STUDY OF ROBOTIC MANIPULATORS SUBJECTED TO BASE DISTURBANCES

BY

ALBERT BUCK TANNER
B.S., UNITED STATES MILITARY ACADEMY
(1979)

SUBMITTED IN PARTIAL FULFILLMENT OF THE DEGREE OF

MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JANUARY 16, 1987

SIGNATURE OF AUTHOR Signature Redacted
DEPARTMENT OF MECHANICAL ENGINEERING
JANUARY 16, 1987

CERTIFIED BY Signature Redacted
STEVEN DUBOWSKY
THESIS ADVISOR

ACCEPTED BY Signature Redacted
AIN A. SONIN
CHAIRMAN, DEPARTMENTAL GRADUATE COMMITTEE

STUDY OF ROBOTIC MANIPULATORS SUBJECTED TO BASE DISTURBANCES

BY

ALBERT BUCK TANNER

Submitted to the Department of Mechanical Engineering on January 16, 1987, in partial fulfillment of the requirements for the Degree of Master of Science in Mechanical Engineering.

ABSTRACT

Manipulators mounted on moving platforms have many practical applications. Unlike industrial manipulators, which operate in highly structured environments, the mobile manipulator is subject to arbitrary base motion disturbances which create control problems and degrade performance. Previous research demonstrated that conventional controllers were unable to compensate for the base motion disturbances. This thesis demonstrates a control strategy to compensate for base motion. A planar, three degree of freedom manipulator is used to perform a typical application. Both an uncoupled and a coupled PD controller are designed which result in satisfactory performance when the manipulator's base is stationary, but which fail to acceptably control the manipulator with the base moving. A compensator is designed which utilizes base motion disturbance measurements. The manipulator with either PD controller, using the sensory base information, can successfully perform the task. The compensator is demonstrated experimentally using a PUMA 250 robot mounted on a mobile platform.

Thesis Advisor: Dr. Steven Dubowsky

Professor of Mechanical Engineering

ACKNOWLEDGEMENTS

The author would like to thank everyone at M.I.T. who provided so much insight and help into this research. Their thoughtfulness, engineering expertise, and guidance was always available. I would especially like to thank Mr. Zia Vafa, Mr. Joe Deck, and Mr. Bert Hootsmans for their advice and help with technical problems.

I would also like to thank Professor Steven Dubowsky for his encouragement and understanding throughout this research. His ability to clarify problems and to stimulate the interest and motivation to solve them proved invaluable.

This research was funded by an Army fellowship and support from the U.S. Army Research Institute under contract MDA903-83-C-0196 and DARPA, the Army Human Engineering Laboratory as agents under Oak Ridge National Laboratory subcontract number 19X-55970C.

I am especially indebted to my wife Lisa for all of her patience, encouragement, and love throughout my studies at M.I.T. She enabled me to devote the necessary time to my studies and still have the love and support of my family. Finally, I would like to thank my daughters, Maria, Melissa, and Melinda Jo for keeping the important aspects of life close at hand.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Robotic manipulators have gained wide acceptance in industry and have demonstrated their effectiveness in accomplishing many tasks, even those which are difficult or dangerous for humans. In such industrial applications current manipulators generally operate in highly structured environments. They are mounted on fixed, rigid bases, and are not subject to unknown base motion disturbances. However, many potential applications, especially for the military, require robots to operate mounted on mobile platforms, and, therefore, subject to arbitrary base motion disturbances.

Many military applications exist which are ideally suited to robotics. The military inherently encompasses tasks which are arduous and dangerous for humans, for which robotic replacements would be advantageous. The modern battlefield is much more dangerous than any previously experienced. Technology has made weapons more lethal, accurate, and with a greater range than ever before. Any robot which can replace humans on the battlefield, therefore, would be of obvious benefit.

The U.S. Army's current strategy for conducting warfare emphasizes mobility and rapid movement. This mobile warfare will place greater demands upon the ability to fight

on the move, and to logistically support these operations [1]. Hence, many military robotic manipulators will be mounted on mobile platforms or vehicles. Potential robotic tasks include a main gun ammunition robotic loader for tanks, forward resupply vehicles, field material handlers, and various robotic mobile weapon platforms [2].

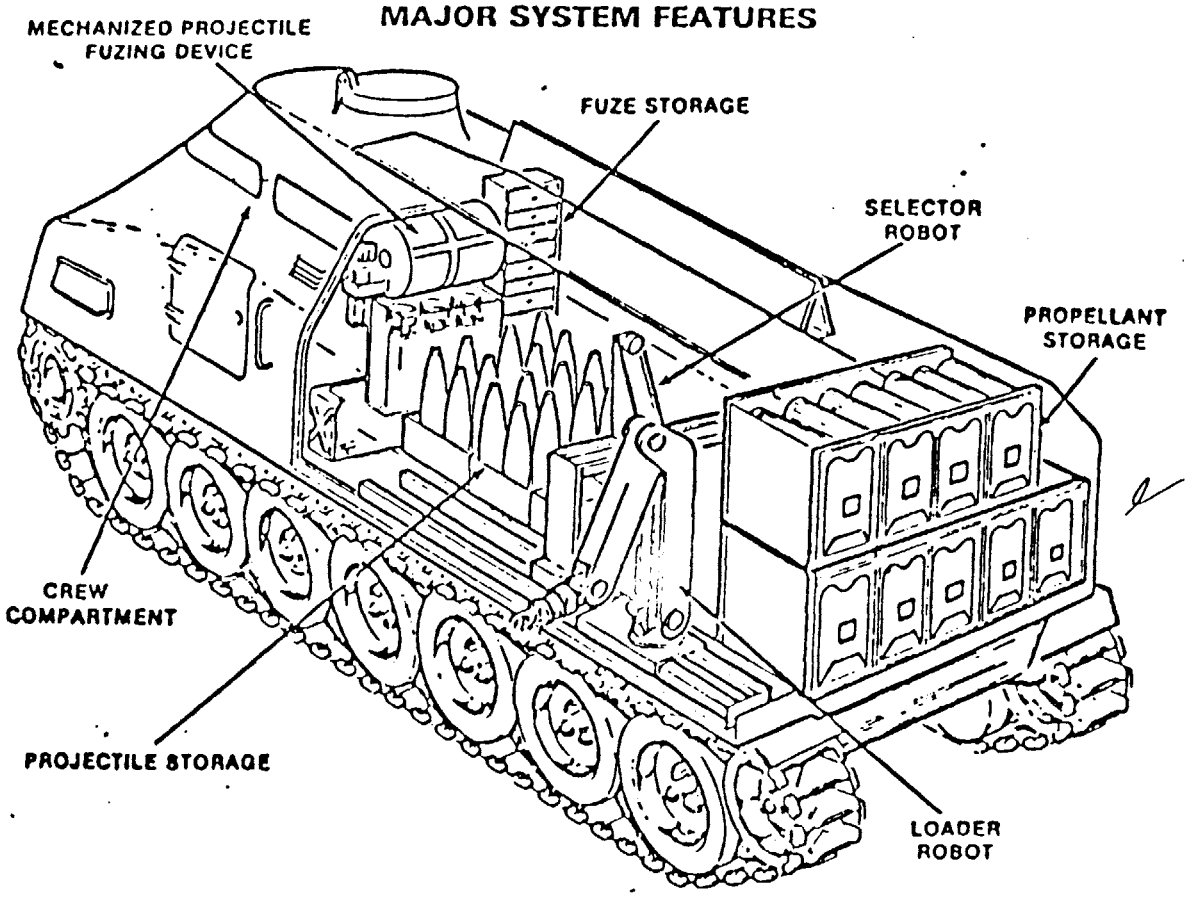

FIGURE 1-1 FORWARD RESUPPLY VEHICLE

The forward resupply vehicle, shown in figure 1-1, would incorporate robotic arms, mounted inside an armored vehicle, to prepare ammunition and supplies for distribution to individual units. The field material handler, shown in

figure 1-2, is a robot mounted on the back of a truck to redistribute material. Robotic manipulators mounted on or inside large vehicles need to be designed to perform tasks which require precise end effector control. Many of these robotic manipulators must also function accurately even while the vehicle is moving quickly across rough terrain. This motion, though attenuated by the vehicle's suspension system, will subject vehicle mounted robots to large arbitary disturbances. These disturbances can degrade system performance [3]. An important technical difference in the above applications is based on the relative mass differences between the robot and the mobile platform. In the tank auto-loader and resupply vehicle, the mass of the vehicle, to which the robot base is mounted, is very much larger than the robot. As a result, the motion of the robot has no effect on the vehicle. Conversely, the motion of manipulators mounted on small trucks or weapons platforms, will cause the vehicle to move. This subsequent robot induced vehicle motion creates additional control problems. This research will focus on the initial problem of controlling a mobile robot mounted in a vehicle with large mass, and will not include robot induced base motion. There has been relatively little previous research into the mobile manipulator problem. Some work has been presented which models vehicle disturbances and incorporates these disturbances into mathamatical models for a simple single link manipulator mounted on a moving platform undergoing

rotations [4]. Additionally, a study was conducted which considered a two link manipulator subject to vertical base motions [3].

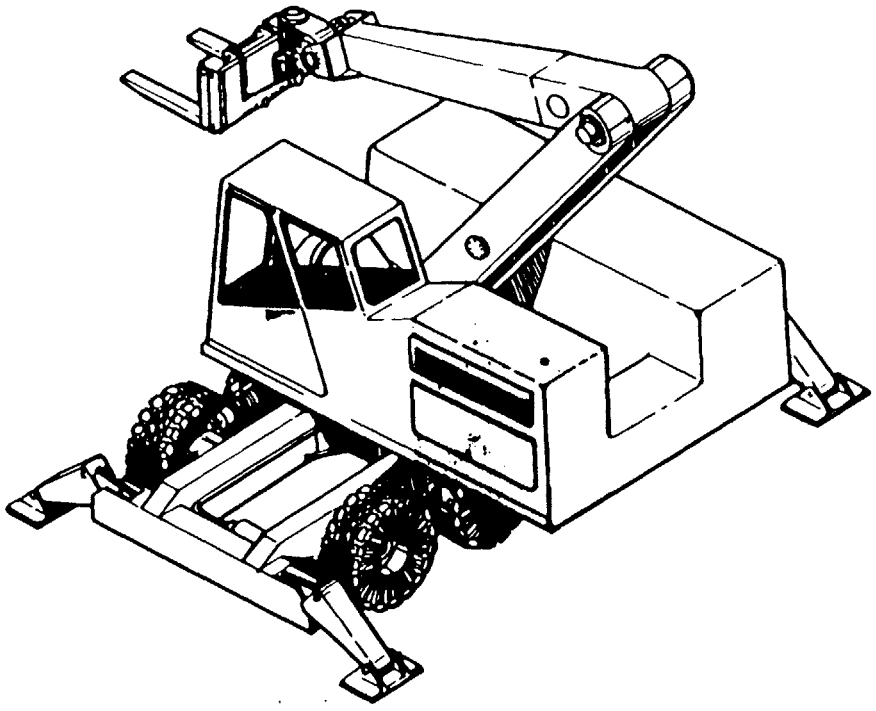

FIGURE 1-2 FIELD MATERIAL HANDLER

The movement of the base generates dynamic forces which are functions of the amplitude and frequency content of the base motion. These forces can create control problems, which among other things, cause the manipulator to leave its prescribed path, saturate its actuators, and induce high stresses in the manipulator. Rick Lynch demonstrated that conventional control strategies are unable to adequately eliminate the tracking errors caused by arbitary base vertical accelerations, without raising the control gains above that allowed by arm structural resonance

[3]. Some current control strategies deal with robustness, the ability to adapt to certain parametric uncertainty, which can be viewed as disturbances. These strategies may be useful for compensating for motion disturbances, but currently have not been used successfully to compensate for the substantial disturbances generated by a vehicle moving over rough terrain.

