# 1991 NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

## JOHN F. KENNEDY SPACE CENTER
## UNIVERSITY OF CENTRAL FLORIDA

# MODELING AND CONTROL OF THE AUTOMATED RADIATOR INSPECTION DEVICE


| | |
|---|---|
| PREPARED BY: | Darren Dawson, Ph.D. |
| ACADEMIC RANK: | Assistant Professor |
| UNIVERSITY AND DEPARTMENT: | Clemson University<br>Electrical & Computer Engineering |
| NASA/KSC | |
| DIVISION: | Mechanical Engineering |
| BRANCH: | Special Projects (RADL) |
| NASA COLLEAGUE: | V. Leon Davis, Chief<br>Robotics and Automation |
| DATE: | July 27, 1991 |
| CONTRACT NUMBER: | University of Central Florida<br>NASA-NGT-60002 Supplement: 6 |

## Acknowledgments

I would like to thank NASA and ASEE for the opportunity to participate in this program. My professional knowledge of launch operations and shuttle support has increased greatly based on my interactions with the robotics group at NASA and the employees of Boeing Aerospace Organization, the engineering contractor for the Robotics Applications Development Laboratory. A special thanks to Leon Davis, my NASA colleague, for giving me the freedom to pursue interesting and challenging problems.

## ABSTRACT

Many of the operations performed at Kennedy Space Center (KSC) are dangerous and repetitive tasks which make them ideal candidates for robotic applications. For one specific application, KSC is currently in the process of designing and constructing a robot, called the Automated Radiator Inspection Device (ARID), to inspect the radiator panels on the orbiter. In this report, the following related aspects of the ARID project are discussed: 1) Modeling of the ARID, 2) Design of control algorithms for the ARID, and 3) Nonlinear-Based simulation of ARID control algorithms. Based on the information delineated in this report, recommendations are then made to assist KSC personnel in the successful completion of the ARID project.

## Summary

To automate the inspection of the radiator panels on the orbiter, Kennedy Space Center (KSC) is designing and constructing the Automated Radiator Inspection Device (ARID). Utilizing a vision-based system, the ARID will be capable of inspecting the entire surface of the radiator panels. Due to accuracy requirements imposed by the vision system, the ARID robot must be precisely controlled to ensure that the end-effector mounted camera is maintained at the proper distance from the radiator panels. To aid KSC personnel in the successful completion of the ARID project, a study was done to examine the associated modeling and control aspects of the ARID robot.

With regard to modeling, several separate sub-areas are analyzed and investigated. The rigid body statics and dynamics are formulated to describe the geometry and dynamic motion of the ARID. Since the ARID is a large robot that utilizes harmonic drives, a model including the effects of dominant mode structural vibrations and joint flexibilities are incorporated into the rigid-body dynamics. Since the ARID utilizes a redundant actuation system, a model including redundant actuators is also formulated.

With regard to control, several separate sub-areas are analyzed and investigated. For the rigid-body model, the performance of a PD controller is compared to a robust saturation controller. For the inclusion of joint flexibilities and dominant mode vibrational effects into the rigid body dynamics, the performance of a PD motor controller is compared to a nested PD motor/link controller and a nested PD motor/link/tip controller. To investigate the effects of redundant actuation on the control system performance, an independent torque controller is compared to a coupled torque controller.

# TABLE OF CONTENTS

## LIST OF ILLUSTRATIONS

# I. INTRODUCTION

## 1.1 Robotics at Kennedy Space Center

The mission of Kennedy Space Center (KSC) is to provide manpower and support for fast, efficient, and safe preparation of launch vehicles. Robotics can be a key ingredient to satisfy this mission. Many of the operations performed at KSC are dangerous and repetitive which make them ideal candidates for robotic applications. The design and servicing procedures of present space vehicles and launch procedures make it difficult to implement robotic applications; however, the next generation space vehicles will no doubt be designed with robots in mind. Therefore, KSC personnel will have to become increasingly familiar with robots and related hardware such as sensors and control systems. The Robotics Applications Development Laboratory (RADL) provides this experience to KSC personnel and its contractors.

## 1.2 Description of the Automated Radiator Inspection Device

KSC is currently in the process of designing and constructing a robot, called the Automated Radiator Inspection Device (ARID), to inspect the radiator panels on the orbiter (See Figure 1.1). These panels, located on the inside of the cargo bay doors, are inspected when the orbiter is horizontally parked in the Orbiter Processing Facility (OPF). After the cargo bay doors are opened to expose the radiator panels, the inspection is presently performed by workers in a crane-assisted bucket over the radiator surface. The radiator surface is divided into grids, and surface defects are cataloged by location in the grid. These surface defects are monitored periodically to determine when repair is needed.

To automate this inspection process, the ARID robot is being constructed to move along the orbiter on a long track. Utilizing a vision-based system, the ARID will be capable of inspecting the entire surface of the radiator panels. The associated vision system will be able to divide the radiator panels into smaller grids and thus provide better cataloging of defects.

## 1.3 Objective of this Research Project

The objective of this research project is to assist KSC personnel in three areas: 1) Modeling of the ARID, 2) Design of control algorithms for the ARID, and 3) Nonlinear-based simulation of ARID control algorithms. The body of this report is dividing among these three areas.

With regard to modeling, several separate sub-areas are analyzed and investigated. The rigid body statics and dynamics [1] are formulated to describe the geometry and dynamic motion of the ARID. Specifically, the ARID kinematics, inverse kinematics, manipulator jacobian, and manipulator dynamics are given. Since the ARID utilizes harmonic drives, a model including effects of joint flexibilities [2] is incorporated into the rigid-body dynamics. Because of the large size of the ARID, structural

vibrations [3] are considered to be a potential problem; therefore, a model including the effects of "dominate mode" vibrations are incorporated into the rigid body/flexible joint model. Since the ARID utilizes a redundant actuation system, motor synchronization is considered to be a potential problem; therefore, a model including the effects of redundant actuators is formulated.

With regard to control, several separate sub-areas are analyzed and investigated. A proportional-derivative (PD) controller [1] and a robust saturation controller [4] are both designed for the rigid-body dynamics. For the inclusion of joint flexibilities into the rigid body dynamics, a PD motor controller [1] and nested PD motor/link controller [5] are formulated. For the inclusion of vibrational effects into the rigid body/joint flexibility model, a PD motor controller [1] and nested PD motor/link/tip controller [6] are formulated. To study the effects of redundant actuation on the control system performance, an independent torque controller [7] and a coupled torque controller [7] are formulated.

With regard to simulation, several separate sub-areas are analyzed and investigated. For the rigid-body model, the performance of a PD controller is compared to a robust saturation controller. For the inclusion of joint flexibilities into the rigid body dynamics, the performance of a PD motor controller is compared to a nested PD motor/link controller. For the inclusion of vibrational effects into the rigid body/joint flexibility model, the performance of a PD motor controller is compared to a nested PD motor/link/tip controller. To investigate the effects of redundant actuation on the control system performance, an independent torque controller is compared to a coupled torque controller.

## II. RIGID BODY STATICS AND DYNAMICS

In this section, we present the static and dynamic rigid body relationships for the ARID robot. The terminology rigid body is used to emphasize that flexibilities due to link and drive transmissions have been neglected. In subsequent subsections, we will discuss modifications of the dynamic model due to link and joint flexibilities.

From Figure 1.1, we can see that the prismatic joint moves perpendicular to the direction of the three revolute joints; therefore, the motion of the prismatic joint is decoupled from the motion of the revolute joints. Since the prismatic motion is decoupled from the motion of the revolute joints, the static and dynamic relationships for the prismatic joint are relatively simple; therefore, in this section, we will only discuss the static and dynamic relationships for the revolute joints. Specifically, we will present the kinematics, inverse kinematics, manipulator jacobian, and the manipulator dynamic relationships for the three revolute joints.

### 2.1 Kinematics

The kinematics [1] (i.e. forward kinematics) problem is associated with finding the end-effector position and orientation given the three joint

angles. The end-effector position and orientation is sometimes called the task space coordinate set. From the geometric description given in Figure 2.1, we define the task space coordinate set as

$$X = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{2.1}$$

where X is a 3x1 vector, x denotes the position of the end-effector in the x-direction, y denotes the position of the end-effector in y-direction, and $\theta$ denotes the end-effector angle of orientation measured off the x-axis. Note that the coordinate set {x,y} defined in Figure 2.1 is a non-rotating coordinate set.

