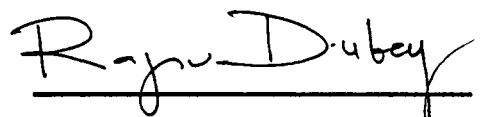


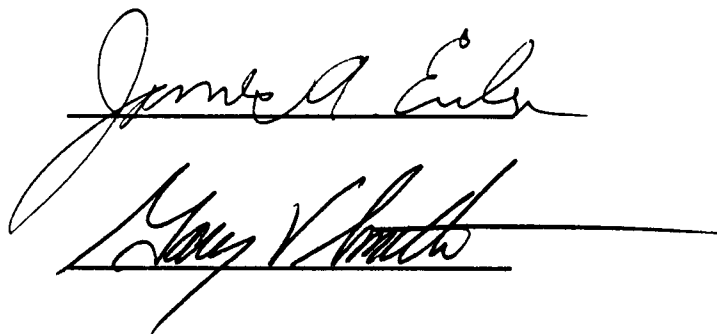
To the Graduate Council:

I am submitting herewith a thesis written by Roy J. Issa entitled "Kinematic Control of Redundant Manipulators for Higher Efficiency and Mechanical Advantage." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mechanical Engineering.

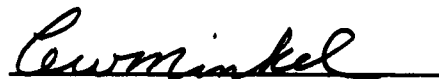


Rajiv V. Dubey, Major Professor

We have read this thesis and
recommend its acceptance:



Accepted for the council:



Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Master of Science degree at the University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotation from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my professor, or in his absence, by the Head of the Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying of use of material in this thesis for financial gain shall not be allowed without my written permission.

Signature

Roy Ino

Date

NOVEMBER 1989

**KINEMATIC CONTROL OF REDUNDANT MANIPULATORS FOR HIGHER
EFFICIENCY AND MECHANICAL ADVANTAGE**

**A Thesis
presented for the
Master of Science
Degree
The University of Tennessee, Knoxville**

Roy J. Issa

December 1989

To the most influential people in my life who made me understand the meaning of life by their own humbleness and sacrifice, and whose love and support to me is like an infinite space and a sea without a shore. To my grandparents who brought so much happiness to my life, to Hanna and Nejme Gabro I dedicate my thesis.

Roy

ACKNOWLEDGMENTS

The research reported here was sponsored by the National Aeronautics And Space Administration, Langley Research Center under contract no. R01-1372-15. I would like to thank NASA for its financial support during the course of my research work.

I would like to thank my committee members, Drs. R. V. Dubey, J. A. Euler, and G. V. Smith who have made themselves readily available to me for consultation and guidance.

I also owe thanks to my parents for their continuous help during my undergraduate and graduate study at the University of Tennessee.

ABSTRACT

The joint torques and the joint velocities of a robot manipulator depend on the joint configuration of the manipulator. In a near singular configuration, the joint velocities required to move the end-effector in certain directions get extremely high. In some configurations, the joint torques required for the end-effector to apply a force along a certain direction may also get extremely high. Redundant manipulators allow the end-effector to reach a desired position and orientation with an infinite number of joint configurations. Thus joint configurations requiring lower joint velocities or torques may be chosen to follow a desired trajectory.

Choosing configurations requiring lower (minimum norm) joint velocities to track a desired end-effector trajectory results in minimum overall movement of the joints. We refer to such manipulator motion as efficient motion. Efficient motion is achieved by maximizing the Manipulator Velocity Ratio (MVR) [2] in the direction of motion, or in the direction in which it is minimum, depending on the nature of the end-effector trajectory. Improving the performance criterion that utilizes the Manipulability Measure [14] also results in low joint velocities in certain directions.

Choosing configurations that require the minimum joint torques for a certain magnitude of external force is also desired in some cases. In order to lower the joint torques, a performance criterion that is based on the Manipulator Mechanical Advantage (MMA) [2] is maximized. Maximizing MMA in the direction of the applied force results in configurations requiring lower joint torques for a desired end-effector force.

The control schemes utilized in this thesis use kinematic and local optimization techniques which are based on the gradient projection method. The control schemes mentioned above are applied to the NASA Laboratory Telerobotic Manipulator (LTM). Simulations were performed using a solid model of the LTM on the Silicon Graphics workstation. The software was written in "C" language for higher speed and portability.

TABLE OF CONTENTS

1 .	INTRODUCTION	1
1.1	BACKGROUND	
1.2	THESIS OBJECTIVES	
1.3	THESIS OUTLINE	
2 .	KINEMATIC CONTROL OF REDUNDANT MANIPULATORS	8
2.1	THE GRADIENT PROJECTION METHOD	
2.2	HOMOGENEOUS AND PARTICULAR SOLUTIONS	
2.3	DETERMINATION OF THE LEAST NORM SOLUTION IN TERMS OF $\dot{\theta}_p$ AND $\dot{\theta}_h$	
2.4	DETERMINATION OF THE NULL SPACE SOLUTION	
2.5	DETERMINATION OF THE SCALAR CONSTANT K	
3 .	EFFICIENCY AND MECHANICAL ADVANTAGE	16
3.1	INTRODUCTION	
3.2	THE STATIC FORCE RELATIONSHIP	
3.3	TRANSFORMATION OF THE KINEMATIC EQUATION AND THE STATIC FORCE RELATIONSHIP	
3.4	MANIPULATOR VELOCITY RATIO	
3.5	GEOMETRIC INTERPRETATION OF THE MANIPULATOR VELOCITY RATIO	
3.6	MATHEMATICAL PROOF FOR THE RANGE IN WHICH MVR OPERATES	
3.7	MVR GRADIENT BASED CONTROL SCHEMES	

- 3.7.1 MVR IN THE DIRECTION OF THE
END-EFFECTOR MOTION
(LINEAR END-EFFECTOR MOTION CASE)
- 3.7.2 MVR IN THE DIRECTION OF THE
END-EFFECTOR MOTION
(ANGULAR END-EFFECTOR MOTION CASE)
- 3.7.3 MVR IN THE DIRECTION OF THE MINIMUM SINGULAR
VALUE (LINEAR END-EFFECTOR MOTION CASE)
- 3.7.4 MVR IN THE DIRECTION OF THE MINIMUM SINGULAR
VALUE (ANGULAR END-EFFECTOR MOTION CASE)
- 3.8 MANIPULATOR MECHANICAL ADVANTAGE
- 3.9 GEOMETRIC INTERPRETATION OF THE MANIPULATOR
MECHANICAL ADVANTAGE
- 3.10 MATHEMATICAL PROOF FOR THE RANGE IN WHICH MMA
OPERATES
- 3.11 MMA GRADIENT BASED CONTROL SCHEMES
 - 3.11.1 MMA BASED ON THE FORCE COMPONENTS
APPLIED AT THE WRIST
 - 3.11.2 MMA BASED ON THE MOMENT COMPONENTS
APPLIED AT THE WRIST
- 3.12 DUALITY OF MVR AND MMA
- 3.13 MANIPULABILITY MEASURE CRITERION

4. CONTROL SCHEME IMPLEMENTATION ON THE NASA LABORATORY TELEROBOTIC MANIPULATOR

48

- 4.1 FORWARD KINEMATICS OF THE LTM
- 4.2 THE WRIST JACOBIAN REFERENCED TO THE BASE COORDINATE
FRAME
- 4.3 THE RELATIONSHIP BETWEEN THE END-EFFECTOR AND THE
WRIST VELOCITIES
- 4.4 INVERSE KINEMATICS OF THE LTM

4.4.1	THE WRIST JACOBIAN REFERENCED TO THE 3 rd COORDINATE FRAME	
4.4.2	THE PARTITIONING OF J_{w3} FOR USE IN THE INVERSE KINEMATIC CONTROL SCHEME	
4.4.3	DETERMINING THE LEAST NORM SOLUTION	
4.4.3.1	ELIMINATING THE FIRST COLUMN	
4.4.3.2	ELIMINATING THE SECOND COLUMN	
4.5	NULL SPACE SOLUTION	
4.6	KINEMATIC CONTROL FOR HIGHER EFFICIENCY	
4.7	KINEMATIC CONTROL FOR HIGHER MECHANICAL ADVANTAGE	
5.	KINEMATIC CONTROL SOFTWARE FOR THE LTM	65
5.1	TIME COORDINATED TRAJECTORIES IN CARTESIAN SPACE	
5.2	GENERATION OF THE EULER ANGLES	
5.3	GENERATION OF THE HAND ERROR AND THE WRIST ERROR	
5.4	GRADIENT OF PERFORMANCE CRITERIA	
5.5	FORWARD POSITION KINEMATICS	
5.6	INVERSE KINEMATICS OF THE LTM	
6.	SIMULATION	80
6.1	MAXIMIZATION OF MVR	
6.1.1	MAXIMIZATION OF MVR IN THE DIRECTION OF MOTION	
6.1.2	MAXIMIZATION OF MVR IN THE DIRECTION CORRESPONDING TO THE MINIMUM SINGULAR VALUE	
6.2	MAXIMIZATION OF MMA	
6.3	THE MINIMUM SINGULAR VALUE AND THE MANIPULABILITY MEASURE	
6.4	EFFICIENT MOTION AND STABILITY	

7. CONCLUSIONS AND RECOMMENDATIONS	111
7.1 CONCLUSIONS	
7.2 RECOMMENDATIONS	
LIST OF REFERENCES	113
APPENDIXES	116
A. THE DENAVIT-HARTENBERG HOMOGENEOUS TRANSFORMATIO MATRICES	
B. DERIVATIONS OF THE POSITION AND THE ORIENTATION OF THE END-EFFECTOR AND THE WRIST w.r.t. THE BASE COORDINATE FRAME	
C. THE WRIST JACOBIAN REFERENCED TO THE BASE FRAME	
D. COMPUTER PROGRAM	
VITA	161

LIST OF FIGURES

Figure	Page
1.1. Cincinnati Milacron T ³ robot arm.	2
1.2. Seven-degree-of-freedom CESAR research manipulator.	3
1.3. Seven-degree-of-freedom NASA Laboratory Telerobotic Manipulator (LTM).	5
3.1. Manipulator Velocity Ratio Ellipsoid.	25
3.2. Manipulator Mechanical Advantage Ellipsoid.	37
4.1. Laboratory Telerobotic Manipulator with coordinate frames associated with every joint.	49
5.1. Starting and stopping motion between successive nodes.	66
5.2. Linear motion between successive nodes.	66
5.3. Path transitions.	67
5.4. Flow chart of the subroutine that determines the desired position of the end-effector.	69
5.5. Time coordinated trajectory consisting of two nodes.	71
5.6. A bubble diagram representing the major routines of the computer software.	73
5.7. The Eulerian angles.	74
5.8. Representation of the current and the desired orientation of the end-effector frame.	76
6.1. MVR in the direction of motion vs. time . (linear end-effector motion case)	81
6.2. MVR in the direction of motion vs. time. (angular end-effector motion case)	83
6.3. MVR in the direction corresponding to the minimum singular value vs. time. (linear end-effector motion case)	84
6.4. MVR in the direction corresponding to the minimum singular value vs. time. (angular end-effector motion case)	86

6.5. MMA vs. time. (for forces applied at the wrist)	87
6.6. MMA vs. time. (for moments applied at the wrist)	88
6.7. Manipulator configuration using the least norm solution for the trajectory in Figure 6.6.	89
6.8. Manipulator configuration using the maximization of MMA in the direction of the wrist force for the trajectory in Figure 6.6.	90
6.9. Minimum singular value vs. time using different control schemes.	92
6.10. Volume of the MVR Ellipsoid vs. time using different control schemes.	94
6.11. Manipulator configuration using the least norm solution for the trajectory in Figure 6.9.	95
6.12. Manipulator configuration using the maximization of the Manipulability Measure for the trajectory in Figure 6.9.	96
6.13. Trajectory motion for case #1, case #2, and case #3.	98
6.14. Total joint motion per cycle for case #1.	99
6.15. Total joint motion per cycle for case #2.	101
6.16. Steady state manipulator configuration cycle using the least norm solution for case #1.	102
6.17. Steady state manipulator configuration cycle using the maximization of MVR in the direction corresponding to the minimum singular value for case #1.	103
6.18. Steady state manipulator configuration cycle using the maximization of MVR in the direction of motion for case #1.	104
6.19. Steady state manipulator configuration cycle using the least norm solution for case #2.	105
6.20. Steady state manipulator configuration cycle using the maximization of MVR in the direction corresponding to the minimum singular value for case #2.	106

6.21. Steady state manipulator configuration cycle using the maximization of MVR in the direction of motion for case #2.	107
6.22. Limiting value for the total joint area per cycle vs. time for case #1.	109
6.23. Limiting value for the total joint area per cycle vs. time for case #2.	109
6.24. Total joint motion per cycle for case #3.	110

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Robot manipulators are computer controlled manipulators consisting of several links connected together either by revolute or prismatic joints. Usually one end of the chain is attached to a supporting base while the other end has a gripper attached to it and is free to move in the work space of the robot. The robot manipulator itself is composed of an arm and a wrist sub-assemblies. For a six-degree-of-freedom robot, such as the Cincinnati Milacron T³ robot (Figure 1.1), the arm sub-assembly can move with three degrees of freedom that will result in positioning the mechanism. The wrist subassembly consists of three rotations referred to as the pitch, yaw, and roll. These rotary motions result in orienting the mechanism. With six degrees of freedom, a robot end-effector can attain any position in the workspace of the robot if no singularities occur within its work space. Three degrees of freedom are required to achieve translational motion and another three degrees of freedom are required to achieve rotational motion in a three dimensional space. Unfortunately, a manipulator invariably reaches singular configurations within its workspace. In such configurations, the joint velocities required to move the end-effector in certain directions get extremely high. Whenever a manipulator encounters a singular configuration, it loses a degree of freedom and becomes unable to move in a certain direction. Therefore, it is highly desirable to use manipulators with more than the minimum number of degrees of freedom required to perform a desired task. Such manipulators are referred to as redundant manipulators.

Most of the redundant manipulators that exist today are confined to research work. Two of these robots are the seven-degree-of-freedom CESAR manipulator at the Oak Ridge National Laboratory (Figure 1.2), and the Laboratory Telerobotic Manipulator (LTM) at the NASA Langley Research Center. The LTM, shown in

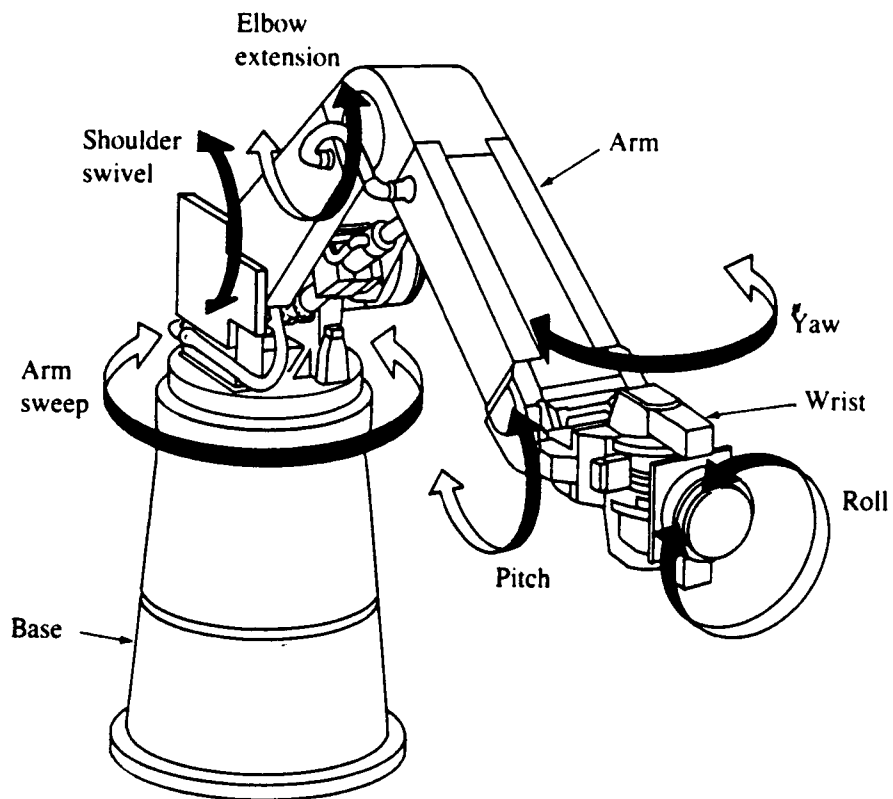


Figure 1.1. Cincinnati Milacron T³ robot arm.

(K.S. FU, R.C. Gonzalez, C.S.G. Lee, "Robotics: Control, Sensing, Vision, and Intelligence", McGraw-Hill Book Company, 1987, p.2)

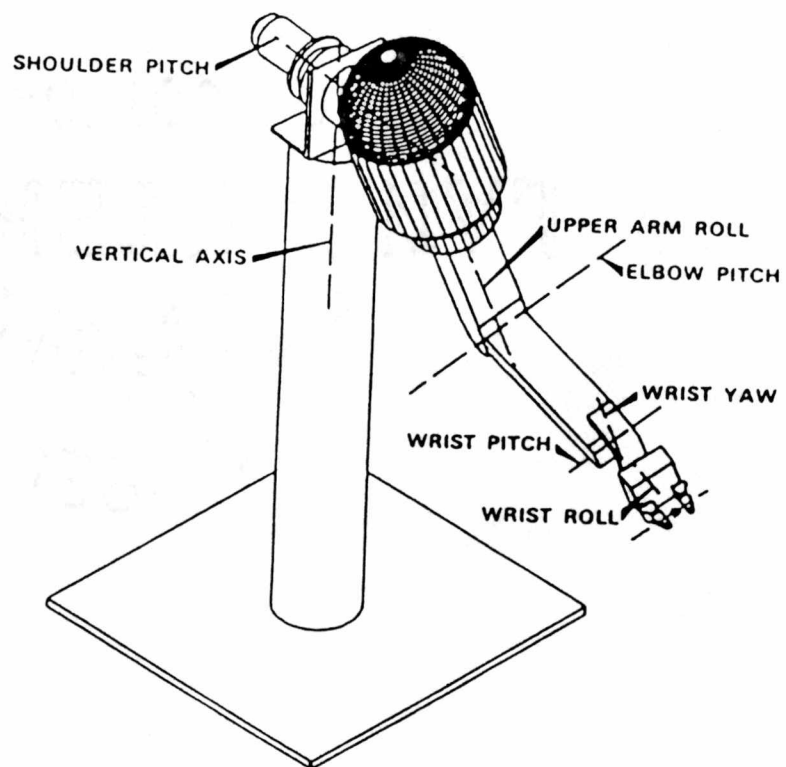


Figure 1.2. Seven-degree-of-freedom CESAR research manipulator.
(Courtesy I&C Division ORNL)

Figure 1.3, is also a seven-degree-of-freedom manipulator. The arm has three pitch/yaw joints that provide the shoulder, the elbow, and the wrist joints. The seventh degree of freedom is provided by the wrist roll motion.

The redundancy of a manipulator plays an important role in increasing its flexibility. Flexibility is defined here as the ability of the manipulator to move in any direction with a minimal encounter in joint singularities. Redundancy occurs when the dimension of the joint space is greater than the dimension of the operational space required to perform a certain task [7]. For redundant manipulators, the kinematic linear equations relating unknown joint velocities to given end-effector velocity components in world coordinates do not have a unique solution. An infinite number of solutions are possible in such a case. Increasing the flexibility of a manipulator requires the utilization of the null space of the Jacobian. The null space of the Jacobian, whose dimension depends on the number of degrees of redundancy of the manipulator, provides the manipulator with infinite number of joint configurations. Since the null space of the Jacobian does not contribute to the motion of the end-effector, it can be used to select a joint configuration that can improve the performance of the manipulator while tracking a certain desired trajectory.

Several performance measures that utilize the null space and improve the performance of the manipulator have been utilized in the past years by several researchers. These performance measures include the Manipulability Measure, $\sqrt{\det(\mathbf{J}\mathbf{J}^T)}$ [14], the Manipulator Velocity Ratio (MVR) and the Manipulator Mechanical Advantage (MMA) [2]. Luh and Gu [8] used the $\text{trace}(\mathbf{J}\mathbf{J}^T)$ to study the efficiency and the flexibility of a redundant manipulator. Paul and Stevenson [11] used the determinant of $\mathbf{J}$ as a performance measure for non-redundant wrists.

Optimization of performance measures can be based on either the global or the local performance, and on either the kinematic or the dynamic performance [13]. Local optimization may not result in a better overall performance for the desired trajectory, while on the other hand global optimization may provide a better solution in terms of overall performance. However, local optimization schemes are preferred to global optimization schemes due to being less computationally complex

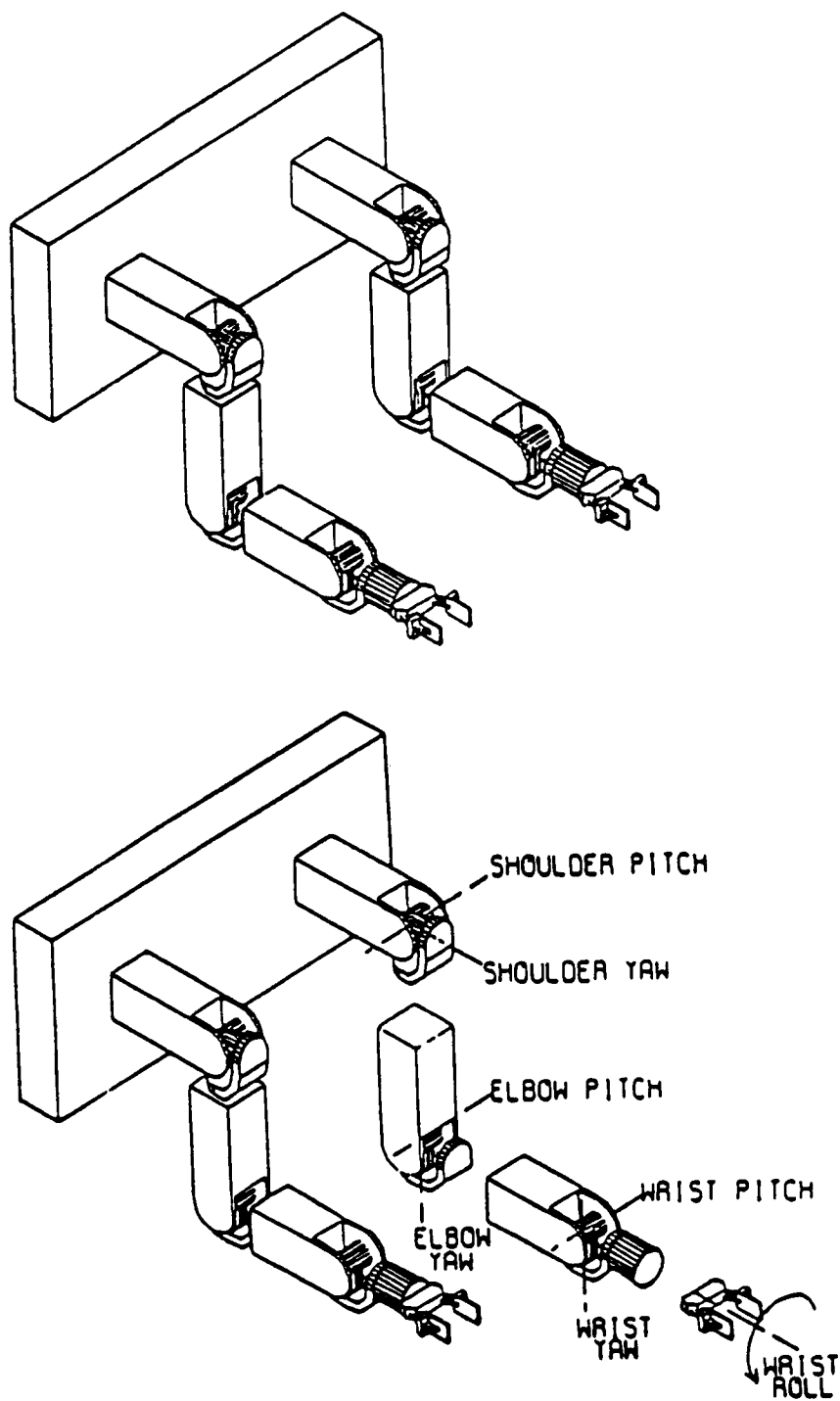


Figure 1.3. Seven-degree-of-freedom NASA Laboratory Telerobotic Manipulator (LTM) (Courtesy I&C Division ORNL).

and faster to operate in real time. Dynamic optimization considers the arm dynamics in the process in order to minimize the required joint torques[6], while static optimization considers only the arm kinematics. Dynamic optimization can be used to optimize the kinetic energy and achieve load balance, while kinematic optimization is used for handling important issues such as avoiding singularities, improving the flexibility of the arm, and avoiding collision with obstacles.

1.2 THESIS OBJECTIVES

The joint velocities required to translate and rotate the end-effector with a given speed depend on the direction of motion as well as the orientation of the end-effector. Different arm configurations result in different joint velocities. However, it is always desirable to choose those joint configurations that require the least joint velocities. Choosing joint configurations that require lower joint velocities to track a desired end-effector trajectory results in the minimum overall movement of the joints. We refer to such manipulator motion as efficient motion. On the other hand, an infinite number of joint torques can result in providing a unique force vector applied by the end-effector on the environment. Choosing joint configurations that result in lower joint torques can be achieved by utilizing the redundancy of the manipulator.

Three performance criteria that improve the performance of the LTM are utilized in this thesis. These are: The Manipulator Velocity Ratio, The Manipulator Mechanical Advantage, and the Manipulability Measure criterion. The Manipulator Velocity Ratio (MVR) is defined as the norm of the transformed end-effector velocity vector to the norm of the least norm transformed joint velocity vector [2]. Control schemes that improve MVR in both the direction of motion and in the direction of the minimum singular value of the Manipulator Velocity Ratio Ellipsoid are applied on the LTM. The Manipulator Mechanical Advantage (MMA) is defined as the ratio of the end-effector transformed force norm to the norm of the transformed joint torque vector. Control schemes that deal with improving MMA in the direction of the applied force are also utilized. The Manipulability Measure

criterion that can be used in a control scheme to avoid singularities in all possible directions is also applied to the LTM. This performance criterion uses the control scheme introduced by Yoshikawa. A comparison between these control schemes that utilize the null space of the Jacobian along with the least norm scheme for different performance measures is presented to reflect the feasibility of these control schemes for certain tasks. These schemes use local optimization techniques and the gradient projection method. By simulation, we will also investigate an important issue concerning the stability of the above control schemes together with the least norm scheme.

1.3 THESIS OUTLINE

Chapter 2 presents a kinematic control of redundant manipulators. Chapter 3 presents a control scheme for improving the efficiency and the mechanical advantage of redundant manipulators. Chapter 4 deals with a control scheme implementation on the NASA Laboratory Telerobotic Manipulator. The kinematic control software is discussed in chapter 5. Chapter 6 contains computer simulations that demonstrate the effectiveness of the control schemes discussed in chapter 4, and chapter 7 presents conclusions and recommendations.

CHAPTER 2

KINEMATIC CONTROL OF REDUNDANT MANIPULATORS

This chapter presents an efficient gradient projection scheme[3] for a seven-degree-of-freedom manipulator with a spherical wrist. This scheme avoids finding the Moore-Penrose inverse of the 6x7 Jacobian and is well suited for real time implementation. This chapter will show that using this scheme, a closed form solution for the joint velocities can be obtained.

2.1 THE GRADIENT PROJECTION METHOD

The kinematic equation relating the end-effector velocity vector $\dot{\mathbf{x}}$ (dimensioned as $m \times 1$, where $m \leq 6$) to the joint velocity vector $\dot{\mathbf{q}}$ ($n \times 1$) is given by the following equation:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}, \quad (2.1)$$

where $\mathbf{J}$ is an $m \times n$ Jacobian matrix. If $m = n$ (i.e. $\mathbf{J}$ is a square matrix) and all columns or rows of $\mathbf{J}$ are linearly independent, then the joint velocities required to achieve a certain end-effector velocity are unique. The joint velocity vector will be given by:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}\dot{\mathbf{x}}. \quad (2.2)$$

If $m < n$ (i.e. $\mathbf{J}$ is rectangular), then numerous joint velocity vectors exist for a desired end-effector velocity. The joint velocity vector can then be computed as:

$$\dot{\mathbf{q}} = \mathbf{J}^+\dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}^+\mathbf{J})\dot{\mathbf{q}}, \quad (2.3)$$

where $\mathbf{J}^+$ is the Moore-Penrose inverse of the Jacobian. Since $\mathbf{J}$ has a rank $\leq m$ and since $m < n$, this means that $\dot{\mathbf{q}}$ which is an $n \times 1$ column vector has components

in two subspaces, namely the row space and the null space. Recall from linear algebra that any space can be divided into two orthogonal complements. The row space of $\mathbf{J}$ has a dimension equal to the $\text{rank}(\mathbf{J})$, and the null space of $\mathbf{J}$ has a dimension equal to $(n - \text{rank}(\mathbf{J}))$. Therefore, $\dot{\underline{\theta}}$ has a component in the row space of $\mathbf{J}$ and another in its null space. The first component in equation (2.3), $\mathbf{J}^+ \dot{\underline{x}}$, represents the projection of $\dot{\underline{x}}$ onto the row space of $\mathbf{J}$. This component is called the least norm solution of $\dot{\underline{\theta}}$ because it is the minimum solution to $\dot{\underline{x}} = \mathbf{J} \dot{\underline{\theta}}$. The second component, $(\mathbf{I} - \mathbf{J}^+ \mathbf{J}) \dot{\underline{\phi}}$, which is the orthogonal complement of $\mathbf{J}^+ \dot{\underline{x}}$, lies in the null space of $\mathbf{J}$ and is referred to as the null space solution since it does not contribute anything to the motion of the end-effector. The vector $\dot{\underline{\phi}}$ is an arbitrary velocity vector, and $(\mathbf{I} - \mathbf{J}^+ \mathbf{J})$ is a projection matrix that projects $\dot{\underline{\phi}}$ onto the null space of $\mathbf{J}$. Since $\dot{\underline{\phi}}$ can be any arbitrary vector, it is often appropriately selected to improve the performance of the robot arm. Let $H(\theta)$ represents a performance criterion which is a function of the joint angles. Using the gradient projection method, this criterion can be improved by substituting $K \nabla H(\theta)$ for $\dot{\underline{\phi}}$. Equation (2.3) will then become:

$$\dot{\underline{\theta}} = \mathbf{J}^+ \dot{\underline{x}} + K(\mathbf{I} - \mathbf{J}^+ \mathbf{J}) \nabla H \quad , \quad (2.4)$$

where the coefficient K is a real number and $\nabla H(\theta)$ is the gradient of $H(\theta)$. The gradient of $H(\theta)$ is computed as follows:

$$\nabla H = \left[\frac{\partial H}{\partial \theta_1} \quad \frac{\partial H}{\partial \theta_2} \quad \dots \quad \frac{\partial H}{\partial \theta_n} \right]^T \quad . \quad (2.5)$$

Since the kinematic control scheme presented in this chapter applies to seven-degree-of-freedom manipulators, $\dot{\underline{\theta}}$ is a 7x1 column vector. Let the end-effector velocity vector be an $m \times 1$ vector which with its maximum components can be represented by:

$$\dot{\underline{x}} = \begin{bmatrix} v_{h,x} & v_{h,y} & v_{h,z} & \omega_{h,x} & \omega_{h,y} & \omega_{h,z} \end{bmatrix}^T, \quad (2.6)$$

where the first three components of this vector represent the linear velocity of the end-effector (hand) and the other three components represent the angular velocity of the end-effector. The Jacobian used in this case will then be a 6x7 matrix.

If we assume that the rank of $\mathbf{J}$ is 6, then there exist six independent column vectors in $\mathbf{J}$. If these independent columns are grouped together separating the dependent column from the other six independent columns, then $\mathbf{J}$ can be partitioned as:

$$\mathbf{J} = \begin{bmatrix} \alpha & \mathbf{J}^* \end{bmatrix}, \quad (2.7)$$

where $\mathbf{J}^*$ consists of the six independent columns of $\mathbf{J}$ and α represents the dependent column of $\mathbf{J}$. If we rearrange the elements of $\dot{\underline{\theta}}$ with the redistribution of the columns of $\mathbf{J}$, the kinematic equation of motion can be written as:

$$\dot{\underline{x}} = \begin{bmatrix} \alpha & \mathbf{J}^* \end{bmatrix} \dot{\underline{\theta}}. \quad (2.8)$$

2.2 HOMOGENEOUS AND PARTICULAR SOLUTIONS

The joint velocity vector $\dot{\underline{\theta}}$ can be represented as a combination of a particular solution ($\dot{\underline{\theta}}_p$) and a homogeneous solution ($\dot{\underline{\theta}}_h$):

$$\dot{\underline{\theta}} = \dot{\underline{\theta}}_p + \mathbf{K} \dot{\underline{\theta}}_h. \quad (2.9)$$

Both the particular and the homogeneous vectors are of dimension 7x1. The particular solution can be determined by choosing the first element in this column

vector as zero, and the remaining elements can be obtained from equation (2.8). $\dot{\underline{\theta}}_p$ can be shown to be:

$$\dot{\underline{\theta}}_p = \begin{bmatrix} 0 \\ J^{*-1} \dot{\underline{x}} \end{bmatrix} . \quad (2.10)$$

The homogeneous solution can be obtained by arbitrarily choosing the first element as 1, and the remaining elements of the homogeneous solution can be obtained by solving for $\dot{\underline{\theta}}_h$ in the following equation:

$$\begin{bmatrix} \alpha J^* \end{bmatrix} \dot{\underline{\theta}}_h = 0 . \quad (2.11)$$

The homogeneous solution for $\dot{\underline{\theta}}$ can then be shown as:

$$\dot{\underline{\theta}}_h = \begin{bmatrix} 1 \\ -J^{*-1} \alpha \end{bmatrix} . \quad (2.12)$$

Substituting equations (2.10) and (2.12) into (2.9), we obtain the following:

$$\dot{\underline{\theta}} = \begin{bmatrix} K \\ J^{*-1}(\dot{\underline{x}} - K\alpha) \end{bmatrix} . \quad (2.13)$$

2.3 DETERMINATION OF THE LEAST NORM SOLUTION IN TERMS OF $\dot{\underline{\theta}}_p$ AND $\dot{\underline{\theta}}_h$

Let us now derive a relationship between the least norm solution, $\dot{\underline{\theta}}_{min}$, and the particular and homogeneous solutions for $\dot{\underline{\theta}}$. From equation (2.9), the square of

the norm of $\dot{\underline{\theta}}$ can be expressed as:

$$\begin{aligned} \|\dot{\underline{\theta}}\|^2 &= \|\dot{\underline{\theta}}_p + K\dot{\underline{\theta}}_h\|^2 \\ &= (\dot{\underline{\theta}}_p + K\dot{\underline{\theta}}_h)^T (\dot{\underline{\theta}}_p + K\dot{\underline{\theta}}_h) . \end{aligned} \quad (2.14)$$

Since we are interested in getting the minimum value of $\dot{\underline{\theta}}$ (i.e. the least norm), all what we need to do is to take the partial derivative of the above equation with respect to K , set the resulting equation equal to zero, solve for K , and then determine $\dot{\underline{\theta}}_{\min}$ based on that value of K . K can then be expressed as:

$$K = -\dot{\underline{\theta}}_p^T \dot{\underline{\theta}}_h / \dot{\underline{\theta}}_h^T \dot{\underline{\theta}}_h . \quad (2.15)$$

Substituting for K in equation (2.9), the least norm solution can be expressed as:

$$\dot{\underline{\theta}}_{\min} = \dot{\underline{\theta}}_p - [\dot{\underline{\theta}}_p^T \dot{\underline{\theta}}_h / \dot{\underline{\theta}}_h^T \dot{\underline{\theta}}_h] \dot{\underline{\theta}}_h . \quad (2.16)$$

Substituting partially for $\dot{\underline{\theta}}_p$ and $\dot{\underline{\theta}}_h$ in the above equation, we can show that:

$$\dot{\underline{\theta}}_{\min} = \begin{bmatrix} -\dot{\underline{\theta}}_p^T \dot{\underline{\theta}}_h / \dot{\underline{\theta}}_h^T \dot{\underline{\theta}}_h \\ J^{*-1} \{ \dot{\underline{x}} - [\dot{\underline{\theta}}_p^T \dot{\underline{\theta}}_h / \dot{\underline{\theta}}_h^T \dot{\underline{\theta}}_h] \alpha \} \end{bmatrix} . \quad (2.17)$$

2.4 DETERMINATION OF THE NULL SPACE SOLUTION

Having found the least norm solution, we still need to determine the null space solution (the second term in equation (2.4)). In this section we will show that the null space solution is the projection of the gradient of the performance criterion

$H(\theta)$ onto the null space of the Jacobian. Chapter 3 presents the gradient of several performance criteria. According to equation (2.4), let the null space solution be given as:

$$K(I - J^+J) \nabla H = K[\nabla H^T \dot{\theta}_h / \dot{\theta}_h^T \dot{\theta}_h] \dot{\theta}_h \quad (2.18)$$

It can be proven that $(I - J^+J)$ is a projection matrix that projects any vector onto the null space of J . Let us rename this projection matrix as P . In order for P to be a projection matrix, it has to satisfy the two following properties [12]:

- (i) It equals its square: $P^2 = P$.
- (ii) It equals its transpose: $P^T = P$.

The above conditions can be shown to be true for $P = I - J^+J$. We still need to show that P projects vectors onto the null space of J . Recall that the least norm solution for $\dot{\theta}$, $J^+ \dot{x}$, lies in the row space of J . Recall also that the only space orthogonal to the row space is the null space. Therefore, if we try to project the least norm solution onto the null space of J we should get a zero for an answer. If we can prove that using the projection matrix P , then P is indeed a projection matrix that projects vectors onto the null space of J . The projection of the least norm solution using P , is represented as:

$$\begin{aligned} P(J^+ \dot{x}) &= (I - J^+J)(J^+ \dot{x}) \\ &= J^+ \dot{x} - J^+J J^+ \dot{x} \\ &= J^+ \dot{x} - J^+ \dot{x} = 0 \end{aligned} \quad (2.19)$$

where $J^+J J^+$ can be shown equal to J^+ [12]. This proves that $(I - J^+J)$ is a projection matrix that project vectors onto the null space of J . Since $\dot{\theta}$ can be decomposed into a homogeneous and a particular solution, the direction in which the null space solution must lie is the direction of the homogeneous solution (because the direction of the least norm solution is orthogonal to the direction of the

homogeneous solution). Therefore equation (2.18) is valid.

2.5 DETERMINATION OF THE SCALER CONSTANT K

The coefficient K in equation (2.4) was mentioned to be a scaler constant whose value is limited by the bounds on the joint velocities. In the following section, we will implement a scheme, proposed by Euler et al. [4], for determining the scaler constant K . The joint velocity vector of a redundant robot consists of a least norm (min) part and a homogeneous part, written as:

$$\dot{\underline{\theta}} = \dot{\underline{\theta}}_{\min} + \dot{\underline{\theta}}_h \quad (2.20)$$

Let the upper and lower bounds on the i^{th} joint velocity be $\dot{\underline{\theta}}_{\max,i}$ and $-\dot{\underline{\theta}}_{\max,i}$ respectively, and the upper and lower limits on K be $K_{\max,i}$ and $K_{\min,i}$ which are determined as follows:

$$K_{\max,i} = \max \left((\dot{\underline{\theta}}_{\max,i} - \dot{\underline{\theta}}_{\min,i}) / \dot{\underline{\theta}}_{h,i} , (-\dot{\underline{\theta}}_{\max,i} - \dot{\underline{\theta}}_{\min,i}) / \dot{\underline{\theta}}_{h,i} \right) \quad (2.21)$$

and

$$K_{\min,i} = \min \left((\dot{\underline{\theta}}_{\max,i} - \dot{\underline{\theta}}_{\min,i}) / \dot{\underline{\theta}}_{h,i} , (-\dot{\underline{\theta}}_{\max,i} - \dot{\underline{\theta}}_{\min,i}) / \dot{\underline{\theta}}_{h,i} \right) \quad (2.22)$$

For the joint velocities of a redundant robot of n joints to be within bounds, the upper and the lower bounds on the scaler constant K should be determined as follows:

$$K_{\max} = \min (K_{\max,1}, K_{\max,2}, \dots, K_{\max,n}) \quad (2.23)$$

and

$$K_{\min} = \max (K_{\min,1}, K_{\min,2}, \dots, K_{\min,n}) \quad (2.24)$$

If $K_{\max}$ is chosen for K , then the criterion $H(\theta)$ will be maximized, otherwise; if $K_{\min}$ is chosen for K , then $H(\theta)$ will be minimized. In this case $K_{\max}$ is greater than $K_{\min}$. If $K_{\max}$ is less than $K_{\min}$, then a value of K does not

exist such that the joint velocities are within their bounds. In this case, even the least norm solution results in high joint velocities.

CHAPTER 3

EFFICIENCY AND MECHANICAL ADVANTAGE

3.1 INTRODUCTION

The joint velocities required to move the end-effector of a robot manipulator with a desired velocity depend on the manipulator configuration. Similarly, the joint torques required for the end-effector to apply a certain force while tracking a prescribed trajectory also depend on the manipulator configuration. Choosing configurations that require the minimal joint velocities results in the minimum overall movement of the joints. Such motion is referred to as efficient motion. On the other hand, choosing configurations that require the minimum joint torques for a certain magnitude of external force is also desired in some cases.

In order to minimize the joint velocities of a manipulator, we can improve the Manipulator Velocity Ratio (MVR) [2] in either the direction of motion of the end-effector or in the direction corresponding to the minimum singular value. Improving the performance criterion that utilizes the Manipulability Measure [14] also results in low joint velocities. In order to lower the joint torques, a performance measure such as the Manipulator Mechanical Advantage (MMA) [2] can be improved. This chapter deals with the derivation of the control schemes that are based on the above performance criteria.

As we mentioned before, the kinematic equation required to describe the relationship between the end-effector velocity and the joint velocities is given by the following:

$$\dot{\mathbf{x}} = \mathbf{J} \dot{\mathbf{q}} \quad , \quad (3.1)$$

where $\dot{\mathbf{x}}$ is an $m \times 1$ vector which at most consists of three translational and three rotational velocities which are V_x, V_y, V_z , and $\omega_x, \omega_y, \omega_z$ respectively. $\dot{\mathbf{q}}$ is an $n \times 1$ vector consisting of all the n joint velocities; $\dot{\theta}_1, \dot{\theta}_2, \dots, \dot{\theta}_n$. $\mathbf{J}$ is an $m \times n$

matrix called the Jacobian, and whose elements are given by $J_{i,j} = \partial x_i / \partial \theta_j$.

3.2 THE STATIC FORCE RELATIONSHIP

Having defined the kinematic relationship between the end-effector and the n joints of a redundant manipulator, let us define another relationship between the actuator drive torques and the resultant force and moment applied at the manipulator end-effector. In this case, our focus will be upon the forces and moments which act on the manipulator arm when it is at rest. Remember that each joint of the manipulator is driven by an individual actuator. The torque inputs from each of these actuators is transmitted through the arm linkages to the end-effector where the resultant force and moment act on the surrounding environment. The relationship between these joint torques and the force and moment acting on the end-effector of the arm can be derived from the principle of virtual work [1]. Recall that the manipulator Jacobian relating the infinitesimal joint displacements $\partial \theta$ to the infinitesimal end-effector displacement ∂x is given by:

$$\partial x = J \partial \theta \quad . \quad (3.2)$$

Considering the virtual displacements at each individual joint and at the end-effector, the virtual work done by the joint torques, and the end-effector forces and moments can be shown as:

$$\partial_{\text{work}} = \underline{\tau}^T \partial \theta - \underline{F}^T \partial x \quad , \quad (3.3)$$

where $\underline{\tau}$ is an $n \times 1$ vector consisting of the joint torques applied by the joint actuators. These torques are $\tau_1, \tau_2, \dots, \tau_n$. We will call $\underline{F}$ the end-effector force and moment vector. It is an $m \times 1$ vector which at most consists of three force terms (f_x, f_y, f_z) and three moment terms (m_x, m_y, m_z). J is the $m \times n$ manipulator Jacobian defined as above.

