

Reducing Flexible Base Vibrations Using Kinematically Redundant Robots

by

Mark Lamon Hanson

B. S. Aeronautics and Astronautics, Massachusetts Institute of Technology, May 1988
M. S. Engineering, University of Michigan-Ann Arbor, December 1989

A Dissertation submitted to

The Faculty of

The School of Engineering and Applied Science
of The George Washington University in partial satisfaction
of the requirements for the degree of Doctor of Science

July 1995

Dissertation directed by

Dr. Robert H. Tolson
Professor of Engineering and Applied Science

This research was conducted at NASA LaRC.

ABSTRACT

Due to costs and mission requirements, future space robots will be used to take the place of humans. When these robots are attached to flexible bases such as the Space Shuttle Remote Manipulator System or the truss of the Space Station Freedom, an important issue concerning their performance is the dynamic coupling between the robot and the flexible base. When the robot moves, reaction forces and moments at the base of the manipulator excite flexible modes in the supporting structure that lead to end-effector tracking errors and performance degradation.

Since redundant robots have the ability to track a desired end-effector trajectory while also optimizing secondary criteria, kinematically redundant robots are investigated in the current work as one means of reducing or eliminating the unwanted effects of flexible vibrations. Using local resolved acceleration control, energy dissipation control laws for the manipulator redundancies are derived and are theoretically and numerically shown to be effective for reducing vibration. However, owing to inherent singularity problems in local optimizations, excessively high joint velocities can occur. To overcome this problem, a fuzzy logic supervisor was designed and implemented for adaptively controlling the kinematic redundancies. Compared to the non-fuzzy solutions, the fuzzy solution is better at avoiding singularities while still decreasing flexible vibrations.

ACKNOWLEDGEMENT

This research was conducted at the Joint Institute for the Advancement of Flight Sciences (JIAFS) at the NASA Langley Research Center under NASA grant NAG1-1353. Much thanks is given to the many groups at Langley who made this work possible. These groups include the administrative staff at GWU, the Office of Education, the library personnel, and last but not least the security guards who let me into the building after hours.

As no man is an island, this work could not have been completed without the assistance from other researchers. First and foremost, thanks is given to Dr. Robert Tolson whose guidance and inquisitiveness helped me to search for deeper understanding of this research, i.e. "why does this curve drop here?" Thanks is also given to Dr. Jer-Nan Juang and Dr. Vladimir Klein whose classes in Control and System Identification of Space Structures and System Identification in Aerospace Systems, respectively, were among the best classes I have ever taken and lit the quest for knowledge fire inside of me. Finally, thanks are given to Mr. J.C. White and my mom, Bertha M. Hanson for their support, encouragement, and love.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENT	iii
LIST OF FIGURES	vi
LIST OF TABLES	ix
NOMENCLATURE	x
1. INTRODUCTION	1
2. KINEMATICS	5
2.1 Fixed Base Kinematics	5
2.2 Moving Base Kinematics	13
3. DYNAMICS	17
3.1 Extension of Moving Base Kinematics	17
3.2 Flexible Base Model	21
3.3 Rigid Robot Model	23
3.4 Integrated Flexible Base, Rigid Robot Model	26
4. VIBRATION REDUCTION CONTROL LAWS	29
4.1 Control Problem Definition and Formulation	30
4.2 End-effector Acceleration Selection	33
4.3 Redundancy Control Methods	35
4.4 Control Algorithms	38
4.4 Performance and Stability Analysis	48
4.5 Summary	58
5. TEST MODEL	60
6. FIXED WEIGHT RESULTS	63
6.1 Vibration Reduction Performance Analysis	67
6.2 Sensitivity Analysis	74
6.3 Discussion	85
7. USING FUZZY LOGIC TO COMPUTE THE WEIGHTING FACTOR	89
7.1 Optimizing Multi-Criteria in Kinematically Redundant Robots	90
7.2 Design of a Fuzzy Logic Supervisor (FLS) for KRR	94
7.3 Application to Test Model	102

8. SUMMARY	110
9. CONCLUDING REMARKS	115
REFERENCE LIST	116
APPENDIX A. Symbolic Expressions for the Joint Velocity and Acceleration of a Planar Three Link Robot	124
APPENDIX B. Symbolic Equations of Motion for a Three Link Robot on a Flexible Base	127
APPENDIX C. Flexible Energy Dissipation using Acceleration Measurements ..	131
APPENDIX D. Proof of Positive Definiteness of Closed-Loop Damping Matrix for the GPM	134
APPENDIX E. Choosing the Velocity Feedback Gain for Minimization Method Redundancy Solutions	136
APPENDIX F. Trajectory Planning Problem	138
APPENDIX G. Sensitivity of Tracking Error, Control Effort, and Maximum Joint Velocity to the Velocity Weighting Matrix	139
APPENDIX H. Sensitivity Curves for Fuzzy Supervisor	143
VITA	147

LIST OF FIGURES

Figure 1. Definition of position vector r_e	6
Figure 2. Coordinate Frame Definitions	13
Figure 3. Inertial, Moving Base, and Robot Coordinate Frames	18
Figure 4. Generic Flexible Structure	22
Figure 5. 1st Stage of Recursive, N-E Equations (for Revolute Joints).	24
Figure 6. 2nd Stage of Recursive, N-E Equations (for Revolute Joints).	25
Figure 7. Generic Flexible Structure with Robot	26
Figure 8. Tracking a Moving Target on a Moving Base	34
Figure 9. Three Link Rigid Robot / Two Model Flexible Base Model.	61
Figure 10. Desired Inertial Trajectories for Three Link Robot.	66
Figure 11. Energy in Flexible Base for Trajectory T, $\gamma = 1 \times 10^{-5}$, $W =$ $M_\theta(\theta)$	68
Figure 12. Energy in Flexible Base for Trajectory Y, $\gamma = 1 \times 10^{-5}$, $W =$ $M_\theta(\theta)$	70
Figure 13. Change in Robot Joint Velocity Norm for RW-DE, $\gamma = 1 \times 10^{-5}$, W $= M_\theta(\theta)$	71
Figure 14. Energy in Flexible Base for Trajectory Y, $\gamma = 1 \times 10^{-2}$, $W =$ $M_\theta(\theta)$	73
Figure 15. Energy in Flexible Base for RW-DE in Trajectory Y as a Function of Velocity Weighting Matrix.	76
Figure 16. Maximum Joint Velocity for RW-DE in Trajectory Y as a Function	

of the Velocity Weighting Matrix.	76
Figure 17. Sensitivity of Mean Tracking Error in Trajectory T, $W=M_0$	79
Figure 18. Sensitivity of Mean Tracking Error in Trajectory Y, $W=M_0$	79
Figure 19. Sensitivity of Required Control Torque in Trajectory T, $W=M_0$	82
Figure 20. Sensitivity of Required Control Torque in Trajectory Y, $W=M_0$	82
Figure 21. Sensitivity of Maximum Joint Velocity in Trajectory T, $W=M_0$	84
Figure 22. Sensitivity of Maximum Joint Velocity in Trajectory Y, $W=M_0$	84
Figure 23. Norm of Control Influence Vector for Stable and Unstable Trajectories.	87
Figure 24. Norm of Desired Control Torque.	88
Figure 25. Hierarchy of Control Strategy.	96
Figure 26. Block Diagram of Adaptive Weighting Factor Computation Algorithm.	97
Figure 27. Fuzzy Membership Functions for Test Model.	104
Figure 28. Energy in Flexible Base using Fuzzy and Non-Fuzzy Weighting with W_1	106
Figure 29. Weighting Factor Variation using Fuzzy Logic Supervisor with W_1 . . .	106
Figure 30. Energy in Flexible Base using Fuzzy and Non-Fuzzy Weighting with W_3	108
Figure 31. Weighting Factor Variation using Fuzzy Logic Supervisor with W_3 . . .	109
Figure A1. Three Link Planar, Revolute, Revolute, Revolute Robot.	125
Figure G1. Sensitivity of Mean Tracking Error to γ and W in Trajectory Y. . . .	141

Figure G2. Sensitivity of Control Effort to γ and W in Trajectory Y.	141
Figure G3. Sensitivity of Maximum Joint Velocity to γ and W in Trajectory Y.	142
Figure H1. Tracking Error Sensitivity for Fuzzy Supervisor in Trajectory Y, RW-DE.	144
Figure H2. Control Effort Sensitivity for Fuzzy Supervisor in Trajectory Y, RW-DE.	145
Figure H3. Maximum Joint Velocity Sensitivity for Fuzzy Supervisor in Trajectory Y, RW-DE.	146

LIST OF TABLES

Table I. Flexible Base Energy Dissipation Solutions.	49
Table II. Test Model Physical Parameters.	62
Table III. Base and Robot Energy Dissipation Solutions.	63
Table IV. Example of Possible Fuzzy Rules	100
Table V. Rate and Truth Factors for Fuzzy Simulation	103
Table VI. Normalized Data Points for Fuzzy Supervisor Sensitivity Plots.	143

NOMENCLATURE

\underline{A}_0	Acceleration of manipulator base coordinate frame
B	Control influence matrix
D	Damping matrix for flexible structure
E	Scalar optimization function
E_f	Energy in flexible base
E_r	Robot joint velocity norm
E_{tot}	Total energy in systems
\underline{F}_i	Inertial force acting on robot joint number i
\underline{F}_r	Modal force on flexible base due to robot joint accelerations
G	Generalized inertia sensitivity matrix for flexible base
H_{ij}	Inertia matrices
I	Identity matrix
$I_{c,i}$	Inertia about the link center of mass for link i
J_H^+	Inertia weighted Jacobian pseudoinverse
J_{1S}^+	One step ahead Jacobian pseudoinverse
J_0	Jacobian matrix which transforms the inertial manipulator base velocity to the inertial end-effector velocity
J_t	3 x n Jacobian matrix between end-effector position and robot joints
J_a	3 x n Jacobian matrix between end-effector orientation and robot joints
J_{MB}	Moving Base Jacobian
${}^i J$	m x n Jacobian matrix between end-effector position and orientation and

	robot joints expressed in coordinate frame i
L	Scalar cost function
M_c	Virtual controller mass matrix
M_i	Mass of link i
M_p	Mass coupling matrix
M_s	Mass matrix for flexible structure
M_θ	Mass matrix of robot
\underline{N}_i	Inertial moment acting on robot joint number i
\underline{N}_τ	Non-linear Coriolis/centrifugal torque vector acting on robot joints
\underline{N}_p	Non-linear Coriolis/centrifugal torque vector acting on flexible base
\underline{N}_w	Non-linear Coriolis/centrifugal force/torque vector acting on joint 1
N_{wf}	Non-linear Coriolis/centrifugal force vector acting on joint 1
$\underline{N}_{w\tau}$	Non-linear Coriolis/centrifugal torque vector acting on joint 1
P	Left singular value matrix
${}^i P_{c,i}$	Center of mass location of link i expressed in i -th coordinate system
${}^i P_{i+1}$	Origin of joint $i+1$ expressed om joint i 's coordinate frame
P_2	Matrix of left nullspace vectors
Q	Right singular value matrix
R^i	i -th fuzzy rule
R	Left singular value matrix of B
R_{ab}	3×3 transformation matrix from frame b to frame a
S	Matrix of right nullspace vectors

T_{ij}	6 x 6 translational and angular velocity transformation matrix from frame j to i
T_m	Maneuver time
T_{pseudo}	Pseudo kinetic energy computed using 1 step ahead velocity
U	Right singular value matrix of B
\underline{V}_b	Translational and angular velocity of flexible base
W, W_i	Robot joint velocity weighting matrices
\underline{W}_i	Applied force/torque vector for i-th grid point on flexible structure
\underline{W}_s	Applied force/torque vector for entire flexible structure
X_j	j-th grid point on flexible structure
\underline{X}_j	Position and orientation of j-th grid point on flexible structure
\underline{X}_{j0}	Nominal position and orientation of j-th grid point on flexible structure
$\underline{X}_{\text{rel}}$	Relative position and orientation between the base of the robot and the target
\underline{X}_T	Position and orientation of moving target
\underline{Z}	Position and orientation vector of entire flexible structure
a_i	Length of link i
$c\theta_i$	$\text{COS}(\theta_i)$
d	Position and orientation of the moving target relative to the end-effector
\underline{d}_m	Desired maneuver move distance
err	Average root sum square tracking error
f_i	3 x 1 force vector acting at i-th location

g	Acceleration due to gravity
g_i	Constraint equations
$(h_{pp})_i$	i -th column in the inertia matrix H_{pp}
k_{GPM}	Flexible velocity feedback gain for the Gradient Projection Method
k_{MM1}	Flexible velocity feedback gain for the Minimization Method
k_1, k_2	Rate and position gains, respectively for target tracking
m	Number of task space variables
m_i	Mass of link i
n	Number of robot degrees of freedom
n_c	Number of virtual controller modes
n_g	Number of grid points on flexible structure
n_p	Number of flexible modes on flexible structure
\underline{n}_i	3×1 moment vector acting at i -th location
\underline{p}	Modal coordinates for flexible base
q	Arbitrary coordinate system
r	Number of redundancies
\underline{r}_{b0}	Translation between origin of flexible base coordinate frame and manipulator base coordinate frame
$\underline{r}_{c,i}$	Center of mass location for i -th link
\underline{r}_e	End-effector position vector relative to the manipulator base
\underline{r}_{ie}	End-effector position vector relative to inertial space
\underline{r}_j	Position of j -th grid point

\mathbf{r}_0	Inertial position vector of the manipulator base
$s\theta_i$	$\text{Sin}(\theta_i)$
\mathbf{u}	$r \times 1$ redundant acceleration
$\underline{\mathbf{v}}_b$	Translational velocity of the flexible base
$\underline{\mathbf{v}}_0$	Translational velocity of the base of the manipulator
w_i	Weighting factor for i-th criterion
wrench	Force and torque applied at a point
$\underline{\mathbf{x}}_c$	Virtual controller position vector
$\underline{\mathbf{z}}$	State vector for flexible base coordinates
$\hat{\mathbf{z}}_i = [0 \ 0 \ 1]^T$	Rotation axis
$\Delta^q \underline{\mathbf{X}}_j$	Position and orientation perturbation for the j-th grid point in coordinate frame q
Λ_c	Virtual control stiffness matrix (diagonal)
Λ_s	Stiffness matrix for flexible structure (non-diagonal)
Λ	Modal Stiffness matrix for flexible structure (diagonal)
Ω	3×1 generalized orientation vector
Φ	Modeshape matrix for entire flexible structure
$\underline{\Psi}_j$	Orientation of j-th grid point
Σ	Singular value matrix
α_0	Coriolis/centrifugal components of acceleration
ϵ_i	Root sum square tracking error at time t_i
$\underline{\eta}_v$	Arbitrary $n \times 1$ robot joint velocity vector

\underline{n}_a	Arbitrary $n \times 1$ robot joint acceleration vector
γ	Composite control weighting factor
$\mu(x)$	Membership function for fuzzy logic supervisor
$\underline{\omega}_b$	Inertial angular velocity of flexible base
$\underline{\omega}_e$	Angular velocity of end-effector relative to the manipulator base
$\underline{\omega}_i$	Angular velocity of the i -th robot link
$\underline{\omega}_{le}$	Inertial angular velocity of end-effector
$\underline{\omega}_0$	Inertial angular velocity of manipulator base
$\underline{\phi}_b$	Modeshape matrix for connecting point between flexible base and robot
$\underline{\rho}$	arbitrary $r \times 1$ robot joint velocity vector
σ	Non-zero diagonal singular value matrix
$\underline{\tau}$	Robot joint torque vector
ξ	Tolerance parameter in KS function
$()^+$	Matrix pseudoinverse operator
$()^T$	Matrix transpose operator
$()^*$	Matrix complex conjugate operator
$()^{-1}$	Matrix inverse operator
(\tilde{x})	Fuzzification operator
$\langle \underline{A} \rangle$	Matrix cross product definition operator
∇_x	Gradient with respect to x operator
$\ \ $	L-2 norm operator
$\mathfrak{R}_{()}$	Rank of the matrix $()$

\mathfrak{R}^n	n-dimensional space
${}^i \underline{x}$	Vector x expressed in coordinate frame i
$\{\underline{X}_i\}$	Cartesian coordinate system i
$\dot{\underline{\theta}}_p, \ddot{\underline{\theta}}_p$	Particular robot joint velocity and acceleration
$\dot{\underline{\theta}}_c, \ddot{\underline{\theta}}_c$	Complementary robot joint velocity and acceleration
$\underline{\theta}, \dot{\underline{\theta}}, \ddot{\underline{\theta}}$	$n \times 1$ robot joint position, velocity, and acceleration vectors, respectively.
${}^i \dot{\underline{x}}_e$	Time derivative of the vector \underline{x}_e expressed in coordinate frame i
$\underline{x}_e, \dot{\underline{x}}_e, \ddot{\underline{x}}_e$	End-effector position, velocity, and acceleration vectors, respectively
\dot{E}_0	Nominal value of the scalar optimization function
${}^I \dot{\underline{X}}_b$	Inertial translational and angular velocity of the flexible base
${}^I \dot{\underline{X}}_e$	Inertial translational and angular velocity of end-effector
${}^I \dot{\underline{X}}_0$	Inertial translational and angular velocity of manipulator base
${}^I \ddot{\underline{X}}_e$	Inertial translational and angular acceleration of end-effector
ARB	Acceleration Relative to Base
ARI	Acceleration Relative to Inertial Space
CBM	Control Base Motion

CE	Control Effort
CPI	Composite Performance Index
DOF	Degrees Of Freedom
FEM	Finite Element Method
FLS	Fuzzy Logic Supervisor
GPM	Gradient Projection Method
KS	Kreisselmeier and Steinhauser
KRR	Kinematically Redundant Robot
MM, MM1	Minimization Method
PDF	Probability Distribution Function
POSE	Position and Orientation
RRR	Revolute, Revolute, Revolute
RSS	Root Sum Square
SF	Scale Factor
SRM	Stabilize Robot Motion
SVD	Singular Value Decomposition
SRMS	Shuttle Remote Manipulator System

1. INTRODUCTION

To minimize the risks to humans and reduce mission costs, future space applications will utilize teleoperated robotic systems such as the Space Station Remote Manipulator System (SSRMS), the Special Purpose Dexterous Manipulator (SPDM), and the Dexterous Orbiter Servicing System (DOSS) [1]. As currently envisioned, these systems will be attached directly to the structure of either the Shuttle Remote Manipulator System (SRSM) or the space station's truss. Since these structures are flexible, a major control issue associated with teleoperated systems is the effect of the motion of the robot on its supporting flexible base [2-4].

According to Newton's third law, forces and moments generated by the robot's motion will create equal and opposite reactions on the base. Since the base is not rigid and massive, as is the case for typical ground-based robots, these reactions can excite flexible modes and cause unwanted base vibration. If uncompensated, these vibrations cause end-effector tracking errors and have the potential to create unstable motion [5].

Since the vibration is caused by robot joint motion, global trajectory optimization schemes can be used to determine joint trajectories that minimize base motion. This, however, requires knowledge of the initial and final states of the end-effector as well as the travel time. If this information is unknown, as would be the case for real-time human controlled operations, local instead of global

optimizations must be performed. As a result, global optimization schemes cannot be applied.

Because global optimizations are not available, local joint optimizations must be used to compute joint trajectories that track the desired end-effector motion, while simultaneously minimizing the effects of the base. Since a six degree of freedom (DOF) robot has only one joint velocity (acceleration) vector for a given, desired end-effector velocity (acceleration), tracking a 6-DOF end-effector trajectory and minimizing flexible base vibrations requires using a robot that has more than six DOF. Thus, a natural question to ask is "How should the additional or redundant degrees of freedom be used to minimize or control flexible base vibration?"

To answer this question, several studies on vibration damping in robotic systems have been done [6-22]. For example, Book and Lee [6,7] showed that a tip robot could be used as an inertial actuator to damp unwanted flexible vibrations in a base such as the SRMS after the base was used for coarse positioning of the tip robot. Likewise, Barbieri et. al. [8,9] used optimal set-point relegation theory to demonstrate that a kinematically redundant tip robot could be used to damp residual or post robot maneuver vibration in a flexible base.

To reduce vibration during the robot maneuver, de Silva et. al. [10-12] and others [13-15] have studied using optimal trajectory shaping. The main emphasis has been on using kinematically redundant robots to locally and globally minimize a quadratic cost function of the forces and moments on the base. However, none of these studies focused on stability issues associated with the minimization.

Related work has also been done in the control of free flying and mobile manipulators. In the case of free flying robots, research has been conducted into finding robot joint paths which reduce the disturbance transmitted to the base [16,17] or use so-called "fixed attitude reference Jacobians" to adjust joint motion to account for base motion [18,19]. In the control of mobile manipulators problem, so-called "moving base" Jacobians have been proposed in conjunction with feedforward control for reducing tracking errors of a robot mounted on a moving or compliant platform [20-22].

Unfortunately, these approaches have drawbacks that limit their effectiveness and applicability to the present problem. For instance, in the vibration suppression approaches, robot motion and base motion were considered separately even though the robot is the primary cause of flexible vibration. Similarly, in the trajectory optimization approaches, robot and base motion were considered jointly but only for known end-effector trajectories (initial and final points specified). Furthermore, none of the above approaches analyzed the stability of the system.

Thus, the goal of the present research is to investigate alternative vibration damping approaches to overcome some of the previous limitations. In particular, this research focuses on direct use of the kinematic redundancies in rigid robots to reduce or damp flexible base motion when the end-effector trajectory is unknown. The principal differences between this and previous studies are:

- 1) end-effector and base motion are considered jointly, not separately,
- 2) no knowledge of the final position or time is used by the controller,

- 3) the direct relationship between the kinematic redundancies and the flexible motion is established, and
- 4) stability issues are considered.

For the purposes of this work, three assumptions were made before undertaking this investigation. First, as was previously mentioned, only rigid robots are considered. This was done to distinguish the base vibration problem from the flexible manipulator vibration problem. Second, the complete end-effector trajectory is assumed to be unknown by the controller. Instead, the controller receives trajectory commands from an operator or other input device. These commands are assumed to be fixed and cannot be altered for the purposes of vibration damping. Finally, all required information is assumed known.

The organization of this document is as follows. First, the kinematic equations of motion for redundant robots are derived and discussed in relation to both fixed and moving bases. Next, the governing equations of motion for a rigid robot on a flexible base are presented. Following this, redundancy control laws that reduce flexible vibration are presented and examined. Then, numerical simulations done on a three link are presented to demonstrate the effectiveness and limitations of the proposed control laws. Next, a fuzzy logic supervisor is designed and shown to be capable of eliminating instability problems associate with high joint velocities and singularities. Finally, some concluding remarks and suggestions for further research are given.

2. KINEMATICS

By definition, a kinematically redundant robot (KRR) is a robot that has more degrees of freedom (DOF) than required for a given task. Typically, the DOF are the robot joint angles for rotational joints and joint lengths for prismatic joints, while the task is the positioning and orienting of the end-effector. Owing to the extra degrees of freedom, additional tasks such as obstacle, singularity, and joint limit avoidance, energy minimization, and joint torque minimization can be accomplished [23].

Accomplishing these tasks requires knowledge of the kinematic constraint equations between the end effector and the robot joint angles¹. In section 2.1 these equations are developed for kinematically redundant robots on fixed bases, while in section 2.2 they are developed for robots on moving bases.

2.1 Fixed Base Kinematics

Consider Figure 1 which shows a generic n-link robot operating in 3-d space described by the Cartesian coordinate system $\{X_0\}$ known as the base frame. In most robotic applications, the primary function of the robot is to accurately move the tip or end-effector of the robot. This requires knowing the complex relationship between the robot joint angles and the position and orientation (POSE) of the end-effector.

¹ For simplicity, hereafter the term "angle" refers to both rotational and prismatic joints.

The position is a non-linear function of the $n \times 1$ vector $\underline{\theta}$ of robot joint angles and can be described in frame 0 by the 3×1 vector 0r_e . The orientation is also a function of the robot joint angles but is not so easily described. Typically, it is represented by either Euler angles, quaternions, or rotation matrices. Regardless of the chosen description, only three generalized coordinates are needed to describe the end-effector's orientation.

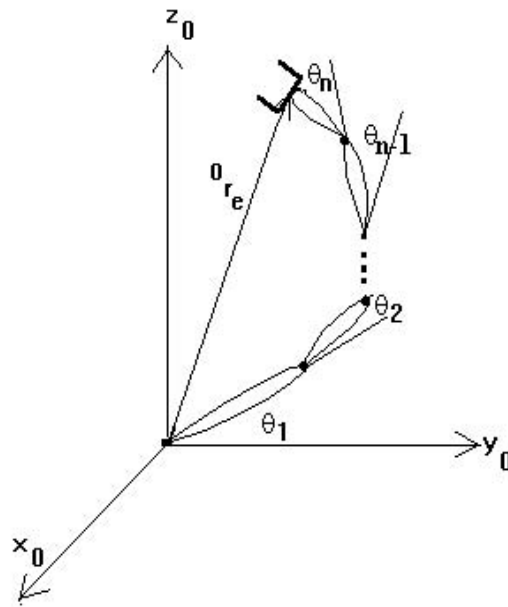


Figure 1. Definition of position vector r_e .

If these three coordinates are designated by $\underline{\Omega}$, then the POSE and first derivative of the POSE of the end-effector are given by

$${}^0\dot{\underline{x}}_e = \begin{bmatrix} {}^0\dot{\underline{r}}_e(\underline{\theta}) \\ \underline{\dot{\Omega}}(\underline{\theta}) \end{bmatrix} \equiv f(\underline{\theta}) \quad (1)$$

and

$$\frac{d}{dt} [{}^0\dot{\underline{x}}_e] = \begin{bmatrix} {}^0\dot{\underline{r}}_e(\underline{\theta}, \dot{\underline{\theta}}) \\ \underline{\dot{\Omega}}(\underline{\theta}, \dot{\underline{\theta}}) \end{bmatrix} = \begin{bmatrix} {}^0\mathbf{J}_t(\underline{\theta}) \dot{\underline{\theta}} \\ \underline{\dot{\Omega}}(\underline{\theta}, \dot{\underline{\theta}}) \end{bmatrix} \quad (2)$$

where the 3 x n matrix ${}^0\mathbf{J}_t$ is the Jacobian between the robot joint velocities and the end-effector translational velocities referenced to coordinate frame 0. Since $\underline{\dot{\Omega}}$ is a non-linear function of the robot joint velocities but can be obtained from the angular velocity ${}^0\dot{\underline{\omega}}_e$ of the end-effector which is linearly related to the robot joint velocities [24], equation (2) is normally replaced by

$${}^0\dot{\underline{x}}_e = \begin{bmatrix} {}^0\dot{\underline{r}}_e \\ {}^0\dot{\underline{\omega}}_e \end{bmatrix} = \begin{bmatrix} {}^0\mathbf{J}_t \\ {}^0\mathbf{J}_a \end{bmatrix} \dot{\underline{\theta}} = {}^0\mathbf{J} \dot{\underline{\theta}} \quad (3)$$

where the 3 x n matrix ${}^0\mathbf{J}_a$ is the angular counterpart to ${}^0\mathbf{J}_t$. It relates robot joint velocities to end-effector angular velocities. Combining ${}^0\mathbf{J}_t$ and ${}^0\mathbf{J}_a$ leads to the total Jacobian ${}^0\mathbf{J}$. Similar to ${}^0\mathbf{J}_a$ and ${}^0\mathbf{J}_t$, ${}^0\mathbf{J}$ is a function of the robot joint angles and represents the linear mapping between the robot joint velocities and the velocity of the end-effector as expressed in frame 0. If the end-effector velocities are with respect to any other frame, say frame i, then ${}^0\dot{\underline{x}}_e$ becomes ${}^i\dot{\underline{x}}_e$ and ${}^0\mathbf{J}$ is replaced by

$${}^i J = \begin{bmatrix} R_{i0} & 0 \\ 0 & R_{i0} \end{bmatrix} {}^0 J \quad (4)$$

where R_{i0} is the 3 x 3 transformation matrix from frame 0 to frame i.²

Equation (3) relates robot joint angle velocities to end-effector velocities. A similar equation exists for the accelerations. Differentiating equation (3) with respect to time yields

$$\ddot{\mathbf{x}}_e = \dot{\mathbf{J}}\dot{\boldsymbol{\theta}} + \mathbf{J}\ddot{\boldsymbol{\theta}}. \quad (5)$$

Inspection of equation (5) shows that the acceleration of the end-effector is the sum of the Coriolis and centrifugal robot joint accelerations ($\dot{\mathbf{J}}\dot{\boldsymbol{\theta}}$) and a linear combination of the second time derivative of the robot joint angles. Thus, similar to equation (3), equation (5) can also be thought of as a linear mapping. It maps robot joint angle accelerations to the term $\ddot{\mathbf{x}}_e - \dot{\mathbf{J}}\dot{\boldsymbol{\theta}}$. As a result, any statements made concerning velocities apply to accelerations.

Together, equations (1), (3) and (5) represent the forward kinematics. That is, given the linear and angular velocity and acceleration of the robot joints, these equations compute the POSE and time rates of change of the POSE. Often however, the reverse situation exists, i.e. given the POSE kinematics, compute the robot kinematics. This situation is known as the reverse kinematics problem. It differs from

² To reduce notation, hereafter all quantities will be assumed to be expressed in frame 0 unless specified. Thus, ${}^0 J$ will be written simply as J .

the forward kinematics in that linear one-to-one mappings are replaced by non-linear one-to-many mappings. For instance, application of equation (1) yields one and only one solution for the POSE of an end-effector, even though the same end-effector POSE may correspond to many different robot configurations [25].

For a redundant robot, similar statements exist for equations (3) and (5). In particular, since the Jacobian matrix of a redundant robot has more columns than rows, equations (3) and (5) are under-determined with respect to solving for either $\underline{\dot{\theta}}$ or $\underline{\ddot{\theta}}$. As a result, inverting equations (3) or (5) yields an infinite number of joint velocity or joint acceleration solutions for a given end-effector velocity or acceleration. As in linear algebra problems, these solutions are the sum of a particular and complementary solutions [26], i.e.

$$\begin{aligned}\underline{\dot{\theta}} &= \underline{\dot{\theta}}_p + \underline{\dot{\theta}}_c \\ &= \mathbf{J}^+ \dot{\mathbf{x}}_e + (\mathbf{I} - \mathbf{J}^+ \mathbf{J}) \underline{\eta}_v\end{aligned}\quad (6)$$

and

$$\begin{aligned}\underline{\ddot{\theta}} &= \underline{\ddot{\theta}}_p + \underline{\ddot{\theta}}_c \\ &= \mathbf{J}^+ (\ddot{\mathbf{x}}_e - \dot{\mathbf{J}} \dot{\underline{\theta}}) + (\mathbf{I} - \mathbf{J}^+ \mathbf{J}) \underline{\eta}_a\end{aligned}\quad (7)$$

where \mathbf{I} is the $n \times n$ identity matrix, $\underline{\eta}_v$ and $\underline{\eta}_a$ are, respectively, arbitrary $n \times 1$ velocity and acceleration vectors, and \mathbf{J}^+ is the $n \times m$ pseudoinverse of \mathbf{J} .

As seen in equations (6) and (7), the particular solution is the solution required to move the end-effector and is dependent on the desired end-effector motion. The homogeneous solution, on the other hand, is dependent on the nullspace projection

matrix $I-J^+J$ and defines all solutions that do not effect the end-effector. It corresponds to self-motion of the robot.

Inspection of equations (6) and (7) also show that both the homogeneous and the particular solutions are functions of J^+ . By definition, J^+ is the unique matrix that satisfies the following four conditions [27]

$$\begin{aligned}
 JJ^+J &= J \\
 J^+JJ^+ &= J^+ \\
 (JJ^+)^* &= JJ^+ \\
 (J^+J)^* &= J^+J
 \end{aligned} \tag{8}$$

where the symbol " $()^*$ " denotes the complex conjugate transpose. If J is full rank, J^+ can be obtained using Lagrange multipliers by minimizing $L = \dot{\theta}^T \dot{\theta} / 2$ subject to equation (3). The net result is $J^+ = J^T (JJ^T)^{-1}$. If J is not full rank, the singular value decomposition can be used. The singular value decomposition [26] is a matrix transformation used to decompose any $m \times n$ matrix J into the product of three matrices, P , Σ , and Q via the transformation

$$\begin{aligned}
 J &= P \Sigma Q^T \\
 &= [P_1 \ P_2] \begin{bmatrix} \sigma & 0 \\ 0 & 0 \end{bmatrix} [Q_1 \ S]^T \\
 &= P_1 \sigma Q_1^T
 \end{aligned} \tag{9}$$

where P and Q are, respectively, $m \times m$ and $n \times n$ unitary matrices, P_1 is $m \times m-r$, P_2 is $m \times r$, Q_1 is $n \times n-r$, S is $n \times r$, σ is an $r \times r$ diagonal matrix of singular values, and r is the rank of J . Therefore, using the orthonormal properties of P and Q , an alternate formulation for J^+ is

$$\mathbf{J}^+ = \mathbf{Q}_1 \boldsymbol{\sigma}^{-1} \mathbf{P}_1^T \quad (10)$$

which is valid regardless of whether or not \mathbf{J} is full rank.

Previously, it was stated that the homogeneous solutions, $(\mathbf{I}-\mathbf{J}^+\mathbf{J})\underline{\mathbf{q}}_v$ and $(\mathbf{I}-\mathbf{J}^+\mathbf{J})\underline{\mathbf{q}}_a$, do not affect the motion of the end-effector. Using equations (9) and (10), this statement can now be proved. Indeed, pre-multiplying equation (6) by \mathbf{J} and substituting equations (9) and (10) yields

$$\begin{aligned} \mathbf{J}\dot{\underline{\mathbf{q}}} &= (\mathbf{P}_1 \boldsymbol{\sigma} \mathbf{Q}_1^T)(\mathbf{Q}_1 \boldsymbol{\sigma}^{-1} \mathbf{P}_1^T) \dot{\underline{\mathbf{x}}}_e + (\mathbf{P}_1 \boldsymbol{\sigma} \mathbf{Q}_1^T) [\mathbf{I} - (\mathbf{Q}_1 \boldsymbol{\sigma}^{-1} \mathbf{P}_1^T)(\mathbf{P}_1 \boldsymbol{\sigma} \mathbf{Q}_1^T)] \underline{\mathbf{q}}_v \\ &= \dot{\underline{\mathbf{x}}}_e. \end{aligned} \quad (11)$$

Similar techniques can be performed for the acceleration equation.

Equations (9) and (10) can also be used to relate the nullspace projection matrix $\mathbf{I}-\mathbf{J}^+\mathbf{J}$ and the matrix of nullspace vectors \mathbf{S} in \mathbf{Q} . Substituting the expressions for \mathbf{J} and \mathbf{J}^+ into $\mathbf{I}-\mathbf{J}^+\mathbf{J}$ gives

$$\begin{aligned} \mathbf{I}-\mathbf{J}^+\mathbf{J} &= \mathbf{I} - (\mathbf{Q}_1 \boldsymbol{\sigma}^{-1} \mathbf{P}_1^T)(\mathbf{P}_1 \boldsymbol{\sigma} \mathbf{Q}_1^T) \\ &= \mathbf{I} - \mathbf{Q}_1 \mathbf{Q}_1^T \\ &= \mathbf{S}\mathbf{S}^T \end{aligned} \quad (12)$$

where the last equality follows from the fact that $\mathbf{Q}\mathbf{Q}^T=\mathbf{I}$. The above form is attractive because it can be used to reduce the complementary solutions to a more useful form.

Substituting equation (12) into the complementary terms of equations (6) and (7)

yields

$$\begin{aligned} \dot{\underline{\mathbf{q}}}_c &= \mathbf{S}\mathbf{S}^T \underline{\dot{\mathbf{q}}}_v \\ \ddot{\underline{\mathbf{q}}}_c &= \mathbf{S}\mathbf{S}^T \underline{\ddot{\mathbf{q}}}_a. \end{aligned} \quad (13)$$

Since \underline{n}_v and \underline{n}_a lie in \mathbb{R}^n and the n columns of Q form a basis for \mathbb{R}^n , they can be written as a linear combination of the m columns of Q_1 and the r columns of S , i.e.

$$\begin{aligned}\underline{n}_v &= Q_1 \underline{y}_1 + S \underline{\rho} \\ \underline{n}_a &= Q_1 \underline{y}_2 + S \underline{u},\end{aligned}\tag{14}$$

where \underline{y}_i ($i=1, 2$) are $m \times 1$ coefficient vectors, while $\underline{\rho}$ and \underline{u} are $r \times 1$ vectors ($r = n - m$). If these relations are substituted into equation (13), the complementary solutions become

$$\begin{aligned}\dot{\theta}_c &= SS^T(Q_1 \underline{a}_1 + S \underline{\rho}) = S \underline{\rho} \\ \ddot{\theta}_c &= SS^T(Q_1 \underline{a}_2 + S \underline{u}) = S \underline{u}\end{aligned}\tag{15}$$

which shows that only r , and not n , parameters can be arbitrarily specified when inverting either the velocity or acceleration equations. For this reason, equations (15) are more desirable than the standard solutions $(I - J^+J)\underline{n}_v$ and $(I - J^+J)\underline{n}_a$. Therefore, the solution to the reverse kinematic problem for fixed bases can be written as³

$$\underline{\dot{\theta}} = J^+ \dot{\underline{x}}_e + S \underline{\rho}\tag{16}$$

and

$$\underline{\ddot{\theta}} = J^+(\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}}) + S \underline{u}.\tag{17}$$

³ To see an example of these, see Appendix A.

2.2 Moving Base Kinematics

The above analysis is valid for fixed base manipulators or when the motion of the end-effector is specified with respect to the base of the manipulator. If the base is moving and the end-effector motion is specified with respect to an inertial coordinate system, the previous results must be modified to include the effects of the non-stationary base. This is accomplished by deriving the kinematic equations for the inertial translational and angular velocities of the end-effector.