Utilizing a geometric approach, it is easy to show that the relationships between the task space variables and the joint space variables are given by

$$X = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = h(q) = \begin{bmatrix} L_3c_{123} + L_2c_{12} + L_1c_1 \\ L_3s_{123} + L_2s_{12} + L_1s_1 \\ q_1 + q_2 + q_3 \end{bmatrix} \tag{2.2}$$

where $q_1$, $q_2$, $q_3$ are the joint variables defined in Figure 2.1; h(q) is the 3x1 vector of kinematic functions; $L_1$, $L_2$, $L_3$ are the link lengths defined in Figure 2.1; and $c_1$, $s_{12}$, $c_{123}$ are used as abbreviations for $\cos(q_1)$, $\sin(q_1 + q_2)$, and $\cos(q_1 + q_2 + q_3)$, respectively.

## 2.2 Inverse Kinematics

The inverse kinematics [1] problem is associated with finding the three joint angles given the end-effector position and orientation. From the geometric description given in Figure 2.1, we define the joint space variable set as

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{2.3}$$

where q is a 3x1 vector composed of the joint variables.

Utilizing a geometric approach, it is easy to show that the relationship between the joint space variables and the task space variables are given by

$$q_1 = \text{Atan2}\left[\frac{y - L_3s_\theta}{x - L_3c_\theta}\right] \pm \text{Acos}\left[\frac{(x - L_3c_\theta)^2 + (y - L_3s_\theta)^2 + L_1^2 - L_2^2}{2L_1\sqrt{(x - L_3c_\theta)^2 + (y - L_3s_\theta)^2}}\right], \tag{2.4}$$

$$q_2 = \text{Atan2}\left[\frac{c_1 y - s_1 x - L_3(c_1 s_\theta - s_1 c_\theta)}{c_1 x + s_1 y - L_3(c_1 s_\theta + s_1 s_\theta) - L_1}\right], \tag{2.5}$$

and

$$q_3 = \theta - q_1 - q_2 \tag{2.6}$$

where Acos(.) is used to denote the inverse cosine operation, Atan2(.) is used to denote the inverse tangent operation with quadrant checking operation, and the positive/negative operation (i.e. $\pm$ in (2.4)) is used to specify the up/down configuration of the ARID elbow.

### 2.3 Manipulator Jacobian

The manipulator jacobian matrix [1] is used to relate the joint space velocity vector to the task space velocity vector. The jacobian matrix is obtained by taking the time derivative of the forward kinematics given in (2.2). That is, the 3x3 Jacobian matrix (i.e. J(q)) satisfies the relationship

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = J(q)\dot{q} = J(q)\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \tag{2.7}$$

where

$$J(q) = \begin{bmatrix} -L_1 s_1 - L_2 s_{12} - L_3 s_{123} & -L_2 s_{12} - L_3 s_{123} & -L_3 s_{123} \\ L_1 c_1 + L_2 c_{12} + L_3 c_{123} & L_2 c_{12} + L_3 c_{123} & L_3 c_{123} \\ 1 & 1 & 1 \end{bmatrix}. \tag{2.8}$$

One of the advantages of calculating the jacobian matrix online is that it can be used to avoid online calculation of the inverse kinematics. For example, one can utilize the task-space PD control [1]

$$\tau = J^T(q)\left[K_v(\dot{x}_d - J(q)\dot{q}) + K_p(x_d - h(q))\right] \tag{2.9}$$

where $K_v$, $K_p$ are positive definite 3x3 gain matrices, $x_d$ is a 3x1 vector used to represent the desired task space trajectory that we wish to track, and $\tau$ is the 3x1 vector used to represent the input torque to each link.

Note that (2.9) does not depend on the calculation of the inverse kinematics. It should also be noted that the task space coordinate set defined in this report is a nonrotating coordinate frame. For actual implementation of the ARID robot, a task space formulation can be used that

ensures that the ARID camera is always perpendicular to the radiator surface. This task-space coordinate set can be formulated if an analytical function of the radiator surface can be obtained.

## 2.4 Manipulator Dynamics

A robot is basically a positioning device. To control the end effector position we must know the dynamic properties of the manipulator in order to know how much force to exert on it to cause it to move. Too little force and the manipulator is slow to react. Too much force and the manipulator may oscillate about the desired position.

In this section, we give the dynamics of the ARID robot in the Lagrange Euler form [1]. The rigid body dynamics for the 3-link revolute ARID robot arm are

$$\tau = M(q)\ddot{q} + V(q,\dot{q}) + G(q) + F(\dot{q}) \tag{2.10}$$

where $M(q)$ is a 3×3 link inertia matrix, $V(q,\dot{q})$ is a 3×1 vector containing the centripetal and Coriolis terms, $G(q)$ is a 3×1 vector containing the gravity terms, $F(\dot{q})$ is a 3×1 vector containing the static and dynamic friction terms, $\ddot{q}$ is a 3×1 vector representing the link accelerations, and $\tau$ is the 3×1 control vector used to represent the torque provided by the actuators in each joint. With regard to ARID robot, we now explicitly define each term associated with (2.10). First, the inertia matrix $M(q)$ is given by

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \tag{2.11}$$

where

$$M_{11} = m_1 L_{c1}^2 + m_2(L_1^2 + L_{c2}^2 + 2L_1 L_{c2} c_2) + I_1 + I_2 + I_3 + m_3(L_1^2 + L_2^2 + L_{c3}^2 + 2L_1 L_2 c_2 + 2L_1 L_{c3} c_{23} + 2L_2 L_{c3} c_3),$$

$$M_{12} = M_{21} = m_2(L_{c2}^2 + L_1 L_{c2} c_2) + I_2 + I_3 + m_3(L_2^2 + L_{c3}^2 + L_1 L_2 c_2 + L_1 L_{c3} c_{23} + 2L_2 L_{c3} c_3),$$

$$M_{13} = M_{31} = I_3 + m_3(L_{c3}^2 + L_1 L_{c3} c_{23} + L_2 L_{c3} c_3),$$

$$M_{22} = m_2 L_{c2}^2 + I_2 + I_3 + m_3(L_2^2 + L_{c3}^2 + 2L_2 L_{c3} c_3),$$

$$M_{23} = M_{32} = I_3 + m_3(L_{c3}^2 + L_2 L_{c3} c_3),$$

$$M_{33} = m_3 L_{c3}^2 + I_3,$$

$I_1$, $I_2$, $I_3$, are the moments of inertia of links $L_1$, $L_2$, $L_3$, respectively; $L_{c1}$, $L_{c2}$, $L_{c3}$, are the distances to the center of mass of links $L_1$, $L_2$, $L_3$, respectively; and $m_1$, $m_2$, $m_3$, are the masses of links $L_1$, $L_2$, $L_3$, respectively (See Figure 2.1). Second, the Coriolis/centripetal terms $V(q,\dot{q})$ are given by

$$V(q,\dot{q}) = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \tag{2.12}$$

where

$$\begin{aligned} V_1 = {} & (-m_2 L_1 L_{c2} - m_3(L_1 L_2 s_2 + L_1 L_{c3} s_{23}))\dot{q}_2^2 + (-m_3(L_1 L_{c3} s_{23} + L_2 L_{c3} s_3))\dot{q}_3^2 \\ & + (-m_2 L_1 L_{c2} - m_3(L_1 L_2 s_2 + L_1 L_{c3} s_{23}))\dot{q}_1\dot{q}_2 \\ & + 2(-m_3(L_1 L_{c3} s_{23} + L_2 L_{c3} s_3))(\dot{q}_1\dot{q}_2 + \dot{q}_2\dot{q}_3), \end{aligned}$$

$$\begin{aligned} V_2 = {} & (m_2 L_1 L_{c2} + m_3(L_1 L_2 s_2 + L_1 L_{c3} s_{23}))\dot{q}_1^2 + (-m_3 L_2 L_{c3} s_3)\dot{q}_3^2 \\ & + 2(-m_3 L_2 L_{c3} s_3)(\dot{q}_1\dot{q}_2 + \dot{q}_2\dot{q}_3), \end{aligned}$$

and

$$V_3 = (m_3(L_1 L_{c3} s_{23} + L_2 L_{c3} s_3))\dot{q}_1^2 + (m_3 L_2 L_{c3} s_3)\dot{q}_2^2 + 2(m_3 L_2 L_{c3} s_3)\dot{q}_1\dot{q}_2.$$

Third, the gravity terms $G(q)$ are given by

$$G(q) = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} \tag{2.13}$$

where

$$G_1 = m_1 g L_{c1} c_1 + m_2 g(L_1 c_1 + L_{c2} c_{12}) + m_3 g(L_1 c_1 + L_2 c_{12} + L_{c3} c_{123}),$$

$$G_2 = m_2 g L_{c2} c_{12} + m_3 g (L_2 c_{12} + L_{c3} c_{123}),$$

$$G_3 = m_3 g L_{c3} c_{123},$$

and

g is used to denote the gravitational constant (i.e. 9.81 $m/s^2$). Lastly, the friction terms $F(\dot{q})$ are given by

$$F(\dot{q}) = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \tag{2.14}$$

where

$$F_1 = v_{f1}\dot{q}_1 + k_{f1}\mathrm{sgn}(\dot{q}_1),$$

$$F_2 = v_{f2}\dot{q}_2 + k_{f2}\mathrm{sgn}(\dot{q}_2),$$

$$F_3 = v_{f3}\dot{q}_3 + k_{f3}\mathrm{sgn}(\dot{q}_3),$$

$v_{f1}$, $k_{f1}$ are positive scalar constants used to denote the static and dynamic coefficients of friction for joint 1; $v_{f2}$, $k_{f2}$ are positive scalar constants used to denote the static and dynamic coefficients of friction for joint 2; $v_{f3}$, $k_{f3}$ are positive scalar constants used to denote the static and dynamic coefficients of friction for joint 3; and the sgn(.) is used to represent the signum function.