The principle of virtual work states that in order for the arm linkage to be at

equilibrium, the virtual work should vanish for any arbitrary joint and end-effector displacements. Substituting equation (3.2) into equation (3.3) we get:

$$\partial_{\text{work}} = (\underline{\tau} - \mathbf{J}^T \underline{F}) \partial \theta . \quad (3.4)$$

So whatever $\partial \underline{\theta}$ may be, ∂_{work} must vanish. For this to happen, the following equation should be satisfied:

$$\underline{\tau} = \mathbf{J}^T \underline{F} . \quad (3.5)$$

3.3 TRANSFORMATION OF THE KINEMATIC EQUATION AND THE STATIC FORCE RELATIONSHIP

One thing we need to deal with when using the above equations is that we do not have a consistency in units. For instance, the $\dot{\underline{x}}$ vector has linear velocities and angular velocities. The linear and angular velocities obviously have different units. The same thing applies to the force vector $\underline{F}$. In addition to that we may also have different units for different entries. Each element of the joint velocity vector $\dot{\underline{\theta}}$, the joint torque vector $\underline{\tau}$, the end-effector velocity vector $\dot{\underline{x}}$, or the end-effector force vector $\underline{F}$ may have different units. Some elements of the $\dot{\underline{\theta}}$ and $\underline{\tau}$ vectors may have a different range of motion when compared with other elements of the same vector. As a result, we may want to restrict the motion of certain joints more than the others, or we may be more interested in the end-effector force in a certain direction than others. As a result of this, we need to introduce weight factors associated with each entry of the following vectors: $\dot{\underline{x}}$, $\dot{\underline{\theta}}$, $\underline{F}$, and $\underline{\tau}$. Taking this into account, the norms of the above vectors will then have to be modified as follows:

$$|\dot{\underline{x}}|_{w_x} = \sqrt{\dot{\underline{x}}^T \mathbf{w}_x \dot{\underline{x}}} , \quad |\dot{\underline{\theta}}|_{w_\theta} = \sqrt{\dot{\underline{\theta}}^T \mathbf{w}_\theta \dot{\underline{\theta}}} , \quad (3.6)$$

$$|\underline{F}|_{w_f} = \sqrt{\underline{F}^T w_f \underline{F}} \quad , \quad |\underline{\tau}|_{w_\tau} = \sqrt{\underline{\tau}^T w_\tau \underline{\tau}} \quad , \quad (3.7)$$

where $w_x \in R^{m \times m}$, $w_\theta \in R^{n \times n}$, $w_f \in R^{m \times m}$, and $w_\tau \in R^{n \times n}$ are symmetric positive definite weighting matrices. Let us introduce the following transformations:

$$\underline{J}_v = w_x^{1/2} \underline{J} w_\theta^{-1/2}, \quad \dot{\underline{x}}_v = w_x^{1/2} \dot{\underline{x}}, \text{ and } \dot{\underline{\theta}}_v = w_\theta^{1/2} \dot{\underline{\theta}} \quad . \quad (3.8)$$

Substituting the above transformation equations in the kinematic equation we get the following:

$$\dot{\underline{x}}_v = \underline{J}_v \dot{\underline{\theta}}_v \quad . \quad (3.9)$$

This represents the new form of the kinematic equation with weighting factors associated with each of its constituents. The norms of the transformed end-effector velocity and the transformed joint velocity vectors will then become:

$$|\dot{\underline{x}}|_{w_x} = \sqrt{\dot{\underline{x}}_v^T \dot{\underline{x}}_v} \quad , \text{ and } |\dot{\underline{\theta}}|_{w_\theta} = \sqrt{\dot{\underline{\theta}}_v^T \dot{\underline{\theta}}_v} \quad . \quad (3.10)$$

Similarly if we introduce the following transformed matrices:

$$\underline{J}_m = w_f^{-1/2} \underline{J} w_\tau^{1/2}, \quad \underline{F}_m = w_f^{1/2} \underline{F}, \text{ and } \underline{\tau}_m = w_\tau^{1/2} \underline{\tau} \quad , \quad (3.11)$$

then the equation relating the static end-effector force to the joint torques will be modified as follows:

$$\underline{\tau}_m = \underline{J}_m^T \underline{F}_m \quad , \quad (3.12)$$

where the transformed norms of the force and torque vectors can be shown as:

$$|\underline{F}|_{w_f} = \sqrt{\underline{F}_m^T \underline{F}_m} \quad , \text{ and } |\underline{\tau}|_{w_\tau} = \sqrt{\underline{\tau}_m^T \underline{\tau}_m} \quad . \quad (3.13)$$

3.4 MANIPULATOR VELOCITY RATIO

For a single-input-single-output mechanism such as the 4-bar mechanism, the velocity ratio is defined as the ratio of output velocity to the ratio of input velocity. This concept can be extended to a seven-degree-of-freedom robot arm as the ratio of the norm of the transformed end-effector velocity vector to the norm of the transformed joint velocities. For a multiple degrees of freedom robot, we call this ratio as the Manipulator Velocity Ratio (MVR)[2]. Let r_v denotes MVR which can be written as:

$$r_v = \|\dot{\underline{x}}\|_{w_x} / \|\dot{\underline{\theta}}\|_{w_\theta} , \quad (3.14)$$

where $\dot{\underline{x}}_{w_x}$ is the transformed end-effector velocity in whose direction the manipulator velocity ratio is obtained, and $\dot{\underline{\theta}}_{w_\theta}$ is the transformed joint velocity vector. Substituting equation (3.10) into equation (3.14), we get:

$$r_v = \sqrt{\dot{\underline{x}}_v^T \dot{\underline{x}}_v / \dot{\underline{\theta}}_v^T \dot{\underline{\theta}}_v} . \quad (3.15)$$

Since $\dot{\underline{\theta}}_v$ consists of both a homogeneous and a particular part, an infinite number of $\dot{\underline{\theta}}_v$ can result for a unique end-effector velocity. In order to show the validity of this, recall that the kinematic equation of motion is:

$$\dot{\underline{x}} = \mathbf{J} \dot{\underline{\theta}} . \quad (3.16)$$

This can also be written as:

$$\begin{aligned} \dot{\underline{x}} &= \mathbf{J} (\dot{\underline{\theta}}_n + \dot{\underline{\theta}}_1) \\ &= \mathbf{J} \dot{\underline{\theta}}_n + \mathbf{J} \dot{\underline{\theta}}_1 \\ &= \mathbf{J} \dot{\underline{\theta}}_1 \\ \Rightarrow \dot{\underline{\theta}}_1 &= \mathbf{J}^+ \dot{\underline{x}} , \end{aligned} \quad (3.17)$$

$$(\mathbf{J} \dot{\underline{\theta}}_n = 0.) \quad (3.18)$$

where $\dot{\underline{\theta}}_n$ is the null space solution for $\dot{\underline{\theta}}$, and $\dot{\underline{\theta}}_1$ is the least norm solution for $\dot{\underline{\theta}}$. This means if we have a least norm solution for $\dot{\underline{\theta}}$ and whatever the null space solution for $\dot{\underline{\theta}}$ may be, we will always end up with the same end-effector velocity because the null space solution does not contribute anything to the end-effector velocity. Thus, the manipulator velocity ratio will be unique if $\dot{\underline{\theta}}_v$ is exchanged with $\dot{\underline{\theta}}_1$. So as a result we have to redefine the manipulator velocity ratio as the norm of the transformed end-effector velocity vector to the norm of the least norm transformed joint velocity vector. r_v can be rewritten as:

$$r_v = \sqrt{\dot{\underline{x}}_v^T \dot{\underline{x}}_v / \dot{\underline{\theta}}_1^T \dot{\underline{\theta}}_1} \quad (3.19)$$

Substituting equation (3.17) after multiplying its parts by the weighting matrices into equation (3.19), we get:

$$r_v = \sqrt{\dot{\underline{x}}_v^T \dot{\underline{x}}_v / (\mathbf{J}_v^+ \dot{\underline{x}}_v)^T (\mathbf{J}_v^+ \dot{\underline{x}}_v)} \quad (3.20)$$

where $\mathbf{J}_v^+$ is the Moore psuedo-inverse of $\mathbf{J}_v$. The above equation shows that the MVR depends on the robot joint configuration as well as the transformed end-effector velocity. In a more compact form, equation (3.20) can be written as:

$$r_v = \sqrt{\dot{\underline{x}}_v^T \dot{\underline{x}}_v / \dot{\underline{x}}_v^T (\mathbf{J}_v \mathbf{J}_v^T)^+ \dot{\underline{x}}_v} \quad (3.21)$$

Equation (3.21) is valid because it can be shown that $(\mathbf{J}_v^+)^T = (\mathbf{J}_v^T)^+$ [12]. Let us define $\underline{u}_v$ as a unit vector in the direction of motion of the end-effector. It is represented as:

$$\underline{u}_v = \dot{\underline{x}}_v / \sqrt{\dot{\underline{x}}_v^T \dot{\underline{x}}_v} \quad . \quad (3.22)$$

Substituting equation (3.22) into equation (3.21), we get:

$$r_v = 1 / \sqrt{\underline{u}_v^T (\mathbf{J}_v \mathbf{J}_v^T)^+ \underline{u}_v} \quad . \quad (3.23)$$

3.5 GEOMETRIC INTERPRETATION OF THE MANIPULATOR VELOCITY RATIO

Our prime objective in this section is to prove that the surface formed by the MVR in every possible direction of motion is an ellipsoid of m dimensions. Also recall that $\mathbf{J}_v$ is defined as an $m \times n$ matrix (where $m < n$), and $\underline{u}_v$ as an $m \times 1$ vector. In matrix algebra, any $m \times n$ matrix $\mathbf{A}$ can be decomposed as:

$$\mathbf{A} = \mathbf{Q}_1 \Sigma \mathbf{Q}_2^T \quad , \quad (3.24)$$

where $\mathbf{Q}_1$ ($m \times m$) and $\mathbf{Q}_2$ ($n \times n$) are orthogonal matrices whose columns constitute the eigenvectors of $\mathbf{A} \mathbf{A}^T$ and $\mathbf{A}^T \mathbf{A}$ respectively. Σ is an $m \times n$ diagonal matrix that consists of the square roots of the non zero eigenvalues of both $\mathbf{A} \mathbf{A}^T$ and $\mathbf{A}^T \mathbf{A}$ [12]. These diagonal entries symbolized as $\sigma_1, \dots, \sigma_m$ ($\sigma_1 > \sigma_2 > \dots > \sigma_m$) are also referred to as the singular values. This decomposition is called the Singular Value Decomposition (SVD). If we apply this SVD decomposition technique to the $m \times n$ transformed Jacobian, we can decompose it as:

$$\mathbf{J}_v = \mathbf{U}_v \Sigma_v \mathbf{V}_v^T \quad , \quad (3.25)$$

where Σ_v is a rectangular diagonal matrix expressed as [12]:

$$\Sigma_v = \begin{bmatrix} \sigma_1 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \sigma_2 & & & & \cdot \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ 0 & \cdot & \cdot & \cdot & 0 & \sigma_m & 0 \end{bmatrix} . \quad (3.26)$$

U_v , Σ_v , and V_v are defined in a fashion similar to Q_1 , Σ , and Q_2 respectively. Since the number of columns of the Jacobian is more than the number of rows, the real inverse of the Jacobian can not be found. Instead, another inverse referred to as the psuedo-inverse can always be found. The psuedo-inverse of the Jacobian can be written as:

$$J_v^+ = V_v \Sigma_v^+ U_v^T , \quad (3.27)$$

where Σ_v^+ is an $n \times m$ diagonal matrix whose diagonal entries are the reciprocal of $\sigma_1, \dots, \sigma_m$. Squaring equation (3.23) and substituting equations (3.25) and (3.27) into it, the following can be shown:

$$1/r_v^2 = \underline{u}_v^T U_v (\Sigma_v \Sigma_v^T)^+ U_v^T \underline{u}_v . \quad (3.28)$$

The above equation can also be written as:

$$1 = (r_v \underline{u}_v^T) U_v (\Sigma_v \Sigma_v^T)^+ U_v^T (r_v \underline{u}_v) , \quad (3.29)$$

where $\Sigma_v \Sigma_v^T$ is an $m \times m$ diagonal matrix with the following diagonal entries: $\sigma_1^2, \dots, \sigma_m^2$. Recall that U_v and $\underline{u}_v$ have the dimensions of $m \times m$ and $m \times 1$ respectively. Let us define a vector $\underline{y}$ as follows:

$$\underline{y} = U_v^T r_v \underline{u}_v . \quad (3.30)$$

Therefore,

$$\underline{y}^T = \underline{r}_v \underline{u}_v^T \underline{U}_v \quad . \quad (3.31)$$

Substituting equations (3.30) and (3.31) into (3.29), it can be shown that equation (3.29) becomes:

$$1 = \underline{y}^T (\underline{\Sigma}_v \underline{\Sigma}_v^T)^+ \underline{y} \quad . \quad (3.32)$$

Equation (3.32) can also be written as:

$$1 = (1 / \sigma_1^2) y_1^2 + (1 / \sigma_2^2) y_2^2 + \dots + (1 / \sigma_m^2) y_m^2 \quad . \quad (3.33)$$

Geometrically $\underline{y}$ represents the projection of $\underline{u}_v$ onto the subspace R^m spanned by the m columns of $\underline{U}_v$ with respect to the m basis vectors of $\underline{U}_v$. Each element of $\underline{y}$, call it y_i , represents the component of $\underline{u}_v$ along the i^{th} column of $\underline{U}_v$. That is, y_i (for $i = 1, \dots, m$) lies in the direction of $\underline{u}_i$ (where $\underline{u}_i$ is the i^{th} column of $\underline{U}_v$). The more interesting fact about equation (3.33) is that geometrically it represents an ellipsoid in an m -dimensional space. The minor axis of this ellipsoid is along y_m that lies in the same direction of $\underline{u}_m$, and the length of this axis is σ_m . Similarly, the major axis of this ellipsoid lies along y_1 that points in the same direction of $\underline{u}_1$, and its length is σ_1 . This ellipsoid, which from now on will be referred to as the Manipulator Velocity Ratio Ellipsoid, is shown in Figure 3.1.

To be more concise, all what we have done in this analysis is proving that the set of all points of $\underline{r}_v \underline{u}_v$ spanning the m -dimensional space is an ellipsoid of m dimensions. The change from $\underline{r}_v \underline{u}_v$ to $\underline{y}$ has rotated the axes of the m -dimensional space to match the axes of the ellipsoid. In another sense, equation (3.29) also represents the same Manipulator Velocity Ratio Ellipsoid except that instead of being referenced to the rotated coordinate frames (y_i) that match the axes of the ellipsoid, it is referenced to the original coordinate frames ($\underline{r}_v \underline{u}_v$).

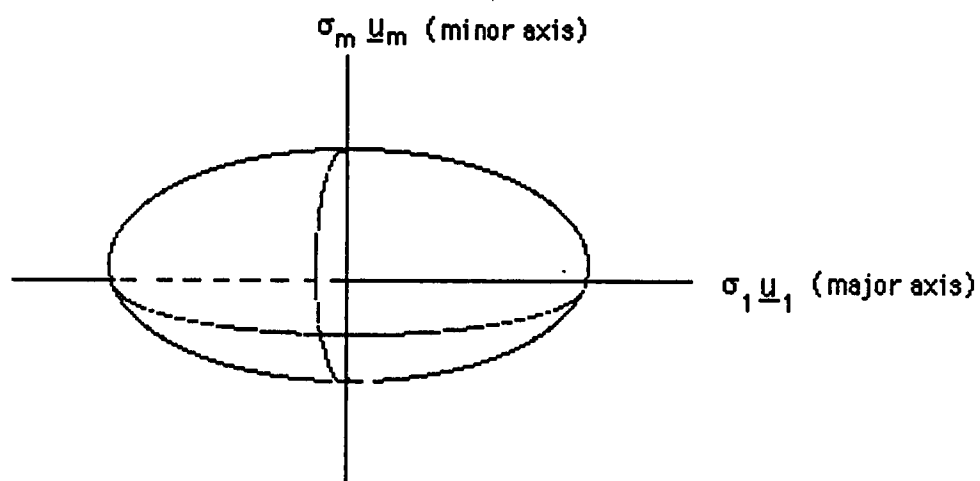


Figure 3.1. Manipulator Velocity Ratio Ellipsoid.

3.6 MATHEMATICAL PROOF FOR THE RANGE IN WHICH MVR OPERATES

We have shown geometrically that the vector $r_v \underline{u}_v$ moves on an elliptical surface whose major and minor axes have lengths of σ_1 and σ_m respectively and lie in the directions of the corresponding columns of the U_v matrix (of the singular value decomposition of J_v). Let us now prove mathematically that:

$$\sigma_m \leq r_v \leq \sigma_1$$

Let

$$\underline{a} = U_v^T \underline{u}_v \quad (3.34)$$

If we substitute this vector and its transpose in equation (3.29), we will get:

$$1/r_v^2 = (1/\sigma_1^2) a_1^2 + (1/\sigma_2^2) a_2^2 + \dots + (1/\sigma_m^2) a_m^2 \quad (3.35)$$

If all the singular values of the MVR Ellipsoid are equal to the maximum singular value, i.e.,

$$\sigma_2 = \sigma_3 = \dots = \sigma_m = \sigma_1 \quad (3.36)$$

then equation (3.35) can be written as:

$$1/\sigma_1^2 = (1/\sigma_1^2) a_1^2 + (1/\sigma_1^2) a_2^2 + \dots + (1/\sigma_1^2) a_m^2 \quad (3.37)$$

Subtracting equation (3.37) from equation (3.35), we obtain the following:

$$1/r_v^2 - 1/\sigma_1^2 = a_2^2 (1/\sigma_2^2 - 1/\sigma_1^2) + \dots + a_m^2 (1/\sigma_m^2 - 1/\sigma_1^2) \quad (3.38)$$

But $(1/\sigma_i^2 - 1/\sigma_1^2) \geq 0$ (for $i = 2, \dots, m$), so $1/r_v^2 \geq 1/\sigma_1^2$, and therefore

$r_v \leq \sigma_1$. In a similar fashion if all the singular values can be set equal to the

minimum singular value, we can prove that $r_v \geq \sigma_m$. Therefore,

$$\sigma_m \leq r_v \leq \sigma_1. \quad (3.39)$$

If σ_i is zero, then the velocity ratio in the direction of $\underline{u}_i$ will be zero. This what is actually referred to as a singularity. In this case, the least norm solution $\mathbf{J}^+ \underline{x}_v$ (where $\underline{x}_v$ lies in the direction of $\underline{u}_i$) will approach infinity. On the other hand, the magnitude of the end-effector velocity will be zero [2]. This means that the end-effector is unable to move in the direction of $\underline{u}_i$.

3.7 MVR GRADIENT BASED CONTROL SCHEMES

We can now define a performance criterion $H_v(\theta)$ that can be used to improve MVR in the direction of motion of the end-effector. Recall that we defined MVR as the ratio of the transformed end-effector velocity norm to the norm of the least norm transformed joint velocity vector. Let us define $H_v(\theta)$ as the inverse of the square of MVR, written as:

$$H_v(\theta) = \underline{u}_v^T (\mathbf{J}_v \mathbf{J}_v^T)^+ \underline{u}_v. \quad (3.40)$$

Any increase for r_v results in a decrease for $H_v(\theta)$. In another way, in order to maximize MVR, we have to decrease its performance criterion. To do that we have to choose a negative sign for the constant K, that has been defined in chapter 2 as a real scalar number that is based on the joint velocities limits. Differentiating equation (3.40) with respect to θ_i , we get:

$$\begin{aligned} \partial H_v / \partial \theta_i &= -\underline{u}_v^T (\mathbf{J}_v \mathbf{J}_v^T)^+^2 [\partial (\mathbf{J}_v \mathbf{J}_v^T) / \partial \theta_i] \underline{u}_v \\ &= -\underline{u}_v^T (\mathbf{J}_v \mathbf{J}_v^T)^+ [\partial (\mathbf{J}_v \mathbf{J}_v^T) / \partial \theta_i] (\mathbf{J}_v \mathbf{J}_v^T)^+ \underline{u}_v. \end{aligned} \quad (3.41)$$

To make the above equation easier to program, let us define a vector $\underline{Z}$ as:

$$\underline{z} = (\mathbf{J}_v \mathbf{J}_v^T)^+ \underline{u}_v \quad . \quad (3.42)$$

Substituting this equation into equation (3.41), we get:

$$\begin{aligned} \partial H_v / \partial \theta_i &= - \underline{z}^T [\partial (\mathbf{J}_v \mathbf{J}_v^T) / \partial \theta_i] \underline{z} \\ &= - \partial (\underline{z}^T \mathbf{J}_v \mathbf{J}_v^T \underline{z}) / \partial \theta_i \\ &= - \partial [(\mathbf{J}_v^T \underline{z})^T (\mathbf{J}_v^T \underline{z})] / \partial \theta_i \\ &= - [\partial (\mathbf{J}_v^T \underline{z})^T / \partial \theta_i] (\mathbf{J}_v^T \underline{z}) + (\mathbf{J}_v^T \underline{z})^T [\partial (\mathbf{J}_v^T \underline{z}) / \partial \theta_i] \quad . \quad (3.43) \end{aligned}$$

But it can be shown [12] that,

$$[\partial (\mathbf{J}_v^T \underline{z})^T / \partial \theta_i] (\mathbf{J}_v^T \underline{z}) = (\mathbf{J}_v^T \underline{z})^T [\partial (\mathbf{J}_v^T \underline{z}) / \partial \theta_i] \quad . \quad (3.44)$$

Therefore, equation (3.43) becomes:

$$\partial H_v / \partial \theta_i = - 2 (\mathbf{J}_v^T \underline{z})^T [\partial (\mathbf{J}_v^T \underline{z}) / \partial \theta_i] \quad . \quad (3.45)$$

As we mentioned earlier in the chapter, the end-effector velocity vector consists of linear and rotational parts. In addition to that, the Jacobian also consists of two parts: one related to linear motion and the other related to rotational motion. As a result of that we can partition the end-effector velocity vector as:

$$\dot{\underline{x}}_L = \begin{bmatrix} v_{h,x} \\ v_{h,y} \\ v_{h,z} \end{bmatrix} , \text{ and } \dot{\underline{x}}_A = \begin{bmatrix} \omega_{h,x} \\ \omega_{h,y} \\ \omega_{h,z} \end{bmatrix} \quad . \quad (3.46)$$

We can also partition the Jacobian as:

$$\mathbf{J}_L = \begin{bmatrix} \mathbf{J}_{L_1} & \mathbf{J}_{L_2} & \dots & \mathbf{J}_{L_7} \end{bmatrix} , \text{ and } \mathbf{J}_A = \begin{bmatrix} \mathbf{J}_{A_1} & \mathbf{J}_{A_2} & \dots & \mathbf{J}_{A_7} \end{bmatrix} . \quad (3.47)$$

Before we proceed further in the analysis, we need to note that the Jacobian used for computing this performance criterion and the ones that will be dealt with later in chapter 3 is the wrist Jacobian referenced to the base coordinate frame. Therefore, the velocity vector used in the kinematic equation of motion is the wrist velocity vector referenced to the base coordinate frame. Recall that the relationship between the end-effector velocity and the wrist velocity is as follows:

$$\begin{aligned}\underline{\omega}_w^0 &= \underline{\omega}_h^0, \text{ and} \\ \underline{v}_w^0 &= \underline{v}_h^0 - \underline{\omega}_h^0 \times \underline{Z}_h^0.\end{aligned}\quad (3.48)$$

$\underline{Z}_h^0$ is a vector representing the unit vector along the z-direction of the hand frame in terms of the unit vectors of the base frame. The wrist velocity vector with respect to base frame coordinates can also be partitioned as:

$$\dot{\underline{x}}_{wL} = \begin{bmatrix} v_{w,x} \\ v_{w,y} \\ v_{w,z} \end{bmatrix}, \text{ and } \dot{\underline{x}}_{wA} = \begin{bmatrix} \omega_{w,x} \\ \omega_{w,y} \\ \omega_{w,z} \end{bmatrix}.\quad (3.49)$$

Since we are now dealing with linear and angular motion, let us define two performance criteria: one related to linear motion and uses the linear wrist velocity vector and the upper half part of the wrist Jacobian, and the other performance criterion related to rotational motion and uses the angular wrist velocity vector and the lower half part of the wrist Jacobian.

3.7.1 MVR IN THE DIRECTION OF THE END-EFFECTOR MOTION (LINEAR END-EFFECTOR MOTION CASE)

The transformed linear velocity for the wrist is shown as:

$$\dot{\underline{x}}_{wvL} = \underline{W}_{x,L}^{1/2} \dot{\underline{x}}_{wL},\quad (3.50)$$

where $w_{x,L}$ is given by:

$$w_{x,L} = \begin{bmatrix} \gamma_1 & 0 & 0 \\ 0 & \gamma_2 & 0 \\ 0 & 0 & \gamma_3 \end{bmatrix}, \quad (3.51)$$

where $\gamma(s)$ are scalar constants. Substituting equations (3.51) and (3.49) into equation (3.50), we get:

$$\dot{x}_{w_v,L} = \begin{bmatrix} \gamma_1^{1/2} v_{w,x} \\ \gamma_2^{1/2} v_{w,y} \\ \gamma_3^{1/2} v_{w,z} \end{bmatrix}. \quad (3.52)$$

Define $u_{v,L}$ as:

$$u_{v,L} = \dot{x}_{w_v,L} / \sqrt{\dot{x}_{w_v,L}^T \dot{x}_{w_v,L}}, \quad (3.53)$$

where

$$\sqrt{\dot{x}_{w_v,L}^T \dot{x}_{w_v,L}} = \gamma_1 v_{w,x}^2 + \gamma_2 v_{w,y}^2 + \gamma_3 v_{w,z}^2. \quad (3.54)$$

Let us also define w_θ as a diagonal matrix represented by:

$$w_\theta = \begin{bmatrix} \epsilon_1 & & & & & & \\ & \epsilon_2 & & & & & \\ & & \epsilon_3 & & & & \\ & & & \epsilon_4 & & & \\ & & & & \epsilon_5 & & \\ & & & & & \epsilon_6 & \\ 0 & & & & & & \epsilon_7 \end{bmatrix}, \quad (3.55)$$

where $\varepsilon(s)$ are scalar constants. The transformed linear wrist Jacobian can be shown as:

$$\mathbf{J}_{w_{v,L}} = \mathbf{w}_x^{1/2} \mathbf{J}_{w_L} \mathbf{w}_\theta^{-1/2} . \quad (3.56)$$

$\mathbf{z}_L$ can also be represented as:

$$\mathbf{z}_L = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = (\mathbf{J}_{w_{v,L}} \mathbf{J}_{w_{v,L}}^T)^+ \mathbf{u}_{v,L} . \quad (3.57)$$

Consequently, an i^{th} element of the gradient vector of the linear part of MVR in the direction of motion can be written as:

$$\partial H_{\text{MVR,L}} / \partial \theta_i = -2 (\mathbf{J}_{w_{v,L}}^T \mathbf{z}_L)^T [\partial (\mathbf{J}_{w_{v,L}}^T \mathbf{z}_L) / \partial \theta_i] , \quad (3.58)$$

where $\mathbf{J}_{w_{v,L}}^T \mathbf{z}_L$ is an $n \times 1$ column vector.

3.7.2 MVR IN THE DIRECTION OF THE END-EFFECTOR MOTION (ANGULAR END-EFFECTOR MOTION CASE)

The transformed wrist velocity vector can also be found as:

$$\dot{\mathbf{x}}_{w_{v,A}} = \mathbf{w}_{x,A}^{1/2} \dot{\mathbf{x}}_{w_A} , \quad (3.59)$$

where $\mathbf{w}_{x,A}$ is given by the following:

$$\mathbf{w}_{x,A} = \begin{bmatrix} \gamma_4 & 0 & 0 \\ 0 & \gamma_5 & 0 \\ 0 & 0 & \gamma_6 \end{bmatrix} . \quad (3.60)$$

Substituting equations (3.60) and (3.49) into equation (3.59), we get:

$$\dot{\underline{x}}_{w,v,A} = \begin{bmatrix} \gamma_4^{1/2} \omega_{w,x} \\ \gamma_5^{1/2} \omega_{w,y} \\ \gamma_6^{1/2} \omega_{w,z} \end{bmatrix} . \quad (3.61)$$

Also define a unit vector in the direction of motion of the transformed angular velocity vector of the wrist as:

$$\underline{u}_{v,A} = \dot{\underline{x}}_{w,v,A} / \sqrt{\dot{\underline{x}}_{w,v,A}^T \dot{\underline{x}}_{w,v,A}} , \quad (3.62)$$

where the norm of this angular velocity vector is given by:

$$\sqrt{\dot{\underline{x}}_{w,v,A}^T \dot{\underline{x}}_{w,v,A}} = \gamma_4 \omega_{w,x}^2 + \gamma_5 \omega_{w,y}^2 + \gamma_6 \omega_{w,z}^2 . \quad (3.63)$$

The transformed wrist Jacobian related to angular motion is:

$$\underline{J}_{w,v,A} = \underline{w}_x^{1/2} \underline{J}_{w,A} \underline{w}_\theta^{-1/2} , \quad (3.64)$$

and $\underline{z}_A$ is:

$$\underline{z}_A = \begin{bmatrix} z_4 \\ z_5 \\ z_6 \end{bmatrix} = (\underline{J}_{w,v,A} \underline{J}_{w,v,A}^T)^+ \underline{u}_{v,A} . \quad (3.65)$$

The i^{th} element of the gradient vector of MVR with respect to angular motion can be shown as:

$$\partial H_{\text{MVR},A} / \partial \theta_i = -2 (\underline{J}_{w,v,A}^T \underline{z}_A)^T [\partial (\underline{J}_{w,v,A}^T \underline{z}_A) / \partial \theta_i] , \quad (3.66)$$

where $\mathbf{J}_{w_{v,A}}^T \mathbf{z}_A$ is an $n \times 1$ vector.

3.7.3 MVR IN THE DIRECTION OF THE MINIMUM SINGULAR VALUE (LINEAR END-EFFECTOR MOTION CASE)

This control scheme utilizes the maximization of MVR (for linear end-effector motion) in the direction of the minimum singular value of the Manipulator Velocity Ratio Ellipsoid. The equations derived in section 3.7.1 also apply to this case. The only change is to replace $\mathbf{u}_{v,L}$ (a unit vector in the direction of motion) with $\mathbf{u}_m$, which is the last column of the $\mathbf{U}_v$ matrix (from the SVD decomposition of the upper half of the wrist Jacobian, $\mathbf{J}_{w_{v,L}}$) that corresponds to the minimum singular value.

3.7.4 MVR IN THE DIRECTION OF THE MINIMUM SINGULAR VALUE (ANGULAR END-EFFECTOR MOTION CASE)

This control scheme utilizes the maximization of MVR (for angular end-effector motion) in the direction of the minimum singular value of the Manipulator Velocity Ratio Ellipsoid. The equations derived in section 3.7.2 also apply to this case. Again the only change is to replace $\mathbf{u}_{v,A}$ (a unit vector in the direction of motion) with $\mathbf{u}_m$, which is the last column of the $\mathbf{U}_v$ matrix (from the SVD decomposition of the lower half of the wrist Jacobian, $\mathbf{J}_{w_{v,A}}$) that corresponds to the minimum singular value.

3.8 MANIPULATOR MECHANICAL ADVANTAGE

For single-input-single-output type mechanisms, the transmission ratio is defined as the ratio of the output force to the input force. In multiple degrees of freedom robot manipulators, a similar performance measure called the Manipulator Mechanical Advantage is defined as the ratio of the end-effector force vector norm to the norm of the joint torque vector [2]. This performance measure abbreviated as MMA is denoted by r_m which is written as:

$$r_m = \|\underline{F}\|_{w_f} / \|\underline{\tau}\|_{w_\tau} , \quad (3.67)$$

where $\underline{F}_{w_f}$ and $\underline{\tau}_{w_\tau}$ are the transformed force and torque vectors defined previously in equation (3.7). Substituting equation (3.7) into (3.67), we get:

$$r_m = \sqrt{\underline{F}_m^T \underline{F}_m / \underline{\tau}_m^T \underline{\tau}_m} . \quad (3.68)$$

If we substitute in the above equation the transformed torque defined in equation (3.12), the result is:

$$r_m = \sqrt{\underline{F}_m^T \underline{F}_m / (\underline{F}_m^T \underline{J}_m \underline{J}_m^T \underline{F}_m)} . \quad (3.69)$$

Let us define the unit vector in the direction of the transformed force as:

$$\underline{u}_m = \underline{F}_m / \sqrt{\underline{F}_m^T \underline{F}_m} . \quad (3.70)$$

It can be shown then that the above equation reduces to:

$$r_m = 1 / \|\underline{J}_m^T \underline{u}_m\| . \quad (3.71)$$

3.9 GEOMETRIC INTERPRETATION OF THE MANIPULATOR MECHANICAL ADVANTAGE

If we decompose J_m using the SVD decomposition technique, we can show that:

$$J_m = U_m \Sigma_m V_m, \quad (3.72)$$

where Σ_m is a triangular diagonal matrix expressed as [12]:

$$\Sigma_m = \begin{bmatrix} \psi_1 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \psi_2 & & & & \cdot \\ \cdot & & \cdot & & & \cdot \\ \cdot & & & \cdot & & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 & \psi_m \\ & & & & & 0 \end{bmatrix}, \quad (3.73)$$

where $\psi_1 > \psi_2 > \dots > \psi_m$. Squaring the right hand side and the left hand side of equation (3.71), we write:

$$r_m^2 = 1/(J_m^T \underline{u}_m)^T (J_m^T \underline{u}_m). \quad (3.74)$$

If we substitute the transpose of equation (3.72) into equation (3.74), we get:

$$1/r_m^2 = \underline{u}_m^T U_m (\Sigma_m \Sigma_m^T) U_m^T \underline{u}_m. \quad (3.75)$$

The above equation can be written in a different form as:

$$1 = \underline{u}_m^T r_m U_m (\Sigma_m \Sigma_m^T) U_m^T r_m \underline{u}_m. \quad (3.76)$$

Let

$$\underline{b} = U_m^T r_m \underline{u}_m. \quad (3.77)$$

If we substitute equation (3.77) into equation (3.76), equation (3.76) becomes:

$$1 = \underline{b}^T (\Sigma_m \Sigma_m^T) \underline{b} \quad , \quad (3.78)$$

which is the same as:

$$1 = \psi_1^2 b_1^2 + \psi_2^2 b_2^2 + \dots + \psi_m^2 b_m^2 \quad . \quad (3.79)$$

Geometrically equation (3.79) represents an ellipsoid of m dimensions. The minor axis of this ellipsoid lies in the direction of $\underline{u}_1$ and its length is $1/\psi_1$. The major axis lies in the direction of $\underline{u}_m$ and its length is $1/\psi_m$. This ellipsoid is the surface formed by the end-points of vectors $r_m \underline{u}_m$ spanning the m -dimensional space (see Figure 3.2).

3.10 MATHEMATICAL PROOF FOR THE RANGE IN WHICH MMA OPERATES

Let us now show mathematically that in general r_m lies between $1/\Psi_1$ and $1/\Psi_m$. If we let:

$$\underline{c} = U_m^T \underline{u}_m \quad , \quad (3.80)$$

and we substitute this equation into equation (3.76), we can show:

$$1/r_m^2 = \psi_1^2 c_1^2 + \psi_2^2 c_2^2 + \dots + \psi_m^2 c_m^2 \quad . \quad (3.81)$$

If all the singular values are equal to the maximum singular value, i.e.

$$\psi_2 = \psi_3 = \dots = \psi_m = \psi_1 \quad , \quad (3.82)$$

then the following can be written:

$$\psi_1^2 = \psi_1^2 c_1^2 + \psi_1^2 c_2^2 + \dots + \psi_1^2 c_m^2 \quad . \quad (3.83)$$

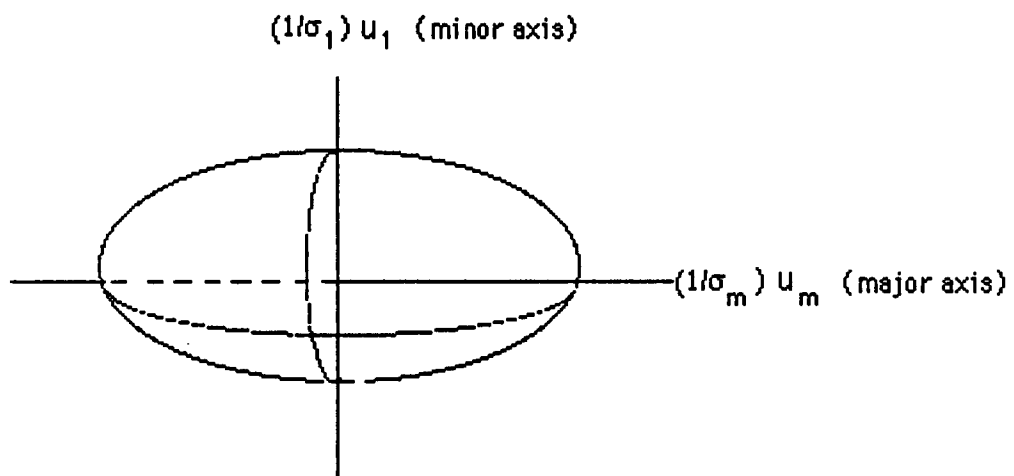


Figure 3.2. Manipulator Mechanical Advantage Ellipsoid.

Subtracting equation (3.83) from equation (3.81), we get:

$$1/r_m^2 - \psi_1^2 = c_2^2 (\psi_2^2 - \psi_1^2) + \dots + c_m^2 (\psi_m^2 - \psi_1^2) . \quad (3.84)$$

But

$$\psi_i^2 - \psi_1^2 \leq 0 . \quad (3.85)$$

Therefore,

$$r_m \geq 1/\psi_1 . \quad (3.86)$$

Similarly we can prove that:

$$r_m \leq 1/\psi_m . \quad (3.87)$$

Thus, r_m lies in the following range:

$$1/\psi_1 \leq r_m \leq 1/\psi_m . \quad (3.88)$$

If ψ_i (for $i = 1, \dots, m$) is zero, then the mechanical advantage in the direction of $\underline{u}_i$ will approach infinity. In this case, the joint torques produced by the actuators will be zero.

3.11 MMA GRADIENT BASED CONTROL SCHEMES

We can now define a performance criterion $H_v(\theta)$ that we can use to improve MMA in the direction of the force vector applied at the wrist. Let us define $H_m(\theta)$ as the inverse of the square of MMA, written as:

$$H_m(\theta) = \left| \mathbf{J}_m^T \underline{u}_m \right|^2 . \quad (3.89)$$

In order to increase MMA, $H_m(\theta)$ should decrease along the trajectory. In this case,

the scalar constant K should be a negative number. Equation (3.89) can be expanded as:

$$H_m(\theta) = (J_m^T \underline{u}_m)^T (J_m^T \underline{u}_m) \quad . \quad (3.90)$$

Differentiating equation (3.90) with respect to θ_i , we get:

$$\partial H_m / \partial \theta_i = [\partial (J_m^T \underline{u}_m)^T / \partial \theta_i] (J_m^T \underline{u}_m) + (J_m^T \underline{u}_m)^T [\partial (J_m^T \underline{u}_m) / \partial \theta_i] \quad . \quad (3.91)$$

It can be shown that the above equation can be reduced to:

$$\partial H_m / \partial \theta_i = 2 (J_m^T \underline{u}_m)^T [\partial (J_m^T \underline{u}_m) / \partial \theta_i] \quad . \quad (3.92)$$

We should make it clear that the force vector (including forces and moments) used in optimizing MMA is the force vector that is applied at the wrist. Since we are using the wrist Jacobian, we have to be consistent in using the force vector applied at the wrist. Knowing the force vector applied at the end-effector, we can determine the forces and moments applied at the wrist joint using the following relations:

$$\underline{F}_w^0 = \underline{F}_h^0 \quad , \quad (3.93)$$

$$\text{and } \underline{M}_w^0 = \underline{M}_h^0 - \underline{F}_h^0 \times \underline{Z}_h^0 \quad . \quad (3.94)$$

The hand moments and forces together with the wrist moments and forces are referred to the base coordinate frame. $\underline{Z}_h^0$ is the unit vector along the z-direction of the end-effector frame but referenced to the base frame.

Since we are dealing with both forces and moments applied at the wrist joint, let us define two control schemes: one control scheme determines MMA based on using only the force components of the force vector applied at the wrist and the upper half part of the wrist Jacobian, while the other control scheme determines MMA based on using the moment components of the force vector applied at the wrist and the lower half part of the wrist Jacobian.

3.11.1 MMA BASED ON THE FORCE COMPONENTS APPLIED AT THE WRIST

The transformed wrist applied force vector is derived as follows:

$$\underline{E}_{m,f} = \underline{w}_{f,f}^{1/2} \underline{E}_{w,f} \quad , \quad (3.95)$$

where $\underline{w}_{f,f}$ is the weighting matrix associated with the force components applied at the wrist. $\underline{E}_{w,f}$ is the wrist force vector, and $\underline{E}_{m,f}$ is the transformed wrist force vector. Let $\underline{w}_{f,f}$ be given by:

$$\underline{w}_{f,f} = \begin{bmatrix} \alpha_1^{1/2} & 0 & 0 \\ 0 & \alpha_2^{1/2} & 0 \\ 0 & 0 & \alpha_3^{1/2} \end{bmatrix} \quad , \quad (3.96)$$

where $\alpha(s)$ are scalar constants. $\underline{E}_{w,f}$ is given by:

$$\underline{E}_{w,f} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \quad . \quad (3.97)$$

$\underline{E}_{m,f}$ can be shown as:

$$\underline{E}_{m,f} = \begin{bmatrix} \alpha_1^{1/2} f_x \\ \alpha_2^{1/2} f_y \\ \alpha_3^{1/2} f_z \end{bmatrix} \quad . \quad (3.98)$$

We can the define a unit vector in the direction of the force applied at the wrist as:

$$\underline{u}_{m,f} = \underline{F}_{m,f} / \sqrt{\underline{F}_{m,f}^T \underline{F}_{m,f}} \quad , \quad (3.99)$$

where

$$\sqrt{\underline{F}_{m,f}^T \underline{F}_{m,f}} = (\alpha_1 f_x^2 + \alpha_2 f_y^2 + \alpha_3 f_z^2) \quad . \quad (3.100)$$

Let us also define $\underline{w}_t$ as:

$$\underline{w}_t = \begin{bmatrix} \beta_1 & & & & & & \\ & \beta_2 & & & & & \\ & & \beta_3 & & & & \\ & & & \beta_4 & & & \\ & & & & \beta_5 & & \\ & & & & & \beta_6 & \\ 0 & & & & & & \beta_7 \end{bmatrix} \quad , \quad (3.101)$$

where $\beta(s)$ are scalar constants. The transformed wrist Jacobian related to the force components is expressed as:

$$\underline{J}_{w_{m,f}} = \underline{w}_{f,f}^{-1/2} \underline{J}_{w_f} \underline{w}_t^{1/2} \quad . \quad (3.102)$$

The i^{th} element of the gradient vector of MMA in the direction of the applied force is then defined as:

$$\partial H_{\text{MMA},f} / \partial \theta_i = 2 (\underline{J}_{w_{m,f}}^T \underline{u}_{m,f})^T [\partial (\underline{J}_{w_{m,f}}^T \underline{u}_{m,f}) / \partial \theta_i] \quad , \quad (3.103)$$

where $\underline{J}_{w_{m,f}}^T \underline{u}_{m,f}$ is an $n \times 1$ column vector.

3.11.2 MMA BASED ON THE MOMENT COMPONENTS APPLIED AT THE WRIST

The transformed moment vector applied at the wrist can also be found as:

$$\underline{F}_{m,m} = \underline{w}_{f,m}^{1/2} \underline{F}_{w,m} \quad , \quad (3.104)$$

where $\underline{w}_{f,m}$ is the weighting matrix associated with the moment applied at the wrist. $\underline{F}_{w,m}$ is the wrist moment vector, and $\underline{F}_{m,m}$ is the transformed wrist

moment vector. Let $w_{f,m}$ be given by:

$$w_{f,m} = \begin{bmatrix} \alpha_4^{1/2} & 0 & 0 \\ 0 & \alpha_5^{1/2} & 0 \\ 0 & 0 & \alpha_6^{1/2} \end{bmatrix}, \quad (3.105)$$

and let $E_{w,m}$ be given by:

$$E_{w,m} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}. \quad (3.106)$$

$E_{m,m}$ can be shown as:

$$E_{m,m} = \begin{bmatrix} \alpha_4^{1/2} m_x \\ \alpha_5^{1/2} m_y \\ \alpha_6^{1/2} m_z \end{bmatrix}. \quad (3.107)$$

A unit vector in the direction of the moment applied at the wrist is expressed as:

$$u_{m,m} = E_{m,m} / \sqrt{E_{m,m}^T E_{m,m}}, \quad (3.108)$$

where

$$\sqrt{E_{m,m}^T E_{m,m}} = (\alpha_4 m_x^2 + \alpha_5 m_y^2 + \alpha_6 m_z^2). \quad (3.109)$$

The transformed wrist Jacobian related to the moment components of the force vector applied at the wrist is written as:

$$J_{w,m} = w_{f,m}^{-1/2} J_{w_m} w_t^{1/2}. \quad (3.110)$$

The i^{th} element of the gradient vector of MMA in the direction of the applied moment

is expressed as:

$$\partial H_{MMA,m} / \partial \theta_i = 2 (J_{w_{m,m}}^T \underline{u}_{m,m})^T [\partial (J_{w_{m,m}}^T \underline{u}_{m,m}) / \partial \theta_i] , \quad (3.111)$$

where $J_{w_{m,m}}^T \underline{u}_{m,m}$ is an $n \times 1$ column vector.