In order for robots to successfully accomplish mobile robot tasks, robotic technology must be increased so that the control problems are solved. A control system must be as accurate in positioning a robot while the robot's base is moving as when it is stationary. This thesis investigates the mobile robot problem and designs a compensator based on sensory information from the robot base.

1.2 ROBOTIC AMMUNITION LOADER

To completely investigate the dynamics and control of the moving base problem, a realistic application is selected. This permits a design to meet performance specifications and the determination of realistic model parameters. Actual parameters, to include disturbance models, result in complete equations of motion. From these equations, the effects of base motion and disturbances can be determined, and a compensator designed to provide satisfactory performance even when the base is moving. The application selected is a robotic main gun ammunition loader

for the current U.S. Army main battle tank.

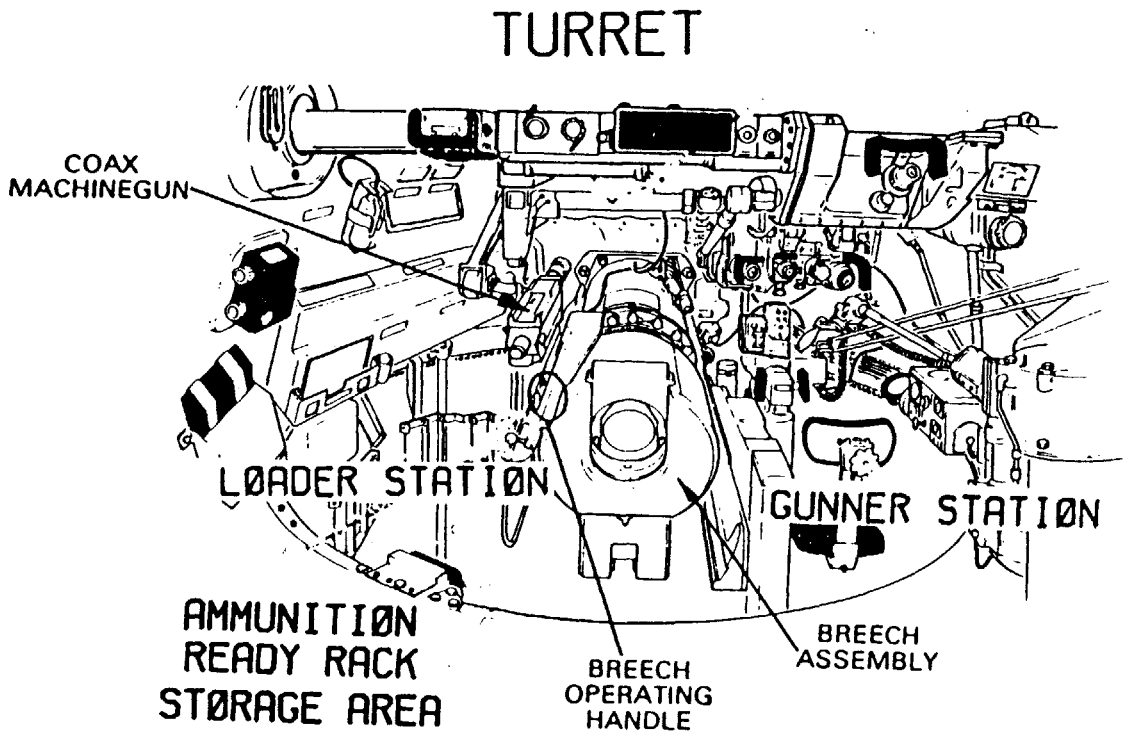

FIGURE 1-3 HUMAN TANK LOADER POSITION

A robotic ammunition loader would be very beneficial. At present, the four-man tank crew consists of a commander, a gunner, a driver, and a loader. The loader receives verbal commands to load a particular type of ammunition, manually selects the designated type of ammunition from a storage rack, removes the round from the storage area, inserts the round into the breach, and reports the gun's readiness to fire [5,6]. The tank crew's "loader" has an extremely difficult and hazardous task physically loading main gun ammunition. It is practically impossible to load the main gun while the vehicle is moving, especially

during cross-country movement over rough terrain. The difficulty and danger is compounded during stabilized operation, when the breach is moving relative to the turret, as movement of the tank is compensated for. The loader must also stay away from the recoil path of the breach, which as shown in figure 1-3 dominates the loader's station, dodge the hot, empty ammunition casings when ejected, and find footing among the empty casings littering the turret floor. This task will become increasingly more difficult as the weight of the main gun ammunition increases. With the increased protection that technology is providing armor vehicles, there is a necessity for larger and more powerful ammunition rounds. This results in heavier rounds which are already difficult for the human loader to handle.

Success and survival on the modern battlefield, with its increased lethality, will depend upon the design of a mechanism to transfer ammunition for the main gun, the tank's primary armament, from the stowage magazine and ram the ammunition round into the chamber quickly and automatically [4]. Many advantages are gained though the use of a automatic loader. A vehicle with a greater rate of fire is more likely to survive, and the endurance of the human loader is limited. Additional benefits of a automatic loader are the reduced work space requirements, which results in the reduction of vehicle size, and the increase of the number of rounds immediately available for use. The advantages of a robotic loader over a mechanical reload

mechanism is the flexibility offered by a robotic system. Existing mechanical reload mechanisms for tanks are limited in ability. These mechanisms use generally one or two ammunition magazines which offer limited round selection. Additionally, the magazines are limited in capacity and must be manually resupplied from other ammunition storage areas. Conversely, the robot can be programmed to access any desired round immediately from the many different type rounds available. The robot may also be programmed to accomplish a variety of tasks between missions, such as rearranging the ready rounds to optimize placement in the magazines and to resupply the ready magazine from the other ammunition storage areas. An additional benefit is the ability to adapt a robot system to current turret designs, and the ability to add it to existing vehicles.

1.3 THESIS ORGANIZATION

In order for robots to meet military requirements, the dynamics and control problems encountered when a robot is mounted on a moving base must be solved. This research determines the effect of base disturbances on robot accuracy, and designs a controller to compensate for the base disturbances. This is accomplished by designing a robotic system, with realistic characteristics for application as a tank main gun loader. Nonlinear equations of motion, which incorporate disturbance motions, are developed using the Lagrangian formulation, and then

linearized in order to design a linear controller. A proportional-derivative (PD) controller is designed which meets the performance specifications, with the vehicle stationary, without exciting the system's structural resonances. The effect of base disturbances is investigated and the dominate disturbance terms determined. A compensator is then designed which uses measured linear acceleration and angular velocity data to calculate the torque generated by the disturbances and then feed forward a corresponding torque to counter the disturbance effects. It was found that a PD controller, with compensator, provided good performance and met all specifications.

This research determines that rotational acceleration measurements are not required for good performance. The robustness of the system to operate with the variety of ammunition weights, or payloads, is investigated. The compensator is then implemented experimentally using a PUMA 250 robot controlled by a PD controller from a DIGITAL PDP 11/73 computer. The experiments demonstrate that the compensator is practical, easily implemented, and effective in maintaining robot accuracy despite base motion disturbances.

CHAPTER 2

ROBOT DESIGN

2.1 MODELING

Modeling is critical in investigating physical systems. Real systems are often extremely complex making them difficult to exactly simulate. Models of physical systems are developed to aid in understanding their functions, investigate their performance, and design better systems. Models are designed by applying certain simplifying assumptions to the real system, without making misleading oversimplifications. For this research, a model of the robotic tank loader system is developed so that significant system behavior, particularly the system's response to base motion disturbances, can be analyzed. The procedure used to accomplish a robotic system design first requires identifying the design specifications. These specifications are described in section 2.3.

The robotic loader is constrained to operate within a predetermined work space, a tank turret, and to manipulate provided payloads, tank main gun ammunition. These constraints, and their representative models are described in section 2.4. Robotic link and actuator parameters are iteratively obtained to achieve the design specifications and system constraints. The parameters are listed in section 2.5. Determining a baseline trajectory for a typical ammunition loading cycle, section 2.6, and

disturbance models, section 2.7, complete the modeling process. The final section describes the performance specifications the robotic ammunition loader must achieve to be successful.

2.2 DESIGN SPECIFICATIONS

The robotic system designed to replace the human "loader" in the current tank crew must accomplish the desired result of loading main gun ammunition. Design is an iterative process which continues to change the system as improvements are made. This process continues until a successful product is obtained. For a successful design, a conceptual system representing a tank robotic loader must accomplish the following specifications:

(1) The robotic loader must be capable of selecting the desired round and inserting the ammunition round through the breach into the chamber of the main gun. Additionally, the robot needs the ability to access all the ammunition storage areas within the tank turret in order to complete resupply and redistribution of ammunition.

(2) The system needs to be robust enough to follow a prescribed trajectory path. The ammunition can not be permitted to strike any object during the loading cycle.

(3) The robotic loader's controller frequency should not excite the robot's structural frequencies. This prevents degraded performance and possible instabilies.

The first specification determines the size and composition of the robot. The loading process requires the

robot to specify the position and orientation of the end effector, which necessitates three actuators. The second and third specifications require an iterative process to obtain the required actuators, to prevent saturation, and to obtain the robotic parameters. The iterative process continues until a robot system is designed which is robust enough to handle the required actuator torques and prevent exciting the natural frequencies, but which can still be controlled and meet performance specifications.

2.3 SYSTEM CONSTRAINTS

The robotic loader's workspace is constrained by the physical dimensions of the U.S. Army's main battle tank. A representative tank is shown in figure 2-1. Approximate dimensions for the turret are indicated, as actual dimensions are classified. Figure 2-2 is the model used to represent the robotic manipulator's work space, the tank turret. This figure depicts a cross sectional view of the side of the turret. The front of the turret, which contains the main gun, faces to the left. The breach is located inside the turret and is opposite the ammunition 'ready rack' storage area. The work space coordinate system (X,Y) is oriented as shown in figure 2-2.