## III. SIMULATION OF CONTROLLERS FOR THE RIGID BODY DYNAMICS

Utilizing the rigid body dynamics developed in Section II, controllers can be developed and tested. Specifically, the dynamic in (2.10) can be rewritten in the form

$$\ddot{q} = M^{-1}(q)[\tau - V(q,\dot{q}) - G(q) - F(\dot{q})]. \tag{3.1}$$

Utilizing a integration package such as SIMNON, the dynamics given by (3.1) can now be integrated over any desired simulation interval for any postulated control input $\tau$.

Typically, the robot control objective is formulated as the tracking problem. That is, we wish to follow a desired trajectory for each joint with as small as deviation possible. To quantify the measure of success, we often

examine the tracking error which is defined by

$$e = q_d - q \tag{3.2}$$

where the 3x1 vector $q_d$ is used to represent the desired trajectory for each joint. From (3.2), we can see that if tracking error (i.e. e) is small then we can be reasonably assured that the robot is performing as desired.

### 3.1 PD Control

As a starting point, we first simulated the simple PD controller [14] given by

$$\tau = K_v\dot{e} + K_p e \tag{3.3}$$

where $K_v$, $K_p$ are 3x3 diagonal, positive definite matrices. The details of the program are given in the SIMNON file ARID1.T [8]. From Figure 3.1, we can see that the PD controller performs poorly, that is, the position and velocity tracking errors are relatively large.

### 3.2 Robust Saturation Control

To illustrate how an advanced control algorithm can improve tracking performance, we proposed the robust saturation controller [4]

$$\tau = K_v\dot{e} + K_p e + \frac{(e + \dot{e})\rho^2}{\|e + \dot{e}\|\rho + \epsilon} \tag{3.4}$$

where $\epsilon$ is a scalar positive constant, $\|\cdot\|$ is the standard Euclidean norm, and $\rho$ is a scalar function that bounds the uncertainty. The details of the program are given in the SIMNON file ARID2.T [8]. From Figure 3.1 and 3.2, we can see that the robust saturation controller outperforms the PD controller by a wide margin. The reason for this improvement in performance is that the robust saturation controller compensates for any "uncertainty" with regard to the manipulator dynamics.

## IV. MODELING AND CONTROL FOR JOINT FLEXIBILITIES

Since the ARID robot is a rather massive robot, the torque delivered to each link must be relatively large. To provide the necessary torque amplification, the ARID robot's actuation system utilizes harmonic drives. While the harmonic drives amplify the torque by a factor of approximately 200, the harmonic drives introduce flexibility at each joint. In order to move the robot accurately, the control strategy must compensate for the "joint flexibility" introduced by the harmonic drive. This very fact was confirmed in a presentation to KSC personnel by ROBOTICS RESEARCH INC.

### 4.1 Model of Joint Flexibilities

The joint flexibilities introduced by the harmonic drives can be modeled by placing springs at each joint (See Figure 4.1). Similar to [2], the rigid body dynamics for the 3-link revolute ARID robot arm with joint flexibilities can then be shown to be

$$M(q)\ddot{q} + V(q,\dot{q}) + G(q) + F(\dot{q}) = K(\Gamma q_m - q) \tag{4.1}$$

and

$$J\ddot{q}_m + B\dot{q}_m + \Gamma K(\Gamma q_m - q) = \tau \tag{4.2}$$

where $q_m(t)$ is a 3×1 vector representing the actuator displacements, K is a constant diagonal 3×3 matrix used to denote the joint flexibilities, J is a 3×3 positive-definite constant diagonal actuator inertia matrix, B is a positive-definite constant diagonal 3×3 matrix used to represent the actuator damping, $\Gamma$ is a 3×3 positive-definite constant diagonal matrix used to represent the gear ratio in each harmonic drive, and all other quantities are the same as those defined in (2.10).

### 4.2 PD Motor Control and Simulation

To illustrate the pitfalls of local feedback (i.e. motor encoder feedback only), we simulated a simple PD motor controller [1] given by

$$\tau = K_v(\dot{q}_d - \dot{q}_m) + K_p(q_d - q_m) \tag{4.3}$$

where $K_v$, $K_p$ are 3x3 diagonal, positive definite matrices. The details of the program are given in the SIMNON file ARID3.T [8]. From Figure 4.2, we can see that the PD controller performs poorly. The information delineated by Figure 4.2 makes sense since if we use motor feedback only, we not even attempting to control the robot end-effector (i.e. link position). That is, we are only controlling the motor.

### 4.3 Nested PD Motor/Link Control and Simulation

To illustrate the improvement in performance of including link encoder feedback along with motor encoder feedback, we simulated a nested PD motor/link controller [5] (i.e. nested feedback control means one feedback loop is inside another feedback loop) given by

$$\tau = K_{mv}K_{mp}[K_v\dot{e} + K_p e - K(q_m - q)] - K_{mv}K(\dot{q}_m - \dot{q}) \tag{4.4}$$

where $K_v$, $K_p$, $K_{mv}$, $K_{mp}$ are 3x3 diagonal, positive definite matrices, and K is the joint flexibility matrix defined in (4.1). The details of the program

are given in the SIMNON file ARID5.T [8]. From Figure 4.3, we can see that the nested PD motor/link controller performs very well. The information delineated by Figure 4.3 also makes sense since if we use link feedback, we are actually measuring the quantity that we attempting to control. Note that in addition to using the motor resolver for motor position information, we must mount a position encoder on each link to implement the controller given in (4.4).

## V. MODELING AND CONTROL FOR VIBRATIONAL EFFECTS

KSC personnel have often stated that the ARID will inspect the radiator panel along the prismatic joint axis [9]. That is, the gross motion of the ARID will be along the track while the revolute joints will move in small increments. Since the ARID is a long slender robot, it has been postulated that link vibrations along the prismatic joint axis may be a potential problem; therefore, the ARID control algorithm may have to be designed to actively compensate for these vibrations. Since these vibrations are the only ones considered in this report, we will refer to these gross motion vibrations as "dominant" vibrations.

### 5.1 Model of Dominant Vibrations

To compensate for the vibrations along the prismatic joint axis, a model for these vibrations must first be formulated. As shown in Figure 5.1, we have selected a lumped model [3] as a first attempt at proposing a possible solution to this problem. It should be emphasized that this problem of link flexibilities or link vibrations is still considered a research area.

From Figure 5.1, the dominant vibration model including joint flexibility is given by

$$m_e\ddot{q}_e + b_v(\dot{q}_e - \dot{q}_L) + k_v(q_e - q_L) = 0, \tag{5.1}$$

$$m_L\ddot{q}_L + b_v(\dot{q}_L - \dot{q}_e) + k_v(q_L - q_e) = k_m(\gamma q_m - q_L), \tag{5.2}$$

and

$$m_m\ddot{q}_m + b_m\dot{q}_m + \gamma k_m(\gamma q_m - q_L) = \tau \tag{5.3}$$

where $m_e$, $m_L$, $m_m$ are positive scalar constants used to represent the lumped mass of the end-effector, link, and motor, respectively; $b_v$, $b_m$ are positive scalar constants used to represent the lumped damping for the vibrational effects and the motor, respectively; $q_e$, $q_L$, $q_m$ are used to represent the position of the end-effector, link, and motor, respectively; $k_v$, $k_m$ are positive scalar constants used to represent the lumped spring constant for the vibrational effects and the joint flexibilities, respectively; $\gamma$ is used

to represent the gear ratio, and $\tau$ is used to represent the control input.

### 5.2 PD Motor Control and Simulation

To illustrate the pitfalls of local feedback (i.e. motor encoder feedback only), we simulated a simple PD motor controller [1] given by

$$\tau = K_v(\dot{q}_d - \dot{q}_m) + K_p(q_d - q_m) \tag{5.4}$$

where $K_v$, $K_p$ are 3x3 diagonal, positive definite matrices. The details of the program are given in the SIMNON file FLEX4.T [8]. From Figure 5.2, we can see that the PD controller performs very poorly. The information delineated by Figure 5.2 makes sense since if we use motor feedback only, we not even attempting to control the robot end-effector (i.e. end-effector position). That is, we are only controlling the motor.