3.12 DUALITY OF MVR AND MMA

Recall that the Manipulator Velocity Ratio Ellipsoid was defined by the following equation:

$$1 = (\underline{r}_v \underline{u}_v^T) \underline{U}_v (\underline{\Sigma}_v \underline{\Sigma}_v^T)^+ \underline{U}_v^T (\underline{r}_v \underline{u}_v) . \quad (3.112)$$

On the other hand, the Manipulator Mechanical Advantage Ellipsoid was defined by this equation:

$$1 = \underline{u}_m^T \underline{r}_m \underline{U}_m (\underline{\Sigma}_m \underline{\Sigma}_m^T) \underline{U}_m^T \underline{r}_m \underline{u}_m . \quad (3.113)$$

Comparing these two equations, we deduce the following:

- 1) The principal axes of the Manipulator velocity Ratio Ellipsoid coincide with the principal axes of the Manipulator Mechanical Advantage Ellipsoid.
- 2) The lengths of the principal axes of the Manipulator Velocity Ratio Ellipsoid are in inverse proportions with the corresponding principal axes of the Manipulator Mechanical Advantage Ellipsoid.

This means that the direction in which MVR is maximum is also the direction in which MMA is minimum. So actually, whenever we maximize MVR in a certain direction, we will be minimizing MMA in that direction.

3.13 MANIPULABILITY MEASURE CRITERION

In this section, we will use the Manipulability Measure [14] as a performance criterion for avoiding singularities as well as for improving the ability of the

manipulator end-effector to move in different directions. This section will present a control scheme based on this criterion. If we define G as:

$$G = J J^T, \quad (3.114)$$

where J is the $m \times n$ Jacobian matrix. The performance criterion for singularity avoidance can then be written as:

$$\begin{aligned} H &= \sqrt{\det(G)} \\ &= \sigma_1 \sigma_2 \dots \sigma_m. \end{aligned} \quad (3.115)$$

In equation (3.115), m is defined as the number of rows of the Jacobian.

An arm configuration approaches a singular configuration when $\det(G)$ approaches zero. This can happen when any singular value, σ_i (for $i = 1, \dots, m$), approaches zero. This results in a situation in which the arm can not move in the direction u_i (the i^{th} column of the U matrix of the SVD decomposition) corresponding to σ_i . In order to avoid any singular configuration, the determinant of G should not approach zero. One way to achieve this is to maximize the product of the singular values (i.e. $\sigma_1 \sigma_2 \dots \sigma_m$). Since the volume of the Manipulability Ellipsoid [14] is proportional to the determinant of G , this means that if we maximize the determinant of G we will also be maximizing the volume of the Manipulability Ellipsoid. This will also result in the Manipulability Ellipsoid being more closer to a sphere than an ellipsoid whose major and minor axes vary considerably in length. The 1^{th} element of the gradient of the Manipulability Measure criterion can be shown as follows:

$$\partial H / \partial \theta_1 = [1/2\sqrt{\det(G)}] [\partial(\det(G))/\partial \theta_1], \quad (3.116)$$

where $1 = 1, 2, \dots, n$. But $\det(G)$ can be expressed as the combination of row #1

and the cofactors of that row. Thus $\det(G)$ can be written as:

$$\det(G) = g_{11}\Delta_{11} + g_{12}\Delta_{12} + \dots + g_{1m}\Delta_{1m} \quad , \quad (3.117)$$

where g_{ij} is an entry in G corresponding to the i^{th} row and the j^{th} column. Δ_{ij} is the cofactor of a sub-matrix a_{ij} for G . Differentiating $\det(G)$ with respect to 1 results in the following expression:

$$\partial[\det(G)]/\partial\theta_1 = \sum_{i=1}^m (\Delta_{1i} \dot{g}_{1i} + \dot{\Delta}_{1i} g_{1i}) \quad . \quad (3.118)$$

Δ_{1i} can be shown as:

$$\Delta_{1i} = (-1)^{1+i} \det(a_{1i}) \quad , \quad (3.119)$$

where $\det(a_{1i})$ can be written as:

$$\det(a_{1i}) = \sum_{i=2}^m g_{2i} \Delta_{2i} \quad . \quad (3.120)$$

Δ_{2i} can also be expressed as:

$$\Delta_{2i} = (-1)^{2+i} \det(a_{2i}) \quad , \quad (3.121)$$

where $\det(a_{2i})$ for instance can be expressed as a combination of either row 1,3,4 ..., or m and the cofactors of that row. Repeating the same procedure for all other cases of cofactors, and substituting these combinations in equation (3.118), the derivative of the singularity avoidance criterion with respect to 1 can be shown as:

$$\partial[\det(G)]/\partial\theta_1 = \sum_{i,j=1}^m (\Delta_{ij} \dot{g}_{ij}) \quad . \quad (3.122)$$

Substituting this equation into equation (3.116), the 1^{th} element of the gradient for the singularity avoidance performance criterion can be written as:

$$\partial H / \partial \theta_1 = [1/2\sqrt{\det(G)}] \sum_{i,j=1}^m \Delta_{ij} \dot{g}_{ij} \quad (3.123)$$

Let us represent g_{ij} in another form as:

$$g_{ij} = \mathbf{J}_i \mathbf{J}_j^T, \quad (3.124)$$

where $\mathbf{J}_i$ is the i^{th} row of $\mathbf{J}$. Differentiating g_{ij} with respect to 1 results in the following:

$$\begin{aligned} \dot{g}_{ij} &= \dot{\mathbf{J}}_i \mathbf{J}_j^T + \mathbf{J}_i \dot{\mathbf{J}}_j^T \\ &= \dot{\mathbf{J}}_i \mathbf{J}_j^T + \mathbf{J}_j \dot{\mathbf{J}}_i^T \end{aligned} \quad (3.125)$$

The relationship between elements in the inverse matrix of $\mathbf{G}$ and elements in the cofactor matrix of $\mathbf{G}$ is expressed as:

$$G_{ij}^{-1} = \Delta_{ij} / \det(\mathbf{G}) \quad (3.126)$$

Substituting equations (3.125) and (3.126) into (3.123), the 1^{th} element of the gradient can then be written as:

$$\partial H / \partial \theta_1 = (1/2)\sqrt{\det(\mathbf{G})} \sum_{i,j=1}^m [G^{-1}]_{ij} (\dot{\mathbf{J}}_i \mathbf{J}_j^T + \mathbf{J}_j \dot{\mathbf{J}}_i^T) \quad (3.127)$$

As we did in other performance criteria, we will consider the Manipulability Measure criterion for two cases: one that corresponds to the linear motion of the wrist and the other that corresponds to the rotational motion of the wrist. This means that in one case $\mathbf{J}$ consists of the upper half part of the wrist Jacobian (referred to as $\mathbf{J}_{w1}$), and in another case $\mathbf{J}$ consists of the lower half part of the wrist Jacobian (referred to as $\mathbf{J}_{wA}$). The gradients of these Manipulability Measure cases are:

$$a) \quad \partial H / \partial \theta_1 = (1/2)\sqrt{\det(\mathbf{G}_1)} \sum_{i,j=1}^m [G_1^{-1}]_{ij} (\dot{\mathbf{J}}_{i_{w1}} \mathbf{J}_{j_{w1}}^T + \mathbf{J}_{j_{w1}} \dot{\mathbf{J}}_{i_{w1}}^T), \quad (3.128)$$

and,

$$b) \quad \partial H / \partial \theta_1 = (1/2) \sqrt{\det(G_A)} \sum_{i,j=1}^m [G_A^{-1}]_{ij} (J_{i_{w_A}}' J_{j_{w_A}}^T + J_{j_{w_A}}' J_{i_{w_A}}^T) \quad . \quad (3.129)$$

G_1 and G_A correspond to J_{w_1} and J_{w_A} respectively.

CHAPTER 4

CONTROL SCHEME IMPLEMENTATION ON THE NASA LABORATORY TELEROBOTIC MANIPULATOR

The LTM, shown in Figure 4.1, is a seven degree of freedom manipulator with a spherical wrist, and whose joints are all revolute joints. Unlike many industrial robots, the wrist is designed in such away that the wrist singularities occur when the hand is pointing to its sides and not when it is pointing straight out. The arm has three pitch/yaw joints that provide the shoulder, the elbow, and the wrist joints. The seventh degree of freedom is provided by the wrist roll motion.

4.1 FORWARD KINEMATICS OF THE LTM

The Denavit-Hartenberg notation [5] is introduced as a way of describing these kinematic relationships between the joints. This method is based on the 4x4 matrix representation called the homogeneous transformation matrix. The homogeneous transformation matrix is represented as:

$$A = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (4.1)$$

where R is a 3x3 rotation matrix, P is 3x1 position vector, and 0 is a 1x3 perspective transformation matrix whose entries are all zeros. The Denavit-Hartenberg parameters for the coordinate frames attached to the LTM joints (Figure 4.1) are listed in table 4.1.

The Denavit-Hartenberg transformation matrix, relating the i^{th} coordinate frame to the $(i-1)^{\text{th}}$ coordinate frame is shown as follows [5]:

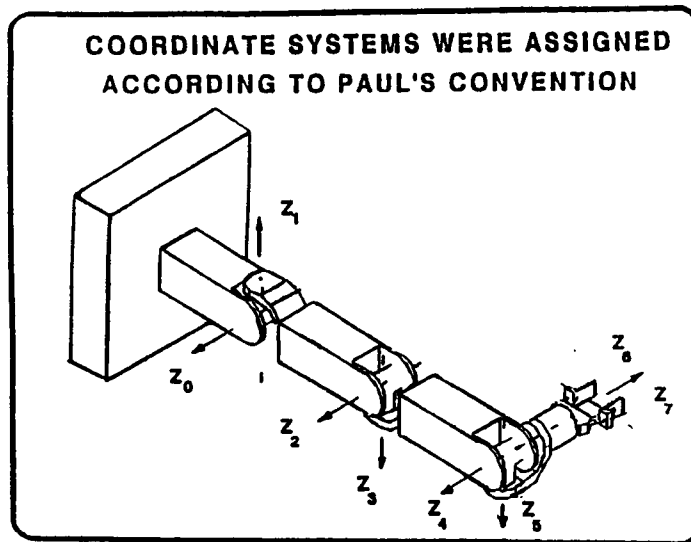


Figure 4.1. Laboratory Telerobotic Manipulator with coordinate frames associated with every joint.

Table 4.1. Denavit-Hartenberg table for the LTM.

θ_i (deg)	d_i (m)	α_i (deg)	a_i (m)
θ_1	0	-90	0
θ_2	0	90	a_2
θ_3	0	90	0
θ_4	0	-90	a_4
θ_5	0	90	0
θ_6	0	90	0
θ_7	d_7	0	0

$$A_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} . \quad (4.2)$$

The homogeneous matrix T_7^0 , which specifies the position and orientation of the 7th coordinate frame with respect to the base frame, is the chain product of successive coordinate transformation matrices of A_i^{i-1} (for $i = 1, 2, \dots, 7$). This transformation is expressed as:

$$T_7^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 A_6^5 A_7^6 . \quad (4.3)$$

The position and orientation of the wrist frame (5th coordinate frame) with respect to the base coordinate frame can in a similar fashion be expressed as follows:

$$T_5^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 . \quad (4.4)$$

Appendix A contains a list of the transformation matrices relating successive coordinate frames to each other. Appendix B contains derivations of the position and orientation of both the end-effector (hand) and the wrist with respect to the base coordinate frame.

4.2 THE WRIST JACOBIAN REFERENCED TO THE BASE COORDINATE FRAME

In the previous section, the position and orientation of the manipulator end-effector as well as the wrist joint were evaluated in terms of joint rotations. In this section we are concerned with the linear as well as the rotational velocity at which the wrist moves. In order for the wrist to move in a specified direction and

with a specified speed, it is necessary that the other joints motions be coordinated in a certain way. In order to coordinate the joints motions, we need to derive the differential relationship between the joints rotations and the wrist location.

Let $d\mathbf{x}_w$ be the (6x1) infinitesimal location vector of the wrist. The first three components of this vector are the linear velocity terms of the wrist, while the other three are the angular velocity terms. Also, let $d\mathbf{\theta}$ be a (7x1) vector representing the infinitesimal displacements (rotations in this case) of the seven joints. $\mathbf{J}_{wrist}$ can then be thought of as a mapping matrix that maps $d\mathbf{\theta}$ in $\mathbf{R}^{7 \times 1}$ to $d\mathbf{x}_w$ in $\mathbf{R}^{6 \times 1}$. $\mathbf{J}_{wrist}$ should then satisfy the following relation:

$$d\mathbf{x}_w = \mathbf{J}_{wrist} d\mathbf{\theta} \quad . \quad (4.5)$$

Dividing both sides of the equation by dt , we get

$$d\mathbf{x}_w/dt = \mathbf{J}_{wrist} d\mathbf{\theta}/dt \quad . \quad (4.6)$$

In a more detailed form, $\mathbf{J}_{wrist}$ represents the following [1]:

$$\mathbf{J}_{wrist} = \begin{bmatrix} \partial x_w / \partial \theta_1 & \partial x_w / \partial \theta_2 & \dots & \partial x_w / \partial \theta_7 \\ \partial y_w / \partial \theta_1 & \partial y_w / \partial \theta_2 & \dots & \partial y_w / \partial \theta_7 \\ \partial z_w / \partial \theta_1 & \partial z_w / \partial \theta_2 & \dots & \partial z_w / \partial \theta_7 \\ \partial \omega_{x_w} / \partial \theta_1 & \partial \omega_{x_w} / \partial \theta_2 & \dots & \partial \omega_{x_w} / \partial \theta_7 \\ \partial \omega_{y_w} / \partial \theta_1 & \partial \omega_{y_w} / \partial \theta_2 & \dots & \partial \omega_{y_w} / \partial \theta_7 \\ \partial \omega_{z_w} / \partial \theta_1 & \partial \omega_{z_w} / \partial \theta_2 & \dots & \partial \omega_{z_w} / \partial \theta_7 \end{bmatrix} \quad . \quad (4.7)$$

Thus, the wrist Jacobian can represent the relationship between the joint velocities and the resulting wrist velocities, as well as the relationship between the

joint displacements and the wrist position. Each column vector of J_{wrist} will then represent the linear and angular velocity of the wrist joint due to unit velocity of the corresponding joints.

Since the LTM is a seven-degree-of-freedom robot manipulator with rotational joints, the wrist Jacobian is a 6x7 matrix whose i^{th} column vector J_i is given by the following equation:

$$J_i(\theta) = \begin{bmatrix} \underline{b}_{i-1} \times \underline{r}_{i-1,5} \\ \underline{b}_{i-1} \end{bmatrix}, \quad (4.8)$$

where

$\underline{r}_{i-1,5}$ = a three dimensional position vector of the origin of the wrist frame from the origin of the $(i-1)^{\text{th}}$ coordinate frame expressed in the base coordinate frame.

$\underline{b}_{i-1}$ = is the unit vector along the axis of motion of joint i expressed in the base coordinate frame.

The wrist Jacobian is then represented as:

$$J_{\text{wrist}}(\theta) = \begin{bmatrix} \underline{b}_0 \times \underline{r}_{0,5} & \underline{b}_1 \times \underline{r}_{1,5} & \dots & \underline{b}_6 \times \underline{r}_{6,5} \\ \underline{b}_0 & \underline{b}_1 & \dots & \underline{b}_6 \end{bmatrix}. \quad (4.9)$$

Appendix C contains derivations for the wrist Jacobian referenced to the base coordinate frame.

4.3 THE RELATIONSHIP BETWEEN THE END-EFFECTOR AND THE WRIST VELOCITIES

So far we have derived the wrist Jacobian relating the wrist velocity vector to the joint velocities. The Kinematic equation of motion using the wrist Jacobian is

described by the following equation:

$$\dot{\underline{x}}_w = J_{\text{wrist}} \dot{\underline{\theta}} \quad , \quad (4.10)$$

where $\dot{\underline{x}}_w$ is a 6x1 velocity vector consisting of the three linear wrist velocities ($v_{w,x}, v_{w,y}, v_{w,z}$) and the three angular wrist velocities ($\omega_{w,x}, \omega_{w,y}, \omega_{w,z}$). $\dot{\underline{\theta}}$ is a 7x1 vector consisting of the joint velocities namely, $\dot{\theta}_1, \dot{\theta}_2, \dots, \dot{\theta}_7$.

Knowing either the end-effector or the wrist velocity vector, we will be able to know the other due to the following relation that exists between the end-effector and the wrist velocities (linear and angular velocities):

$$\underline{\omega}_w^0 = \underline{\omega}_h^0 \quad , \quad (4.11)$$

$$\underline{v}_w^0 = \underline{v}_h^0 - \underline{\omega}_h^0 \times \underline{Z}_h^0 \quad , \quad (4.12)$$

where $\underline{Z}_h^0$ is the unit vector along the z-direction of the end-effector frame but referenced to the base coordinate frame. $\omega_{h,x}, \omega_{h,y}, \omega_{h,z}$ are the angular velocities of the hand (or end-effector), and $v_{h,x}, v_{h,y}, v_{h,z}$ are the linear velocities of the hand. Since $\underline{Z}_h^0$ is actually the third column in the orientation matrix of the hand, equation (4.12) can be expressed as:

$$\begin{bmatrix} v_{w,x} \\ v_{w,y} \\ v_{w,z} \end{bmatrix} = \begin{bmatrix} v_{h,x} \\ v_{h,y} \\ v_{h,z} \end{bmatrix} - \begin{bmatrix} \omega_{h,x} \\ \omega_{h,y} \\ \omega_{h,z} \end{bmatrix} \times d_7 \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad , \quad (4.13)$$

where d_7 is the distance between the fifth joint and the end-effector.

4.4 INVERSE KINEMATICS OF THE LTM

The inverse kinematics of the LTM is based on using the wrist Jacobian referenced to the third coordinate frame. The inverse kinematic control scheme discussed in chapter 2 is applied to the LTM. Closed form solutions for the least norm joint angles are then obtained for two illustrative cases.

4.4.1 THE WRIST JACOBIAN REFERENCED TO THE 3rd COORDINATE FRAME

Let the Jacobian used in the inverse kinematic control scheme be the wrist Jacobian referenced to the third coordinate frame and not the wrist Jacobian referenced to the base frame. As a result, we will refer the end-effector velocity vector and the wrist velocity vector to the third coordinate frame. The wrist Jacobian referenced to the base frame will only be used in the control schemes that optimize several performance criteria. If $\dot{\underline{x}}_h$ is the hand (end-effector) velocity vector referenced to the base coordinate frame, then the hand velocity vector referenced to the third coordinate frame, $\dot{\underline{x}}_h^3$, will be determined by the following equation:

$$\dot{\underline{x}}_h^3 = R_0^3 \dot{\underline{x}}_h \quad , \quad (4.14)$$

where R_0^3 is a 3x3 matrix relating the base coordinate frame to the third coordinate frame. It is given by:

$$R_0^3 = \begin{bmatrix} C_1 C_2 C_3 - S_1 S_3 & S_1 C_2 C_3 + C_1 S_3 & -S_2 C_3 \\ C_1 S_2 & S_1 S_2 & C_2 \\ C_1 C_2 S_3 + S_1 C_3 & S_1 C_2 S_3 - C_1 C_3 & -S_2 S_3 \end{bmatrix} \quad , \quad (4.15)$$

where $C_i = \cos \theta_i$ and $S_i = \sin \theta_i$. The wrist velocity vectors can be obtained

from the following relationships:

$$\underline{\omega}_w^3 = \underline{\omega}_h^3, \quad (4.16)$$

$$\underline{v}_w^3 = \underline{v}_h^3 - \underline{\omega}_h^3 \times d_7 \underline{Z}_h^3, \quad (4.17)$$

where

$$\underline{Z}_h^3 = \begin{bmatrix} S_4 C_6 + C_4 C_5 S_6 \\ S_4 C_5 S_6 - C_4 C_6 \\ -S_5 S_6 \end{bmatrix}. \quad (4.18)$$

By using the wrist Jacobian referenced to the third coordinate frame, $\underline{J}_w^3$, the kinematic equation of motion can then be represented as:

$$\dot{\underline{x}}_w^3 = \underline{J}_w^3 \dot{\underline{\theta}}, \quad (4.19)$$

where $\underline{J}_w^3$ is given by:

$$\underline{J}_w^3 = \begin{bmatrix} a_4 S_2 S_3 S_4 + a_2 C_2 S_3 & a_4 C_3 S_4 & 0 & -a_4 S_4 & 0 & 0 & 0 \\ -a_4 S_2 S_3 C_4 & -a_4 C_3 C_4 - a_2 & 0 & a_4 C_4 & 0 & 0 & 0 \\ -a_4 S_2 C_3 S_4 - a_4 C_2 C_4 - a_2 C_2 C_3 & a_4 S_3 S_4 & -a_4 C_4 & 0 & 0 & 0 & 0 \\ -S_2 C_3 & S_3 & 0 & 0 & -S_4 & C_4 S_5 & C_4 C_5 S_6 + S_4 C_6 \\ C_2 & 0 & 1 & 0 & C_4 & S_4 S_5 & S_4 C_5 S_6 - C_4 C_6 \\ -S_2 S_3 & -C_3 & 0 & 1 & 0 & C_5 & -S_5 S_6 \end{bmatrix} \quad (4.20)$$

4.4.2 THE PARTITIONING OF J_w^3 FOR USE IN THE INVERSE KINEMATIC CONTROL SCHEME

If J_w^3 is a full rank matrix, and if $\underline{\alpha}$ represents the dependant column, then $\dot{\underline{x}}_w^3$ can be written as:

$$\dot{\underline{x}}_w^3 = [\underline{\alpha} J_w^{3*}] \dot{\underline{\theta}} \quad (4.21)$$

J_w^{3*} can be partitioned as:

$$J_w^{3*} = \begin{bmatrix} J_{w,1}^{3*} & 0 \\ J_{w,2}^{3*} & J_{w,3}^{3*} \end{bmatrix}, \quad (4.22)$$

where each of the constituents of J_w^{3*} is a 3x3 sub-matrix. If we replace J^* by J_w^{3*} in equations (2.10) and (2.12), then we can compute $\dot{\underline{\theta}}_p$ and $\dot{\underline{\theta}}_h$ by simply determining $J_w^{3* -1} \dot{\underline{x}}_w^3$ and $-J_w^{3* -1} \underline{\alpha}$. If $\underline{z}$ and $\underline{y}$ are set equal to $\dot{\underline{x}}_w^3$ and $J_w^{3* -1} \dot{\underline{x}}_w^3$ respectively in one case (particular solution case), and $\underline{z}$ and $\underline{y}$ are set equal to $\underline{\alpha}$ and $-J_w^{3* -1} \underline{\alpha}$ respectively in another case (homogeneous solution case), then we can determine $\underline{y}$ by solving for it in the following equation:

$$J_w^{3*} \underline{y} = \underline{z}, \quad (4.23)$$

where $\underline{y}$ and $\underline{z}$ are 6x1 vectors. If $\underline{y}$ and $\underline{z}$ are also partitioned into two 3x1 vectors, namely $\underline{y}_1$ and $\underline{y}_2$, and $\underline{z}_1$ and $\underline{z}_2$ respectively, then it can be shown [3] that $\underline{y}_1$ and $\underline{y}_2$ are determined as:

$$\underline{y}_1 = J_{w,1}^{3* -1} \underline{z}_1, \quad (4.24)$$

and

$$\underline{y}_2 = J_{w,3}^{3* -1} (\underline{z}_2 - J_{w,2}^{3*} \underline{y}_1) \quad . \quad (4.25)$$

Knowing $\underline{y}_1$ and $\underline{y}_2$ means that we already know $\dot{\underline{\theta}}_p$ and $\dot{\underline{\theta}}_h$.

4.4.3 DETERMINING THE LEAST NORM SOLUTION

Recall from equation (2.17) in chapter 2 that the least norm solution was expressed in terms of $\dot{\underline{\theta}}_p$ and $\dot{\underline{\theta}}_h$. In this section we will get a closed form solution for the least norm solution using equations (4.24) and (4.25) above, and equations (2.10) and (2.12) from chapter 2. We will concentrate on getting the least norm solution for two cases in the following analysis: by eliminating the first column of the wrist Jacobian for one case, and the second column of the wrist Jacobian for another case.

4.4.3.1 ELIMINATING THE FIRST COLUMN

The particular solution for $\dot{\underline{\theta}}$ in closed form can be expressed as follows:

$$\dot{\theta}_{p1} = 0 \quad . \quad (4.26)$$

$$\begin{bmatrix} \dot{\theta}_{p2} \\ \dot{\theta}_{p4} \end{bmatrix} = (1/\Delta) \begin{bmatrix} a_4 C_4 & a_4 S_4 \\ a_2 + a_4 C_3 C_4 & a_4 C_3 S_4 \end{bmatrix} \begin{bmatrix} v_{w,x}^3 \\ v_{w,y}^3 \end{bmatrix} \quad , \quad (4.27)$$

where $\Delta = -a_2 a_4 S_4$,

$$\dot{\theta}_{p3} = [(a_4 S_3 S_4) \dot{\theta}_{p2} - v_{w,z}^3] / a_4 C_4 \quad . \quad (4.28)$$

$$\begin{bmatrix} \dot{\theta}_{p5} \\ \dot{\theta}_{p6} \\ \dot{\theta}_{p7} \end{bmatrix} = (1/S_6) \begin{bmatrix} C_4 C_5 C_6 - S_4 S_6 & S_4 C_5 C_6 + C_4 S_6 & -S_5 C_6 \\ C_4 S_5 S_6 & S_4 S_5 S_6 & C_5 S_6 \\ C_4 C_5 & S_4 C_5 & -S_5 \end{bmatrix} \begin{bmatrix} \omega_{w,x}^3 - S_3 \dot{\theta}_{p2} \\ \omega_{w,y}^3 - \dot{\theta}_{p3} \\ \omega_{w,z}^3 + C_3 \dot{\theta}_{p2} - \dot{\theta}_{p4} \end{bmatrix} \quad (4.29)$$

The vector $\dot{\theta}_h$ can be obtained by setting the element $\dot{\theta}_{h1} = 1$, and the rest of the elements of $\dot{\theta}_h$ are obtained from equations (4.27), (4.28) and (4.29) by replacing $\dot{\theta}_{pi}$ with $\dot{\theta}_{hi}$. The i^{th} element of the wrist velocity vector is also replaced by the i^{th} element of vector $-\underline{\alpha}$. The following equations can then be shown:

$$\dot{\theta}_{h1} = 1 \quad (4.30)$$

$$\begin{bmatrix} \dot{\theta}_{h2} \\ \dot{\theta}_{h4} \end{bmatrix} = (1/\Delta) \begin{bmatrix} a_4 C_4 & a_4 S_4 \\ a_2 + a_4 C_3 C_4 & a_4 C_3 S_4 \end{bmatrix} \begin{bmatrix} -(a_4 S_2 S_3 S_4 + a_2 C_2 S_3) \\ a_4 S_2 S_3 C_4 \end{bmatrix} \quad (4.31)$$

$$\dot{\theta}_{h3} = [(a_4 S_3 S_4) \dot{\theta}_{h2} - a_4 (S_2 C_3 S_4 + C_2 C_4) - a_2 C_2 C_3] / a_4 C_4 \quad (4.32)$$

$$\begin{bmatrix} \dot{\theta}_{h5} \\ \dot{\theta}_{h6} \\ \dot{\theta}_{h7} \end{bmatrix} = (1/S_6) \begin{bmatrix} C_4 C_5 C_6 - S_4 S_6 & S_4 C_5 C_6 + C_4 S_6 & -S_5 C_6 \\ C_4 S_5 S_6 & S_4 S_5 S_6 & C_5 S_6 \\ C_4 C_5 & S_4 C_5 & -S_5 \end{bmatrix} \begin{bmatrix} S_2 C_3 - S_3 \dot{\theta}_{h2} \\ -C_2 - \dot{\theta}_{h3} \\ S_2 S_3 + C_3 \dot{\theta}_{h2} - \dot{\theta}_{h4} \end{bmatrix} \quad (4.33)$$

4.4.3.2 ELIMINATING THE SECOND COLUMN

By eliminating the second column of J_w^3 , the particular solution for $\underline{\theta}$ in closed form can be expressed as follows:

$$\dot{\theta}_{p2} = 0 \quad (4.34)$$

$$\begin{bmatrix} \dot{\theta}_{p1} \\ \dot{\theta}_{p3} \\ \dot{\theta}_{p4} \end{bmatrix} = (1/\Delta) \begin{bmatrix} a_4^2 C_4^2 v_{w,x}^3 + a_4^2 C_4 S_4 v_{w,y}^3 \\ \{ a_4 C_4 (a_4 S_2 C_3 S_4 + a_4 C_2 C_4 + a_2 C_2 C_3) v_{w,x}^3 \\ + a_4 S_4 (a_4 S_2 C_3 S_4 + a_4 C_2 C_4 + a_2 C_2 C_3) v_{w,y}^3 - a_2 a_4 C_2 C_4 S_3 v_{w,z}^3 \} \\ a_4^2 S_2 S_3 C_4^2 v_{w,x}^3 + a_4 C_4 (a_4 S_2 S_3 S_4 + a_2 C_2 S_3) v_{w,y}^3 \end{bmatrix} \quad (4.35)$$

where $\Delta = a_4^2 a_2 C_2 C_4^2 S_3$.

$$\begin{bmatrix} \dot{\theta}_{p5} \\ \dot{\theta}_{p6} \\ \dot{\theta}_{p7} \end{bmatrix} = (1/S_6) \begin{bmatrix} -S_4 S_6 + C_4 C_5 C_6 & C_4 S_6 + S_4 C_5 C_6 & -C_6 S_5 \\ C_4 S_5 S_6 & S_4 S_5 S_6 & C_5 S_6 \\ C_4 C_5 & S_4 C_5 & -S_5 \end{bmatrix} \begin{bmatrix} \omega_{w,x}^3 + S_2 C_3 \dot{\theta}_{p1} \\ \omega_{w,y}^3 - C_2 \dot{\theta}_{p1} - \dot{\theta}_{p3} \\ \omega_{w,z}^3 + S_2 S_3 \dot{\theta}_{p1} - \dot{\theta}_{p4} \end{bmatrix} \quad (4.36)$$

Again if we set $\dot{\theta}_{h2} = 1$, replace $\dot{\theta}_{pi}$ with $\dot{\theta}_{hi}$, and replace the i^{th} element of the wrist velocity vector with the i^{th} element of the vector $-\underline{\alpha}$, we should get the following solution for $\dot{\theta}_{hi}$:

$$\dot{\theta}_{h2} = 1 \quad (4.37)$$

$$\begin{bmatrix} \dot{\theta}_{h1} \\ \dot{\theta}_{h3} \\ \dot{\theta}_{h4} \end{bmatrix} = (1/\Delta) \begin{bmatrix} -a_4^3 C_4^2 C_3 S_4 + a_4^2 C_4 S_4 (a_4 C_3 C_4 + a_2) \\ \{ -a_4^2 C_3 S_4 C_4 (a_4 S_2 C_3 S_4 + a_4 C_2 C_4 + a_2 C_2 C_3) \\ + a_4 S_4 (a_4 C_3 C_4 + a_2) (a_4 S_2 C_3 S_4 + a_4 C_2 C_4 + a_2 C_2 C_3) + a_2 a_4^2 C_2 S_3^2 C_4 S_4 \} \\ -a_4^3 S_2 C_3 S_3 C_4^2 S_4 + a_4 C_4 (a_4 C_3 C_4 + a_2) (a_4 S_2 S_3 S_4 + a_2 C_2 S_3) \end{bmatrix} \quad (4.38)$$

$$\begin{bmatrix} \dot{\theta}_{h5} \\ \dot{\theta}_{h6} \\ \dot{\theta}_{h7} \end{bmatrix} = (1/S_6) \begin{bmatrix} -S_4 S_6 + C_4 C_5 C_6 & C_4 S_6 + S_4 C_5 C_6 & -C_6 S_5 \\ C_4 S_5 S_6 & S_4 S_5 S_6 & C_5 S_6 \\ C_4 C_5 & S_4 C_5 & -S_5 \end{bmatrix} \begin{bmatrix} -S_3 + S_2 C_3 \dot{\theta}_{h1} \\ -C_2 \dot{\theta}_{h1} - \dot{\theta}_{h3} \\ C_3 + S_2 S_3 \dot{\theta}_{h1} - \dot{\theta}_{h4} \end{bmatrix} \quad (4.39)$$

4.4.4 NULL SPACE SOLUTION

The null space solution, which is the projection of the gradient of the performance criterion $H(\theta)$ onto the null space of the wrist Jacobian (see chapter 2), can be found as follows:

$$K(I - J^+J) \nabla H = K[\nabla H^T \dot{\theta}_h / \dot{\theta}_h^T \dot{\theta}_h] \dot{\theta}_h \quad (4.40)$$

where ∇H is the gradient of the performance criterion H . K is a scalar constant whose value is limited by the bounds on the joint velocities.

4.5 KINEMATIC CONTROL FOR HIGHER EFFICIENCY

Recall that in chapter 3 Efficiency was defined as the ability of the manipulator to acquire certain configurations that require the minimal joint velocities that can result in the minimum overall movement of the joints. A performance criterion based on MVR was defined in chapter 3. A control scheme that uses the gradient of the MVR criterion is applied to the LTM. An i^{th} element of the gradient vector of the linear part of MVR in the direction of motion can be written as:

$$\partial H_{\text{MVR,L}} / \partial \theta_i = -2 (\mathbf{J}_{\text{w,v,L}}^T \mathbf{z}_L)^T [\partial (\mathbf{J}_{\text{w,v,L}}^T \mathbf{z}_L) / \partial \theta_i] \quad , \quad (4.41)$$

where $\mathbf{J}_{\text{w,v,L}}^T \mathbf{z}_L$ is a 7x1 column vector, whose elements are given as:

$$\mathbf{J}_{\text{w,v,L}}^T \mathbf{z}_L[1] = -\gamma_1^{1/2} p_{5_y} \epsilon_1^{-1/2} z_1 + \gamma_2^{1/2} p_{5_x} \epsilon_1^{-1/2} z_2 \quad , \quad (4.42)$$

$$\mathbf{J}_{\text{w,v,L}}^T \mathbf{z}_L[2] = \gamma_1^{1/2} C_1 p_{5_z} \epsilon_2^{-1/2} z_1 + \gamma_2^{1/2} S_1 p_{5_z} \epsilon_2^{-1/2} z_2 + \gamma_3^{1/2} (-S_1 p_{5_y} - C_1 p_5) \epsilon_2^{-1/2} z_3 \quad , \quad (4.43)$$

$$\begin{aligned} \mathbf{J}_{\text{w,v,L}}^T \mathbf{z}_L[3] = & \gamma_1^{1/2} [S_1 S_2 (p_{5_z} + a_2 S_2) - C_2 (p_{5_y} - a_2 C_2 S_1)] \epsilon_3^{-1/2} z_1 \\ & + \gamma_2^{1/2} [-C_1 S_2 (p_{5_z} + a_2 S_2) + C_2 (p_{5_x} - a_2 C_2 C_1)] \epsilon_3^{-1/2} z_2 \\ & + \gamma_3^{1/2} [C_1 S_2 (p_{5_y} - a_2 C_2 S_1) - S_1 S_2 (p_{5_x} - a_2 C_2 C_1)] \epsilon_3^{-1/2} z_3 \quad , \quad (4.44) \end{aligned}$$

$$\begin{aligned} \mathbf{J}_{\text{w,v,L}}^T \mathbf{z}_L[4] = & \gamma_1^{1/2} [(S_1 C_2 S_3 - C_1 C_3) (p_{5_z} + a_2 S_2) + (S_2 S_3) (p_{5_y} - a_2 C_2 S_1)] \epsilon_4^{-1/2} z_1 \\ & + \gamma_2^{1/2} [-(C_1 C_2 S_3 + S_1 C_3) (p_{5_z} + a_2 S_2) - (S_2 S_3) (p_{5_x} - a_2 C_2 C_1)] \epsilon_4^{-1/2} z_2 \\ & + \gamma_3^{1/2} [(C_1 C_2 S_3 + S_1 C_3) (p_{5_y} - a_2 C_2 S_1) - (p_{5_x} - a_2 C_2 C_1) (S_1 C_2 S_3 - C_1 C_3)] \epsilon_4^{-1/2} z_3 \quad (4.45) \end{aligned}$$

$$\mathbf{J}_{\text{w,v,L}}^T \mathbf{z}_L[5] = \mathbf{J}_{\text{w,v,L}}^T \mathbf{z}_L[6] = \mathbf{J}_{\text{w,v,L}}^T \mathbf{z}_L[7] = 0 \quad . \quad (4.46)$$

The i^{th} element of the gradient vector of MVR with respect to angular motion can be deduced as:

$$\partial H_{\text{MVR,A}} / \partial \theta_i = -2 (\mathbf{J}_{\mathbf{w}_{v,A}}^T \mathbf{z}_A)^T [\partial (\mathbf{J}_{\mathbf{w}_{v,A}}^T \mathbf{z}_A) / \partial \theta_i] \quad , \quad (4.47)$$

where $\mathbf{J}_{\mathbf{w}_{v,A}}^T \mathbf{z}_A$ is a 7x1 vector whose entries are shown below:

$$\mathbf{J}_{\mathbf{w}_{v,A}}^T \mathbf{z}_A[1] = \gamma_6^{1/2} \epsilon_1^{-1/2} z_6 \quad , \quad (4.48)$$

$$\mathbf{J}_{\mathbf{w}_{v,A}}^T \mathbf{z}_A[2] = -\gamma_4^{1/2} S_1 \epsilon_2^{-1/2} z_4 + \gamma_5^{1/2} C_1 \epsilon_2^{-1/2} z_5 \quad , \quad (4.49)$$

$$\mathbf{J}_{\mathbf{w}_{v,A}}^T \mathbf{z}_A[3] = \gamma_4^{1/2} C_1 S_2 \epsilon_3^{-1/2} z_4 + \gamma_5^{1/2} S_1 S_2 \epsilon_3^{-1/2} z_5 + \gamma_6^{1/2} C_2 \epsilon_3^{-1/2} z_6 \quad , \quad (4.50)$$

$$\begin{aligned} \mathbf{J}_{\mathbf{w}_{v,A}}^T \mathbf{z}_A[4] = & \gamma_4^{1/2} (C_1 C_2 S_3 + S_1 C_3) \epsilon_4^{-1/2} z_4 + \gamma_5^{1/2} (S_1 C_2 S_3 - C_1 C_3) \epsilon_4^{-1/2} z_5 \\ & + \gamma_6^{1/2} (-S_2 S_3) \epsilon_4^{-1/2} z_6 \quad , \end{aligned} \quad (4.51)$$

$$\begin{aligned} \mathbf{J}_{\mathbf{w}_{v,A}}^T \mathbf{z}_A[5] = & -\gamma_4^{1/2} (-S_6 C_7 n_x + S_6 S_7 s_x + C_6 a_x) \epsilon_5^{-1/2} z_4 \\ & -\gamma_5^{1/2} (-S_6 C_7 n_y + S_6 S_7 s_y + C_6 a_y) \epsilon_5^{-1/2} z_5 \\ & -\gamma_6^{1/2} (-S_6 C_7 n_z + S_6 S_7 s_z + C_6 a_z) \epsilon_5^{-1/2} z_6 \quad , \end{aligned} \quad (4.52)$$

$$\begin{aligned} \mathbf{J}_{\mathbf{w}_{v,A}}^T \mathbf{z}_A[6] = & \gamma_4^{1/2} (S_7 n_x + C_7 s_x) \epsilon_6^{-1/2} z_4 + \gamma_5^{1/2} (S_7 n_y + C_7 s_y) \epsilon_6^{-1/2} z_5 \\ & + \gamma_6^{1/2} (S_7 n_z + C_7 s_z) \epsilon_6^{-1/2} z_6 \quad , \end{aligned} \quad (4.53)$$

$$\begin{aligned} \mathbf{J}_{\mathbf{w}_{v,A}}^T \mathbf{z}_A[7] = & \gamma_4^{1/2} [(p_x - p_{5_x}) / d_7] \epsilon_7^{-1/2} z_4 + \gamma_5^{1/2} [(p_y - p_{5_y}) / d_7] \epsilon_7^{-1/2} z_5 \\ & + \gamma_6^{1/2} [(p_z - p_{5_z}) / d_7] \epsilon_7^{-1/2} z_6 \quad . \end{aligned} \quad (4.54)$$

4.6 KINEMATIC CONTROL FOR HIGHER MECHANICAL ADVANTAGE

The MMA performance criterion that is based on minimizing the joint torques for a certain external force applied at the wrist was defined in chapter 3. A control scheme using the gradient of MMA is developed for the LTM. The i^{th} element of the

gradient vector of MMA in the direction of the applied force components is defined as:

$$\partial H_{MMA,f} / \partial \theta_i = 2 (J_{w_{m,f}}^T \underline{u}_{m,f})^T [\partial (J_{w_{m,f}}^T \underline{u}_{m,f}) / \partial \theta_i] \quad , \quad (4.55)$$

where $J_{w_{m,f}}^T \underline{u}_{m,f}$ is a 7x1 column vector, whose components are as follows:

$$J_{w_{m,f}}^T \underline{u}_{m,f}[1] = [-f_x p_{5_y} \beta_1^{1/2} + f_y p_{5_x} \beta_1^{1/2}] [1 / \sqrt{E_{m,f}^T E_{m,f}}] \quad , \quad (4.56)$$

$$J_{w_{m,f}}^T \underline{u}_{m,f}[2] = \{ f_x C_1 p_{5_z} \beta_2^{1/2} + f_y S_1 p_{5_z} \beta_2^{1/2} + f_z (-S_1 p_{5_y} - C_1 p_{5_x}) \beta_2^{1/2} \} [1 / \sqrt{E_{m,f}^T E_{m,f}}] \quad (4.57)$$

$$\begin{aligned} J_{w_{m,f}}^T \underline{u}_{m,f}[3] = & \{ f_x \beta_3^{1/2} [S_1 S_2 (p_{5_z} + a_2 S_2) - C_2 (p_{5_y} - a_2 C_2 S_1)] \\ & + f_y \beta_3^{1/2} [-C_1 S_2 (p_{5_z} + a_2 S_2) + C_2 (p_{5_x} - a_2 C_2 C_1)] \\ & + f_z \beta_3^{1/2} [C_1 S_2 (p_{5_y} - a_2 C_2 S_1) - S_1 S_2 (p_{5_x} - a_2 C_2 C_1)] \} [1 / \sqrt{E_{m,f}^T E_{m,f}}] \end{aligned} \quad (4.58)$$

$$\begin{aligned} J_{w_{m,f}}^T \underline{u}_{m,f}[4] = & \{ f_x \beta_4^{1/2} [(S_1 C_2 S_3 - C_1 C_3) (p_{5_z} + a_2 S_2) + S_2 S_3 (p_{5_y} - a_2 C_2 S_1)] \\ & + f_y \beta_4^{1/2} [-(C_1 C_2 S_3 + S_1 C_3) (p_{5_z} + a_2 S_2) - S_2 S_3 (p_{5_x} - a_2 C_2 C_1)] \\ & + f_z \beta_4^{1/2} [(C_1 C_2 C_3 + S_1 C_3) (p_{5_y} - a_2 C_2 S_1) - (S_1 C_2 S_3 - C_1 C_3) (p_{5_x} - a_2 C_2 C_1)] \} \\ & \times [1 / \sqrt{E_{m,f}^T E_{m,f}}] \end{aligned} \quad (4.59)$$

$$J_{w_{m,f}}^T \underline{u}_{m,f}[5] = J_{w_{m,f}}^T \underline{u}_{m,f}[6] = J_{w_{m,f}}^T \underline{u}_{m,f}[7] = 0 \quad . \quad (4.60)$$

The i^{th} element of the gradient vector of MMA in the direction of the applied moment is defined as:

$$\partial H_{MMA,m} / \partial \theta_i = 2 (J_{w_{m,m}}^T \underline{u}_{m,m})^T [\partial (J_{w_{m,m}}^T \underline{u}_{m,m}) / \partial \theta_i] \quad , \quad (4.61)$$

where $J_{w_{m,m}}^T \underline{u}_{m,m}$ is a 7x1 column vector, whose components are as follows:

$$J_{w_{m,m}}^T \underline{u}_{m,m}[1] = [\beta_1^{1/2} m_z] [1/\sqrt{F_{m,m}^T F_{m,m}}] , \quad (4.62)$$

$$J_{w_{m,m}}^T \underline{u}_{m,m}[2] = [-S_1 m_x \beta_2^{1/2} + C_1 m_y \beta_2^{1/2}] [1/\sqrt{F_{m,m}^T F_{m,m}}] , \quad (4.63)$$

$$J_{w_{m,m}}^T \underline{u}_{m,m}[3] = [m_x C_1 S_2 \beta_3^{1/2} + m_y S_1 S_2 \beta_3^{1/2} + m_z C_2 \beta_3^{1/2}] [1/\sqrt{F_{m,m}^T F_{m,m}}] , \quad (4.64)$$

$$J_{w_{m,m}}^T \underline{u}_{m,m}[4] = \{ m_x (C_1 C_2 S_3 + S_1 C_3) \beta_4^{1/2} + m_y (S_1 C_2 S_3 - C_1 C_3) \beta_4^{1/2} + m_z (-S_2 S_3) \beta_4^{1/2} \} [1/\sqrt{F_{m,m}^T F_{m,m}}] , \quad (4.65)$$

$$J_{w_{m,m}}^T \underline{u}_{m,m}[5] = \{ -m_x [-S_6 C_7 n_x + S_6 S_7 s_x + C_6 a_x] \beta_5^{1/2} - m_y [-S_6 C_7 n_y + S_6 S_7 s_y + C_6 a_y] \beta_5^{1/2} - m_z [-S_6 C_7 n_z + S_6 S_7 s_z + C_6 a_z] \beta_5^{1/2} \} [1/\sqrt{F_{m,m}^T F_{m,m}}] , \quad (4.66)$$

$$J_{w_{m,m}}^T \underline{u}_{m,m}[6] = \{ [S_7 n_x + C_7 s_x] \beta_6^{1/2} + [S_7 n_y + C_7 s_y] \beta_6^{1/2} + [S_7 n_z + C_7 s_z] \beta_6^{1/2} \} \times [1/\sqrt{F_{m,m}^T F_{m,m}}] , \quad (4.67)$$

$$J_{w_{m,m}}^T \underline{u}_{m,m}[7] = \{ m_x [(p_x - p_{5_x})/d_7] \beta_7^{1/2} + m_y [(p_y - p_{5_y})/d_7] \beta_7^{1/2} + m_z [(p_z - p_{5_z})/d_7] \beta_7^{1/2} \} [1/\sqrt{F_{m,m}^T F_{m,m}}] . \quad (4.68)$$

CHAPTER 5

KINEMATIC CONTROL SOFTWARE FOR THE LTM

This chapter presents a summary of the major parts of the computer code developed to move the arm on a certain specified trajectory. For a more detailed reference on the computer code, the reader can refer to the computer program [9]. The following components of the computer code are addressed in this chapter:

- Time Coordinated Trajectories in Cartesian Space
- Generation of the Euler Angles
- Generation of the Hand Error And the Wrist Error
- Forward Kinematics (Position And Orientation)
- Gradients of Performance Criteria
- Inverse Kinematics

5.1 TIME COORDINATED TRAJECTORIES IN CARTESIAN SPACE

In order to move the end-effector across a certain trajectory we need to specify the starting point as well as the ending point of the trajectory. The position and orientation of the end-effector at these points, which will be referred to as nodes), are specified in terms of homogeneous transformations. There are many ways to move the end-effector from one position to another. To name a few, the end-effector for instance may stop at each node (Figure 5.1) in one case. In another case, the motion between successive nodes can be represented by straight line motion (Figure 5.2). However, both cases are undesirable. In the first case, it is desirable for the end-effector to stop at the last node and not at every node across the trajectory. In the second case, the velocities and the accelerations are discontinuous at the node points. The control scheme used in our research to move the end-effector across a specified trajectory is based on the algorithm developed by Paul [10]. In order to provide for continuity in position, velocity, and acceleration during the transition from one node to another (Figure 5.3), Paul defined the

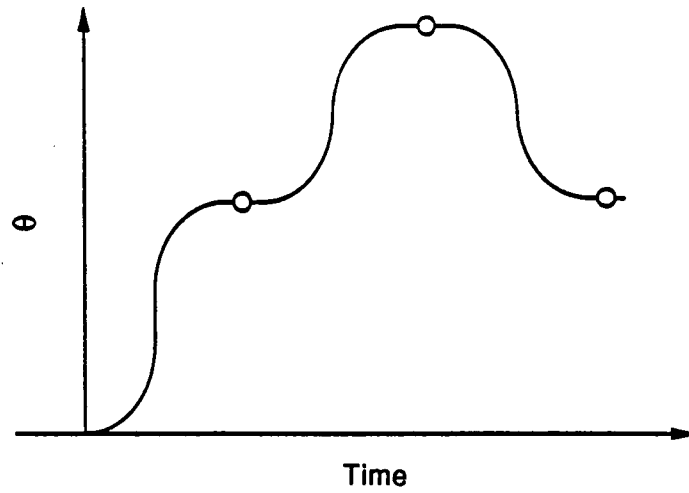


Figure 5.1. Starting and stopping motion between successive nodes.