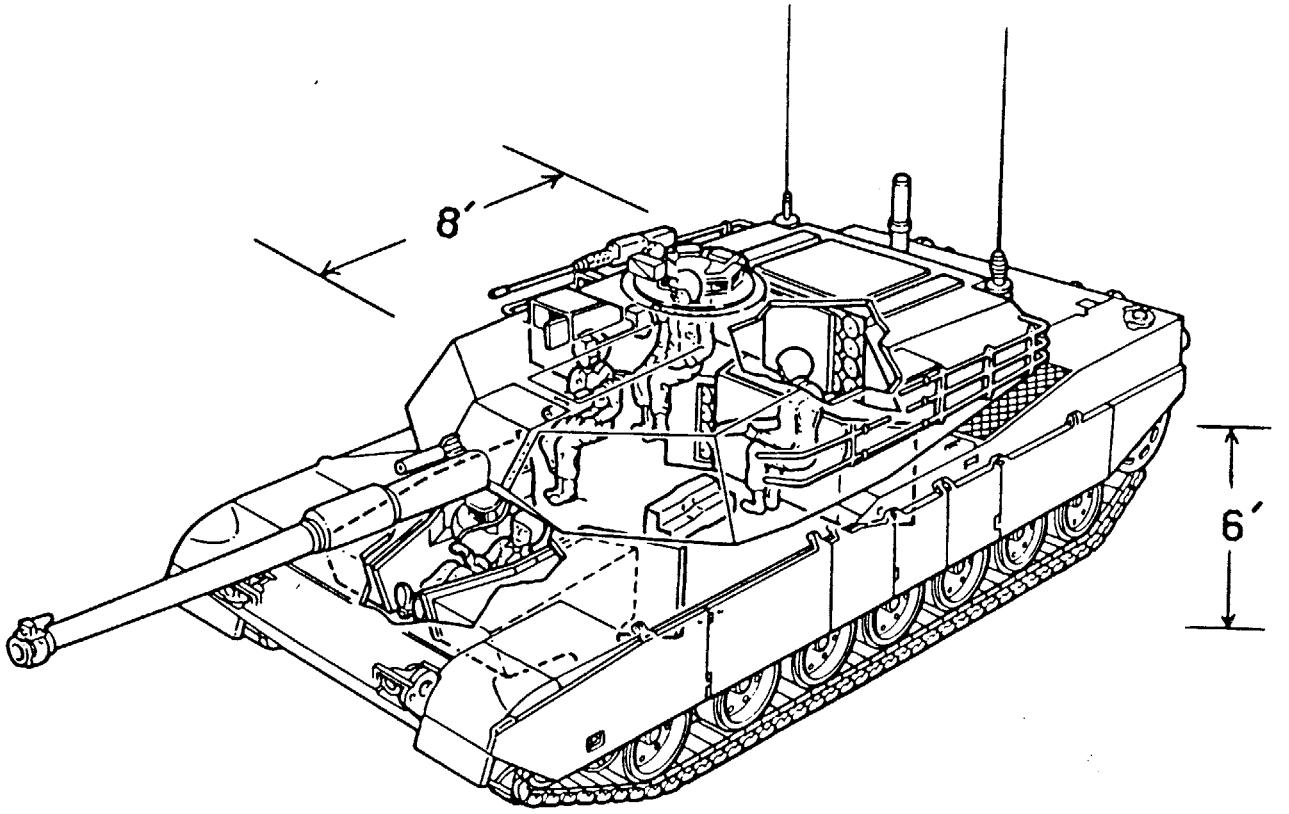

FIGURE 2-1 MAIN BATTLE TANK

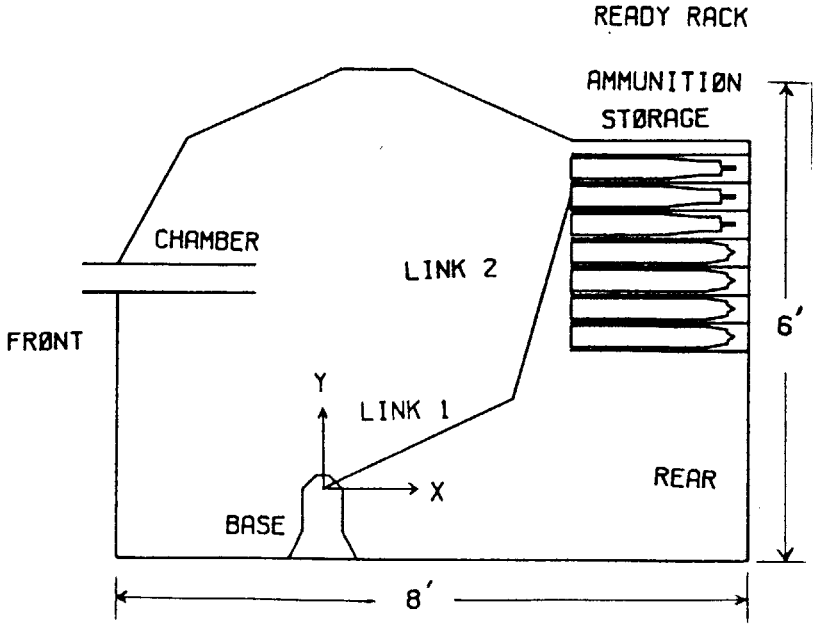

FIGURE 2-2 TURRET MODEL

Payloads for military robots will vary greatly. This requires a robustness which can tolerate a wide range of payload parameters. The Army utilizes many different tank main gun ammunitions, but uses primarily anti-tank ammunition. These different ammunitions vary in weight and size. Additionally, the caliber of the round has a great impact on the weight of a round. Currently, 105mm caliber ammunition is used, but 120mm will soon be incorporated. For this research, two representative types of anti-tank ammunition are employed [5,7]:

(1) HEAT is a high explosive anti-tank round of main gun ammunition. 105 mm HEAT rounds weigh 48.06 pounds and are 39.02 inches long. The model used to represent the HEAT ammunition round is shown in figure 2-3. A model weight of 60.0 pounds is used to make this model representive of the heavier 120mm rounds.

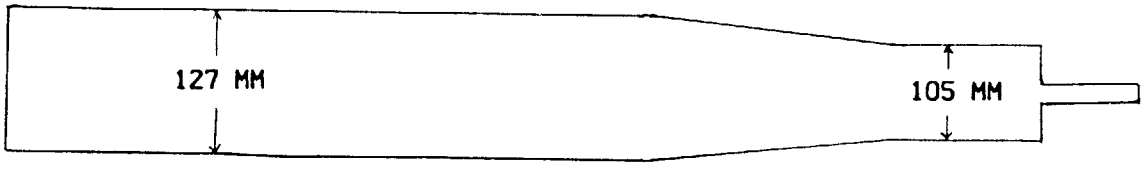

FIGURE 2-3 HEAT AMMUNITION MODEL

(2) APFSDS-T is an armor piercing, fin stabilized, discarding sabot ammunition round. 105mm APFSDS-T rounds weigh 38.14 pounds and are 36.57 inches in length. The

corresponding model for the APFSDS-T ammunition round is shown in figure 2-4, and a model weight of 40.0 pounds is used. The two modeled ammunitions provide a representation of the various weights a robotic loader will need to manipulate.

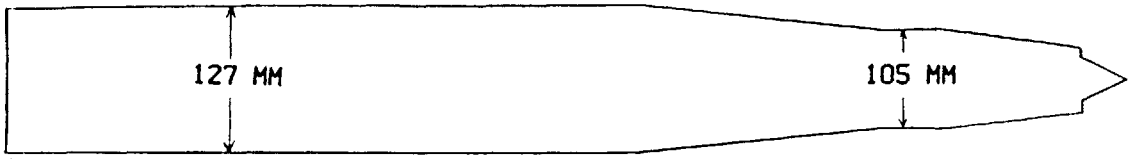

FIGURE 2-4 APFSDS-T AMMUNITION MODEL

2.4 MODEL PARAMETERS

Model parameters are determined based on work space dimensions and the payload requirements. The robotic loader model employed to represent the loader system is shown in figure 2-5.

Hydraulic rotary actuators are selected with enough torque capacity for gravity compensation, joint acceleration, and disturbance rejection. Hydraulic actuators provide good weight to torque capability and are robust. Bird-Johnson hydraulic rotary actuators are selected [8]. Details for model parameter calculations are in Appendix A. The final model parameters, after several iterations adjusting actuators and link parameters to achieve good structural frequency, are presented in table 2-1.