### 5.3 Nested PD Motor/Link/Tip Control and Simulation

To illustrate the improvement in performance of including link encoder feedback and end-effector measurements along with motor encoder feedback, we simulated a nested PD motor/link/tip controller [6] given by

$$\tau = K_{p3}K_{p2}K_{v1}[K_p\dot{e} - (\dot{q}_L - \dot{q}_e)] + K_{p3}K_{p2}K_{p1}[K_p e - (q_L - q_e)] \tag{5.5}$$

$$- K_{p3}K_{p2}(q_m - q_L) - K_{p3}(\dot{q}_m - \dot{q}_L)$$

where $K_{p3}$, $K_{p2}$, $K_{v1}$, $K_p$, $K_{p1}$ are positive controller gains. The details of the program are given in the SIMNON file FLEX3.T [8]. From Figure 5.3, we can see that the nested PD motor/link/tip controller performs very well. The information delineated by Figure 5.3 also makes sense since if we use end-effector feedback, we are actually measuring the quantity that we are attempting to control.

As we have already stated the controller given in (5.5) requires measurement of end-effector position as well as link and motor position; therefore, if vibrations become a problem, we must have a way of measuring end-effector position. A common method for measuring end-effector position for a vibrating robot is to mount an accelerometer on the end of the robot. The signal from the accelerometer can be integrated once to obtain end-effector velocity, that is

$$\dot{q}_e(t) = \int_0^t \ddot{q}_e(t)\, dt + \dot{q}_e(0). \tag{5.6}$$

The end-effector position can then be obtained by integrating (5.6) to obtain

$$q_e(t) = \int_0^t \dot{q}_e(t)\, dt + q_e(0). \tag{5.7}$$

## VI. SYNCHRONIZATION OF REDUNDANT MOTORS

Since the ARID robot is the first robot to be installed next to flight hardware, the reliability of the ARID robot and associated control system is extremely important. That is, the ARID robot should be designed to minimize the chance of damaging the orbiter. To enhance reliability of the ARID robot, the electromechanical design of each joint includes redundant drive shafts, bearings, harmonic drives, brakes, transmission chains, motors, and computer control systems.

As pointed out in [9], two motors in parallel could cause a potential problem in that the motors might "fight" each other. That is, since the two motor/control systems will not have exactly the same dynamic characteristics, the torque delivered by each motor to the corresponding link will not be exactly the same at any instant of time. In this section, we develop a model and some control strategies to examine this motor synchronization problem.

### 6.1 Model of Motor Synchronization Problem

The model used to study the motor synchronization problem was based on the same model that was used to study the joint flexibility effects. For simplicity, we will only consider a one-link problem as illustrated in Figure 6.1. From Figure 6.1, we can see that the model is composed of two motors connected to the same link. The associated dynamic equations for this system are given by

$$mL^2\ddot{q} + mgL\sin(q) + v_f\dot{q} + k_f\,\mathrm{sgn}(\dot{q}) = K_1(\Gamma_1 q_{m1} - q) + K_2(\Gamma_2 q_{m2} - q), \tag{6.1}$$

$$J_1\ddot{q}_{m1} + B_1\dot{q}_{m1} + \Gamma_1 K_1(\Gamma_1 q_{m1} - q) = \tau_1, \tag{6.2}$$

and

$$J_2\ddot{q}_{m2} + B_2\dot{q}_{m2} + \Gamma_2 K_2(\Gamma_2 q_{m2} - q) = \tau_2 \tag{6.3}$$

where $q_{m1}(t)$, $q_{m2}(t)$ represent the actuator displacements of motors 1 and 2, respectively; $\tau_1$, $\tau_2$ represent the torque delivered by motors 1 and 2, respectively; $K_1$, $K_2$ are positive constants used to denote the joint flexibilities of harmonic drives 1 and 2, respectively; $J_1$, $J_2$ are positive constants used to represent the inertia of motors 1 and 2, respectively; $B_1$, $B_2$ are positive constants used to represent the damping in motors 1 and 2, respectively; $\Gamma_1$, $\Gamma_2$ are positive constants used to represent the gear ratio

of harmonic drives 1 and 2, respectively; and all other quantities are the same as those defined in (4.1) and (4.2).

### 6.2 Independent Torque Control and Simulation

To simulate the effects of redundant motors, we first simulated the nested PD motor/link controller given in Section 4.3 for each redundant system. That is, each control system, designated control system 1 and 2, has no sensory input based on its counterpart's performance. From Section 4.3, the two "independent" torque controllers are given by

$$\tau_1 = K_{mv1}K_{mp1}[K_{v1}\dot{e} + K_{p1}e - K_1(q_{m1} - q)] - K_{mv1}K_1(\dot{q}_{m1} - \dot{q}) \qquad (6.4)$$

and

$$\tau_2 = K_{mv2}K_{mp2}[K_{v2}\dot{e} + K_{p2}e - K_2(q_{m2} - q)] - K_{mv2}K_2(\dot{q}_{m2} - \dot{q}) \qquad (6.5)$$

where $K_{v1}$, $K_{p1}$, $K_{mv1}$, $K_{mp1}$, $K_{v2}$, $K_{p2}$, $K_{mv2}$, $K_{mp2}$ are positive controller gains; $K_1$, $K_2$ are the joint flexibility constants defined in (6.1); and the link tracking error is defined in (3.2).

To simulate a possible failure of control system 2, $\tau_2$ was set to zero at 4 seconds. To simulate mismatch in the redundant systems, the parameters representing the two motors/drives were assumed to be mismatched by 50%. The details of the program are given in the SIMNON file SYC1.T [8]. From Figure 6.2, we can see that the independent nested PD motor/link controllers perform very well. Even after motor 2 is shut completed down at 4 seconds, controller 1 compensates for this failure. Figure 6.2 seems also to confirm that the motor fighting problem may not be a problem if a proper torque based controller is utilized.

### 6.3 Coupled Torque Control and Simulation

To simulate the effects of coupled control of redundant motors, we simulated a "coupled" nested PD motor/link controller similar to that given in Section 4.3. That is, each control system has sensory input based on its counterpart's performance. The two coupled torque controllers are given by

$$\tau_1 = \tau_2 = K_{mv}K_{mp}[K_v\dot{e} + K_pe - K(q_{m1} + q_{m2} - 2q)] \qquad (6.6)$$

$$- K_{mv}K(\dot{q}_{m1} + \dot{q}_{m2} - 2\dot{q})$$

where $K_v$, $K_p$, $K_{mv}$, $K_{mp}$ are positive controller gains; and K is the average joint flexibility constant (i.e. $(K_1 + K_2)/2$.

To simulate a possible failure of control system 2, $\tau_2$ was set to zero at 4 seconds. To simulate mismatch in the redundant systems, the parameters representing the two motors/drives were assumed to be mismatched by 50%. The details of the program are given in the SIMNON file SYC2.T [8]. From Figure 6.3, we can see that the coupled nested PD motor/link controllers perform very well. Even after motor 2 is shut completed down at 4 seconds, controller 1 compensates for this failure. Figure 6.3 seems also to confirm that the motor fighting problem may not be a problem if a proper torque based controller is utilized.

It should be noted that the coupled torque control approach can be designed to maintain the redundant concept by installing redundant motor encoders at each motor shaft. In this way, the coupled torque control approach can be used without requiring that control systems 1 and 2 have any electrical connections that violate the integrity of the redundant channels.

## VII. CONCLUSIONS AND RECOMMENDATIONS

In this report, the following related aspects of the ARID project have been discussed 1) Modeling of the ARID, 2) Design of control algorithms for the ARID, and 3) Nonlinear-based simulation of ARID control algorithms. Based on the information delineated in this report, conclusions and recommendations are now made to assist KSC personnel in the successful completion of the ARID project

### 7.1 Conclusions

To aid KSC personnel in the successful completion of the ARID project, a study was done to examine the associated modeling and control aspects of the ARID robot. With regard to modeling, the rigid body statics and dynamics were formulated to describe the geometry and dynamic motion of the ARID. A model including the effects of joint flexibilities and dominant vibrations were then incorporated into the rigid-body dynamics. Since the ARID utilizes a redundant actuation system, a model including redundant actuation was formulated.

With regard to control, a robust saturation controller was shown to outperform a PD controller for the rigid body dynamics. For the inclusion of joint flexibilities and dominant vibrational effects into the rigid body dynamics, a nested PD motor/link controller and a nested PD motor/link/tip controller were both shown to outperform a PD motor controller. To investigate the effects of redundant actuation on the control system performance, an independent nested PD motor/link torque controller and a coupled nested PD motor/link torque controller were both shown to give good performance inspite of a single motor failure.