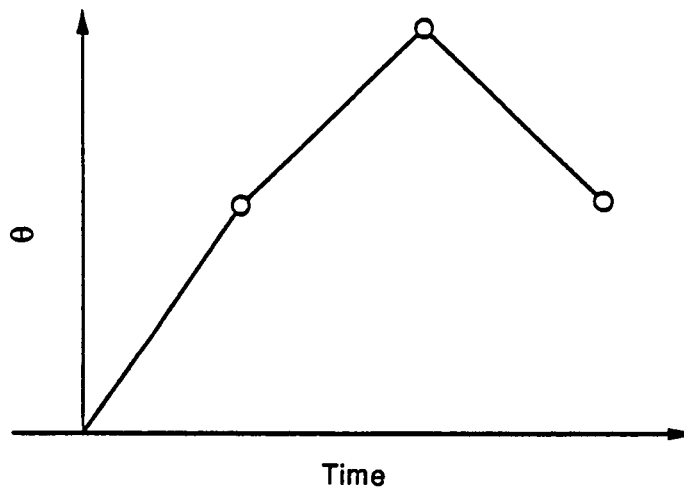


Figure 5.2. Linear motion between successive nodes.

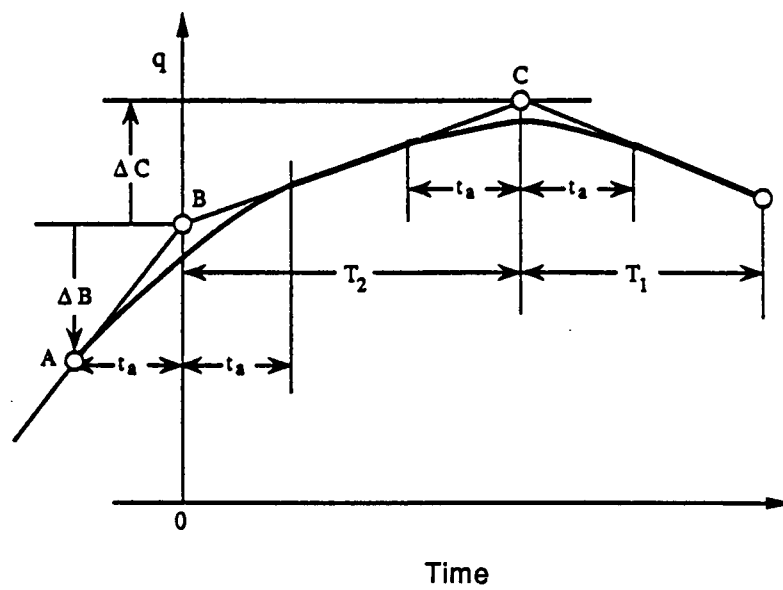


Figure 5.3. Path transitions.

(Paul R.P. "Robot Manipulators:Mathematics, Programming, and Control",
Cambridge: MIT Press)

position of the end-effector as a quartic polynomial which is a function of time:

$$q = a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 \quad . \quad (5.1)$$

Since we have six boundary conditions at both ends of the transition between two successive nodes, a fifth order polynomial may be used. However, because of the symmetry of transition, it can be proven that a quartic polynomial is sufficient. When the first three derivatives of the quartic polynomial are taken, and the boundary conditions are applied, the equation describing the end-effector motion can be shown as:

$$q = [(\Delta C t_{acc}/T_1 + \Delta B)(2 - h)^2 - 2\Delta B] h + A \quad , \quad (5.2)$$

where

$$\Delta C = C - B \quad ,$$

$$\Delta B = A - B \quad ,$$

$$\text{and } h = (t + t_{acc})/2t_{acc} \quad . \quad (5.3)$$

A , B, and C are end-effector locations shown in Figure 5.3. t is the total move time between B and C, and t_{acc} is the acceleration time. t is divided into several time steps whose total number is referred to as nt . The total number of time steps in the acceleration time (t_{acc}) is denoted by na . Equation (5.2) applies only when the end-effector is in the acceleration or deceleration phase.

After $t = t_{acc}$, the acceleration phase is completed and the end-effector maintains a constant velocity motion. The motion of the end-effector can then be described as:

$$q = h \Delta C + B \quad , \quad (5.4)$$

where

$$h = t/T_1 \quad . \quad (5.5)$$

There are three cases which must be dealt with for any motion between two successive nodes. As seen in the flow chart (Figure 5.4), these are: the zeroth case,

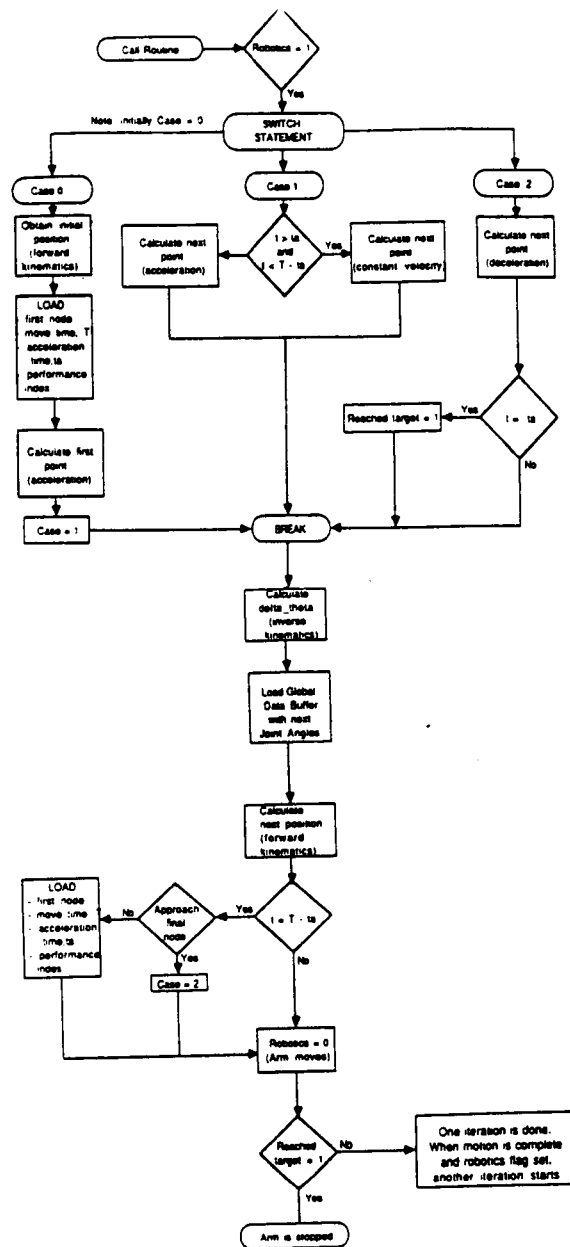


Figure 5.4. Flow chart of the subroutine that determines the desired position of the end-effector.

(Magness, R.B. Robotic Control of the Seven-Degree-of-Freedom NASA Laboratory Telerobotic Manipulator. Master Thesis, University of Tennessee, Knoxville)

the first case, and the second case.

The zeroth case is the case in which the initial current position and orientation of the end-effector are obtained from the initial joint angles supplied to the program interactively. Thus, the zeroth case is chosen only during the first iteration. In Figure 5.3, node A is referred to as the zeroth node (initial node). The Cartesian coordinates for node 1 are assigned to C and not to B. Because the manipulator starts from rest and must accelerate, it must be that A and B have the same cartesian coordinates, and node 1 is therefore assigned to C. ΔB will then be set equal to zero. If the first node is also the last node then we must subtract $2n_a$ from n_t ; otherwise, we subtract n_a from n_t .

Case 1 applies whenever the end-effector is either in acceleration phase or constant velocity motion phase. Case 1 is executed immediately after the first iteration. Again the desired position and orientation (Euler Angles) of the end-effector are calculated. If at $t = -t_{acc}$ the last node is not loaded, then the current position is assigned to A and the coordinate values of C are assigned to B. ΔB and ΔC are then evaluated.

Case 2 is dealt with whenever the end-effector is decelerating to a stop. When the last node is loaded, this node which is assigned to C must actually be node B, and therefore ΔB is set to zero. Figure 5.5 shows the corresponding positions of points A, B, and C on a trajectory that consists of two nodes.

In all these cases, the desired position and orientation of the end-effector (in terms of Euler Angles) are obtained from either equation (5.2) or equation (5.4). The current position and orientation of the end-effector is calculated from the joint angles using the forward kinematic equations. The hand error, which is the difference between the current position and orientation and the desired position and orientation of the end-effector, is then determined. Because the inverse kinematic control scheme uses the wrist error instead of the hand error, the values of the wrist error are obtained from the hand error using the appropriate equations for that. The wrist error is then projected onto the third coordinate frame before

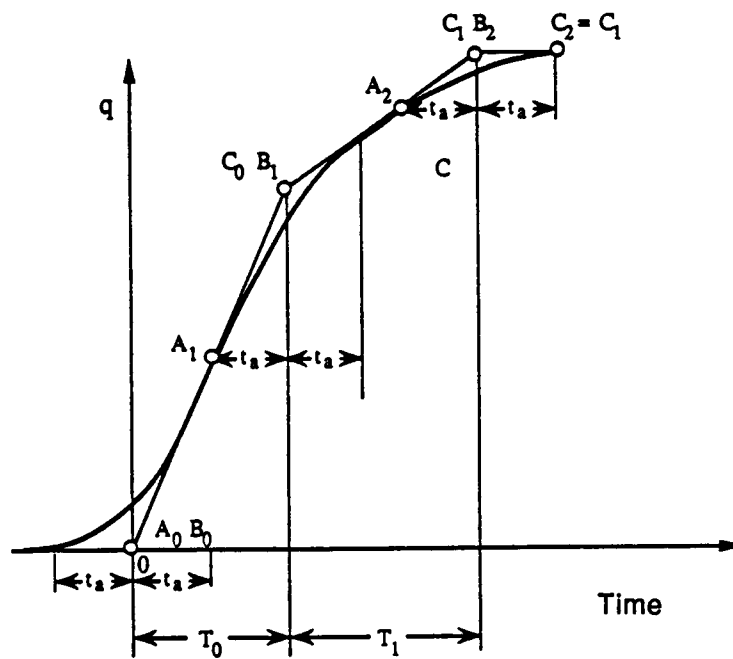


Figure 5.5. Time coordinated trajectory consisting of two nodes.

passing it to the inverse kinematic routine. The inverse kinematic routine then returns the incremental change in the joint angles, $\Delta\theta$. The joint angles are then updated by the amount $\Delta\theta$. The updated values of the joint angles are then passed to the forward kinematic routine where the new current position and orientation of the end-effector are obtained. Also the world coordinates of the manipulator elbow and wrist are obtained. These are sent to graphic routines to draw the LTM arm. Figure 5.6 shows a bubble diagram representing the major routines of the computer software and their interactions.

5.2 GENERATION OF THE EULER ANGLES

In the computer code, the orientation part of the homogeneous transformation matrix is used in obtaining the Euler Angles necessary for the orientation. Since the orientation of the hand is given in base coordinates, the Euler Angles are also expressed in base coordinates. Let the end-effector transformation matrix be represented by:

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} . \quad (5.6)$$

Let the Euler Angles be denoted by ϕ , θ , and ψ . ϕ is a positive rotation about the z-axis, θ is a positive rotation about the new y-axis, and ψ is a positive rotation about the new z-axis (Figure 5.7). It can then be shown that the Euler Angles can be expressed as [10]:

$$\phi = \tan^{-1} [a_y/a_x] , \quad (5.7)$$

$$\theta = \tan^{-1} [(a_x \cos \phi + a_y \sin \phi) / a_z] , \quad (5.8)$$

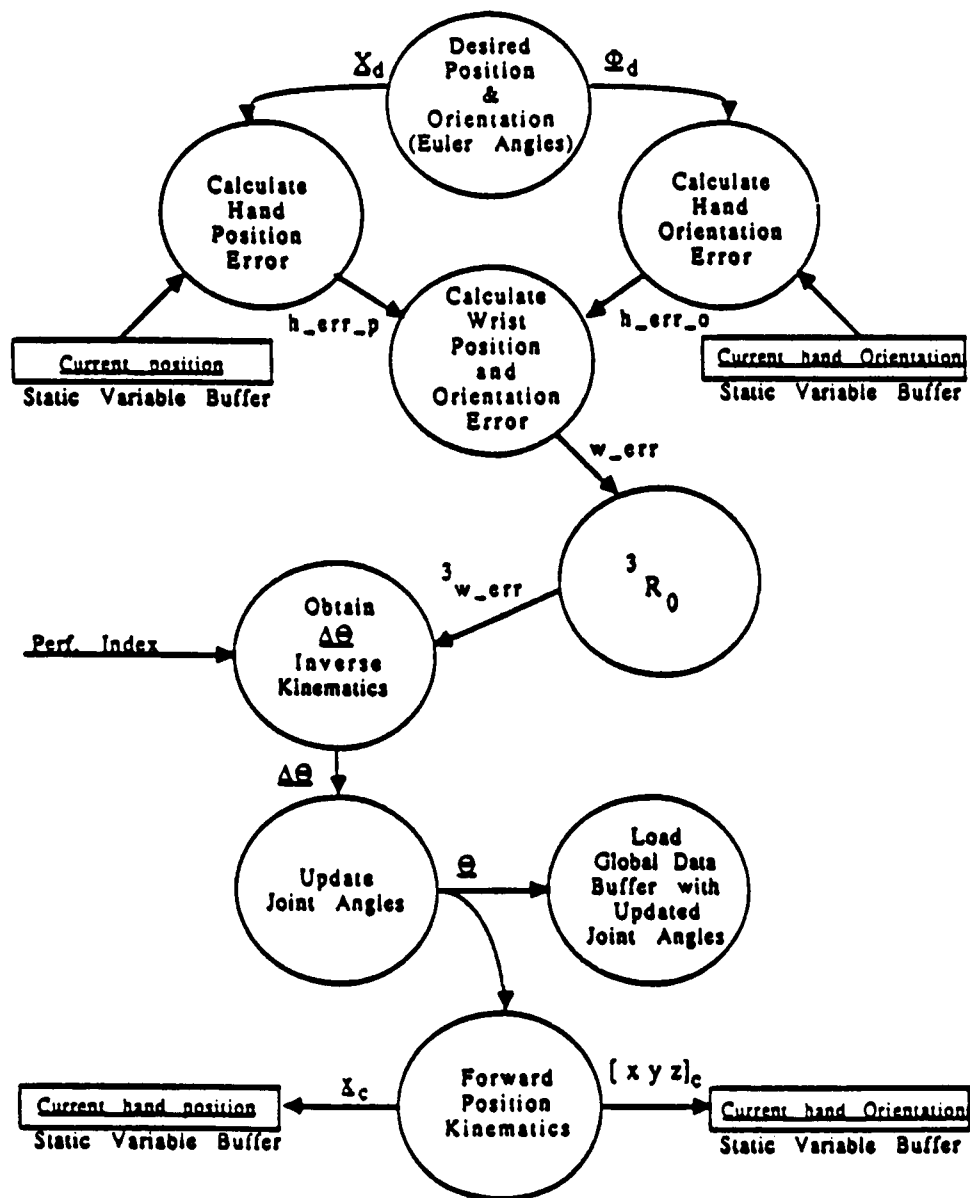


Figure 5.6. A bubble diagram representing the major routines of the computer software.

(Magness, R.B. Robotic Control of the Seven-Degree-of-Freedom NASA Laboratory Telerobotic Manipulator. Master Thesis, University of Tennessee, Knoxville)

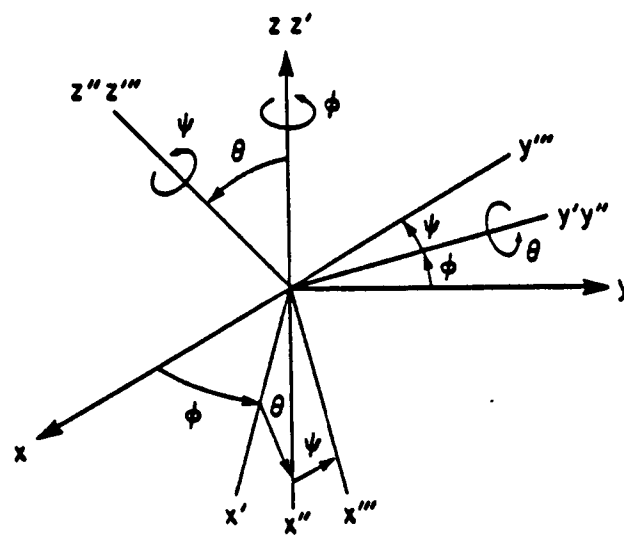


Figure 5.7. The Eulerian angles.

(Paul R.P. "Robot Manipulators: Mathematics, Programming, and Control",
Cambridge: MIT Press)

$$\text{and } \psi = \tan^{-1}[(n_y \cos \phi - n_x \sin \phi)/(o_y \cos \phi - o_x \sin \phi)] \quad . \quad (5.9)$$

Thus having known the orientation matrix of the end-effector, we can easily obtain the Euler Angles using the above equations.

5.3 GENERATION OF THE HAND ERROR AND THE WRIST ERROR

The hand error is the difference between the current and the desired position and orientation of the hand (end-effector), while the wrist error is the difference between the current and the desired position and orientation of the wrist. The positions and orientations are referenced to the base coordinate frame. As the time step approaches zero, these finite changes result in linear and angular velocities if divided by the time increment. Let us represent the desired orientation of the end-effector in terms of the Euler Angles. This orientation matrix can be obtained by the following products of matrices:

$$E(\phi, \theta, \psi) = R(z, \phi) R(y, \theta) R(z, \psi) = \begin{bmatrix} x_d & y_d & z_d \end{bmatrix} \quad , \quad (5.10)$$

where $\begin{bmatrix} x_c & y_c & z_c \end{bmatrix}$ and $\begin{bmatrix} x_d & y_d & z_d \end{bmatrix}$ are the current and desired orientation of the end-effector respectively. The rotation necessary to carry the end-effector coordinate frame from its current orientation to its desired orientation (Figure 5.8) can be derived as follows [9]:

$$d\theta = d\theta_x + d\theta_y + d\theta_z \quad , \quad (5.11)$$

where

$$\begin{aligned} d\theta_x &= (1/2)[y_c \times y_d + z_c \times z_d] \quad , \\ d\theta_y &= (1/2)[x_c \times x_d + z_c \times z_d] \quad , \text{ and} \\ d\theta_z &= (1/2)[x_c \times x_d + y_c \times y_d] \quad . \end{aligned} \quad (5.12)$$

The above three terms represent the orientation error of the hand. The hand error

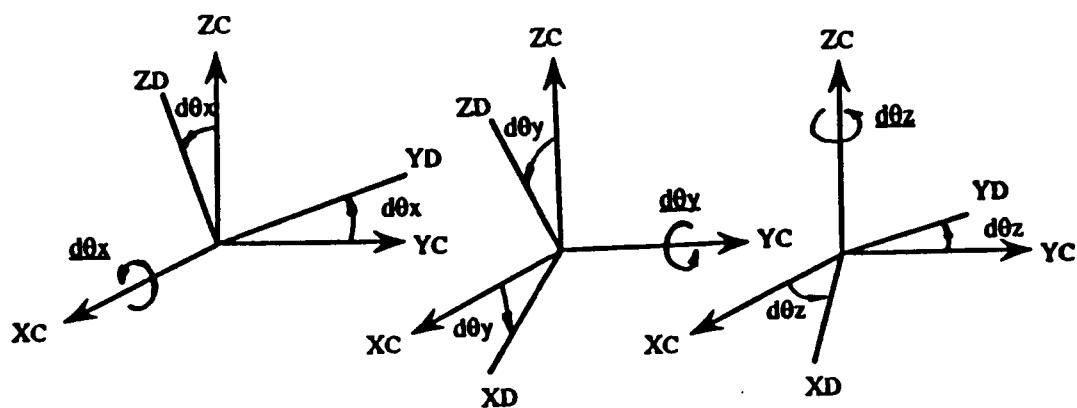


Figure 5.8. Representation of the current and the desired orientation of the end-effector frame.

that corresponds to the linear motion of the hand is given by:

$$\begin{aligned}h \, dx &= x_d - x_c \quad , \\h \, dy &= y_d - y_c \quad , \text{ and} \\h \, dz &= z_d - z_c \quad .\end{aligned}\tag{5.13}$$

The wrist error can be deduced from the hand error using the following relationships:

$$\begin{aligned}\omega_{\text{wrist}} &= \omega_{\text{hand}} \quad , \\ \text{and, } \underline{v}_{\text{wrist}} &= \underline{v}_{\text{hand}} - d_7(\omega_{\text{hand}} \times \underline{z}_{\text{hand}}) \quad ,\end{aligned}\tag{5.14}$$

where d_7 is the length of the most distal link, and $\underline{z}_{\text{hand}}$ is the the z-axis of the hand coordinate frame.

5.4 GRADIENTS OF PERFORMANCE CRITERIA

The routine that determines the gradients of performance criteria is implemented on a module separate from the rest of the program. This routine consists of finding the gradients of performance criteria for the following cases:

- 1) Maximization of MVR in the direction of the motion of the end-effector
(linear motion of the end-effector)
- 2) Maximization of MVR in the direction of the motion of the end-effector
(angular motion of the end-effector)
- 3) Maximization of MMA in the direction of the applied wrist force
- 4) Maximization of MMA in the direction of the applied wrist moment
- 5) Maximization of the Manipulability Measure
(linear part of the wrist Jacobian)
- 6) Maximization of MVR in the direction corresponding to the minimum
singular value (linear motion of the end-effector)

- 7) Maximization of MVR in the direction corresponding to the minimum singular value (angular motion of the end-effector)

The performance criterion MVR, MMA, and the Manipulability Measure are discussed in chapter 3.

5.5 FORWARD POSITION KINEMATICS

In this routine, the current position and orientation of the end-effector and the wrist are calculated with reference to the third coordinate frame and the base frame too. The wrist position and orientation referenced to the third coordinate frame are used in the inverse kinematic calculations, while the wrist position and orientation referenced to the base coordinate frame are passed to graphic routines. Derivations of the equations of the position and the orientation of the end-effector and the wrist with respect to the base frame are presented in appendix B. Appendices A and C provide detailed references on the homogeneous transformation matrices and the relationship between the wrist and the hand velocities respectively.

5.6 INVERSE KINEMATICS OF THE LTM

The Inverse kinematics control scheme is based on the approach outlined by Euler et al.[4] which was discussed in chapters 2 and 3. This approach avoids determining the Moore-pseudo inverse and is therefore less computationally complex. As a reminder, the Jacobian used in the inverse kinematic control scheme is the wrist Jacobian referenced to the third coordinate frame. The inverse kinematic routine is supplied with information such as the wrist error and the gradient of the performance criterion. It returns the incremental change in the joint angles, $\Delta\theta$. If we are interested in a least norm solution (see equation (2.17)), then $\Delta\theta$ is based on the difference between the particular solution and the component of the particular solution in the direction of the homogeneous solution.

However, if a performance criterion is optimized, then $\Delta\theta$ will consist of the least norm solution and the null space solution (see equations 2.4 and 2.18). The direction of the null space solution is obtained from the gradient, while the magnitude of the null space solution is determined from K which is bounded by the joint velocities (see chapter 2). The inverse kinematic control scheme operates as follows [9]:

- If joint angle 4 is between -10 and 10 degrees, then $\underline{\alpha}$ is chosen as column #4
- If joint angle 4 is between -80 and 80 degrees, then $\underline{\alpha}$ is chosen as column #3
- otherwise eliminate column #1

$\underline{\alpha}$ denotes the column of the wrist Jacobian that is dependant on the rest of the columns.

CHAPTER 6

SIMULATION

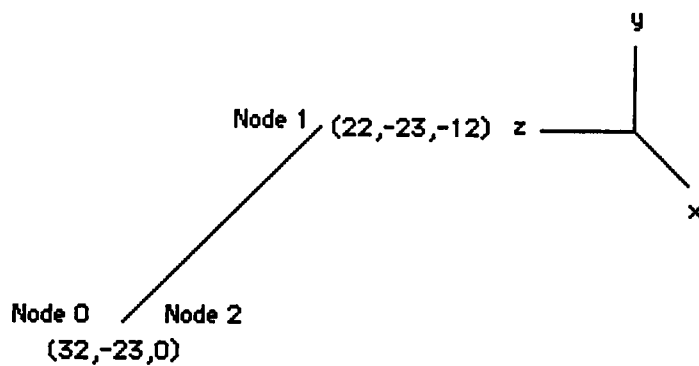
The kinematic control schemes for higher efficiency and mechanical advantage were applied to the NASA Laboratory Telerobotic Manipulator (LTM). Simulations for the LTM using these control schemes were performed using a 3-D solid model of the LTM on the Silicon Graphics workstation. The control software was written in the "C" programming language for higher speed and portability. The LTM manipulator with the coordinate axes attached to its joints is shown in Figure 4.1. The weighting matrices defined in chapter 3 are chosen as identity matrices for the simulations presented in this chapter.

6.1 MAXIMIZATION OF MVR

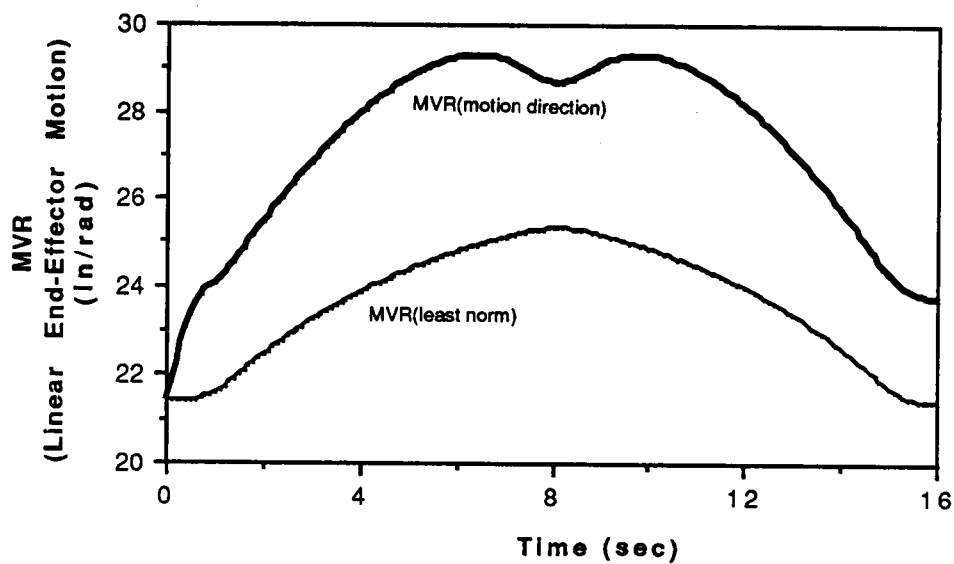
This section presents simulation results for reducing the joint velocities of the robot manipulator by maximizing MVR in the direction of motion of the end-effector or in the direction corresponding to the minimum singular value of the wrist Jacobian.

6.1.1 MAXIMIZATION OF MVR IN THE DIRECTION OF MOTION

Figure 6.1(a) shows a trajectory that consists of two nodes that form a closed path (1 cycle). MVR is defined here as the ratio of the norm of the transformed linear wrist velocity to the norm of the least norm transformed joint velocities. MVR in Figure 6.1(b) was obtained using two control schemes: one that increases MVR in the direction of motion of the end-effector, and one that uses the least norm control scheme. As shown in the plot, the redundancy of the LTM arm played a major role in improving MVR, and thus in obtaining joint configurations that resulted in lower joint velocities. The least norm control scheme which minimizes the instantaneous joint velocities, resulted in configurations that required higher



(a) End-effector trajectory motion



(b) MVR vs. time

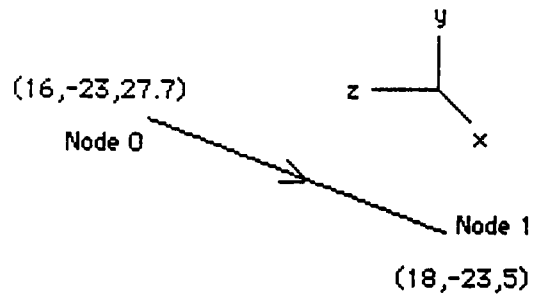
Figure 6.1. MVR in the direction of motion vs. time .
(linear end-effector motion case)

minimum joint velocities. As a result the mobility of the manipulator remained poor or got worse. It should be noted that when the end-effector returned to its initial position, the least norm scheme resulted in almost the same MVR value for both the starting position and the ending position of the trajectory. As expected, the control scheme that improved MVR in the direction of motion resulted in a much higher MVR value than it did when initially the manipulator started from rest. Even though the end-effector is returning to the same position, the joint configuration of the manipulator arm is different from the starting configuration. As we optimized MVR along the trajectory, the manipulator configuration at the end position was substantially improved because of the utilization of the extra degree of joint redundancy.

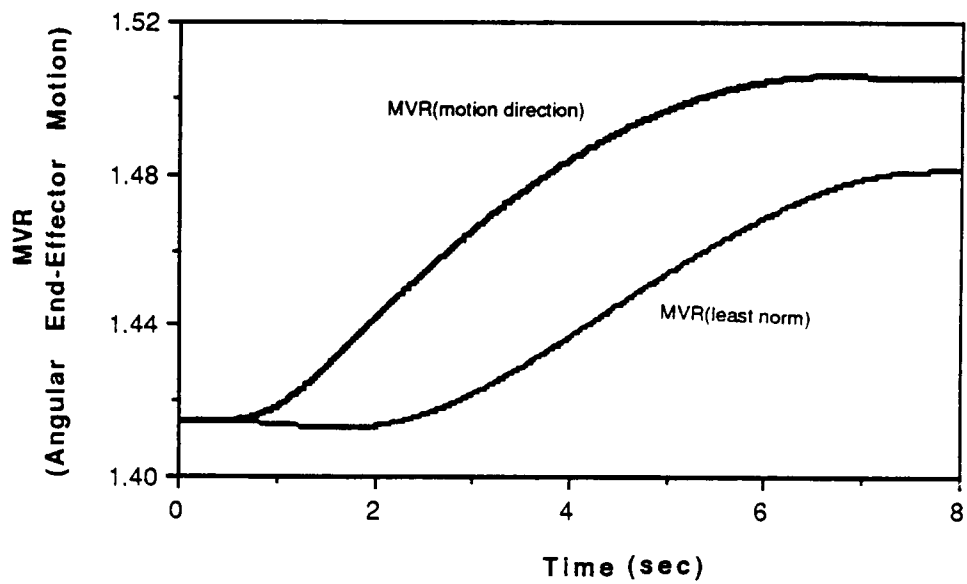
Figure 6.2(a) shows a trajectory that consists of a single node (1 cycle). Figure 6.2(b) shows a plot of MVR versus time for this trajectory using the least norm control scheme and the control scheme that improves MVR in the direction of motion. In this case, MVR was defined as the norm of the transformed wrist angular velocity to the norm of the least norm transformed joint velocities. Again optimization using the null space resulted in better MVR performance than the least norm solution.

6.1.2 MAXIMIZATION OF MVR IN THE DIRECTION CORRESPONDING TO THE MINIMUM SINGULAR VALUE

Figure 6.3(a) shows a closed loop trajectory that consists of two nodes (1 cycle). MVR, which in this case is based on the upper three rows of the wrist Jacobian that correspond to the linear motion of the joints, is improved in the direction of the minimum singular value of the Manipulator Velocity Ratio Ellipsoid. The plots of MVR versus time shown in Figure 6.3(b) are obtained using the control scheme that improves MVR in the direction corresponding to the minimum singular value, and the least norm control scheme. Comparing the results of both schemes, it is shown that as expected the least norm scheme results in lower values for MVR and therefore higher joint velocities. The starting and ending position of the

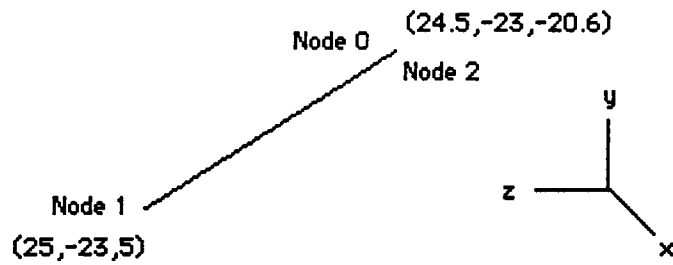


(a) End-effector trajectory motion

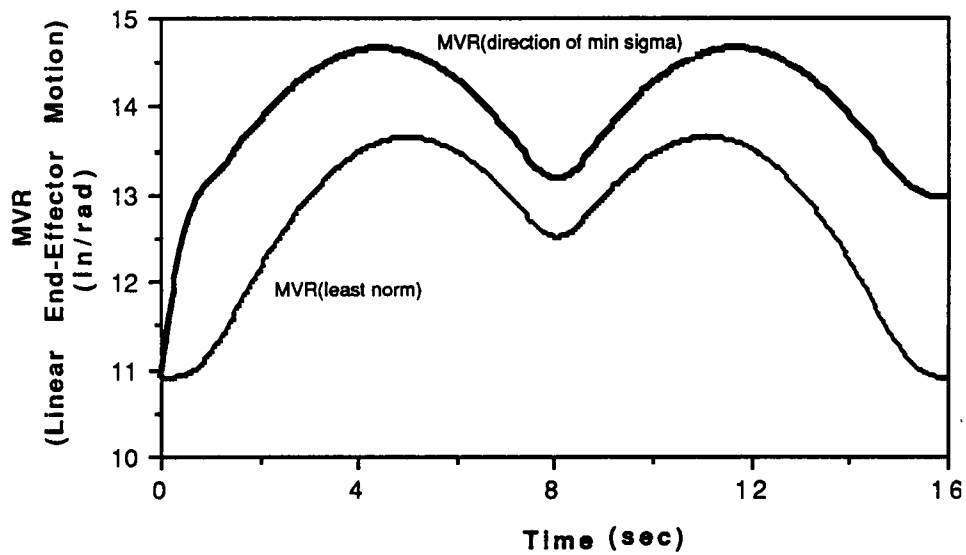


(b) MVR vs. time

Figure 6.2. MVR in the direction of motion vs. time.
(angular end-effector motion case)



(a) End-effector trajectory motion



(b) MVR vs. time

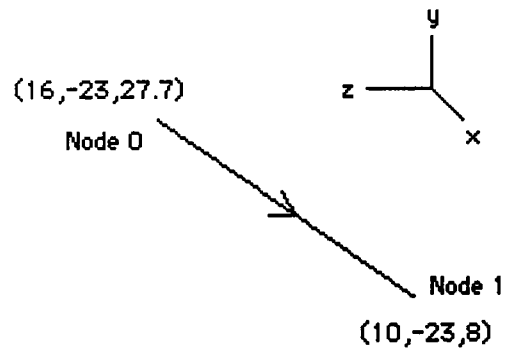
Figure 6.3. MVR in the direction corresponding to the minimum singular value vs. time. (linear end-effector motion case)

end-effector result in almost the same value of MVR (higher joint velocities) using the least norm scheme, while the optimization scheme considerably improves the manipulator joint configuration, and thus results in a higher value for MVR (lower joint velocities) in the direction corresponding to the minimum singular value at the stopping position.

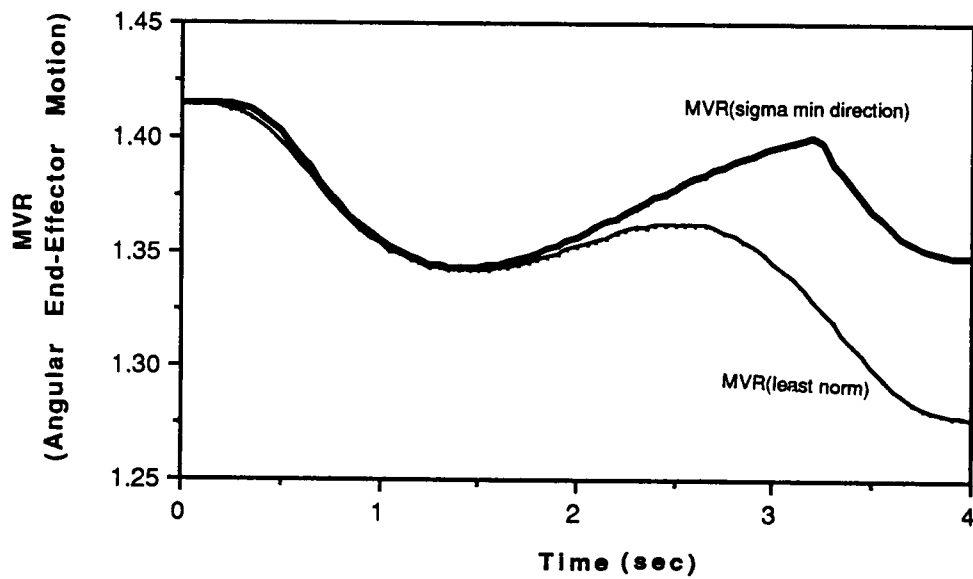
Figure 6.4(a) shows a trajectory that consists of one node (1 cycle). In this case, MVR is based on the lower three rows of the wrist Jacobian that correspond to the rotational motion of the joints. Figure 6.4(b) shows MVR versus time using the least norm solution and the null space solution where the utilization of the redundancy of the manipulator resulted in higher MVR in the direction of the minimum singular value. In this case the orientation of the end-effector was changing throughout the entire motion.