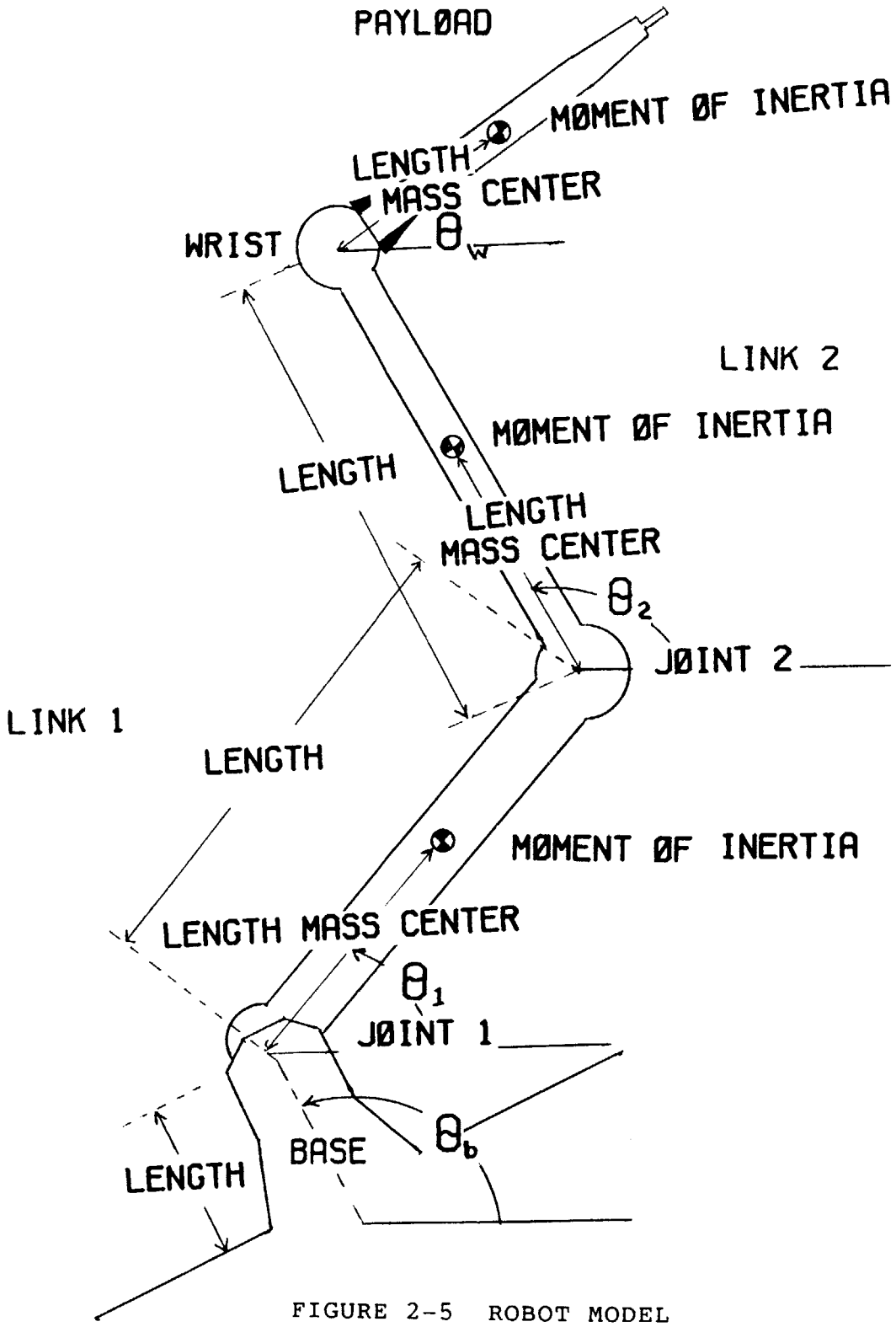

FIGURE 2-5 ROBOT MODEL

Parameter	Link 1	Link 2	Wrist with payload	Wrist without payload
inner radius (inch)	3.4	3.0		
outer radius (inch)	4.4	3.9	5.0	5.0
length (inch)	36.0	36.0	30.0	3.0
mass ($\frac{\text{lb sec}^2}{\text{feet}}$)	9.32	6.77	2.64	0.78
mass center (inch)	1.57	1.49	0.75	0.17
moment of inertia (lbf sec^2 ft)	7.25	5.22	1.47	0.03

actuators:

type	SS 12/25	SS 4/8	SS 3/3
weight (lb)	100	40	43
capacity (in lb)	70965	21700	6840

TABLE 2-1 SYSTEM PARAMETERS

To properly construct a controller, the mechanism's structural frequency must be calculated, or at least bounded. Generally, the controller frequency must be lower than the structural frequency to prevent exciting the mechanism's structual resonances [9]. There is currently research being conducted that investigates the use of controllers which operate within the natural frequency range, but this is only a research topic and not ready for implementation [10]. A lumped parameter method, with the robot arm held horizontal, is used to estimate the system's structural resonances. Utilizing beam theory, assuming locked actuators, and using lumped parameter modeling, results in a structural resonance of 15 hertz with payload and 24 hertz without payload. The equations and procedure used to perform these calculations are in Appendix B.

Certain assumptions are made for developing this model to simplify the actual system, but still emulate the actual system behavior. These assumptions are listed below:

(1) Manipulator motion is restricted to planar motion. Since the tank turret is circular and the manipulator is centrally located the manipulator motion out of plane will be similiar to the modeled planar motion.

(2) Manipulator links are assumed to be rigid. This is valid considering the robust robotic system necessary to move the heavy payloads within the required times and during large dynamic disturbance forces. Weight is not a consideration on a tank as it is in space applications.

(3) Actuator friction, backlash, inertia, etc. are neglected. Hydraulic actuators are direct drive

motors so there is no transmission losses. Compressibility of the hydraulic fluid is not considered. The relative effect of actuator parameters to the rest of the system is small.

(4) The manipulator does not impose any motion upon the base, the tank. This assumption is valid considering the differences in masses, 3200 slugs (102,000 pounds) for the tank versus 18 slugs (500 pounds) for the complete robot system including payload.

2.5 TRAJECTORY

The typical trajectory used to conduct this research is shown by figure 2-6, which depicts the movement of the end effector relative to the tank turret cross-section. The work space coordinate system, (X,Y), used in figure 2-6 is shown in figure 2-2, where the coordinate axes are positioned at the first joint. The straight line sections, of the trajectory, are required for the extraction of the round of ammunition from the storage racks and for alignment and insertion of the round through the breach into the chamber. The wrist swings the round of ammunition around, reversing the direction of the round, as soon as the tip of the round has cleared the storage rack. The wrist completes the swing prior to inserting the round into the breach. This trajectory is representive of a loading cycle. Actual trajectories will depend upon the location of the desired round. Probability analysis and optimization routines could be utilized to determine the best locations and order of use for ammunition, and improve upon the developed trajectory. While not optimal, this trajectory provides a realistic

baseline for conducting simulations.

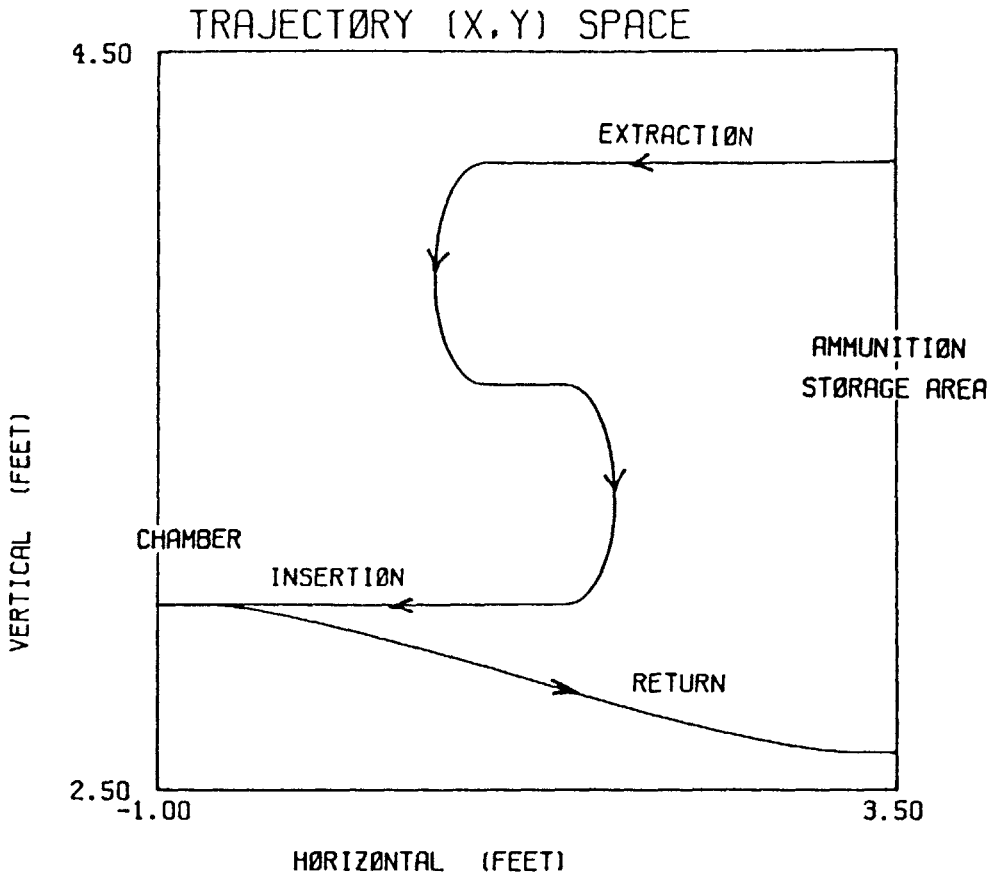

FIGURE 2-6 SAMPLE TRAJECTORY

To obtain the given trajectory in joint coordinates, equations of path segments in turret work space coordinates (X,Y) are created and then transformed by inverse kinematics to the required joint angles (Θ_1 - joint 1, Θ_2 - joint 2, Θ_w - wrist) defined in figure 2-5. The inverse kinematics derivation and the computer program which computes the trajectory joint angles are in Appendix C and Appendix E respectively. The resulting equations for joint 1 and joint 2 are given by equations 2.2 and 2.3.

$$\phi_2 = -\cos^{-1}\left[\frac{(X^2 + Y^2 - L_1^2 - L_2^2)}{(2L_1L_2)}\right] \qquad (2.1)$$

$$\theta_1 = \tan^{-1}\left[\frac{Y}{X}\right] - \sin^{-1}\left[\frac{L_2 \sin\phi_2}{(X^2 + Y^2)}\right] \qquad (2.2)$$

$$\theta_2 = \theta_1 + \phi_2 \qquad (2.3)$$

where: θ_1 is absolute joint angle 1.

θ_2 is absolute joint angle 2.

ϕ_2 is relative joint angle 2.

The resulting joint angles for the baseline loading cycle are shown in figure 2-7. These angles result in the desired baseline trajectory, where the robotic loader extracts a round from the 'ready rack' storage area, inserts this round into the breach, and then returns for the next round. The robot motion to complete these actions is shown in figure 2-8.