### 7.2 Recommendations

Since it is widely recognized by robotic engineers that the robot control problem must be solved at the torque input level, it is suggested that the motor drive units be changed to accommodate torque based control development.

It is important to note that the synchronization of the redundant motors appears not to be a potential problem if a proper torque based control is used.

To compensate for joint flexibilities induced by the harmonic drives, it is suggested that redundant "base-mounted" link encoders be added to the design. Simply stated, if you do not measure link position, you can not control link position.

If link flexibilities (i.e. vibrations) are noted to be a problem during the initial testing phase, it is suggested that redundant link tip measurement devices (such as accelerometers) be added to the design.

Determine the failure mode of each motor to see if it continues to turn if it stops running due to a failure.

Since the host computer for the ARID control system is a relatively slow computer, it is suggested that a high speed DSP expansion board be added to the design to speed up online transmission and collection of data.

To reduce the time spent writing code and doing hardware interfacing, it is suggesting that a high level software system (for example LABVIEW) be used. Moving the robot accurately is a hard enough task as it is; therefore, re-design of low-level software and hardware should be avoided. It should also be noted that National Instruments sells a high-speed DSP expansion board (i.e. NB-DSP2300, 32-bit, 33 MFLOPS, high speed data transfer) for LABVIEW applications.

Utilize Jacobian matrix in conjunction with task-space formulation to avoid solving the inverse-kinematics online.

Since large movements of the prismatic joint may cause excessive vibration, KSC personnel should plan to make large movements with the revolute joints as a backup mode of operation.

Since exact position of the radiator is not known, it suggested that the controller incorporate feedback (i.e. range sensors) to accurately place the ARID end-effector. If the radiator panel does warp, it is believed that no "warping model" will model these nonlinear effects accurately enough to move the camera at 24" $\pm$ 1/8" tolerance in an open loop configuration.

Since it would be desirable for computer system 1 to know the position and velocity of motor 2, it is suggested that redundant motor encoders be added to the the design (This concept maintains the validity of the redundant specification). Note that redundant motor encoders will be needed if the ARID robot requires a coupled control approach.

If the ARID can not be accurately moved with simple nested PD control laws as the ones given in this report, KSC personnel should use advanced control techniques such as the ones given in [5] and [6].

To make the programs written in this report more useful, the following steps should be taken: 1) the prismatic program should be combined with the revolute program, 2) the redundant motor model should then be included, and 3) actual task space desired trajectories related to the radiator surface should then be included. If all these steps are taken, one program can then be used to evaluate the overall system performance.

If the PC23 indexer is used and a torque based controller is not used then vibration can be reduced by using the S-curve velocity profiling algorithm that comes with the indexer.