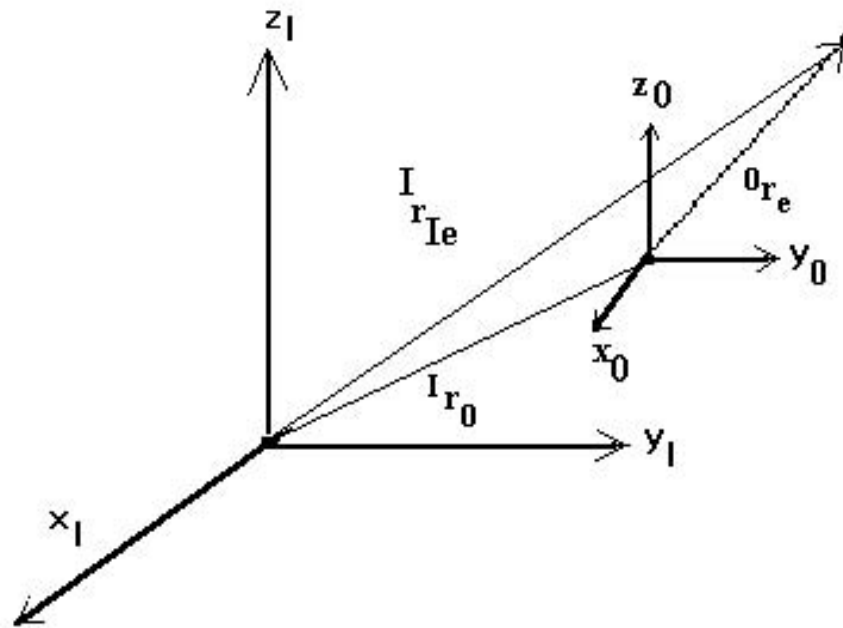


Figure 2. Coordinate Frame Definitions

The inertial translational velocity of the end-effector is found by differentiating the inertial position vector. Referring to Figure 2, the inertial position of the end-

effector, \underline{r}_{1e} is given by

$${}^I \underline{r}_{1e} = {}^I \underline{r}_0 + R_{10} {}^0 \underline{r}_e \quad (18)$$

where ${}^I \underline{r}_0$ is the inertial position of the manipulator base and R_{10} is a 3 x 3 transformation matrix which transforms vectors in the manipulator base frame 0 to the inertial frame I. Differentiating equation (18) and using the nonlinear relationship between the time derivative of R_{10} and the inertial angular velocity of the base ${}^I \underline{\omega}_0$ yields [24]

$${}^I \dot{\underline{r}}_{1e} = \dot{{}^I \underline{r}}_0 + {}^I \underline{\omega}_0 \times R_{10} {}^0 \underline{r}_e + R_{10} {}^0 \dot{\underline{r}}_e. \quad (19)$$

For notational purposes, the expression for ${}^I \dot{\underline{r}}_{1e}$ can be put into matrix form and

written as a linear relationship between robot and base velocities. This is

accomplished by substituting ${}^0 \dot{\underline{r}}_e = \underline{J} \dot{\underline{\theta}}$ from equation (3) and using the matrix

formulation of the cross product, i.e. $\underline{A} \times \underline{B} \equiv \langle \underline{A} \rangle_x \underline{B}$ where $\langle \underline{A} \rangle_x$ is defined by

$$\langle \underline{A} \rangle_x \equiv \begin{bmatrix} 0 & -A_3 & A_2 \\ A_3 & 0 & -A_1 \\ -A_2 & A_1 & 0 \end{bmatrix} \quad (20)$$

to obtain

$${}^I \underline{\dot{r}}_{Ie} = \begin{bmatrix} R_{I0} & {}^0 J_t & I & -\langle R_{I0} & {}^0 r_e \rangle_x \\ & & & & \end{bmatrix} \begin{bmatrix} \dot{\underline{\theta}} \\ I \dot{\underline{r}}_0 \\ I \underline{\omega}_0 \end{bmatrix}. \quad (21)$$

Similarly, the inertial angular velocity of the end-effector, $\underline{\omega}_{Ie}$, is the sum of the angular velocity of the base and the angular velocity of the end-effector relative to the base,

$$\begin{aligned} {}^I \underline{\omega}_{Ie} &= {}^I \underline{\omega}_o + R_{I0} {}^0 \underline{\omega}_e \\ &= \begin{bmatrix} R_{I0} & {}^0 J_a & 0 & I \end{bmatrix} \begin{bmatrix} \dot{\underline{\theta}} \\ I \dot{\underline{r}}_o \\ I \underline{\omega}_o \end{bmatrix}. \end{aligned} \quad (22)$$

Therefore, the inertial POSE velocity of the end-effector is given by

$$\begin{aligned} {}^I \underline{\dot{X}}_e &= \begin{bmatrix} {}^I \dot{\underline{r}}_{Ie} \\ {}^I \underline{\omega}_{Ie} \end{bmatrix} = \begin{bmatrix} R_{I0} & {}^0 J_t & I & -\langle R_{I0} & {}^0 r_e \rangle_x \\ R_{I0} & {}^0 J_a & 0 & I \end{bmatrix} \begin{bmatrix} \dot{\underline{\theta}} \\ I \dot{\underline{r}}_0 \\ I \underline{\omega}_o \end{bmatrix} \\ &= {}^I J_{MB} \begin{bmatrix} \dot{\underline{\theta}} \\ I \dot{\underline{X}}_0 \end{bmatrix} \end{aligned} \quad (23)$$

Compared to the fixed base equation, equation (23) shows that the inertial POSE velocity of an end-effector is the product of a moving base Jacobian matrix ${}^I J_{MB}$ and a system velocity $\begin{bmatrix} \dot{\underline{\theta}}^T & I \dot{\underline{X}}_0^T \end{bmatrix}^T$. However, because the base velocities $I \dot{\underline{X}}_0$ are not control inputs, they do not add extra degrees of freedom. Therefore, equation (23) cannot be

inverted in the same manner as equations (3) and (5).

To circumvent this problem, ${}^I J_{MB}$ can be partitioned into ${}^I J_{MB} = [{}^I J \quad {}^I J_0]$ where ${}^I J$ is the $6 \times n$ fixed base Jacobian referenced to the inertial frame and ${}^I J_0$ consists of the last six columns of ${}^I J_{MB}$. Equation (23) then becomes

$${}^I \dot{\underline{X}}_e = {}^I \underline{J} \dot{\underline{\theta}} + {}^I J_0 \dot{}^I \underline{X}_0. \quad (24)$$

As a result, the inertial end-effector POSE acceleration is given by

$${}^I \ddot{\underline{X}}_e = {}^I \underline{J} \ddot{\underline{\theta}} + {}^I J_0 \ddot{}^I \underline{X}_0 + {}^I \dot{\underline{J}}_{MB} \begin{bmatrix} \dot{\underline{\theta}} \\ \dot{}^I \underline{X}_0 \end{bmatrix} \quad (25)$$

where the time derivative of ${}^I J_{MB}$ is

$${}^I \dot{\underline{J}}_{MB} = \begin{bmatrix} R_{I0} \quad {}^0 \underline{j}_t & 0 & -\langle {}^I \underline{\omega}_0 \times (R_{I0} \quad {}^0 \underline{r}_e) + 2(R_{I0} \quad {}^0 \underline{j}_t \dot{\underline{\theta}}) \rangle_x \\ R_{I0} \quad {}^0 \underline{j}_a & 0 & -\langle R_{I0} \quad {}^0 \underline{j}_a \dot{\underline{\theta}} \rangle_x \end{bmatrix}. \quad (26)$$

Finally, using equations (24), (25), and (15), the reverse kinematics for a manipulator on a moving base are

$$\dot{\underline{\theta}} = {}^I J^+ \left({}^I \dot{\underline{X}}_e - {}^I J_0 \dot{}^I \underline{X}_0 \right) + S \underline{\rho} \quad (27)$$

and

$$\ddot{\underline{\theta}} = {}^I J^+ \left({}^I \ddot{\underline{X}}_e - {}^I J_0 \ddot{}^I \underline{X}_0 - {}^I \dot{\underline{J}}_{MB} \begin{bmatrix} \dot{\underline{\theta}} \\ \dot{}^I \underline{X}_0 \end{bmatrix} \right) + S \underline{u}. \quad (28)$$

3. DYNAMICS

The previously developed equations describe the kinematic relationship between the motion of the end-effector and the motion of a robot on a moving base. The equations do not, however, describe the forces and moments necessary to cause the motion. Thus, the present chapter derives the dynamic or governing equations of motion for a rigid robot - moving base system.

The outline of this chapter is as follows. First, for reasons that will become clear later, the moving base forward kinematic equations developed in Section 2.2 are extended to the case for which the origin of the robot base frame $\{\underline{X}_0\}$ is known with respect to a point on the flexible base. Next, component mode synthesis ideas are used to derive the dynamic equations of motion for the flexible base and the robot. Finally, the models for the base and robot are joined to obtain the dynamic equations for the total system.

3.1 Extension of Moving Base Kinematics

Figure 3 shows an inertial reference frame $\{\underline{X}_1\}=\{x_1, y_1, z_1\}$, a flexible base reference frame $\{\underline{X}_b\}=\{x_b, y_b, z_b\}$, and a robot reference frame $\{\underline{X}_0\}=\{x_0, y_0, z_0\}$ which is translated with respect to $\{\underline{X}_b\}$. In section 2.2, kinematic equations were developed for the motion of an end-effector of a robot on a moving base. The equations were

shown to be a function of the inertial velocity ${}^I\dot{\underline{X}}_0$ and acceleration ${}^I\ddot{\underline{X}}_0$ of the robot reference frame. Often however, ${}^I\dot{\underline{X}}_0$ and ${}^I\ddot{\underline{X}}_0$ are functions of the velocity and acceleration, ${}^I\dot{\underline{X}}_b$ and ${}^I\ddot{\underline{X}}_b$, of another point, say X_b on the moving base. As a result, the moving base kinematic equations, i.e. equations (24) and (25), must be modified.

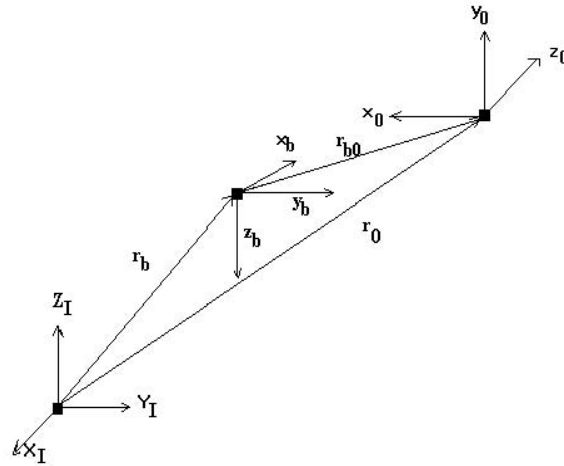


Figure 3. Inertial, Moving Base, and Robot Coordinate Frames

From a physical standpoint, this situation is equivalent to having a model for the motion of a single point X_b on the moving base and expressing the kinematics of the robot in terms of the kinematics of X_b . With respect to equations (24) and (25), this is accomplished by expressing the velocity and acceleration of the robot base in terms of the velocity and acceleration of the origin of $\{X_b\}$. Assuming the translation

${}^b\mathbf{r}_{b0}$ between $\{\underline{X}_b\}$ and $\{\underline{X}_0\}$ is fixed⁴, the translational velocity and acceleration of the robot base are found from differentiating the position vector ${}^b\mathbf{r}_{b0}$ to obtain [28]

$$\begin{aligned} {}^b\mathbf{v}_0 &= \frac{d}{dt}({}^b\mathbf{r}_0) = \frac{d}{dt}({}^b\mathbf{r}_b) + {}^b\boldsymbol{\omega}_b \times {}^b\mathbf{r}_{b0} \\ &\equiv {}^b\mathbf{v}_b + {}^b\boldsymbol{\omega}_b \times {}^b\mathbf{r}_{b0} \end{aligned} \quad (29)$$

and

$$\begin{aligned} {}^b\dot{\mathbf{v}}_0 &= \frac{d^2}{dt^2}({}^b\mathbf{r}_0) = \frac{d}{dt}({}^b\mathbf{v}_b) + {}^b\dot{\boldsymbol{\omega}}_b \times {}^b\mathbf{r}_{b0} + {}^b\boldsymbol{\omega}_b \times ({}^b\boldsymbol{\omega}_b \times {}^b\mathbf{r}_{b0}) \\ &\equiv {}^b\dot{\mathbf{v}}_b + {}^b\dot{\boldsymbol{\omega}}_b \times {}^b\mathbf{r}_{b0} + {}^b\boldsymbol{\omega}_b \times ({}^b\boldsymbol{\omega}_b \times {}^b\mathbf{r}_{b0}) \end{aligned} \quad (30)$$

where ${}^b\boldsymbol{\omega}_b$ and ${}^b\dot{\boldsymbol{\omega}}_b$ are respectively, the inertial angular velocity and acceleration of the known point expressed in the flexible base coordinate frame. Since the robot frame is not rotating relative to the base frame, the total velocity of the robot frame expressed in $\{\underline{X}_0\}$ is given by

$$\begin{aligned} {}^0\mathbf{V}_0 \equiv {}^0\dot{\underline{X}}_0 &= \begin{bmatrix} {}^0\mathbf{v}_0 \\ {}^0\boldsymbol{\omega}_0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{0b} & -\mathbf{R}_{0b} \langle {}^b\mathbf{r}_{b0} \rangle_x \\ \mathbf{0} & \mathbf{R}_{0b} \end{bmatrix} \begin{bmatrix} {}^b\dot{\mathbf{v}}_b \\ {}^b\boldsymbol{\omega}_b \end{bmatrix} \\ &\equiv \mathbf{T}_{0b} {}^b\dot{\underline{X}}_b \end{aligned} \quad (31)$$

while the total acceleration is given by

⁴ Note, assuming ${}^b\mathbf{r}_{b0}$ is fixed implies that the robot is rigidly attached to the flexible base. This assumption eliminates the need to model the coupling between the robot and base.

$$\begin{aligned}
{}^0\mathbf{A}_0 &\equiv {}^0\ddot{\mathbf{X}}_0 = T_{0b} {}^b\ddot{\mathbf{X}}_b + \begin{bmatrix} R_{0b}({}^b\boldsymbol{\omega}_b \times ({}^b\boldsymbol{\omega}_b \times {}^b\mathbf{r}_{b0})) \\ \mathbf{0} \end{bmatrix} \\
&\equiv T_{0b} {}^b\ddot{\mathbf{X}}_b + {}^0\boldsymbol{\alpha}_0.
\end{aligned} \tag{32}$$

Therefore, the modified end-effector velocity and acceleration equations are

$$\begin{aligned}
{}^I\dot{\mathbf{X}}_e &= {}^I\mathbf{J}\dot{\boldsymbol{\theta}} + {}^I\mathbf{J}_0 {}^I\dot{\mathbf{X}}_0 \\
&= {}^I\mathbf{J}\dot{\boldsymbol{\theta}} + {}^I\mathbf{J}_0 \begin{bmatrix} R_{Ib} & -R_{Ib} \langle {}^b\mathbf{r}_{0b} \rangle_x \\ \mathbf{0} & R_{Ib} \end{bmatrix} {}^b\dot{\mathbf{X}}_b \\
&= {}^I\mathbf{J}\dot{\boldsymbol{\theta}} + {}^I\mathbf{J}_0 T_{Ib} {}^b\dot{\mathbf{X}}_b
\end{aligned} \tag{33}$$

and

$$\begin{aligned}
{}^I\ddot{\mathbf{X}}_e &= {}^I\mathbf{J}\ddot{\boldsymbol{\theta}} + {}^I\mathbf{J}_0 {}^I\ddot{\mathbf{X}}_0 + \mathbf{J}_{MB} \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ {}^I\dot{\mathbf{X}}_0 \end{bmatrix} \\
&= {}^I\mathbf{J}\ddot{\boldsymbol{\theta}} + {}^I\mathbf{J}_0 (T_{Ib} {}^b\ddot{\mathbf{X}}_b + {}^I\boldsymbol{\alpha}_0) + \mathbf{J}_{MB} \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ T_{Ib} {}^b\dot{\mathbf{X}}_b \end{bmatrix}.
\end{aligned} \tag{34}$$

It follows that inverse or reverse kinematics are

$$\dot{\boldsymbol{\theta}} = {}^I\mathbf{J}^+ \left({}^I\dot{\mathbf{X}}_e - {}^I\mathbf{J}_0 T_{Ib} {}^b\dot{\mathbf{X}}_b \right) + S\boldsymbol{\rho} \tag{35}$$

and

$$\ddot{\boldsymbol{\theta}} = {}^I\mathbf{J}^+ \left({}^I\ddot{\mathbf{X}}_e - {}^I\mathbf{J}_0 (T_{Ib} {}^b\ddot{\mathbf{X}}_b + {}^I\boldsymbol{\alpha}_0) - \mathbf{J}_{MB} \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ T_{Ib} {}^b\dot{\mathbf{X}}_b \end{bmatrix} \right) + S\mathbf{u} \tag{36}$$

3.2 Flexible Base Model

Considerable research has been done on modelling flexible structures and includes using singular perturbation theory [29], finite element models [30], modal testing [31], and admittance theory [32]. However, for a flexible structure with an attached robot, the ideas behind component mode synthesis [33] or sub-structuring are particularly useful. In sub-structuring, an overall system model is constructed by joining the models of individual components. The advantage to this approach is that it is often easier and cheaper to first construct models for individual elements and then build the overall model. For example, creating a high fidelity finite element model of a space station is costly and requires using a large number of elements, whereas creating and then merging models of individual components is much more manageable [33]. Furthermore, applying sub-structuring to the present problem is appealing since this allows the models of the flexible structure and robot to be developed independently of each other.

Thus, first consider only the flexible structure (see Figure 4) and assume that the structure has been discretized into n_g grid points denoted by X_j . Let the translational and angular displacement of each grid point with respect to a coordinate system q be denoted by ${}^q\underline{X}_j = [{}^q\underline{r}_j \quad {}^q\underline{\Psi}_j]^T$. Assuming small perturbations, the motion of ${}^q\underline{X}_j$ is given by

$${}^q\mathbf{X}_j = {}^q\mathbf{X}_{j0} + \Delta {}^q\mathbf{X}_j, \quad {}^q\mathbf{X}_j = \begin{bmatrix} {}^q\mathbf{r}_j \\ {}^q\mathbf{\Psi}_j \end{bmatrix} \quad (37)$$

where ${}^q\mathbf{X}_{j0}$ is the nominal value and $\Delta {}^q\mathbf{X}_j$ is the perturbation from the nominal.

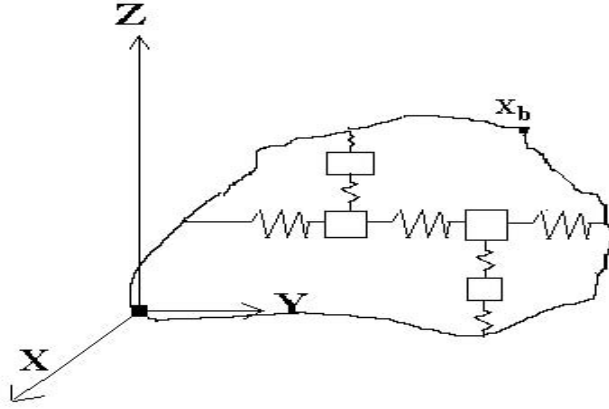


Figure 4. Generic Flexible Structure

Using finite element theory, a linear model for the perturbations is [30]

$$\mathbf{M}_s \ddot{\mathbf{Z}} + \mathbf{\Lambda}_s \mathbf{Z} = \mathbf{W}_s \quad (38)$$

where $\mathbf{Z} = [(\Delta {}^q\mathbf{X}_1)^T \ (\Delta {}^q\mathbf{X}_2)^T \ \dots \ (\Delta {}^q\mathbf{X}_{n_g})^T]^T$ is a $6*n_g \times 1$ vector of perturbations, \mathbf{M}_s is a $6*n_g \times 6*n_g$ mass matrix, $\mathbf{\Lambda}_s$ is a $6*n_g \times 6*n_g$ stiffness matrix, and $\mathbf{W}_s = [\mathbf{W}_1^T \ \mathbf{W}_2^T \ \dots \ \mathbf{W}_{n_g}^T]^T$ is the $6*n_g \times 1$ vector of applied forces and torques often called wrenches. At the i -th grid point, the applied wrench \mathbf{W}_i , is a 6×1 vector composed of the 3×1 applied force \mathbf{f}_i and moment \mathbf{n}_i vectors, i.e.

$$\underline{W}_i = \begin{bmatrix} \underline{f}_i \\ \underline{n}_i \end{bmatrix}. \quad (39)$$

For computational purposes, equation (38) is often written in terms of n_p modal coordinates (denoted by \underline{p}) as

$$\ddot{\underline{p}} + \Lambda \underline{p} = ({}^q\Phi)^T \underline{W}_s \quad (40)$$

where Λ is an $n_p \times n_p$ diagonal matrix with $\Lambda_{ii} = \omega_i^2$, ω_i is the i -ith natural frequency, and ${}^q\Phi$ is the $6 \cdot n_g \times n_p$ mass normalized modeshape matrix defined by

$$\underline{Z} = \begin{Bmatrix} \Delta^q \underline{X}_1 \\ \Delta^q \underline{X}_2 \\ \vdots \\ \Delta^q \underline{X}_{n_g} \end{Bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \cdots & \Phi_{1n_p} \\ \Phi_{21} & \Phi_{22} & \cdots & \Phi_{2n_p} \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{n_g 1} & \Phi_{n_g 2} & \cdots & \Phi_{n_g n_p} \end{bmatrix} \underline{p} \equiv \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n_g} \end{bmatrix} \underline{p} = \Phi \underline{p}. \quad (41)$$

Equation (41) also gives the velocity and acceleration at each grid point as

$$\Delta^q \dot{\underline{X}}_j = \phi_j \dot{\underline{p}} \quad (42)$$

and

$$\Delta^q \ddot{\underline{X}}_j = \phi_j \ddot{\underline{p}}. \quad (43)$$

3.3 Rigid Robot Model

Equations (40) and (41) describe the small displacement motion of a flexible base. A similar set of equations can be developed for the motion of a robot on a moving base. Although there are many methods for obtaining these equations (e.g. see

[24], pp. 219-220), this work uses the two stage, recursive algorithm developed by Luh, Walker, and Paul [34] and by Walker and Orin [35].

In the first stage (outlined in Figure 5), the inertial force ${}^i\mathbf{F}_i$ and moment ${}^i\mathbf{N}_i$ on link i ($i=1, \dots, n$) are calculated in terms of the mass M_i and inertia $I_{c,i}$ about the link center of mass by (1) recursively propagating the angular velocity ${}^i\boldsymbol{\omega}_i$ and the total acceleration ${}^i\ddot{\mathbf{X}}_i$ from the base of the robot to the last link, (2) computing the translational acceleration of the center of mass ${}^i\ddot{\mathbf{r}}_{c,i}$, and (3) applying Newton's second law and Euler's equation.

$$\begin{aligned}
 & \text{for } i=0, 1, \dots, n-1 \\
 & \quad {}^{i+1}\boldsymbol{\omega}_{i+1} = \mathbf{R}_{i+1,i} \quad {}^i\boldsymbol{\omega}_i + \dot{\boldsymbol{\theta}}_{i+1} \quad {}^{i+1}\hat{\mathbf{Z}}_{i+1} \\
 & \quad {}^{i+1}\ddot{\mathbf{X}}_{i+1} = \begin{bmatrix} {}^{i+1}\ddot{\mathbf{r}}_{i+1} \\ {}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{i+1,i} \left({}^i\dot{\boldsymbol{\omega}}_i \times {}^i\mathbf{P}_{i+1} + {}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{\omega}_i \times {}^i\mathbf{P}_{i+1}) \right) \\ \mathbf{R}_{i+1,i} \quad {}^i\boldsymbol{\omega}_i \times \dot{\boldsymbol{\theta}}_{i+1} \quad {}^{i+1}\hat{\mathbf{Z}}_{i+1} + \ddot{\boldsymbol{\theta}}_{i+1} \quad {}^{i+1}\hat{\mathbf{Z}}_{i+1} \end{bmatrix} + \begin{bmatrix} \mathbf{R}_{i+1,i} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{i+1,i} \end{bmatrix} \quad {}^i\ddot{\mathbf{X}}_i \\
 & \quad {}^{i+1}\ddot{\mathbf{r}}_{c,i+1} = \left[\mathbf{I} \quad -\langle {}^{i+1}\mathbf{P}_{c,i+1} \rangle \right] \quad {}^{i+1}\ddot{\mathbf{X}}_{i+1} + {}^{i+1}\boldsymbol{\omega}_{i+1} \times ({}^{i+1}\boldsymbol{\omega}_{i+1} \times {}^{i+1}\mathbf{P}_{c,i+1}) \\
 & \quad {}^{i+1}\mathbf{F}_{i+1} = M_{i+1} \quad {}^{i+1}\ddot{\mathbf{r}}_{c,i+1} \\
 & \quad {}^{i+1}\mathbf{N}_{i+1} = I_{c,i+1} \quad {}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} + {}^{i+1}\boldsymbol{\omega}_{i+1} \times I_{c,i+1} \quad {}^{i+1}\boldsymbol{\omega}_{i+1} \\
 & \text{end}
 \end{aligned}$$

Figure 5. 1st Stage of Recursive, N-E Equations (for Revolute Joints).

Then, in the second stage (See Figure 6.), the joint force ${}^i\mathbf{f}_i$ and moment ${}^i\mathbf{n}_i$ acting on link i are calculated by recursively balancing the net force and moment on each link beginning with link n and working backwards to link 1. In the figures, the previously un-defined terms are $\hat{\mathbf{Z}}=[0 \ 0 \ 1]^T$, ${}^i\mathbf{P}_{i+1}$ is the origin of joint $i+1$ expressed

in joint i 's coordinate frame, and ${}^iP_{c,i}$ is the center of mass location of link i .

$$\begin{array}{l}
 \text{for } i = n, n-1, \dots, 1 \quad (2^{\text{nd}} \text{ Stage}) \\
 {}^iW_i = \begin{bmatrix} {}^if_i \\ {}^in_i \end{bmatrix} = \begin{bmatrix} I & 0 & R_{i,i+1} & 0 \\ \langle {}^iP_{c,i} \rangle & I & \langle {}^iP_{i+1} \rangle R_{i,i+1} & R_{i,i+1} \end{bmatrix} \begin{bmatrix} {}^iF_i \\ {}^iN_i \\ {}^{i+1}f_{i+1} \\ {}^{i+1}n_{i+1} \end{bmatrix} \\
 \text{end}
 \end{array}$$

Figure 6. 2nd Stage of Recursive, N-E Equations (for Revolute Joints).

Assuming zero external forces, the resulting robot equations of motion and the wrench on joint one are

$$\underline{\tau} = M_\theta(\theta)\ddot{\theta} + M_p(\theta) {}^0A_0 + \underline{N}_\tau(\theta, \dot{\theta}, {}^0V_0) \quad (44)$$

and

$$\begin{aligned}
 {}^1W_1 &= \begin{bmatrix} {}^1f_1 \\ {}^1\tau_1 \end{bmatrix} = \begin{bmatrix} H_{f\theta}(\theta) \\ H_{\tau\theta}(\theta) \end{bmatrix} \ddot{\theta} + \begin{bmatrix} H_{fp}(\theta) \\ H_{\tau p}(\theta) \end{bmatrix} {}^0A_0 + \begin{bmatrix} \underline{N}_{wf}(\theta, \dot{\theta}, {}^0V_0) \\ \underline{N}_{w\tau}(\theta, \dot{\theta}, {}^0V_0) \end{bmatrix} \\
 &= H_\theta \ddot{\theta} + H_p {}^0A_0 + \underline{N}_w
 \end{aligned} \quad (45)$$

where $\underline{\tau} = [{}^1n_{1,z} \quad {}^2n_{2,z} \quad \dots \quad {}^nn_{n,z}]^T$ is the $n \times 1$ vector of robot joint torques, M_θ is a symmetric positive definite $n \times n$ inertia matrix, M_p is an $n \times 6$ inertia matrix, \underline{N}_τ is an $n \times 1$ vector of non-linear joint torques, $H_{f\theta}$ and H_{fp} are $3 \times n$ and 3×6 base force inertia matrices, $H_{\tau\theta}$ and $H_{\tau p}$ are $3 \times n$ and 3×6 base moment inertia matrices, and \underline{N}_{wf} and $\underline{N}_{w\tau}$ are 3×1 vectors which contain the non-linear components of the wrench on joint one. Inspection of equations (44) and (45) shows that, compared to traditional, fixed base robots where ${}^0A_0 = [0 \ g \ 0]^T$ if the robot is in a gravity field, the

primary effect of a moving base is to cause the "gravity loading" torques $M_p^0 \underline{A}_0$ and $H_p^0 \underline{A}_0$ to be a function of both the configuration of the robot and the acceleration of the moving base.

3.4 Integrated Flexible Base, Rigid Robot Model

Using the idea of sub-structuring, equations (40) and (41) for the flexible base and equations (44) and (45) for the rigid robot were developed independently of each other. However, to model a rigid robot on a flexible base, the two sets of equations must be connected. This is accomplished by expressing the acceleration of the base of the robot and the wrench applied to the flexible structure in terms of the kinematics of the modal coordinates and the robot joint angles.

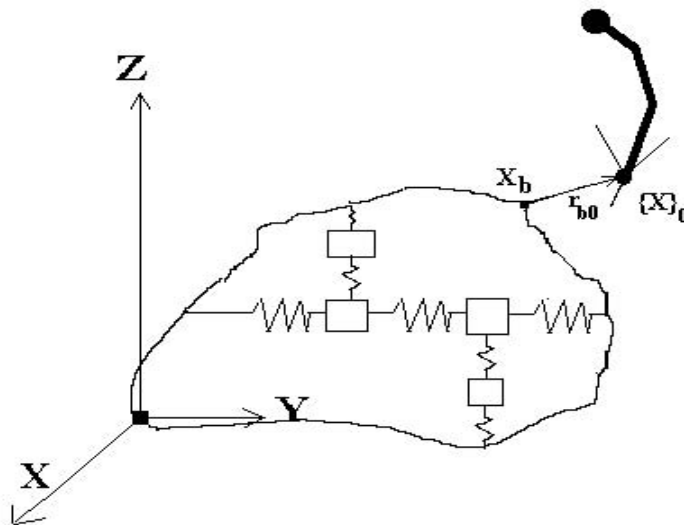


Figure 7. Generic Flexible Structure with Robot

First assume the robot is attached to the flexible structure at grid point X_b via a rigid mount shown as \underline{r}_{b0} in Figure 7. Furthermore, suppose that the only external force acting on the flexible structure occurs at X_b . Under this assumption, the modal equations of motion reduce to

$$\ddot{\underline{p}} + \Lambda \underline{p} = \phi_b^T \underline{W}_b \quad (46)$$

where ϕ_b is the $6 \times n_p$ modeshape matrix for grid point X_b and \underline{W}_b is the wrench applied by the robot due to the desired motion of the end-effector robot. Using equation (45), the wrench acting on the flexible structure is found by rotating the wrench on joint one from coordinate system $\{X_1\}$ to $\{X_0\}$ and then transferring the result to $\{X_b\}$. Mathematically, this process is equivalent to the second stage of the recursive Newton-Euler equations and can be expressed by

$$\begin{aligned} {}^b \underline{W}_b &= - \begin{bmatrix} R_{b0} & 0 \\ \langle {}^b \underline{r}_{ob} \rangle R_{b0} & R_{b0} \end{bmatrix} \left(\begin{bmatrix} R_{01} & 0 \\ 0 & R_{01} \end{bmatrix} {}^1 \underline{W}_1 \right) \\ &\equiv -T_{ob}^T {}^0 \underline{W}_1. \end{aligned} \quad (47)$$

which becomes

$$\begin{aligned} {}^b \underline{W}_b &= -T_{ob}^T \left(\begin{bmatrix} R_{01} H_{j\theta} \\ R_{01} H_{\tau\theta} \end{bmatrix} \ddot{\underline{\Theta}} + \begin{bmatrix} R_{01} H_{fp} \\ R_{01} H_{\tau p} \end{bmatrix} {}^0 \underline{A}_0 + \begin{bmatrix} R_{01} \underline{N}_{wf} \\ R_{01} \underline{N}_{w\tau} \end{bmatrix} \right) \\ &\equiv -T_{ob}^T (\bar{H}_{p\theta} \ddot{\underline{\Theta}} + \bar{H}_{pp} {}^0 \underline{A}_0 + \bar{N}_p) \end{aligned} \quad (48)$$

after substituting equation (45). Note that the minus sign in equation (48) is needed because the wrench acting on the flexible base is the negative of the wrench acting on

the first joint of the robot. Finally, using equations (32), (37), and (43) with $q = j = b$, the expression for the wrench applied to the flexible base becomes

$$\begin{aligned} {}^b\mathbf{W}_b &= -T_{0b}^T(\bar{\mathbf{H}}_{p\theta}\ddot{\Theta} + \bar{\mathbf{H}}_{pp}(T_{0b}\Delta {}^b\ddot{\mathbf{X}}_b + {}^0\alpha_0) + \bar{\mathbf{N}}_p) \\ &= -\mathbf{H}_{p\theta}\ddot{\Theta} - \mathbf{H}_{pp}\Phi_b\ddot{\mathbf{p}} - \mathbf{N}_p. \end{aligned} \quad (49)$$

where $\mathbf{H}_{pp} \equiv T_{0b}^T\bar{\mathbf{H}}_{pp}T_{0b}$, $\mathbf{N}_p \equiv T_{0b}^T(\bar{\mathbf{N}}_p + \bar{\mathbf{H}}_{pp} {}^0\alpha_0)$, and $\mathbf{H}_{p\theta} \equiv T_{0b}^T\bar{\mathbf{H}}_{p\theta}$. Therefore, from

equations (44) and (46), the governing equations of motion for the coupled system are

$$\begin{bmatrix} M_\theta & M_p \\ \Phi_b^T\mathbf{H}_{p\theta} & I + \Phi_b^T\mathbf{H}_{pp}\Phi_b \end{bmatrix} \begin{bmatrix} \ddot{\Theta} \\ \ddot{\mathbf{p}} \end{bmatrix} + \begin{bmatrix} \mathbf{N}_\tau \\ \Phi_b^T\mathbf{N}_p \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \Lambda\mathbf{p} \end{bmatrix} = \begin{bmatrix} \tau \\ \mathbf{0} \end{bmatrix} \quad (50)$$

where due to symmetry $M_p = (\Phi_b^T\mathbf{H}_{pp})^T$. Inspection of equation (50) shows that, because $\Phi_b^T\mathbf{H}_{pp}\Phi_b$ is in general non-diagonal and \mathbf{N}_p contains non-linear terms, the governing equations of motion for the flexible base are coupled, non-linear time varying differential equations. This in contrast to traditional finite element models of flexible structures which are linear and time invariant.

4. VIBRATION REDUCTION CONTROL LAWS

In Chapters 2 and 3, the governing equations of motion were developed for a rigid robot on a flexible base. These equations were shown to consist of a set of coupled, non-linear ordinary differential equations that can be decomposed into a subset for the motion of the flexible base⁵,

$$(I + \Phi_b^T H_{pp} \Phi_b) \ddot{\underline{p}} + \Lambda \underline{p} = -\Phi_b^T (H_{p\theta} \ddot{\underline{\theta}} + \underline{N}_p) \quad (51)$$

and a subset for the rigid robot

$$M_0 \ddot{\underline{\theta}} + M_p \ddot{\underline{p}} + \underline{N}_\tau = \underline{\tau}. \quad (52)$$

Using these equations, the purpose of this chapter is to develop control laws for the kinematic redundancies of the manipulator that reduce or control vibration in the flexible base.

The outline of this chapter is as follows. In Section 4.1, the control problem is formally defined. Then, in Section 4.2, issues regarding the selection of the end-effector trajectory are discussed. Next, in Section 4.3, redundancy control techniques and methods are presented. In Section 4.4, control algorithms for the kinematic redundancies are derived. Finally, for stability reasons, Section 4.5 introduces composite redundancy control.

⁵ For an example of these equations, see Appendix B which presents the equations for a three link rigid robot on a modal model of a flexible base.

4.1 Control Problem Definition and Formulation

The fundamental problem which this work addresses is unwanted vibrations in the supporting flexible structure of a robot. As seen in equation (51) if the base is initially at rest, i.e. $\underline{p} = \dot{\underline{p}} = \underline{0}$, the primary cause of vibration is the modal torque

$$\underline{F}_r = -\Phi_b^T H_{p0} \ddot{\underline{\theta}}.$$

\underline{F}_r adds energy to the flexible structure that excites flexible modes and causes flexible vibration. Therefore, the fundamental problem is to determine robot joint accelerations that cancel or reduce \underline{F}_r .

Generally speaking, there are three approaches for reducing \underline{F}_r . First, if an accurate model of the modal inertia matrix $\Phi_b^T H_{p0}$ does not exist, \underline{F}_r can be reduced by minimizing the robot joint accelerations. However, because the particular joint accelerations are already the minimum norm solutions [27], this approach cannot be taken.

Alternatively, one could choose to reduce the desired end-effector acceleration $\ddot{\underline{x}}_e$ or ${}^I\ddot{\underline{X}}_e$. Although reducing the end-effector acceleration would reduce flexible vibrations by requiring smaller joint accelerations, this approach does have two drawbacks. First, because the duration of the maneuver is inversely related to the magnitude of the end-effector acceleration, reducing $\ddot{\underline{x}}_e$ or ${}^I\ddot{\underline{X}}_e$ increases the time of the maneuver. For earth based applications, the added time may be acceptable but not for space applications where time is a critical factor.