6.2 MAXIMIZATION OF MMA

Figures 6.5(b) and 6.6(b) show the plots of MMA versus time for the least norm solution and for the case where MMA is maximized in the direction of the applied force. Figures 6.5(a) and 6.6(a) show two end-effector trajectories (for 1 cycle) tracked by the LTM. Each trajectory is tracked using the least norm solution and the maximization of MMA in the direction of the applied force. In Figure 6.5(b), MMA considered only the forces applied at the wrist, while Figure 6.6(b) considered only the moments applied at the wrist. The robot's ability to apply a force or a moment in the direction of the applied force or moment is considerably improved using the optimization control schemes based on MMA as compared to the least norm control scheme. In Figure 6.6(b), it is shown that maximizing MMA in the direction of the applied moment did not result in quite an improvement in MMA as compared to the least norm solution for the short period of time before the end-effector stopped. This is due to the fact that the manipulator had reached a configuration that whatever the null space solution was, the manipulator could not get itself in a configuration that could result in a much better improvement of MMA in the direction of that applied moment. In both Figures 6.7 and 6.8, the LTM arm is shown

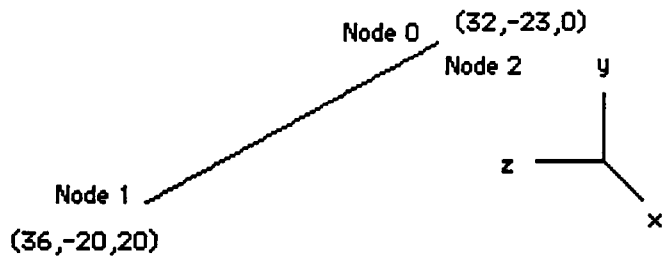


(a) End-effector trajectory motion

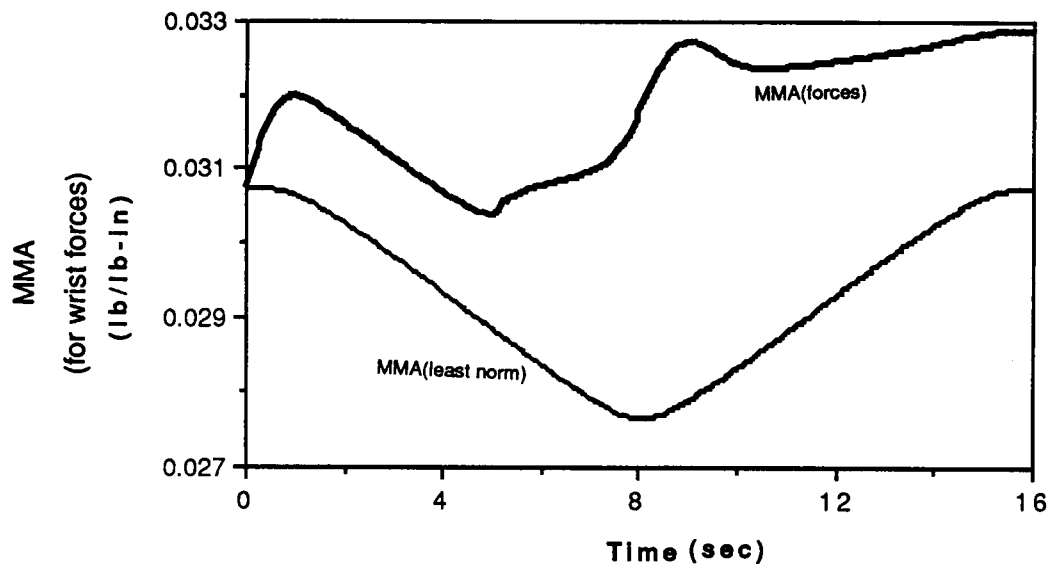


(b) MVR vs. time

Figure 6.4. MVR in the direction corresponding to the minimum singular value vs. time. (angular end-effector motion case)

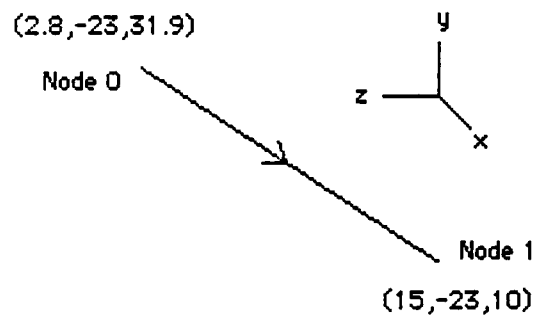


(a) End-effector trajectory motion

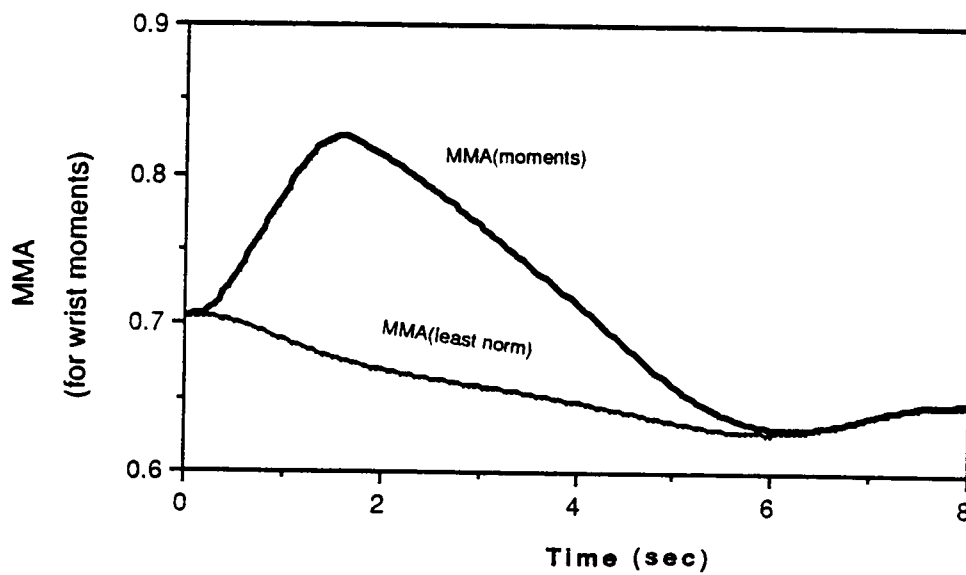


(b) MMA vs. time

Figure 6.5. MMA vs. time. (for forces applied at the wrist)



(a) End-effector trajectory motion



(b) MMA vs. time

Figure 6.6. MMA vs. time. (for moments applied at the wrist)

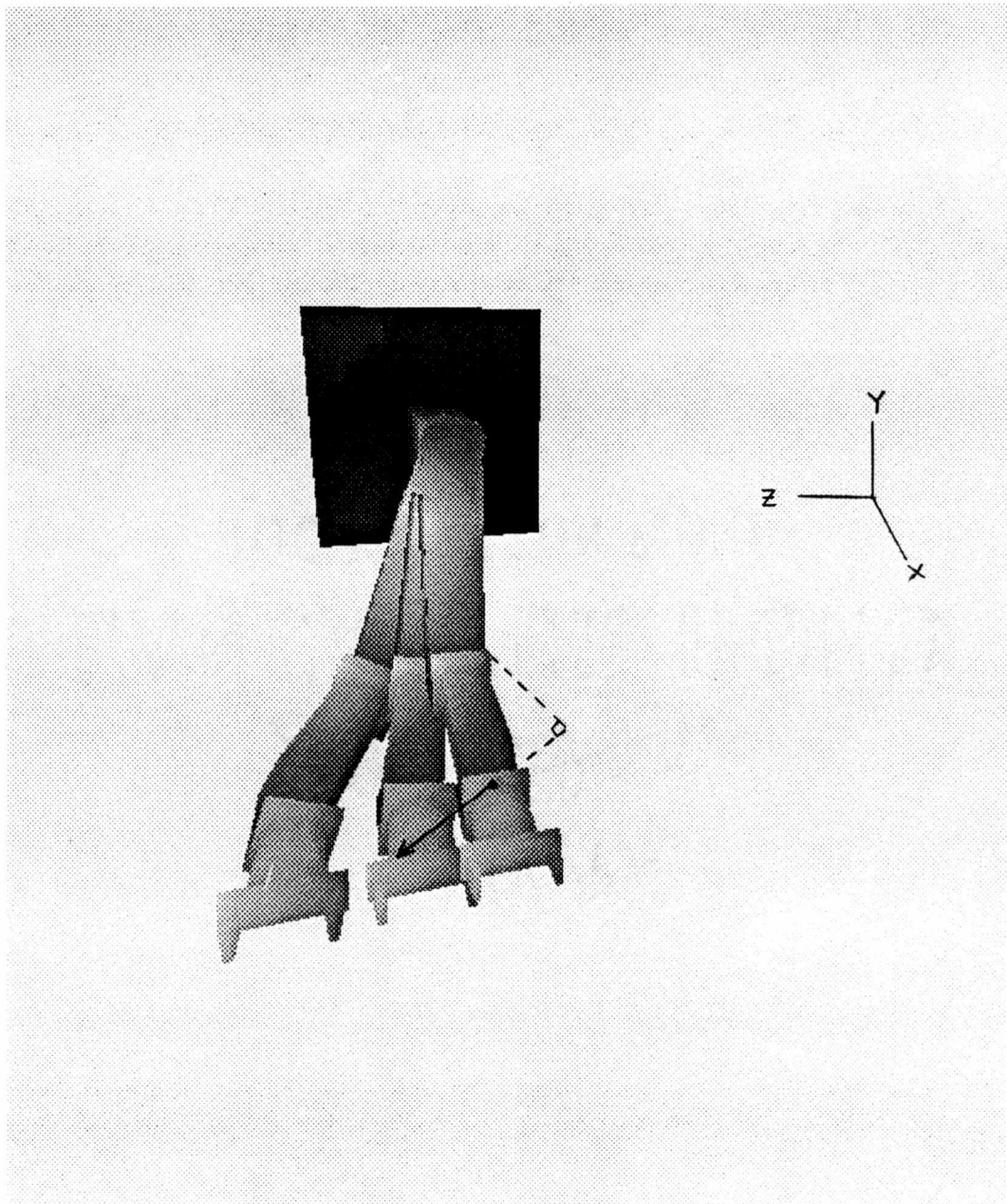


Figure 6.7. Manipulator configuration using the least norm solution for the trajectory in Figure 6.6.

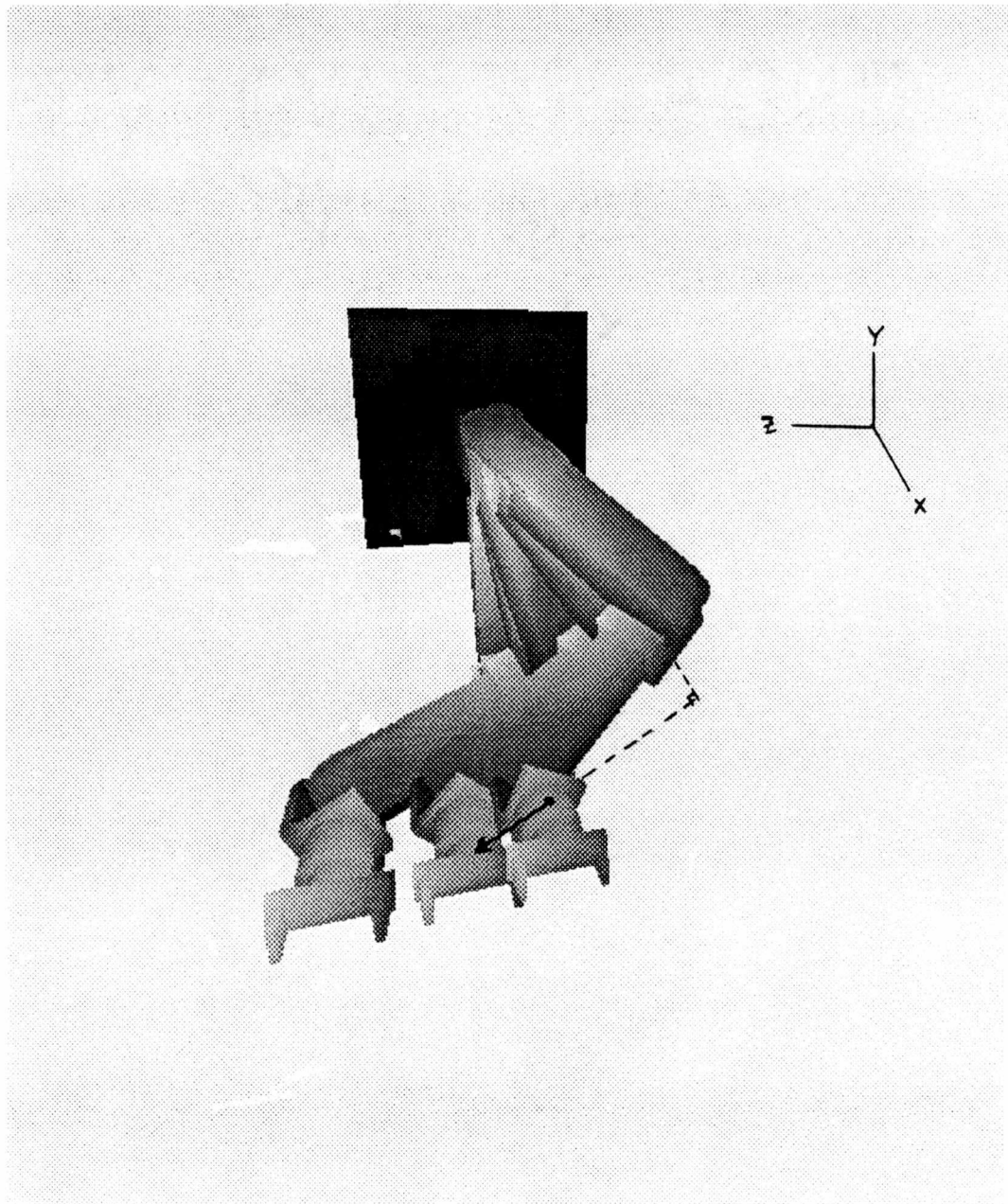
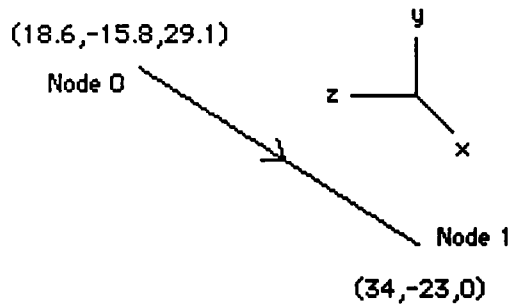


Figure 6.8. Manipulator configuration using the maximization of MMA in the direction of the wrist force for the trajectory in Figure 6.6.

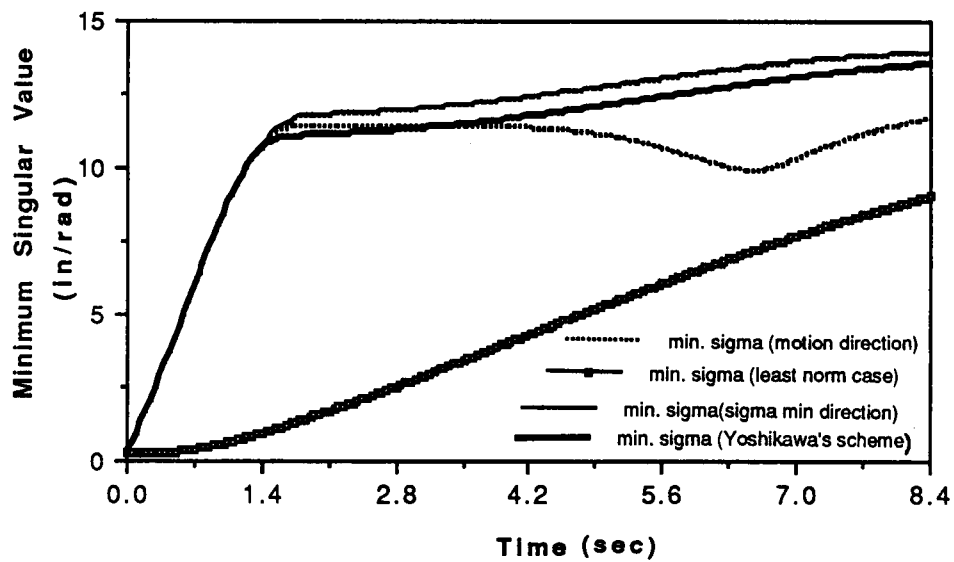
to be continuously changing to improve the manipulator configuration that resulted in the minimum joint torques required for a specified external force applied at the wrist. In this case the direction of the external force applied at the wrist is parallel to a unit vector whose x, y, and z components are positive, equal, and constant. If this force vector is projected onto the xz-plane in both Figures 6.7 and 6.8, then its component in the xz-plane lies in the direction shown in these Figures. Observing these Figures, it can be deduced that the moment arm of this projected force at the elbow joint is shorter in Figure 6.8 than in Figure 6.7. This results in a higher moment applied at the elbow using the least norm scheme than if we maximized MMA in the direction of the applied force. Therefore the MMA based control schemes are well suited for minimizing joint torques.

6.3 THE MINIMUM SINGULAR VALUE AND THE MANIPULABILITY MEASURE

Figure 6.9(b) is a plot of the minimum singular value ($\sigma_{\min}$) versus time for four different control schemes: The control scheme that improves MVR in the direction of motion of the end-effector, the control scheme that improves MVR in the direction corresponding to the minimum singular value, the least norm control scheme, and the Manipulability Measure based control scheme[14]. Figure 6.9(a) shows the corresponding trajectory for the end-effector (1 cycle). As shown in Figure 6.9(b), the least norm scheme results in the lowest value of $\sigma_{\min}$. In this trajectory, the end-effector started from a near singular configuration where initially $\sigma_{\min}$ was very close to zero. Using the least norm scheme, the end-effector hardly escaped the singular configuration mentioned above, and in fact the value of $\sigma_{\min}$ for the first several iterations was almost the same. However, the other schemes resulted in joint configurations that avoided this singular



(a) End-effector trajectory motion



(b) $\sigma_{\min}$ vs. time

Figure 6.9. Minimum singular value vs. time using different control schemes.

configuration and improved $\sigma_{\min}$ at a much faster rate than the least norm scheme.

The value of $\sigma_{\min}$ obtained using the Manipulability Measure based control scheme and the control scheme that improved MVR in the direction corresponding to the minimum singular value were almost very close. Improving MVR in the direction of motion resulted in values that fluctuated between the values obtained using the least norm control scheme and control scheme that improved MVR in the direction corresponding to $\sigma_{\min}$. It should be noted that the control scheme that results in the highest value of $\sigma_{\min}$ is the most recommendable in situations where directional changes are frequent because whenever $\sigma_{\min}$ gets smaller, the chances that the manipulator will encounter a singular configuration in the corresponding direction will increase.

Figure 6.10 shows a plot of the volume of the MVR Ellipsoid versus time for the different control schemes used for the end-effector trajectory in Figure 6.9(a). In a near singular configuration one of the axes of the ellipsoid will vanish, and the volume of the ellipsoid will approach zero. An ellipsoid that has the largest possible volume results in an overall better mobility in any direction of motion than any other ellipsoid because the ellipsoid becomes more circular. If the major and minor axis of the ellipsoid vary considerably, then any change in the direction of motion of the end-effector may result in a near singular configuration for the robot manipulator. In Figure 6.10, the control scheme that improves MVR in the $\sigma_{\min}$ direction and the control scheme that uses the Manipulability Measure criterion, resulted in the same volume values for the ellipsoids, and therefore one of them was omitted to avoid confusion while plotting. Since the Manipulability Measure based control scheme is computationally complex, using the control scheme that increases MVR in the $\sigma_{\min}$ direction will be almost as good in avoiding singular configurations and improving the mobility of the end-effector in near singular configurations. Figures 6.11 and 6.12 show the manipulator configuration using the least norm

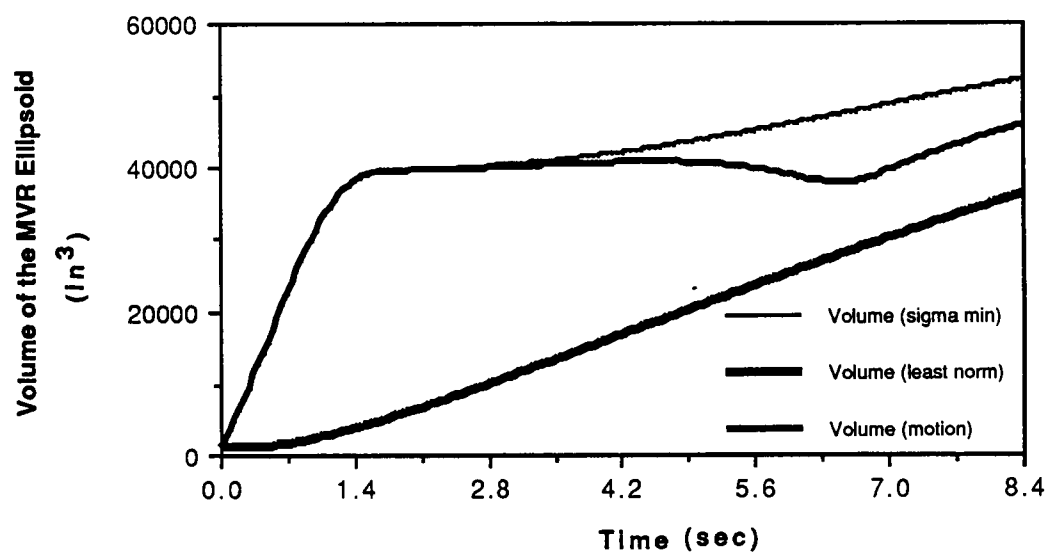


Figure 6.10. Volume of the MVR Ellipsoid vs. time using different control schemes.

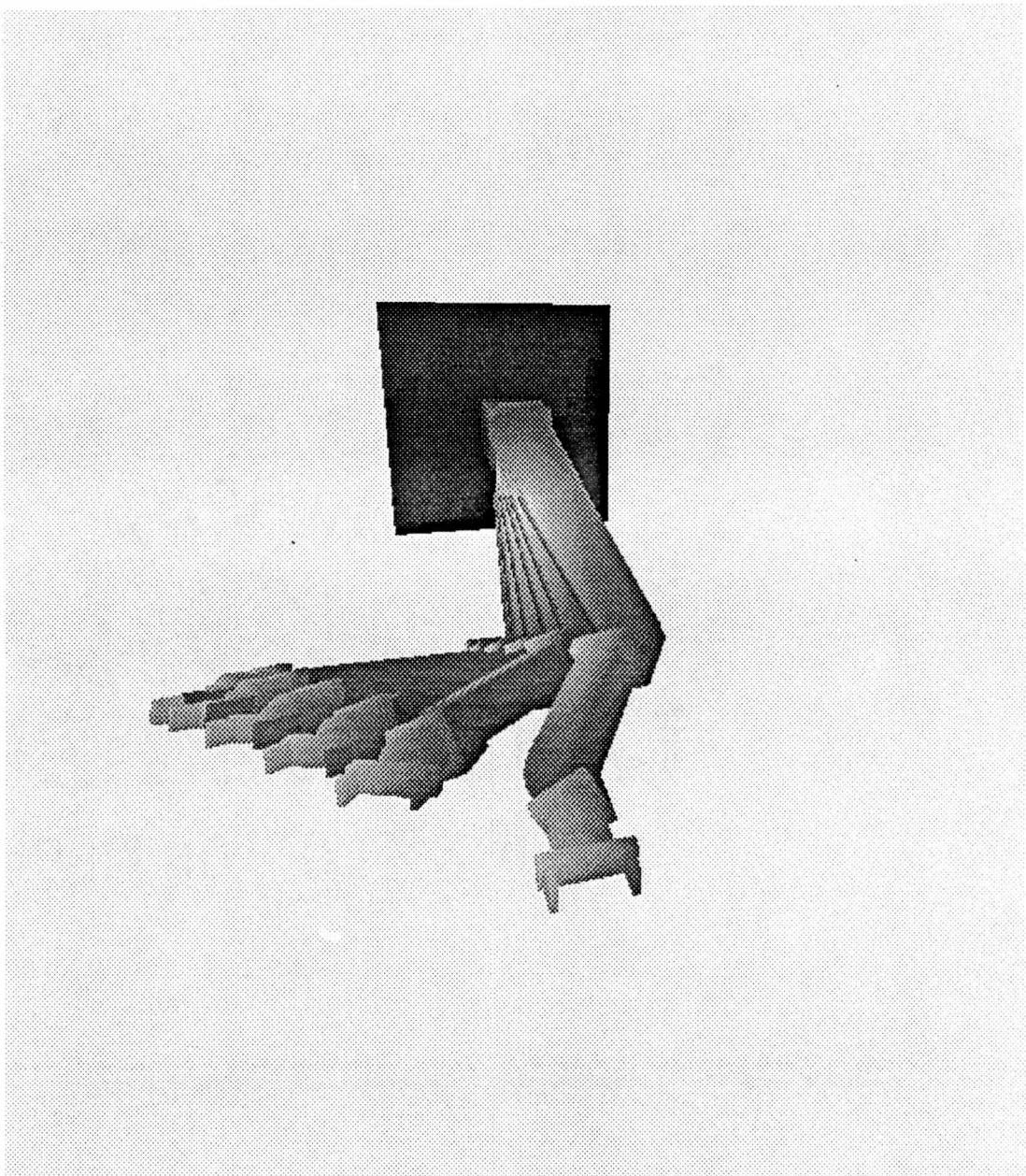


Figure 6.11. Manipulator configuration using the least norm solution for the trajectory in Figure 6.9.

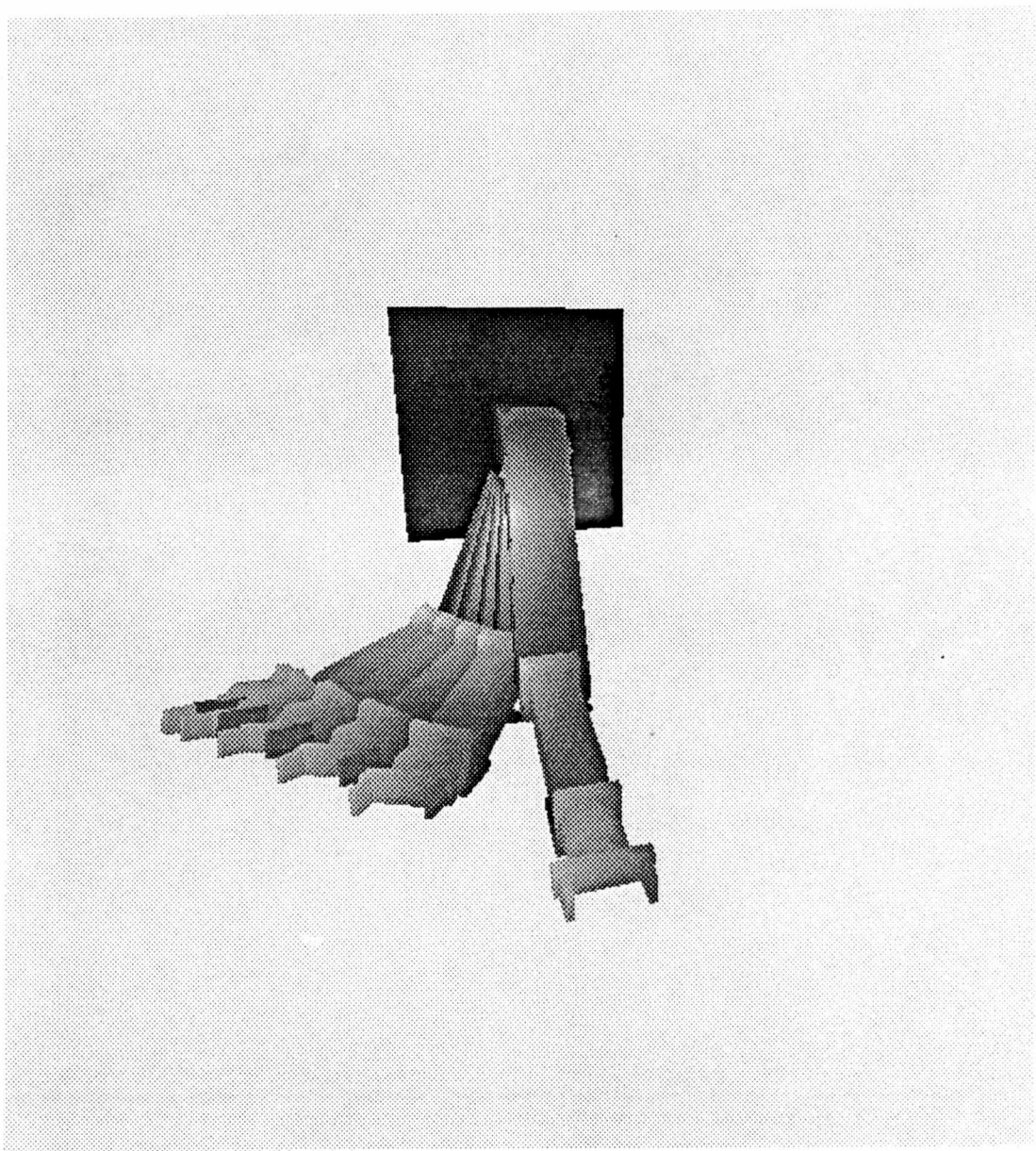


Figure 6.12. Manipulator configuration using the maximization of the Manipulability Measure for the trajectory in Figure 6.9.

solution and the maximization of the Manipulability Measure respectively for the trajectory of Figure 6.9.

6.4 EFFICIENT MOTION AND STABILITY

Figure 6.13 shows three trajectory paths referred to as case #1, case #2, and case #3. The trajectory for case #1 is a closed loop trajectory consisting of 4 nodes. As shown in Figure 6.13, the directional changes in case #1 occur more frequently at short distances. In Figure 6.14, the total joint motion per cycle for case #1 was plotted using the above mentioned control schemes and the least norm scheme. The path for case #1 was tracked repeatedly for 44 cycles. The total joint motion per cycle is defined here as the summation of the absolute values of the motion of the seven joints per cycle. It was obtained by numerically integrating this absolute value of the joint velocities for each cycle. The least norm control scheme resulted in high joint motion after several repeated cycles, but finally a limit value for joint motion per cycle was reached. On the other hand, maximizing MVR in the direction of motion or in the direction of the minimum singular value, resulted in a steady joint motion attained from the second cycle. The trajectory tracked by the end-effector during the first cycle is different than the trajectories tracked for other cycles because the arm was initially at rest (see chapter 5). Due to this reason, the first cycle was omitted. The total joint motion per cycle resulted by maximizing MVR in the direction of the minimum singular value was less than the motion resulted by maximizing MVR in the direction of motion. Maximizing MVR in the direction of minimum singular value results in MVR ellipsoids with minimum singular values as large as possible. For the three dimensional ellipsoid this would be closer to a sphere. Thus the manipulator's ability to arbitrarily change the direction of the end-effector motion improves. Therefore, for case #1, where directional changes occur more frequently, maximizing MVR in the direction of minimum singular value results in more efficient motion. If MVR is maximized in the direction of motion for case #1, the manipulator is not prepared for frequent directional changes resulting in higher value of overall motion.

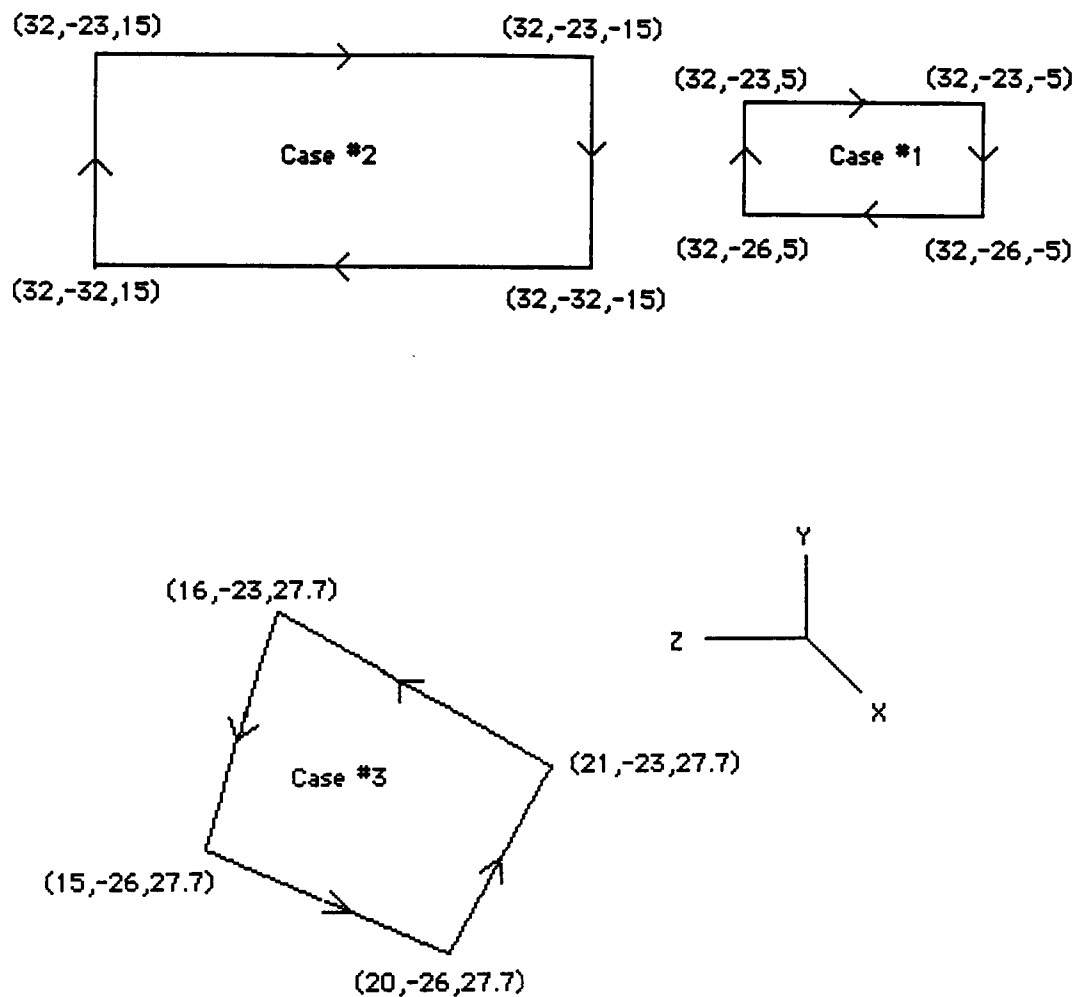


Figure 6.13. Trajectory motion for case #1, case #2, and case #3.

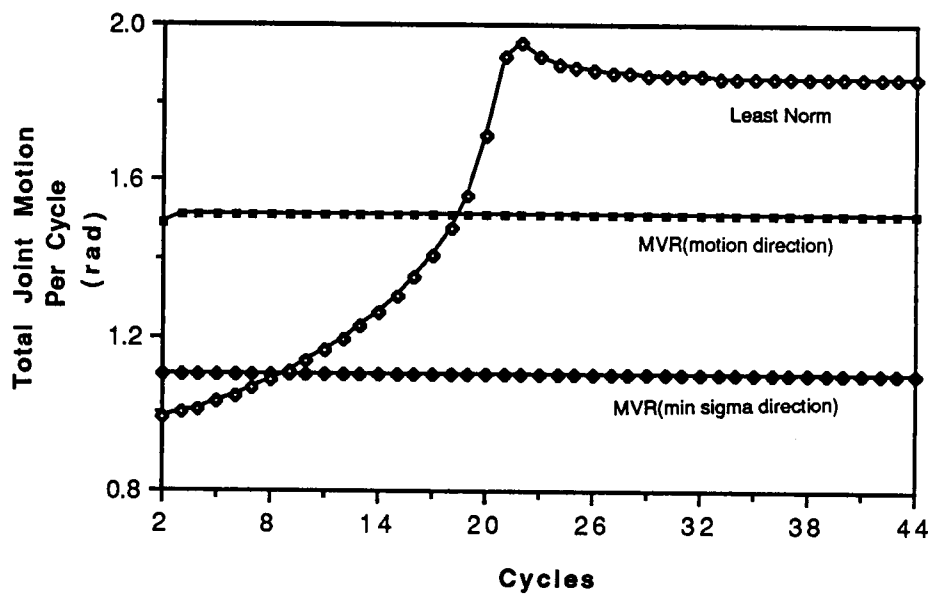


Figure 6.14. Total joint motion per cycle for case #1.

Figure 6.15 shows the total joint motion per cycle using the two MVR control schemes and the least norm scheme for the longer path referred to as case #2. In this case also the total cycle joint motion reaches a limiting value for the least norm solution, the maximization of MVR in the direction of minimum singular value, and the maximization of MVR in the direction of motion.

The limiting value for the total cycle joint motion using the maximization of MVR in the direction of motion was less than that using the maximization of MVR in the direction of the minimum singular value. In case #2, as the directional changes of the end-effector trajectory occur after longer distances, the manipulator can re-orient itself considerably to improve its ability to move in one particular direction without the need to change suddenly. Therefore more efficient motion is achieved by maximizing MVR in the direction of motion as compared to maximizing MVR in the direction of minimum singular value.

Classification of a trajectory as short or long will depend on several factors such as: the manipulator geometry, the constraints on the motors speeds, and the desired maximum end-effector velocity. It is recommended to investigate the determination of such a classification analytically in future work. Due to the complexity of this problem, such a classification may be determined empirically.

Figures 6.16 through 6.21 show the manipulator configuration at steady state using the least norm control scheme, the maximization of MVR in the direction of the minimum singular value, and the maximization of MVR in the direction of motion for case #1 and case #2. Figure 6.17 shows the manipulator configuration at steady state using the maximization of MVR in the direction of the minimum singular value for case #1. As shown in this Figure, the manipulator barely changed its joint configurations over this short cycle. If we examine Figures 6.16 and 6.18, we see that the manipulator configuration at steady state for the same case using the least norm solution and the maximization of MVR in the direction of motion resulted in more joint motion and the links of the manipulator opened up. Similarly in Figures 6.19 and 6.21, the manipulator links opened up while in Figure 6.20 the links remained quite in close locations. Since the trajectory tracked in this case is larger than that for case #1, some of the manipulator joints in Figure 6.20 required

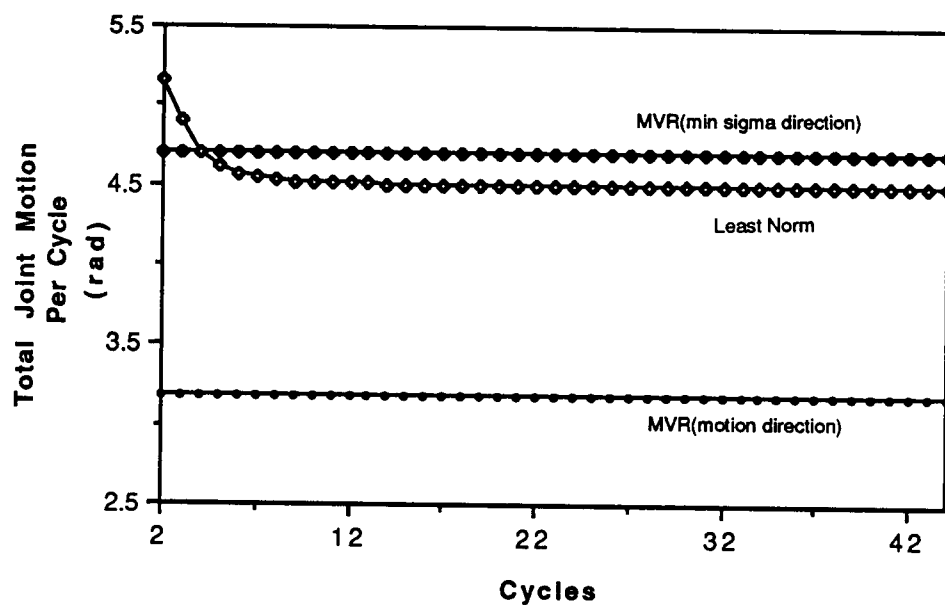


Figure 6.15. Total joint motion per cycle for case #2.

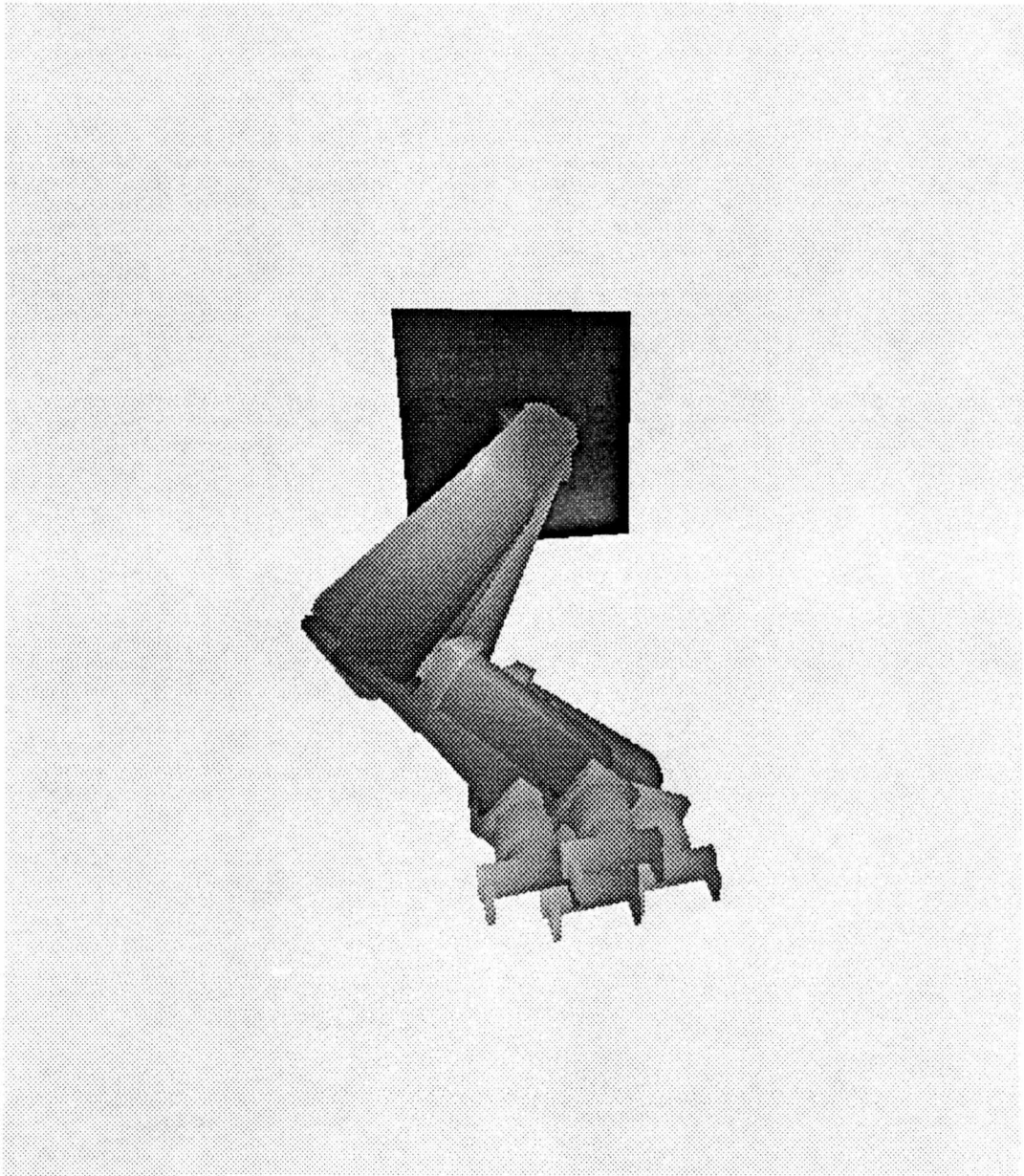


Figure 6.16. Steady state manipulator configuration cycle using the least norm solution for case #1.

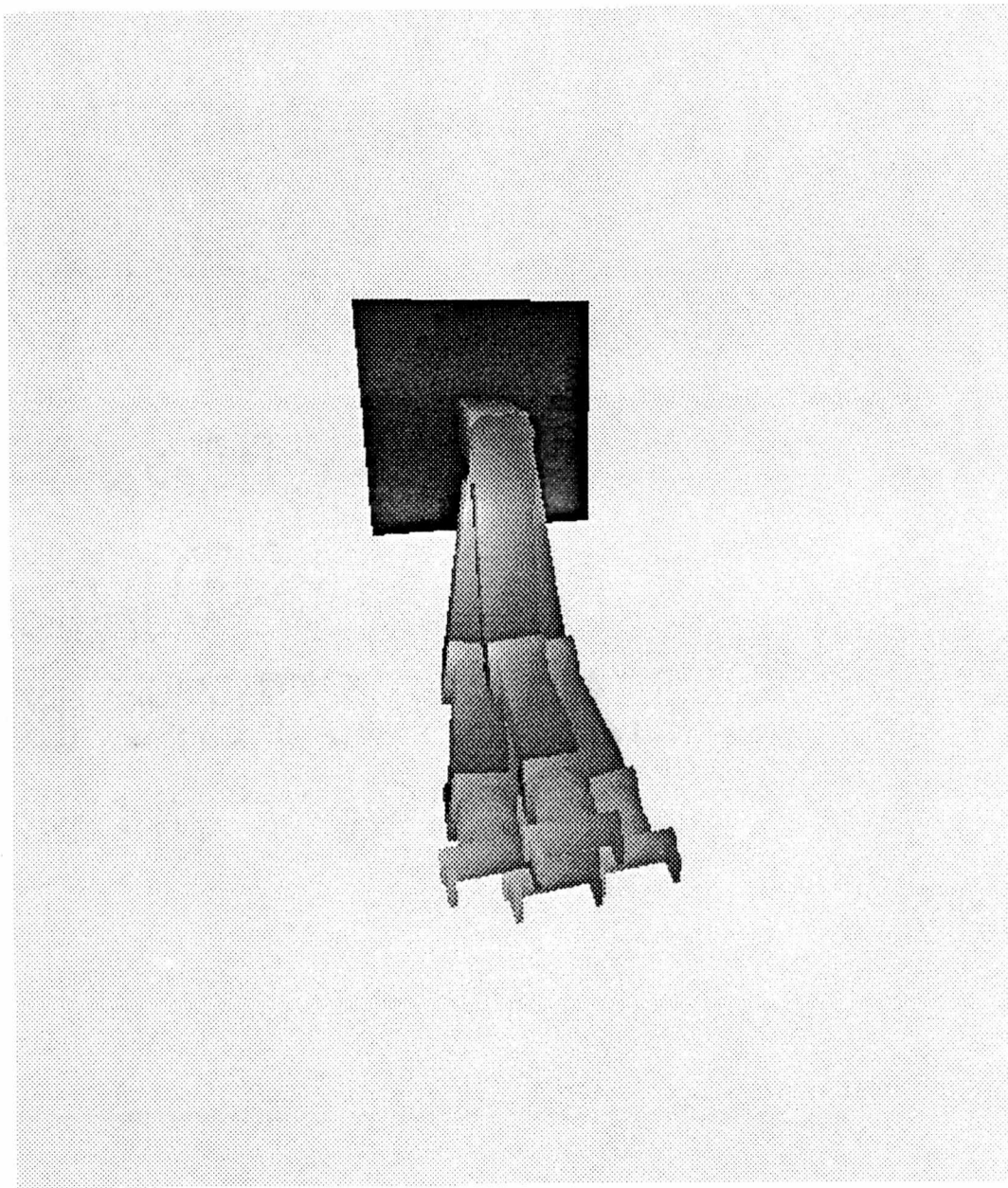


Figure 6.17. Steady state manipulator configuration cycle using the maximization of MVR in the direction corresponding to the minimum singular value for case #1.