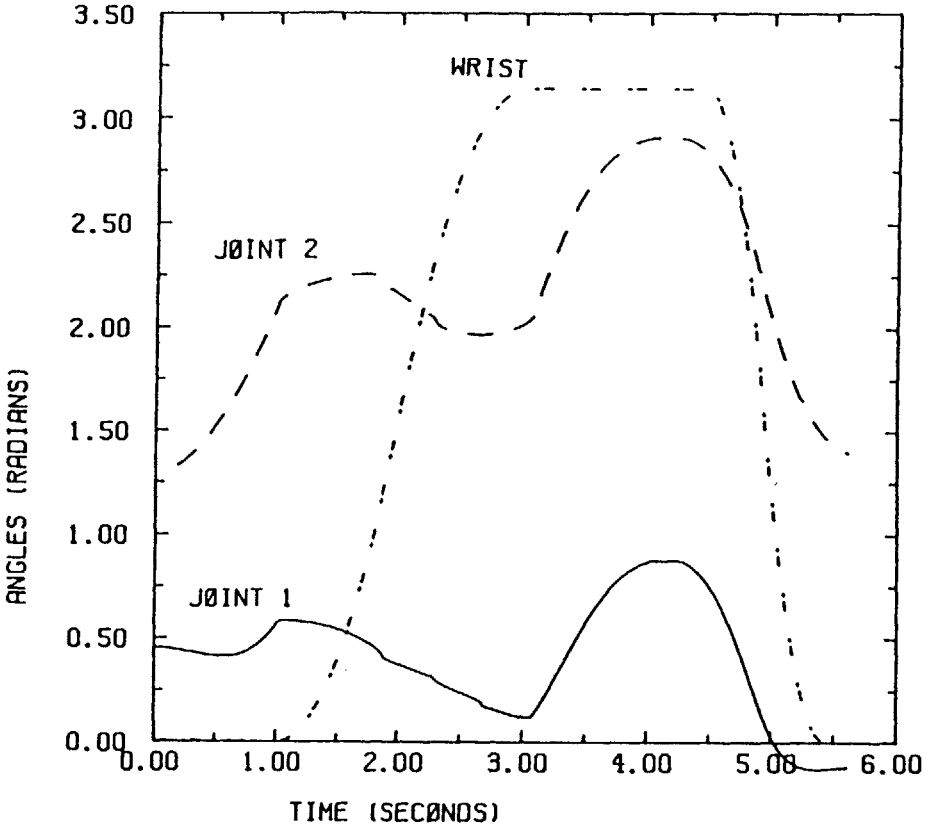

FIGURE 2-7 REQUIRED JOINT ANGLES FOR TRAJECTORY

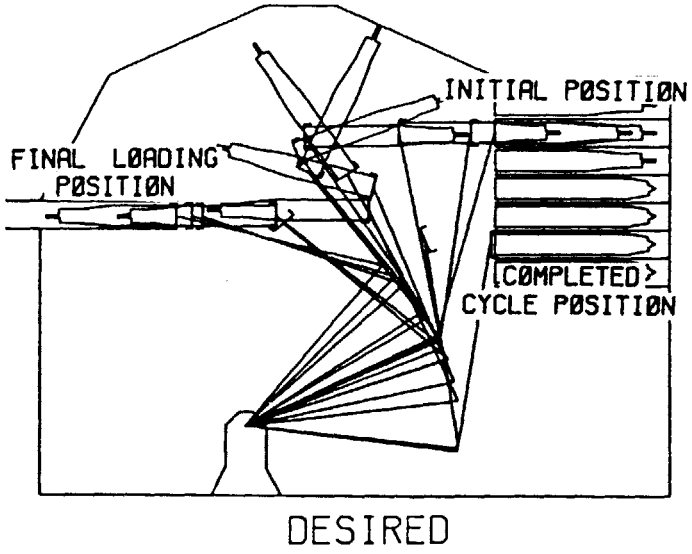

FIGURE 2-8 DESIRED TRAJECTORY

2.6 DISTURBANCES

Tanks are now more powerful and capable of higher rates of speed, close to sixty miles per hour. When a tank is maneuvering, the tank will move quickly cross country, not necessarily taking the time to travel on the smoothest route. Due to the ground's irregularities, the tank is subject to random accelerations of high magnitude. It is when the tank is experiencing these accelerations, that a robotic loader must load ammunition. Information was initially obtained which gave vertical and pitch accelerations for an armored vehicle [11]. To reconstruct this information and adjust it for more severe terrain conditions, the amplitudes are increased by fifty percent. The frequency content is obtained by combining sine waves.

$$Y = 30\cos 5t + 8\sin 45t + 5\sin 66t \qquad (2.4)$$

$$\theta_b = 6\cos 16t + 3.5\sin 43t + 2.5\sin 3t \qquad (2.5)$$

These models, shown in figures 2-9 and 2-11, are used to design and test the controller and compensator. Later, data was received from the U.S. Army Tank Automotive Command which was actually measured on the M1 Abrams main battle tank [12]. These disturbances, shown in figures 2-10 and 2-12, are used to verify the performance of the designed controller and compensator. The tank's rotational center is placed below the robot's first joint, so that rotational

disturbances impose vertical and horizontal accelerations to the robot's base.