Another drawback to this approach is that the ability to reduce the end-effector acceleration and still reach the final goal requires knowing the complete maneuver in advance. However, as stated in Chapter 1, a fundamental assumption of this work is that the end-effector trajectory is not known in advance.

The last approach is to assume that $\Phi_b^T H_{p0}$ is known and to recall that the redundancies have no direct effect on the desired end-effector trajectory.⁶ For this reason, the redundancies can be used to reduce flexible base motion without changing the desired end-effector acceleration. In this fashion, the available redundancy, not the entire robot, is used as an inertial actuator. The role of the redundancies is to decrease base motion so that (ideally) when the maneuver is completed, there is no motion in the base. In the event of residual base motion, the redundancies are then used to damp the remaining motion while also maintaining the desired final end-effector state.

Compared to the other two approaches, the principal advantage of this approach is that the use of the redundancies for vibration minimization decouples the end-effector tracking problem from the flexible structure vibration problem. This, in turn, reduces the overall complexity of the controller since any existing end-effector tracking algorithm such as that in ref. [36] can be used. As a result, new algorithms need only be developed for reducing flexible vibration instead of algorithms for end-effector tracking and vibration damping.

On the other hand, one disadvantage to this approach is the added computing

⁶ Although the redundancies do not directly affect $\ddot{\underline{X}}$, they do have an indirect effect because the redundancies affect the motion of the base.

expense and the additional knowledge required. For example, at the very least, directly controlling flexible vibrations requires accurate knowledge of the aforementioned modal inertia matrix. Since this matrix is time-varying and is a complex function of the robot joint angles as well as the inertia characteristics of the robot and payload, computing this matrix in real-time poses significant computational challenges.

Notwithstanding the additional computational requirements, it should also be mentioned that the above controller strategy says nothing about how to actually compute the required joint torques in equation (52). That is, the above control strategy only generates the desired particular and complementary robot joint accelerations but does not suggest how to actually achieve them. Since the development of joint torque control laws is an open research problem in and of itself, the present work *assumes* a suitable joint torque control law (e.g., feedforward control [37], configuration control [38], fuzzy logic control [39], and neural-net control [40]) exists such that the desired and actual robot joint accelerations are equivalent.

Assuming this equivalence, the present control problem can be formally defined as follows: Given a particular set of robot joint accelerations needed to move the end-effector, $\ddot{\underline{\theta}}_p$, and a set of Jacobian nullspace vectors, choose the redundancies to reduce vibration in the flexible structure. Mathematically, this amounts to substituting the general robot joint acceleration solution, i.e. $\ddot{\underline{\theta}} = \ddot{\underline{\theta}}_p + \mathcal{S}\underline{u}$, into equation (51) to obtain

$$(I + \Phi_b^T H_{pp} \Phi_b) \ddot{\underline{p}} + \Lambda \underline{p} = -\Phi_b^T (H_{p\theta} (S \underline{u} + \ddot{\underline{\theta}}_p) + \underline{N}_p), \quad (53)$$

and then choosing \underline{u} to damp flexible vibrations.

4.2 End-effector Acceleration Selection

In Section 4.1, it was shown that the particular robot joint accelerations are the primary cause for flexible base vibration. Since the acceleration of the end-effector is an integral part of the particular joint acceleration solution, it is worthwhile to briefly discuss the selection of the end-effector acceleration.

Inspection of the kinematic equations in Chapters 2 and 3 shows that there are two possible choices for specifying the end-effector acceleration: (1) specifying with respect to the base of the manipulator and (2) specifying with respect to an inertial space. For convenience, the governing kinematic equations for these two choices are repeated below

- end-effector Acceleration specified Relative to Base of robot (ARB)

$$\ddot{\underline{\theta}}_p = {}^0J^+ ({}^0\ddot{\underline{x}}_e - {}^0J\dot{\underline{\theta}}) \quad (54)$$

- end-effector Acceleration specified Relative to Inertial space (ARI)

$$\ddot{\underline{\theta}}_p = {}^I J^+ \left(\left\{ {}^I \ddot{\underline{x}}_e - {}^I J_0 {}^I \underline{A}_0 \right\} - {}^I \dot{J}_{MB} \begin{bmatrix} \dot{\underline{\theta}} \\ T_{Ib} {}^b \dot{\underline{x}}_b \end{bmatrix} \right) \quad (55)$$

Traditionally, ARB has been used for fixed based manipulators. Its use is appropriate when the trajectory to be followed is specified relative to the base of the manipulator. ARI, on the other hand, is appropriate when the trajectory to track is

specified in inertial space and the base of the manipulator is moving. If the base is not moving, either approach can be used.

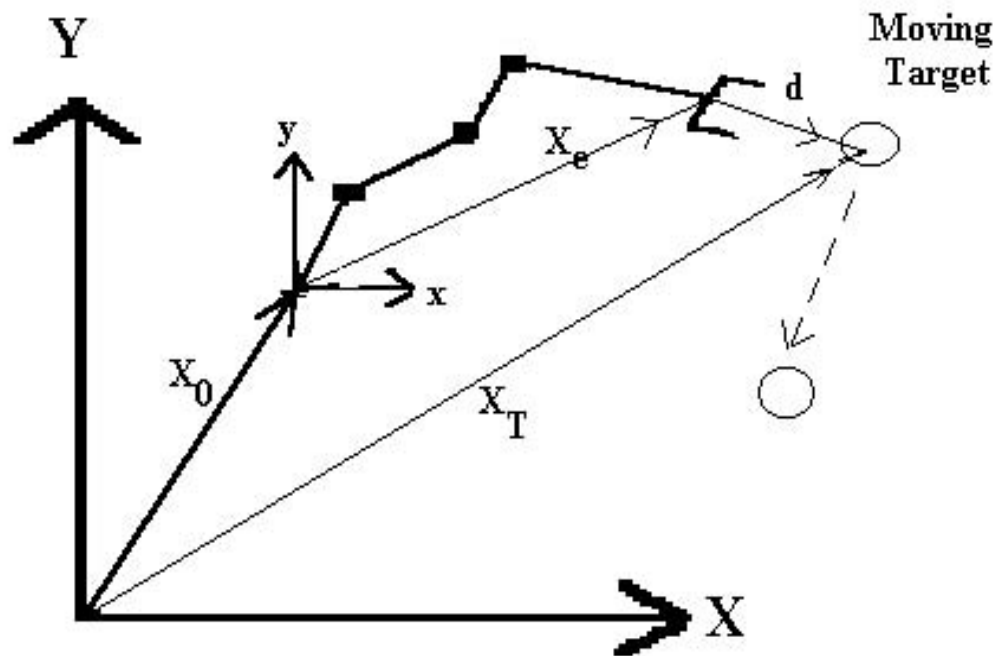


Figure 8. Tracking a Moving Target on a Moving Base

In the present problem, both the base and the tracking object may be moving, Since it assumed that an operator is controlling the motion of the end-effector, the end-effector acceleration inputs will most probably be specified relative to the base of the robot. An example of this scenario is an astronaut controlling a robot that is attached to the SRMS and trying to grasp a slowly moving satellite (Figure 8). Using cameras or other sensing devices located on the robot, the astronaut would control the end-effector by commanding accelerations that are relative to the robot's base. Since

the target and the base are both moving, the desired end-effector acceleration⁷ ${}^0\ddot{\mathbf{x}}_e$ in equation (54) must include the kinematics of the moving base-moving target system.

This is accomplished by noting that the range information the sensor would measure is

$$\underline{\mathbf{d}} = \underline{\mathbf{X}}_T - \underline{\mathbf{X}}_0 - \mathbf{x}_e. \quad (56)$$

Therefore, to ensure that the tracking errors asymptotically go to zero, the acceleration of $\underline{\mathbf{d}}$ should be chosen as

$$\ddot{\underline{\mathbf{d}}} + k_1 \dot{\underline{\mathbf{d}}} + k_2 \underline{\mathbf{d}} = \mathbf{0} \quad (57)$$

which leads to the following desired end-effector acceleration:

$$\ddot{\mathbf{x}}_e = \ddot{\underline{\mathbf{X}}}_{rel} + k_1 (\dot{\underline{\mathbf{X}}}_{rel} - \dot{\mathbf{x}}_e) + k_2 (\underline{\mathbf{X}}_{rel} - \mathbf{x}_e) \quad (58)$$

where $\underline{\mathbf{X}}_{rel} \equiv \underline{\mathbf{X}}_T - \underline{\mathbf{X}}_0$ is the relative motion between the base of the robot and the target.

4.3 Redundancy Control Methods

Before deriving vibration reduction redundancy control laws, some background on redundancy control is useful. As was mentioned earlier, redundant manipulators have been used to accomplish a wide variety of tasks. Typically, these tasks have been performed by using the redundancies to optimize a single criterion, such as the joint torque norm or the distance to an obstacle.

For the optimization algorithm, most applications use the Gradient Projection

⁷ To reduce notation, the end-effector acceleration will be assumed to be specified relative to the robot base frame, $\{\mathbf{X}_0\}$ so that the superscript 0 can be dropped.

Method (GPM) [41]. At the acceleration level, the GPM optimizes a scalar function $E(\theta, \dot{\theta})$ by choosing \underline{u} either to increase or decrease the time derivative of E. Since the time derivative of E is given by

$$\begin{aligned}\dot{E} &= \nabla_{\dot{\theta}} E^T \ddot{\theta} + \nabla_{\theta} E^T \dot{\theta} \\ &= \nabla_{\dot{\theta}} E^T S \underline{u} + (\nabla_{\dot{\theta}} E^T J^+ (\ddot{x}_e - \dot{J}\dot{\theta}) + \nabla_{\theta} E^T \dot{\theta}),\end{aligned}\quad (59)$$

where ∇_y denotes the gradient with respect to y , optimizing E using the GPM is equivalent to choosing k_{GPM} in

$$\underline{u} = k_{GPM} (S^T \nabla_{\dot{\theta}} E) \quad (60)$$

such that the term $\nabla_{\dot{\theta}} E^T S \underline{u}$ is either greater than zero (maximization) or less than zero (minimization). To determine the sign of k_{GPM} , substitute equation (60) into equation (59) to obtain

$$\dot{E} = (S^T \nabla_{\dot{\theta}} E)^T k_{GPM} (S^T \nabla_{\dot{\theta}} E) + (\nabla_{\dot{\theta}} E^T J^+ (\ddot{x}_e - \dot{J}\dot{\theta}) + \nabla_{\theta} E^T \dot{\theta}). \quad (61)$$

Since the first term in equation (61) is a quadratic, the control over the time rate of change of E is determined by making $k_{GPM} > 0$ (positive definite) for maximization and $k_{GPM} < 0$ (negative definite) for minimization. It should be noted that the choice of k_{GPM} does not affect the time rate of change due to the particular acceleration.

Another technique used in deriving redundancy control laws is the "minimization" method (MM) [42]. In the MM, the redundancies are considered as unknown parameters and are chosen by directly minimizing or solving a matrix equation. For example, if $E(\theta, \dot{\theta})$ is once again to be minimized, one choice for using

the MM is to choose the redundancies to drive \dot{E} given in equation (59) to some nominal or optimal value \dot{E}_0 by solving the following 1 x r matrix equation

$$\nabla_{\theta} E^T S \underline{u} + (\nabla_{\theta} E^T J^+ (\ddot{\mathbf{x}}_e - \dot{J}\dot{\theta}) + \nabla_{\theta} E^T \dot{\theta}) = \dot{E}_0 \quad (62)$$

to obtain

$$\underline{u}_{MM} = (\nabla_{\theta} E^T S)^+ (\dot{E}_0 - \nabla_{\theta} E^T J^+ (\ddot{\mathbf{x}}_e - \dot{J}\dot{\theta}) - \nabla_{\theta} E^T \dot{\theta}). \quad (63)$$

Compared to GPM, MM minimizes a function $E(\theta, \dot{\theta})$ by making the *entire* derivative, not just one term, less than zero. This requires substantially more information and calculations, but it has one advantage of some improvement in minimization. A comparison also shows that for the same optimization problem, MM is more flexible than GPM. For instance, in the above example, GPM has only the single solution given by equation (60) whereas MM could have minimized

$$L(\underline{u}) = \nabla_{\theta} E^T S \underline{u} - \dot{E}_0 \quad (64)$$

as a way of reducing E. However, if the 1 x r matrix $\nabla_{\theta} E^T S$ does not equal zero, the two methods are identical if

$$k_{GPM} = \frac{(\dot{E}_0 - \nabla_{\theta} E^T J^+ (\ddot{\mathbf{x}}_e - \dot{J}\dot{\theta}) - \nabla_{\theta} E^T \dot{\theta})}{\|\nabla_{\theta} E^T S\|^2}. \quad (65)$$

Although GPM and MM has been used with some success (see [23] for a list), several problems still exist. First, from the above equations, the performance of GPM

and MM is particularly sensitive to the choice of k_{GPM} and \dot{E}_0 . Unfortunately, there are no guidelines or theory for choosing these parameters, so their selection is limited to trial and error. Also, since both GPM and MM are local optimizations, their use does not imply that the final value of E is the global minimum or that the resulting joint accelerations lead to stable robot joint motion.

4.4 Control Algorithms

As a prelude to designing vibration damping control laws, it is illustrative first to compare the present control problem with previous vibration control problems. Based on the analysis presented in Section 4.2, assume an operator controls the end-effector by entering the desired end-effector accelerations relative to the base of the manipulator. Then, for convenience define $M_b(\underline{\theta}) \equiv I + \phi_b^T H_{pp}(\underline{\theta}) \phi_b$, $B(\underline{\theta}) \equiv H_{p0} S$, and $\underline{v} \equiv H_{p0} J^+(\underline{\ddot{x}} - J\dot{\underline{\theta}}) + \underline{N}_p$, so that equation (53) becomes

$$M_b(\underline{\theta})\ddot{\underline{p}} + \Lambda \underline{p} = -\Phi_b^T (B(\underline{\theta})\underline{u} + \underline{v}(\underline{p}, \dot{\underline{p}}, \underline{\theta}, \dot{\underline{\theta}}, \ddot{\underline{\theta}})). \quad (66)$$

Then, using standard state space definitions, equation (66) can be written in first order form as

$$\begin{aligned} \dot{\underline{z}} = \frac{d}{dt} \begin{bmatrix} \underline{p} \\ \dot{\underline{p}} \end{bmatrix} &= \begin{bmatrix} \mathbf{0} & I \\ -M_b(\underline{\theta})^{-1}\Lambda & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{p} \\ \dot{\underline{p}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ M_b(\underline{\theta})^{-1} \Phi_b^T B(\underline{\theta}) \end{bmatrix} \underline{u} + \begin{bmatrix} \mathbf{0} \\ M_b(\underline{\theta})^{-1} \Phi_b^T \underline{v} \end{bmatrix} \\ &= \bar{A}(t)\underline{x} + \bar{B}(t)\underline{u} + \bar{v}(t) \end{aligned} \quad (67)$$

where $\mathbf{0}$ and I are appropriately dimensioned zero and identity matrices.

First consider equation (66). Unlike many FEM based vibration damping problems [43-46], equation (66) is unusual in that it has a non-diagonal, time-varying mass matrix, a time-varying control influence matrix, and a non-linear "disturbing" force \underline{v} which is related to the control u through the integration of the robot joint accelerations. Because the mass matrix is time varying and is a function of the robot's payload, the eigenvalues of the flexible structure are not constant and can be quite different from the "open-loop" eigenvalues of the original flexible structure [47]. As a result, active and passive vibration damping schemes using tuned frequencies, e.g. [48,49] are particularly difficult to use. Also, since the control influence matrix is a function of the Jacobian's nullspace vectors, it is possible for B to equal zero. Thus, vibration control approaches that require controllability are not valid.

Next, consider the first order representation in equation (67). Since the joint angles are a function of the robot configuration which varies with time, $A(t)$ must be continuously updated. For large order flexible structures ($n_p \gg 1$), updating is very expensive because it requires inverting a fully populated $n_p \times n_p$ matrix. To overcome this problem, one could try to linearize the equations of motion about a reference trajectory, $\underline{\theta} \approx \underline{\theta}_{ref}$ and $\dot{\underline{\theta}} \approx \dot{\underline{\theta}}_{ref}$. However, since the trajectory is computed on-line, there is no reference trajectory. Therefore, a linear model cannot be developed. As a result, control laws developed for linear models such as linear quadratic regulators or pole placement [50] cannot be applied.

On account of the aforementioned complications, control for energy dissipation [51] is selected as the candidate method for choosing redundancy control laws. The

rationale for this choice is that since the principal cause of flexible vibration is energy imparted to the flexible structure by the robot particular joint accelerations, i.e. $\ddot{\underline{\theta}}_p$, it is logical to use the redundancies to dissipate any energy already in the base and reduce the energy imparted to the base. Other reasons for this choice are design simplicity and robustness [52]. In terms of choosing the manipulator redundancies, energy dissipation is also attractive because energy is a scalar function. Because of this, the previously discussed GPM and MM can be used without much difficulty.

Therefore, assume GPM or MM is to be used and that the velocity of the flexible base can be measured⁸. Then, the first step in deriving redundancy control laws is the selection of an appropriate optimization function. Using equation (50), this function is chosen as follows. From equation (50), total energy in the system is given by

$$E_{tot} = \frac{1}{2} \begin{bmatrix} \dot{\underline{\theta}}^T & \dot{\underline{p}}^T \end{bmatrix} \begin{bmatrix} M_\theta & H_{p\theta}^T \Phi_b \\ \Phi_b^T H_{p\theta} & I + \Phi_b^T H_{pp} \Phi_b \end{bmatrix} \begin{bmatrix} \dot{\underline{\theta}} \\ \dot{\underline{p}} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \underline{\theta}^T & \underline{p}^T \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \Lambda \end{bmatrix} \begin{bmatrix} \underline{\theta} \\ \underline{p} \end{bmatrix} \quad (68)$$

which also equals

$$E_{tot} = \frac{1}{2} \dot{\underline{\theta}}^T M_\theta \dot{\underline{\theta}} + \dot{\underline{\theta}}^T H_{p\theta}^T \Phi_b \dot{\underline{p}} + \frac{1}{2} \dot{\underline{p}}^T (I + \Phi_b^T H_{pp} \Phi_b) \dot{\underline{p}} + \frac{1}{2} \underline{p}^T \Lambda \underline{p}. \quad (69)$$

Inspection of equation (69) shows that the first term represents the kinetic energy due solely to the motion of the robot, the second term represents the kinetic energy due to

⁸ If velocity measurements are not available, energy dissipation can still be accomplished using acceleration measurements and a virtual controller. See Appendix C for more details.

the coupled motion of the flexible base and the robot, while the last two terms represent the kinetic and potential energy of the flexible base due to the motion of the flexible base. Therefore, to use MM or GPM to locally reduce flexible vibrations via energy dissipation, the appropriate optimization function is

$$E_f = \frac{1}{2} \dot{\mathbf{p}}^T (I + \Phi_b^T H_{pp}(\Theta) \Phi_b) \dot{\mathbf{p}} + \frac{1}{2} \mathbf{p}^T \Lambda \mathbf{p}. \quad (70)$$

Proceeding as before, the objective in either GPM or MM is to make the time derivative of E_f given by

$$\dot{E}_f = \dot{\mathbf{p}}^T [(I + \Phi_b^T H_{pp}(\Theta) \Phi_b) \ddot{\mathbf{p}} + \Lambda \dot{\mathbf{p}}] + \frac{1}{2} \dot{\mathbf{p}}^T \Phi_b^T \dot{H}_{pp} \Phi_b \dot{\mathbf{p}} \quad (71)$$

as negative as possible. Since the term in brackets is equal to the left hand side of equation (51), and the end-effector accelerations are specified with respect to the base of the manipulator, and the velocity at the mounting point between the flexible structure and the base of the robot is $\mathbf{V}_b = \Phi_b \dot{\mathbf{p}}$, an alternate expression for \dot{E}_f is

$$\dot{E}_f = -\mathbf{V}_b^T [H_{p\theta} (S\mathbf{u} + \mathbf{J}^+ (\ddot{\mathbf{x}}_e - \mathbf{J}\dot{\Theta})) + \mathbf{N}_p] + \frac{1}{2} \mathbf{V}_b^T \dot{H}_{pp}(\Theta) \mathbf{V}_b - \dot{\mathbf{p}}^T D \dot{\mathbf{p}}. \quad (72)$$

Furthermore, since the time derivative of the $m \times m$ matrix H_{pp} can be written as

$$\begin{aligned} \dot{H}_{pp}(\Theta) &= \frac{d}{dt} [\mathbf{h}_{pp_1}(\Theta) \mid \mathbf{h}_{pp_2}(\Theta) \mid \dots \mid \mathbf{h}_{pp_m}(\Theta)] \\ &= \left[\nabla_{\Theta} \mathbf{h}_{pp_1} \dot{\Theta} \mid \nabla_{\Theta} \mathbf{h}_{pp_2} \dot{\Theta} \mid \dots \mid \nabla_{\Theta} \mathbf{h}_{pp_m} \dot{\Theta} \right] \end{aligned} \quad (73)$$

where the j -th sensitivity for the i -th column of \dot{H}_{pp} is given by

$$(\nabla_{\underline{\theta}} \underline{h}_{pp_i})_j = \frac{\partial \underline{h}_{pp_i}(\underline{\theta})}{\partial \theta_j}, \quad (74)$$

the time rate of change of flexible energy due to changes in H_{pp} can be expressed by

$$\begin{aligned} \underline{V}_b^T \dot{\underline{H}}_{pp}(\underline{\theta}) \underline{V}_b &= \underline{V}_b^T \left[\nabla_{\underline{\theta}} \underline{h}_{pp_1} \dot{\underline{\theta}} \mid \nabla_{\underline{\theta}} \underline{h}_{pp_2} \dot{\underline{\theta}} \mid \dots \mid \nabla_{\underline{\theta}} \underline{h}_{pp_m} \dot{\underline{\theta}} \right] \begin{bmatrix} \underline{V}_{b,1} \\ \underline{V}_{b,2} \\ \vdots \\ \underline{V}_{b,m} \end{bmatrix} \\ &= \underline{V}_b^T (\underline{V}_{b,1} \cdot \nabla_{\underline{\theta}} \underline{h}_{pp_1} + \underline{V}_{b,2} \cdot \nabla_{\underline{\theta}} \underline{h}_{pp_2} + \dots + \underline{V}_{b,m} \cdot \nabla_{\underline{\theta}} \underline{h}_{pp_m}) \dot{\underline{\theta}} \\ &\equiv \underline{V}_b^T \underline{G} \dot{\underline{\theta}} \end{aligned} \quad (75)$$

where G is an $m \times n$ matrix of "generalized" sensitivities. As a result, the time rate of change of the desired MM/GPM optimization function is

$$\begin{aligned} \dot{\underline{E}}_f &= -\underline{V}_b^T \left[\underline{H}_{p\theta} (\underline{S}\underline{u} + \underline{J}^+ (\ddot{\underline{x}}_e - \underline{J}\dot{\underline{\theta}})) + \underline{N}_p - \frac{1}{2} \underline{G}\dot{\underline{\theta}} \right] - \dot{\underline{p}}^T \underline{D}\dot{\underline{p}} \\ &= -\underline{V}_b^T \left[\underline{B}\underline{u} + \underline{H}_{p\theta} \underline{J}^+ (\ddot{\underline{x}}_e - \underline{J}\dot{\underline{\theta}}) + \underline{N}_p - \frac{1}{2} \underline{G}\dot{\underline{\theta}} \right] - \dot{\underline{p}}^T \underline{D}\dot{\underline{p}}. \end{aligned} \quad (76)$$

It must be pointed out that the only free parameters in equation (76) are the redundancies contained in \underline{u} . Using the GPM and the MM, several choices for \underline{u} will now be given.

First, in the GPM, \underline{u} is chosen to ensure that $-\underline{V}_b^T \underline{B}\underline{u}$ is non-positive. This is accomplished by computing the $r \times 1$ vector \underline{u} according to

$$\underline{u}_{GPM} = k_{GPM} \underline{B}^T \underline{V}_b \quad (77)$$

where k_{GPM} is any symmetric $r \times r$ matrix. However, for simplicity, typically k_{GPM} is

chosen as a scalar whose sign is determined by substituting equation (77) into equation (76). Making this substitution yields

$$\dot{E}_f(\underline{u}_{GPM}) = -k_{GPM}(\underline{V}_b^T \mathbf{B} \mathbf{B}^T \underline{V}_b) - \underline{V}_b^T \left[H_{p\theta} \mathbf{J}^+(\ddot{\mathbf{x}}_e - \dot{\mathbf{J}}\dot{\mathbf{Q}}) + \underline{N}_p - \frac{1}{2} G\dot{\mathbf{Q}} \right] - \dot{\mathbf{p}}^T D \dot{\mathbf{p}}. \quad (78)$$

Since $\underline{V}_b^T \mathbf{B} \mathbf{B}^T \underline{V}_b$ is non-negative, flexible energy is dissipated when k_{GPM} is greater than zero. As with all GPM optimization solutions, the present solution does not ensure that $\dot{E}_f < 0$. Instead, equation (77) ensures that whenever \underline{V}_b does not equal zero and \mathbf{B} is not orthogonal to \underline{V}_b , the instantaneous time rate of change of flexible energy will be decreased by the amount $-k_{GPM} \underline{V}_b^T \mathbf{B} \mathbf{B}^T \underline{V}_b$.

The second choice \underline{u} is to use the MM and feedback linearization to cancel the non-linear terms in equation (76). Since there is some freedom in the selection of the minimization function, consider using velocity feedback [52] to make the term in the brackets of equation (76) equal to $k_{MMI} \underline{V}_b$, i.e.

$$\mathbf{B}\underline{u} + H_{p\theta} \mathbf{J}^+(\ddot{\mathbf{x}}_e - \dot{\mathbf{J}}\dot{\mathbf{Q}}) + \underline{N}_p - \frac{1}{2} G\dot{\mathbf{Q}} = k_{MMI} \underline{V}_b. \quad (79)$$

where k_{MMI} is a symmetric, positive definite $m \times m$ matrix. Under this assumption, the resulting time rate of change of flexible energy would be

$$\dot{E}_f(\underline{u}_{MMI}) = -\underline{V}_b^T k_{MMI} \underline{V}_b \quad (80)$$

which is non-positive as long as k_{MMI} is positive, semi-definite. However, because \mathbf{B} is configuration dependent and $m \times r$, \mathbf{B} may not be square or full rank for all configurations. Therefore, in general, it is not possible to exactly solve equation (79)

for all time. As a result, the best that can be done is to choose the redundancies to minimize the error between $k_{MMI}\underline{V}_b$ and the left hand side of equation (79) by minimizing $L(\underline{u}_{MMI})$ defined by

$$L(\underline{u}_{MMI}) = \left\| \underline{B}\underline{u} + H_{p\theta} \mathbf{J}^+ (\ddot{\underline{x}}_e - \dot{\mathbf{J}}\dot{\underline{\theta}}) + \underline{N}_p - \frac{1}{2} \underline{G}\dot{\underline{\theta}} - k_{MMI}\underline{V}_b \right\|^2 \quad (81)$$

to obtain

$$\underline{u}_{MMI} = \underline{B}^+ \left(k_{MMI}\underline{V}_b - H_{p\theta} \mathbf{J}^+ (\ddot{\underline{x}}_e - \dot{\mathbf{J}}\dot{\underline{\theta}}) - \underline{N}_p + \frac{1}{2} \underline{G}\dot{\underline{\theta}} \right) \quad (82)$$

Because of the least squares property of the pseudoinverse, \underline{u}_{MMI} in equation (82) represents the best the redundancies can do to use feedback linearization to achieve energy dissipation as defined by equation (80).

Since \underline{u}_{MMI} is generally not an exact solution, \underline{u}_{MMI} must be shown to provide energy dissipation. Mathematically, this proof is accomplished by examining the closed-loop, time rate of change of flexible energy. Substituting \underline{u}_{MMI} back into equation (76) yields

$$\begin{aligned} \dot{E}_f(\underline{u}_{MMI}) &= -\underline{V}_b^T \left[\underline{B}\underline{u}_{MMI} + \mathbf{J}^+ (\ddot{\underline{x}}_e - \dot{\mathbf{J}}\dot{\underline{\theta}}) + \underline{N}_p - \frac{1}{2} \underline{G}\dot{\underline{\theta}} \right] - \dot{\underline{p}}^T \underline{D}\dot{\underline{p}} \\ &= -\underline{V}_b^T (\underline{B}\underline{B}^+ k_{MMI}) \underline{V}_b - \underline{V}_b^T (\underline{I} - \underline{B}\underline{B}^+) \left(H_{p\theta} \mathbf{J}^+ (\ddot{\underline{x}}_e - \dot{\mathbf{J}}\dot{\underline{\theta}}) + \underline{N}_p - \frac{1}{2} \underline{G}\dot{\underline{\theta}} \right) - \dot{\underline{p}}^T \underline{D}\dot{\underline{p}}. \end{aligned} \quad (83)$$

Inspection of equation (83) shows that irrespective of natural damping in the flexible base, energy is dissipated if

$$\underline{V}_b^T \mathbf{B} \mathbf{B}^+ k_{MMI} \underline{V}_b > 0. \quad (84)$$

To determine when this condition is true, the singular value decomposition can be used to decompose \mathbf{B} , \mathbf{B}^+ , and $\mathbf{B}\mathbf{B}^+$ into

$$\mathbf{B} = \mathbf{H}_{p\theta} \mathbf{S} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 \end{bmatrix} \begin{bmatrix} \sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}_1^T \\ \mathbf{U}_2^T \end{bmatrix} = \mathbf{R}_1 \sigma \mathbf{U}_1^T, \quad (85)$$

$$\mathbf{B}^+ = \mathbf{U}_1 \sigma^{-1} \mathbf{R}_1^T, \quad (86)$$

and

$$\mathbf{B}\mathbf{B}^+ = \mathbf{R}_1 \mathbf{R}_1^T = \mathbf{I}_m - \mathbf{R}_2 \mathbf{R}_2^T \quad (87)$$

where if \mathfrak{R}_B is the rank of \mathbf{B} , then \mathbf{R}_1 , \mathbf{R}_2 , \mathbf{U}_1 , and \mathbf{U}_2 are respectively, $m \times \mathfrak{R}_B$, $m \times m - \mathfrak{R}_B$, $r \times \mathfrak{R}_B$, and $r \times r - \mathfrak{R}_B$ matrices. Depending on the number of redundancies and the rank of \mathbf{B} , there are two cases to consider.

Case 1. If there are more redundancies than task space variables ($r > m$) and \mathbf{B} is full rank, then $\mathfrak{R}_B = m$ and \mathbf{R}_2 is zero. As a result, $\mathbf{B}\mathbf{B}^+ = \mathbf{I}$ and the closed-loop change in energy becomes

$$\dot{\mathbf{E}}_f = -\underline{V}_b^T k_{MMI} \underline{V}_b. \quad (88)$$

Thus, equation (80) is identically true. Similarly, if \mathbf{B} is full rank and $r = m$, \mathbf{B} is a square matrix and once again equation (80) is satisfied exactly.

Case 2. If there are more task space variables than redundancies ($m > r$) and B is full rank or if B is not full rank, R_2 is non-zero. This leads to

$$\dot{E}_F = -\underline{V}_b^T (R_1 R_1^T k_{MM1}) \underline{V}_b - \underline{V}_b^T (I - BB^+) \left(H_{p\theta} J^+ (\ddot{\underline{x}}_e - \dot{J}\dot{\theta}) + \underline{N}_p - \frac{1}{2} G\dot{\theta} \right) \quad (89)$$

Inspection of equation (89) shows that energy is always dissipated if k_{MM1} is chosen such that $R_1 R_1^T k_{MM1}$ is positive definite. Since R_1 is time varying, energy dissipation requires that k_{MM1} be a positive scalar. Furthermore, inspection also shows that because $(I - BB^+) (H_{p\theta} J^+ (\ddot{\underline{x}}_e - \dot{J}\dot{\theta}) + \underline{N}_p - 0.5G\dot{\theta})$ is the least squares result obtained from solving

$$B\underline{u} + (H_{p\theta} J^+ (\ddot{\underline{x}}_e - \dot{J}\dot{\theta}) + \underline{N}_p - \frac{1}{2} G\dot{\theta}) = \mathbf{0}, \quad (90)$$

the effects of the particular acceleration, Coriolis torques, and generalized inertia sensitivity matrix are attenuated. Thus, depending on the number of redundancies and the number of task space variables, \underline{u}_{MM1} either produces total energy dissipation or adds some damping and reduces the effects of the non-linear terms.

However, in actual implementation, all of the information needed to compute \underline{u}_{MM1} may not be available. For example, it may not be possible to compute G or \underline{N}_p in real time. Therefore, instead of using equation (82), a suboptimal feedback linearization MM solution can be obtained by minimizing

$$L(\underline{u}) = \left\| \mathbf{B}\underline{u} + \mathbf{H}_{p0}\mathbf{J}^+(\ddot{\mathbf{x}}_e - \dot{\mathbf{J}}\dot{\boldsymbol{\theta}}) + c_1\mathbf{N}_p - c_2\frac{\mathbf{G}\dot{\boldsymbol{\theta}}}{2} - c_3k_{MMI}\mathbf{V}_b \right\|^2 \quad (91)$$

where depending on the available information, $c_i = 1$ or 0 . Due to the lack of information, this formulation is equivalent to trying to cancel one or more of the terms in $\dot{\mathbf{E}}_f$. Two possible scenarios will now be given.

Feedback Linearization MM Scenario 1. Reduce the effects of the Particular Acceleration and Dissipate Energy (RPA-DE)

If velocity measurements are available, but the Coriolis torques and the generalized inertia sensitivity matrix cannot be computed, $c_1=0$, $c_2=0$, and $c_3=1$. For these conditions, the redundancies attenuate the effects of the particular acceleration by making the total robot joint accelerations as orthogonal as possible to the mass coupling matrix \mathbf{H}_{p0} while also adding an energy dissipating velocity feedback term. The solution for \underline{u} is given by

$$\underline{u}_{RPA-DE} = \mathbf{B}^+(k_{MMI}\mathbf{V}_b - \mathbf{H}_{p0}\mathbf{J}^+(\ddot{\mathbf{x}}_e - \dot{\mathbf{J}}\dot{\boldsymbol{\theta}})). \quad (92)$$

Feedback Linearization MM Scenario 2. Reduce Applied Wrench and Dissipate Energy (RW-DE)

The second scenario is when all of the required information is available except the generalized inertia sensitivity matrix \mathbf{G} , i.e. ($c_1=1$; $c_2=0$, and $c_3=1$). As seen in equation (51), RW-DE chooses the redundancies to reduce the total applied Cartesian wrench on the flexible structure, while increasing the system damping level. Using

equation (82), the corresponding redundancy solution is

$$\underline{u}_{RW-DE} = \mathbf{B}^+ (k_{MMI} \underline{V}_b - \mathbf{H}_{p\theta} \mathbf{J}^+ (\ddot{\underline{x}}_e - \dot{\mathbf{J}}\dot{\underline{\theta}}) - \underline{N}_p) \quad (93)$$

4.4 Performance and Stability Analysis

In Section 4.3, the GPM and the MM were used to develop two new solutions, \underline{u}_{GPM} and \underline{u}_{MMI} , for the manipulator redundancies that locally reduce the energy in the flexible base by adding damping. Because all of the information need to compute \underline{u}_{MMI} may not be accessible, two additional MM solutions, \underline{u}_{RPA-DE} and \underline{u}_{RW-DE} , were developed based on the available information. For convenience, these solutions are listed in Table I.

Because velocity feedback is used, all of the solutions shown in Table I require choosing a feedback gain, either k_{GPM} or k_{MMI} . Due to non-linearities and the local nature of the problem, there is no general theory or method for choosing these gains that guarantees global stability and acceptable performance. However, if the equations of motion for the combined rigid robot-flexible base system are written as

$$\begin{bmatrix} I & 0 \\ 0 & I + \phi_b^T \mathbf{H}_{pp} \phi_b \end{bmatrix} \begin{bmatrix} \ddot{\underline{\theta}} \\ \ddot{\underline{p}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} \dot{\underline{\theta}} \\ \dot{\underline{p}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \Lambda \end{bmatrix} \begin{bmatrix} \dot{\underline{\theta}} \\ \dot{\underline{p}} \end{bmatrix} = \begin{bmatrix} I \\ -\phi_b^T \mathbf{H}_{p\theta} \end{bmatrix} \mathbf{J}^+ (\ddot{\underline{x}}_e - \dot{\mathbf{J}}\dot{\underline{\theta}}) + \begin{bmatrix} S \\ -\phi_b^T \mathbf{B} \end{bmatrix} \underline{u} + \begin{bmatrix} 0 \\ -\phi_b^T \underline{N}_p \end{bmatrix}, \quad (94)$$

energy dissipation concepts can be used to facilitate choosing k_{GPM} or k_{MMI} . Before continuing, it should be pointed out that the reason for the apparent difference between

equation (94) and equation (50) is that since a suitable torque control is assumed to exist (see Section 4.1), the dynamic equations for the motion of the robot can be replaced by the kinematic relations given in equation (17).

Table I. Flexible Base Energy Dissipation Solutions.