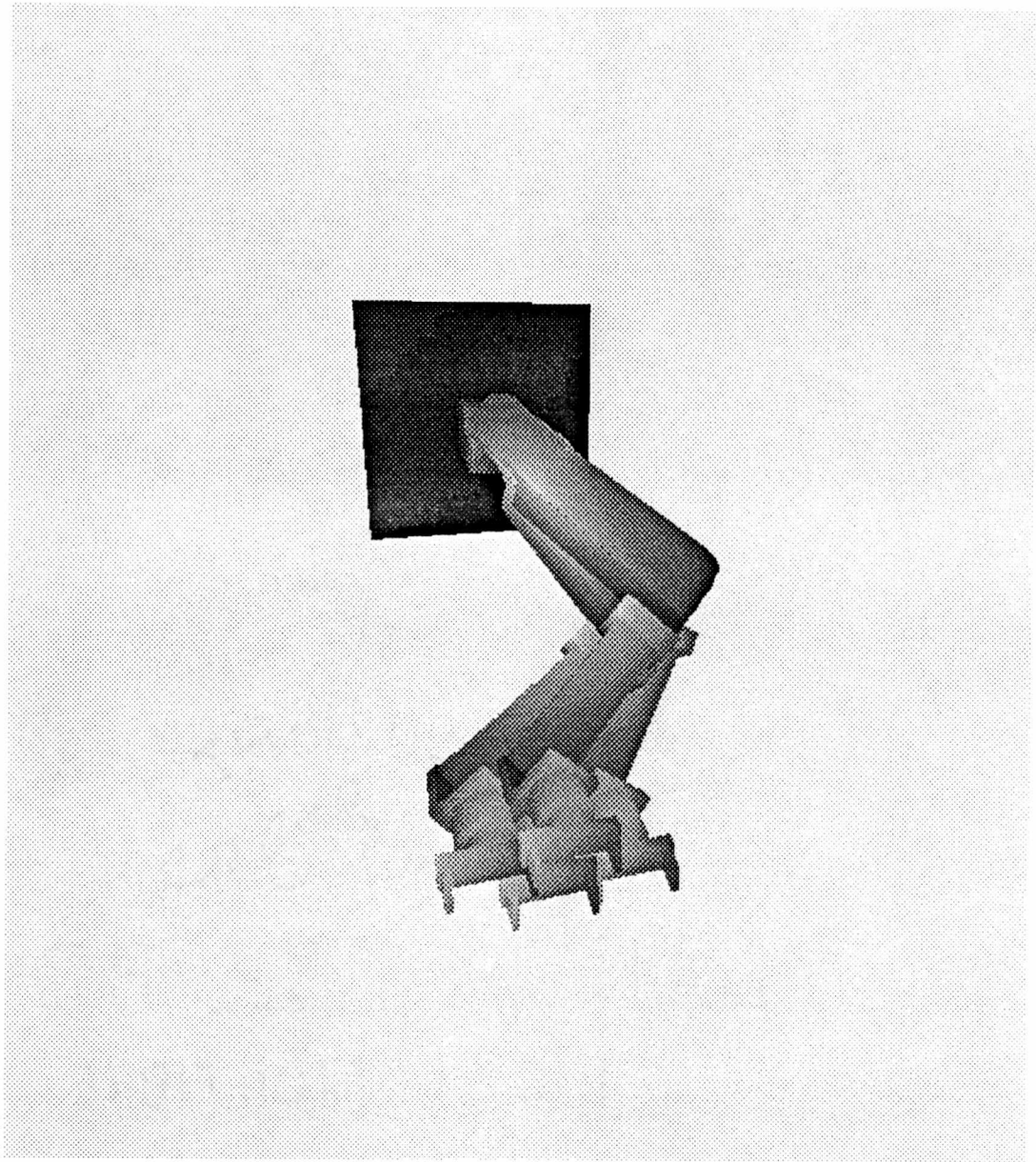


Figure 6.18. Steady state manipulator configuration cycle using the maximization of MVR in the direction of motion for case #1.

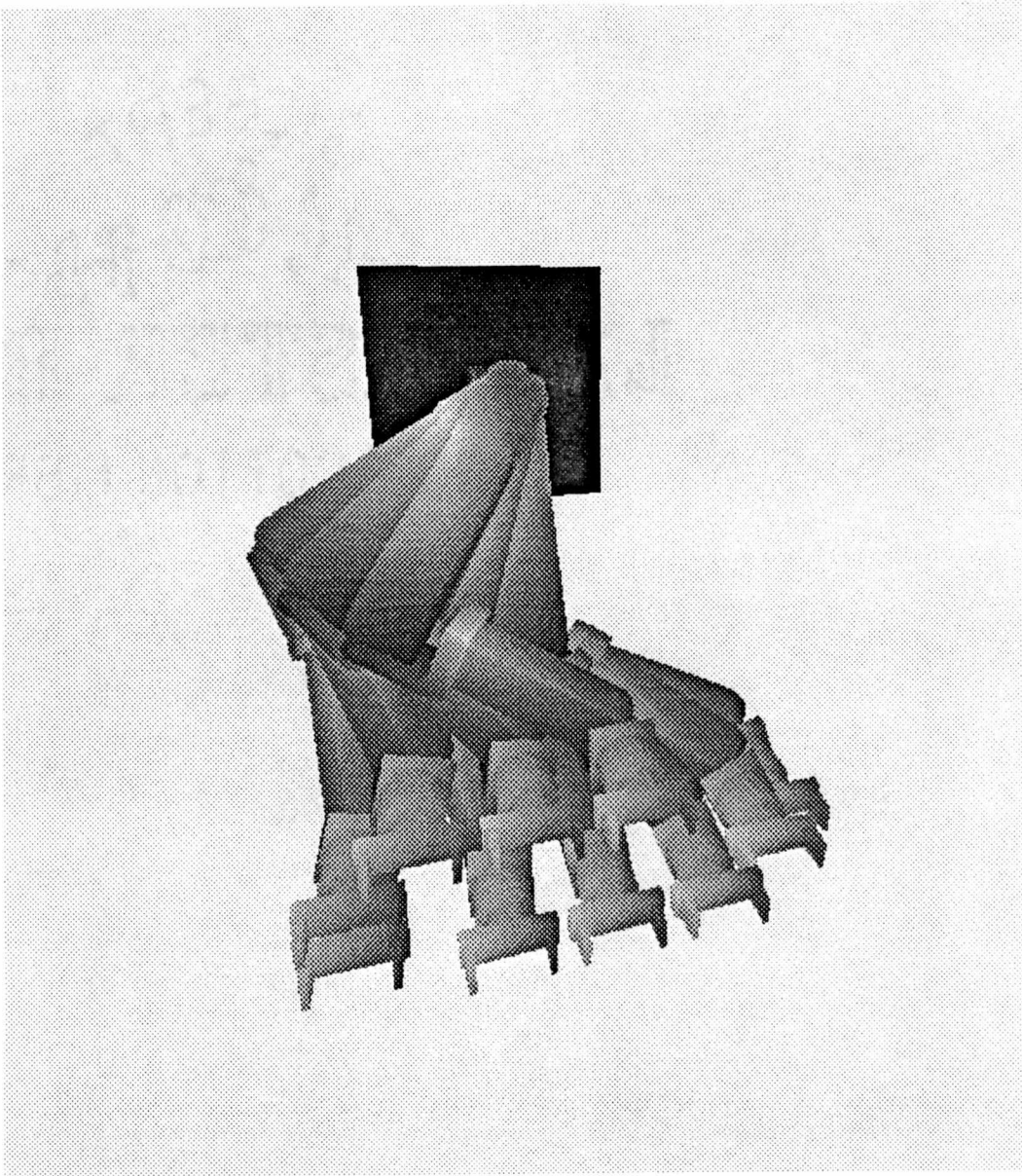


Figure 6.19. Steady state manipulator configuration cycle using the least norm solution for case #2.

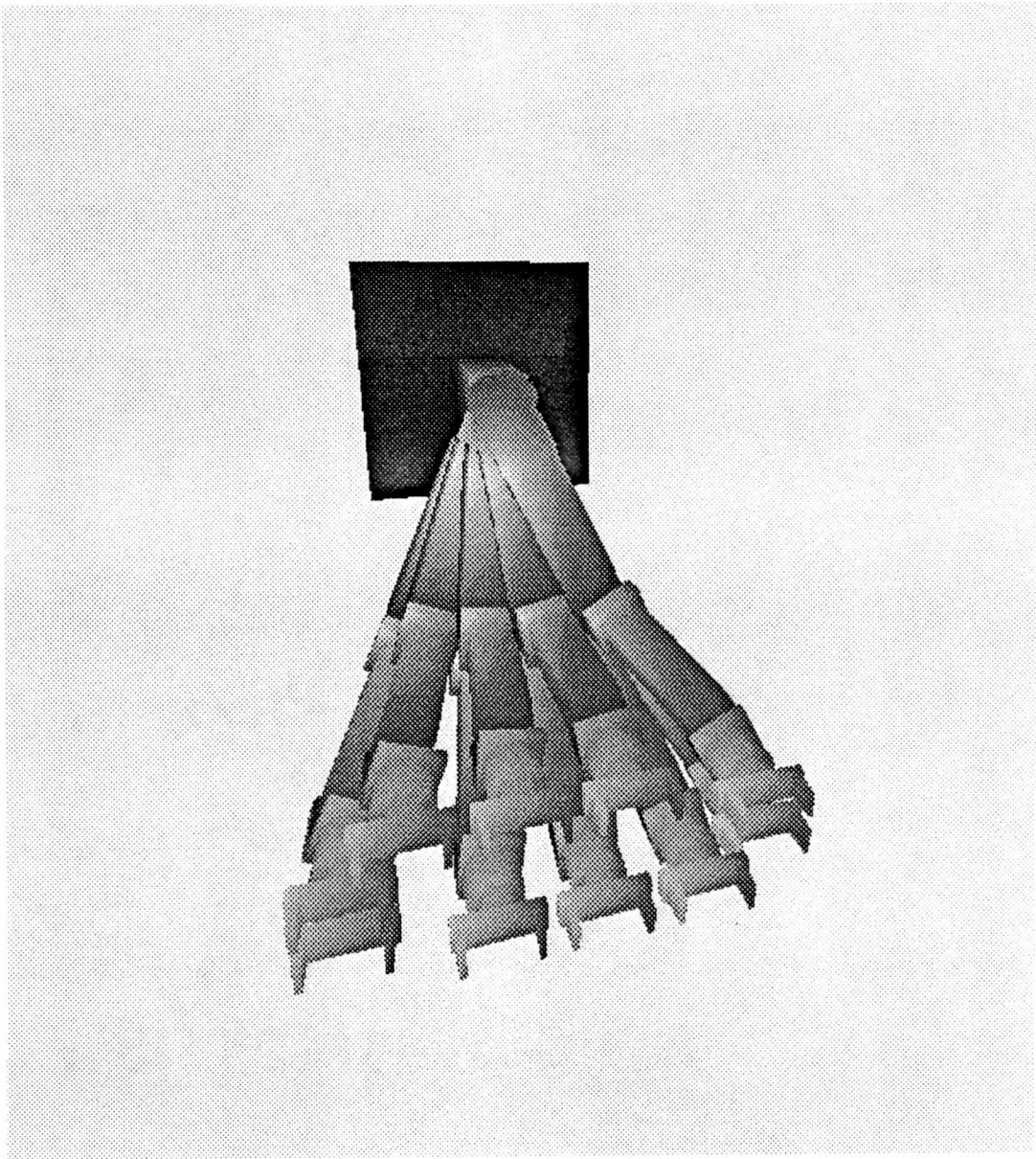


Figure 6.20. Steady state manipulator configuration cycle using the maximization of MVR in the direction corresponding to the minimum singular value for case #2.

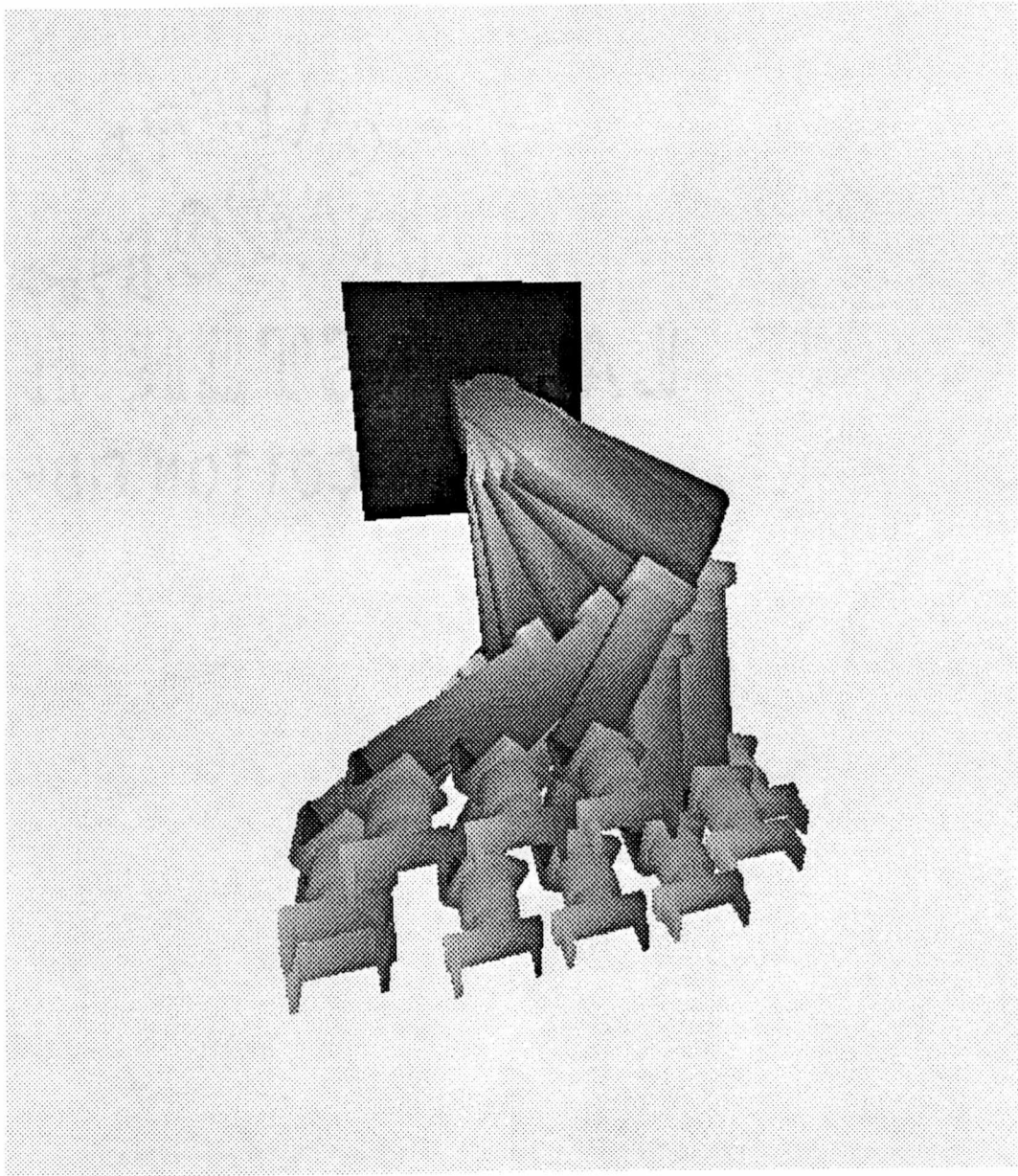


Figure 6.21. Steady state manipulator configuration cycle using the maximization of MVR in the direction of motion for case #2.

more motion than others, while in Figure 6.21 the motion was distributed more evenly on every joint. As a result the total joint motion at steady state for the manipulator in Figure 6.21 (where the maximization of MVR in the direction of motion was utilized) was less than the total joint motion at steady state for the manipulator in Figure 6.20 (where the maximization of MVR in the direction corresponding to the minimum singular value was utilized). It is worth mentioning here that for both cases, case #1 and case #2, the end-effector was moving in a clockwise motion while tracking the specified trajectory. In the case where the least norm scheme was utilized, the manipulator links opened up in a direction opposite to the direction obtained using the control scheme that maximizes MVR in the direction of motion.

Figures 6.22 and 6.23 correspond to case #1 and case #2 respectively. These graphs show that the limit value of the total joint motion per cycle varies with different starting configurations. However, as the move time between the consecutive nodes increases, the total joint motion per cycle converges to a limiting value.

Figure 6.24 corresponds to case #3 in which the end-effector follows a closed loop trajectory that consists of four nodes also. In this case, results obtained using the least norm control scheme are compared with those obtained using the control scheme that maximizes MVR in the direction of motion of the end-effector. Using the least norm scheme, the manipulator became unstable and very high joint velocities were reached. On the contrary the MVR control scheme resulted in a stable joint motion all the way through from the second cycle.

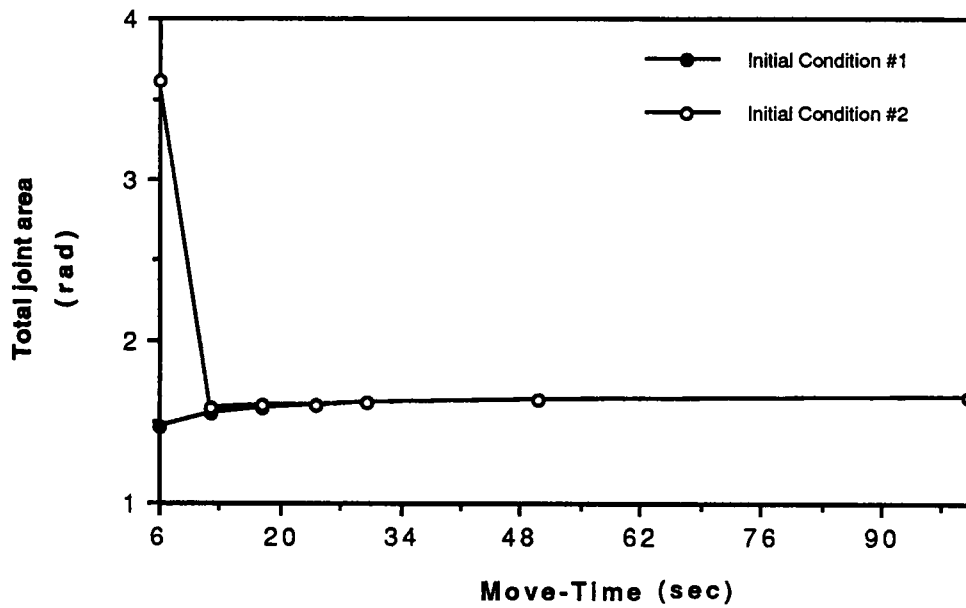


Figure 6.22. Limiting value for the total joint area per cycle vs. time for case #1.

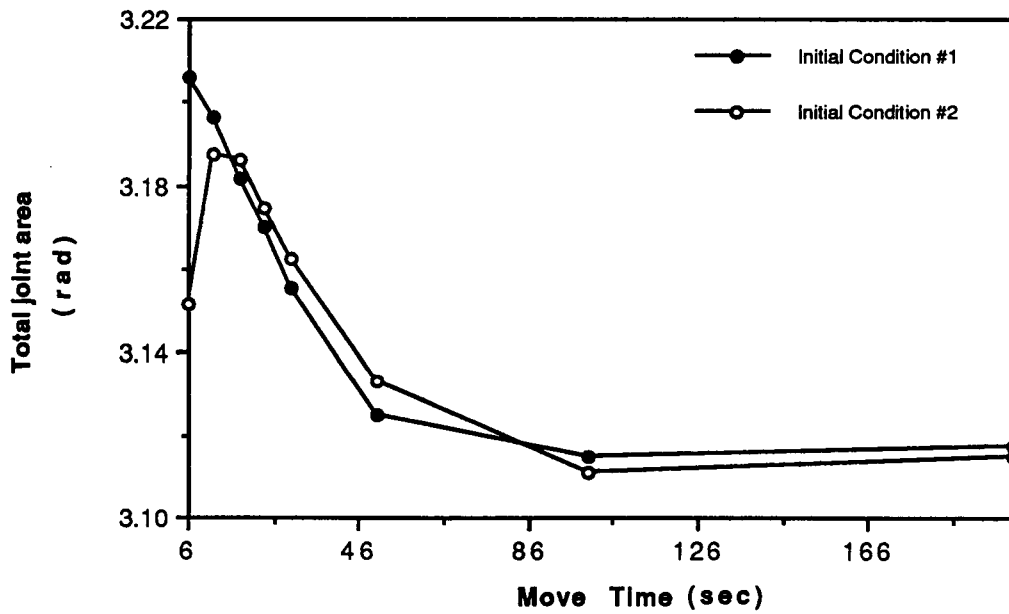


Figure 6.23. Limiting value for the total joint area per cycle vs. time for case #2.

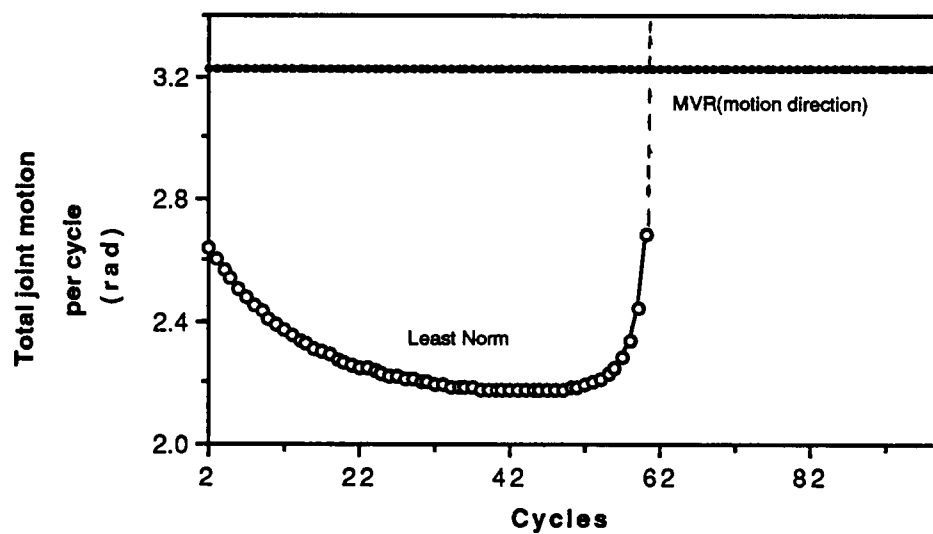


Figure 6.24. Total joint motion per cycle for case #3.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

7.1 CONCLUSIONS

Control Schemes that maximize MVR in the direction of motion of the end-effector and in the direction in which MVR is minimum were implemented on the LTM. Efficient motion was defined as the manipulator motion that results in the minimum overall movement of the joints. For short trajectories in which the direction of motion of the end-effector changes rapidly and quite often, it was shown that maximizing MVR in the direction of the minimum singular value of the manipulability ellipsoid results in minimum overall movement of the joints. It was also shown that for long trajectory paths in which the change in the direction of motion of the end-effector does not happen rapidly, maximizing MVR in the direction of motion results in more efficient motion. The MVR control schemes result in stable joint motion for repetitive cycles, while on the contrary the Least Norm scheme was shown to be least efficient, was often unstable and occasionally led to high joint velocity configurations. The limiting value of the total joint motion per cycle was shown to vary with different starting configurations. However, these variations almost disappeared as the total cycle time was increased. This was due to more optimization time available while following the desired end-effector trajectory. It was also shown that maximizing MVR in the direction corresponding to the minimum singular value was almost as good in avoiding singular configurations as well as improving the mobility of the end-effector in near singular configurations as the control scheme that is based on improving the Manipulability Measure.

It was shown that using the MMA control scheme, joint configurations that required the minimum joint torques for a certain force applied at the wrist were achieved. Thus the MMA based control scheme is well suited for avoiding high joint torques.

7.2 RECOMMENDATIONS

In this thesis, end-effector trajectories were classified as long or short, and according to this classification maximizing MVR in the direction of motion of the end-effector or in the direction in which MVR is minimum resulted in better efficient motion for one control scheme than the other depending on the nature of the trajectory. It is recommended to investigate the determination of such a classification analytically for future work. The geometry of the LTM arm is not well suited for operating the manipulator with rare encounter of joint limits. In many cases where the end-effector was tracking a prescribed trajectory joint limits were encountered frequently. Thus, it is recommended to take into account multiple performance criteria. The manipulator can for instance improve MVR in a certain direction while at the same time avoid joint limits. Combining other criteria such as MVR and/or MMA with a criterion to avoid an obstacle is also needed in many applications. When the manipulator encounters an obstacle, the weighting factor associated with the obstacle avoidance criterion will increase, while in an obstacle-free environment more weight will be imposed on the MVR or the MMA criterion. Also, combination of MVR and MMA may be optimized in cases where both obtaining a certain end-effector force and lowering the joint velocities to a certain range are of importance.

LIST OF REFERENCES

REFERENCES

- [1] Asada, H., and Slotine, Jean-Jacques E. 1986 Robot Analysis and Control, Wiley Inc, New York, NY.
- [2] Dubey, R.V. and Luh, J.Y.S. (1988), "Redundant Robot Control Using Task Based Performance Measures," *Journal of Robotic Systems*, 5(5), 409-432, October 1988.
- [3] Dubey, R.V., Euler, J.A., and Babcock, S.M. 1988. "An Efficient Gradient Projection Optimization Scheme for a Seven-Degree-Of-Freedom Redundant Robot With a Spherical Wrist." *IEEE International Conference on Robotics and Automation*, April 25-29 1988.
- [4] Euler, J.A., Dubey, R.V. and Babcock, S.M. (1988), "Self Motion Determination Based on Joint Velocity Bounds for Redundant Robots," *Proc. of the Second International Symposium of Robotics and Mfg Research, Education and Application*, Albuquerque, November, 1988, pp. 291-298, ASME Press.
- [5] Fu, K.S., Gonzalez, R.C., LEE, C.S.G. 1987 Robotics: Control, Sensing, Vision, and Intelligence, McGraw-Hill, New York, NY.
- [6] Hollerbach, J.M. and Suh, K.C., "Redundancy resolution of manipulators through torque optimization," *Proc. IEEE Conf. Robotics and Automation*, March, 1985, pp. 1016-1021.
- [7] Lee, S., "Dual Redundant Arm Configuration Optimization with Task-Oriented Dual Arm Manipulability," *IEEE Transactions on Robotics and Automation*, Vol. 5, No.1, February 1989, pp. 78-97.
- [8] Luh, J.Y.S. and Gu, Y.L., "Efficiency and Flexibility of Industrial Robots with Redundancy," *Proc. ASME Winter Annual Meeting, PED Vol. 15, Robotics and*

Manufacturing Automation, ed. M. Donath and M. Leu, Nov. 17-22, 1985, Miami-Beach, Florida, pp.37-41.

- [9] Magness, R.B. Robotic Control of the Seven-Degree-of-Freedom NASA Laboratory Telerobotic Manipulator. Master Thesis, University of Tennessee, Knoxville (1989).
- [10] Paul, R.P. Robot Manipulators: Mathematics, Programming, and Control, Cambridge: MIT Press.
- [11] Paul, R.P., and Stevenson, C.N. (1983), "Kinematics of robot wrists," The Int. Journal of Robotics Research, Vol. 1, No. 2, pp. 33-38.
- [12] Strang, Gilbert. 1988 Linear Algebra and its Applications, Harcourt Brace Javanovich.
- [13] Vukobratovic, M., and Kircanski, M., "A dynamic approach to nominal trajectory synthesis for redundant manipulators," IEEE Trans. Syst., Man, Cybern., Vol. SMC-14, No. 4, July/Aug. 1984.
- [14] Yoshikawa, T. (1984), "Analysis and Control of Robot Manipulators with Redundancy," Robotics Research: The First Int. Symp., ed. M. Brady and R. Paul, Cambridge, Mass., MIT Press, pp. 735-748.

APPENDIXES

APPENDIX A

THE DENAVIT-HARTENBERG HOMOGENEOUS TRANSFORMATION MATRICES

The homogeneous transformation matrices relating the i^{th} coordinate frame to the $(i-1)^{\text{th}}$ coordinate frame (for $i = 1, 2, \dots, 7$) can be shown to be as follows :

$$A_1^0 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} C_2 & 0 & S_2 & a_2 C_2 \\ S_2 & 0 & -C_2 & a_2 S_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} C_3 & 0 & S_3 & 0 \\ S_3 & 0 & -C_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4^3 = \begin{bmatrix} C_4 & 0 & -S_4 & a_4 C_4 \\ S_4 & 0 & C_4 & a_4 S_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^4 = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6^5 = \begin{bmatrix} C_6 & 0 & S_6 & 0 \\ S_6 & 0 & -C_6 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_7^6 = \begin{bmatrix} C_7 & -S_7 & 0 & 0 \\ S_7 & C_7 & 0 & 0 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $S_i = \sin(\theta_i)$ and $C_i = \cos(\theta_i)$.

APPENDIX B

DERIVATION OF THE POSITION AND THE ORIENTATION OF THE END-EFFECTOR AND THE WRIST w.r.t. THE BASE COORDINATE FRAME

The homogeneous matrix T_7^0 which specifies the position and orientation of the 7th coordinate frame with respect to the base frame is expressed as:

$$T_7^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 A_6^5 A_7^6$$

$$= \begin{bmatrix} x_7 & y_7 & z_7 & p_7 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \underline{n} & \underline{s} & \underline{a} & \underline{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where

$\underline{n}$ = normal vector of the hand.

$\underline{s}$ = sliding vector of the hand. It is pointing in the direction of the finger motion as the gripper opens and closes.

$\underline{a}$ = approach vector of the hand. It is pointing in the direction normal to the tool mounting plate of the arm.

$\underline{p}$ = position vector of the hand. It points from the origin of the base coordinate frame to the origin of the hand coordinate frame.

$$\begin{aligned}
n_x = & C_1 C_2 C_3 C_4 C_5 C_6 C_7 - S_1 S_3 C_4 C_5 C_6 C_7 + C_1 S_2 S_4 C_5 C_6 C_7 - C_1 C_2 S_3 S_5 C_6 C_7 - S_1 C_3 S_5 C_6 C_7 \\
& - C_1 C_2 C_3 S_4 S_6 C_7 + S_1 S_3 S_4 S_6 C_7 + C_1 S_2 C_4 S_6 C_7 + C_1 C_2 C_3 C_4 S_5 S_7 - S_1 S_3 C_4 S_5 S_7 \\
& + C_1 S_2 S_4 S_5 S_7 + C_1 C_2 S_3 C_5 S_7 + S_1 C_3 C_5 S_7
\end{aligned}$$

$$\begin{aligned}
n_y = & S_1 C_2 C_3 C_4 C_5 C_6 C_7 + C_1 S_3 C_4 C_5 C_6 C_7 + S_1 S_2 S_4 C_5 C_6 C_7 - S_1 C_2 S_3 S_5 C_6 C_7 \\
& + C_1 C_3 S_5 C_6 C_7 - S_1 C_2 C_3 S_4 S_6 C_7 - C_1 S_3 S_4 S_6 C_7 + S_1 S_2 C_4 S_6 C_7 + S_1 C_2 C_3 C_4 S_5 S_7 \\
& + C_1 S_3 C_4 S_5 S_7 + S_1 S_2 S_4 S_5 S_7 + S_1 C_2 S_3 C_5 S_7 - C_1 C_3 C_5 S_7
\end{aligned}$$

$$\begin{aligned}
n_z = & - S_2 C_3 C_4 C_5 C_6 C_7 + C_2 S_4 C_5 C_6 C_7 + S_2 S_3 S_5 C_6 C_7 + S_2 C_3 S_4 S_6 C_7 + C_2 C_4 S_6 C_7 \\
& - S_2 C_3 C_4 S_5 S_7 + C_2 S_4 S_5 S_7 - S_2 S_3 C_5 S_7
\end{aligned}$$

$$\begin{aligned}
s_x = & - C_1 C_2 C_3 C_4 C_5 C_6 S_7 + S_1 S_3 C_4 C_5 C_6 S_7 - C_1 S_2 S_4 C_5 C_6 S_7 + C_1 C_2 S_3 S_5 C_6 S_7 \\
& + S_1 C_3 S_5 C_6 S_7 + C_1 C_2 C_3 S_4 S_6 S_7 - S_1 S_3 S_4 S_6 S_7 - C_1 S_2 C_4 S_6 S_7 + C_1 C_2 C_3 C_4 S_5 C_7 \\
& - S_1 S_3 C_4 S_5 C_7 + C_1 S_2 S_4 S_5 C_7 + C_1 C_2 S_3 C_5 C_7 + S_1 C_3 C_5 C_7
\end{aligned}$$

$$\begin{aligned}
s_y = & - S_1 C_2 C_3 C_4 C_5 C_6 S_7 - C_1 S_3 C_4 C_5 C_6 S_7 - S_1 S_2 S_4 C_5 C_6 S_7 + S_1 C_2 S_3 S_5 C_6 S_7 \\
& - C_1 C_3 S_5 C_6 S_7 + S_1 C_2 C_3 S_4 S_6 S_7 + C_1 S_3 S_4 S_6 S_7 - S_1 S_2 C_4 S_6 S_7 + S_1 C_2 C_3 C_4 S_5 C_7 \\
& + C_1 S_3 C_4 S_5 C_7 + S_1 S_2 S_4 S_5 C_7 + S_1 C_2 S_3 C_5 C_7 - C_1 C_3 C_5 C_7
\end{aligned}$$

$$\begin{aligned}
s_z = & S_2 C_3 C_4 C_5 C_6 S_7 - C_2 S_4 C_5 C_6 S_7 - S_2 S_3 S_5 C_6 S_7 - S_2 C_3 S_4 S_6 S_7 \\
& - C_2 C_4 S_6 S_7 - S_2 C_3 C_4 S_5 C_7 + C_2 S_4 S_5 C_7 - S_2 S_3 C_5 C_7
\end{aligned}$$

$$a_x = C_1 C_2 C_3 C_4 C_5 S_6 - S_1 S_3 C_4 C_5 S_6 + C_1 S_2 S_4 C_5 S_6 - C_1 C_2 S_3 S_5 S_6 - S_1 C_3 S_5 S_6 \\ + C_1 C_2 C_3 S_4 C_6 - S_1 S_3 S_4 C_6 - C_1 S_2 C_4 C_6$$

$$a_y = S_1 C_2 C_3 C_4 C_5 S_6 + C_1 S_3 C_4 C_5 S_6 + S_1 S_2 S_4 C_5 S_6 - S_1 C_2 S_3 S_5 S_6 + C_1 C_3 S_5 S_6 \\ + S_1 C_2 C_3 S_4 C_6 + C_1 S_3 S_4 C_6 - S_1 S_2 C_4 C_6$$

$$a_z = -S_2 C_3 C_4 C_5 S_6 + C_2 S_4 C_5 S_6 + S_2 S_3 S_5 S_6 - S_2 C_3 S_4 C_6 - C_2 C_4 C_6$$

$$p_x = d_7 (C_1 C_2 C_3 C_4 C_5 S_6 - S_1 S_3 C_4 C_5 S_6 + C_1 S_2 S_4 C_5 S_6 - C_1 C_2 S_3 S_5 S_6 - S_1 C_3 S_5 S_6 \\ + C_1 C_2 C_3 S_4 C_6 - S_1 S_3 S_4 C_6 - C_1 S_2 C_4 C_6) + a_4 C_1 C_2 C_3 C_4 - a_4 S_1 S_3 C_4 + a_4 C_1 S_2 S_4 \\ + a_2 C_1 C_2$$

$$p_y = d_7 (S_1 C_2 C_3 C_4 C_5 S_6 + C_1 S_3 C_4 C_5 S_6 + S_1 S_2 S_4 C_5 S_6 - S_1 C_2 S_3 S_5 S_6 + C_1 C_3 S_5 S_6 \\ + S_1 C_2 C_3 S_4 C_6 + C_1 S_3 S_4 C_6 - S_1 S_2 C_4 C_6) + a_4 S_1 C_2 C_3 C_4 + a_4 C_1 S_3 C_4 + a_4 S_1 S_2 S_4 \\ + a_2 S_1 C_2$$

$$p_z = d_7 (-S_2 C_3 C_4 C_5 S_6 + C_2 S_4 C_5 S_6 + S_2 S_3 S_5 S_6 - S_2 C_3 S_4 C_6 - C_2 C_4 C_6) - a_4 S_2 C_3 C_4 \\ + a_4 C_2 S_4 - a_2 S_2$$

The position and orientation of the wrist frame (5th coordinate frame) with respect to the base coordinate frame can be expressed as follows :

$$T_5^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4$$

$$= \begin{bmatrix} x_5 & y_5 & z_5 & p_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x_{5_x} & y_{5_x} & z_{5_x} & p_{5_x} \\ x_{5_y} & y_{5_y} & z_{5_y} & p_{5_y} \\ x_{5_z} & y_{5_z} & z_{5_z} & p_{5_z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where

$\underline{x}_5$ = a vector representing the unit vector along the x-direction of the 5th coordinate frame in terms of the unit vectors of the base frame.

$\underline{y}_5$ = a vector representing the unit vector along the y-direction of the 5th coordinate frame in terms of the unit vectors of the base frame.

$\underline{z}_5$ = a vector representing the unit vector along the z-direction of the 5th coordinate frame in terms of the unit vectors of the base frame.

$$x_{5_x} = C_1 C_2 C_3 C_4 C_5 - S_1 S_3 C_4 C_5 + C_1 S_2 S_4 C_5 - C_1 C_2 S_3 S_5 - S_1 C_3 S_5$$

$$x_{5_y} = S_1 C_2 C_3 C_4 C_5 + C_1 S_3 C_4 C_5 + S_1 S_2 S_4 C_5 - S_1 C_2 S_3 S_5 + C_1 C_3 S_5$$

$$x_{5_z} = -S_2 C_3 C_4 C_5 + C_2 S_4 C_5 + S_2 S_3 S_5$$

$$y_{5_x} = -C_1 C_2 C_3 S_4 + S_1 S_3 S_4 + C_1 S_2 C_4$$

$$y_{5_y} = -S_1 C_2 C_3 S_4 - C_1 S_3 S_4 + S_1 S_2 C_4$$

$$y_{5_z} = S_2 C_3 S_4 + C_2 C_4$$

$$z_{5_x} = C_1 C_2 C_3 C_4 S_5 - S_1 S_3 C_4 S_5 + C_1 S_2 S_4 S_5 + C_1 C_2 S_3 C_5 + S_1 C_3 C_5$$

$$z_{5_y} = S_1 C_2 C_3 C_4 S_5 + C_1 S_3 C_4 S_5 + S_1 S_2 S_4 S_5 + S_1 C_2 S_3 C_5 - C_1 C_3 C_5$$

$$z_{5_z} = -S_2 C_3 C_4 S_5 + C_2 S_4 S_5 - S_2 S_3 C_5$$

$$p_{5_x} = a_4 C_1 C_2 C_3 C_4 - a_4 S_1 S_3 C_4 + a_4 C_1 S_2 S_4 + a_2 C_1 C_2$$

APPENDIX C

THE WRIST JACOBIAN REFERENCED TO THE BASE FRAME

The wrist Jacobian is represented by the following:

$$J_{\text{wrist}}(\theta) = \begin{bmatrix} \underline{b}_0 \times \underline{r}_{0,5} & \underline{b}_1 \times \underline{r}_{1,5} & \cdots & \underline{b}_6 \times \underline{r}_{6,5} \\ \underline{b}_0 & \underline{b}_1 & \cdots & \underline{b}_6 \end{bmatrix}$$

where

$\underline{r}_{i-1,5}$ = a three dimensional position vector of the origin of the wrist frame from the origin of the $(i-1)^{\text{th}}$ coordinate frame expressed in the base coordinate frame.

$\underline{b}_{i-1}$ = is the unit vector along the axis of motion of joint i expressed in the base coordinate frame.

$$\underline{b}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \underline{b}_1 = \begin{bmatrix} -S_1 \\ C_1 \\ 0 \end{bmatrix} \quad \underline{b}_2 = \begin{bmatrix} C_1 S_2 \\ S_1 S_2 \\ C_2 \end{bmatrix}$$

$$\underline{b}_3 = \begin{bmatrix} C_1 C_2 S_3 + S_1 C_3 \\ S_1 C_2 S_3 - C_1 C_3 \\ -S_2 S_3 \end{bmatrix} \quad \underline{b}_4 = \begin{bmatrix} (S_6 C_7) n_x - (S_6 S_7) s_x - (C_6) a_x \\ (S_6 C_7) n_y - (S_6 S_7) s_y - (C_6) a_y \\ (S_6 C_7) n_z - (S_6 S_7) s_z - (C_6) a_z \end{bmatrix}$$

$$\underline{b}_5 = \begin{bmatrix} S_7 n_x + C_7 s_x \\ S_7 n_y + C_7 s_y \\ S_7 n_z + C_7 s_z \end{bmatrix} \quad \underline{b}_6 = \begin{bmatrix} (p_x - p_{5_x})/d_7 \\ (p_y - p_{5_y})/d_7 \\ (p_z - p_{5_z})/d_7 \end{bmatrix}$$

and

$$\underline{b}_0 \times \underline{r}_{0,5} = \begin{bmatrix} -p_{5_y} \\ p_{5_x} \\ 0 \end{bmatrix} \quad \underline{b}_1 \times \underline{r}_{1,5} = \begin{bmatrix} C_1 p_{5_z} \\ S_1 p_{5_z} \\ -S_1 p_{5_y} - C_1 p_{5_x} \end{bmatrix}$$

$$\underline{b}_2 \times \underline{r}_{2,5} = \begin{bmatrix} S_1 S_2 (p_{5_z} + a_2 S_2) - C_2 (p_{5_y} - a_2 C_2 S_1) \\ -C_1 S_2 (p_{5_z} + a_2 S_2) + C_2 (p_{5_x} - a_2 C_2 C_1) \\ C_1 S_2 (p_{5_y} - a_2 C_2 S_1) - S_1 S_2 (p_{5_x} - a_2 C_2 C_1) \end{bmatrix}$$

$$\underline{b}_3 \times \underline{r}_{3,5} = \begin{bmatrix} (S_1 C_2 S_3 - C_1 C_3) (p_{5_z} + a_2 S_2) + (S_2 S_3) (p_{5_y} - a_2 C_2 S_1) \\ -(C_1 C_2 S_3 + S_1 C_3) (p_{5_z} + a_2 S_2) - (S_2 S_3) (p_{5_x} - a_2 C_2 C_1) \\ (C_1 C_2 S_3 + S_1 C_3) (p_{5_y} - a_2 C_2 S_1) - (p_{5_x} - a_2 C_2 C_1) (S_1 C_2 S_3 - C_1 C_3) \end{bmatrix}$$

$$\underline{b}_4 \times \underline{r}_{4,5} = \underline{b}_5 \times \underline{r}_{5,5} = \underline{b}_6 \times \underline{r}_{6,5} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

COMPUTER PROGRAM

125

```

/*****
      A ROUTINE FOR DETERMINING THE GRADIENTS OF SEVERAL
      PERFORMANCE CRITERIA.
*****/

GetGradientH ( CaseNum,theta,gripper,endeffector,gH,kKind )

JOINT_ANGLES      theta, *gH;
int               CaseNum, *kKind;
HOMOGEN_TRANS     gripper;
INCREMENTS        endeffector;

{

double ddo1[8][2],ddo2[8][2],ddo3[8][2],ddo4[8][2];
double ddo5[8][2],ddo6[8][2],ddo7[8][2];
int i,ii,iii,j,jj,jp;
double c1,c2,c3,c4,c5,c6,c7,s1,s2,s3,s4,s5,s6,s7,a1,a2,a4;
static float etal,eta2,eta3,eta4,eta5,eta6,eta7,e1,e2,e3,
e4,e5,e6,e7;
static float betal,beta2,beta3,beta4,beta5,beta6,beta7;
static float gamal,gama2,gama3,gama4,gama5,gama6,gama7,
g1,g2,g3,g4,g5,g6,g7;
static float alfa1,alfa2,alfa3,alfa4,alfa5,alfa6,alfa7;
double Xa,Xb,Ya,Yb,Za,Zb;
double timec,time_inc;
double px,psx,py,psy,pz,psz;
double x17,x27,x37,y17,y27,y37,z17,z27,z37;
double i1,i2,i3;
double dhdo[8],z[7],z1[7];
static float fx,fy,fz,Mx,My,Mz, CoeF1,CoeF2;
double vx,vy,vz,wx,wy,wz,mx,my,mz;
double zh0x,zh0y,zh0z;
double ja[8][8],jza[8][2],jzat[2][8],jat[8][8],jjati[4][4],
ua[4][2];
double mag_xa;
double jzl[8][2],jzlt[2][8],jl[8][8],ul[4][2],mag_xl,
jjlti[4][4];
double jlt[8][8],empty[8][8],zero[8][8],um[8][8],select[1];
double rva,rvl,rml,m1;
double b1,b2,b3,b4,b5,b6,b7;
int cycle,ND,node_num,iii,cycle_num,iiii;
double loop_time1, loop_time2,time_need;
double jul[8][2],jult[2][8],modulus,mo,rm2,ju2[8][2];
double ju2t[2][8];
double eta,tau,temp,sumsq,sign,sigma,pro,det;
int m,n,numcol,col,numrow,k,row;
double dtal,dtav,magtv;
double a[8][6],at[8][6],r[8][6],ai[6][6];
double att[6][6],aa[6][6],tmpv[6][6],num[6][6];
double qt[6][6],inv[6][6],g[8][8];
double s[8],ss[8],zeta[8];
double jac[7][8],jac1[7][8],jacp[9][8][9],b[7][7],non[8];
int l;

/*      DETERMINATION OF THE TRIGONOMETRIC VALUES
      OF THE JOINT ANGLES      */

c1 = cos(theta.one);
c2 = cos(theta.two);
c3 = cos(theta.three);
c4 = cos(theta.four);
c5 = cos(theta.five);
c6 = cos(theta.six);