# VIII. FIGURES

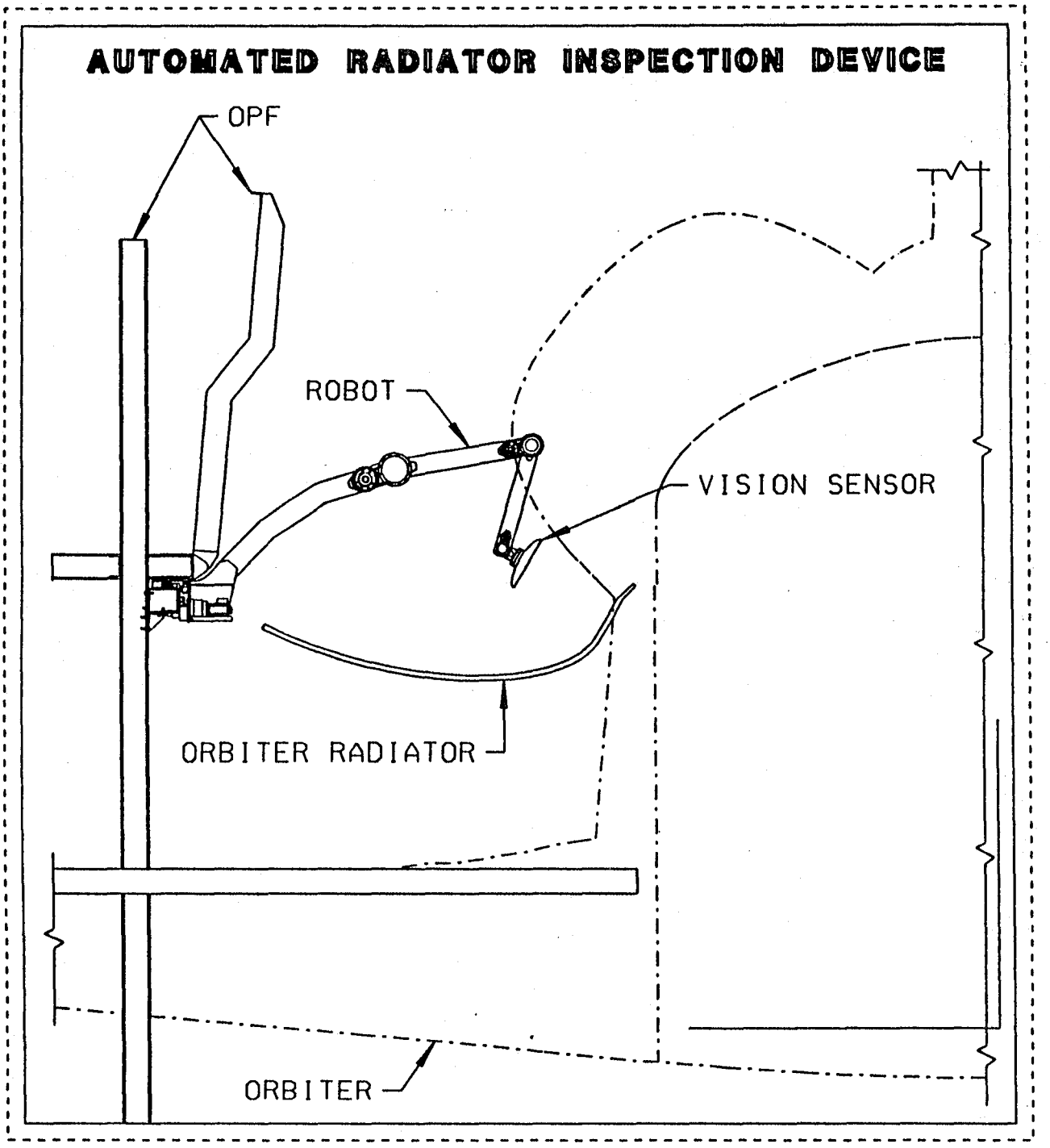


Figure 1.1 Diagram of the ARID Robot

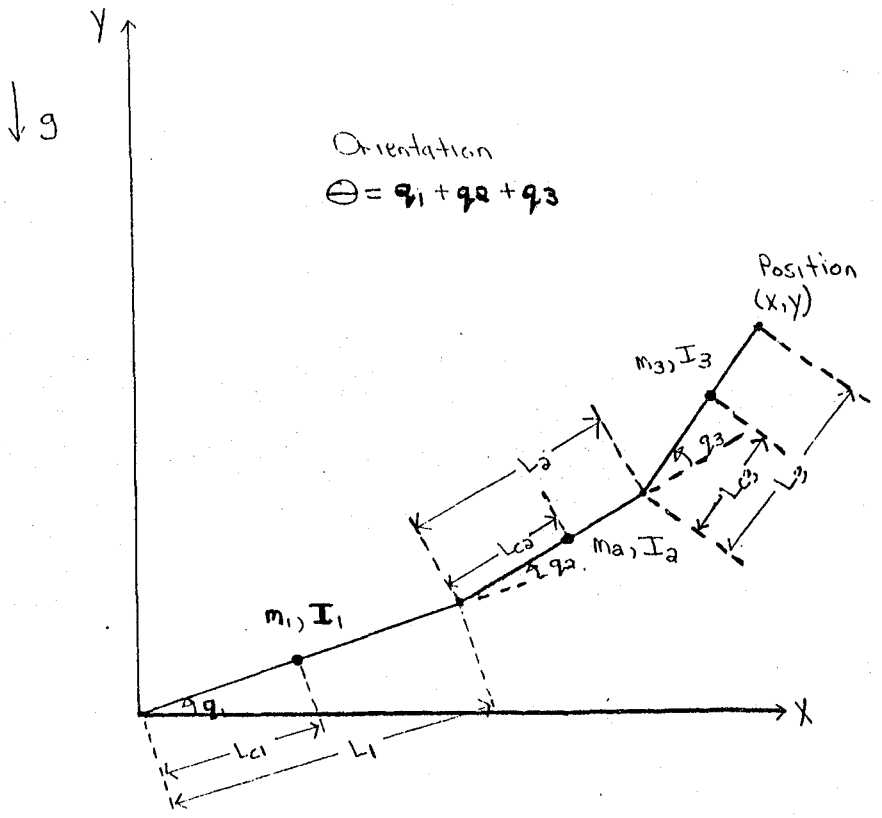


Figure 2.1 3-Link Robot Freebody Diagram

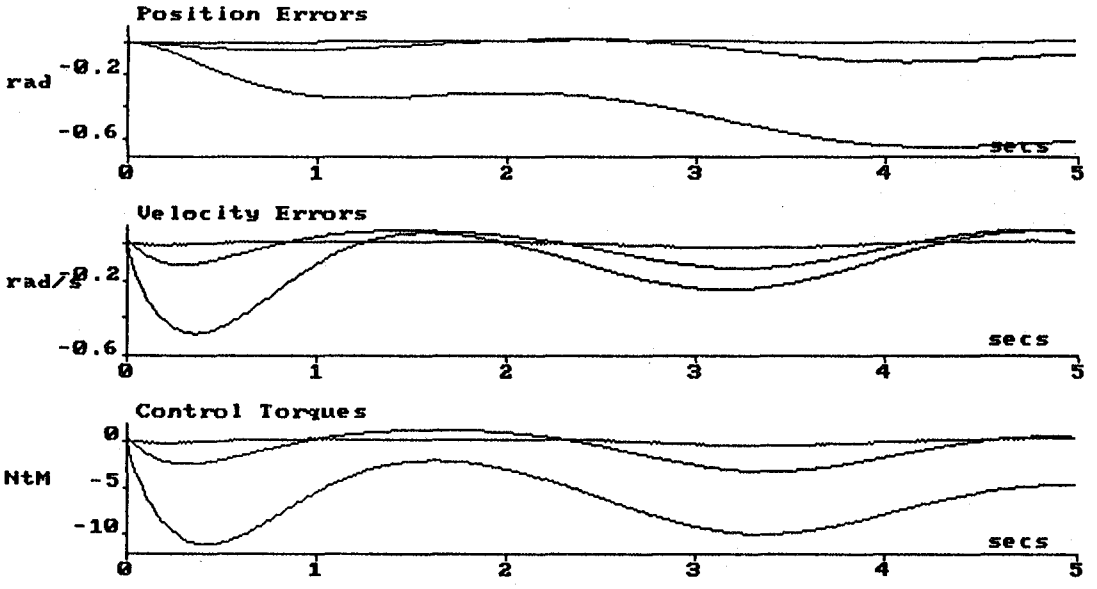


Figure 3.1 Rigid-Link PD Control Simulation

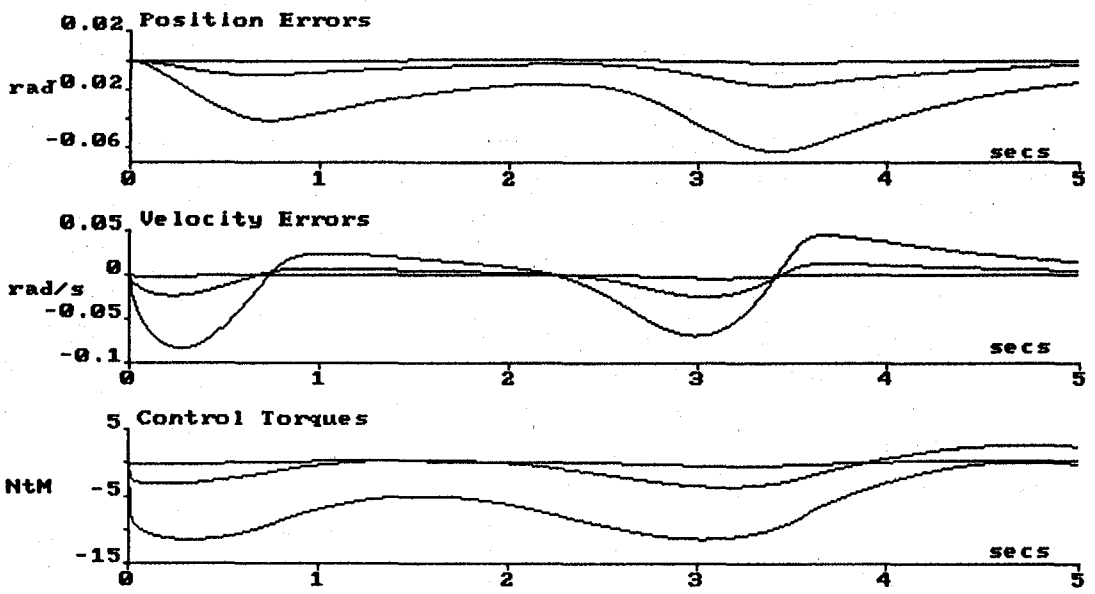


Figure 3.2 Rigid-Link Robust Saturation Control Simulation

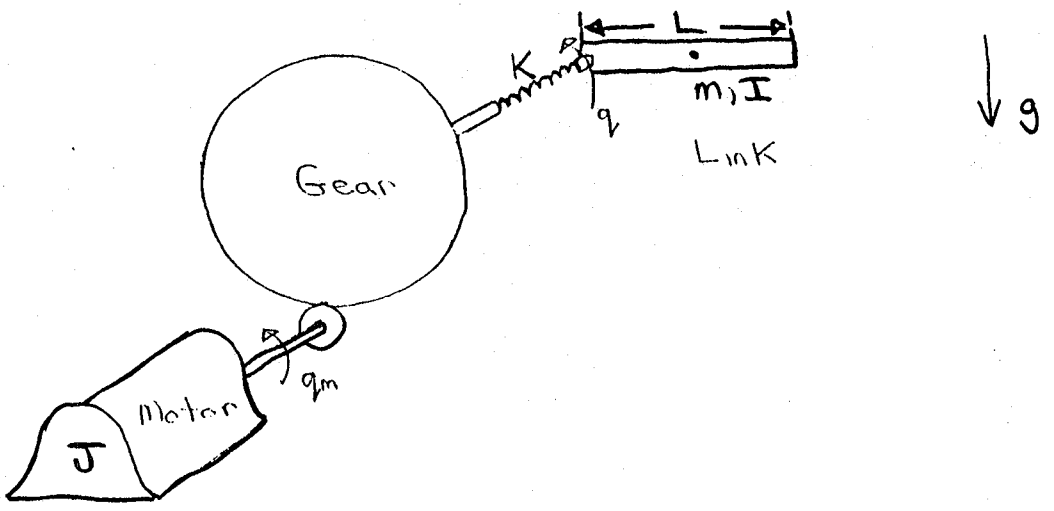


Figure 4.1 Model of Joint Flexibilities

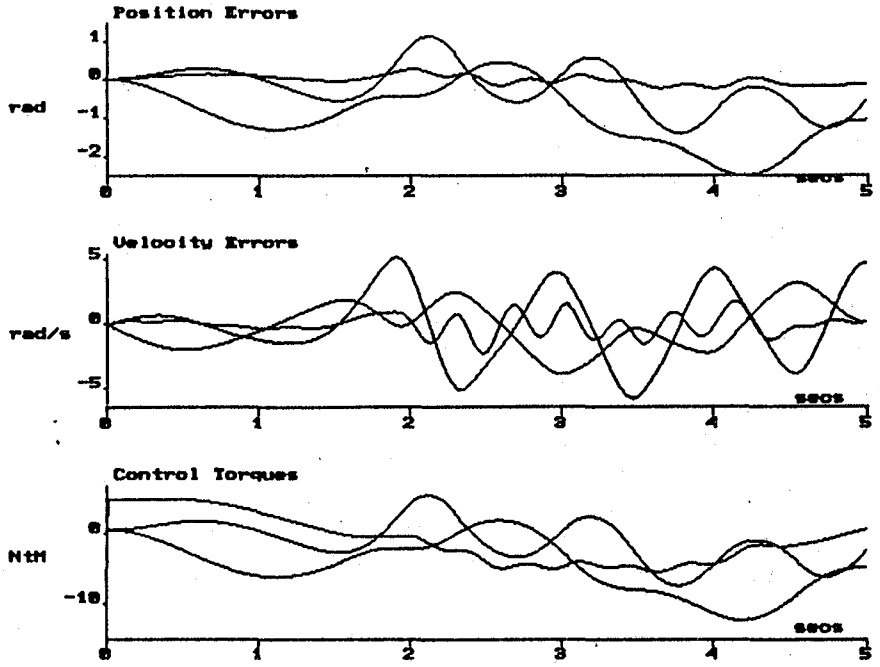


Figure 4.2 PD Motor Control Simulation

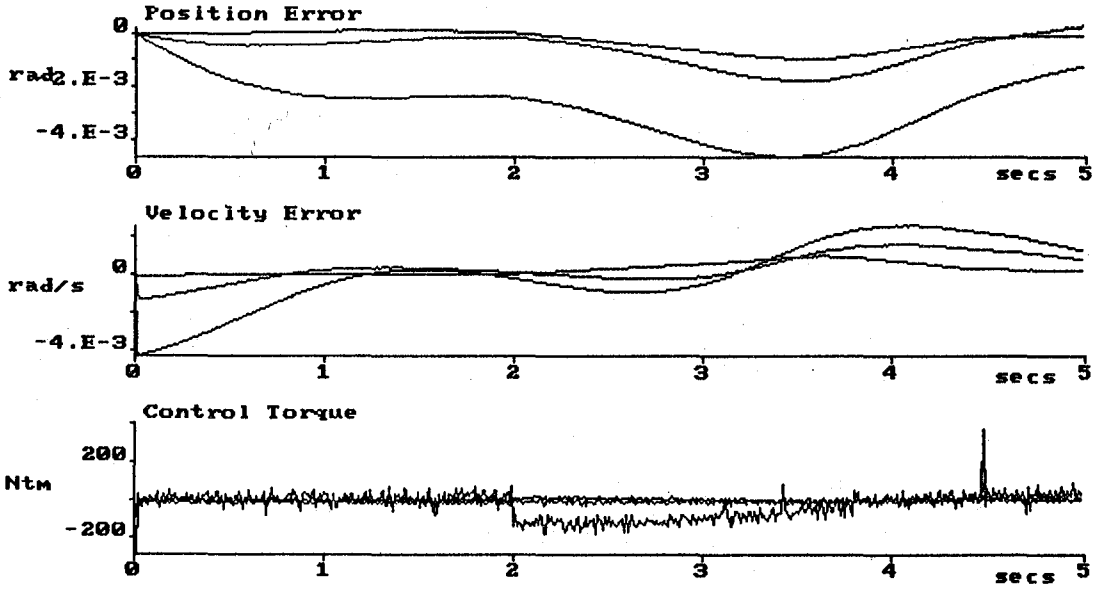


Figure 4.3 Nested PD Motor/Link Control Simulation

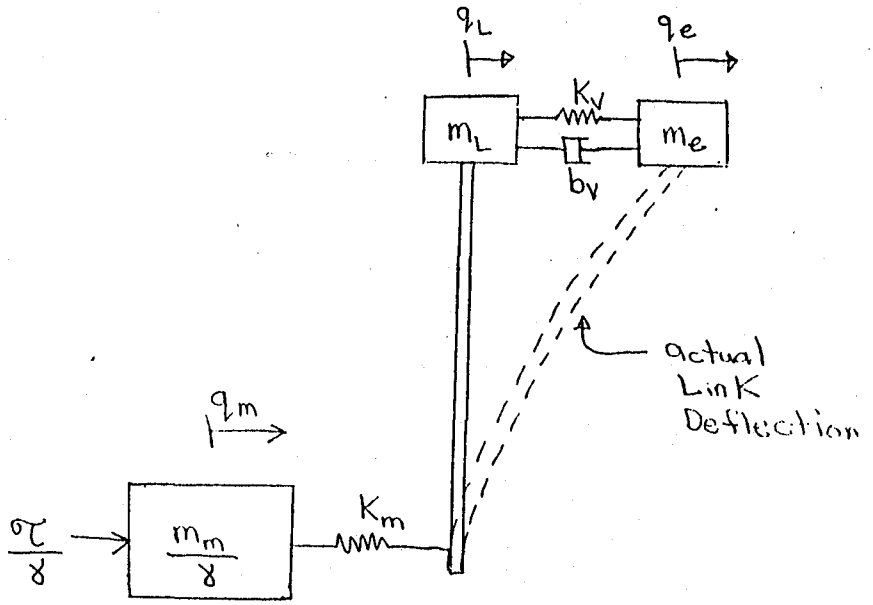


Figure 5.1 Model of Dominant Vibrations

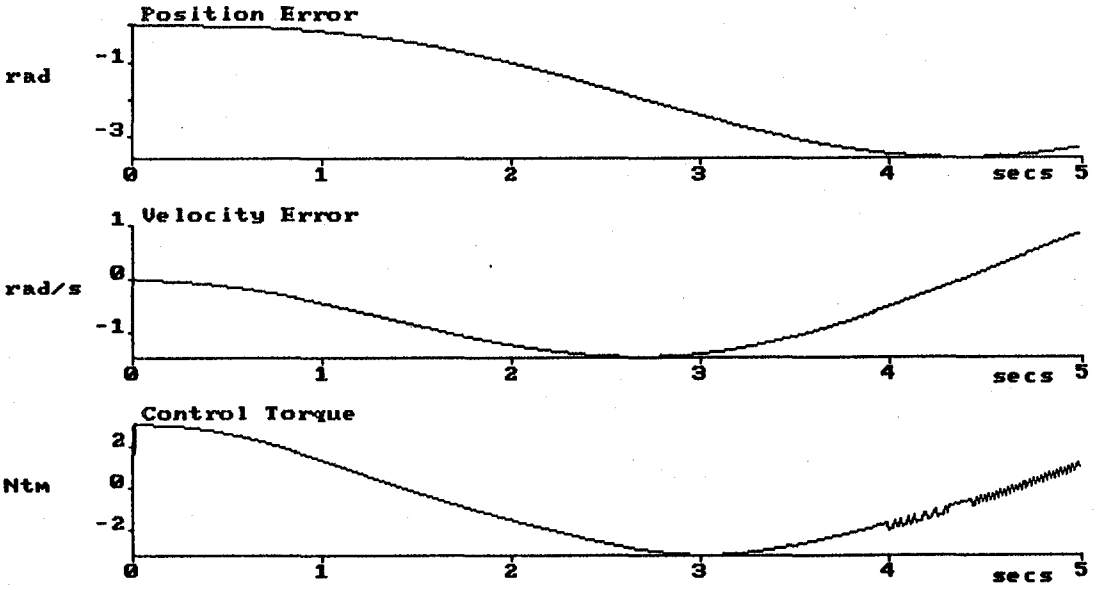


Figure 5.2 PD Motor Control Simulation