Optimization Method	Title	Redundancy Solution
		$\ddot{\theta}_p = J^+(\ddot{x}_e - \dot{J}\dot{\theta})$
Gradient Projection Method	GPM	$\underline{u}_{GPM} = k_{GPM} B^T \underline{V}_b$
Minimization Method	Complete Feedback Linearization	$\underline{u}_{MMI} = B^+ \left(k_{MMI} \underline{V}_b - H_{p0} \ddot{\theta}_p - \underline{N}_p + \frac{1}{2} G \dot{\theta} \right)$
	Reduce Particular Acceleration-Dissipate Energy	$\underline{u}_{RPA-DE} = B^+ \left(k_{MMI} \underline{V}_b - H_{p0} \ddot{\theta}_p \right)$
	Reduce Wrench-Dissipate Energy	$\underline{u}_{RW-DE} = B^+ \left(k_{MMI} \underline{V}_b - H_{p0} \ddot{\theta}_p - \underline{N}_p \right)$

To systematically choose k_{GPM} and k_{MM1} , consider its effects on the combined system when the $m \times r$ control influence matrix B is full rank with $m > r$ ⁹. Under this assumption, the pseudo-inverse of B is given by $B^+ = (B^T B)^{-1} B^T$. Inspection of Table I then shows that each of the four redundancy solutions can be decomposed into a term that is proportional to the modal velocities and another term dependent on the particular acceleration. Here, the emphasis is on the velocity dependent term, \underline{u}_p , which can be written as

$$\underline{u}_p = k_p W_p B^T \phi_b \dot{\underline{p}}. \quad (95)$$

where $k_p = k_{\text{GPM}}$ (an $r \times r$ matrix) and $W_p = I_r$ for GPM and $k_p = k_{\text{MM1}}$ (a scalar) and $W_p = (B^T B)^{-1}$ for MM1, RPA-DE, and RW-DE. If this term is substituted into equation (94), the resulting closed-loop damping matrix is

$$D_{cl} = \begin{bmatrix} \mathbf{0} & -k_p S W_p B^T \phi_b \\ \mathbf{0} & D + k_p \phi_b^T B B^T \phi_b \end{bmatrix}. \quad (96)$$

Since this matrix is not symmetric, positive semi-definite, energy is not dissipated in the combined system of equations. Thus, even though the redundancy solutions listed in Table I dissipate some energy in the flexible base, they do not dissipate energy in the robot. As a result, using the redundancies to only dissipate energy in the flexible base can cause unstable robot joint motion.

Mathematically, this is due to the fact that calculating \underline{u} to optimize a function of the flexible base does not guarantee that the subsequent robot joint motion resulting

⁹ If B is not full rank, equations (85), (86), and (87) can be used to derive similar results.

from

$$\ddot{\underline{\theta}} + \mathbf{J}^+ \dot{\mathbf{J}} \dot{\underline{\theta}} = \mathbf{J}^+ \ddot{\underline{x}}_e + \mathbf{S} \underline{u} \quad (97)$$

will be stable or remain within the workspace of the manipulator. This type of behavior is common in locally resolving the redundancies. For example, in the local torque optimization problem, other researchers [54-58] have shown that choosing \underline{u} to locally optimize a torque performance function often causes global stability problems and large robot joint velocities. Due to the similarities between torque optimization and flexible base vibration damping, this implies that choosing \underline{u} to minimize flexible vibrations at time t_1 may cause unacceptably high joint velocities at a later time $t_2 > t_1$ that lead to unstable robot motion.

From a physical standpoint, the unstable motion is the result of the redundancies locally affecting the robot joint accelerations at t_1 , which affects the velocities at later times. Since the robot velocities affect the particular acceleration via $-\mathbf{J}^+ \dot{\mathbf{J}} \dot{\underline{\theta}}$, the control at later times has to counteract the (possibly negative) effects of \underline{u} from previous times. Depending on the desired end-effector trajectory and the redundancy optimization function, this cycle can cause uncontrolled joint velocity buildups that cause instabilities. It should be pointed out that this phenomenon is true regardless of whether or not B satisfies the conditions listed in Case 1 and Case 2 or if all of the required information is available, i.e., this is a fundamental characteristic in local optimizations.

Since uncontrolled or unstable motion is accompanied by large velocity

buildups, in order to improve system stability, \underline{u} must be chosen to simultaneously minimize a function of the flexible base and the robot joint velocities. However, before deriving such a function, it is useful to qualitatively consider the effects on \dot{E}_f of choosing \underline{u} to reduce the robot joint velocities. First, note that the \dot{E}_f can be written as

$$\dot{E}_f = -V_b^T (B\underline{u} + H_{p\theta} J^+ \ddot{\underline{x}}_e) - V_b^T \left[N_p(\dot{\underline{p}}, \dot{\underline{\theta}}) + \left(\frac{1}{2} G - H_{p\theta} J^+ \dot{J} \right) \dot{\underline{\theta}} \right] - \dot{\underline{p}}^T D \dot{\underline{p}}. \quad (98)$$

Then if small angular velocities are assumed in the flexible base, in the limit as the robot joint velocities go to zero, the term in brackets and \underline{u} go to zero. Thus, if the end-effector acceleration is treated as an external disturbance and if enough of \underline{u} is chosen to reduce robot joint velocities, the energy in the flexible base will be bounded. Indeed, if \underline{u} is chosen to only reduce the robot joint velocities, when the robot stops moving, equation (98) shows that due to the inherent damping, \dot{E}_f is less than zero.

Now, it remains to compute a \underline{u} that damps both the flexible base and robot. To accomplish this, an approach similar to Ma, Hirose, and Nenchev [57] in the local torque optimization problem can be taken. Instead of minimizing a function of the robot torque norm, they proposed minimizing a "damped-squared" cost function of the norm of the robot joint accelerations and the robot joint torque norm, i.e.,

$$\underline{u} \leftarrow \min_{\underline{\ddot{\theta}}} \left[(1-\gamma) \|\underline{\tau} - \underline{\tau}_{des}\|^2 + \gamma \|\underline{\ddot{\theta}}\|^2 \right] \quad (99)$$

where γ is a weighting or balancing factor and $\underline{\tau}_{des}$ is the desired or nominal torque

value. However, in order to eliminate the problems associated with adding quantities with different units (e.g. squared accelerations and squared torques), in the current problem, composite control is used to partition \underline{u} into the weighted sum of individual \underline{u} 's independently calculated to Stabilize Robot Motion \underline{u}_{SRM} and to Control Base Motion \underline{u}_{CBM} (e.g. GPM or RPA-DE) , i.e.,

$$\underline{u} = \gamma \underline{u}_{SRM} + (1-\gamma)\underline{u}_{CBM} . \quad (100)$$

To compute a robot stabilizing control, recall that based on the work done in torque optimization, the underlying cause of robot instability is uncontrolled robot joint velocity buildups due to the influence of the control at time t_1 on the robot joint velocities at some later time t_2 . Therefore, to prevent instabilities, \underline{u}_{srm} should be chosen to minimize some function of *future* robot joint velocities. If the complete end-effector trajectory is known, the ideal function would be to minimize *all* future robot joint velocities such as was done in reference [55]:

$$L = \int_{t_0}^{t_f} \dot{\underline{q}}^T W \dot{\underline{q}} dt \quad (101)$$

where W is a symmetric, positive definite (possibly time varying) velocity weighting matrix. But since the entire trajectory is unknown in local optimization problems, only the *next* velocity can be minimized. As a result, \underline{u}_{srm} must be chosen to minimize some approximation to L . Although many possible approximations to L exist, this work uses the weighted, one step ahead robot joint velocity norm defined by where the " $()_k$ " implies evaluation at time t_k and the one-step ahead robot joint

$$L_k = \dot{\underline{\theta}}_{k+1}^T W \dot{\underline{\theta}}_{k+1}. \quad (102)$$

velocity is defined by the first order discrete approximation

$$\dot{\underline{\theta}}_{k+1} = \dot{\underline{\theta}}_k + (S_k \underline{u} + J_k^+ (\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}})|_k) \Delta t \quad (103)$$

for a given sampling time Δt . Therefore, a robot stabilizing control can be calculated by substituting equation (103) into equation (102) and minimizing to obtain

$$\begin{aligned} \underline{u}_{SRM} &= -(W^{1/2} S)^+ W^{1/2} (\dot{\underline{\theta}}/\Delta t + J^+ (\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}})) \\ &= -(S^T W S)^{-1} S^T W (\dot{\underline{\theta}}/\Delta t + J^+ (\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}})) \end{aligned} \quad (104)$$

where for simplicity, the subscripts have been left out.

To verify that the above solution is in fact stabilizing, consider its effects on the time rate of change of the weighted velocity norm T_w ,

$$\dot{T}_w = \frac{d}{dt} \left(\frac{1}{2} \dot{\underline{\theta}}^T W \dot{\underline{\theta}} \right) = \dot{\underline{\theta}}^T W (S \underline{u} + J^+ (\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}})) + \frac{1}{2} \dot{\underline{\theta}}^T \dot{W} \dot{\underline{\theta}}. \quad (105)$$

Substituting \underline{u}_{SRM} into equation (105) and simplifying yields

$$\begin{aligned} \dot{T}_w &= -[S^T W \dot{\underline{\theta}}]^T \frac{[S^T W S]^{-1}}{\Delta t} [S^T W \dot{\underline{\theta}}] + \frac{1}{2} \dot{\underline{\theta}}^T \dot{W} \dot{\underline{\theta}} \\ &\quad + \dot{\underline{\theta}}^T [I - S(S^T W S)^{-1} S^T] W J^+ (\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}}). \end{aligned} \quad (106)$$

Since $S^T W S$ and its inverse are positive definite, the first term in equation (106) is always negative. Therefore, \underline{u}_{SRM} adds a kinetic energy dissipative torque to the robot joints which helps to stabilize the joint motion of the robot.

Using \underline{u}_{SRM} and equation (100), k_{GPM} and k_{MMI} can now be specified. To accomplish this, first re-write the system of equations as

$$\begin{bmatrix} W_\theta & 0 \\ 0 & I + \Phi_b^T H_{pp} \Phi_b \end{bmatrix} \begin{bmatrix} \ddot{\underline{x}} \\ \ddot{\underline{p}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{p}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \Lambda \end{bmatrix} \begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{p}} \end{bmatrix} = \begin{bmatrix} W_\theta \\ -\Phi_b^T H_{p\theta} \end{bmatrix} J^+ (\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}}) + \begin{bmatrix} W_\theta S \\ -\Phi_b^T B \end{bmatrix} \underline{u} + \begin{bmatrix} 0 \\ -\Phi_b^T N_p \end{bmatrix}, \quad (107)$$

where W_θ is a to-be-determined symmetric, positive definite inertia matrix not necessarily equal to the inertia matrix M_θ of the robot. For GPM, equations (77), (100), and (104) yield the following partitioned control redundancy solution:

$$\underline{u}_{GPM} = (1-\gamma)k_{gpm} B^T \Phi_b \dot{\underline{p}} - \gamma (S^T W S)^{-1} S^T W (\dot{\underline{\theta}}/\Delta t + J^T (\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}})). \quad (108)$$

Using this relation, the closed-loop force generated by the partitioned control redundancy solution is given by

$$\begin{bmatrix} W_\theta S \\ -\Phi_b^T B \end{bmatrix} \underline{u}_{GPM} = - \begin{bmatrix} \gamma W_\theta S (S^T W S)^{-1} S^T W / \Delta t & -(1-\gamma) W_\theta S k_{gpm} B^T \Phi_b \\ -\gamma \Phi_b^T B (S^T W S)^{-1} S^T W / \Delta t & (1-\gamma) \Phi_b^T B k_{gpm} B^T \Phi_b \end{bmatrix} \begin{bmatrix} \dot{\underline{\theta}} \\ \dot{\underline{p}} \end{bmatrix} - \gamma \begin{bmatrix} W_\theta S \\ -\Phi_b^T B \end{bmatrix} (S^T W S)^{-1} S^T W J^+ (\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}}) \quad (109)$$

which for notational purposes can also be written as

$$\begin{bmatrix} W_\theta S \\ -\Phi_b^T B \end{bmatrix} \underline{u}_{GPM} = -D_{GPM} \begin{bmatrix} \dot{\underline{\theta}} \\ \dot{\underline{p}} \end{bmatrix} - \gamma \begin{bmatrix} W_\theta S \\ -\Phi_b^T B \end{bmatrix} S^T W J^+ (\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}}). \quad (110)$$

For energy dissipation, D_{GPM} must be symmetric and positive semi-definite. Inspection of equation (109) shows that these conditions are met if (1) the one step-ahead weighting matrix W equals the fictitious inertia matrix W_θ and (2) the velocity

feedback gain is calculated by

$$k_{GPM} = \frac{\gamma}{(1-\gamma)\Delta t} [(S^T W S)^{-1}]^T, 0 < \gamma < 1. \quad (111)$$

Using equation (111), the closed-loop damping matrix is

$$D_{GPM} = \frac{\gamma}{\Delta t} \begin{bmatrix} WS(S^T WS)^{-1}S^T W & -WS(S^T WS)^{-1}B^T\phi_b \\ -\phi_b^T B(S^T WS)^{-1}S^T W & \phi_b^T B(S^T WS)^{-1}B^T\phi_b \end{bmatrix}. \quad (112)$$

(For a proof of positive definiteness, see Appendix D.) Inspection of the equations (111) and (112) shows that since S and (possibly W) are configuration dependent, enforcing energy dissipation causes k_{GPM} to be time-varying and makes the closed-loop damping matrix linearly proportional to the weighting factor γ . Furthermore, although equation (111) gives an analytic expression for k_{GPM} , the formula was derived at the expense of increasing the number of unknown parameters from one (k_{GPM}) to two (W and γ). However, since γ must lie between 0 and 1, and setting W equal to the inertia matrix of the robot dissipates robot kinetic energy, choosing W and γ is somewhat simpler than choosing k_{GPM} . That is, if closed-loop damping is not considered, the range of k_{GPM} is $[0, \infty]$. On the other hand, if closed-loop damping is considered and if W is the inertia matrix of the robot, the range of the free parameter is reduced to $[0,1]$.

Equation (111) gives the gain given the weighting between reducing flexible velocities and stabilizing the robot. A similar expression can be derived to compute the weighting given a value for the gain. Indeed, if k_{GPM} is a scalar, the relation between γ and k_{GPM} is given by

$$\gamma = \frac{k_{GPM} \Delta t S^T W S}{k_{GPM} \Delta t S^T W S + 1}. \quad (113)$$

The preceding equations were developed for GPM. For MM based solutions, a similar analysis can be done (see Appendix E) to obtain

$$k_{MMI} = \frac{\gamma}{(1-\gamma) \Delta t} \frac{B^T B}{S^T W S}, \quad (114)$$

$$\gamma = \frac{k_{MMI} \Delta t S^T W S}{k_{MMI} \Delta t S^T W S + B^T B}, \quad (115)$$

and

$$D_{MMI} = \frac{\gamma}{\Delta t} \begin{bmatrix} WS(S^T WS)^{-1} S^T W & -WS B^T \phi_b \\ -\phi_b^T B(S^T WS)^{-1} S^T W & \phi_b^T B(S^T WS)^{-1} B^T \phi_b \end{bmatrix} \quad (116)$$

where it must be remembered that k_{mm1} is a scalar (See page 46). Compared to equation (112), inspection of equation (116) shows enforcing energy dissipation yields the same closed-loop damping matrix for MM1 and GPM. Therefore, the primary difference between the gradient projection solution (GPM) and the minimization solutions (MM1, RPA-DE, and RW-DE) is the inclusion of terms responsible for reducing the effects of the torque due to the particular acceleration and the Coriolis torque. Indeed, using equations (60), (104), and (111), the weighted GPM solution is

$$\underline{u}_{GPM} = \frac{\gamma}{\Delta t} (S^T W S)^{-1} (B^T \underline{V}_b - S^T W \dot{\underline{\theta}}) - \gamma (S^T W S)^{-1} S^T W J^+ (\ddot{\underline{x}}_e - \dot{J} \dot{\underline{\theta}}), \quad (117)$$

while from equations (82), (104), and (114), the weighted MM solutions are

$$\underline{u}_{MMI} = \underline{u}_{gpm} - (1 - \gamma)B^+(H_{p\theta}J^+(\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}}) + \underline{N}_p - G\dot{\underline{\theta}}/2), \quad (118)$$

$$\underline{u}_{RPA-DE} = \underline{u}_{gpm} - (1 - \gamma)B^+H_{p\theta}J^+(\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}}), \quad (119)$$

and

$$\underline{u}_{RW-DE} = \underline{u}_{gpm} - (1 - \gamma)B^+(H_{p\theta}J^+(\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}}) + \underline{N}_p). \quad (120)$$

4.5 Summary

Due to the large number of equations, it is worthwhile to review and summarize the results presented so far. Using the coupled equations of motion for a rigid robot on a flexible base, this chapter derived energy dissipation control algorithms for the kinematic redundancies in a kinematically redundant robot. These algorithms are previously un-published research results and represent new uses for kinematic redundancies. They were calculated as follows.

First, it was argued that because an operator would most likely be entering the desired trajectory relative to the base of the manipulator, fixed-base instead of moving base Jacobian relations should be used to resolve the redundancies. Next, to simplify the problem, a suitable torque control law was assumed so that the dynamic equation for the robot could be replaced by the kinematic relations between the acceleration of the end-effector and the robot joints. This reduced the control problem to optimally determining the redundancies to reduce flexible vibrations.

Next, due to the non-linear and time-varying nature of the problem, two common redundancy control methods, the minimization and the gradient projection

method, were used to derive local, energy dissipation algorithms for the flexible base (equations (60), (82), (92), and (93)). Next, it was shown that although these algorithms locally increase the damping in the flexible base, only optimizing flexible base motion can lead to high joint velocities and unstable robot joint motion. As a result, partitioned or weighted composite control was introduced as one method to simultaneously add damping to both the robot and the flexible base. This method adds damping by computing the total redundancy solution (equation (100)) as the weighted sum of two solutions independently chosen to dissipate flexible energy and to locally minimize a weighted function of the robot joint velocities (equation (102)). To ensure that the partitioned control solutions were energy dissipating, analytic expressions (equations (111) and (114)) were then derived for the flexible velocity feedback gains. Finally, these expressions were used to develop flexible base and robot energy dissipation solutions (equations (117) through (120)).

5. TEST MODEL

To investigate the effectiveness of reducing flexible vibrations using kinematic redundancies, a model of a planar, five link manipulator with joint flexibility was developed. As seen in Figure 9, the first two joints model a two mode flexible base, while the upper three joints represent a revolute, revolute, revolute (RRR) rigid robot. For the purposes of this work, the lower two joints are assumed to be used for coarse positioning, while the upper three joints are used for fine positioning. As envisioned here, after the lower two joints position the base of the upper three links and the brakes are locked, due to the coupling between the two subsystems, motion by the robot causes the flexible modes (p_1 and p_2) to vibrate about some nominal position. If the orientation of the end-effector is neglected, the upper three links have one degree of redundancy available to reduce the unwanted vibration.

Due to its structure, the model in Figure 9 can also be thought of as a planar model of the SRMS/DOSS manipulator system. From this viewpoint, the lower two joints and links represent, respectively, the shoulder and pitch joints, and the upper and lower booms of the SRMS. Similarly, the upper three links represent the shoulder, elbow, and wrist pitch joints of the DOSS robot. To model damping in the SRMS joints, two percent modal damping was added to the lower two links. The actual values used in the model are listed in Table II.

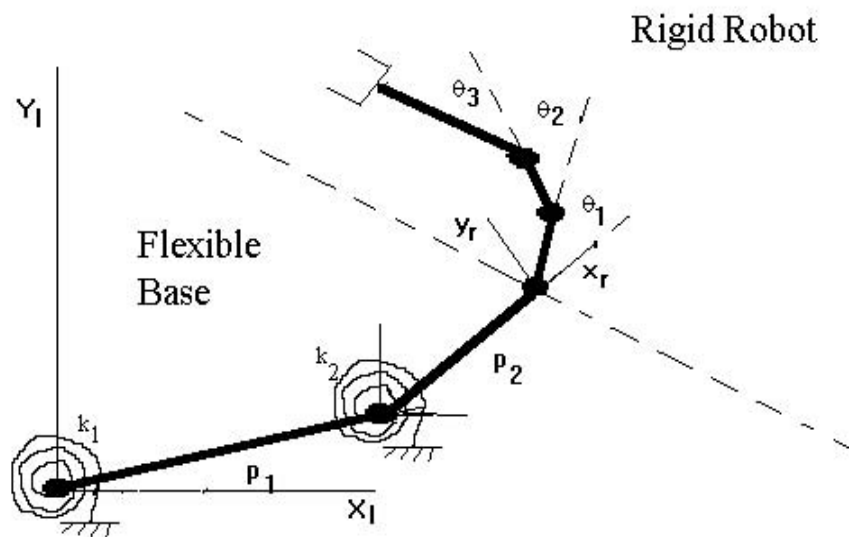


Figure 9. Three Link Rigid Robot / Two Model Flexible Base Model.

Table II. Test Model Physical Parameters.

Subsystem	Link #	Length (m)	Mass (kg)	Inertia (kg-m ²)	Stiffness (Nm) x 10 ⁶
Flexible Base	1	6.4	138.6	471.8	1.02
	2	7.1	87.6	363.4	2.6
Rigid Robot	1	1.0	56.8	56.8	N/A
	2	0.5	34.1	34.1	N/A
	3	0.5	22.7	22.7	N/A

6. FIXED WEIGHT RESULTS

Since actual hardware was not available, numerical simulations were done on the test model in Figure 9. Using a fourth order Adams-Moulton predictor corrector integration algorithm written in the MATLAB programming environment [59], the simulations consisted of integrating the dynamic equations of motion for the three link rigid robot, two mode flexible base system using the previously developed energy dissipation solutions. For convenience, Table III lists these solutions once again.

Table III. Base and Robot Energy Dissipation Solutions.

Method	Redundancy Solutions
GPM	$\underline{u}_{GPM} = \frac{\gamma}{\Delta t} (S^T W S)^{-1} (B^T \phi_b \dot{p} - S^T W \dot{\theta}) - \gamma (S^T W S)^{-1} S^T W J^+ (\ddot{x}_e - \dot{J} \dot{\theta})$
MM1	$\underline{u}_{MM1} = \underline{u}_{gpm} - (1 - \gamma) B^+ (H_{p\theta} J^+ (\ddot{x} - \dot{J} \dot{\theta}) + \underline{N}_p - G \dot{\theta} / 2)$
RPA-DE	$\underline{u}_{RPA-DE} = \underline{u}_{gpm} - (1 - \gamma) B^+ H_{p\theta} J^+ (\ddot{x} - \dot{J} \dot{\theta})$
RW-DE	$\underline{u}_{RW-DE} = \underline{u}_{gpm} - (1 - \gamma) B^+ (H_{p\theta} J^+ (\ddot{x} - \dot{J} \dot{\theta}) + \underline{N}_p)$

To study the sensitivity of the algorithms to the weighting factor γ and the velocity weighting matrix W , ten γ values and three velocity weighting matrices were used. The ten values of γ were 0.00001, 0.001, 0.01, 0.05, 0.1, 0.25, 0.5, .0.75, 0.9, and 1. The three velocity weighting matrices were

$$\begin{aligned}
 1) W &\equiv W_1 = M_\theta(\theta) \\
 2) W &\equiv W_2 = M_\theta(\theta_0) \\
 3) W &\equiv W_3 = \text{diag} [sf * (1 - \dot{\theta}_i / \dot{\theta}_{i,\max})^{-2}]
 \end{aligned} \tag{121}$$

These choices minimize, respectively, the pseudo robot kinetic energy defined by

$$T_{pseudo} \equiv \frac{1}{2} \dot{\theta}_{k+1}^T M_\theta \dot{\theta}_{k+1}, \tag{122}$$

the "initial inertia" weighted robot joint velocities, and the inverse square robot joint velocity norm, where sf is a scale factor with units $\text{kg}\cdot\text{m}^2$ chosen to scale W and $\dot{\theta}_{i,\max}$ is the maximum joint velocity for joint i .

For reference purposes, a baseline algorithm was also simulated. In the baseline algorithm, the redundancy control vector \underline{u} was set identically equal to zero. This represents the case when no attempt is made to control flexible vibrations or add dissipative forces to the robot joints. This choice is also the global solution to minimizing the integral of the squared robot joint velocity norm [55]. In all simulations, perfect knowledge of the system matrices and states of robot and flexible base is assumed.

To compare the different redundancy control algorithms, time history plots and performance criteria were analyzed. The time history plots were (1) the energy E_b in the flexible base, (2) the instantaneous maximum robot joint velocity, and (3) the root sum square (RSS) tracking error ϵ_i defined by

$$\epsilon_i = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \quad (123)$$

where x_i, y_i are respectively, the desired inertial x and y locations at time step i, and \hat{x}_i and \hat{y}_i are the actual inertial x and y end-effector locations. The performance criteria used were: (1) the mean RSS endpoint tracking error,

$$e = \frac{1}{N} \sum_{i=1}^N \epsilon_i \quad (124)$$

where N is the total number of simulation data points, (2) the maximum joint velocity required during the maneuver, $\dot{\theta}_{\max}$, and (3) the control effort, CE, defined by

$$CE \equiv \frac{1}{N} \sum_{i=1}^N \tau_i^T \tau_i \approx \frac{1}{T_m} \int_0^{T_m} \tau^T \tau dt. \quad (125)$$

where T_m is the maneuver end time. For convenience, $e, \dot{\theta}_{\max}$, and CE were

normalized with respect to the baseline case. Thus, values greater than one indicate cases where the performance of the local optimization solution was less than the baseline.

To study the effects of different trajectories and payloads, two desired end-effector trajectories, labelled Trajectory T and Trajectory Y shown in Figure 10 were

simulated. In Trajectory T, the end-effector was required to move a 2500 pound concentrated mass payload a distance of $\underline{d}_m = [-78\text{cm}, 15\text{cm}, 0\text{cm}]^T$ in the local robot coordinates in 3 seconds before stopping, while in Trajectory Y, the end-effector had

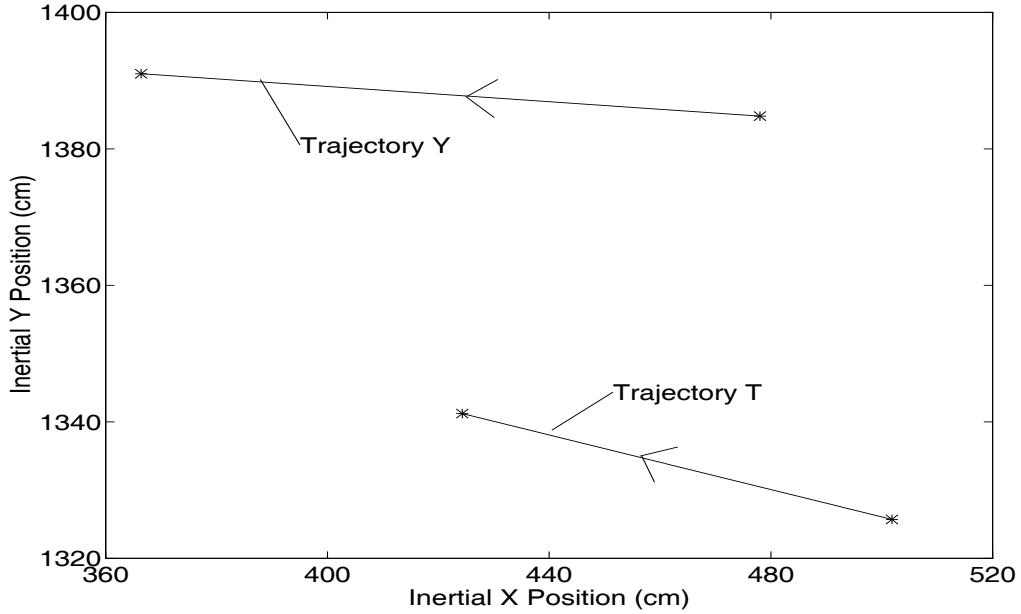


Figure 10. Desired Inertial Trajectories for Three Link Robot.

to move a 1000 pound payload a distance of $\underline{d}_m = [-112\text{cm}, 6\text{cm}, 0\text{cm}]^T$ in 3.5 seconds before stopping. In both trajectories, the desired end-effector acceleration input to the controllers was

$$\ddot{\underline{x}} = \frac{2\pi}{T_m^2} \underline{d}_m \sin\left(\frac{2\pi}{T_m} t\right). \quad (126)$$

(For information on the derivation of equation (126), see Appendix F.) It should be noted that to simulate the response of an actual operator, the local optimization solutions did *not* use any knowledge about the desired, future end-effector state when

optimizing their respective objective functions.

6.1 Vibration Reduction Performance Analysis

To analyze the performance of the energy dissipation controls, Figure 11 through Figure 14 show some simulation results for the four algorithms. To facilitate the analysis, these results are for $\gamma = 1 \times 10^{-5}$ and 1×10^{-2} instead of the full set of γ values. Note: these values were selected because they are representative of the entire set of simulations.

Figure 11 shows the time history of energy in the flexible base for $\gamma = 1 \times 10^{-5}$ and $W = M_0$. As seen in the figure, energy reduction varied considerably amongst the four redundancy control algorithms. Compared to the baseline, GPM yielded virtually no energy reduction, RPA-DE reduced energy but went unstable, while RW-DE and MM1 were nearly identical and provided the greatest reduction in energy. Inspection of Table III shows that GPM's performance is the result of γ being nearly equal to zero. Since the closed-loop damping matrix D_{gpm} is not symmetric for $\gamma = 0$, energy dissipation forces the redundancy to go to zero to make $D_{\text{gpm}} = 0$. As a result, for small γ 's, the GPM solution converges to the baseline solution where u is identically equal to zero.

For RW-DE and MM1, inspection of Table III shows that for any arbitrary robot joint velocity, these two solutions yield the same results only when the time rate of change of the generalized sensitivity matrix of the flexible base inertia matrix (see equation (75)) is negligible. Examination of Figure 11 shows this to be the case, as

RW-DE and MM1 yielded the same solution. Furthermore, because the redundancy solution for RW-DE with small weighting factors is

$$\underline{u}_{MM1}|_{\gamma=0} = -\mathbf{B}^+ (\mathbf{H}_{p\theta} \mathbf{J}^+ (\ddot{\mathbf{x}}_e - \dot{\mathbf{J}} \dot{\boldsymbol{\theta}}) + \underline{N}_p), \quad (127)$$

closed-loop energy dissipation forces the redundancies to minimize the wrench applied by the robot. Thus, if the only goal is to reduce flexible motion, satisfying closed-loop energy dissipation does not allow flexible velocity feedback alone! Instead, closed-loop energy dissipation requires both flexible and robot joint velocity feedback ($\gamma > 0$). Although appealing from an implementation standpoint, the consequences of this choice are unstable motion and increased energy absorption.

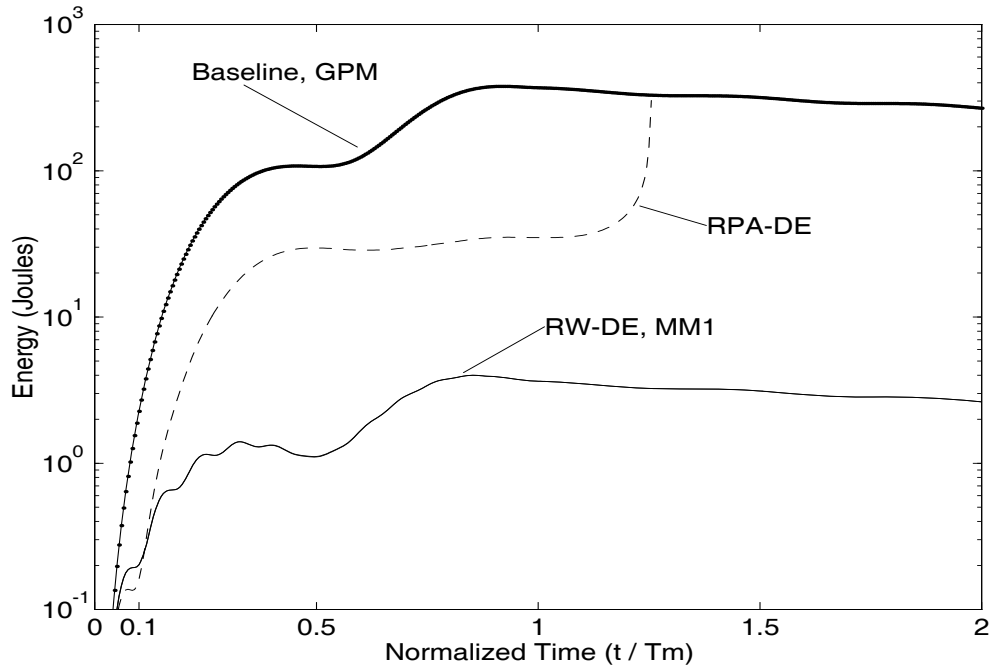


Figure 11. Energy in Flexible Base for Trajectory T, $\gamma = 1 \times 10^{-5}$, $\mathbf{W} = \mathbf{M}_\theta(\theta)$.

As an example, consider the energy results obtained using RPA-DE. Inspection of Figure 11 shows that up until approximately 1/10th of the way through the maneuver, RPA-DE and RW-DE had similar energy absorption curves. Then, around one tenth of the way into the maneuver, when the robot and flexible base velocities built up, the non-linear Coriolis torque \underline{N}_p became the dominant energy source and added energy into the base. This continued until 25% normalized time after the maneuver, where due to insufficient robot joint damping, the flexible base energy dissipation redundancy solution significantly increased the energy in the robot and drove the robot unstable. It should be pointed out that this behavior was not due to the pseudoinverse used in calculating RPA-DE, but was the result of insufficient robot joint damping that allowed $u_{\text{RPA-DE}}$ to cause unstable robot joint velocity buildups.¹⁰ In contrast, RW-DE did not go unstable because in the latter part of the maneuver, including \underline{N}_p decreased the robot joint velocities.

However, from a theoretical standpoint, including \underline{N}_p does not guarantee that robot or base energy will dissipated. Indeed, Figure 12 shows that by changing the desired trajectory and payload, using RW-DE and MM1 without any robot and flexible base damping can also result in unstable flexible base motion. Whereas the redundancy solution $u = -B^+(H_{p\theta}J^+(\ddot{x}_e - \dot{J}\dot{\theta}) + \underline{N}_p)$ dissipated robot energy in the latter part of Trajectory T, in Trajectory Y, a singularity was encountered just before

¹⁰ The term "unstable" is often used when discussing the behavior of redundant robots. Typically, it is used to refer to "excessively high" joint velocities. However, due to joint rate limits, in practice the robot would not actually go dynamically unstable, but the joint velocities would reach the maximum level and cause end-effector tracking errors.

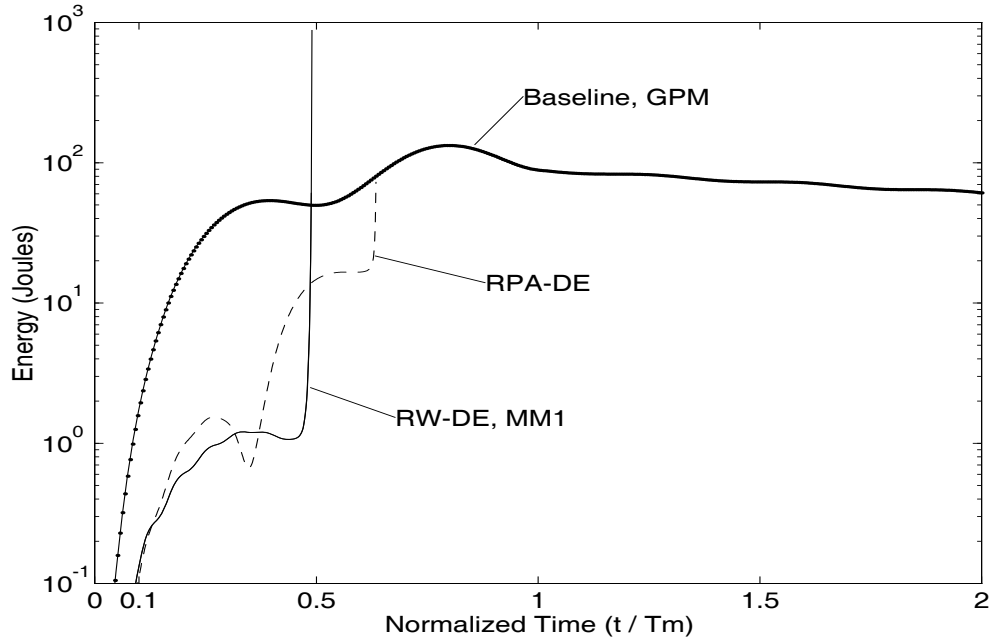


Figure 12. Energy in Flexible Base for Trajectory Y, $\gamma = 1 \times 10^{-5}$, $W = M_0(\theta)$.

the midpoint of the maneuver which caused the time rate of change of robot velocity energy due to the control, $\dot{E}_r|_u$ defined in

$$\begin{aligned} \dot{E}_r &= \frac{d}{dt} \left(\frac{1}{2} \dot{\theta}^T \dot{\theta} \right) = \dot{\theta}^T S \underline{u} + \dot{\theta}^T J^+ (\ddot{x}_e - J \dot{\theta}) \\ &\equiv \dot{E}_r|_u + \dot{E}_r|_{PA} \end{aligned} \quad (128)$$

to increase faster than the time rate of change due to the particular acceleration $\dot{E}_r|_{PA}$ could dissipate energy (see Figure 13). As a result, energy was continually added to the robot which resulted in unstable robot joint motion. Then, because the time rate of change of flexible energy is dependent upon the robot joint velocities, i.e.

$$\dot{E}_f = -\mathbf{V}_b^T [\mathbf{B}\mathbf{u} + \mathbf{H}_{p\theta}\mathbf{J}^+ \ddot{\mathbf{x}}_e + \mathbf{N}_p] - \dot{\mathbf{p}}^T \mathbf{D}\dot{\mathbf{p}} + \mathbf{V}_b^T [\mathbf{H}_{p\theta}\mathbf{J}^+ \dot{\mathbf{J}} + \mathbf{G}/2] \dot{\boldsymbol{\theta}}, \quad (129)$$

where $\dot{\mathbf{J}}$ is also a function of the joint velocities, the increasing joint velocities caused large increases in flexible base energy as shown in Figure 12.