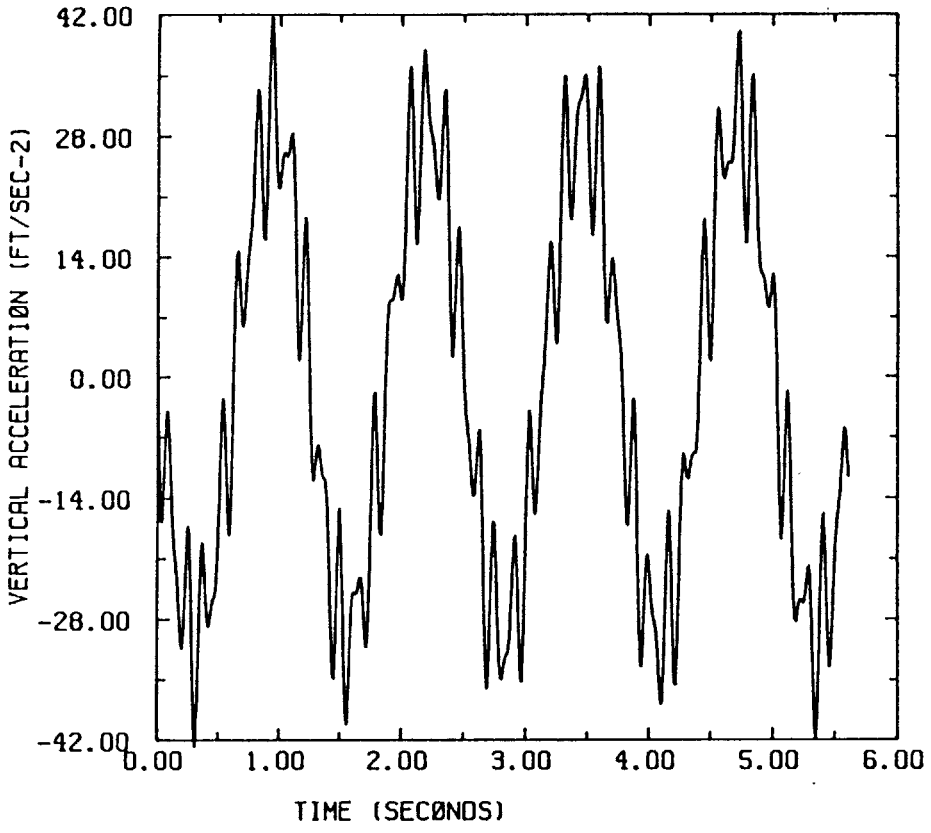

FIGURE 2-9 MODEL FOR VERTICAL DISTURBANCE

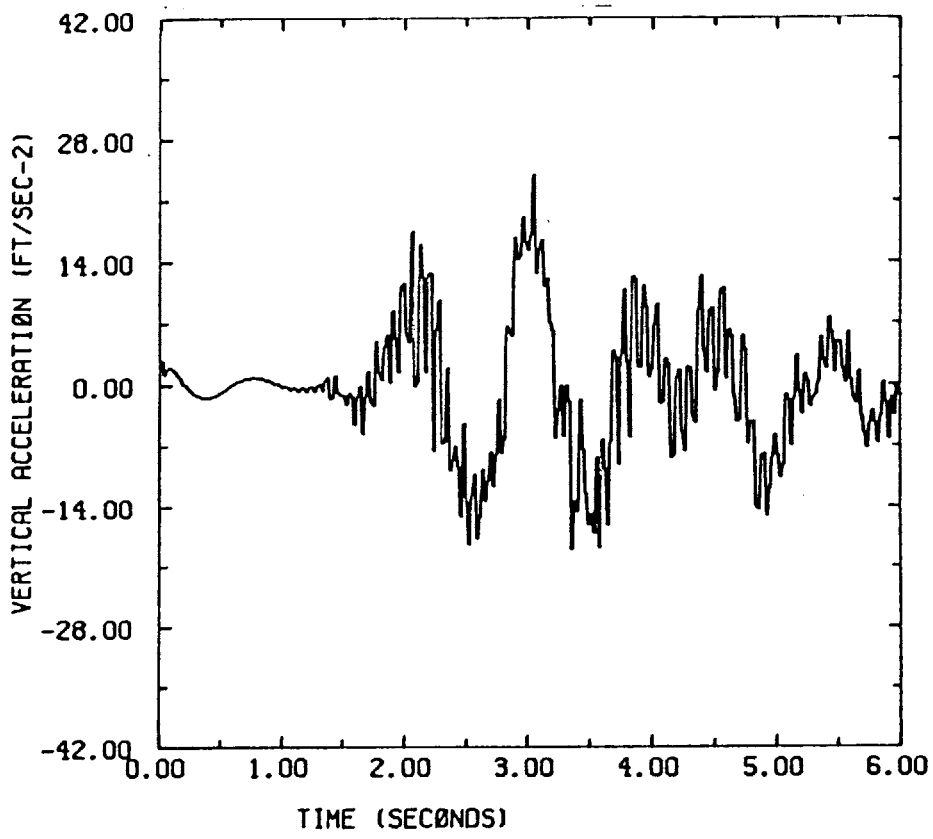

FIGURE 2-10 ACTUAL TANK VERTICAL ACCELERATION

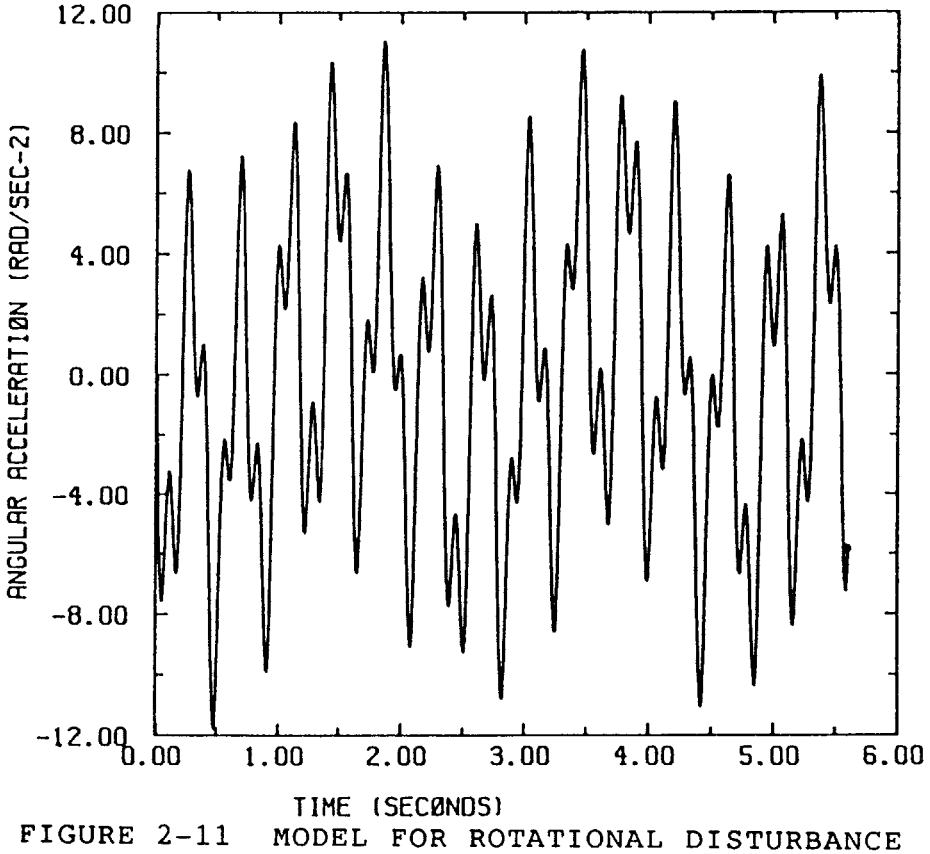

FIGURE 2-11 MODEL FOR ROTATIONAL DISTURBANCE

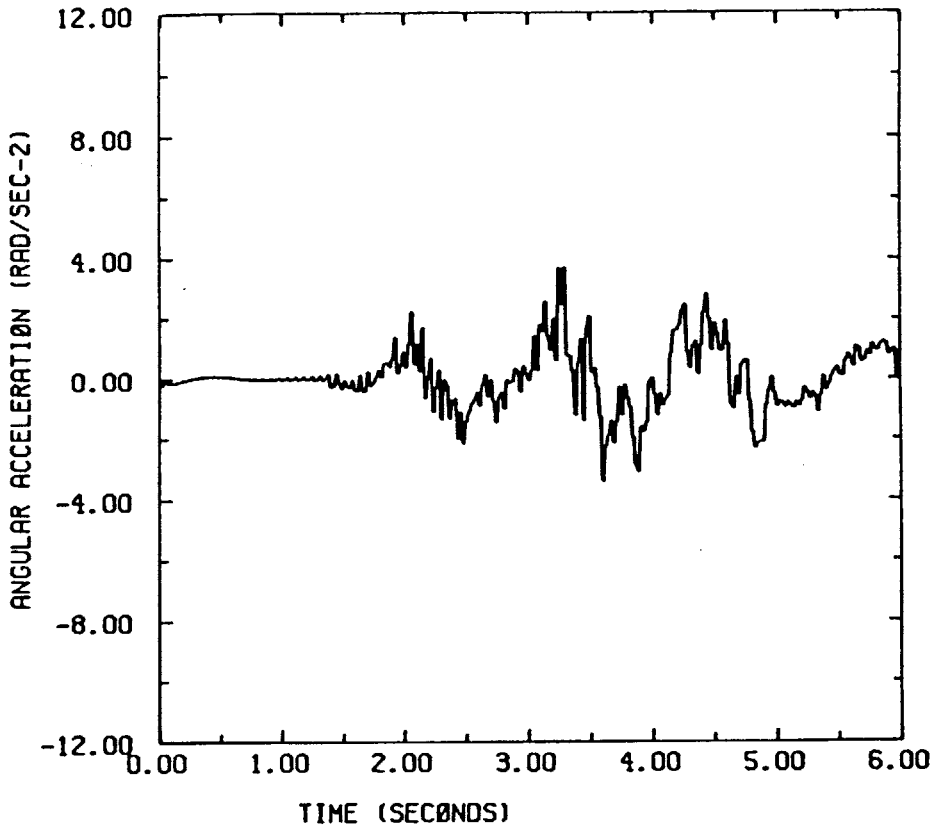

FIGURE 2-12 ACTUAL ROTATIONAL DISTURBANCE

2.7 CONTROL SYSTEM PERFORMANCE SPECIFICATIONS

The robot designed to replace the human "loader" in the current tank crew needs to meet certain minimum performance specifications. These specifications are determined from the system's design mechanical parameters and to match the capabilities of a human loader. The resulting performance specifications which must be met are as follows:

(1) Maximum Torque: The modeled robot has limited torque capability. This limitation is determined by the

selected actuators. The maximum torque, at which saturation occurs is the following for each joint:

Joint 1: +/- 5913 (ft lb)

Joint 2: +/- 1808 (ft lb)

Joint 3: +/- 570 (ft lb)

(2) Eigenvalues: The closed-loop system eigenvalue, with controller, are bounded by the natural frequency of the robot arm. The controller frequencies need to be substantially lower than the natural frequencies to prevent reduced performance from excitation of natural frequencies. This means that the imaginary portion of the eigenvalues should be less than 30 radians/second, since the lowest natural frequency is 94 radians/second.

(3) Cycle Time: The robot loader needs to meet or exceed the performance expected from a human loader. Currently, the human loader for the stationary M1 Abrams tank is required to load a round and be prepared to load a second round within six seconds [6]. No requirement currently exists for a moving tank.

(4) Tolerance (during insertion): The shape of the tank ammunition allows for an error during insertion. The

tolerance values are described by figure 2-13. This tolerance decreases as the round enters the chamber, and is listed in table 2-2 for different insertion distances.

Insertion Distance (feet)	0.0	0.2	0.5
Vertical Tolerance (feet)	0.12	0.08	0.05
Angular Tolerance (degrees)	15	15	0

TABLE 2-2 INSERTION TOLERANCES

As the round enters the chamber, the round slides against the chamber walls and there is no tolerance. The controller must switch to force control from position control after the insertion distance is greater than 0.5 feet to push the round the remaining distance into the chamber and to lock the breach. During extraction of the round from the ammunition storage rack, the round is assumed to be constrained by the rack to move straight out. The round slides against the sides of the rack, necessitating force control during extraction. Hence, the only critical tolerance limits are the tolerances imposed during insertion of the ammunition into the breach.

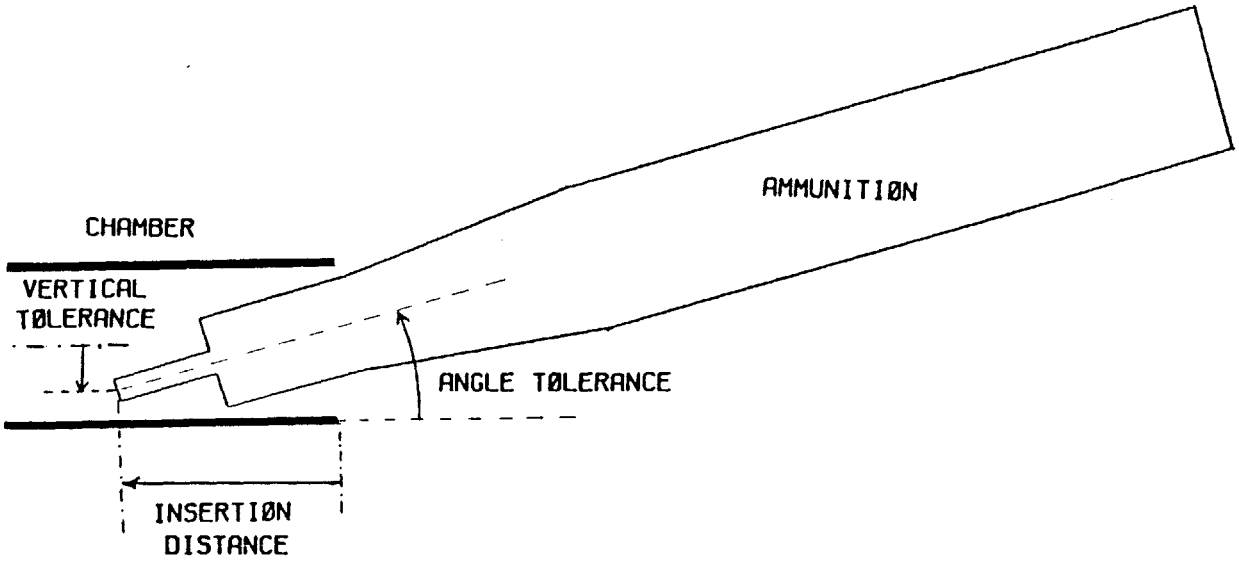

FIGURE 2-13 ACCURACY TOLERANCE

CHAPTER 3

EQUATIONS OF MOTION

3.1 INTRODUCTION

The dynamic behavior of the robot is described in terms of the time rate of change of the arm configuration in relation to the joint torques exerted by the actuators. These relationships, expressed as a set of differential equations, are called equations of motion. These equations are used to develop a mathematical model of the designed conceptual robotic loader system. Simulations and tests can then be conducted digitally with a computer to obtain information on the system's performance. The Lagrangian formulation will be used to develop the equations. The Lagrangian formulation describes the system's dynamic behavior in terms of work and energy stored in the system using generalized coordinates (Θ_1, Θ_2, Θ_w), which are shown in figure 3-1 [13,14]. All angles are measured relative to the inertial reference frame (absolute joint angles).

3.2 DERIVATION OF EQUATIONS OF MOTION

The model for the robotic tank ammunition loader consists of two links and a wrist with end effector. These three degrees of freedom permit locating the end effector in work space coordinates and specifying the orientation of the wrist. Two disturbances are modeled, vertical motions and rotational movement in the plane of motion, pitch. The

rotations are about a point below the robot's first joint. Horizontal motion was not considered significant as these motions are generally created through driver commands and are limited by vehicle capibilities for accelerating and stopping. It should be noted that horizontal motion is imposed on the robot base through components of rotational motion. The system is described in figure 3-1.

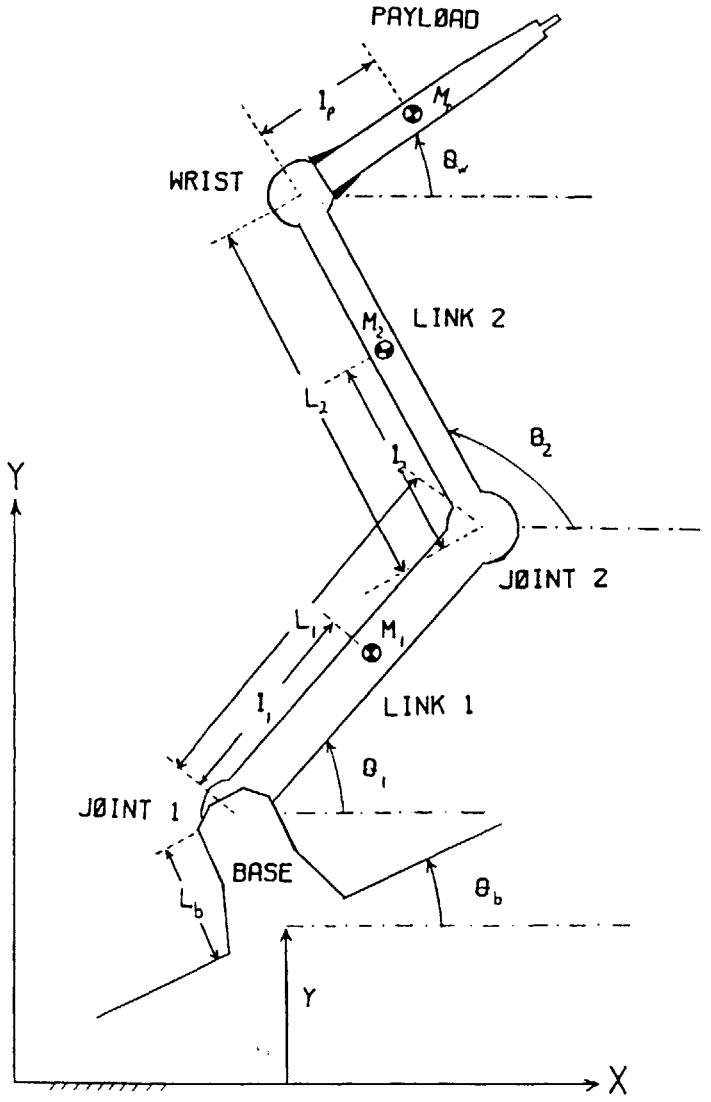

FIGURE 3-1 SYSTEM MODEL

where:

M_1 = mass of link 1

M_2 = mass of link 2

M_p = mass of wrist and payload

l_1 = length to mass center of link 1

l_2 = length to mass center of link 2

l_p = length to mass center of payload

L_b = length to tank rotational center

L_1 = length of link 1

L_2 = length of link 2

J_1 = moment of inertia of link 1 about link centroid

J_2 = moment of inertia of link 2 about link centroid

J_p = moment of inertia of payload about centroid

The Lagrangian formulation is based on equation 3.1.

$$\frac{d}{dt}\frac{\partial L}{\partial q_i} - \frac{\partial L}{\partial q_i} = Q_i \quad \text{for } i = 1,..,n \qquad (3.1)$$

where: $L = T - U$

T = kinetic energy

U = potential energy

Q_i = generalized force

q_i = generalized coordinate

The generalized coordinates are the three joint angles, θ_1, θ_2, θ_w. The terms used in equation 3.1 are determined as follows:

KINETIC ENERGY

The system kinetic energy is the sum of the kinetic energies of the individual elements. The kinetic energy is determined by the linear and angular velocity of the links and in general is determined by equation 3.2.

$$T = \sum_{i=1}^{n} [.5M_i V_i V_i^t + .5w_i^t J_i w_i] \qquad (3.2)$$

where: V_i = linear velocity vector of CG link i

w_i = angular velocity vector of link i at joint i

n = number of links in the system

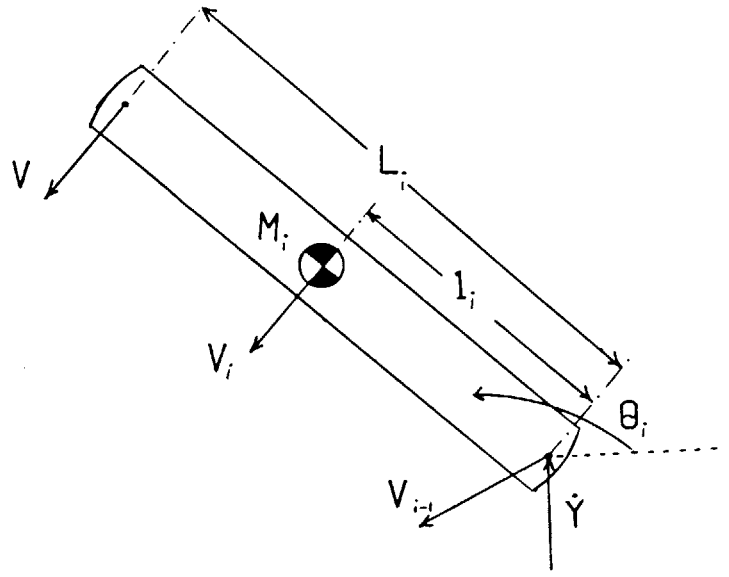

FIGURE 3-2 LINK i KINETIC ENERGY

POTENTIAL ENERGY

The system potential energy is the sum of the potential energy for the individual elements. Potential is due to gravity and, in general, is calculated using equation 3.3.

$$U = \sum_{i=1}^{n} M_i g h_i \qquad (3.3)$$

where: g = gravity

h_i = height CG of link i is above reference
$= l_i \sin\theta_i$

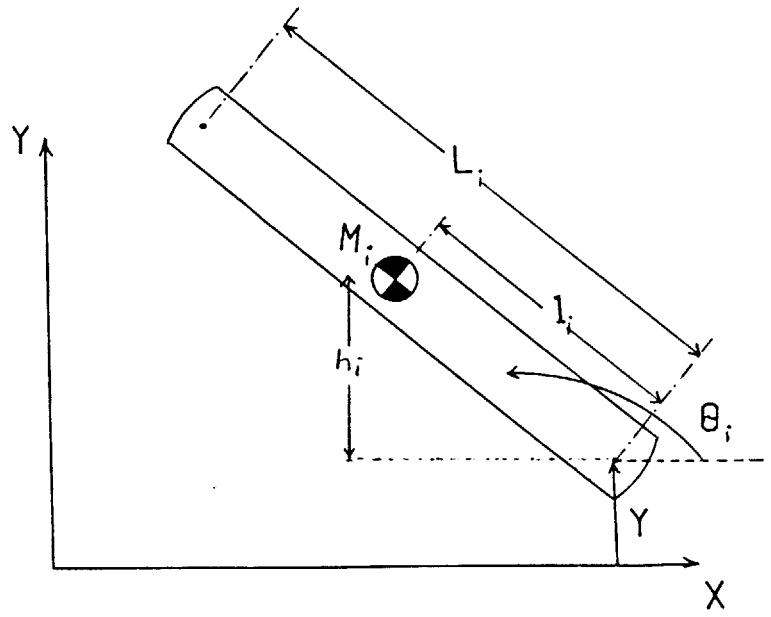

FIGURE 3-3 LINK i POTENTIAL ENERGY

GENERALIZED FORCES

The generalized forces are the nonconservative forces acting within the system, actuator torques. In general, the generalized forces are determined using equation 3.4.

$$Q_i = \sum_{i=1}^{n} F \frac{\partial X}{\partial q_i} + \sum_{i=1}^{n} M \frac{\partial \theta}{\partial q_i} \qquad (3.4)$$

where: F = forces

M = torques

q_i = generalized coordinates

X = translation

θ = rotation

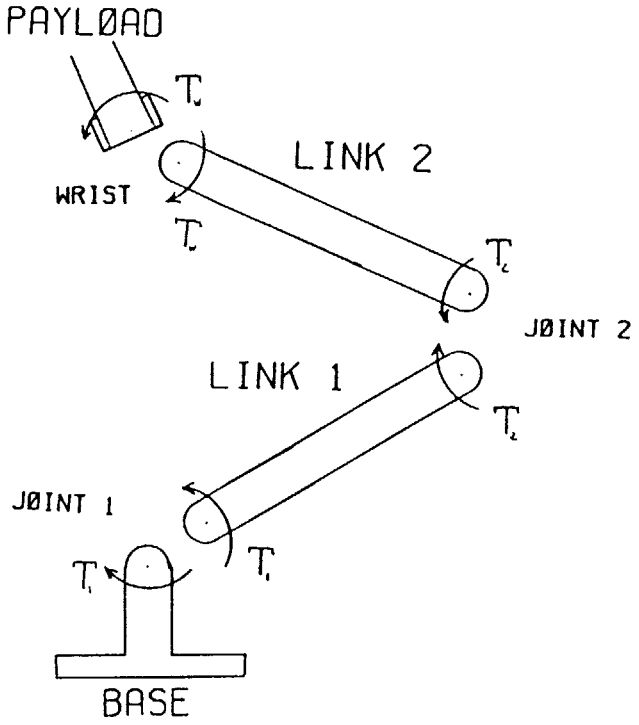

FIGURE 3-4 GENERALIZED FORCES

Equations 3.2, 3.3, and 3.4 are solved for each link and substituted into equation 3.1. The details for solving these equations are in Appendix D. The three resulting nonlinear equations are given by equations 3.5, 3.6 and 3.7. Disturbance terms are underlined.

$$(1)\ \underline{(M_1L_bl_1+M_2L_bL_1+M_pL_bL_1)\cos(\theta_b-\theta_1)\ddot{\theta}_b} +$$
$$(M_1L_1^2+M_2L_1^2+M_pL_1^2+J_1)\ddot{\theta}_1 + (M_2L_1l_2+M_pL_1L_2)\cos(\theta_1-\theta_2)\ddot{\theta}_2 +$$
$$M_pL_1l_p\cos(\theta_1-\theta_w)\ddot{\theta}_w + \underline{(M_1l_1+M_2L_1+M_pL_1)\cos\theta_1(\ddot{Y}+g)} -$$
$$\underline{(M_1L_bl_1+M_2L_bL_1+M_pL_bL_1)\sin(\theta_b-\theta_1)\dot{\theta}_b^2} +$$
$$(M_2L_1l_2+M_pL_1L_2)\sin(\theta_1-\theta_2)\dot{\theta}_2^2 + M_pL_1l_p\sin(\theta_1-\theta_w)\dot{\theta}_w^2 =$$

$$T_1 - T_2 \tag{3.5}$$

$$(2)\ \underline{(M_2 L_b l_2 + M_p L_b L_2)\cos(\theta_b - \theta_2)\ddot{\theta}_b} + (M_2 L_1 l_2 + M_p L_1 l_2)\cos(\theta_1 - \theta_2)\ddot{\theta}_1 + (M_2 l_2^2 + M_p L_2^2 + J_2)\ddot{\theta}_2 + M_p L_2 l_p \cos(\theta_2 - \theta_w)\ddot{\theta}_w + \underline{(M_2 l_2 + M_p L_2)\cos\theta_2(\ddot{Y}+g)} - \underline{(M_2 L_b L_2)\sin(\theta_b - \theta_2)\dot{\theta}_b^2} - (M_2 L_1 l_2 + M_p L_1 L_2)\sin(\theta_1 - \theta_2)\dot{\theta}_1^2 + M_p L_2 l_p \sin(\theta_2 - \theta_w)\dot{\theta}_w^2 = T_2 - T_w \tag{3.6}$$

$$(3)\ \underline{M_p L_b l_p \cos(\theta_b - \theta_w)\ddot{\theta}_b} + M_p L_1 l_p \cos(\theta_1 - \theta_w)\ddot{\theta}_1 + M_p L_2 l_p \cos(\theta_2 - \theta_w)\ddot{\theta}_2 + (M_p l_p^2 + J_p)\ddot{\theta}_w + \underline{M_p l_p \cos\theta_w(\ddot{Y}+g)} - \underline{M_p L_b l_p \sin(\theta_b - \theta_w)\dot{\theta}_b^2} - M_p L_1 l_p \sin(\theta_1 - \theta_w)\dot{\theta}_1^2 - M_p L_2 l_p \sin(\theta_2 - \theta_w)\dot{\theta}_2^2 = T_w \tag{3.7}$$

These equations are digitally solved using a FORTRAN computer program listed in Appendix E. To verify that the equations of motion correctly simulate the performance of the modeled system, various checks are conducted. Nonlinear simulations are conducted with the system placed under different initial conditions. The results of these simulations are as would be intuitively expected for a mechanical system, thus leading to increased confidence in the model. The first test, shown in figure 3-5, has the arm's initial condition horizontal, and no external forces are applied to the system. Gravity is set to zero and no motor torques are applied. The arm remained horizontal as is expected of a mechanical arm without gravity imposed.