```



```

{
    cycle_num = i;
    time_need = loop_time1 + (loop_time2 + EndOfLoopTime)
                *(i-1);
}
}

if ( iteration*delta_t >= timec && iteration*delta_t
    <= timec+total_time[ND] && ND <= The_num_of_nodes )

{
    timec = timec + total_time[ND];
    Xa = x_d[ND-1];
    Xb = x_d[ND];
    Ya = y_d[ND-1];
    Yb = y_d[ND];
    Za = z_d[ND-1];
    Zb = z_d[ND];
    ND = ND + 1;
}
if ( ND == The_num_of_nodes +1 && iteration*delta_t
    >= time_need )

{
    timec = timec + EndOfLoopTime;
    Xa = x_d[ND-1];
    Xb = x_d[1];
    Ya = y_d[ND-1];
    Yb = y_d[1];
    Za = z_d[ND-1];
    Zb = z_d[1];
    ND = 2;
}

loop_time1 = 0.0;
loop_time2 = 0.0;

/*      DETERMINATION OF THE ORIENTATION OF THE END-EFFECTOR
        w.r.t. THE BASE COORDINATE FRAME.      */

x17 = c1*c2*c3*c4*c5*c6*c7 - s1*s3*c4*c5*c6*c7 + c1*s2*s4*c5*c6*c7
- c1*c2*s3*s5*c6*c7 - s1*c3*s5*c6*c7 - c1*c2*c3*s4*s6*c7 +
s1*s3*s4*s6*c7 + c1*s2*c4*s6*c7 + c1*c2*c3*c4*s5*s7 -
s1*s3*c4*s5*s7 + c1*s2*s4*s5*s7 + c1*c2*s3*c5*s7 + s1*c3*c5*s7;

x27 = s1*c2*c3*c4*c5*c6*c7 + c1*s3*c4*c5*c6*c7 + s1*s2*s4*c5*c6*c7
- s1*c2*s3*s5*c6*c7 + c1*c3*s5*c6*c7 - s1*c2*c3*s4*s6*c7 -
c1*s3*s4*s6*c7 + s1*s2*c4*s6*c7 + s1*c2*c3*c4*s5*s7 +
c1*s3*c4*s5*s7 + s1*s2*s4*s5*s7 + s1*c2*s3*c5*s7 - c1*c3*c5*s7;

x37 = -s2*c3*c4*c5*c6*c7 + c2*s4*c5*c6*c7 + s2*s3*s5*c6*c7
+ s2*c3*s4*s6*c7 + c2*c4*s6*c7 - s2*c3*c4*s5*s7 + c2*s4*s5*s7
- s2*s3*c5*s7;

y17 = -c1*c2*c3*c4*c5*c6*s7 + s1*s3*c4*c5*c6*s7 - c1*s2*s4*c5*c6*s7
+ c1*c2*s3*s5*c6*s7 + s1*c3*s5*c6*s7 + c1*c2*c3*s4*s6*s7 -
s1*s3*s4*s6*s7 - c1*s2*c4*s6*s7 + c1*c2*c3*c4*s5*c7 -
s1*s3*c4*s5*c7 + c1*s2*s4*s5*c7 + c1*c2*s3*c5*c7 + s1*c3*c5*c7;

y27 = -s1*c2*c3*c4*c5*c6*s7 - c1*s3*c4*c5*c6*s7 - s1*s2*s4*c5*c6*s7
+ s1*c2*s3*s5*c6*s7 - c1*c3*s5*c6*s7 + s1*c2*c3*s4*s6*s7 +
c1*s3*s4*s6*s7 - s1*s2*c4*s6*s7 + s1*c2*c3*c4*s5*c7 + c1*s3*c4*s5*c7

```

```

+ s1*s2*s4*s5*c7 + s1*c2*s3*c5*c7 - c1*c3*c5*c7;

y37 = s2*c3*c4*c5*c6*s7 - c2*s4*c5*c6*s7 - s2*s3*s5*c6*s7 -
s2*c3*s4*s6*s7 - c2*c4*s6*s7 - s2*c3*c4*s5*c7 + c2*s4*s5*c7
- s2*s3*c5*c7;

z17 = c1*c2*c3*c4*c5*s6 - s1*s3*c4*c5*s6 + c1*s2*s4*c5*s6 -
c1*c2*s3*s5*s6 - s1*c3*s5*s6 + c1*c2*c3*s4*c6 - s1*s3*s4*c6 -
c1*s2*c4*c6;

z27 = s1*c2*c3*c4*c5*s6 + c1*s3*c4*c5*s6 + s1*s2*s4*c5*s6
- s1*c2*s3*s5*s6 + c1*c3*s5*s6 + s1*c2*c3*s4*c6 + c1*s3*s4*c6
- s1*s2*c4*c6;

z37 = -s2*c3*c4*c5*s6 + c2*s4*c5*s6 + s2*s3*s5*s6 - s2*c3*s4*c6
- c2*c4*c6;

/*      DETERMINATION OF THE POSITION OF THE END-EFFECTOR
w.r.t. THE BASE FRAME      */

px = d7*c1*c2*c3*c4*c5*s6 - d7*s1*s3*c4*c5*s6 + d7*c1*s2*s4*c5*s6
- d7*c1*c2*s3*s5*s6 - d7*s1*c3*s5*s6 + d7*c1*c2*c3*s4*c6 -
d7*s1*s3*s4*c6 - d7*c1*s2*c4*c6 + c1*c2*c3*a4*c4 - s1*s3*a4*c4
+ c1*s2*a4*s4 + a2*c2*c1;

py = d7*s1*c2*c3*c4*c5*s6 + d7*c1*s3*c4*c5*s6 + d7*s1*s2*s4*c5*s6
- d7*s1*c2*s3*s5*s6 + d7*c1*c3*s5*s6 + d7*s1*c2*c3*s4*c6 +
d7*c1*s3*s4*c6 - d7*s1*s2*c4*c6 + s1*c2*c3*a4*c4 + c1*s3*a4*c4
+ s1*s2*a4*s4 + a2*c2*s1;

pz = -d7*s2*c3*c4*c5*s6 + d7*c2*s4*c5*s6 + d7*s2*s3*s5*s6 -
d7*s2*c3*s4*c6 - d7*c2*c4*c6 - s2*c3*a4*c4 + c2*a4*s4 - a2*s2;

/*      LINEAR AND ANGULAR VELOCITY OF THE END-EFFECTOR */

vx = endeffector.dx;
vy = endeffector.dy;
vz = endeffector.dz;
wx = endeffector.dThetaX;
wy = endeffector.dThetaY;
wz = endeffector.dThetaZ;

/*      THE END-EFFECTOR Z-AXIS EXPRESSED IN BASE COORDINATES */

zh0x = c1*c2*c3*c4*c5*s6 - s1*s3*c4*c5*s6 + c1*s2*s4*c5*s6
- c1*c2*s3*s5*s6 - s1*c3*s5*s6 + c1*c2*c3*s4*c6
- s1*s3*s4*c6 - c1*s2*c4*c6;
zh0y = s1*c2*c3*c4*c5*s6 + c1*s3*c4*c5*s6 + s1*s2*s4*c5*s6
- s1*c2*s3*s5*s6 + c1*c3*s5*s6 + s1*c2*c3*s4*c6
+ c1*s3*s4*c6 - s1*s2*c4*c6;
zh0z = -s2*c3*c4*c5*s6 + c2*s4*c5*s6 + s2*s3*s5*s6 - s2*c3*s4*c6
- c2*c4*c6;

/*      THE VELOCITY OF THE WRIST REFERENCED TO THE BASE
COORDINATE FRAME      */

vx = vx - d7*( zh0z*wy - zh0y*wz );
vy = vy - d7*(-zh0z*wx + zh0x*wz );
vz = vz - d7*( zh0y*wx - zh0x*wy );

/*      THE MOMEMENT PART OF THE FORCE VECTOR APPLIED
.      AT THE WRIST      */

```

```

mx = Mx + d7*( fz*zh0y - fy*zh0z );
my = My + d7*(-fz*zh0x + fx*zh0z );
mz = Mz + d7*( fy*zh0x - fx*zh0y );

/*      THE POSITION OF THE WRIST w.r.t. THE BASE FRAME      */

psx = px - ( d7*c1*c2*c3*c4*c5*s6 - d7*s1*s3*c4*c5*s6 +
d7*c1*s2*s4*c5*s6 - d7*c1*c2*s3*s5*s6 - d7*s1*c3*s5*s6 +
d7*c1*c2*c3*s4*c6 - d7*s1*s3*s4*c6 - d7*c1*s2*c4*c6 );

psy = py - ( d7*s1*c2*c3*c4*c5*s6 + d7*c1*s3*c4*c5*s6 +
d7*s1*s2*s4*c5*s6 - d7*s1*c2*s3*s5*s6 + d7*c1*c3*s5*s6 +
d7*s1*c2*c3*s4*c6 + d7*c1*s3*s4*c6 - d7*s1*s2*c4*c6 );

psz = pz - ( -d7*s2*c3*c4*c5*s6 + d7*c2*s4*c5*s6 +
d7*s2*s3*s5*s6 - d7*s2*c3*s4*c6 - d7*c2*c4*c6 );

/* IF kKind = 0 THEN THE PERFORMANCE CRITERION IS MINIMIZED
   IF kKind = 1 THEN THE PERFORMANCE CRITERION IS MAXIMIZED */

*kKind = 0;

/*      SELECTING A PERFORMANCE CRITERION

IF CaseNum = 1 THEN THE MANIPULATOR VELOCITY RATIO
IN THE DIRECTION OF MOTION OF THE END-
EFFECTOR IS MAXIMIZED
(FOR LINEAR END-EFFECTOR MOTION CASE)

IF CaseNum = 2 THEN THE MANIPULATOR VELOCITY RATIO
IN THE DIRECTION OF MOTION OF THE END-
EFFECTOR IS MAXIMIZED
(FOR ANGULAR END-EFFECTOR MOTION CASE)

IF CaseNum = 3 THEN THE MANIPULATOR MECHANICAL
ADVANTAGE IN THE DIRECTION OF THE FORCE
APPLIED AT THE WRIST IS MAXIMIZED

IF CaseNum = 4 THEN THE MANIPULATOR MECHANICAL
ADVANTAGE IN THE DIRECTION OF THE MOMENT
APPLIED AT THE WRIST IS MAXIMIZED

IF CaseNum = 5 THEN THE MANIPULABILITY MEASURE USING
THE UPPER HALF PART OF THE WRIST JACOBIAN
IS MAXIMIZED

IF CaseNum = 6 THEN THE MANIPULATOR VELOCITY RATIO IN THE
DIRECTION CORRESPONDING TO THE MINIMUM
SINGULAR VALUE IS MAXIMIZED
(LINEAR END-EFFECTOR MOTION CASE)

IF CaseNum = 7 THEN THE MANIPULATOR VELOCITY RATIO IN THE
DIRECTION CORRESPONDING TO THE MINIMUM
SINGULAR VALUE IS MAXIMIZED
(ANGULAR END-EFFECTOR MOTION CASE)

*/

switch (CaseNum)
(

```

```

case 1:

/*****
      MANIPULATOR VELOCITY RATIO
      IN THE DIRECTION OF MOTION
      (LINEAR MOTION CASE)
*****/

/*      INITIALIZING VARIABLES      */

for ( i = 0; i <= 3; ++i )
{
    for ( ii = 0; ii <= 3; ++ii )
    {
        jlti[i][ii] = 0.0;
    }
}

for ( i = 0; i <= 7; ++i )
{
    for ( ii = 0; ii <= 1; ++ii )
    {
        ddo1[i][ii] = 0.0;
        ddo2[i][ii] = 0.0;
        ddo3[i][ii] = 0.0;
        ddo4[i][ii] = 0.0;
        ddo5[i][ii] = 0.0;
        ddo6[i][ii] = 0.0;
        ddo7[i][ii] = 0.0;
        jz1[i][ii] = 0.0;
    }
}

for ( i = 0; i <= 1; ++i )
{
    for ( ii = 0; ii <= 7; ++ii )
    {
        jzlt[i][ii] = 0.0;
    }
}

for ( i = 0; i <= 7; ++i )
{
    dhdo[i] = 0.0;
    z[i] = 0.0;
}

for ( i = 0; i <= 7; ++i )
{
    for ( ii = 0; ii <= 3; ++ii )
    {
        jlt[i][ii] = 0.0;
    }
}

for ( i = 0; i <= 7; ++i )
{
    for ( ii = 0; ii <= 7; ++ii )
    {
        jl[i][ii] = 0.0;
        empty[i][ii] = 0.0;
    }
}

/*      WEIGHTING FACTORS ASSOCIATED WITH THE

```

VELOCITY OF THE WRIST AND THE JACOBIAN */

```
g1 = pow(gama1,0.5);
g2 = pow(gama2,0.5);
g3 = pow(gama3,0.5);
e1 = pow(eta1,-0.5);
e2 = pow(eta2,-0.5);
e3 = pow(eta3,-0.5);
e4 = pow(eta4,-0.5);
```

/* DETERMINING THE END-EFFECTOR DESIRED VELOCITY */

```
mag_x1 = (Xb-Xa)*(Xb-Xa) + (Yb-Ya)*(Yb-Ya) + (Zb-Za)*(Zb-Za);
mag_x1 = pow(mag_x1,0.5);
ul[1][1] = (1.0/mag_x1)*(Xb-Xa);
ul[2][1] = (1.0/mag_x1)*(Yb-Ya);
ul[3][1] = (1.0/mag_x1)*(Zb-Za);
```

/* THE TRANSFORMED WRIST JACOBIAN RELATED TO
LINEAR MOTION */

```
j1[1][1] = -g1*psy*e1;
j1[1][2] = g1*c1*psz*e2;
j1[1][3] = g1*( s1*s2*(psz+a2*s2) - c2*(psy-a2*c2*s1) )*e3;
j1[1][4] = g1*( (s1*c2*s3-c1*c3)*(psz+a2*s2) + (s2*s3)*
    (psy-a2*c2*s1) )*e4;
j1[2][1] = g2*psx*e1;
j1[2][2] = g2*s1*psz*e2;
j1[2][3] = g2*( -c1*s2*(psz+a2*s2) + c2*(psx-a2*c2*c1) )*e3;
j1[2][4] = g2*( -(c1*c2*s3+s1*c3)*(psz+a2*s2) - (s2*s3)*
    (psx-a2*c2*c1) )*e4;
j1[3][2] = g3*(-s1*psy-c1*psx)*e2;
j1[3][3] = g3*( c1*s2*(psy-a2*c2*s1)-s1*s2*(psx-a2*c2*c1))*e3;
j1[3][4] = g3*( (c1*c2*s3+s1*c3)*(psy-a2*c2*s1) -
    (psx-a2*c2*c1)*(s1*c2*s3-c1*c3) )*e4;
```

/* THE TRANSPOSE OF THE JACOBIAN */

```
for ( i = 1; i <= 7; ++i )
{
    for ( ii = 1; ii <= 3; ++ii )
    {
        jlt[i][ii] = j1[ii][i];
    }
}
```

/* CALLING THE PSUEDO-INVERSE ROUTINE */

```
psuedo_inv ( j1,empty,select,w );
/* j1 is now a 3x3 inverse matrix that contains
   the inverse of JJT */
```

/* THE INVERSE OF JJT */

```
for ( i = 1; i <= 3; ++i )
{
    for ( ii = 1; ii <= 3; ++ii )
    {
        jjlti[i][ii] = j1[i][ii];
    }
}
```

```

/*      z IS THE PRODUCT OF THE INVERSE OF JJT BY
      THE VECTOR IN THE DIRECTION OF MOTION      */

for ( i = 1; i <= 3; ++i )
{
    for ( iii = 1; iii <= 3; ++iii )
    {
        z[i] = jjlti[i][iii]*ul[iii][1] + z[i];
    }
}

/*      THE MVR FOR LINEAR MOTION      */

rvl = 1.0/(sqrt(ul[1][1]*z[1] + ul[2][1]*z[2] + ul[3][1]*z[3]));

fprintf(singularity, "%d %f\n", iteration, rvl);

/*      THESE ARE THE ENTRIES OF THE FOLLOWING PRODUCT:

      (TRANSPOSE OF THE JACOBIAN)*(Z-VECTOR)      */

i1 = psz + a2*s2;
i2 = psy - a2*c2*s1;
i3 = psx - a2*c2*c1;

jzl[1][1] = -g1*psy*e1*z[1] + g2*psx*e1*z[2];
jzl[2][1] = g1*c1*psz*e2*z[1] + g2*s1*psz*e2*z[2]
+ g3*(-s1*psy-c1*psx)*e2*z[3];
jzl[3][1] = g1*( s1*s2*i1 - c2*i2 )*e3*z[1]
+ g2*( -c1*s2*i1 + c2*i3 )*e3*z[2]
+ g3*( c1*s2*i2 - s1*s2*i3 )*e3*z[3];
jzl[4][1] = g1*( (s1*c2*s3-c1*c3)*i1 + (s2*s3)*i2 )*e4*z[1]
+ g2*( -(c1*c2*s3+s1*c3)*i1 - (s2*s3)*i3 )*e4*z[2]
+ g3*( (c1*c2*s3+s1*c3)*i2 - i3*(s1*c2*s3-c1*c3) )*e4*z[3];
jzl[5][1] = 0.0;
jzl[6][1] = 0.0;
jzl[7][1] = 0.0;

/*      PARTIAL DIFFERENTIAL OF jzl W.r.t. THETA      */

ddol[1][1] = -g1*(c1*c2*c3*a4*c4 - s1*s3*a4*c4 + c1*s2*a4*s4
+ a2*c2*c1)*e1*z[1] + g2*(-s1*c2*c3*a4*c4 - c1*s3*a4*c4 -
s1*s2*a4*s4 - a2*c2*s1)*e1*z[2];

ddol[2][1] = g1*(s1*s2*c3*a4*c4 - s1*c2*a4*s4 + s1*a2*s2)*e2*z[1]
+ g2*(-c1*s2*c3*a4*c4 + c1*c2*a4*s4 - c1*a2*s2)*e2*z[2];

ddol[3][1] = g1*(-c1*c3*a4*c4 + s1*c2*s3*a4*c4)*e3*z[1] +
g2*(-s1*c3*a4*c4 - c1*c2*s3*a4*c4)*e3*z[2];

ddol[4][1] = g1*(-s2*s1*a4*c4 + c1*s3*a4*s4 +
s1*c3*c2*a4*s4)*e4*z[1] + g2*(c1*s2*a4*c4 + s1*s3*a4*s4 -
c1*c3*c2*a4*s4)*e4*z[2];

ddo2[1][1] = -g1*(-s1*s2*c3*a4*c4 + s1*c2*a4*s4 -
a2*s2*s1)*e1*z[1] + g2*(-c1*s2*c3*a4*c4 + c1*c2*a4*s4 -
a2*s2*c1)*e1*z[2];

ddo2[2][1] = g1*(-c1*c2*c3*a4*c4 - c1*s2*a4*s4 -
c1*a2*c2)*e2*z[1] + g2*(-s1*c2*c3*a4*c4 - s1*s2*a4*s4 -
s1*a2*c2)*e2*z[2] + g3*(s2*c3*a4*c4 - c2*a4*s4 + a2*s2)*e2*z[3];

ddo2[3][1] = g1*(c1*s2*s3*a4*c4)*e3*z[1] +
g2*(s1*s2*s3*a4*c4)*e3*z[2] + g3*(c2*s3*a4*c4)*e3*z[3];

```

```

ddo2[4][1] = g1*(c2*c1*a4*c4 + c1*c3*s2*a4*s4)*e4*z[1] +
g2*(s1*c2*a4*c4 + s1*c3*s2*a4*s4)*e4*z[2] + g3*(-s2*a4*c4
+ c3*c2*a4*s4)*e4*z[3];

ddo3[1][1] = -g1*(-s1*c2*s3*a4*c4 + c1*c3*a4*c4)*e1*z[1]
+ g2*(-c1*c2*s3*a4*c4 - s1*c3*a4*c4)*e1*z[2];

ddo3[2][1] = g1*(c1*s2*s3*a4*c4)*e2*z[1] +
g2*(s1*s2*s3*a4*c4)*e2*z[2] + g3*(c2*s3*a4*c4)*e2*z[3];

ddo3[3][1] = g1*(s1*s3*a4*c4 - c1*c2*c3*a4*c4)*e3*z[1]
+ g2*(-c1*s3*a4*c4 - s1*c2*c3*a4*c4)*e3*z[2] +
g3*(s2*c3*a4*c4)*e3*z[3];

ddo3[4][1] = g1*(s1*c3*a4*s4 + c1*s3*c2*a4*s4)*e4*z[1] +
g2*(-c1*c3*a4*s4 + s1*s3*c2*a4*s4)*e4*z[2] +
g3*(-s3*s2*a4*s4)*e4*z[3];

ddo4[1][1] = -g1*(-s1*c2*c3*a4*s4 - c1*s3*a4*s4 +
s1*s2*a4*c4)*e1*z[1] + g2*(-c1*c2*c3*a4*s4 + s1*s3*a4*s4 +
c1*s2*a4*c4)*e1*z[2];

ddo4[2][1] = g1*(c1*s2*c3*a4*s4 + c1*c2*a4*c4)*e2*z[1] +
g2*(s1*s2*c3*a4*s4 + s1*c2*a4*c4)*e2*z[2] + g3*(c2*c3*a4*s4
- s2*a4*c4)*e2*z[3];

ddo4[3][1] = g1*(s1*c3*a4*s4 + c1*c2*s3*a4*s4)*e3*z[1]
+ g2*(-c1*c3*a4*s4 + s1*c2*s3*a4*s4)*e3*z[2] +
g3*(-s2*s3*a4*s4)*e3*z[3];

ddo4[4][1] = g1*(-s2*c1*a4*s4 + s1*s3*a4*c4 -
c1*c3*c2*a4*c4)*e4*z[1] + g2*(-s1*s2*a4*s4 - c1*s3*a4*c4
-s1*c3*c2*a4*c4)*e4*z[2] + g3*(-c2*a4*s4 + c3*s2*a4*c4)*e4*z[3];

for ( i = 1; i <= 7; ++i )
{
    jzlt[1][i] = jz1[i][1];
}

/*      DETERMINING THE GRADIENT OF THE PERFORMANCE
        CRITERION      */

j = 1;
for ( i = 1; i <= 7; ++i )
{
    dhdo[j] = -2.0*jzlt[1][i]*ddo1[i][1] + dhdo[j];
    dhdo[j+1] = -2.0*jzlt[1][i]*ddo2[i][1] + dhdo[j+1];
    dhdo[j+2] = -2.0*jzlt[1][i]*ddo3[i][1] + dhdo[j+2];
    dhdo[j+3] = -2.0*jzlt[1][i]*ddo4[i][1] + dhdo[j+3];
    dhdo[j+4] = -2.0*jzlt[1][i]*ddo5[i][1] + dhdo[j+4];
    dhdo[j+5] = -2.0*jzlt[1][i]*ddo6[i][1] + dhdo[j+5];
    dhdo[j+6] = -2.0*jzlt[1][i]*ddo7[i][1] + dhdo[j+6];
}

gH->one = dhdo[1];
gH->two = dhdo[2];
gH->three = dhdo[3];
gH->four = dhdo[4];
gH->five = dhdo[5];
gH->six = dhdo[6];
gH->seven = dhdo[7];

```

```
break;
```

```
case 2:
```

```
/******  
      MANIPULATOR VELOCITY RATIO IN THE DIRECTION  
      OF MOTION (ANGULAR MOTION CASE)  
******/
```

```
/*      INITIALIZING VARIABLES      */
```

```
for ( i = 0; i <= 3; ++i )  
{  
    for ( ii = 0; ii <= 3; ++ii )  
    {  
        jjati[i][ii] = 0.0;  
    }  
}
```

```
for ( i = 0; i <= 3; ++i )  
{  
    for ( ii = 0; ii <= 1; ++ii )  
    {  
        ua[i][ii] = 0.0;  
    }  
}
```

```
for ( i = 0; i <= 7; ++i )  
{  
    for ( ii = 0; ii <= 1; ++ii )  
    {  
        ddo1[i][ii] = 0.0;  
        ddo2[i][ii] = 0.0;  
        ddo3[i][ii] = 0.0;  
        ddo4[i][ii] = 0.0;  
        ddo5[i][ii] = 0.0;  
        ddo6[i][ii] = 0.0;  
        ddo7[i][ii] = 0.0;  
        jza[i][ii] = 0.0;  
    }  
}
```

```
for ( i = 0; i <= 1; ++i )  
{  
    for ( ii = 0; ii <= 7; ++ii )  
    {  
        jzat[i][ii] = 0.0;  
    }  
}
```

```
for ( i = 0; i <= 7; ++i )  
{
```

```
    dhdo[i] = 0.0;  
    z[i] = 0.0;
```

```
}  
for ( i = 0; i <= 7; ++i )  
{  
    for ( ii = 0; ii <= 3; ++ii )  
    {  
        jat[i][ii] = 0.0;  
    }  
}
```

```

for ( i = 0; i <= 7; ++i )
{
    for ( ii = 0; ii <= 7; ++ii )
    {
        ja[i][ii] = 0.0;
        zero[i][ii] = 0.0;
    }
}

/*      WEIGHTING FACTORS      */

g6 = pow(gama6,0.5);
g5 = pow(gama5,0.5);
g4 = pow(gama4,0.5);
e1 = pow(eta1,-0.5);
e2 = pow(eta2,-0.5);
e3 = pow(eta3,-0.5);
e4 = pow(eta4,-0.5);
e5 = pow(eta5,-0.5);
e6 = pow(eta6,-0.5);
e7 = pow(eta7,-0.5);

/*      THE ANGULAR VELOCITY OF THE WRIST      */

mag_xa = g4*g4*wx*wx + g5*g5*wy*wy + g6*g6*wz*wz;
mag_xa = pow(mag_xa,0.5);

ua[1][1] = (1.0/mag_xa)*g4*wx;
ua[2][1] = (1.0/mag_xa)*g5*wy;
ua[3][1] = (1.0/mag_xa)*g6*wz;

/*      THE ANGULAR MOTION PART OF THE WRIST JACOBIAN      */

ja[1][1] = 0.0;
ja[1][2] = -g4*s1*e2;
ja[1][3] = g4*c1*s2*e3;
ja[1][4] = g4*( c1*c2*s3+s1*c3 )*e4;
ja[1][5] = -g4*( (-s6*c7)*x17 + (s6*s7)*y17 + c6*z17 )*e5;
ja[1][6] = g4*( s7*x17 + c7*y17 )*e6;
ja[1][7] = g4*( (px-psx)/d7 )*e7;
ja[2][1] = 0.0;
ja[2][2] = g5*c1*e2;
ja[2][3] = g5*s1*s2*e3;
ja[2][4] = g5*( s1*c2*s3-c1*c3 )*e4;
ja[2][5] = -g5*( (-s6*c7)*x27 + (s6*s7)*y27 + c6*z27 )*e5;
ja[2][6] = g5*( s7*x27 + c7*y27 )*e6;
ja[2][7] = g5*( (py-psy)/d7 )*e7;
ja[3][1] = g6*e1;
ja[3][2] = 0.0;
ja[3][3] = g6*c2*e3;
ja[3][4] = g6*( -s2*s3 )*e4;
ja[3][5] = -g6*( (-s6*c7)*x37 + (s6*s7)*y37 + c6*z37 )*e5;
ja[3][6] = g6*( s7*x37 + c7*y37 )*e6;
ja[3][7] = g6*( (pz-psz)/d7 )*e7;

for ( i = 1; i <= 7; ++i )
{
    for ( ii = 1; ii <= 3; ++ii )
    {
        jat[i][ii] = ja[ii][i];
    }
}

```

```

/*      CALLING THE PSUEDO-INVERSE ROUTINE      */

psuedo_inv ( ja,zero,select,w );
/* ja is now a 3x3 inverse matrix */

/*      THE INVERSE OF JJT      */

for ( i = 1; i <= 3; ++i )
{
    for ( ii = 1; ii <= 3; ++ii )
    {
        jjati[i][ii] = ja[i][ii];
    }
}

/*      z IS THE PRODUCT OF THE INVERSE OF JJT BY
THE VECTOR IN THE DIRECTION OF ANGULAR
ROTATION OF THE WRIST      */

for ( i = 1; i <= 3; ++i )
{
    for ( iii = 1; iii <= 3; ++iii )
    {
        z[i+3] = jjati[i][iii]*ua[iii][1] + z[i+3];
    }
}

/*      MVR FOR ANGULAR MOTION      */

rva = 1.0/(sqrt(ua[1][1]*z[4] + ua[2][1]*z[5] + ua[3][1]*z[6]));
fprintf(singularity, "%f\n", rva);

/*      PRODUCT OF:
      (TRANSPOSE OF THE JACOBIAN)*(Z-VECTOR)      */

jza[1][1] = g6*e1*z[6];
jza[2][1] = -g4*s1*e2*z[4] + g5*c1*e2*z[5];
jza[3][1] = g4*c1*s2*e3*z[4] + g5*s1*s2*e3*z[5] + g6*c2*e3*z[6];
jza[4][1] = g4*(c1*c2*s3+s1*c3)*e4*z[4]
+g5*(s1*c2*s3-c1*c3)*e4*z[5]
+ g6*(-s2*s3)*e4*z[6];
jza[5][1] = -g4*( -s6*c7*x17 + s6*s7*y17 + c6*z17 )*e5*z[4]
- g5*( -s6*c7*x27 + s6*s7*y27 + c6*z27 )*e5*z[5]
- g6*( -s6*c7*x37 + s6*s7*y37 + c6*z37 )*e5*z[6];
jza[6][1] = g4*(s7*x17 + c7*y17)*e6*z[4]
+ g5*(s7*x27 + c7*y27)*e6*z[5]
+ g6*(s7*x37 + c7*y37)*e6*z[6];
jza[7][1] = g4*( (px-psx)/d7 )*e7*z[4] + g5*( (py-psy)/d7 )*e6*z[5]
+ g6*( (pz-psz)/d7 )*e7*z[6];

/*      PARTIAL DIFFERENTIAL OF jza w.r.t. THETA      */

ddol[2][1] = -g4*c1*e2*z[4] - g5*s1*e2*z[5];
ddol[3][1] = -g4*s1*s2*e3*z[4] + g5*c1*s2*e3*z[5];
ddol[4][1] = g4*(-s1*c2*s3+c1*c3)*e4*z[4]
+ g5*(c1*c2*s3+s1*c3)*e4*z[5];

```

```

ddo2[3][1] = g4*c1*c2*e3*z[4] + g5*s1*c2*e3*z[5] - g6*s2*e3*z[6];
ddo2[4][1] = g4*(-c1*s2*s3)*e4*z[4] + g5*(-s1*s2*s3)*e4*z[5]
              + g6*(-c2*s3)*e4*z[6];

ddo3[4][1] = g4*(c1*c2*c3-s1*s3)*e4*z[4]
              + g5*(s1*c2*c3+c1*s3)*e4*z[5]
              + g6*(-s2*c3)*e4*z[6];

ddo6[5][1] = -g4*( -c6*c7*x17 + c6*s7*y17 - s6*z17 )*e5*z[4]
              - g5*( -c6*c7*x27 + c6*s7*y27 - s6*z27 )*e5*z[5]
              - g6*( -c6*c7*x37 + c6*s7*y37 - s6*z37 )*e5*z[6];

ddo7[5][1] = -g4*( s6*s7*x17 + s6*c7*y17 )*e5*z[4]
              - g5*( s6*s7*x27 + s6*c7*y27 )*e5*z[5]
              - g6*( s6*s7*x37 + s6*c7*y37 )*e5*z[6];
ddo7[6][1] = g4*(c7*x17 - s7*y17)*e6*z[4]
              + g5*(c7*x27 - s7*y27)*e6*z[5]
              + g6*(c7*x37 - s7*y37)*e6*z[6];

for ( i = 1; i <= 7; ++i )
{
    jzat[1][i] = jza[i][1];
}

/*      DETERMINATION OF THE GRADIENT      */

j = 1;
for ( i = 1; i <= 7; ++i )
{
    dhdo[j] = -2.0*jzat[1][i]*ddo1[i][1] + dhdo[j];
    dhdo[j+1] = -2.0*jzat[1][i]*ddo2[i][1] + dhdo[j+1];
    dhdo[j+2] = -2.0*jzat[1][i]*ddo3[i][1] + dhdo[j+2];
    dhdo[j+3] = -2.0*jzat[1][i]*ddo4[i][1] + dhdo[j+3];
    dhdo[j+4] = -2.0*jzat[1][i]*ddo5[i][1] + dhdo[j+4];
    dhdo[j+5] = -2.0*jzat[1][i]*ddo6[i][1] + dhdo[j+5];
    dhdo[j+6] = -2.0*jzat[1][i]*ddo7[i][1] + dhdo[j+6];
}

gH->one = dhdo[1];
gH->two = dhdo[2];
gH->three = dhdo[3];
gH->four = dhdo[4];
gH->five = dhdo[5];
gH->six = dhdo[6];
gH->seven = dhdo[7];

break;

case 3:

/*****
    MANIPULATOR MECHANICAL ADVANTAGE IN THE
    DIRECTION OF THE FORCE APPLIED AT THE
    WRIST
*****/

/*      INITIALIZING VARIABLES      */

for ( i = 1; i <= 7; ++i )
{
    dhdo[i] = 0.0;

```

```

    }

    for ( i = 0; i <= 7; ++i )
    {
        for ( ii = 0; ii <= 1; ++ii )
        {
            ddo1[i][ii] = 0.0;
            ddo2[i][ii] = 0.0;
            ddo3[i][ii] = 0.0;
            ddo4[i][ii] = 0.0;
            ddo5[i][ii] = 0.0;
            ddo6[i][ii] = 0.0;
            ddo7[i][ii] = 0.0;
        }
    }

    /*      WEIGHTING FACTORS      */

    b1 = pow(beta1,0.5);
    b2 = pow(beta2,0.5);
    b3 = pow(beta3,0.5);
    b4 = pow(beta4,0.5);

    /*      THE MAGNITUDE OF THE FORCE VECTOR      */

    m1 = sqrt( alfa1*fx*fx + alfa2*fy*fy + alfa3*fz*fz );
    m1 = 1.0/m1;

    /*      THESE ARE THE ENTRIES OF THE FOLLOWING VECTOR
           THAT IS THE PRODUCT OF :
           (TRANSPOSE OF THE JACOBIAN)*(DIRECTION OF FORCE) */

    i1 = psz + a2*s2;
    i2 = psy - a2*c2*s1;
    i3 = psx - a2*c2*c1;

    jul[1][1] = m1*(-fx*psy*pow(beta1,0.5) + fy*psx*pow(beta1,0.5));
    jul[2][1] = m1*(fx*c1*psz*pow(beta2,0.5)
        + fy*s1*psz*pow(beta2,0.5)
        + fz*(-s1*psy -c1*psx)*pow(beta2,0.5));
    jul[3][1] = m1*(fx*pow(beta3,0.5)*( s1*s2*i1 - c2*i2 )
        + fy*pow(beta3,0.5)*( -c1*s2*i1 + c2*i3 )
        + fz*pow(beta3,0.5)*( c1*s2*i2 - s1*s2*i3 ));
    jul[4][1] = m1*(fx*pow(beta4,0.5)*((s1*c2*s3-c1*c3)*i1+(s2*s3)*i2)
        + fy*pow(beta4,0.5)*(-(c1*c2*s3+s1*c3)*i1 - (s2*s3)*i3 )
        + fz*pow(beta4,0.5)*((c1*c2*c3+s1*c3)*i2-(s1*c2*s3-c1*c3)*i3));
    jul[5][1] = 0.0;
    jul[6][1] = 0.0;
    jul[7][1] = 0.0;

    /*      THE MMA IN THE DIRECTION OF THE APPLIED FORCE      */

    rml = sqrt(jul[1][1]*jul[1][1]+jul[2][1]*jul[2][1]+jul[3][1]*jul[3][1]
        + jul[4][1]*jul[4][1]+jul[5][1]*jul[5][1]+jul[6][1]*jul[6][1]
        + jul[7][1]*jul[7][1]);
    rml = 1.0/rml;

    fprintf(singularity, "%f\n", rml);

    /*      PARTIAL DIFFERENTIAL OF jul w.r.t. THETA      */

```

```

ddol[1][1] = -m1*fx*(c1*c2*c3*a4*c4 - s1*s3*a4*c4 +
c1*s2*a4*s4 + a2*c2*c1)*b1 + m1*fy*(-s1*c2*c3*a4*c4 -
c1*s3*a4*c4 - s1*s2*a4*s4 - a2*c2*s1)*b1;

ddol[2][1] = m1*fx*(s1*s2*c3*a4*c4 - s1*c2*a4*s4 +
s1*a2*s2)*b2 + m1*fy*(-c1*s2*c3*a4*c4 + c1*c2*a4*s4 -
c1*a2*s2)*b2;

ddol[3][1] = m1*fx*(-c1*c3*a4*c4 + s1*c2*s3*a4*c4)*b3 +
m1*fy*(-s1*c3*a4*c4 - c1*c2*s3*a4*c4)*b3;

ddol[4][1] = m1*fx*(-s2*s1*a4*c4 + c1*s3*a4*s4 +
s1*c3*c2*a4*s4)*b4 + m1*fy*(c1*s2*a4*c4 + s1*s3*a4*s4 -
c1*c3*c2*a4*s4)*b4;

ddo2[1][1] = -m1*fx*(-s1*s2*c3*a4*c4 + s1*c2*a4*s4 -
a2*s2*s1)*b1 + m1*fy*(-c1*s2*c3*a4*c4 + c1*c2*a4*s4 -
a2*s2*c1)*b1;

ddo2[2][1] = m1*fx*(-c1*c2*c3*a4*c4 - c1*s2*a4*s4 -
c1*a2*c2)*b2 + m1*fy*(-s1*c2*c3*a4*c4 - s1*s2*a4*s4 -
s1*a2*c2)*b2 + m1*fz*(s2*c3*a4*c4 - c2*a4*s4 + a2*s2)*b2;

ddo2[3][1] = m1*fx*(c1*s2*s3*a4*c4)*b3 +
m1*fy*(s1*s2*s3*a4*c4)*b3 + m1*fz*(c2*s3*a4*c4)*b3;

ddo2[4][1] = m1*fx*(c2*c1*a4*c4 + c1*c3*s2*a4*s4)*b4 +
m1*fy*(s1*c2*a4*c4 + s1*c3*s2*a4*s4)*b4 + m1*fz*(-s2*a4*c4
+ c3*c2*a4*s4)*b4;

ddo3[1][1] = -m1*fx*(-s1*c2*s3*a4*c4 + c1*c3*a4*c4)*b1 +
m1*fy*(-c1*c2*s3*a4*c4 - s1*c3*a4*c4)*b1;

ddo3[2][1] = m1*fx*(c1*s2*s3*a4*c4)*b2 +
m1*fy*(s1*s2*s3*a4*c4)*b2 + m1*fz*(c2*s3*a4*c4)*b2;

ddo3[3][1] = m1*fx*(s1*s3*a4*c4 - c1*c2*c3*a4*c4)*b3 +
m1*fy*(-c1*s3*a4*c4 - s1*c2*c3*a4*c4)*b3 +
m1*fz*(s2*c3*a4*c4)*b3;

ddo3[4][1] = m1*fx*(s1*c3*a4*s4 + c1*s3*c2*a4*s4)*b4 +
m1*fy*(-c1*c3*a4*s4 + s1*s3*c2*a4*s4)*b4 +
m1*fz*(-s3*s2*a4*s4)*b4;

ddo4[1][1] = -m1*fx*(-s1*c2*c3*a4*s4 - c1*s3*a4*s4 +
s1*s2*a4*c4)*b1 + m1*fy*(-c1*c2*c3*a4*s4 + s1*s3*a4*s4 +
c1*s2*a4*c4)*b1;

ddo4[2][1] = m1*fx*(c1*s2*c3*a4*s4 + c1*c2*a4*c4)*b2 +
m1*fy*(s1*s2*c3*a4*s4 + s1*c2*a4*c4)*b2 +
m1*fz*(c2*c3*a4*s4 - s2*a4*c4)*b2;

ddo4[3][1] = m1*fx*(s1*c3*a4*s4 + c1*c2*s3*a4*s4)*b3 +
m1*fy*(-c1*c3*a4*s4 + s1*c2*s3*a4*s4)*b3 +
m1*fz*(-s2*s3*a4*s4)*b3;

ddo4[4][1] = m1*fx*(-s2*c1*a4*s4 + s1*s3*a4*c4 -
c1*c3*c2*a4*c4)*b4 + m1*fy*(-s1*s2*a4*s4 - c1*s3*a4*c4 -
s1*c3*c2*a4*c4)*b4 + m1*fz*(-c2*a4*s4 + c3*s2*a4*c4)*b4;

for ( i = 1; i <= 7; ++i )
{
    jult[1][i] = jul[i][1];
}

```

```

}

/*      GRADIENT OF THE PERFORMANCE CRITERION      */

j = 1;
for ( i = 1; i <= 7; ++i )
{
    dhdo[j] = 2.0*jult[1][i]*ddo1[i][1] + dhdo[j];
    dhdo[j+1] = 2.0*jult[1][i]*ddo2[i][1] + dhdo[j+1];
    dhdo[j+2] = 2.0*jult[1][i]*ddo3[i][1] + dhdo[j+2];
    dhdo[j+3] = 2.0*jult[1][i]*ddo4[i][1] + dhdo[j+3];
    dhdo[j+4] = 2.0*jult[1][i]*ddo5[i][1] + dhdo[j+4];
    dhdo[j+5] = 2.0*jult[1][i]*ddo6[i][1] + dhdo[j+5];
    dhdo[j+6] = 2.0*jult[1][i]*ddo7[i][1] + dhdo[j+6];
}

    gH->one = dhdo[1];
    gH->two = dhdo[2];
    gH->three = dhdo[3];
    gH->four = dhdo[4];
    gH->five = dhdo[5];
    gH->six = dhdo[6];
    gH->seven = dhdo[7];

break;

case 4:

/*****
    MANIPULATOR MECHANICAL ADVANTAGE IN THE
    DIRECTION OF THE MOMENT APPLIED AT THE
    WRIST
*****/

/*      INITIALIZING VARIABLES      */

for ( i = 1; i <= 7; ++i )
{
    dhdo[i] = 0.0;
}

for ( i = 0; i <= 7; ++i )
{
    for ( ii = 0; ii <= 1; ++ii )
    {
        ddo1[i][ii] = 0.0;
        ddo2[i][ii] = 0.0;
        ddo3[i][ii] = 0.0;
        ddo4[i][ii] = 0.0;
        ddo5[i][ii] = 0.0;
        ddo6[i][ii] = 0.0;
        ddo7[i][ii] = 0.0;
    }
}

/*      WEIGHTING FACTORS      */

b1 = pow(beta1,0.5);
b2 = pow(beta2,0.5);
b3 = pow(beta3,0.5);
b4 = pow(beta4,0.5);

```

```

b5 = pow(beta5,0.5);
b6 = pow(beta6,0.5);
b7 = pow(beta7,0.5);

/*      MAGNITUDE OF THE APPLIED MOMENT      */

modulus = sqrt(alfa4*mx*mx + alfa5*my*my + alfa6*mz*mz);
mo = 1.0/modulus;

/*      THESE ARE THE ENTRIES OF THE VECTOR THAT IS THE
      PRODUCT OF:
      (TRANSPOSE OF THE JACOBIAN)*(MOMENT DIRECTION)
      */

ju2[1][1] = mo*mz*pow(beta1,0.5);
ju2[2][1] = mo*(-s1*mx*pow(beta2,0.5) + c1*my*pow(beta2,0.5));
ju2[3][1] = mo*(mx*c1*s2*pow(beta3,0.5) + my*s1*s2*pow(beta3,0.5)
      + mz*c2*pow(beta3,0.5));
ju2[4][1] = mo*(mx*(c1*c2*s3 + s1*c3)*pow(beta4,0.5)
      + my*(s1*c2*s3 - c1*c3)*pow(beta4,0.5)
      + mz*(-s2*s3)*pow(beta4,0.5));
ju2[5][1] = -mo*(mx*(-s6*c7*x17+s6*s7*y17+c6*z17)*pow(beta5,0.5)
      + my*(-s6*c7*x27+s6*s7*y27+c6*z27)*pow(beta5,0.5)
      + mz*(-s6*c7*x37+s6*s7*y37+c6*z37)*pow(beta5,0.5));
ju2[6][1] = mo*(mx*(s7*x17 + c7*y17)*pow(beta6,0.5)
      + my*(s7*x27 + c7*y27)*pow(beta6,0.5)
      + mz*(s7*x37 + c7*y37)*pow(beta6,0.5));
ju2[7][1] = mo*(mx*((px-psx)/d7)*pow(beta7,0.5)
      + my*((py-psy)/d7)*pow(beta7,0.5)
      + mz*((pz-psz)/d7)*pow(beta7,0.5));

/*      THE MMA IN THE DIRECTION OF THE APPLIED MOMENT      */

rm2 = sqrt(ju2[1][1]*ju2[1][1]+ju2[2][1]*ju2[2][1]+ju2[3][1]*ju2[3][1]
      + ju2[4][1]*ju2[4][1]+ju2[5][1]*ju2[5][1]+ju2[6][1]*ju2[6][1]
      + ju2[7][1]*ju2[7][1]);
rm2 = 1.0/rm2;

fprintf(singularity, "%f\n", rm2);

/*      DIFFERENTIAL OF ju2 w.r.t. THETA      */

ddol[2][1] = -mo*c1*mx*pow(beta2,0.5) - mo*s1*my*pow(beta2,0.5);
ddol[3][1] = -mo*mx*s1*s2*pow(beta3,0.5)+mo*my*c1*s2*pow(beta3,0.5);
ddol[4][1] = mo*mx*(-s1*c2*s3 + c1*c3)*pow(beta4,0.5)
      + mo*my*(c1*c2*s3 + s1*c3)*pow(beta4,0.5);

ddo2[3][1] = mo*mx*c1*c2*pow(beta3,0.5) + mo*my*s1*c2*pow(beta3,0.5)
      - mo*mz*s2*pow(beta3,0.5);
ddo2[4][1] = mo*mx*(-c1*s2*s3)*pow(beta4,0.5)
      + mo*my*(-s1*s2*s3)*pow(beta4,0.5)
      + mo*mz*(-c2*s3)*pow(beta4,0.5);

ddo3[4][1] = mo*mx*(c1*c2*c3 - s1*s3)*pow(beta4,0.5)
      + mo*my*(s1*c2*c3 + c1*s3)*pow(beta4,0.5)
      + mo*mz*(-s2*c3)*pow(beta4,0.5);

ddo6[5][1] = -mo*mx*(-c6*c7*x17 + c6*s7*y17 - s6*z17)*pow(beta5,0.5)
      - mo*my*(-c6*c7*x27+c6*s7*y27-s6*z27)*pow(beta5,0.5)
      - mo*mz*(-c6*c7*x37+c6*s7*y37-s6*z37)*pow(beta5,0.5);

ddo7[5][1] = -mo*mx*(s6*s7*x17 + s6*c7*y17)*pow(beta5,0.5)
      - mo*my*(s6*s7*x27 + s6*c7*y27)*pow(beta5,0.5)