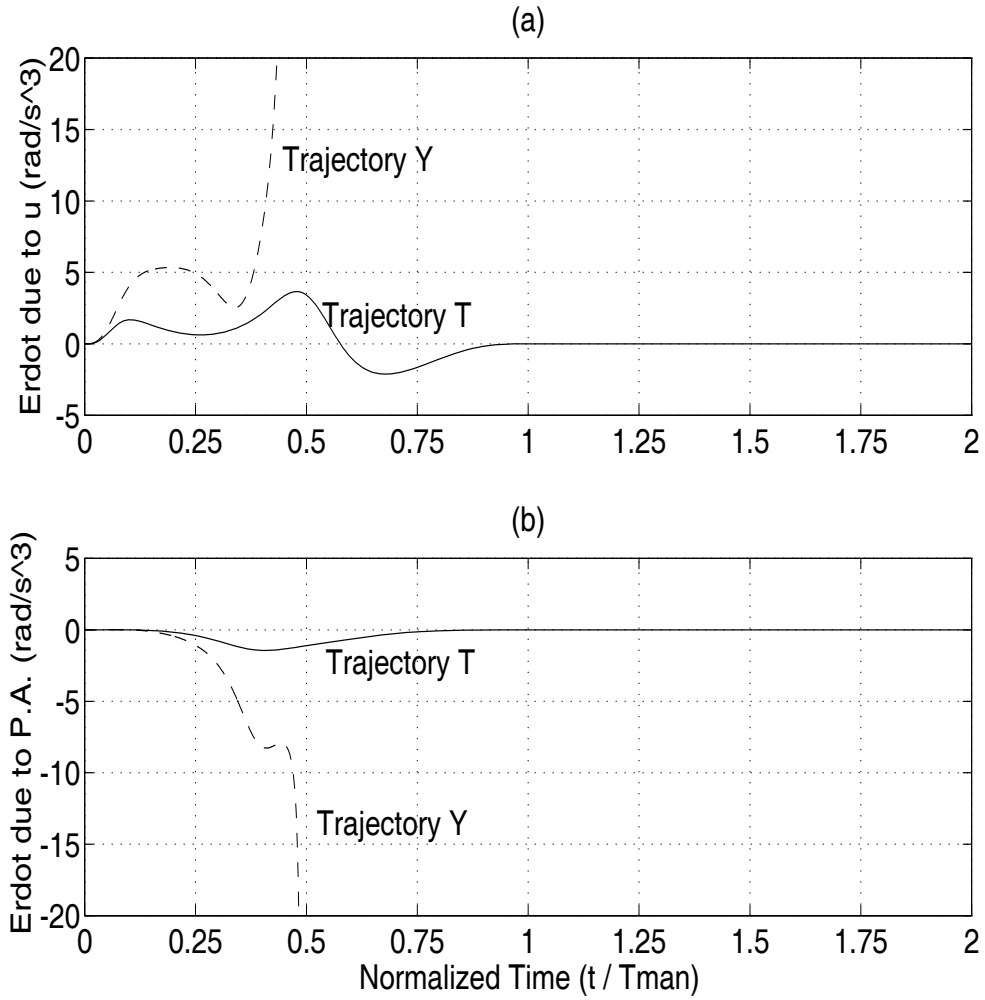


Figure 13. Change in Robot Joint Velocity Norm for RW-DE, $\gamma = 1 \times 10^{-5}$, $\mathbf{W} = \mathbf{M}_\theta(\boldsymbol{\theta})$.

To eliminate unstable robot motion, more damping can be applied to the robot by increasing the weighting factor γ . Inspection of Table III shows that the maximum amount of robot and base damping occurs when $\gamma = 1$. For this value, the redundancy control algorithms converge to

$$\underline{u}|_{\gamma=1} = \frac{1}{\Delta t(S^T W S)} \left[B^T \phi_b \dot{p} - S^T W \dot{\theta} - \Delta t S^T W J^+ (\ddot{x}_e - J \dot{\theta}) \right]. \quad (130)$$

Inspection of this equation shows at maximum damping, no control is used to reduce the rate of change of energy caused by the robot particular joint accelerations and the Coriolis wrench torques. Instead, the effects on the weighted robot joint velocity norm due to the particular accelerations are minimized, while using robot and flexible base velocity feedback to add damping.

Figure 14 shows the energy in the flexible base when γ was increased from 1×10^{-5} to 1×10^{-2} . As seen in the figure, the primary effect of increasing γ was the addition of joint dissipative torques which prevented unstable robot joint velocity buildups that allowed the robot to complete the maneuver with less absorbed flexible base energy than the baseline solution. However, compared to Trajectory T which did not include velocity feedback, inspection of Figure 14 shows that because the robot joint velocity term in u_{GPM} in Table III can be larger than the flexible base velocity term, one drawback to increasing robot velocity feedback is a decrease in flexible vibration suppression.

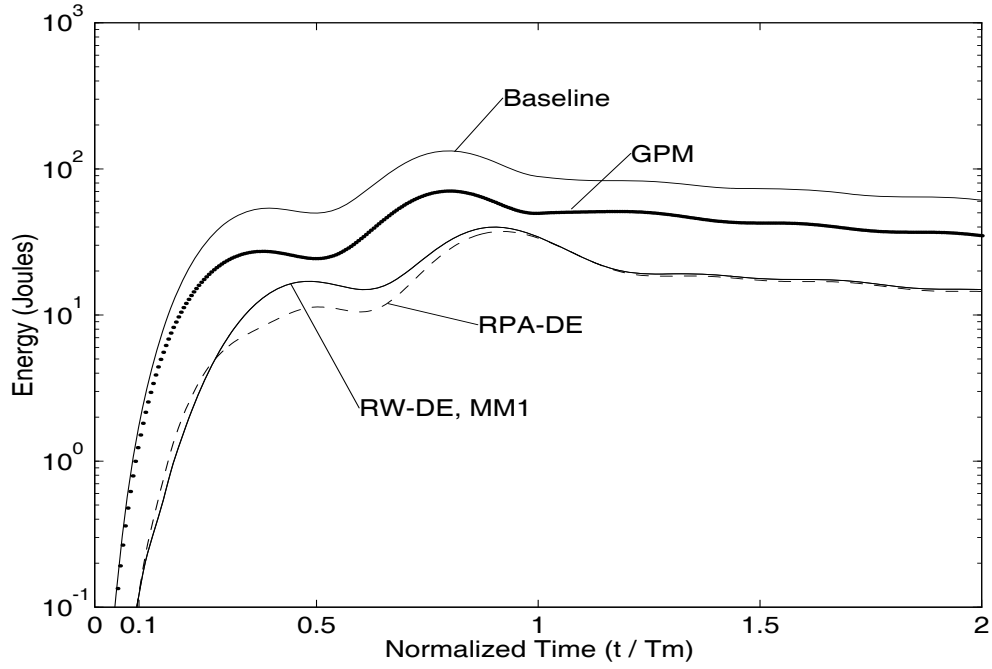


Figure 14. Energy in Flexible Base for Trajectory Y, $\gamma = 1 \times 10^{-2}$, $W = M_0(\theta)$.

Inspection of Figure 14 also shows that compared to ignoring the wrench applied by the robot (GPM), minimizing either all or part of the wrench applied by the robot (RW-DE or RPA-DE) can be energy dissipating. Indeed, if the wrench applied by the robot is adding energy to the base, i.e.

$$-\dot{\mathbf{p}}^T [H_{p\theta} \mathbf{J}^+ (\ddot{\mathbf{x}}_e - \dot{\mathbf{J}}\dot{\theta}) + \underline{N}_p] > 0, \quad (131)$$

choosing \mathbf{u} to cancel the wrench by solving

$$\mathbf{B}\mathbf{u} = -(1 - \gamma) [H_{p\theta} \mathbf{J}^+ (\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\theta}) + \underline{N}_p] \quad (132)$$

is energy dissipating whenever

$$-\dot{\mathbf{p}}^T \mathbf{B} \mathbf{u} = -\left[-\dot{\mathbf{p}}^T (\mathbf{H}_{p\theta} \mathbf{J}^+ (\ddot{\mathbf{x}}_e - \dot{\mathbf{J}} \dot{\boldsymbol{\theta}}) + \mathbf{N}_p) \right] (1 - \gamma) < 0. \quad (133)$$

is true. Likewise, if the applied wrench is dissipating energy, the wrench need not be included when calculating the redundancy.

Unfortunately, equation (132) is valid only when B is square and non-singular or when the rank of B is larger than the number of task variables, m. Thus, since B is a 2 x 1 vector and m = 2 in the test model, minimizing the applied wrench is not necessarily energy dissipating even though the applied wrench is. Because the majority of redundant robots have only one degree of redundancy, in a practical sense this implies that minimizing all or part of the applied wrench does not necessarily improve performance. Indeed, due to the pseudoinverse, in Trajectory T, minimizing the entire wrench (RW-DE) had less energy and smaller tracking errors than minimizing part of the wrench (RPA-DE), while the results were switched in Trajectory Y.

6.2 Sensitivity Analysis

The above results were obtained using the time varying robot inertia matrix, $M_\theta(\theta)$, as the one step ahead robot joint velocity norm matrix W in equation (102). Because the one step ahead velocity is not the current velocity, the robot stabilizing control minimizes the pseudo robot kinetic energy, T_p . Although kinetic energy dissipation is attractive, computing the elements of $M_\theta(\theta)$ is, in general, difficult and time consuming. Therefore, since the choice of W is arbitrary, a natural question to ask is "how does performance vary with W?" To answer this question, two additional

robot joint velocity norm matrices, W_2 and W_3 in equation (121), were used in the control algorithms. Some sample results are given in Figure 15 and Figure 16.

Inspection of Figure 15 shows that for the $\gamma = 1 \times 10^{-5}$, neither the fixed velocity weighting matrix W_2 nor the time-varying inverse velocity squared matrix W_3 prevented unstable motion. However, similar to the results for $W_1 = M_0(\theta)$, increasing γ to 1×10^{-2} resulted in stable motion. Since W_2 and W_3 were arbitrarily chosen, this result suggests that stability is primarily controlled by the value of γ .

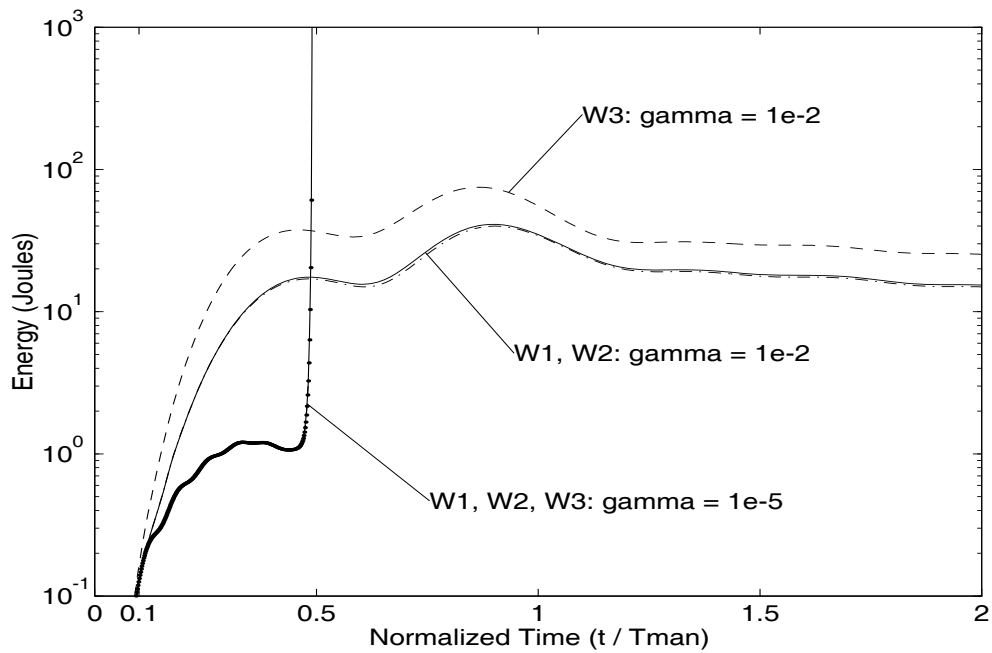


Figure 15. Energy in Flexible Base for RW-DE in Trajectory Y as a Function of Velocity Weighting Matrix.

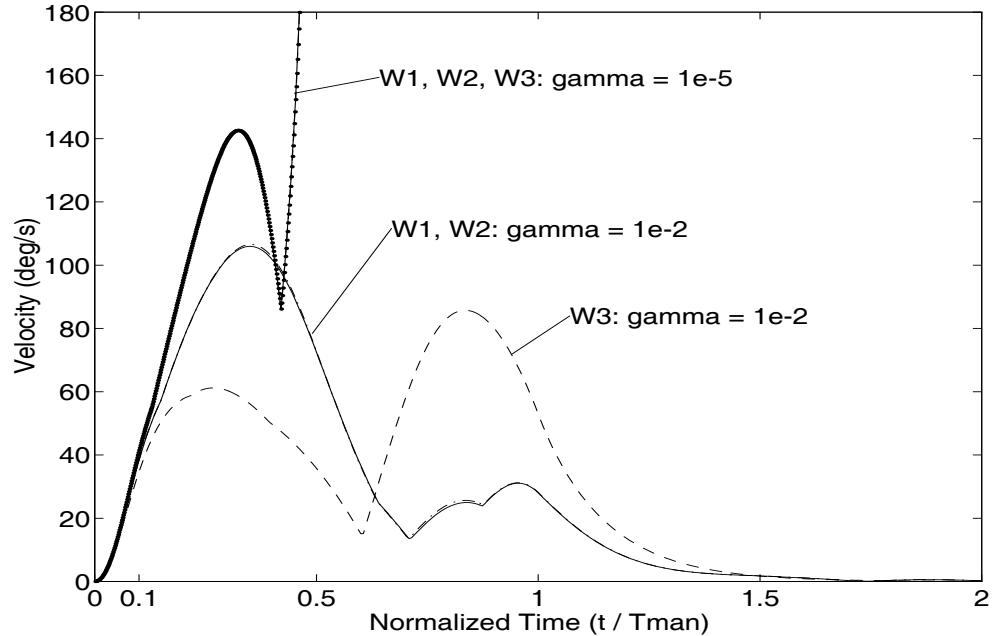


Figure 16. Maximum Joint Velocity for RW-DE in Trajectory Y as a Function of the Velocity Weighting Matrix.

Performance, on the other hand, is controlled by both γ and W . Indeed, Figure 15 shows that although there was little difference between the time varying inertia matrix and the initial inertia matrix (W_1 and W_2), using the inverse squared velocity matrix resulted in greater energy being absorbed by the flexible base. Since the vibration suppression performance of W_3 was less than W_1 and W_2 regardless of the value of γ (See Appendix G.), non-diagonal matrices which incorporate the dynamic coupling in the joints appear to be preferred for vibration damping over diagonal matrices that only include kinematics.

However, if decreasing the maximum joint velocity is a primary objective, examination of Figure 16 and the maximum velocity sensitivity curves in Appendix G shows that the inverse squared velocity matrix is to be preferred. Unlike W_1 or W_2 , W_3 changes the weight applied to each velocity proportional to the square of the distance between the current and the maximum allowable velocity. Thus, as a particular joint velocity approaches its maximum, minimizing the velocity norm using W_3 automatically uses the redundancies to reduce that particular velocity.

Notwithstanding, it should be pointed out that the joint velocities are not reduced unless γ has the proper value. Indeed, the preceding results show that the key factor in determining performance is γ . If γ is too small, insufficient robot joint damping torques will be produced which can lead to instabilities. Alternatively, if γ is too large, vibration reduction will be sacrificed. Thus, an automatic method for computing γ is needed. Before developing such a method, it is first worthwhile to use the previously defined performance criteria to investigate sensitivity with respect to γ .

Tracking Error Sensitivity to γ

Figure 17 and Figure 18 show the variation in the mean RSS inertial tracking error for Trajectories T and Y. In the figures, the symbol ' \exists ' is used to denote the stability boundary defined as the minimum value of γ for which instabilities did not occur. Consistent with the results from the time history plots, inspection of the figures shows that with the exception of GPM, tracking error is directly proportional to γ . In GPM, this does not hold because as γ goes to zero, so does u . Then since the baseline is for u identically equal to zero, the normalized performance of GPM approaches one.

Inspection also shows that as γ goes to one, the four control algorithms have the same performance. This occurs because as γ goes to one, less emphasis is placed on minimizing components of the applied wrench on the base while more emphasis is placed on adding dissipative forces. For γ equal to one, these dissipative forces reduce the motion of the robot and the base resulting in stable motion.

In terms of performance, the sensitivity curves show that amongst the minimization solutions (MM1, RW-DE, and RPA-DE), vibration suppression is particularly sensitive to singularities and the energy dissipation resulting from minimizing components in the applied wrench. In Trajectory T, RW-DE and MM1 had the most vibration reduction, while in Trajectory Y, RPA-DE had the best. The difference between these two cases is related to the non-linear effects of the Coriolis torque. In Trajectory T, minimizing this torque increased energy dissipation, while in Trajectory Y including this term decreased energy dissipation.

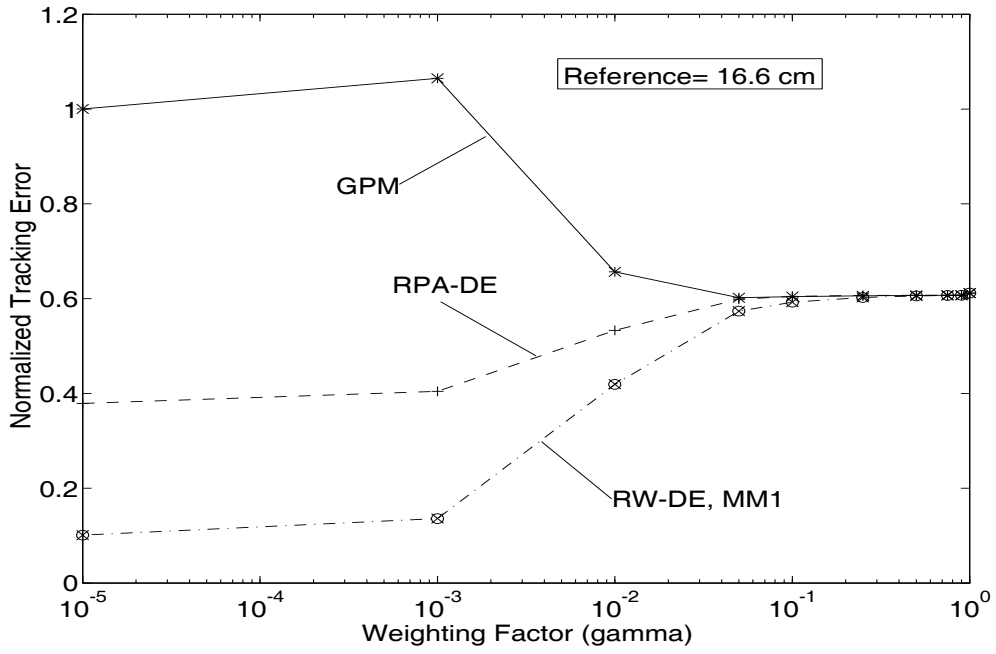


Figure 17. Sensitivity of Mean Tracking Error in Trajectory T, $W=M_0$.

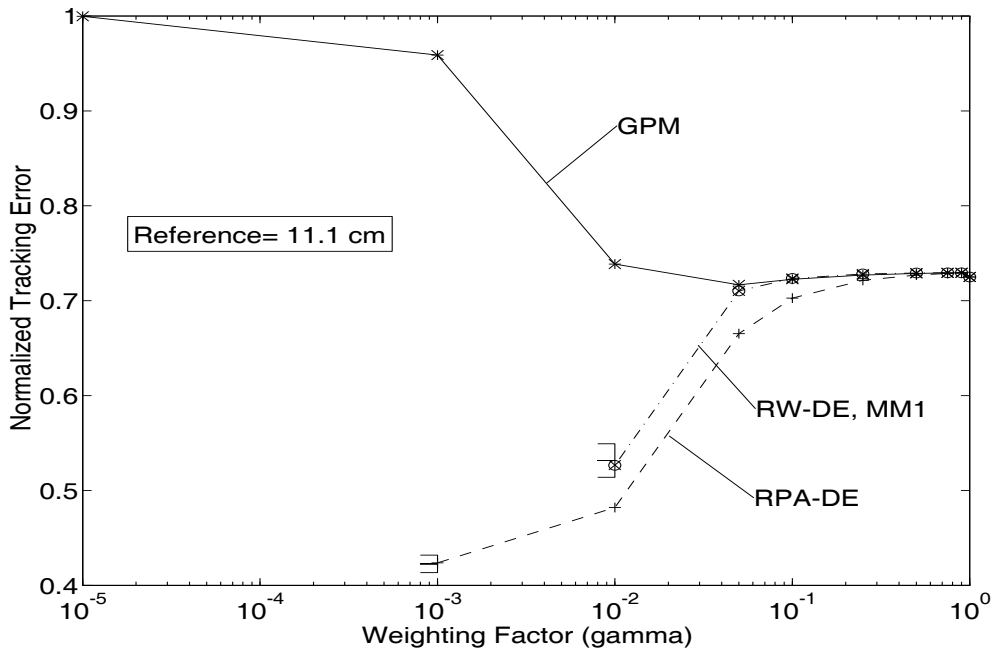


Figure 18. Sensitivity of Mean Tracking Error in Trajectory Y, $W=M_0$.

Since all of the control algorithms use inertia information while the baseline algorithm does not, these results suggest that choosing robot trajectories which follow inertia weighted paths are preferred for reducing flexible base vibrations. Furthermore, when compared to the results obtained using the local minimum robot energy redundancy solution [60] and the global minimum robot energy redundancy solution [55], the new redundancy solutions, i.e. GPM, MM1, RW-DE, and RPA-DE, had greater vibration suppression.

Control Effort Sensitivity to γ

Figure 19 and Figure 20 show the sensitivity of the required control torque as a function of γ . Inspection of the figures shows that (1) unlike the mean tracking error, for the most part, the required control effort is inversely proportional to γ and (2) the greatest vibration suppression requires the most amount of control. Since smaller γ 's place more weight on reducing flexible vibration, these plots indicate that reducing flexible vibrations is more expensive than adding dissipative forces via minimizing the pseudo kinetic energy.

Because the normalized solutions for $\gamma = 1$ are stable by virtue of the robot dissipative torques and are much less than 1, these results also indicate that minimizing the one step ahead inertia weighted robot joint velocity norm is a suitable approach for solving the joint torque minimization problem. Indeed, it has been shown that globally minimizing the kinetic energy is equivalent to locally minimizing the inverse inertia weighted joint torque norm [56], i.e.

$$\min \underline{x}^T M_\theta^{-1} \underline{x} \equiv \min \frac{1}{2} \int_{t_0}^{T_m} \dot{\underline{\theta}}^T M_\theta \dot{\underline{\theta}} dt. \quad (134)$$

The joint accelerations obtained from minimizing equation (134) are

$$\ddot{\underline{\theta}} = \mathbf{J}_H^+ (\ddot{\underline{x}}_e - \dot{\mathbf{J}} \dot{\underline{\theta}}) - (I - \mathbf{J}_H^+ \mathbf{J}) M_\theta^{-1} \underline{N}_\theta \quad (135)$$

where inertia weighted Jacobian pseudoinverse \mathbf{J}_H^+ is

$$\mathbf{J}_H^+ = M_\theta^{-1} \mathbf{J}^T (\mathbf{J} M_\theta^{-1} \mathbf{J}^T)^{-1} \quad (136)$$

and \underline{N}_θ is the Coriolis torque acting on the robot joints. In contrast, the joint accelerations obtained using the one step ahead inertia weighted robot joint velocity norm are

$$\ddot{\underline{\theta}} = \mathbf{J}_{1S}^+ (\ddot{\underline{x}}_e - \dot{\mathbf{J}} \dot{\underline{\theta}}) - \mathbf{S} (\mathbf{S}^T M_\theta \mathbf{S})^{-1} \mathbf{S}^T M_\theta \frac{\dot{\underline{\theta}}}{\Delta t} \quad (137)$$

where the one step ahead Jacobian pseudoinverse \mathbf{J}_{1S}^+ is given by

$$\mathbf{J}_{1S}^+ = (I - \mathbf{S} (\mathbf{S}^T M_\theta \mathbf{S})^{-1} \mathbf{S}^T M_\theta) \mathbf{J}^+. \quad (138)$$

Because it does not require computing the inverse of the fully populated inertia matrix or the non-linear Coriolis torque, the one step ahead solution requires fewer computations than the inverse inertia weighted solutions. For example, in the three degree of freedom robot used in the test model, computing the global kinetic energy solution required 980 floating point operations while the one step solution required only 500. Thus, using the one step ahead joint velocity norm to stabilize the robot has the added benefit of also reducing the required joint torques.

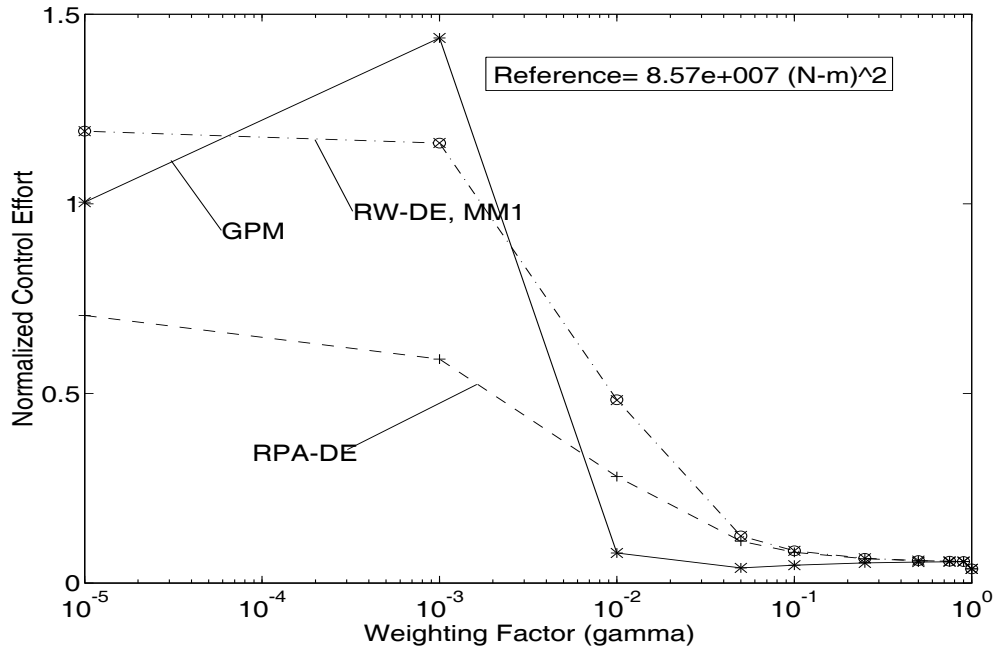


Figure 19. Sensitivity of Required Control Torque in Trajectory T, $W=M_0$.

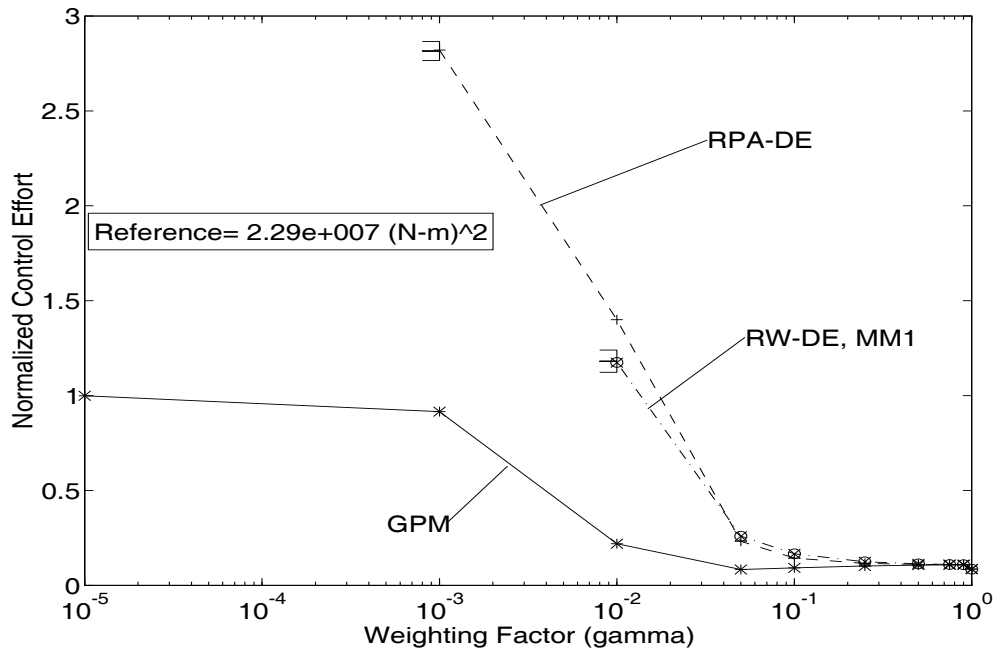


Figure 20. Sensitivity of Required Control Torque in Trajectory Y, $W=M_0$.

Maximum Joint Velocity Sensitivity

The variation in maximum robot joint velocity is shown in Figure 21 and Figure 22. Examination of the curves for high values of γ shows that increasing γ decreases the variation in the maximum joint velocity. Indeed, for $\gamma \geq 0.1$ in Trajectory T and $\gamma \geq 0.3$ in Trajectory Y, the curves for the different redundancy solutions were nearly identical. Since increasing γ places more weight on stabilizing the robot, this result is the consequence of the effect of larger joint dissipative torques.

In contrast, inspection of the curves for low values of γ shows that there is considerable variation. Due to the effects of the non-linear torques, RW-DE and MM1 required the largest velocities in Trajectory T, while in Trajectory Y, RPA-DE required the largest velocity. Similarly, because GPM requires smaller controls as γ decreases, GPM had the smallest peak velocities.

Since the largest velocities occurred for the smallest weighting and vice versa, these results indicate that reducing flexible vibrations requires higher peak robot joint velocities. Similarly, because the baseline case is the globally minimum robot joint velocity norm solution, the performance for $\gamma \approx 1$ indicates that minimizing robot kinetic energy also requires larger joint velocities.

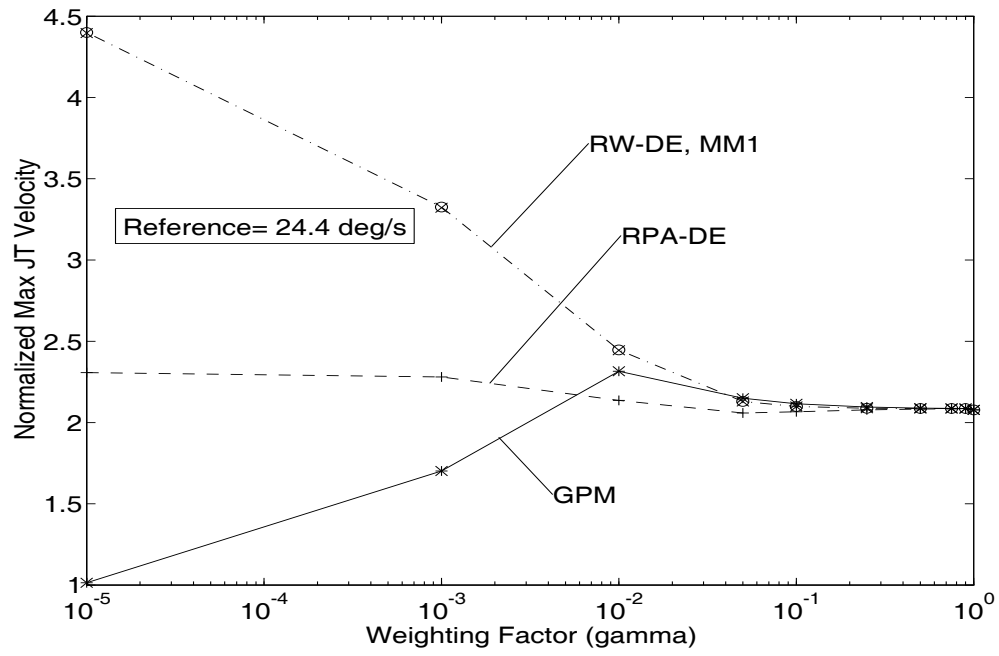


Figure 21. Sensitivity of Maximum Joint Velocity in Trajectory T, $W=M_0$.

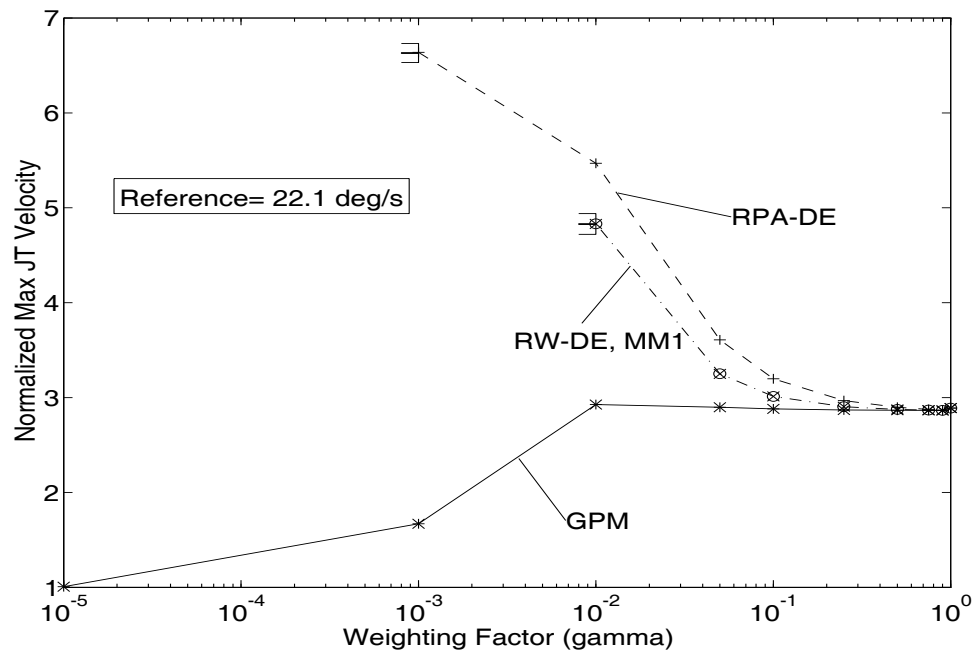


Figure 22. Sensitivity of Maximum Joint Velocity in Trajectory Y, $W=M_0$.

6.3 Discussion

The above results indicate that if γ is properly chosen, partitioned control can be used to stabilize local optimizations. They also show that following reduced base motion paths ($\gamma \rightarrow 0$) requires more robot effort than following minimum robot kinetic energy paths ($\gamma \rightarrow 1$). This suggests that there is a tradeoff between reducing flexible vibrations and reducing robot motion. Although consistent with the idea of partitioned control, this result is somewhat contrary to intuition since one would think that decreasing, not increasing, robot motion should reduce base motion. This apparent contradiction is a result of two facts.

First, as implemented here, reducing robot motion chooses joint accelerations that add joint dissipative torques that reduce the effects of future robot joint velocities. Minimizing flexible base motion, on the other hand, does not add joint dissipative torques or penalize future joint velocities. Thus, reducing base motion actually increases robot motion. As a result, more robot control and energy effort are required to overcome the effects of velocity buildups.

The second reason for the apparent contradiction is the inherent limitations of having fixed, desired end-effector accelerations. If the end-effector trajectory were variable, input pre-shaping techniques [61-63] could be used to modify the desired acceleration such that the motion of the end-effector does not excite flexible modes. However, as was mentioned before, a fundamental assumption of this work is that end-effector accelerations cannot be altered.

In Chapter 4, it was mentioned that, in the torque optimization problem, locally

resolving redundancies at the acceleration level often leads to large joint velocities and torques. The results in the previous sections showed that this same phenomenon occurs when resolving redundancies to control flexible base vibrations. The results also showed that adding joint dissipative torques can alleviate this problem but only if γ is properly chosen. Choosing values too small can lead to instabilities, while choosing values too high decreases vibration damping performance. Thus, it is worthwhile to investigate the instability problem in local redundancy resolutions with particular emphasis on the effects of adding a robot stabilization term to the total redundancy solution.

In [57], Ma et al reported that instabilities occurred when the singular values of the optimization or information matrix decreased. Since this matrix must be inverted, decreased singular values indicate optimization difficulties. In RW-DE, RPA-DE, and RPA, the optimization matrix is the control influence matrix $B = H_{p0}S$ which is 2×1 for the three link model. Thus, with respect to the work done by Ma in torque optimization, decreased singular values of the information matrix imply a reduction in controllability in the present problem. Indeed, as shown in Figure 23, one distinct difference between stable and unstable trajectories is a sharp drop in controllability.

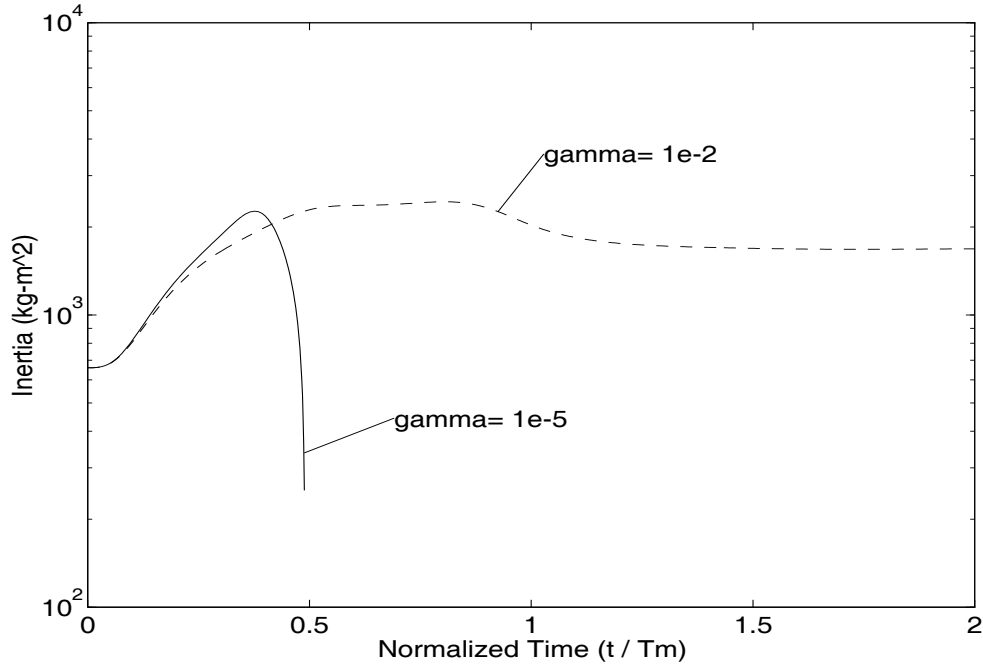


Figure 23. Norm of Control Influence Vector for Stable and Unstable Trajectories.