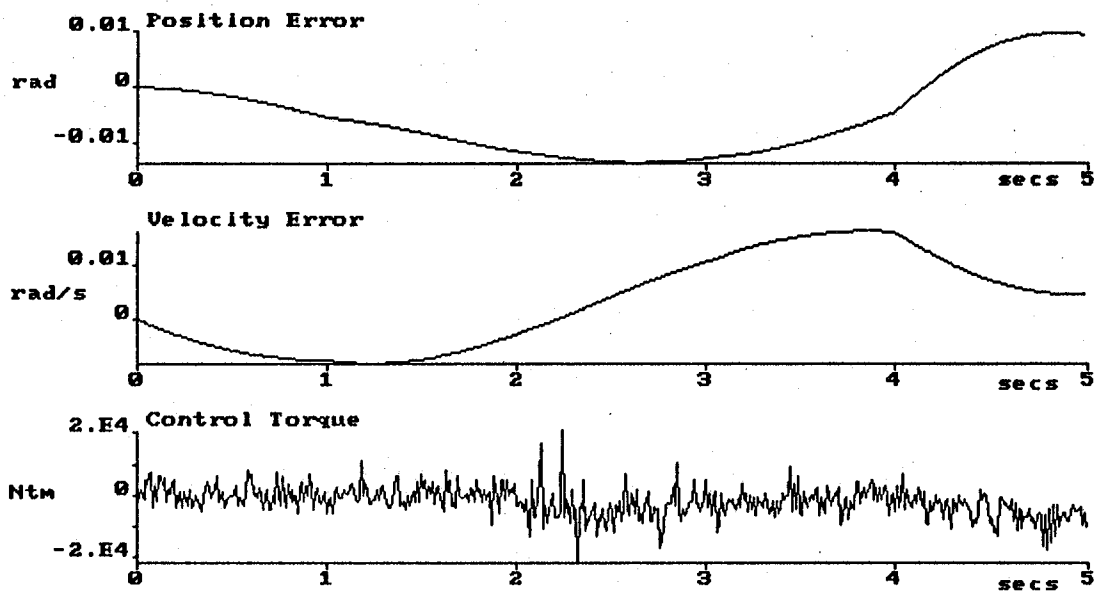


Figure 5.3 Nested PD Motor/Link/Tip Control Simulation

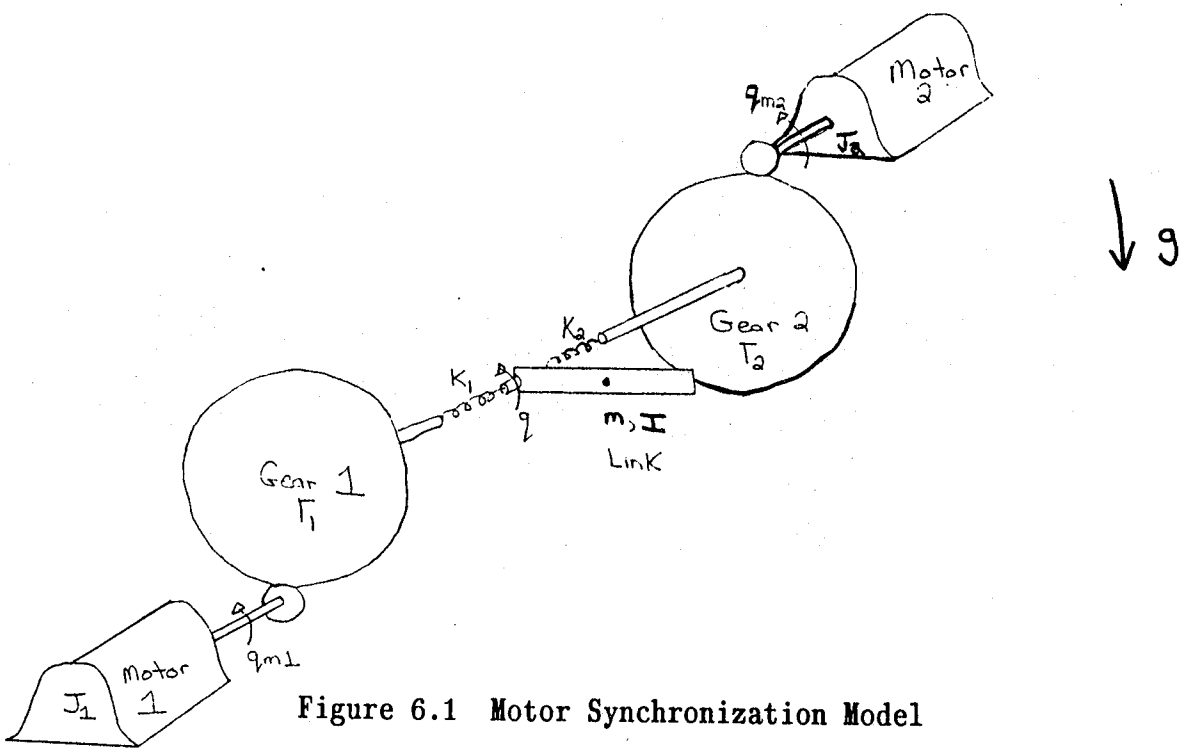


Figure 6.1 Motor Synchronization Model

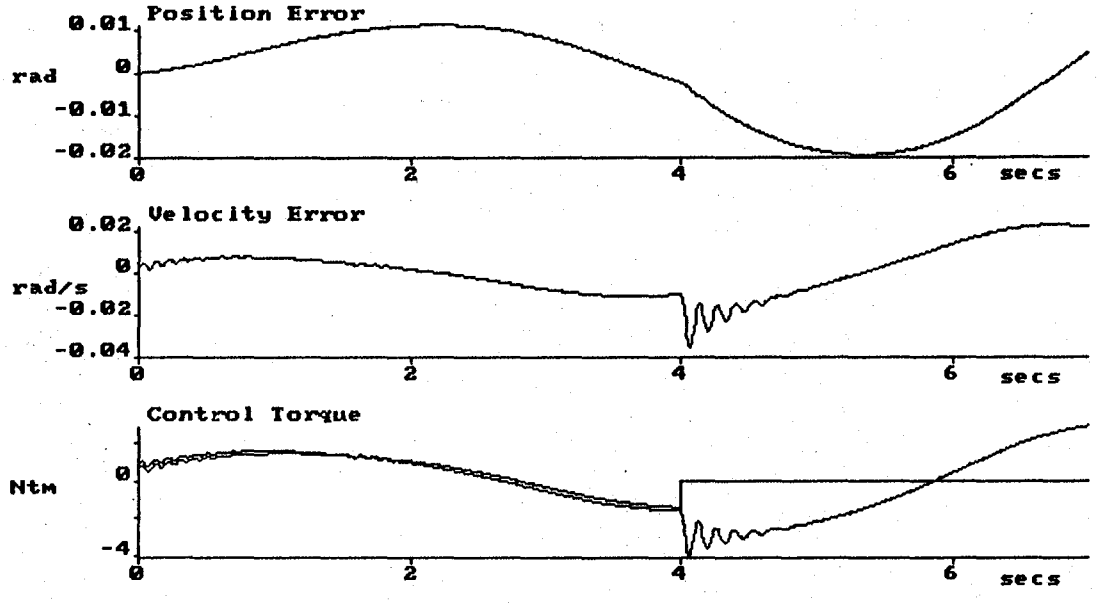


Figure 6.2 Independent Torque Control Simulation

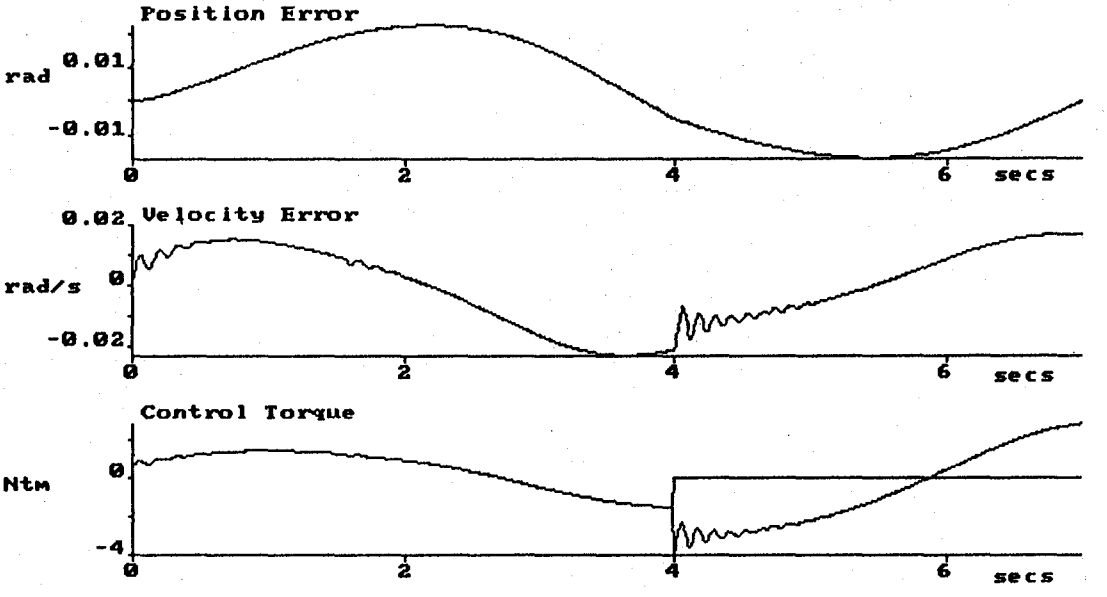


Figure 6.3 Coupled Torque Control Simulation

## IX. REFERENCES

[1] M. Spong and M. Vidyasagar, Robot Dynamics and Control, New York: John Wiley and Sons, Inc., 1989.

[2] M. Spong, "Modeling and Control of Elastic Joint Robots", Journal of Dynamic Systems, Measurement, and Control, Vol. 109, Dec. 1987, pp 310-319.

[3] R. R. Craig, Jr., Structural Dynamics: An Introduction to Computer Methods, New York: John Wiley and Sons, Inc., 1981.

[4] D. Dawson, Z. Qu, F. Lewis and J. Dorsey, "Robust Control for the Tracking of Robot Motion", International Journal of Control, 1990, Vol. 52, No. 3, pp 581-595.

[5] D. Dawson, Z. Qu, and M. Bridges, "Robust Tracking of Rigid-Link Flexible-Joint Robots", Automatica, June 1991, submitted for publication.

[6] D. Dawson, Z. Qu, and S. Deepak, "Robust Tracking Control of Vibrating Robots", American Controls Conference, June 1992, in preparation.

[7] D. Dawson, Z. Qu, and M. Bridges, "Robust Sychronization of Redundantly-Driven Rigid-Link Flexible-Joint Robots", American Controls Conference, June 1992, in preparation.

[8] D. Dawson, "Simulation and Control Routines for the Automated Radiator Inspection Device", Informally submitted to KSC Robotics Group.

[9] V. Davis, "Robotic Control and Inspection Verification", KSC internal report.