```

```

        - mo*mz*( s6*s7*x37 + s6*c7*y37 )*pow(beta5,0.5);
ddo7[6][1] = mo*mx*( c7*x17 - s7*y17 )*pow(beta6,0.5)
        + mo*my*( c7*x27 - s7*y27 )*pow(beta6,0.5)
        + mo*mz*( c7*x37 - s7*y37 )*pow(beta6,0.5);

for ( i = 1; i <= 7; ++i )
{
    ju2t[1][i] = ju2[i][1];
}

/*      GRADIENT OF THE PERFORMANCE CRITERION      */

j = 1;
for ( i = 1; i <= 7; ++i )
{
    dhdo[j] = 2.0*ju2t[1][i]*ddo1[i][1] + dhdo[j];
    dhdo[j+1] = 2.0*ju2t[1][i]*ddo2[i][1] + dhdo[j+1];
    dhdo[j+2] = 2.0*ju2t[1][i]*ddo3[i][1] + dhdo[j+2];
    dhdo[j+3] = 2.0*ju2t[1][i]*ddo4[i][1] + dhdo[j+3];
    dhdo[j+4] = 2.0*ju2t[1][i]*ddo5[i][1] + dhdo[j+4];
    dhdo[j+5] = 2.0*ju2t[1][i]*ddo6[i][1] + dhdo[j+5];
    dhdo[j+6] = 2.0*ju2t[1][i]*ddo7[i][1] + dhdo[j+6];
}

    gH->one = dhdo[1];
    gH->two = dhdo[2];
    gH->three = dhdo[3];
    gH->four = dhdo[4];
    gH->five = dhdo[5];
    gH->six = dhdo[6];
    gH->seven = dhdo[7];

break;

case 5:

*kKind = 1;      /* MAXIMIZING PERFORMANCE CRITERION */

/*****
      MANIPULABILITY MEASURE CRITERION
      (using the upper part of the wrist Jacobian)
*****/

n = 7;
m = 7;

/*      INITIALIZING VARIABLES      */

for ( i = 1; i <= 7; ++i )
{
    for ( ii = 1; ii <= 7; ++ii )
    {
        jac[i][ii] = 0.0;
    }
}

/*      THE UPPER 3x7 PART OF THE WRIST JACOBIAN
      REFERENCED TO THE BASE FRAME      */

```

```

jac[1][1] = -psy;
jac[1][2] = c1*psz;
jac[1][3] = s1*s2*(psz+a2*s2) - c2*(psy-a2*c2*s1);
jac[1][4] = (s1*c2*s3-c1*c3)*(psz+a2*s2) + (s2*s3)*(psy-a2*c2*s1);

jac[2][1] = psx;
jac[2][2] = s1*psz;
jac[2][3] = -c1*s2*(psz+a2*s2) + c2*(psx-a2*c2*c1);
jac[2][4] = -(c1*c2*s3+s1*c3)*(psz+a2*s2) - (s2*s3)*(psx-a2*c2*c1);

jac[3][2] = -s1*psy-c1*psx;
jac[3][3] = c1*s2*(psy-a2*c2*s1) - s1*s2*(psx-a2*c2*c1);
jac[3][4] = (c1*c2*s3+s1*c3)*(psy-a2*c2*s1)
            -(psx-a2*c2*c1)*(s1*c2*s3-c1*c3);

for ( i = 1; i <= 7; ++i )
{
    for ( ii = 1; ii <= 7; ++ii )
    {
        g[i][ii] = 0.0;
    }
}

for ( i = 1; i <= n; ++i )
{
    for ( ii = 1; ii <= n; ++ii )
    {
        g[i][ii] = jac[i][ii];
    }
}

for ( i = 1; i <= 9; ++i )
{
    for ( ii = 1; ii <= 8; ++ii )
    {
        for ( j = 1; j <= 9; ++j )
        {
            jacp[i][ii][j] = 0.0;
        }
    }
}

/*      PARTIAL DIFFERENTIAL OF EVERY ENTRY IN THE
      WRIST JACOBIAN w.r.t. THETA      */

jacp[1][1][1] = -(c1*c2*c3*a4*c4 - s1*s3*a4*c4 +
                  c1*s2*a4*s4 + a2*c2*c1);
jacp[1][1][2] = (s1*s2*c3*a4*c4 - s1*c2*a4*s4 + s1*a2*s2);
jacp[1][1][3] = (-c1*c3*a4*c4 + s1*c2*s3*a4*c4);
jacp[1][1][4] = (-s2*s1*a4*c4 + c1*s3*a4*s4 + s1*c3*c2*a4*s4);
jacp[2][1][1] = -(-s1*s2*c3*a4*c4 + s1*c2*a4*s4 - a2*s2*s1);
jacp[2][1][2] = (-c1*c2*c3*a4*c4 - c1*s2*a4*s4 - c1*a2*c2);
jacp[2][1][3] = (c1*s2*s3*a4*c4);
jacp[2][1][4] = (c2*c1*a4*c4 + c1*c3*s2*a4*s4);
jacp[3][1][1] = -(-s1*c2*s3*a4*c4 + c1*c3*a4*c4);
jacp[3][1][2] = (c1*s2*s3*a4*c4);
jacp[3][1][3] = (s1*s3*a4*c4 - c1*c2*c3*a4*c4);
jacp[3][1][4] = (s1*c3*a4*s4 + c1*s3*c2*a4*s4);
jacp[4][1][1] = -(-s1*c2*c3*a4*s4 - c1*s3*a4*s4 + s1*s2*a4*c4);

```

```

jacp[4][1][2] = (c1*s2*c3*a4*s4 + c1*c2*a4*c4);
jacp[4][1][3] = (s1*c3*a4*s4 + c1*c2*s3*a4*s4);
jacp[4][1][4] = (-s2*c1*a4*s4 + s1*s3*a4*c4 - c1*c3*c2*a4*c4);
jacp[1][2][1] = (-s1*c2*c3*a4*c4 - c1*s3*a4*c4 -
                  s1*s2*a4*s4 - a2*c2*s1);
jacp[1][2][2] = (-c1*s2*c3*a4*c4 + c1*c2*a4*s4 - c1*a2*s2);
jacp[1][2][3] = (-s1*c3*a4*c4 - c1*c2*s3*a4*c4);
jacp[1][2][4] = (c1*s2*a4*c4 + s1*s3*a4*s4 - c1*c3*c2*a4*s4);
jacp[2][2][1] = (-c1*s2*c3*a4*c4 + c1*c2*a4*s4 - a2*s2*c1);
jacp[2][2][2] = (-s1*c2*c3*a4*c4 - s1*s2*a4*s4 - s1*a2*c2);
jacp[2][2][3] = (s1*s2*s3*a4*c4);
jacp[2][2][4] = (s1*c2*a4*c4 + s1*c3*s2*a4*s4);
jacp[3][2][1] = (-c1*c2*s3*a4*c4 - s1*c3*a4*c4);
jacp[3][2][2] = (s1*s2*s3*a4*c4);
jacp[3][2][3] = (-c1*s3*a4*c4 - s1*c2*c3*a4*c4);
jacp[3][2][4] = (-c1*c3*a4*s4 + s1*s3*c2*a4*s4);
jacp[4][2][1] = (-c1*c2*c3*a4*s4 + s1*s3*a4*s4 + c1*s2*a4*c4);
jacp[4][2][2] = (s1*s2*c3*a4*s4 + s1*c2*a4*c4);
jacp[4][2][3] = (-c1*c3*a4*s4 + s1*c2*s3*a4*s4);
jacp[4][2][4] = (-s1*s2*a4*s4 - c1*s3*a4*c4 - s1*c3*c2*a4*c4);
jacp[1][3][1] = 0.0;
jacp[1][3][2] = (-c2*c3*a4*c4 - s2*a4*s4 - a2*c2);
jacp[1][3][3] = 0.0;
jacp[1][3][4] = 0.0;
jacp[2][3][1] = 0.0;
jacp[2][3][2] = (s2*c3*a4*c4 - c2*a4*s4 + a2*s2);
jacp[2][3][3] = (c2*s3*a4*c4);
jacp[2][3][4] = (-s2*a4*c4 + c3*c2*a4*s4);
jacp[3][3][1] = 0.0;
jacp[3][3][2] = (c2*s3*a4*c4);
jacp[3][3][3] = (s2*c3*a4*c4);
jacp[3][3][4] = (-s3*s2*a4*s4);
jacp[4][3][1] = 0.0;
jacp[4][3][2] = (c2*c3*a4*s4 - s2*a4*c4);
jacp[4][3][3] = (-s2*s3*a4*s4);
jacp[4][3][4] = (-c2*a4*s4 + c3*s2*a4*c4);

/*      CALLING THE PSUEDO-INVERSE ROUTINE      */

psuedo_inv (g,empty,select,w);
/* g is now the inverse of JJT */

/*      INITIALIZING AND RECALCULATING THE ENTRIES
      OF THE JACOBIAN      */

for ( i = 1; i <= 7; ++i )
{
    for ( ii = 1; ii <= 7; ++ii )
    {
        jac[i][ii] = 0.0;
    }
}

jac[1][1] = -psy;
jac[1][2] = c1*psz;
jac[1][3] = s1*s2*(psz+a2*s2) - c2*(psy-a2*c2*s1);
jac[1][4] = (s1*c2*s3-c1*c3)*(psz+a2*s2) + (s2*s3)*(psy-a2*c2*s1);

jac[2][1] = psx;
jac[2][2] = s1*psz;
jac[2][3] = -c1*s2*(psz+a2*s2) + c2*(psx-a2*c2*c1);
jac[2][4] = -(c1*c2*s3+s1*c3)*(psz+a2*s2) - (s2*s3)*(psx-a2*c2*c1);

```

```

jac[3][2] = -s1*psy-c1*psx;
jac[3][3] = c1*s2*(psy-a2*c2*s1) - s1*s2*(psx-a2*c2*c1);
jac[3][4] = (c1*c2*s3+s1*c3)*(psy-a2*c2*s1)
            -(psx-a2*c2*c1)*(s1*c2*s3-c1*c3);

/*      THE DETERMINANT OF:
          (JACOBIAN)*(TRASPOSE OF THE JACOBIAN) */

det = 1.0;
for ( i = 1; i <= 7; ++i )
{
    if ( non[i] != 0.0 )
    {
        det = det*non[i];
    }
}

if ( det<0.0 )
{
    det = - det;
}

/*      THE GRADIENT OF THE MANIPULABILITY MEASURE
          CRITERION */

for ( i = 0; i <= 7; ++i )
{
    s[i] = 0.0;
    ss[i] = 0.0;
}

for ( l = 1; l <= 7 ; ++l )
{
    for ( i = 1; i <= 6; ++i )
    {
        for ( j = 1; j <= 6; ++j )
        {
            for ( iii = 1; iii <= 7; ++iii )
            {
                s[l] = s[l] + jacp[l][i][iii]*jac[j][iii]
                    + jacp[l][j][iii]*jac[i][iii];
            }
            ss[l] = ss[l] + g[i][j]*s[l];
            s[l] = 0.0;
        }
    }
}

zeta[l] = 0.5*(sqrt(det))*ss[l];
}

gH->one = zeta[1];
gH->two = zeta[2];
gH->three = zeta[3];
gH->four = zeta[4];
gH->five = zeta[5];
gH->six = zeta[6];
gH->seven = zeta[7];

```

break;

case 6:

```

/*****
      MANIPULATOR VELOCITY RATIO IN THE DIRECTION
      CORRESPONDING TO THE MINIMUM SINGULAR
      VALUE (LINEAR MOTION CASE)
*****/

```

```

/*      INITIALIZING VARIABLES      */

```

```

for ( i = 0; i <= 3; ++i )
{
    for ( ii = 0; ii <= 3; ++ii )
    {
        jlti[i][ii] = 0.0;
    }
}

```

```

for ( i = 0; i <= 7; ++i )
{
    for ( ii = 0; ii <= 1; ++ii )
    {
        ddo1[i][ii] = 0.0;
        ddo2[i][ii] = 0.0;
        ddo3[i][ii] = 0.0;
        ddo4[i][ii] = 0.0;
        ddo5[i][ii] = 0.0;
        ddo6[i][ii] = 0.0;
        ddo7[i][ii] = 0.0;
        jzl[i][ii] = 0.0;
    }
}

```

```

for ( i = 0; i <= 1; ++i )
{
    for ( ii = 0; ii <= 7; ++ii )
    {
        jzlt[i][ii] = 0.0;
    }
}

```

```

for ( i = 0; i <= 7; ++i )
{
    dhdo[i] = 0.0;
    z[i] = 0.0;
    zl[i] = 0.0;
}

```

```

for ( i = 0; i <= 7; ++i )
{
    for ( ii = 0; ii <= 3; ++ii )
    {
        jlt[i][ii] = 0.0;
    }
}

```

```

for ( i = 0; i <= 7; ++i )
{
    for ( ii = 0; ii <= 7; ++ii )
    {
        jl[i][ii] = 0.0;
        um[i][ii] = 0.0;
        ul[i][ii] = 0.0;
    }
}

```

```

/*      WEIGHTING FACTORS      */

g1 = pow(gama1,0.5);
g2 = pow(gama2,0.5);
g3 = pow(gama3,0.5);
e1 = pow(eta1,-0.5);
e2 = pow(eta2,-0.5);
e3 = pow(eta3,-0.5);
e4 = pow(eta4,-0.5);

/*      THE TRANSFORMED WRIST JACOBIAN
      (UPPER HALF-PART)      */

j1[1][1] = -g1*psy*e1;
j1[1][2] = g1*c1*psz*e2;
j1[1][3] = g1*( s1*s2*(psz+a2*s2) - c2*(psy-a2*c2*s1) )*e3;
j1[1][4] = g1*( (s1*c2*s3-c1*c3)*(psz+a2*s2) + (s2*s3)*
      (psy-a2*c2*s1) )*e4;
j1[2][1] = g2*psx*e1;
j1[2][2] = g2*s1*psz*e2;
j1[2][3] = g2*( -c1*s2*(psz+a2*s2) + c2*(psx-a2*c2*c1) )*e3;
j1[2][4] = g2*( -(c1*c2*s3+s1*c3)*(psz+a2*s2) - (s2*s3)*
      (psx-a2*c2*c1) )*e4;
j1[3][2] = g3*(-s1*psy-c1*psx)*e2;
j1[3][3] = g3*( c1*s2*(psy-a2*c2*s1)-s1*s2*(psx-a2*c2*c1) )*e3;
j1[3][4] = g3*( (c1*c2*s3+s1*c3)*(psy-a2*c2*s1) -
      (psx-a2*c2*c1)*(s1*c2*s3-c1*c3) )*e4;

for ( i = 1; i <= 7; ++i )
{
    for ( ii = 1; ii <= 3; ++ii )
    {
        jlt[i][ii] = j1[ii][i];
    }
}

/*      CALLING THE PSUEDO-INVERSE ROUTINE      */

psuedo_inv ( j1,um,select,w );
/* j1 is now a 3x3 inverse matrix of JJT */

/*      THE INVERSE OF JJT      */

for ( i = 1; i <= 3; ++i )
{
    for ( ii = 1; ii <= 3; ++ii )
    {
        jjlti[i][ii] = j1[i][ii];
    }
}

/*      z IS THE PRODUCT OF THE INVERSE OF JJT BY
      THE VECTOR IN THE DIRECTION OF THE MINIMUM
      SINGULAR VALUE      */

for ( i = 1; i <= 3; ++i )

```

```

{
    for ( iii = 1; iii<= 3; ++iii )
    {
        z[i] = jjlti[i][iii]*um[iii][select[0]] + z[i];
    }
}

/*      MVR IN THE DIRECTION CORRESPONDING TO THE          */
/*      MINIMUM SINGULAR VALUE                                */

rv1 = 1.0/(sqrt(um[1][select[0]]*z[1] + um[2][select[0]]*z[2]
+ um[3][select[0]]*z[3]));

fprintf(singularity,"%d %f\n",iteration, rv1);

/*      ENTRIES OF THE VECTOR THAT CONSISTS OF THE PRODUCT
OF:
      (TRANPOSE OF THE JACOBIAN)*(Z-VECTOR)      */

i1 = psz + a2*s2;
i2 = psy - a2*c2*s1;
i3 = psx - a2*c2*c1;

jz1[1][1] = -g1*psy*e1*z[1] + g2*psx*e1*z[2];
jz1[2][1] = g1*c1*psz*e1*z[1] + g2*s1*psz*e2*z[2]
+ g3*(-s1*psy-c1*psx)*e2*z[3];
jz1[3][1] = g1*( s1*s2*i1 - c2*i2 )*e3*z[1]
+ g2*( -c1*s2*i1 + c2*i3 )*e3*z[2]
+ g3*( c1*s2*i2 - s1*s2*i3 )*e3*z[3];
jz1[4][1] = g1*( (s1*c2*s3-c1*c3)*i1 + (s2*s3)*i2 )*e4*z[1]
+ g2*( -(c1*c2*s3+s1*c3)*i1 - (s2*s3)*i3 )*e4*z[2]
+ g3*( (c1*c2*s3+s1*c3)*i2 - i3*(s1*c2*s3-c1*c3))*e4*z[3];
jz1[5][1] = 0.0;
jz1[6][1] = 0.0;
jz1[7][1] = 0.0;

/*      PARTIAL DIFFERENTIAL OF jz1 w.r.t. THETA          */

ddo1[1][1] = -g1*(c1*c2*c3*a4*c4 - s1*s3*a4*c4 +
c1*s2*a4*s4 + a2*c2*c1)*e1*z[1] + g2*(-s1*c2*c3*a4*c4 -
c1*s3*a4*c4 - s1*s2*a4*s4 - a2*c2*s1)*e1*z[2];

ddo1[2][1] = g1*(s1*s2*c3*a4*c4 - s1*c2*a4*s4 +
s1*a2*s2)*e2*z[1] + g2*(-c1*s2*c3*a4*c4 + c1*c2*a4*s4 -
c1*a2*s2)*e2*z[2];

ddo1[3][1] = g1*(-c1*c3*a4*c4 + s1*c2*s3*a4*c4)*e3*z[1] +
g2*(-s1*c3*a4*c4 - c1*c2*s3*a4*c4)*e3*z[2];

ddo1[4][1] = g1*(-s2*s1*a4*c4 + c1*s3*a4*s4 +
s1*c3*c2*a4*s4)*e4*z[1] + g2*(c1*s2*a4*c4 + s1*s3*a4*s4 -
c1*c3*c2*a4*s4)*e4*z[2];

ddo2[1][1] = -g1*(-s1*s2*c3*a4*c4 + s1*c2*a4*s4 -
a2*s2*s1)*e1*z[1] + g2*(-c1*s2*c3*a4*c4 + c1*c2*a4*s4 -
a2*s2*c1)*e1*z[2];

ddo2[2][1] = g1*(-c1*c2*c3*a4*c4 - c1*s2*a4*s4 -
c1*a2*c2)*e2*z[1] + g2*(-s1*c2*c3*a4*c4 - s1*s2*a4*s4 -
s1*a2*c2)*e2*z[2] + g3*(s2*c3*a4*c4 - c2*a4*s4 + a2*s2)*e2*z[3];

```

```

ddo2[3][1] = g1*(c1*s2*s3*a4*c4)*e3*z[1] +
g2*(s1*s2*s3*a4*c4)*e3*z[2] + g3*(c2*s3*a4*c4)*e3*z[3];

ddo2[4][1] = g1*(c2*c1*a4*c4 + c1*c3*s2*a4*s4)*e4*z[1] +
g2*(s1*c2*a4*c4 + s1*c3*s2*a4*s4)*e4*z[2] + g3*(-s2*a4*c4 +
c3*c2*a4*s4)*e4*z[3];

ddo3[1][1] = -g1*(-s1*c2*s3*a4*c4 + c1*c3*a4*c4)*e1*z[1] +
g2*(-c1*c2*s3*a4*c4 - s1*c3*a4*c4)*e1*z[2];

ddo3[2][1] = g1*(c1*s2*s3*a4*c4)*e2*z[1] +
g2*(s1*s2*s3*a4*c4)*e2*z[2] + g3*(c2*s3*a4*c4)*e2*z[3];

ddo3[3][1] = g1*(s1*s3*a4*c4 - c1*c2*c3*a4*c4)*e3*z[1] +
g2*(-c1*s3*a4*c4 - s1*c2*c3*a4*c4)*e3*z[2] +
g3*(s2*c3*a4*c4)*e3*z[3];

ddo3[4][1] = g1*(s1*c3*a4*s4 + c1*s3*c2*a4*s4)*e4*z[1] +
g2*(-c1*c3*a4*s4 + s1*s3*c2*a4*s4)*e4*z[2] +
g3*(-s3*s2*a4*s4)*e4*z[3];

ddo4[1][1] = -g1*(-s1*c2*c3*a4*s4 - c1*s3*a4*s4 +
s1*s2*a4*c4)*e1*z[1] + g2*(-c1*c2*c3*a4*s4 + s1*s3*a4*s4 +
c1*s2*a4*c4)*e1*z[2];

ddo4[2][1] = g1*(c1*s2*c3*a4*s4 + c1*c2*a4*c4)*e2*z[1] +
g2*(s1*s2*c3*a4*s4 + s1*c2*a4*c4)*e2*z[2] + g3*(c2*c3*a4*s4 -
s2*a4*c4)*e2*z[3];

ddo4[3][1] = g1*(s1*c3*a4*s4 + c1*c2*s3*a4*s4)*e3*z[1] +
g2*(-c1*c3*a4*s4 + s1*c2*s3*a4*s4)*e3*z[2] +
g3*(-s2*s3*a4*s4)*e3*z[3];

ddo4[4][1] = g1*(-s2*c1*a4*s4 + s1*s3*a4*c4 -
c1*c3*c2*a4*c4)*e4*z[1] + g2*(-s1*s2*a4*s4 - c1*s3*a4*c4 -
s1*c3*c2*a4*c4)*e4*z[2] + g3*(-c2*a4*s4 + c3*s2*a4*c4)*e4*z[3];

for ( i = 1; i <= 7; ++i )
{
    jzlt[1][i] = jz1[i][1];
}

/*      THE GRADIENT OF THE PERFORMANCE CRITERION      */

j = 1;
for ( i = 1; i <= 7; ++i )
{
    dhdo[j] = -2.0*jzlt[1][i]*ddo1[i][1] + dhdo[j];
    dhdo[j+1] = -2.0*jzlt[1][i]*ddo2[i][1] + dhdo[j+1];
    dhdo[j+2] = -2.0*jzlt[1][i]*ddo3[i][1] + dhdo[j+2];
    dhdo[j+3] = -2.0*jzlt[1][i]*ddo4[i][1] + dhdo[j+3];
    dhdo[j+4] = -2.0*jzlt[1][i]*ddo5[i][1] + dhdo[j+4];
    dhdo[j+5] = -2.0*jzlt[1][i]*ddo6[i][1] + dhdo[j+5];
    dhdo[j+6] = -2.0*jzlt[1][i]*ddo7[i][1] + dhdo[j+6];
}

gH->one = dhdo[1];
gH->two = dhdo[2];
gH->three = dhdo[3];
gH->four = dhdo[4];
gH->five = dhdo[5];
gH->six = dhdo[6];

```

```

        gH->seven = dhdo[7];

break;

case 7:

/*****
      MANIPULATOR VELOCITY RATIO IN THE DIRECTION
      CORRESPONDING TO THE MINIMUM SINGULAR VALUE
      (ANGULAR MOTION CASE)
*****/

/*      INITIALIZING VARIABLES      */

for ( i = 0; i <= 3; ++i )
{
    for ( ii = 0; ii <= 3; ++ii )
    {
        jjati[i][ii] = 0.0;
    }
}

for ( i = 0; i <= 3; ++i )
{
    for ( ii = 0; ii <= 1; ++ii )
    {
        ua[i][ii] = 0.0;
    }
}

for ( i = 0; i <= 7; ++i )
{
    for ( ii = 0; ii <= 1; ++ii )
    {
        ddo1[i][ii] = 0.0;
        ddo2[i][ii] = 0.0;
        ddo3[i][ii] = 0.0;
        ddo4[i][ii] = 0.0;
        ddo5[i][ii] = 0.0;
        ddo6[i][ii] = 0.0;
        ddo7[i][ii] = 0.0;
        jza[i][ii] = 0.0;
    }
}

for ( i = 0; i <= 1; ++i )
{
    for ( ii = 0; ii <= 7; ++ii )
    {
        jzat[i][ii] = 0.0;
    }
}

for ( i = 0; i <= 7; ++i )
{
    dhdo[i] = 0.0;
    z[i] = 0.0;
}

for ( i = 0; i <= 7; ++i )
{

```

```

        for ( ii = 0; ii <= 3; ++ii )
        {
            jat[i][ii] = 0.0;
        }
    }

    for ( i = 0; i <= 7; ++i )
    {
        for ( ii = 0; ii <= 7; ++ii )
        {
            ja[i][ii] = 0.0;
            um[i][ii] = 0.0;
        }
    }

    /*      WEIGHTING FACTORS      */

    g6 = pow(gama6,0.5);
    g5 = pow(gama5,0.5);
    g4 = pow(gama4,0.5);
    e1 = pow(eta1,-0.5);
    e2 = pow(eta2,-0.5);
    e3 = pow(eta3,-0.5);
    e4 = pow(eta4,-0.5);
    e5 = pow(eta5,-0.5);
    e6 = pow(eta6,-0.5);
    e7 = pow(eta7,-0.5);

    /*      THE WRIST JACOBIAN (3x7 LOWER HALF PART)      */

    ja[1][1] = 0.0;
    ja[1][2] = -g4*s1*e2;
    ja[1][3] = g4*c1*s2*e3;
    ja[1][4] = g4*( c1*c2*s3+s1*c3 )*e4;
    ja[1][5] = -g4*( (-s6*c7)*x17 + (s6*s7)*y17 + c6*z17 )*e5;
    ja[1][6] = g4*( s7*x17 + c7*y17 )*e6;
    ja[1][7] = g4*( (px-psx)/d7 )*e7;
    ja[2][1] = 0.0;
    ja[2][2] = g5*c1*e2;
    ja[2][3] = g5*s1*s2*e3;
    ja[2][4] = g5*( s1*c2*s3-c1*c3 )*e4;
    ja[2][5] = -g5*( (-s6*c7)*x27 + (s6*s7)*y27 + c6*z27 )*e5;
    ja[2][6] = g5*( s7*x27 + c7*y27 )*e6;
    ja[2][7] = g5*( (py-psy)/d7 )*e7;
    ja[3][1] = g6*e1;
    ja[3][2] = 0.0;
    ja[3][3] = g6*c2*e3;
    ja[3][4] = g6*( -s2*s3 )*e4;
    ja[3][5] = -g6*( (-s6*c7)*x37 + (s6*s7)*y37 + c6*z37 )*e5;
    ja[3][6] = g6*( s7*x37 + c7*y37 )*e6;
    ja[3][7] = g6*( (pz-psz)/d7 )*e7;

    for ( i = 1; i <= 7; ++i )
    {
        for ( ii = 1; ii <= 3; ++ii )
        {
            jat[i][ii] = ja[ii][i];
        }
    }

    /*      CALLING THE PSUEDO-INVERSE ROUTINE      */

    psuedo_inv ( ja,um,select,w );

```

```

/* ja is now a 3x3 inverse matrix of JJT */

/*      THE INVERSE OF JJT      */
for ( i = 1; i <= 3; ++i )
{
    for ( ii = 1; ii <= 3; ++ii )
    {
        jjati[i][ii] = ja[i][ii];
    }
}

/*      z IS THE PRODUCT OF THE INVERSE OF JJT BY
THE VECTOR IN THE DIRECTION OF THE MINIMUM
SINGULAR VALUE      */
for ( i = 1; i <= 3; ++i )
{
    for ( iii = 1; iii <= 3; ++iii )
    {
        z[i+3] = jjati[i][iii]*um[iii][select[0]] + z[i+3];
    }
}

/*      MVR IN THE DIRECTION CORRESPONDING TO THE MINIMUM
SINGULAR VALUE      */
rva = 1.0/(sqrt(um[1][select[0]]*z[4] + um[2][select[0]]*z[5]
+ um[3][select[0]]*z[6]));

fprintf(singularity, "%f\n", rva);

/*      ENTRIES OF THE VECTOR CONSISTING OF THE PRODUCT
OF:
(TRANSPOSE OF THE JACOBIAN)*(Z-VECTOR) */

jza[1][1] = g6*e1*z[6];
jza[2][1] = -g4*s1*e2*z[4] + g5*c1*e2*z[5];
jza[3][1] = g4*c1*s2*e3*z[4] + g5*s1*s2*e3*z[5] + g6*c2*e3*z[6];
jza[4][1] = g4*(c1*c2*s3+s1*c3)*e4*z[4]+g5*(s1*c2*s3-c1*c3)*e4*z[5]
+ g6*(-s2*s3)*e4*z[6];
jza[5][1] = -g4*(-s6*c7*x17 + s6*s7*y17 + c6*z17)*e5*z[4]
- g5*(-s6*c7*x27 + s6*s7*y27 + c6*z27)*e5*z[5]
- g6*(-s6*c7*x37 + s6*s7*y37 + c6*z37)*e5*z[6];
jza[6][1] = g4*(s7*x17+c7*y17)*e6*z[4] + g5*(s7*x27+c7*y27)*e6*z[5]
+ g6*(s7*x37 + c7*y37)*e6*z[6];
jza[7][1] = g4*( (px-psx)/d7 )*e7*z[4] + g5*( (py-psy)/d7 )*e6*z[5]
+ g6*( (pz-psz)/d7 )*e7*z[6];

/*      PARTIAL DIFFERENTIAL OF jza w.r.t. THETA      */
ddol[2][1] = -g4*c1*e2*z[4] - g5*s1*e2*z[5];
ddol[3][1] = -g4*s1*s2*e3*z[4] + g5*c1*s2*e3*z[5];
ddol[4][1] = g4*(-s1*c2*s3+c1*c3)*e4*z[4]
+ g5*(c1*c2*s3+s1*c3)*e4*z[5];
ddo2[3][1] = g4*c1*c2*e3*z[4] + g5*s1*c2*e3*z[5] - g6*s2*e3*z[6];
ddo2[4][1] = g4*(-c1*s2*s3)*e4*z[4] + g5*(-s1*s2*s3)*e4*z[5]
+ g6*(-c2*s3)*e4*z[6];
ddo3[4][1] = g4*(c1*c2*c3-s1*s3)*e4*z[4]

```

```

        + g5*(s1*c2*c3+c1*s3)*e4*z[5]
        + g6*(-s2*c3)*e4*z[6];

ddo6[5][1] = -g4*( -c6*c7*x17 + c6*s7*y17 - s6*z17 )*e5*z[4]
             - g5*( -c6*c7*x27 + c6*s7*y27 - s6*z27 )*e5*z[5]
             - g6*( -c6*c7*x37 + c6*s7*y37 - s6*z37 )*e5*z[6];

ddo7[5][1] = -g4*( s6*s7*x17 + s6*c7*y17 )*e5*z[4]
             - g5*( s6*s7*x27 + s6*c7*y27 )*e5*z[5]
             - g6*( s6*s7*x37 + s6*c7*y37 )*e5*z[6];
ddo7[6][1] = g4*(c7*x17-s7*y17)*e6*z[4]+g5*(c7*x27-s7*y27)*e6*z[5]
             + g6*(c7*x37 - s7*y37)*e6*z[6];

for ( i = 1; i <= 7; ++i )
{
    jzat[1][i] = jza[i][1];
}

/*      THE GRADIENT OF THE PERFORMANCE CRITERION      */

j = 1;
for ( i = 1; i <= 7; ++i )
{
    dhdo[j] = -2.0*jzat[1][i]*ddo1[i][1] + dhdo[j];
    dhdo[j+1] = -2.0*jzat[1][i]*ddo2[i][1] + dhdo[j+1];
    dhdo[j+2] = -2.0*jzat[1][i]*ddo3[i][1] + dhdo[j+2];
    dhdo[j+3] = -2.0*jzat[1][i]*ddo4[i][1] + dhdo[j+3];
    dhdo[j+4] = -2.0*jzat[1][i]*ddo5[i][1] + dhdo[j+4];
    dhdo[j+5] = -2.0*jzat[1][i]*ddo6[i][1] + dhdo[j+5];
    dhdo[j+6] = -2.0*jzat[1][i]*ddo7[i][1] + dhdo[j+6];
}

gH->one = dhdo[1];
gH->two = dhdo[2];
gH->three = dhdo[3];
gH->four = dhdo[4];
gH->five = dhdo[5];
gH->six = dhdo[6];
gH->seven = dhdo[7];

break;
}
}

/*****
THIS ROUTINE DECOMPOSES THE JACOBIAN USING THE
SVD DECOMPOSITION
GIVEN A MATRIX a[1...m][1...n], THIS ROUTINE
COMPUTES ITS SINGULAR VALUE DECOMPOSITION,
J = U.W.V(T) . THE MATRIX U REPLACES a ON OUTPUT
AND THE DIAGONAL MATRIX OF SINGULAR VALUES W IS
OUTPUT AS A VECTOR W[1...n]. THE MATRIX V, NOT
THE TRASPOSE OF V, IS OUTPUT AS V[1...n][1...n].
*****/

psuedo_inv ( a,direction,selection,w )
double a[8][8],direction[8][8],w[8];
int selection[1];

```

```

{
int flag,i,its,j,jj,k,l,nm;
double c,f,h,s,x,y,z;
double anorm=0.0,g=0.0,scale=0.0;
double rv1[8];
double w[8],v[8][8];
int m,n;
double p[8][8],inv[8][8],wi[8][8],sig[8][8],atransp[8][8];
double psig[8][8],sigt[8][8],pinv[8][8],prod_inv[8][8],iden[4][4];
int ii,iii;
double att[7][7];
double smallest;
double det;
int selection1[1];
double largest;

m=7;
n=7;

/*      INITIALIZING VARIABLES      */

for ( i = 1; i <= n; ++i )
{
    for ( ii = 1; ii <= n; ++ii )
    {
        p[i][ii] = 0.0;
        v[i][ii] = 0.0;
        atransp[i][ii] = 0.0;
        inv[i][ii] = 0.0;
        prod_inv[i][ii] = 0.0;
        pinv[i][ii] = 0.0;
        direction[i][ii] = 0.0;
    }
}

/*      DECOMPOSING THE mxn MATRIX      */

for ( i = 1; i <= n; i++ ) {
    l = i+1;
    rv1[i] = scale*g;
    g=s=scale=0.0;
    if ( i <= m ) {
        for (k=i;k<=m;k++) scale += fabs(a[k][i]);
        if (scale) {
            for (k=i;k<=m;k++) {
                a[k][i] /= scale;
                s += a[k][i]*a[k][i];
            }
            f=a[i][i];
            g = -SIGN(sqrt(s),f);
            h=f*g-s;
            a[i][i]=f-g;
            if (i != n) {
                for (j=l;j<=n;j++) {
                    for(s=0.0,k=i;k<=m;k++) s +=a[k][i]*a[k][j];
                    f=s/h;
                    for (k=i;k<=m;k++) a[k][j] += f*a[k][i];
                }
            }
            for (k=i;k<=m;k++) a[k][i] *= scale;
        }
    }
    w[i] = scale*g;
    g=s=scale=0.0;
    if (i <= m && i != n) {
        for (k=l;k<=n;k++) scale += fabs(a[i][k]);
    }
}

```

```

        if (scale) {
            for (k=1;k<=n;k++) {
                a[i][k] /= scale;
                s += a[i][k]*a[i][k];
            }
            f=a[i][1];
            g = -SIGN(sqrt(s),f);
            h=f*g-s;
            a[i][1]=f-g;
            for (k=1;k<=n;k++) rvl[k]=a[i][k]/h;
            if (i != m) {
                for (j=1;j<=m;j++) {
                    for (s=0.0,k=1;k<=n;k++) s +=a[j][k]*a[i][k];
                    for (k=1;k<=n;k++) a[j][k] += s*rvl[k];
                }
            }
            for (k=1;k<=n;k++) a[i][k] *= scale;
        }
    }
    anorm=MAX(anorm, (fabs(w[i])+fabs(rvl[i])));
}

for (i=n;i>=1;i--) {
    if (i < n) {
        if (g) {
            for (j=1;j<=n;j++)
                v[j][i]=(a[i][j]/a[i][1])/g;
            for (j=1;j<=n;j++) {
                for (s=0.0,k=1;k<=n;k++) s += a[i][k]*v[k][j];
                for (k=1;k<=n;k++) v[k][j] += s*v[k][i];
            }
            for (j=1;j<=n;j++) v[i][j]=v[j][i]=0.0;
        }
        v[i][i]=1.0;
        g=rvl[i];
        l=i;
    }
    for (i=n;i>=1;i--) {
        l=i+1;
        g=w[i];
        if (i < n)
            for (j=1;j<=n;j++) a[i][j]=0.0;
        if (g) {
            g=1.0/g;
            if (i != n) {
                for (j=1;j<=n;j++) {
                    for (s=0.0,k=1;k<=m;k++) s += a[k][i]*a[k][j];
                    f=(s/a[i][i])*g;
                    for (k=i;k<=m;k++) a[k][j] += f*a[k][i];
                }
            }
            for (j=i;j<=m;j++) a[j][i] *= g;
        } else {
            for (j=i;j<=m;j++) a[j][i]=0.0;
        }
        ++a[i][i];
    }
}

for (k=n;k>=1;k--) {
    for (its=1;its<=30;its++) {
        flag=1;
        for (l=k;l>=1;l--) {
            nm=l-1;
            if (fabs(rvl[l])+anorm == anorm) {
                flag=0;

```

```

        break;
    }
    if (fabs(w[nm])+anorm == anorm) break;
}
if (flag) {
    c=0.0;
    s=1.0;
    for (i=1;i<=k;i++) {
        f=s*rvl[i];
        if (fabs(f)+anorm != anorm) {
            g=w[i];
            h=PYTHAG(f,g);
            w[i]=h;
            h=1.0/h;
            c=g*h;
            s=(-f*h);
            for (j=1;j<=m;j++) {
                y=a[j][nm];
                z=a[j][i];
                a[j][nm]=y*c+z*s;
                a[j][i]=z*c-y*s;
            }
        }
    }
}
z=w[k];
if (l == k) {
    if (z < 0.0) {
        w[k] = -z;
        for (j=1;j<=n;j++) v[j][k]=(-v[j][k]);
    }
    break;
}

x=w[l];
nm=k-1;
y=w[nm];
g=rvl[nm];
h=rvl[k];
f=((y-z)*(y+z)+(g-h)*(g+h))/(2.0*h*y);
g=PYTHAG(f,1.0);
f=((x-z)*(x+z)+h*((y/(f+SIGN(g,f)))-h))/x;

c=s=1.0;
for (j=1;j<=nm;j++) {
    i=j+1;
    g=rvl[i];
    y=w[i];
    h=s*g;
    g=c*g;
    z=PYTHAG(f,h);
    rvl[j]=z;
    c=f/z;
    s=h/z;
    f=x*c+g*s;
    g=g*c-x*s;
    h=y*s;
    y=y*c;
    for (jj=1;jj<=n;jj++) {
        x=v[jj][j];
        z=v[jj][i];
        v[jj][j]=x*c+z*s;
        v[jj][i]=z*c-x*s;
    }
    z=PYTHAG(f,h);
    w[j]=z;
}

```

```

        if (z) {
            z=1.0/z;
            c=f*z;
            s=h*z;
        }
        f=(c*g)+(s*y);
        x=(c*y)-(s*g);
        for (jj=1;jj<=m;jj++) {
            y=a[jj][j];
            z=a[jj][i];
            a[jj][j]=y*c+z*s;
            a[jj][i]=z*c-y*s;
        }
    }
    rv1[l]=0.0;
    rv1[k]=f;
    w[k]=x;
}

/*      THE DIRECTIONS CORRESPONDING TO THE SINGULAR
VALUES                                     */

for ( i = 1; i <= n; ++i )
{
    for ( j = 1; j <= n; ++j )
    {
        direction[i][j] = a[i][j];
    }
}

for ( i = 1; i <= m; ++i )
{
    for ( j = 1; j <= m; ++j )
    {
        atransp[i][j] = a[j][i];
    }
}

/*      THE SINGULAR VALUES OF THE DECOMPOSITION      */

for ( i = 1; i <= n; ++i )
{
    if ( fabs(w[i]) < 0.00001 )
    {
        w[i] = 0.0;
    }
}

for ( i = 1; i <= m; ++i )
{
    for ( ii = 1; ii <= n; ++ii )
    {
        if ( i == ii )
        {
            sig[i][ii] = w[i];
        }
        if ( i != ii )
        {
            sig[i][ii] = 0.0;
        }
    }
}
}

```

```

for ( i = 1; i <= n; ++i )
{
    if ( i == 1 && (fabs(sig[1][1]) > 0.00001) )
    {
        largest = sig[1][1];
        selectionl[0] = i;
        smallest = sig[1][1];
        selection[0] = i;
    }

    if ( i == 1 && (sig[1][1] == 0.0) )
    {
        largest = sig[2][2];
        selectionl[0] = i;
        smallest = sig[2][2];
        selection[0] = i+1;
    }

    if ((sig[i][i] < smallest) && (fabs(sig[i][i]) > 0.00001 ))
    {
        smallest = sig[i][i];
        selection[0] = i;
    }

    if ((sig[i][i] > largest) && (fabs(sig[i][i]) > 0.00001 ))
    {
        largest = sig[i][i];
        selectionl[0] = i;
    }
}

/*      THE DETERMINANT OF THE DECOMPOSED MATRIX      */
det = 1.0;
for ( i = 1; i <= 7; ++i )
{
    if ( sig[i][i] != 0.0 )
    {
        det = det*sig[i][i];
    }
}

/*      DETERMINING THE INVERSE OF THE JACOBIAN      */
for ( i = 1; i <= m; ++i )
{
    for ( ii = 1; ii <= m; ++ii )
    {
        if ( (i == ii) && (fabs(sig[i][ii]) > 0.0) )
        {
            wi[i][ii] = 1.0/sig[i][ii];
        }
        if ( (i == ii) && (fabs(sig[i][ii]) == 0.0) )
        {
            wi[i][ii] = 0.0;
        }
        if ( i != ii )
        {
            wi[i][ii] = 0.0;
        }
    }
}
for ( i = 1; i <= m; ++i )
{

```

```

        for ( j = 1; j <= m; ++j )
        {
            for ( iii = 1; iii <= m; ++iii )
            {
                p[i][j] = v[i][iii]*wi[iii][j]
                    +p[i][j];
            }
        }
    }
    for ( i = 1; i <= m; ++i )
    {
        for ( j = 1; j <= m; ++j )
        {
            for ( iii = 1; iii <= m; ++iii )
            {
                inv[i][j] = p[i][iii]*atransp[iii][j] + inv[i][j];
            }
        }
    }

    /*      THE INVERSE OF THE PRODUCT OF:
            (JACOBIAN)*(TRANSPOSE OF THE JACOBIAN)      */

    for ( i = 1; i <= n; ++i )
    {
        for ( ii = 1; ii <= m; ++ii )
        {
            for ( iii = 1; iii <= n; ++iii )
            {
                prod_inv[i][ii] = inv[iii][i]*inv[iii][ii]
                    + prod_inv[i][ii];
            }
        }
    }

    /* changing the name of a that has contents of u to contents
    of prod_inv */

    for ( i = 1; i <= n; ++i )
    {
        for ( ii = 1; ii <= n; ++ii )
        {
            a[i][ii] = 0.0;
        }
    }

    for ( i = 1; i <= n; ++i )
    {
        for ( ii = 1; ii <= n; ++ii )
        {
            a[i][ii] = prod_inv[i][ii];
        }
    }
}

```

VITA

Roy J. Issa was born in Beirut, Lebanon in 1966. He spent his childhood partly in his home town and partly in his grandparent's mountain resort overlooking the Mediterranean sea. He received his education in a private school in Beirut, from which he graduated with a high school degree in summer of 1984. Few days later, he travelled to the United States to attend Mohegan college in Norwich, Connecticut. After leaving the United States for a while in 1985, he immigrated back and settled in Knoxville, Tennessee. He entered the University of Tennessee at Knoxville where he graduated in June 1988 with a Bachelor of Science degree in Mechanical Engineering. Shortly after that, he decided to go to graduate school. In December of 1989, he received his Master of Science degree in Mechanical Engineering from the same university.

Roy is a member of the following societies: The American Society of Mechanical Engineers, Tau Beta Pi, Pi Tau Sigma, and Golden Key honor society.

Roy is blessed to have such a generous and loving parents who were supportive to him during the course of his undergraduate and graduate studies. Currently he is living with both his parents in Knoxville. He plans to get married in the near future and live in the north eastern part of the United States.