Another difference between stable and unstable trajectories is the increase in the particular acceleration term of the optimization function. To see this, re-write the original optimization problem (equation (91) on page 47) as

$$L(u_{CBM}) = \|Bu - \underline{T}_{des}\|^2 \quad (139)$$

where \underline{T}_{des} is the desired control torque on the flexible base. Figure 24 shows plots of the norm of \underline{T}_{des} for the same cases as in Figure 23. Inspection of the figure shows that another aspect of unstable trajectories is that as the control influence matrix became singular, the desired control torque also increased. Since \underline{T}_{des} is proportional to the square of the velocities ($\underline{J}\dot{\underline{\theta}}$), the effects of high joint velocities are amplified near a singularity. This leads to even larger joint velocities and unstable joint velocities.

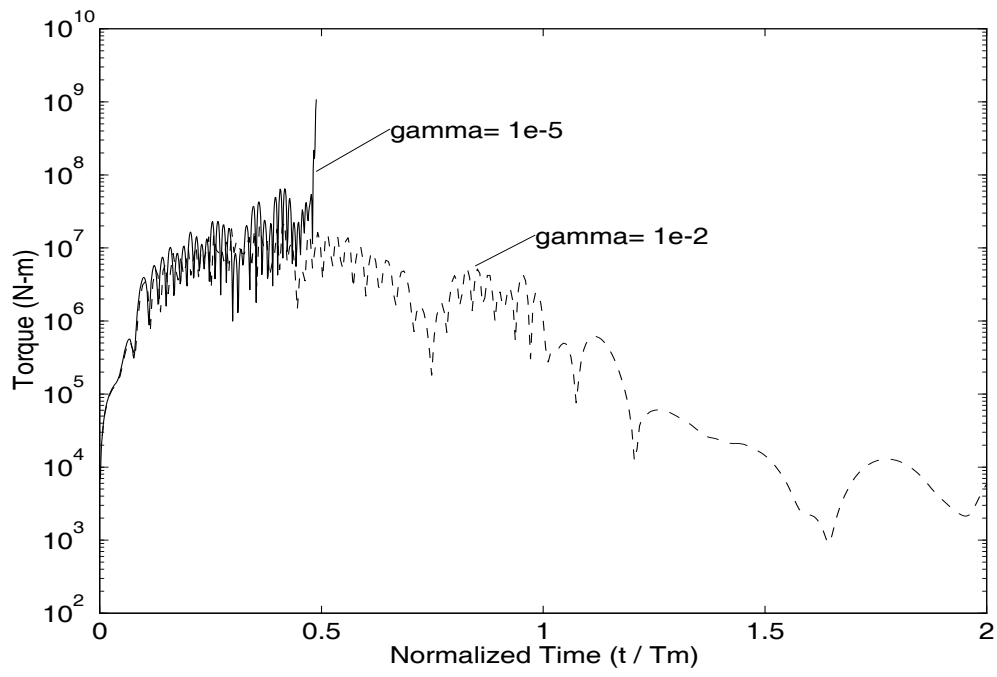


Figure 24. Norm of Desired Control Torque.

7. USING FUZZY LOGIC TO COMPUTE THE WEIGHTING FACTOR

The results in Chapter 6 showed that if the weighting factor γ was properly selected, choosing part of \underline{u} to minimize some measure of robot motion can prevent the instabilities associated with local optimizations. Unfortunately, choosing a proper value of γ without knowing the complete end-effector acceleration trajectory beforehand is difficult, if not impossible. If γ is chosen too small, instabilities can arise, while if γ is too high, smaller robot joint velocities and less control effort are obtained at the expense of vibration damping. Since the fuzzy or vague qualifiers "too small" and "too high" are intimately related to the desired end-effector trajectory, the robotic payload, and the state of the flexible structure, an automated means of determining γ is needed.

With this in mind, the present chapter presents and discusses using fuzzy logic as one way of automatically computing γ . First introduced by Zadeh in 1968 [65], fuzzy logic is a mathematical way of handling vague and imprecise information, particularly information in the form of human knowledge expressed in linguistic terms, such as "too small" or "very hot". As such, fuzzy logic is a candidate for solving the present problem. The outline of the chapter is as follows. First, because partitioning \underline{u} into $(1-\gamma)*\underline{u}_{CBM}$ and $\gamma*\underline{u}_{SRM}$ is an attempt to simultaneously optimize two secondary criteria, an overview of the current status of multi-criteria optimizations in redundant robots is presented. Then, a Fuzzy Logic Supervisor (FLS) is proposed and designed

for optimizing multiple criteria in KRRs with emphasis on flexible vibration damping.

7.1 Optimizing Multi-Criteria in Kinematically Redundant Robots

As was mentioned earlier, kinematically redundant robots have been used in a variety of applications. In most of these applications, only a single criterion or subtask was optimized. However, in dynamically changing and ill-defined environments, optimizing a single criterion is limiting and can lead to problems [66]. For instance, choosing the redundancy to minimize kinetic energy does not imply that the robot will avoid any obstacles or hardware limitations such as position and velocity limits. Thus, multiple criteria need to be optimized in order to achieve adequate performance.

The notion of optimizing multiple criteria is not new. Indeed, Nakamura et al [67] were among the first to consider multiple optimization. Their approach was to first divide the total task into l subtasks, each with its own constraint relation defined by

$$\dot{\mathbf{x}}_i = \mathbf{J}_i \dot{\boldsymbol{\theta}}_i, i = 1, 2, \dots, l \quad (140)$$

where $\dot{\mathbf{x}}_i$ and \mathbf{J}_i are, respectively, the desired end-effector velocity and Jacobian associated with the i -th subtask. Using equation (140), they then obtained a recursive solution for robot joint velocities that simultaneously optimize multiple criteria. For two tasks, their solution was

$$\begin{aligned}
i=1 & \rightarrow \dot{\Theta} = J_1^+ \dot{x}_1 + (I - J_1^+ J_1) \gamma = J_1^+ \dot{x}_1 + N_1 \gamma \\
i=2 & \rightarrow \dot{x}_2 = J_2 (J_1^+ \dot{x}_1 + N_1 \gamma) \\
\gamma & = \tilde{J}_2^+ (\dot{x}_2 - J_2 J_1^+ \dot{x}_1) + (I - \tilde{J}_2^+ \tilde{J}_2) \rho
\end{aligned} \tag{141}$$

where $\tilde{J}_2 = J_2 N_1$ and ρ is an arbitrary vector. It should be noted that this method handles priority or importance in the initial ordering of the tasks, i.e., task number one is considered more important than task number two, and so on. Thus, this method is inflexible in terms of handling tasks whose priorities change. As a result, sub-optimum solutions are generated when a higher ordered task becomes more important than a lower ordered task.

Other researchers (e.g. [68, 69]) have taken a different approach. They resolved the redundancies by defining and minimizing a weighted cost function composed of each subtask's individual performance function. If E_i are the individual performance functions and w_i are fixed weights that also take into account the units of each function, this approach calculates the redundancies by minimizing

$$L = \sum_{i=1}^q w_i E_i(\Theta) . \tag{142}$$

Related work has also been done in the end-effector trajectory planning problem. There, Cleary [70] and Panames [71] used normalized cost functions with fixed weights to study the effects of considering multiple subtasks when generating the end-effector trajectory profile. For future comparisons, Cleary's cost function, also known as the composite performance index, is given

$$CPI \equiv \frac{\sum_{i=1}^q w_i E_i}{\sum_{i=1}^q w_i} . \quad (143)$$

It should be noted that Cleary did not actually use equation (143) to resolve the redundancies by optimizing multiple criteria. Instead, he used CPI only as a performance measure to consider when planning trajectories. In contrast, in Section 6.2, an expression similar to CPI will be derived that does resolve the redundancies.

Although the above solutions have produced some successes, using fixed weights has been shown to be restrictive. For example, consider the following situation. Suppose at some point in time, the i -th joint velocity is "VERY CLOSE" to its limit while the joint position is "MEDIUM CLOSE" to the maximum joint excursion and the flexible base is "NOT MOVING VERY FAST." It is obvious that controlling the i -th joint velocity is more important than minimizing the joint position which is more important than reducing flexible vibrations. Thus, at this instant in time most of the weight should be placed on reducing the joint velocity followed by reducing the joint position followed by minimizing flexible vibrations. At a later time, the importance can change and so should the weights.

Similarly, in the obstacle avoidance problem, Maciejewski [66] pointed out that obstacles need not be considered unless they are close to the manipulator. Since it is difficult to choose fixed weights that reflect closeness, Maciejewski introduced a *variable weighting function* dependent on distance from an obstacle where distance was defined by the following three regions: (1) a sphere of influence, (2) a unity gain

region, and (3) a task abort distance. Similarly, Seraji et al also recognized the limitations of fixed weights and used variable weighting functions in an improved version of configuration control. [72]

In spite of these improvements, variable weighting functions were only used to handle a single constraint. Real world applications, on the other hand, require simultaneously optimizing more than a single constraint or subtask, e.g., keeping all of the joints and joint rates within limits, avoiding obstacles, minimizing flexible base vibrations, etc. To overcome this problem, McGhee et al [73] used probability distribution functions (PDF) defined for each subtask. Though the use of PDFs did allow variable weighting optimization of more than one subtask, a sample size of 750,000 experimentally determined data points was needed to construct the PDFs.

Since collecting such a large number of data points may be costly, an alternative method is needed. One choice is to use a Fuzzy Logic Supervisor (FLS). The rationale for using an FLS is that at every instant in time, each criterion or objective is important to some degree, and therefore should contribute to the overall redundancy solution. Thus, the job of the supervisor is to use fuzzy logic to determine the relative importance of each criterion and assign time-varying weights. As will be shown later, compared to generating PDFs, this method is simple to design and implement and does not require large amounts of data.

7.2 Design of a Fuzzy Logic Supervisor (FLS) for KRR

As was mentioned earlier, fuzzy logic is a mathematical method to handle imprecise information, particularly information presented in natural language, i.e. "if the error is LARGE and POSITIVE, and the rate of change of error is VERY SMALL, then the output is LARGE and POSITIVE." Although long been accepted in the Far East and in Europe, fuzzy logic has only recently gained acceptance in the United States. Notwithstanding this fact, fuzzy logic has been successfully applied to many problems in such diverse areas as subways, helicopters, engines, and household appliances [74].

Here, fuzzy logic is applied to controlling the redundant motion of a robot. This is in contrast to previous uses of fuzzy logic in redundant manipulators which use fuzzy logic to eliminate computing the Jacobian and its pseudoinverse [75-77]. As will be shown, when computing the Jacobian and its pseudoinverse is not an issue, controlling the redundancies using fuzzy logic greatly expands the usefulness of the robot.

Although many definitions have been given and debated, at its essence, using fuzzy logic is another way to model human reasoning. As a result, fuzzy logic can be considered a form of artificial intelligence. Using this definition, an FLS can be described as a mechanism that uses a priori knowledge about the process(es) to be controlled and the power of fuzzy inferencing to emulate the response of a human supervisor. Using an FLS is appropriate when "good" controls exist for particular problems but no single control is adequate. For example, in a gas turbine power plant

control problem [78], six low level controllers developed for different operating conditions were used with an FLS to develop a single overall control. Similarly, in a flexible robot control problem [79], an FLS was used in controller selection.

Analogously, a FLS is appropriate for KRRs, when individual redundancy solutions exist for particular subtasks but not all of the subtasks together. Indeed, in Chapter 4 controls were first developed to reduce flexible vibrations and then to add dissipative forces to the robot joints but not to do both simultaneously. To accomplish both, a trajectory and payload dependent weighting factor γ was introduced. However, as was demonstrated in Chapter 6, choosing γ is difficult. Moreover, if q subtasks instead of two are considered, the optimization problem becomes even more complicated since then q weights instead of one must be chosen.

To use an FLS to compute the weighting, a bi-level hierarchical control strategy similar to [79] can be taken. As seen in Figure 25, the first level consists of separate, independent controls for each objective, while the second level consists of a fuzzy adaptive weighting factor computation algorithm that takes auxiliary information in the form of membership functions, hardware limits, and linguistic rules and adaptively changes the weights w_i assigned to each control.

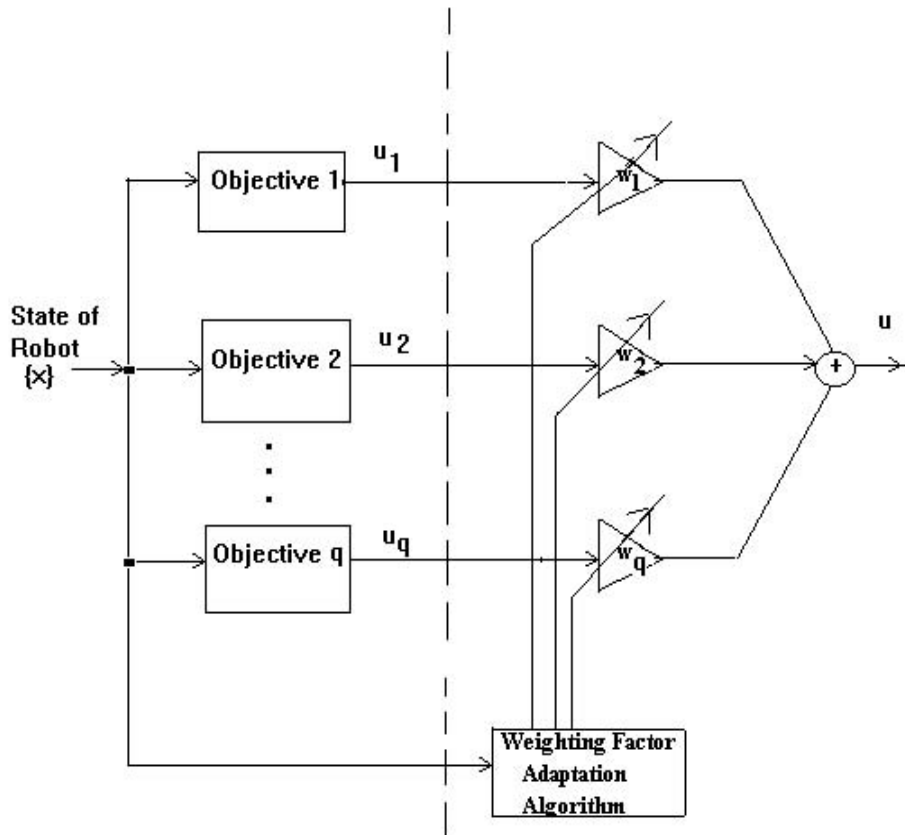


Figure 25. Hierarchy of Control Strategy.

As with all fuzzy systems, the fuzzy weighting factor adaptation algorithm requires three major components: a fuzzification method, an inference engine, and a de-fuzzification method (See Figure 26.) and whose basic operation is as follows. The fuzzification method takes the raw or "crisp" inputs and measurements (x) and converts them into fuzzy values (\tilde{x}) defined for all inputs in some universe of discourse. Next, the inference engine takes the fuzzy values and applies heuristic, common sense linguistic rules to determine the truth or strength of each rule and its output u_i . Using the computed truth values and fuzzy outputs \tilde{u}_i , the de-fuzzification algorithm then computes a single crisp, non-fuzzy output.

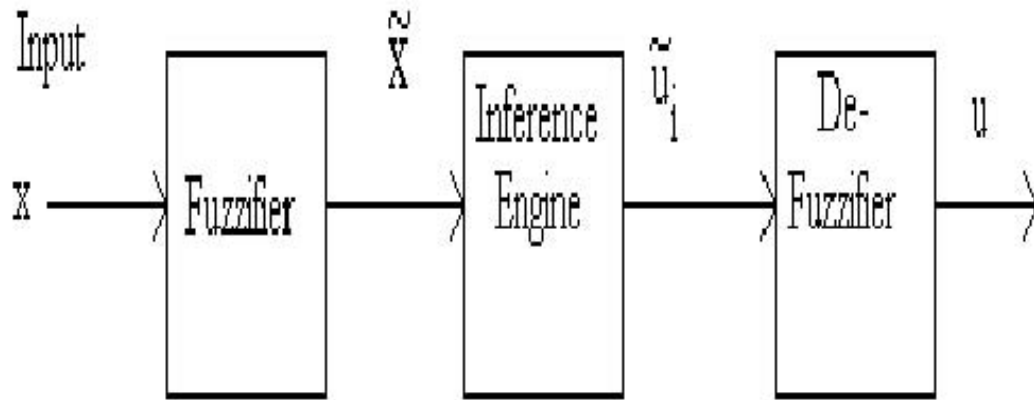


Figure 26. Block Diagram of Adaptive Weighting Factor Computation Algorithm.

Fuzzification is accomplished using membership functions. Membership functions express the degree of truth that an input x belongs to a fuzzy set. Since membership functions can be arbitrarily specified, assume that the degree of truth is an exponential function given by

$$\mu(x) = I_0(d) \left(\exp\left[\frac{b^2}{(x/x^* - 1)^2}\right] - e^{-b^2}\right) \quad (144)$$

where the three parameters (b , d , x^*) are known as, respectively, the rate factor, the truth factor, and the max factor. The definition of these factors is as follows. The truth factor is the normalized distance which defines completely true or 100% important. It reflects a priori knowledge concerning when something is or should be considered important. Similarly, since the derivative of $\mu(x)$ is linearly related to b , the rate factor reflects a priori knowledge about how fast something becomes important. The final parameter is the max factor. The max factor is the maximum value of the input. It normalizes the input to make the universe of discourse lie

between [0, 1].

Although it appears that the above definition for $\mu(x)$ is somewhat arbitrary, there are several reasons for this choice. First, since the fundamental problem is to optimize multiple criteria, one choice is to use techniques developed for other multi-criteria problems. In structural optimization problems, one such technique is to use an exponential weighting function called a KS function [80]. For k constraints, the KS is defined by

$$KS = \frac{1}{\xi} \ln \left\{ \sum_{i=1}^k e^{\xi g_i} \right\} \quad (145)$$

where ξ is a tolerance parameter and g_i are the l constraint equations normalized such that $g_i < 0$ is the feasible domain. Inspection of equation (145) shows that the closer a constraint gets to its feasible boundary, the more it contributes to the KS function. This is analogous to the present situation except that here the "KS-like" membership function is a measure of importance.

A second reason for the above choice is that since the FLS is proposed as an alternative to the computationally intensive PDFs, a reasonable definition for fuzzy importance is to use something similar to the experimentally determined PDFs. As it turns out, these functions were also exponentials.

A third reason for the above choice is that because x/x^* is normalized distance, equation (144) is a distance function. Therefore, the notions of "close" and "far" introduced by Maciejewski in obstacle avoidance are easily represented as $x/x^* \rightarrow 1$ and $x/x^* \rightarrow 0$, respectively. Furthermore, since $\mu(x) \rightarrow \infty$ as $x/x^* \rightarrow 1$ and $\mu(x) \rightarrow 0$ as $x/x^* \rightarrow 0$, the

above also reduces the number of rules needed because the linguistic qualifiers "VERY SMALL" and "VERY LARGE" are automatically included in the membership function definition.

Compared to the majority of multi-criteria approaches which use one parameter (the fixed weight) per criterion, inspection of equation (144) shows that the FLS requires more input on the part of the engineer because it uses three parameters per criterion. However, since x^* is often known from hardware specifications, e.g. the maximum joint limit, while b and d have physical attributes (which the fixed weights don't), it can be argued that choosing the three membership function parameters is easier than determining the single fixed weight.

More importantly, since the FLS automatically adjusts the weight based on the distance from a boundary, the exact values of x^* , b , and d are relatively unimportant. For example, since d is the normalized distance where $\mu=1$, $1-d$ can be viewed as a sphere of influence. Inside the sphere, the importance increases rapidly so that a large weight is placed on the criterion. Thus, the system is less sensitive to the exact value of d than it is to an arbitrarily chosen fixed weight.

Returning to Figure 26, the second component of a fuzzy supervisor is the inference engine. The inference engine uses fuzzy, user defined linguistic rules to make judgements about the system. To simplify matters, Takagi and Sugeno's parametric format [81] can be used to write the i th rule R^i as:

$$R^i: \text{ IF } A_i(\underline{X}, \underline{C}) \text{ THEN } \underline{u}_i=f(\underline{X};\underline{D})$$

where \underline{X} is the state of the robot, \underline{C} and \underline{D} are vectors of auxiliary parameters such as

physical limits, and A_i are linguistic conditional statements. As an example of rules, consider controlling the flexible vibrations of a rigid redundant robot mounted a compliant base. Three criteria for selecting the redundancies might be to (1) reduce the base vibration caused by the motion of the robot, (2) keep the joint angles within their physical limits, and (3) limit the joint rates. Three fuzzy logic supervisor rules might then be:

Table IV. Example of Possible Fuzzy Rules

Rule Number	Fuzzy Rule
1	IF θ_1 is near $\theta_{1, \text{lim}}$ OR θ_2 is near $\theta_{2, \text{lim}}$ OR ... θ_n is near $\theta_{n, \text{max}}$, THEN move θ away from limit.
2	IF $d\theta_1/dt$ is near $(d\theta_1/dt)_{\text{lim}}$ OR $d\theta_2/dt$ is near $(d\theta_2/dt)_{\text{lim}}$ OR ... $d\theta_n/dt$ is near $(d\theta_n/dt)_{\text{lim}}$, THEN reduce joint velocities.
3	IF the base is vibrating too much, THEN damp flexible vibrations.

The strength or truth of the rules is obtained by evaluating the membership function. Thus, with respect to the above rules the truth of the conditionals "is near" and "is moving" is given by evaluating equation (144) with x replaced by each θ_i for rule one, each $d\theta_i/dt$ for rule two, and any measure of base motion such as the norm of the base velocities for rule three. Since rules one and two contain the fuzzy 'OR' operator, the strength of these rules $\mu(R^i)$ is calculated using the maximum of the individual strengths. Thus, $\mu(R^1)$ and $\mu(R^2)$ are given by

$$\begin{aligned}\mu(R^1) &= \text{MAX}[\mu(\theta_1), \mu(\theta_2), \dots, \mu(\theta_n)] \\ \mu(R^2) &= \text{MAX}[\mu(\dot{\theta}_1), \mu(\dot{\theta}_2), \dots, \mu(\dot{\theta}_n)].\end{aligned}\tag{146}$$

From the above rules, it should be noted that the job of the supervisor is to weight each criterion but not to compute the individual redundancy control. Thus, there is no need for rules involving linguistic hedging such as "VERY NEAR" or "MOVING VERY FAST." Any hedging is handled by the individual controls. This allows the supervisor to be developed independently of the specific criteria.

The last component of the fuzzy supervisor is the choice for the fuzzification / de-fuzzification method. Since x^* normalizes x within the universe of discourse, fuzzification is already provided. For de-fuzzification, the center of gravity algorithm is used [82]. For q rules with strengths $\mu(R^i)$ and outputs u_i , the center of gravity algorithm calculates the crisp output by

$$\underline{u} = \frac{\sum_{i=1}^q \mu(R^i) u_i}{\sum_{i=1}^q \mu(R^i)}.\tag{147}$$

For comparison to fixed weight algorithms, equation (147) can be re-written as

$$\underline{u} = \frac{\mu(R^1)}{\sum_{i=1}^q \mu(R^i)} u_1 + \dots + \frac{\mu(R^q)}{\sum_{i=1}^q \mu(R^i)} u_q = w_1 u_1 + w_2 u_2 + \dots + w_q u_q.\tag{148}$$

This form shows that, as desired, the final output is a time-varying weighted sum of the individual controls. This form is also similar to Cleary's CPI. The primary difference is that here the weights are the importance values as measured by the membership functions and redundancy optimization solutions are used instead of the

values of the optimization cost function.

From a structural standpoint, equation (148) is also the engine of the weighting factor adaptation algorithm. Inspection of equation (148) shows that for each criterion or objective, the time varying weight is computed by dividing the individual importance by the summation of all of the importances.

7.3 Application to Test Model

Since low values of γ often produced unstable joint motion while high values did not, a FLS can potentially be used to improve stability by adaptively changing γ . To test this hypothesis, a FLS with two rules was implemented and applied to the three link test model. For implementation the original composite control problem is re-written as

$$\begin{aligned}\underline{u} &= \gamma \underline{u}_{SRM} + (1 - \gamma) \underline{u}_{CBM} \\ &= w_1 \underline{u}_1 + w_2 \underline{u}_2.\end{aligned}\tag{149}$$

By inspection, the fuzzy weights w_1 and w_2 correspond to the weighting factors γ and $1-\gamma$, respectively. Likewise, the two objective controls \underline{u}_1 and \underline{u}_2 correspond to the robot stabilizing control \underline{u}_{SRM} and the vibration suppression control \underline{u}_{CBM} . Since the primary objective is reducing flexible vibration while keeping the robot joints within certain limits, only two fuzzy logic rule are needed. Heuristically speaking, these rules can be stated as "Damp flexible vibration unless the joint velocities are too high." Two fuzzy rules corresponding to this heuristic are:

R¹: IF $d\theta_1/dt$ is near $(d\theta_1/dt)_{lim}$ OR $d\theta_2/dt$ is near $(d\theta_2/dt)_{lim}$ OR ... $d\theta_n/dt$ is near $(d\theta_n/dt)_{lim}$, THEN choose the redundancies to reduce joint velocities

R²: IF the base is vibrating, THEN choose the redundancies to damp flexible vibrations.

To complete the design of a FLS, the membership functions must be specified.

Since there are many ways to define and measure vibration, for the purposes of this study, the membership function for vibration was assumed to be

$$\mu(R^2) = \begin{cases} 0, & \underline{V}_b^T \underline{V}_b = 0 \\ 1, & \underline{V}_b^T \underline{V}_b \neq 0 \end{cases} \quad (150)$$

For R¹, the exponential membership function given in equation (144) was used. For sensitivity purposes, four different rate and truth factor pairs, listed in Table V and plotted in Figure 27, were investigated as a function of three maximum joint velocities: 145°/s, 90°/s, and 45°/s.

Table V. Rate and Truth Factors for Fuzzy Simulation

Case #	Rate Factor (b)	Truth Factor (d)
1	0.15	0.75
2	0.15	0.80
3	0.15	0.85
4	0.20	0.85

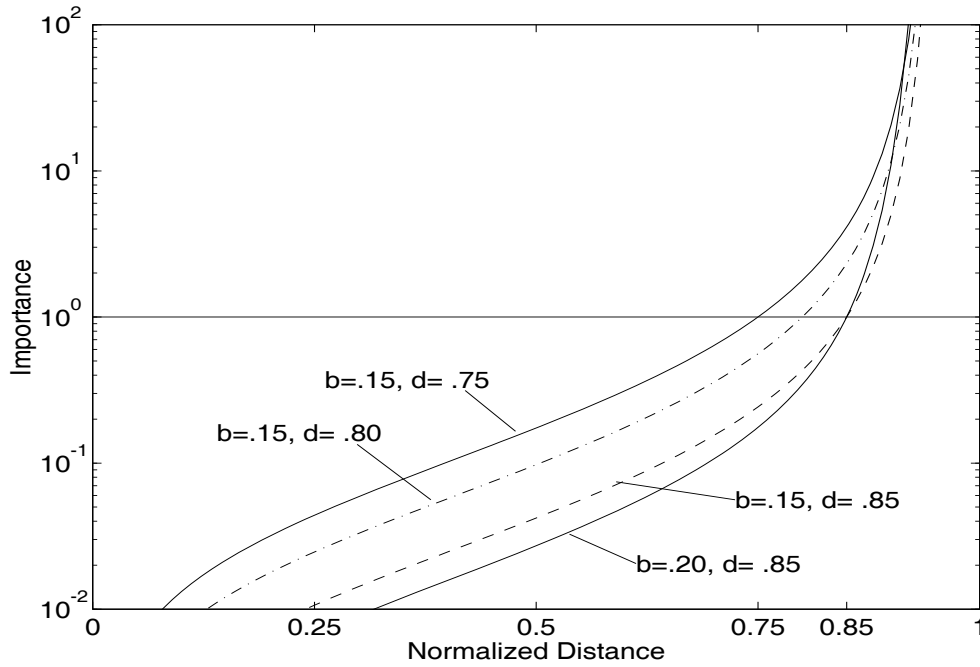


Figure 27. Fuzzy Membership Functions for Test Model.

To simplify matters, once again, only representative plots will be shown. These plots are the energy in the flexible base and the time variation in the weighting factor for $\dot{\theta}_{\max} = 135^\circ/\text{s}$ and $45^\circ/\text{s}$ for the RW-DE algorithm using W_1 and W_3 . A more complete set of plots is found in Appendix H.

Figure 28 and Figure 29 show the fuzzy results using W_1 with a maximum joint velocity of $135^\circ/\text{s}$. In the figure, 1, 2, ..., 4 denotes the case number for the fuzzy membership function parameters. For comparative purposes, the baseline and three fixed weight (non-fuzzy) solutions are also given. The three fixed weight solutions were for $\gamma = 1 \times 10^{-5}$, 1×10^{-2} , and 5×10^{-2} and correspond to the unstable motion, stable motion with a maximum joint velocity of $106^\circ/\text{s}$, and stable motion with a maximum joint velocity of $85^\circ/\text{s}$. As seen in Figure 28, adaptively choosing γ

yielded stable motion whose performance fell in between that of $\gamma = 1 \times 10^{-2}$ and 5×10^{-2} . During the maneuver, there was some variance amongst the fuzzy solutions, but afterwards they remained roughly the same. Inspection of the curves shows that performance appears to be linearly related to the truth factor. Indeed, for a fixed rate factor, increasing the truth factor decreased the energy absorbed by the base. This is attributed to the fact that higher truth factors delay importance which allows the redundancy to apply more effort into reducing flexible motion.

Comparing case numbers 3 to 4 also show that for the same truth factor, larger rate factors yield better performance. Examination of the curves for case 3 ($b=.15$ and $d=.85$) and case 4 ($b=.20$ and $d=.85$) in Figure 27 shows that again, this is due to delayed importance. Although cases 3 and 4 define importance at 0.85, case 4 has a lower importance profile from 0 to 0.85 which results in more vibration suppression.

Increased vibration suppression is also shown in Figure 29. Comparing Figure 28 and Figure 29 shows that reduced energy absorption was directly related to the amount of weighting. Inspection of the curves shows that, as was previously mentioned, smaller importances result in less robot stabilization and more vibration suppression weightings.

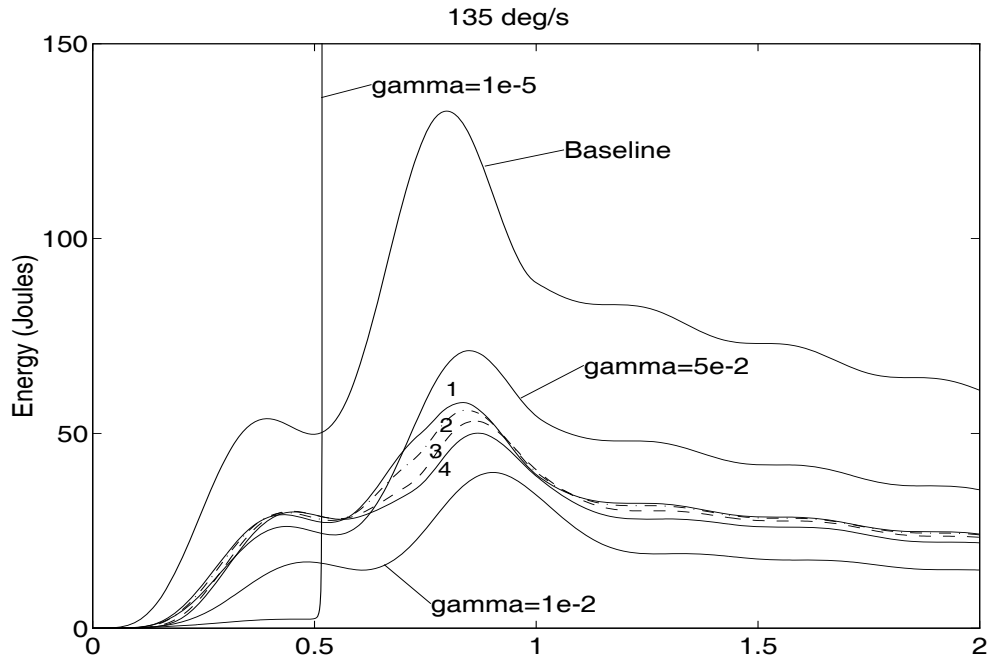


Figure 28. Energy in Flexible Base using Fuzzy and Non-Fuzzy Weighting with W_1 .

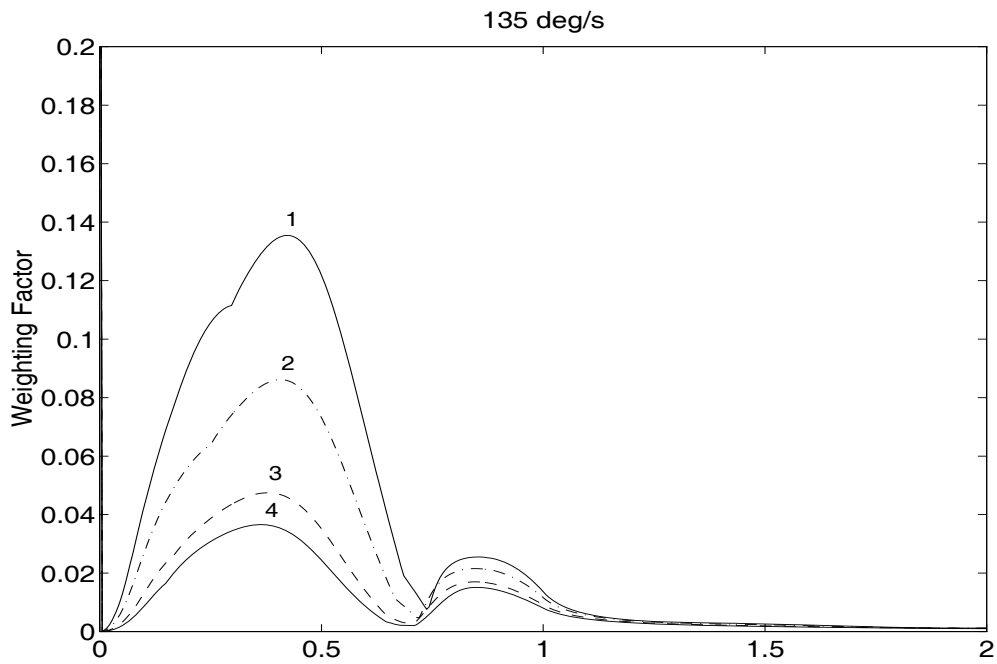


Figure 29. Weighting Factor Variation using Fuzzy Logic Supervisor with W_1 .

Similar results were obtained when using the inverse velocity squared robot joint velocity weighting matrix. Indeed, Figure 30 shows that for maximum robot joint velocities of 135°/s and 45°/s, delayed importance resulted in improved energy dissipation. Furthermore, compared to the results for W_1 , Figure 30a shows that once again, using the inertia matrix of the robot yields better vibration damping. The tradeoff is in the maximum robot joint velocities. The maximum velocities obtained for a velocity limit of 135°/s using the inverse squared velocity matrix were between 35°/s for case 1 and 60°/s for case 4; while the maximum velocities using the robot inertia matrix ranged from 65°/s for case 1 up to 80°/s for case 4.

If the maximum allowable velocity is decreased to 45°/s, Figure 30b shows that very little energy was dissipated.¹¹ Since this velocity limit is relatively low compared to the velocities naturally generated by the vibration control, larger weighting factors were required (see Figure 31b) which resulted in smaller vibration controls.

¹¹ Note, no curves are presented for W_1 with $\dot{\theta}_{\max} = 45^\circ/\text{s}$ because W_1 failed to keep the velocities below this limit. Inspection of Figure 23 for $\gamma = 1$ shows that this occurred because minimizing the robot kinetic energy required at least a 55°/s maximum.

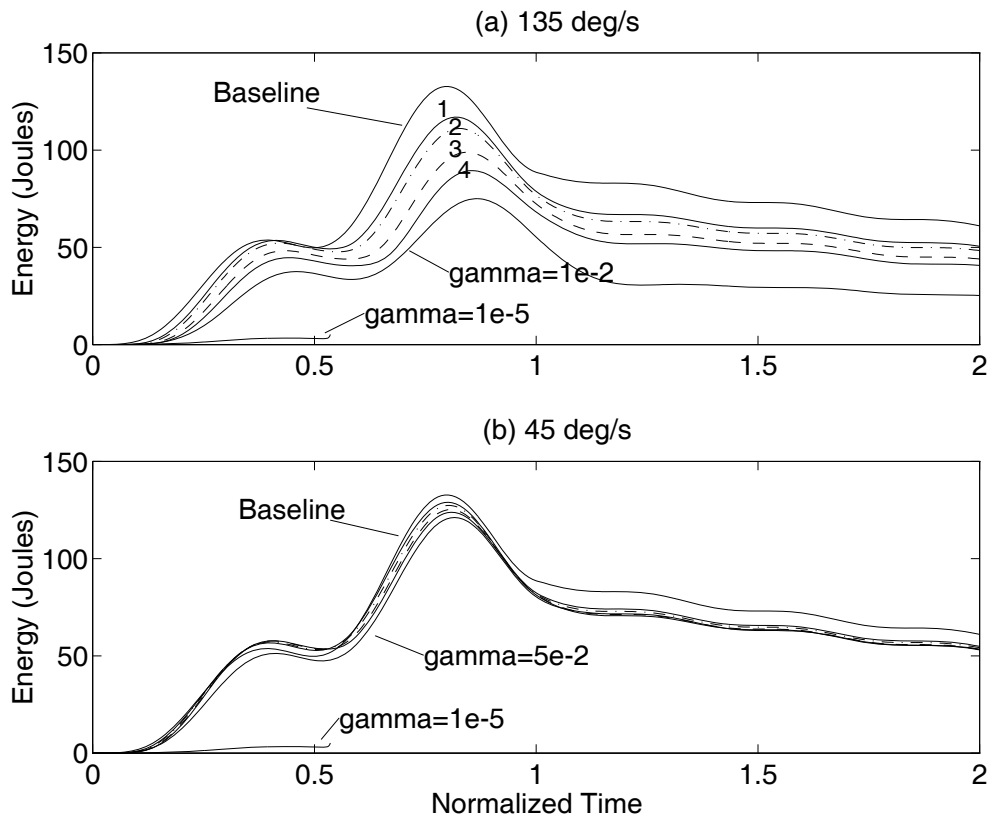


Figure 30. Energy in Flexible Base using Fuzzy and Non-Fuzzy Weighting with W_3 .

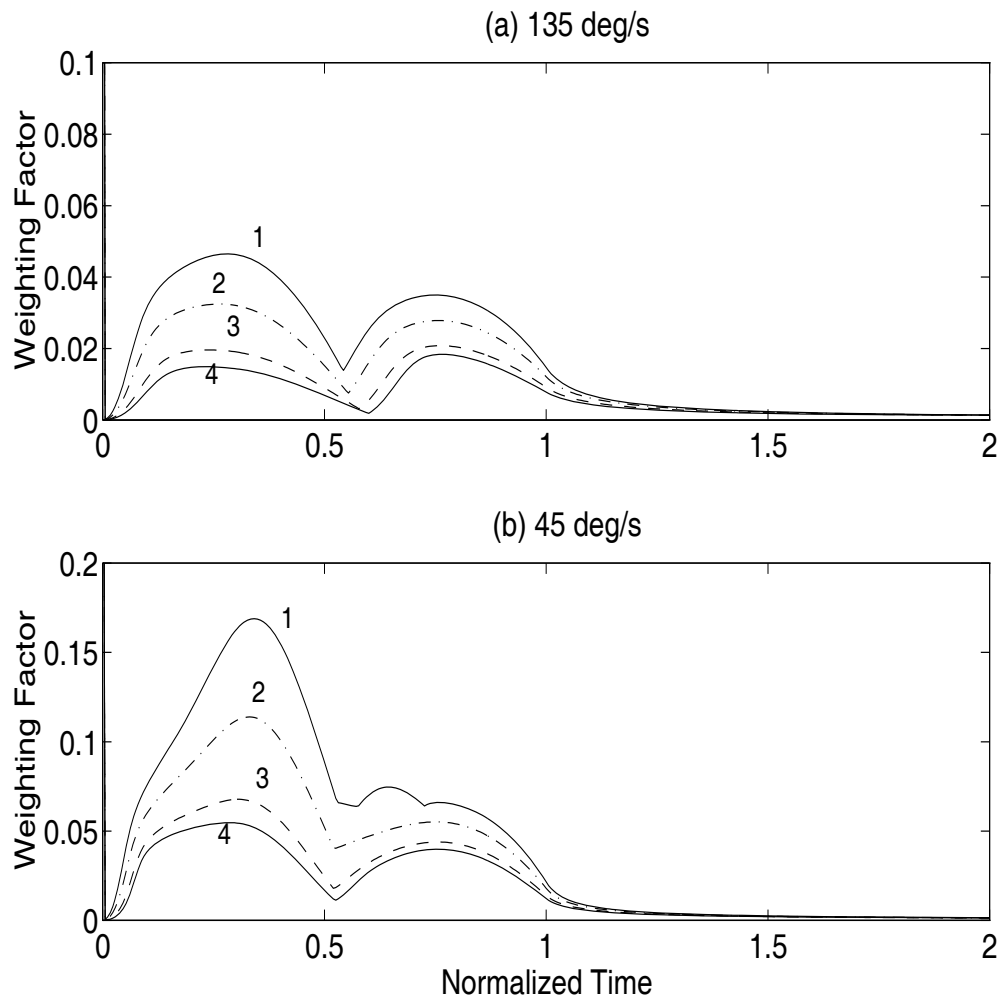


Figure 31. Weighting Factor Variation using Fuzzy Logic Supervisor with W_3 .

8. SUMMARY

To deal with dynamically changing environments and to overcome limitations imposed by non-redundant robots, future robotic applications will use kinematically redundant robots. One such application is aiding in space construction. Due to the high cost of space transportation, these robots will be attached to light, flexible bases. Since these bases vibrate when excited by the robot, accurately controlling the end-effector is a difficult problem. Thus, the purpose of this research was to investigate using the extra degrees of freedom in a kinematically redundant robot for vibration suppression.

From a theoretical, standpoint, this investigation was comprised of three parts: (1) deriving a model for the problem, (2) deriving vibration suppression redundancy control laws, and (3) designing a fuzzy logic supervisor to increase robustness. In the model derivation part, model sub-structuring was used to joint independently derived models for robot and the flexible base. Due to Coriolis torques and robot configuration terms, this model was shown to be a coupled set of non-linear, time varying ordinary differential equations. This is in contrast to standard "textbook" structural vibration problems which are linear and time invariant.

To circumvent these problems, the Gradient Projection Method (GPM) and the Minimization Method (MM) were used to develop four energy dissipation redundancy control algorithms. These algorithms were (1) GPM, a straightforward application of

the gradient projection method, (2) MM1, a straightforward application of the minimization method, 3) RPA-DE, the minimization method used to Reduce the effects of the Particular Acceleration while Dissipating Energy, and (4) RW-DE, the minimization method used to Reduce the applied Wrench while Dissipating Energy. The difference between these algorithms is as follows.

In GPM, the redundancies are selected by feeding back flexible velocities to use rate feedback to dissipate energy in the flexible base. In contrast, MM1 is a feedback linearization approach that chooses the redundancies by attempting to cancel all of the forces and torques which can add energy to the flexible base while also incorporating rate feedback. Since all of the information and computing power necessary for MM1 may not be available, RPA-DE and RW-DE use feedback linearization to cancel some of the terms in the force/torque applied by the robot on the base while still including rate feedback. In RPA-DE, the redundancies are calculated by only cancelling the components due to the particular acceleration of the robot joints necessary to move the end-effector. Similarly, in RW-DE, the redundancies are computed to cancel the entire applied force/torque vector.

Due to the limitations imposed by the Gradient Projection Method and the Minimization Method, none of these algorithms are truly dissipating in the sense that energy is always dissipated as a result of the redundancies. These limitations are as follows. In the Gradient Projection Method, the redundancies are chosen by ensuring that only the redundancy term is energy dissipation. In the Minimization Method, the control algorithms are totally energy dissipating only if the rank of the control

influence matrix is larger than the number of task variables. For one degree of redundancy robots, this implies that only one direction is controllable. When this is not true, the Minimization Method chooses redundancies which (1) have at least one term that is energy dissipating and (2) minimize the norm of a particular force/torque (e.g., the force/torque due to the particular acceleration). Although minimizing this force/torque is attractive from the standpoint of less input excitation, the resulting minimized force may actually increase rather than decrease energy in the flexible base.

Even when these limitations are acceptable, a greater problem exists for the redundancies. Similar to work done in the torque optimization problem, choosing the redundancies to only minimize flexible base motion does not guarantee that the robot motion will be stable. On the contrary, unstable robot joint velocities can result because while the redundancy is dissipating energy in the base, it may be increasing energy in the robot. This leads to high joint velocities and eventual instabilities.

To overcome this problem, composite control was used to develop a new redundancy control that attempts to simultaneously dissipate flexible base energy and stabilize the robot by adding robot joint dissipative torques. By minimizing a weighted robot joint velocity norm, these torques add robot joint velocity feedback and reduce the effects of the particular joint acceleration. Although the robot stabilization control effectively adds robot joint damping, the ensuing closed loop damping matrix is not symmetric, positive definite. Therefore, the closed-loop velocity matrix is not energy dissipating.

However, by imposing symmetry and positive definiteness, a constraint relation

can be developed which relates the flexible velocity feedback gain to the composite control weighting factor and removes any arbitrariness in its choice. An interesting consequence of this constraint is that if the primary goal is vibration reduction, both robot and flexible velocity feedback must be used or else the redundancies should be used to simply minimize the applied input force.

To analyze the performance of the control laws, numerical simulations were conducted for a three link planar robot attached to a two mode flexible base. The results demonstrated that if the composite weighting factor is properly chosen, kinematic redundancies can be used to reduce flexible motion when compared to the global minimum velocity norm solution. In terms of performance, GPM, which is a special case of MM1, RPA-DE, and RW-DE, produced the least amount of vibration suppression while RPA-DE and RW-DE yielded the greatest amount.

However, if the weighting factor was "too low", insufficient robot joint dissipative torques were generated which often lead to high joint velocities and unstable motion in RPA-DE, RW-DE, and MM1. Unstable motion did not occur in GPM for small weighting values because the closed-loop energy dissipation forces the redundancy to zero as the weighting factor decreases. On the other hand, in RPA-DE, RW-DE, and MM1, instabilities resulted because the flexible base energy dissipation solution drove the robot into singular or decreased controllability regions. In these regions, the rank of the controllability matrix fell sharply and was followed by a steep increase in the desired control torque. To counteract this problem, robot joint velocity damping was added by increasing the composite weighting factor. This reduced the

joint velocities and effectively kept the robot away from the singular regions.

Though this is a viable technique in numerical simulations, in practical applications, the weight cannot be chosen after the fact. As a result, an automated method for computing the weighting factor was needed. Due to its design simplicity and ease of implementation, fuzzy logic supervisor was used to develop this method.

9. CONCLUDING REMARKS

Kinematic redundancies offer the ability to perform several tasks at once. In this research kinematic redundancies were used to solve a new research problem, namely flexible base vibration suppression. Using theoretical developments and through numerical simulations, it was shown that kinematic redundancies can reduce flexible vibrations provided certain requirements are met. These requirements are (1) sufficient robot joint damping is present to prevent instable motion, (2) the flexible velocities are measurable, and (3) the rank of the control influence matrix is larger than the number of task space variables. If these conditions are not met (particularly conditions 1 and 3), the redundancies can still dissipate flexible energy but stability and performance cannot be guaranteed.

Numerical simulations were the only testing means available in this work. As a result, practical issues such as computation, system identification, and measurability were not considered to any great extent. Therefore, the next step in development should consider these as well other implementation issues.

REFERENCE LIST

1. Lavery, Dave, "Perspectives on Future Space Robotics," *Aerospace America*, May 1994, pp. 32-37.
2. Holt, K. and Deschrochers, A. A., "Disturbance Rejection for Space-Based Manipulators," *Proceedings NASA Workshop on Selected Topics in Robotics for Space Exploration*, March 1993, Hampton, VA, NASA-CP 10131, pg. 38-43.
3. Miller, Jeffrey H., Lawrence, Charles, and Rohn, Douglas A., "The Dynamic Effects of Internal Robots on Space Station Freedom," *AIAA Guidance, Navigation and Control Conference*, August 1991, pg. 1865-1878.
4. Vafa, Zia and Dubowsky, Steven, "Minimization of Spacecraft Disturbances in Space-Robotic Systems," *AAS 88-006*.
5. Vasquez, R.L., and Cannon, R.H., Jr., "Initial Experiments in Cooperative Manipulation From A Moving Platform," *Advances in Dynamics and Control of Flexible Spacecraft and Space-Based Manipulations*, ASME, November 25-30, 1990, pp. 43-48.
6. Book, Wayne J. and Lee, Soo Han, "Vibration Control of a Large Flexible Manipulator by a Small Robotic Arm," *1989 American Control Conference*, June 1989, Pittsburgh, PA.
7. Lee, Soo Han and Book, Wayne J., "Robot Vibration Control Using Inertial Damping Forces," *NASA-CR 186808*.
8. Barbieri, Enrique and Wang, Qinyou, "An Example of Optimal Set-point Relegation for Self-Motion Control in Redundant Flexible Robots," *Proc. 1990 IEEE Intl. Conference on Robotics and Automation*, pp 632-637.
9. Barbieri, Enrique and Özgüner, Ümit, "A Control Approach for Robots with Flexible Links and Rigid End-Effectors," *Proc. NASA Conference on Space Telerobotics*, January 1989, Pasadena, CA, pp. 41-50.
10. De Silva, C. W., Chung, C. L., and C. Lawrence, "Base Reaction Optimization of Robotic Manipulators for Space Applications," *Proc. 19th Intl. Symposium on Industrial Robots*, Sydney, Australia, November 1988, pp. 829-852.
11. De Silva, C. W., "Trajectory Design for Robotic Manipulators in Space Applications," *Journal of Guidance*, Vol. 14, No. 3, 1990.

12. De Silva, C. W., Chung, C. L., and C. Lawrence, "Base Reaction Optimization of Redundant Manipulators for Space Applications," Cambridge Univ. (England) Dept. of Engineering/F-INFENG/TR.13, June 1988.
13. Chung, C.L. and Desa, S., "A Global Approach for Using Kinematic Redundancy to Minimize Base Reactions of Manipulators," Carnegie Mellon University, CMU-RI-TR-89-9, March 1989.
14. Lin, N. J., and Quinn, R. D., "The Use of Locally Optimal Trajectory Management for Base Reaction Control of Robots in a Microgravity Environment," AIAA Guidance, Navigation and Control Conference, pp. 1879-1888, New Orleans, LA, 1991.
15. Rohn, Douglas, and Lawrence, Charles, "Reaction-Compensation Technology for Microgravity Laboratory Robots," NASA TM-103271.
16. Torres, Miguel A. and Dubowsky, Steven, "Minimizing Spacecraft Attitude Disturbance in Space Manipulator Systems," Journal of Guidance, Control, and Dynamics, Vol 15, No. 4, July-August 1992, pp. 1010-1017.
17. Dubowsky, Steven and Torres, Miguel A., "Path Planning for Space Manipulators to Minimize Spacecraft Attitude Disturbances," Proceedings of the 1991 IEEE Intl. Conference on Robotics and Automation, Sacramento, CA, April 1991, pp. 2522-2528.
18. Umetani, Yoji, and Yoshida, Kazuya, "Continuous Path Control of Space Manipulators Mounted on OMV," Acta Astronautica, Vol. 15, No. 12, 1987, pp. 981-986.
19. Nechev, Dragomir, Umetani, Yoji, and Yoshida, Kazuya, "Analysis of a Redundant Free Flying Spacecraft/Manipulator System," IEEE Transactions on Robotics and Automation, Vol. 8, No. 1, Feb. 1992.
20. Wiens, Gloria J., "Effects of Dynamic Coupling in Mobile Robotic Systems," 1989 World Conference on Robotics Research, Gaithersburg, MD, May 1989, pp.3-43 - 3-55.
21. Hootsman, Norbert A. and Dubowsky, Steven, "Large Motion Control of Mobile Manipulators Including Vehicle Suspension Characteristics," Proc. of 1991 IEEE Intl. Conf. on Robotics and Automation, Sacramento, CA, April 1991, pp.2336-2341.

22. Hootsman, Norbert A., Dubowsky, Steven, and Mo, Patrick Z., "The Experimental Performance of a Mobile Manipulator Control Algorithm," Proc. of 1992 IEEE Intl. Conf. on Robotics and Automation, Nice, France, May 1992, pp. 1948-1954.
23. Nenchev, Dragomir N., "Redundancy Resolution through Local Optimization: A Review," Journal of Robotic Systems, 1989, pp.769-798.
24. Craig, John J., Introduction to Robotics, Mechanics and Control, 2nd Edition, Addison-Wesley Publishing Co., 1989.
25. Selig, J. M., Introductory Robotics, Prentice Hall International, Copyright 1992.
26. Ben-Israel, Adi and Greville, Thomas N. E., Generalized Inverses: Theory and Applications, John Wiley and Sons, Inc., Copyright 1974.
27. Klein, Charles A. and Huang, Ching-Hsiang, "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-13, No. 3, March-April 1983, pp. 245-250.
28. Meirovitch, Methods of Analytical Dynamics, McGraw-Hill Book Co., Copyright 1970.
29. Siciliano, Bruno and Book, Wayne J., "A Singular Perturbation Approach to Control of Lightweight Flexible Manipulators," The Intl. Journal of Robotics Research, Vol. 7, No. 4, August 1988, pp. 79-90.
30. Wang, Ji C., "Synthesis of Active Vibration Control Systems for A Space Structure Test Article," Vibration Control of Mechanical, Structural, and Fluid-Structural Systems, ASME Annual Meeting, Dallas, TX, November 1990, pp.15-23.
31. Juang, J.-N. and Pappa, R.S., "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction," Journal of Guidance, Control, and Dynamics, Vol. 8, No. 5, pp. 620-627.
32. Smith, Kevin E. and Kienholz, David A., "Introduction to Admittance Modelling," CSA Engineering, Inc., Palo Alto, CA, Report No. 87-06-01.
33. Widrick, Timothy, "Determining the Effect of Modal Truncation and Modal Errors in Component Mode Synthesis Methods," M.S. Thesis, George Washington University, Washington, DC, July 1992.

34. Luh, J.Y.S., Walker, M.W., and Paul, R.P.C., "On-Line Computational Scheme for Mechanical Manipulators," *Journal of Dynamic Systems, Measurements, and Control*, Vol. 102, June 1980, pp. 69-76.
35. Walker, M.W. and Orin, D.E., "Efficient Dynamic Computer Simulation of Robotic Mechanisms," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 104, September 1982, pp. 205-211.
36. Hsu, Ping, Hauser, John, and Sastry, Shankar, "Dynamic Control of Redundant Manipulators," *Proc. of 1988 IEEE Intl. Conf. on Robotics and Automation*, Philadelphia, PA, April 1988, pp.183-187.
37. An, Chae H., Atkeson, Christopher G., and Hollerbach, John M., "Experimental Determination of the Effect of Feedforward Control on Trajectory Tracking Errors," *Proc. of 1986 IEEE Intl. Conf. on Robotics and Automation*, pp. 55-60.
38. Seraji, Hodayoun, "Configuration Control of Redundant Manipulators: Theory and Implementation," *IEEE Trans. on Robotics and Automation*, Vol. 5, No. 4, August 1989, pp. 472-490.
39. de Silva, C. W. and MacFarlane, A. G. J., Knowledge-Based Control with Application to Robots, Lecture Notes in Control and Information Sciences, Springer-Verlag, Copyright 1989.
40. Er, M. J., "Recent Developments and Futuristic Trends in Robot Manipulator Control," *ASIA-Pacific Workshop on Advances in Motion Control*, Singapore, July 1993, pp. 106-111.
41. Yoshikawa, Tsuneo, "Analysis and Control of Robot Manipulators with Redundancy," *The First Intl. Symp. on Robotics Research*, 1984, pp.735-747.
42. Walker, Ian D. and Marcus, Steven I. "Subtask Performance by Redundancy Resolution for Redundant Robot Manipulators," *IEEE Journal of Robotics and Automation*, Vol. 4, No. 3, June 1988, pp. 350-354.
43. Duke, John P. Webb, Steven, Vu, Hung, "Optimal Passive Control of Multi-Degree of Freedom Systems Using a Vibration Absorber," *AIAA Guidance, Navigation and Control Conf.*, Portland, OR, August 1990, pp.1657-1663.
44. Juang, Jer-Nan and Phan, Minh, "Robust Controller Designs for Second- Order Dynamic Systems: A Virtual Passive Approach," *NASA-TM 102666*, May 1990.

45. Bruner, A. M., Belvin, W. K., Horta, L. G., and Juang, J. -N, "Active Vibration Absorber for the CSI Evolutionary Model: Design and Experimental Results, AIAA 32nd Structures, Structural Dynamics & Materials Conf., Baltimore, MD, April, 1991.
46. Williams, Trevor, Xu, Jianfan, and Juang, Jer-Nan, "Design of Virtual Passive Controllers for Flexible Space Structures," AIAA/AAS Astrodynamics Conf., Hilton Head Island, SC, August 1992, 217-226.
47. Papadopoulos, Loukas, and Tolson, Robert H., "Admittance Model for the Shuttle Remote Manipulator System in Four Configurations," NASA CR-4555, Nov 1993.
48. von Flotow, Andreas H., "An Overview of Possible and Not-so-Possible Tasks for Active Control of Sound and Vibration," Intl. Technical Specialists Meeting, Rotorcraft Acoustics, Valley Forge, PA, October 1991.
49. Lee-Glauser, Gina, Juang, Jer-Nan, and Sulla, Jeffrey L., "Optimal Active Vibration Absorber: Design and Experimental Results," 34th AIAA/ASME/ASCE Structures, Structural Dynamics & Materials Conf., La Jolla, CA, April 1993, pp.3146-3155.
50. Bernard, Friedland, Control Systems Design: An Introduction to State-Space Methods, McGraw-Hill, Inc., Copyright 1986.
51. Joshi, S. P., Vincent, T. L., and Lin, Y. C., "Control for Energy Dissipation in Structures," 29th AIAA/ASME/ASCE Structures, Structural Dynamics & Materials Conf., Williamsburg, VA, April 1988, pp.479-490.
52. Balas, Mark J., "Direct Velocity Feedback Control of Large Space Structures," Journal of Guidance and Control, Vol. 2, No. 3, May-June 1978, pp. 252-253.
53. Hollerbach, John M. and Suh, Ki C., "Redundancy Resolution of Manipulators through Torque Optimization," IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, August 1987, pp. 308-316.
54. Suh, Ki C., and Hollerbach, John M., "Local versus Global Torque Optimization of Redundant Manipulators," Proc. of the 1987 IEEE International Conference on Robotics and Automation, pp. 619-624.

55. Kazerounian, K., and Wang, Z, "Global versus Local Optimization in Redundancy Resolution of Robotic Manipulators," *International Journal of Robotics Research*, Oct.1988, pp. 3-12.
56. Nedungadi, A., and Kazerounian, K., "A Local Solution with Global Characteristics for the Joint Torque Optimization of a Redundant Robot," *Journal of Robotic Systems*, May 1981, pp. 631-654.
57. Ma, Shugen, Hirose, Shigeo, and Nenchev, Dragomir, "Improving Local Optimization Techniques for Redundant Robotic Mechanisms," *Journal of Robotic Systems*, Sep. 1990, pp. 75-91.
58. Pueh, Lee Heow, "Motions with Minimal Joint Torques for Redundant Manipulators," *Journal of Mechanical Design*, Vol. 115, September 1993, pp. 599-603.
59. MATLAB, The MathWorks, Inc.
60. O. Khatib, "Dynamic Control of Manipulators in Operational Space," 6th IFToMM Congress on Theory of Machines and Mechanisms, New Delhi, India, 1983.
61. Singer, Neil C., and Seering, Warren P., "Preshaping Command Inputs to Reduce Telerobotic System Oscillations," *Proc. of NASA Conference on Space Telerobotics*, Vol. V, Jan 1989, pp. 53-62.
62. Singer, Neil C., and Seering, Warren P., "Preshaping Command Inputs to Reduce System Vibration," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 112, March 1990, pp.76-81.
63. Singer, Neil C., and Seering, Warren P., "Experimental Verification of Command Shaping Methods for Controlling Residual Vibration in Flexible Robots," *Proc. 1990 American Control Conference*, San Diego, CA, May 1990, pp. 1738-1744.
64. Cleary, Kevin, Nguyen, Loc, and Frisch, Harry, "A Demonstration of Vibration Reduction for aFlexible Beam Using Input Shaping," *Proc of 1992 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Raleigh, NC, July 1992, pp. 1711-1716.
65. Zadeh, L. A., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Trans. Syst., Man, and Cybernetics*, 1973, Vol. 3, pp. 28-44

66. Maciejewski, A. A. and Klein, C. A., "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," Intl. Journal of Robotics Research, Vol. 4, No. 3, 1985, pp. 109-117.
67. Nakaumura, Y., Hanafusa, H., and Yoshikawa, T., "Task-Based Redundancy Control of Robot Manipulators," Intl. Journal of Robotics Research, Vol. 6, No. 2, 1987, pp. 3-15.
68. Mohamed, A.A. and Chevallereau, C., "Resolution of Robot Redundancy in the Cartesian Space by Criteria Optimization," 1993 IEEE, pp. 646-651.
69. Seraji, H. and Colbaugh, R., "Singularity-Robustness and Task-Prioritization in Configuration Control of Redundant Robots," Proc. of 29th Conf. on Decision and Control, Dec. 1990, pp. 3089-3095.
70. Cleary, K., "Incorporating Multiple Criteria in the Operation of Redundant Manipulators," 1990 IEEE Intl. Conf. on Robotics and Automation, pp. 618-624.
71. Pamanes, J. A. and Zeghloul, S., "Optimal Placement of Robotic Manipulators Using Multiple Kinematic Criteria," 1991 IEEE Intl. Conf. on Robotics and Automation, pp. 993-938.
72. Seraji, H. and Colbaugh, R., " Improved Configuration Control for Redundant Robots," Journal of Robotic Systems, Vol. 7, No. 6, 1990, 897-928.
73. McGhee, S., Chan, T-F., Dubey, R.V., and Kress, R.L., "Simultaneous Optimization of Multiple Performance Criteria for a Redundant Manipulator," DSC-Vol. 49, Advances in Robotics, Mechatronics, and Haptic Interfaces, ASME 1993, pp. 239-245.
74. Quail, Alanna and Shaout, Adnan, "State-of-the-Art in Household Appliances Using Fuzzy Logic," The 2nd Intl. Workshop on Industrial Fuzzy Control and Intelligent Systems, College Station, TX, December 1992, pp. 204-211.
75. Palm, Rainer, "Control of a Redundant Manipulator Using Fuzzy Rules," Fuzzy Sets and Systems, Vol. 45, 1992, pp. 279-298.
76. Kim, Sung-Woo and Lee, Ju-Jang, "Resolved Motion Rate Control of Redundant Robots using Fuzzy Logic," 2nd IEEE Intl. Conf. on Fuzzy Systems, San Francisco, CA, March 1993, pp. 333-338.

77. Kim, Sung-Woo, Lee, Ju-Jang, and Sugisaka, Masanori, "Inverse Kinematics Solution Based on Fuzzy Logic for Redundant Manipulators," Proc. 1993 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, Yokohama, Japan, July 1993, pp. 904-909.
78. Chiang, K. H. and Bonissone, P. P., "Fuzzy Logic Control (FLC): A Knowledge-base System Perspective," Proc. SPIE Conf. on Applications of Fuzzy Logic Technology, Sept. 1993, pp. 8-20.
79. Garcia-Benitez, E., Yurkovich, S., and Passino, K. M., "A Fuzzy Supervisor for Flexible Manipulator Control," Proc. 1991 IEEE Intl. Symposium on Intelligent Control, Arlington, VA, August 1991, pp.37-41.
80. Kreisselmeier, G. and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," Intl. Federation of Active Controls Symposium on Computer-Aided Design of Control Systems, Zurich, Switzerland, August 1979.
81. Takagi, T. and Sugeno, M., "Fuzzy Identification of Systems and Its Applications to Modeling and Control," IEEE Trans. On Sys., Man, and Cyber., Vol. SMC-15, No. 1, Jan. 1985, pp. 116-132.
82. Jackson, A., "Fuzzy Logic vs Traditional Approaches to the Design of Microcontroller-Based Systems," Proc. 1994 IEEE Aerospace Sciences Conf., pp. 19-33.

APPENDIX A. Symbolic Expressions for the Joint Velocity and Acceleration of a Planar Three Link Robot

To use the additional degrees of freedom of a kinematically redundant robot to optimize secondary tasks requires knowing the non-linear kinematic relationships between the end-effector and the robot joints. For a three link robot, these relationships can easily be derived. Indeed, consider the three link revolute, revolute, revolute manipulator shown in Figure A1. In the figure, the base of the manipulator base is denoted by $\{X_0\}$, while the link coordinate frames are $\{X_i\}$, $i=1,2$, and 3. The axes of rotations are z_i , $i=1,2$, and 3. Because all three rotation axes are parallel, the payload is constrained to move in the X_0 - Y_0 frame. Since only two degrees of freedom are necessary to position the payload, this system has one degree of redundancy available for optimizing secondary tasks.

Using straightforward geometry, the position, velocity, and acceleration of the end-effector (x_e and y_e) are given by:

$$\begin{bmatrix} {}^0x_e \\ {}^0y_e \end{bmatrix} = \begin{bmatrix} a_1c\theta_1 + a_2c\theta_{12} + a_3c\theta_{123} \\ a_1s\theta_1 + a_2s\theta_{12} + a_3s\theta_{123} \end{bmatrix}, \quad (\text{A1})$$

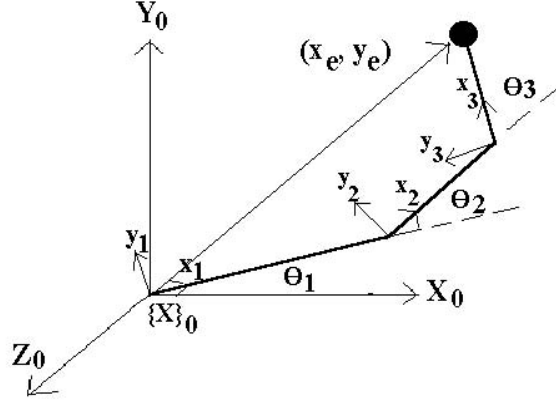


Figure A1. Three Link Planar, Revolute, Revolute, Revolute Robot.

$$\begin{aligned} \dot{\underline{x}}_e &= \frac{d}{dt} \begin{bmatrix} {}^0x_e \\ {}^0y_e \end{bmatrix} = \begin{bmatrix} -a_1s\theta_1 - a_2s\theta_{12} - a_3s\theta_{123} & -a_2s\theta_{12} - a_3s\theta_{123} & -a_3s\theta_{123} \\ a_1c\theta_1 + a_2c\theta_{12} + a_3c\theta_{123} & a_2c\theta_{12} + a_3c\theta_{123} & a_3c\theta_{123} \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix} \\ &\equiv {}^0\mathbf{J}\dot{\underline{\theta}}, \end{aligned} \quad (\text{A2})$$

and

$$\ddot{\underline{x}}_e = {}^0\mathbf{J}\dot{\underline{\theta}} + {}^0\mathbf{J}\ddot{\underline{\theta}} \quad (\text{A3})$$

where the time rate of change of the Jacobian matrix is

$$\dot{\mathbf{J}} = \begin{bmatrix} -\dot{\theta}_{123}a_3c\theta_{123} - \dot{\theta}_{12}a_2c\theta_{12} - \dot{\theta}_1a_1c\theta_1 & -\dot{\theta}_{123}a_3c\theta_{123} - \dot{\theta}_{12}a_2c\theta_{12} & -\dot{\theta}_{123}a_3c\theta_{123} \\ -\dot{\theta}_{123}a_3s\theta_{123} - \dot{\theta}_{12}a_2s\theta_{12} - \dot{\theta}_1a_1s\theta_1 & -\dot{\theta}_{123}a_3s\theta_{123} - \dot{\theta}_{12}a_2s\theta_{12} & -\dot{\theta}_{123}a_3s\theta_{123} \end{bmatrix} \quad (\text{A4})$$

and $c\theta_1 = \cos(\theta_1)$, $s\theta_1 = \sin(\theta_1)$, $c\theta_{23} = \cos(\theta_2 + \theta_3)$, $s\theta_{23} = \sin(\theta_2 + \theta_3)$, $c\theta_{123} = \cos(\theta_1 + \theta_2 + \theta_3)$, $s\theta_{123} = \sin(\theta_1 + \theta_2 + \theta_3)$, and a_i is the length of link i . It then follows that the robot joint velocities and accelerations are calculated as

$$\begin{aligned}
\dot{\underline{\theta}} &= {}^0\mathbf{J}^+\dot{\underline{x}}_e + \underline{\mathbf{S}}u \\
&= {}^0\mathbf{J}^+\dot{\underline{x}}_e + \begin{bmatrix} 1 \\ -\frac{a_1s\theta_{23}}{a_2s\theta_3} - 1 \\ \frac{a_1s\theta_2}{a_3s\theta_3} + \frac{a_1s\theta_{23}}{a_2s\theta_3} \end{bmatrix} u
\end{aligned} \tag{A5}$$

and

$$\begin{aligned}
\ddot{\underline{\theta}} &= {}^0\mathbf{J}^+(\ddot{\underline{x}}_e - {}^0\mathbf{J}\dot{\underline{\theta}}) + \underline{\mathbf{S}}u \\
&= {}^0\mathbf{J}^+(\ddot{\underline{x}}_e - {}^0\mathbf{J}\dot{\underline{\theta}}) + \begin{bmatrix} 1 \\ -\frac{a_1s\theta_{23}}{a_2s\theta_3} - 1 \\ \frac{a_1s\theta_2}{a_3s\theta_3} + \frac{a_1s\theta_{23}}{a_2s\theta_3} \end{bmatrix} u
\end{aligned} \tag{A6}$$

where ${}^0\mathbf{J}^+$ is the 3 x 2 pseudoinverse of the 2 x 3 Jacobian matrix ${}^0\mathbf{J}$ referenced to $\{X_0\}$, $\underline{\mathbf{S}}$ is the nullspace vector of \mathbf{J} , and u is an arbitrary scalar which due to the nullspace properties in $\underline{\mathbf{S}}$ does not affect the motion of the end-effector.

APPENDIX B. Symbolic Equations of Motion for a Three Link Robot on a Flexible Base

The governing equations of motion for an n_p degree of freedom flexible base with an attached n degree of freedom rigid robot are

$$\begin{aligned} I\ddot{\underline{p}} + D\dot{\underline{p}} + \Lambda\underline{p} &= -\phi_b^T \begin{bmatrix} R_{01} & 0 \\ 0 & R_{01} \end{bmatrix} {}^1\underline{W}_1 \\ &= -\phi_b^T (H_{p\theta} \ddot{\underline{\theta}} + H_{pp} \phi_b \ddot{\underline{p}} + \underline{N}_p) \end{aligned} \quad (\text{B1})$$

where

- \underline{p} is an $n_p \times 1$ vector of modal coordinates,
- $\underline{\theta}$ is the $n \times 1$ vector of robot joint coordinates,
- I is an $n_p \times n_p$ identity matrix,
- D is an $n_p \times n_p$ positive semi-definite damping matrix,
- Λ is an $n_p \times n_p$ diagonal matrix that contains the square of the natural frequencies of the flexible base without the robot,
- ϕ_b is the $6 \times n_p$ modeshape matrix for the mounting point between the robot and the flexible base in frame $\{X_b\}$,
- R_{01} is the 3×3 rotation matrix between coordinate frames $\{X_0\}$ and $\{X_1\}$,
- ${}^1\underline{W}_1$ is the 6×1 force/torque wrench vector acting on link 1,
- $H_{p\theta}$ is a $6 \times n$ control inertia matrix,
- H_{pp} is a 6×6 symmetric, positive definite inertia matrix, and
- \underline{N}_p is the 6×1 Coriolis and centrifugal wrench vector.

Assuming that the orientation between the robot base coordinate frame $\{X_0\}$ and the flexible base coordinate frame $\{X_b\}$ is zero, symbolically evaluating the recursive

Newton-Euler equations for a three link planar robot yields ¹²

$${}^1\underline{W}_1 = [{}^1F_{1,x} \quad {}^1F_{1,y} \quad 0 \quad 0 \quad 0 \quad {}^1\tau_{1,z}]^T, \quad (\text{B2})$$

$$H_{p\theta} = \begin{bmatrix} H_{p\theta}(1,1) & H_{p\theta}(1,2) & -s\theta_{23}(a_3m_4 + Pc_{3x}m_3) \\ H_{p\theta}(2,1) & H_{p\theta}(2,2) & c\theta_{23}(a_3m_4 + Pc_{3x}m_3) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ H_{p\theta}(6,1) & H_{p\theta}(6,2) & H_{p\theta}(6,3) \end{bmatrix}, \quad (\text{B3})$$

and

$$H_{pp} = \begin{bmatrix} c\theta_1 M & s\theta_1 M & 0 & 0 & 0 & H_{pp}(1,6) \\ -s\theta_1 M & c\theta_1 M & 0 & 0 & 0 & H_{pp}(2,6) \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ H_{pp}(6,1) & H_{pp}(6,2) & 0 & 0 & 0 & H_{pp}(6,3) \end{bmatrix}, \quad (\text{B4})$$

where ${}^1F_{1,x}$ = x-component of force on joint 1, ${}^1F_{1,y}$ = y-component of force on joint 1, ${}^1\tau_{1,z}$ = z-component of torque on joint 1, $c\theta_1 = \cos(\theta_1)$, $s\theta_1 = \sin(\theta_1)$, $c\theta_{23} = \cos(\theta_2+\theta_3)$, $s\theta_{23} = \sin(\theta_2+\theta_3)$, a_i , m_i and I_{zi} ($i=1,2,3$) are respectively, the link, the mass, and the inertia of link i , m_4 and I_{z4} , are respectively the mass and inertia of the payload, Pc_{ij} = center of mass location of link i along link i 's j -th axis, $M \equiv$

¹² A symbolic solution for \underline{N}_p is not given because during the evaluation of the recursive Newton-Euler equations, the string for \underline{N}_p exceeded the 10,000 character limit imposed by the MATLAB SYMBOLIC TOOLBOX and caused an error.

$m_1+m_2+m_3+m_4$, and $I_z \equiv I_{z1}+I_{z2}+I_{z3}+I_{z4}$. The non-zero elements of $H_{p\theta}$ and H_{pp} are:

$$H_{p\theta}(1,1) = -(s\theta_{23}a_3 + a_2s\theta_2)m_4 - (s\theta_{23}Pc_{3x} + a_2s\theta_2)m_3 - Pc_{2x}s\theta_2m_2 \quad (\text{B5})$$

$$H_{p\theta}(1,2) = -(s\theta_{23}a_3 + a_2s\theta_2)m_4 - (s\theta_{23}Pc_{3x} + a_2s\theta_2)m_3 - Pc_{2x}s\theta_2m_2 \quad (\text{B6})$$

$$H_{p\theta}(2,1) = (c\theta_{23}a_3 + a_1 + a_2c\theta_2)m_4 + (c\theta_{23}Pc_{3x} + a_1 + a_2c\theta_2)m_3 + (Pc_{2x}c\theta_2 + a_1)m_2 + m_1Pc_{1x} \quad (\text{B7})$$

$$H_{p\theta}(2,2) = (a_2c\theta_2 + c\theta_{23}a_3)m_4 + (c\theta_{23}Pc_{3x} + a_2c\theta_2)m_3 + Pc_{2x}c\theta_2m_2 \quad (\text{B8})$$

$$H_{p\theta}(6,1) = (a_3^2 + 2a_2c\theta_2a_1 + a_1^2 + a_2^2 + 2c\theta_3a_2a_3 + 2a_1a_3c\theta_{23})m_4 + (2a_2c\theta_2a_1 + a_1^2 + a_2^2 + Pc_{3x}^2 + (2c\theta_3a_2 + 2c\theta_{23}a_1)Pc_{3x})m_3 + (a_1^2 + Pc_{2x}^2 + 2Pc_{2x}c\theta_2a_1)m_2 + Pc_1x^2m_1 + I_z \quad (\text{B9})$$

$$H_{p\theta}(6,2) = (a_3^2 + a_2^2 + a_2c\theta_2a_1 + a_1a_3c\theta_{23} + 2c\theta_3a_2a_3)m_4 + ((c\theta_{23}a_1 + 2c\theta_3a_2)Pc_{3x} + a_2^2 + a_2c\theta_2a_1 + Pc_{3x}^2)m_3 + ((Pc_{2x}c\theta_2)a_1 + Pc_{2x}^2)m_2 + I_{z4} + I_{z3} + I_{z2} \quad (\text{B10})$$

$$H_{p\theta}(6,3) = (a_1a_3c\theta_{23} + a_3^2 + c\theta_3a_2a_3)m_4 + (Pc_{3x}^2 + (c\theta_3a_2 + c\theta_{23}a_1)Pc_{3x})m_3 + I_{z3} + I_{z4} \quad (\text{B11})$$

$$H_{pp}(1,6) = (-s\theta_2a_2 - s\theta_{23}a_3)m_4 + (-s\theta_{23}Pc_{3x} - s\theta_2a_2)m_3 - Pc_{2x}m_2s\theta_2 \quad (\text{B12})$$

$$H_{pp}(2,6) = (c\theta_{23}a_3 + a_2c\theta_2 + a_1)m_4 + (m_2 + m_3)a_1 + Pc_{2x}m_2c\theta_2 + (a_2c\theta_2 + c\theta_{23}Pc_{3x})m_3 + m_1Pc_{1x} \quad (\text{B13})$$

$$H_{pp}(6,1) = (-s\theta_{12}a_2 - s\theta_1a_1 - s\theta_{123}a_3)m_4 + (-m_3s\theta_1 - s\theta_1m_2)a_1 - m_1Pc_{1x}s\theta_1 - s\theta_{12}m_2Pc_{2x} + (-s\theta_{123}Pc_{3x} - s\theta_{12}a_2)m_3 \quad (\text{B14})$$

$$H_{pp}(6,2) = (c\theta_{12}a_2 + c\theta_1a_1 + c\theta_{123}a_3)m_4 + (c\theta_1m_2 + c\theta_1m_3)a_1 + m_1Pc_{1x}c\theta_1 + c\theta_{12}m_2Pc_{2x} + (c\theta_{123}Pc_{3x} + c\theta_{12}a_2)m_3 \quad (\text{B15})$$

$$H_{pp}(6,6) = (a_1^2 + 2c\theta_3a_2a_3 + a_3^2 + a_2^2 + (2a_2c\theta_2 + 2c\theta_{23}a_3)a_1)m_4 + (m_2 + m_3)a_1^2 + (2Pc_{2x}m_2c\theta_2 + (2a_2c\theta_2 + 2c\theta_{23}Pc_{3x})m_3)a_1 + (2c\theta_3a_2Pc_{3x} + Pc_{3x}^2 + a_2^2)m_3 + Pc_{2x}^2m_2 + I_z + Pc_{1x}^2m_1 \quad (\text{B16})$$

APPENDIX C. Flexible Energy Dissipation using Acceleration Measurements

If the acceleration ${}^b\mathbf{A}_b$ is measured instead of the velocity ${}^b\mathbf{V}_b$, energy dissipation can still be accomplished. One way to accomplish this is to use a virtual controller developed for non-linear systems [C1]. Following the developments in [C1], a redundancy control algorithm is developed as follows.

First, define a virtual physical system composed of virtual masses, springs, and dampers that is attached to the actual flexible base system. As the name implies, this system is not a real system. Instead, the virtual system is control algorithm that can be viewed in terms of a physical mass-spring-damper system. Using this concept, the total kinetic and potential energy of the combined flexible base/virtual system for n_c virtual masses are

$$T = \frac{1}{2} \dot{\mathbf{p}}^T (I + \Phi_b^T H_{pp} \Phi_b) \dot{\mathbf{p}} + \frac{1}{2} (\Phi_b \dot{\mathbf{p}} + \dot{\mathbf{x}}_c)^T M_c (\Phi_b \dot{\mathbf{p}} + \dot{\mathbf{x}}_c) \quad (\text{C1})$$

and

$$V = \frac{1}{2} \mathbf{p}^T \Lambda \mathbf{p} + \frac{1}{2} \mathbf{x}_c^T \Lambda_c \mathbf{x}_c \quad (\text{C2})$$

where the new symbols are \mathbf{x}_c , the $n_c \times 1$ virtual position vector which is measured relative to the flexible base, M_c , a symmetric, positive definite $n_c \times n_c$ virtual mass matrix, and Λ_c , a symmetric, positive semi-definite $n_c \times n_c$ virtual stiffness matrix. For simplicity, assume that the number of virtual controller modes n_c equals the number of

task space variables, m . Thus, for 6 degree of freedom systems, $n_c=6$.

As before, the goal for the r redundancies is to dissipate energy in the flexible base. If $E_{vc} \equiv T + V$ and \underline{A}_b is the acceleration of the attachment point X_b on the flexible base, energy dissipation requires that the time derivative of E_{vc} given by,

$$\dot{E}_{vc} = -\underline{V}_b^T (H_{p\theta} J^T (\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}}) + B\underline{u} + \underline{N}_p - G\dot{\underline{\theta}}) + (\underline{V}_b + \underline{V}_c)^T M_c (\underline{A}_b + \ddot{\underline{x}}_c) + \dot{\underline{x}}_c^T \Lambda_c \dot{\underline{x}}_c \quad (C3)$$

be less than zero. To make $\dot{E}_{vc} < 0$, first assume that the $m \times r$ control influence matrix B is such that \underline{u} can be chosen to ensure that

$$H_{p\theta} J^T (\ddot{\underline{x}}_e - \dot{J}\dot{\underline{\theta}}) + B\underline{u} + \underline{N}_p - G\dot{\underline{\theta}} = M_c (\underline{A}_b + \ddot{\underline{x}}_c). \quad (C4)$$

If this is possible, the time rate of change of E_{vc} becomes

$$\begin{aligned} \dot{E}_{vc} &= -\underline{V}_b^T M_c (\underline{A}_b + \ddot{\underline{x}}_c) + (\underline{V}_b + \dot{\underline{x}}_c)^T M_c (\underline{A}_b + \ddot{\underline{x}}_c) + \dot{\underline{x}}_c^T \Lambda_c \dot{\underline{x}}_c \\ &= \dot{\underline{x}}_c^T (M_c (\underline{A}_b + \ddot{\underline{x}}_c) + \Lambda_c \dot{\underline{x}}_c). \end{aligned} \quad (C5)$$

Inspection of equation (C5) then shows that energy will always be dissipated if the virtual dynamics are chosen such that

$$M_c (\underline{A}_b + \ddot{\underline{x}}_c) + \Lambda_c \dot{\underline{x}}_c = -D_c \dot{\underline{x}}_c \quad (C6)$$

which can be re-arranged to yield

$$M_c \ddot{\underline{x}}_c + D_c \dot{\underline{x}}_c + \Lambda_c \dot{\underline{x}}_c = -M_c \underline{A}_b. \quad (C7)$$

Thus, in the absence of velocity measurements, a virtual controller can be used to dissipate energy if the rank of the control influence matrix B is greater than m so that equation (C4) can be identically satisfied to give

$$\underline{u} = \mathbf{B}^{-1} \left(\mathbf{M}_c (\underline{A}_b + \ddot{\underline{x}}_c) + \mathbf{G}\underline{\theta}/2 - \mathbf{H}_{p\theta} \mathbf{J}^T (\ddot{\underline{x}}_e - \dot{\mathbf{J}}\dot{\underline{\theta}}) + \underline{N}_p \right) \quad (\text{C8})$$

where the virtual dynamics are governed by equation (C7) and the control parameters are \mathbf{M}_c , \mathbf{D}_c , and Λ_c .

[C1] Juang, J-N, Wu, S-C, Phan, Minh, and Longman, Richard, "Passive Dynamic Controllers for Nonlinear Mechanical Systems," NASA TM-104047, March 1991.

APPENDIX D. Proof of Positive Definiteness of Closed-Loop Damping Matrix for the GPM

For an n degree of freedom robot with m task variables, the $(n + m) \times (n + m)$ closed-loop damping matrix for the GPM with partitioned control is

$$D_{GPM} = \begin{bmatrix} \gamma W_\theta S (S^T W S)^{-1} S^T W / \Delta t & -(1-\gamma) W_\theta S k_{gpm} B^T \phi_b \\ -\gamma \phi_b^T B (S^T W S)^{-1} S^T W / \Delta t & (1-\gamma) \phi_b^T B k_{gpm} B^T \phi_b \end{bmatrix}. \quad (D1)$$

Using symmetry arguments, the symmetric $n \times n$ one-step ahead weighting matrix W and the $r \times r$ velocity feedback gain k_{GPM} are

$$W = W_\theta \quad (D2)$$

and

$$k_{gpm} = \frac{\gamma}{(1-\gamma) \Delta t} (S^T W S)^{-1}. \quad (D3)$$

Therefore, the expression for the closed-loop damping matrix expression can be re-written as

$$D_{GPM} = \frac{\gamma}{\Delta t} \begin{bmatrix} W_\theta S (S^T W_\theta S)^{-1} S^T W_\theta & -W_\theta S (S^T W_\theta S)^{-1} B^T \phi_b \\ -\phi_b^T B (S^T W_\theta S)^{-1} S^T W_\theta & \phi_b^T B (S^T W_\theta S)^{-1} B^T \phi_b \end{bmatrix}. \quad (D4)$$

For D_{GPM} to be positive semi-definite, the following condition must be true for any arbitrary $(n \times 1)$ \underline{x} and $(m \times 1)$ \underline{y} vectors:

$$\begin{bmatrix} \mathbf{x}^T & \mathbf{y}^T \end{bmatrix} \mathbf{D}_{GPM} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \geq 0. \quad (\text{D5})$$

Substituting in \mathbf{D}_{GPM} yields

$$\begin{aligned} \begin{bmatrix} \mathbf{x}^T & \mathbf{y}^T \end{bmatrix} \mathbf{D}_{GPM} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} &= \mathbf{x}^T \mathbf{W}_\theta \mathbf{S} (\mathbf{S}^T \mathbf{W}_\theta \mathbf{S})^{-1} \mathbf{S}^T \mathbf{W}_\theta \mathbf{x} + \mathbf{y}^T \boldsymbol{\phi}_b^T \mathbf{B} (\mathbf{S}^T \mathbf{W}_\theta \mathbf{S})^{-1} \mathbf{B}^T \boldsymbol{\phi}_b \mathbf{y} \\ &\quad - 2 \mathbf{x}^T \mathbf{W}_\theta \mathbf{S} (\mathbf{S}^T \mathbf{W}_\theta \mathbf{S})^{-1} \mathbf{B}^T \boldsymbol{\phi}_b \mathbf{y} \end{aligned} \quad (\text{D6})$$

which simplifies to

$$\begin{aligned} \begin{bmatrix} \mathbf{x}^T & \mathbf{y}^T \end{bmatrix} \mathbf{D}_{GPM} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} &= \left((\mathbf{S}^T \mathbf{W}_\theta \mathbf{S})^{-1/2} \mathbf{S}^T \mathbf{W}_\theta \mathbf{x} - (\mathbf{S}^T \mathbf{W}_\theta \mathbf{S})^{-1/2} \mathbf{B}^T \boldsymbol{\phi}_b \mathbf{x} \right)^T \cdot \\ &\quad \left((\mathbf{S}^T \mathbf{W}_\theta \mathbf{S})^{-1/2} \mathbf{S}^T \mathbf{W}_\theta \mathbf{x} - (\mathbf{S}^T \mathbf{W}_\theta \mathbf{S})^{-1/2} \mathbf{B}^T \boldsymbol{\phi}_b \mathbf{x} \right) \geq 0. \end{aligned} \quad (\text{D7})$$

Since this expression is greater than or equal to zero, \mathbf{D}_{GPM} is positive semi-definite.

APPENDIX E. Choosing the Velocity Feedback Gain for Minimization Method

Redundancy Solutions

The closed-loop equations of motion for a rigid robot on a flexible can be written as

$$\begin{bmatrix} W_\theta & 0 \\ 0 & I + \phi_b^T H_{pp} \phi_b \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{p} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{p} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \Lambda \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} W_\theta \\ -\phi_b^T H_{p\theta} \end{bmatrix} J^+ (\ddot{x}_e - \dot{J}\dot{\theta}) + \begin{bmatrix} W_\theta S \\ -\phi_b^T B \end{bmatrix} \underline{u} + \begin{bmatrix} 0 \\ -\phi_b^T N_p \end{bmatrix}, \quad (\text{E1})$$

where W_θ is a to-be-determined symmetric, positive definite inertia matrix not necessarily equal to the inertia matrix M_θ of the robot. For the MM, the partitioned control redundancy solution, from equations (82), (100), and (104), is

$$\begin{aligned} \underline{u}_{MMI} = & (1-\gamma)(B^T B)^{-1} B^T (k_{MMI} \phi_b \dot{p} - H_{p\theta} J^T (\ddot{x}_e - \dot{J}\dot{\theta}) - N_p) \\ & - \gamma (S^T W S)^{-1} S^T W \cdot (\dot{\theta} / \Delta t + J^T (\ddot{x}_e - \dot{J}\dot{\theta})). \end{aligned} \quad (\text{E2})$$

Using this relation, the closed-loop force generated by the redundancies is

$$\begin{bmatrix} W_\theta S \\ -\phi_b^T B \end{bmatrix} \underline{u}_{MMI} = -D_{MMI} \begin{bmatrix} \dot{\theta} \\ \dot{p} \end{bmatrix} - \begin{bmatrix} W_\theta S \\ -\phi_b^T B \end{bmatrix} \underline{v}_{MMI} \quad (\text{E3})$$

where the closed-loop damping matrix D_{MMI} and non-linear force \underline{v}_{MMI} are

$$D_{MMI} = \begin{bmatrix} \frac{\gamma}{\Delta t} W_\theta S (S^T W S)^{-1} S^T W & -(1-\gamma) W_\theta S k_{mm1} (B^T B)^{-1} B^T \phi_b \\ -\frac{\gamma}{\Delta t} \phi_b^T B (S^T W S)^{-1} S^T W & (1-\gamma) \phi_b^T B k_{mm1} (B^T B)^{-1} B^T \phi_b \end{bmatrix} \quad (\text{E4})$$

and

$$\underline{v}_{MMI} = \left(\left\{ (1-\gamma) B^+ H_{p\theta} + \gamma (S^T W S)^{-1} S^T W \right\} J^+ (\ddot{x}_e - \dot{J}\dot{\theta}) + (1-\gamma) B^+ \underline{N}_p \right). \quad (\text{E5})$$

For energy dissipation, D_{MMI} must be symmetric and positive semi-definite. With respect to equation (E4), symmetry and positive semi-definiteness are met if

$$W = W_\theta \quad (\text{E6})$$

and k_{MMI} is a scalar given by

$$k_{MMI} = \frac{\gamma B^T B}{(1-\gamma) \Delta t (S^T W_\theta S)}. \quad (\text{E7})$$

APPENDIX F. Trajectory Planning Problem

In this research, the trajectory of the end-effector was specified by

$$\underline{x}_e(t) = \underline{A} \sin(2\pi t/T_m) + \underline{B}t + \underline{C} \quad (\text{F1})$$

with the boundary conditions:

$$\begin{aligned} \underline{x}_e(0) &= \underline{x}_0 \\ \dot{\underline{x}}_e(0) &= \mathbf{0} \\ \underline{d}_m &\equiv \underline{x}_e(t_m) - \underline{x}_e(0). \end{aligned} \quad (\text{F2})$$

where \underline{x}_e is the 2 x 1 vector of the end-effector's relative x and y locations, T_m is the time of the end of the maneuver or trajectory, \underline{d}_m is a 2 x 1 vector of the desired move distance, and \underline{A} , \underline{B} , and \underline{C} are 2 x 1 to-be-determined constants. Because the redundancy are resolved at the acceleration level, the end-effector trajectory must be given in terms of the acceleration of the end-effector, i.e.

$$\ddot{\underline{x}} = -\underline{A}(2\pi/T_m)^2 \sin(2\pi t/T_m). \quad (\text{F3})$$

As a result, \underline{A} must be written in terms of the boundary conditions. Using equations (F1) and (F2), \underline{A} is $-\underline{d}_m/(2\pi)$. Therefore, the desired end-effector trajectory is

$$\ddot{\underline{x}} = (2\pi/T_m^2) \underline{d}_m \sin(2\pi t/T_m). \quad (\text{F4})$$

APPENDIX G. Sensitivity of Tracking Error, Control Effort, and Maximum Joint Velocity to the Velocity Weighting Matrix

To stabilize the motion of the robot, robot joint dissipative forces are added by choosing part of \underline{u} to minimize the weighted, one step ahead robot joint velocity norm, L_k defined by

$$L_k = \dot{\underline{\theta}}_{k+1}^T W \dot{\underline{\theta}}_{k+1} \quad (\text{G1})$$

which can also be written as

$$L_k = \left\| W^{1/2} (\dot{\underline{\theta}} + \Delta t J^+ (\ddot{\underline{x}}_e - \dot{J} \dot{\underline{\theta}}) + \Delta t S \underline{u}) \right\|^2 \quad (\text{G2})$$

where Δt is the sampling time and W is a to-be-determined velocity weighting matrix.

As part of this research, three velocity weighting matrices,

$$\begin{aligned} 1) W &\equiv W_1 = M_\theta(\theta) \\ 2) W &\equiv W_2 = M_\theta(\theta_0) \\ 3) W &\equiv W_3 = \text{diag} \left[sf * (1 - \dot{\theta}_i / \dot{\theta}_{i,\max})^{-2} \right] \end{aligned} \quad (\text{G3})$$

were evaluated. For scaling purposes, the scaling factor 'sf' in W_3 was set to one third of the trace of $M_\theta(\theta_0)$. This choice gives W_3 the units of inertia and scales it in proportion to the initial inertia matrix. For the fixed weight study, the maximum joint velocity was set to 135°/s.

To investigate the performance sensitivity of the redundancy algorithms, the RW-

DE solution, given by

$$\underline{u}_{GPM} = \frac{\Upsilon}{\Delta t} (S^T W_i S)^{-1} (B^T \underline{V}_b - S^T W_i \dot{\underline{\theta}}) - \gamma (S^T W_i S)^{-1} S^T W_i J^+ (\ddot{\underline{x}}_e - \dot{J} \dot{\underline{\theta}}) \quad (G4)$$

$$\underline{u}_{RW-DE} = \underline{u}_{gpm} - (1 - \gamma) B^+ (H_{p0} J^+ (\ddot{\underline{x}}_e - \dot{J} \dot{\underline{\theta}}) + \underline{N}_p). \quad (G5)$$

where W_i is one of the three weighting matrices was implemented for several values of γ . Figures G1 through G3 show sensitivity plots of the mean tracking error (e), the control effort (CE), and the maximum joint velocity where e and CE are defined as follows:

$$e = \frac{1}{N} \sum_{i=1}^N \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \quad (G6)$$

and

$$CE \equiv \frac{1}{N} \sum_{i=1}^N \underline{\tau}_i^T \underline{\tau}_i. \quad (G7)$$

where N is the number of simulation data points, $\underline{\tau}$ is the robot joint torque vector, x_i , y_i are respectively, the desired inertial x and y locations at time step i , and \hat{x}_i and \hat{y}_i are the actual inertial x and y end-effector locations. For scaling purposes, the curves have been normalized by the zero redundancy solution.

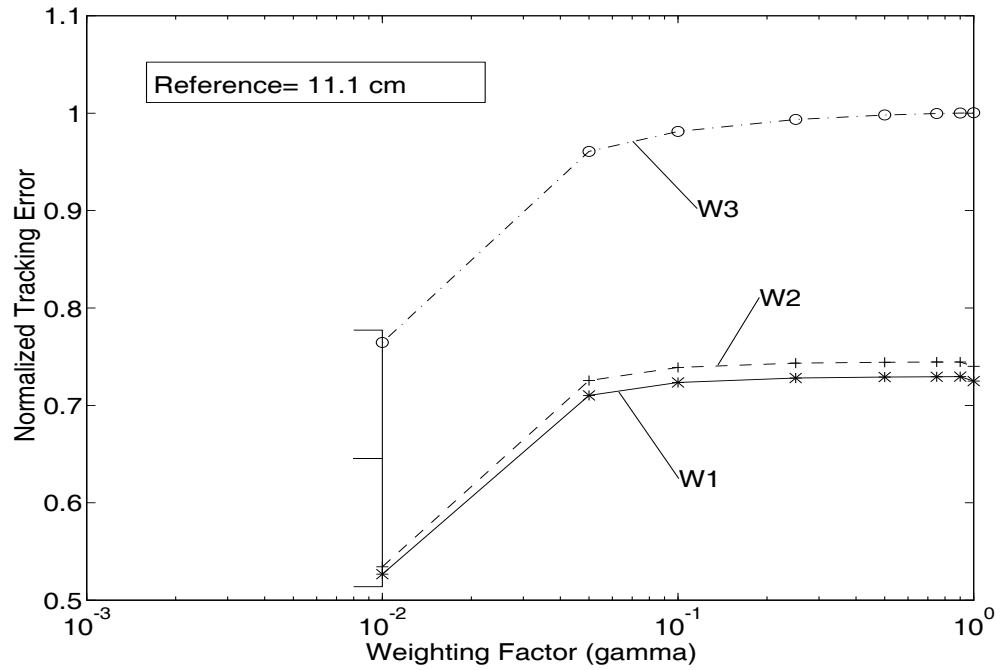


Figure G1. Sensitivity of Mean Tracking Error to γ and W in Trajectory Y.

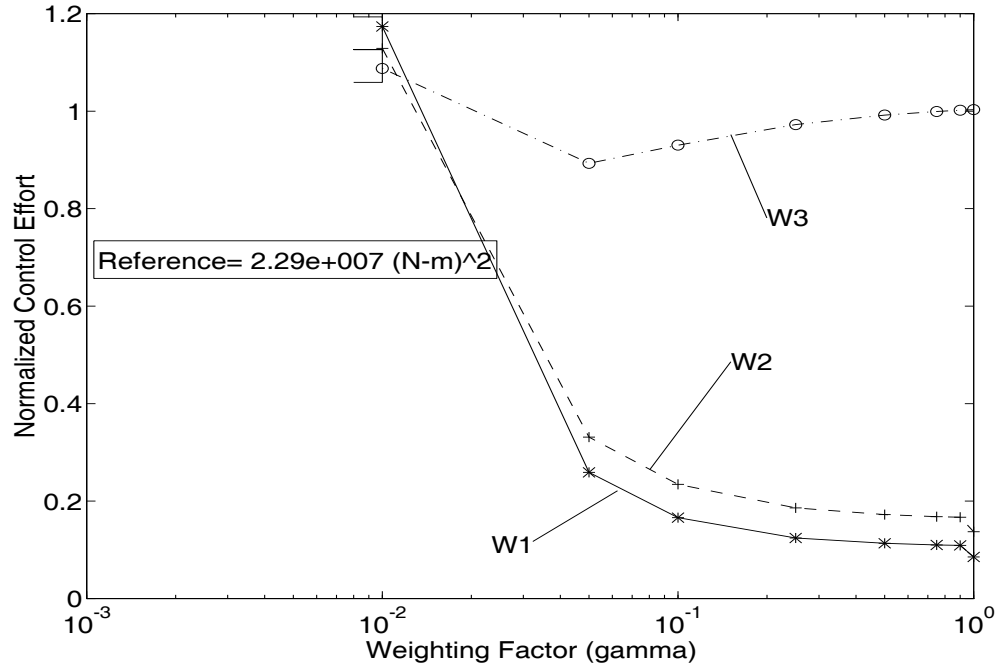


Figure G2. Sensitivity of Control Effort to γ and W in Trajectory Y.

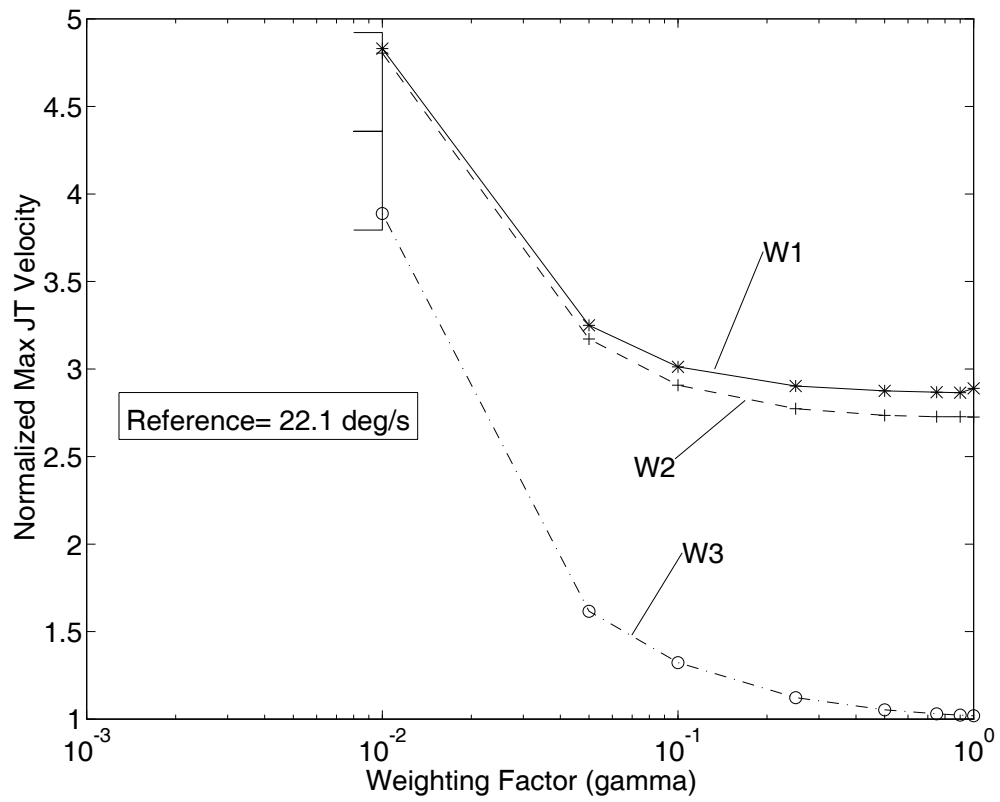


Figure G3. Sensitivity of Maximum Joint Velocity to γ and W in Trajectory Y.

APPENDIX H. Sensitivity Curves for Fuzzy Supervisor

To investigate sensitivity, 3-d sensitivity plots (Figures H1 through H3) were made of the normalized performance criteria for the FLS results. In these plots, the x axis is labelled 1 through 4 corresponding to fuzzy case number 1 through 4 where the differences between the cases are the rate and truth factors listed in Table V. Similarly, the y axis of the plots corresponds to the robot joint velocity limit. On each plot, six data points, labelled A through F, are annotated. Below, Table VI gives the values for these points.

Table VI. Normalized Data Points for Fuzzy Supervisor Sensitivity Plots.

Weighting Matrix	Letter	Figure H1	Figure H2	Figure H3
W_1	A	0.65	3.4	0.99
W_1	B	0.70	2.9	0.40
W_1	C	0.70	2.9	0.29
W_3	D	0.83	2.6	0.95
W_3	E	0.94	1.6	0.89
W_3	F	0.97	1.4	0.91

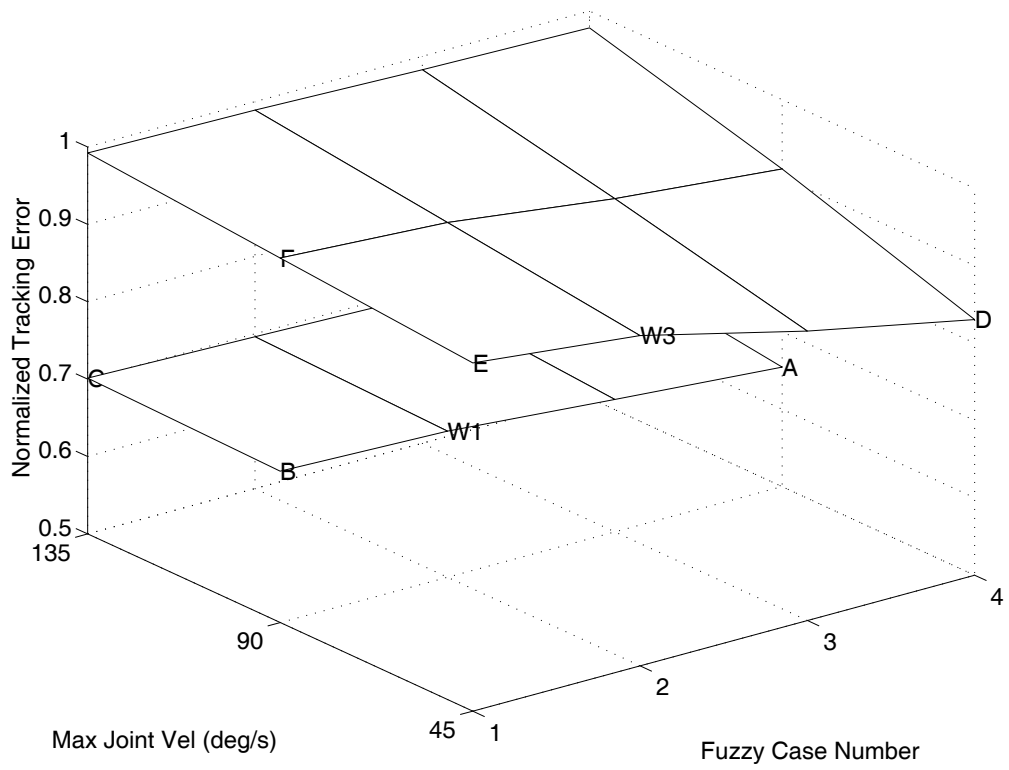


Figure H1. Tracking Error Sensitivity for Fuzzy Supervisor in Trajectory Y, RW-DE.

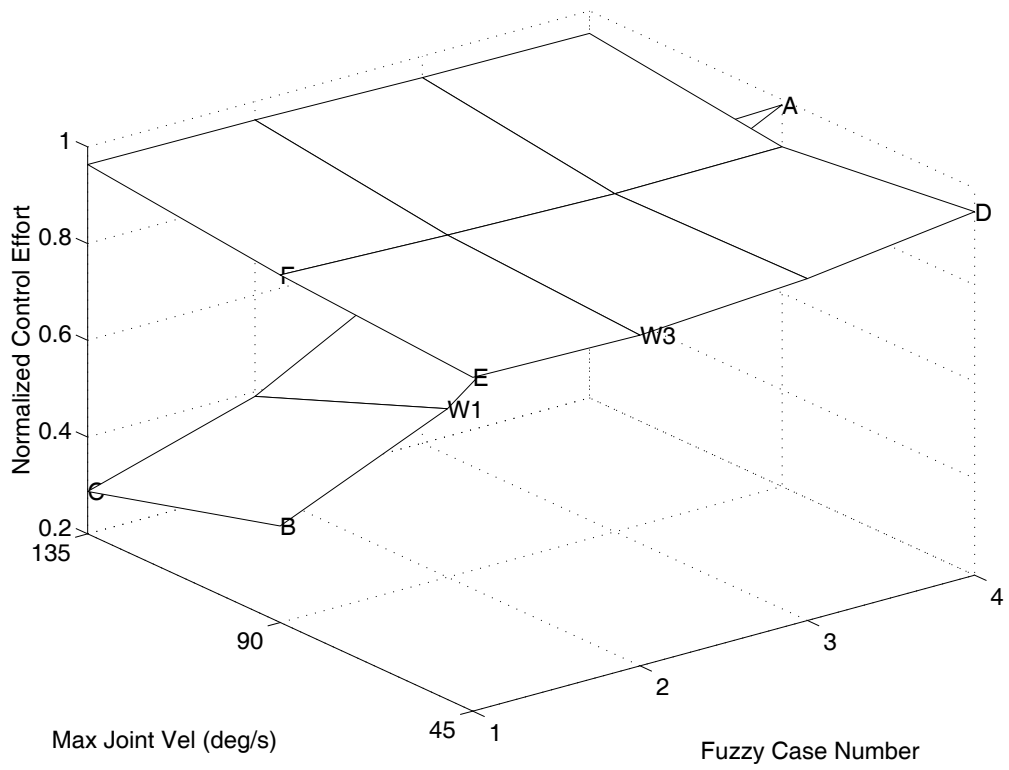


Figure H2. Control Effort Sensitivity for Fuzzy Supervisor in Trajectory Y, RW-DE.

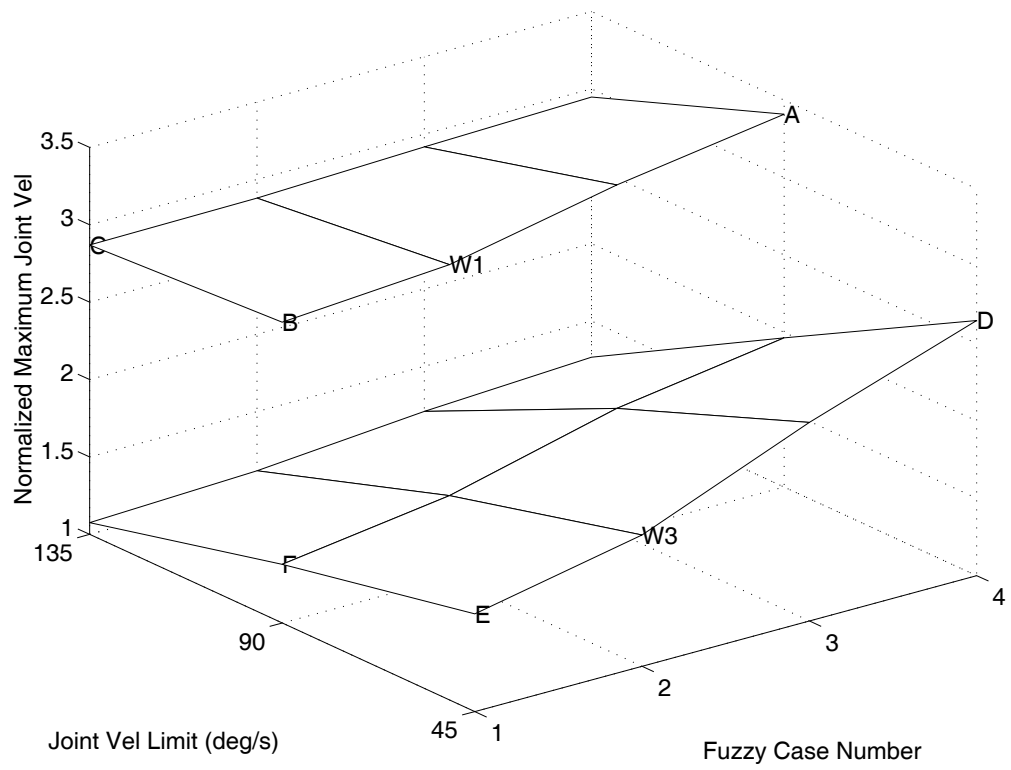


Figure H3. Maximum Joint Velocity Sensitivity for Fuzzy Supervisor in Trajectory Y, RW-DE.

VITA

The author was born in Peoria, IL, on July 7, 1966. He lived in Peoria and graduated from Peoria High School in May 1984. In September 1984, he entered the Massachusetts Institute of Technology where he received a Bachelor of Science in Aeronautics and Astronautics in May 1988. His senior research project was entitled "Development of a Water Jet Propulsion System."

In September 1988, the author entered the University of Michigan at Ann Arbor, MI. He received a Masters of Science in Engineering in December 1989. During this period, he was awarded a National Consortium for Graduate Degrees for Minorities in Engineering (GEM) fellowship. From January 1990 to August 1990, he worked at TRW Space and Defense Sector in Redondo Beach, CA in the spacecraft attitude controls group.

In August 1990, he was granted a research assistantship with the George Washington University/NASA Langley Research Center Joint Institute for the Advancement of Flight Sciences to begin studies leading to the degree of Doctor of Sciences in Aerospace Engineering. His research interests are robotics, control of flexible structures, and system identification.