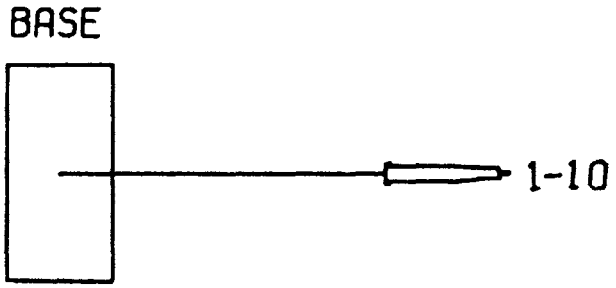

FIGURE 3-5 SIMULATION #1 NO GRAVITY

The second simulation, figure 3-6, has the arm initially horizontal, but now gravity is applied to the system. The arm swings back and forth, like a pendulum, without any loss of motion.

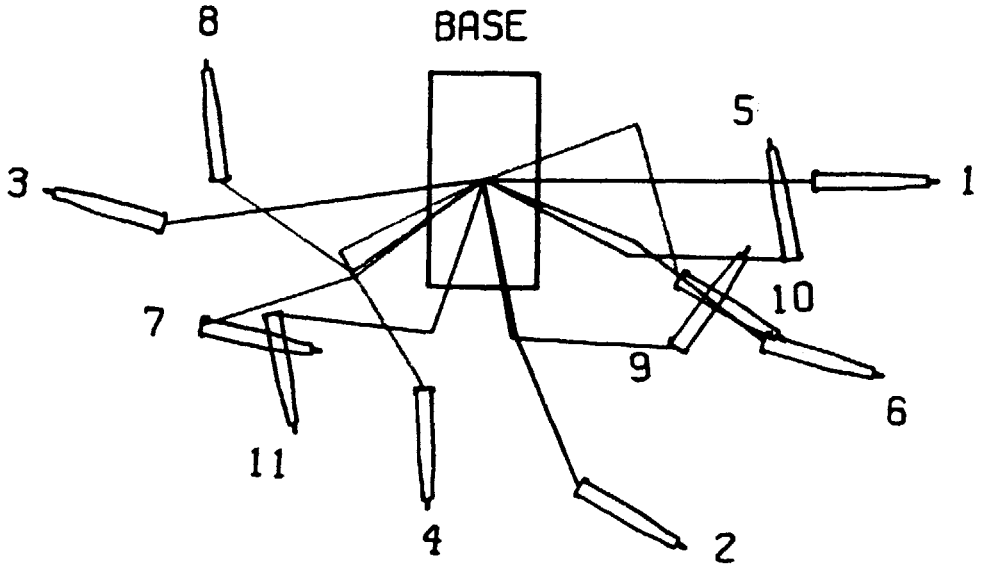

FIGURE 3-6 SIMULATION #2 WITH GRAVITY

Simulation #3, shown in figure 3-7, has the arm initially horizontal, with gravity applied, but now damping

is applied to the joints. A damping term is applied to each joint so that energy is lost from the system. The pendulum motion of the arm becomes smaller with time as the energy decreases.

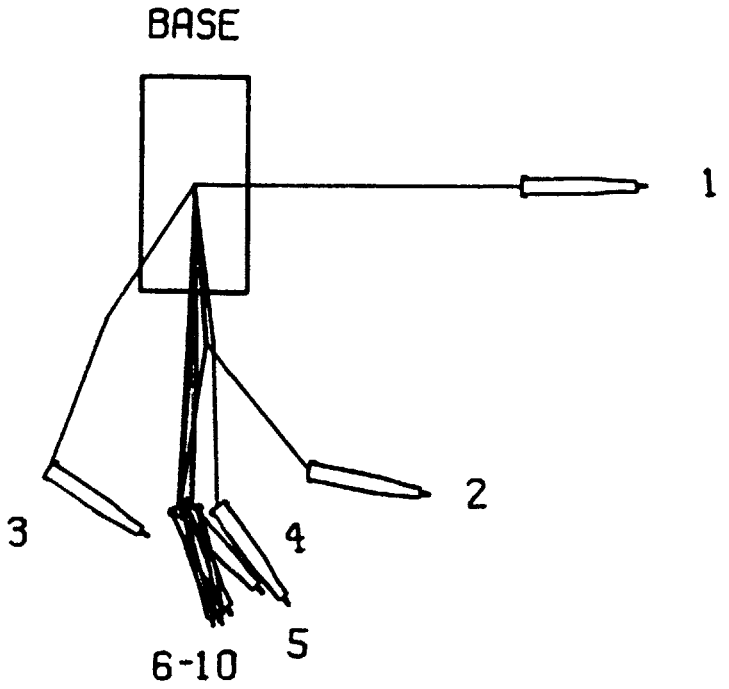

FIGURE 3-7 SIMULATION #3 WITH DAMPING

Simulation #4, shown in figure 3-8, has the arm straight down initially with a small torque applied to each joint. A constant base rotation is then applied. The base rotation causes the links to also begin rotating. This induced motion demonstrates the expected coupling which exists between base rotations and the robot. The last simulation, shown in figure 3-9, has the arm initially vertical. No torques are applied to the joints, so the arm is balanced vertically. A constant base vertical acceleration is then applied. The arm remains balanced vertically until computation roundoff errors cause the arm

to leave the vertical position slightly, then the gravity and vertical acceleration causes the arm to fall.

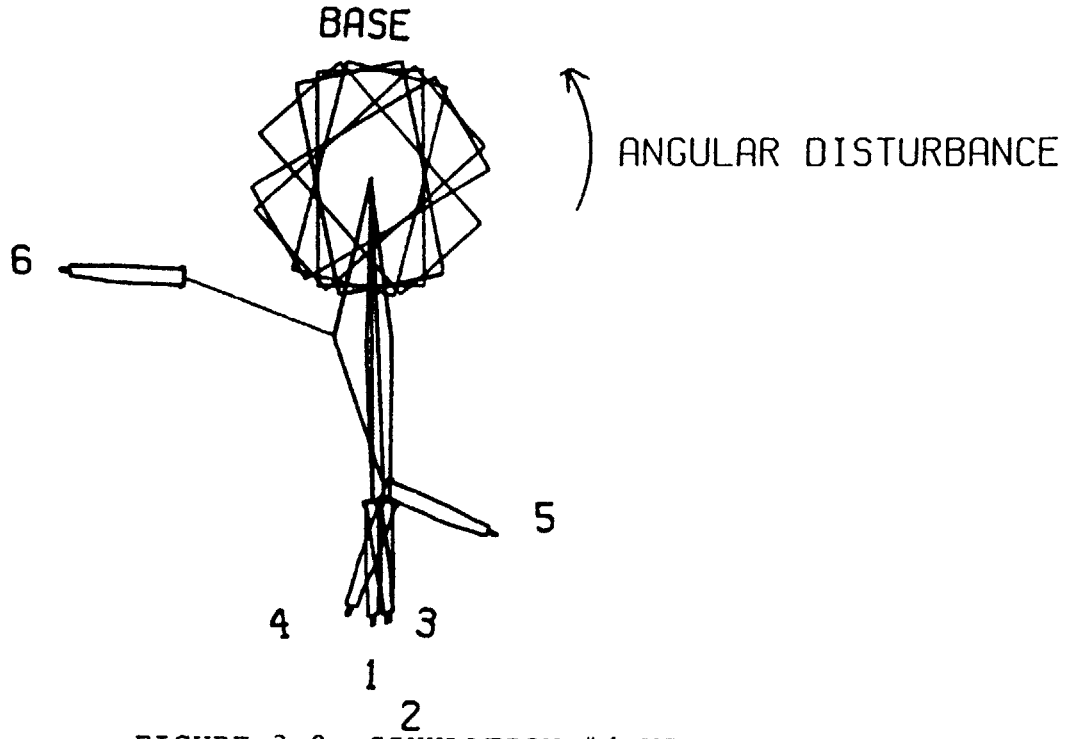

FIGURE 3-8 SIMULATION #4 WITH BASE ROTATION

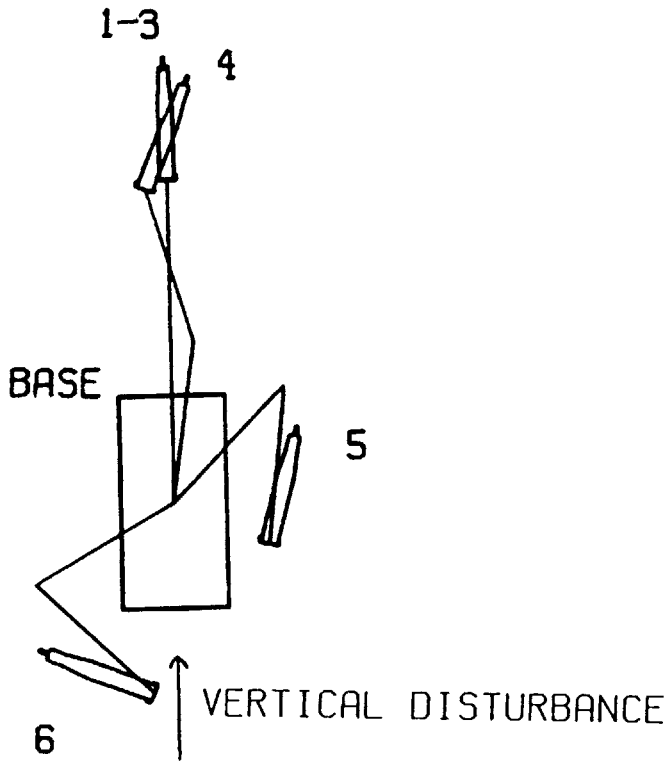

FIGURE 3-9 SIMULATION #5 WITH VERTICAL ACCELERATION

Additional checks are conducted utilizing the system's total energy, the sum of kinetic and potential energies. In the absence of any external forces, the total energy remained constant during simulations with the arm in various initial conditions, as expected.

3.3 LINEARIZATION

The resulting equations of motion are highly nonlinear, (example: $M_1 L_b l_1 \sin(\theta_2)\dot{\theta}_b\dot{\theta}_2$). In order to design a linear controller, the nonlinear equations are linearized about selected nominal operating points. This linear controller will then be used to conduct simulations over the entire range of operating points. The nonlinear system will then be at least assympotically stable for small deviations from the selected nominal operating point. The linearization is completed through a Taylor series expansion, with higher order terms truncated [13,15]. Small perturbations about the nominal operating points are obtained as follows:

$$\theta_i = \theta_i^{\,n} + \partial\theta_i \qquad i=1,2,w \tag{3.8}$$

$$\dot{\theta}_i = \dot{\theta}_i^{\,n} + \partial\dot{\theta}_i \tag{3.9}$$

$$\ddot{\theta}_i = \ddot{\theta}_i^{\,n} + \partial\ddot{\theta}_i \tag{3.10}$$

where:

$\theta_i^{\,n}$, $\dot{\theta}_i^{\,n}$, $\ddot{\theta}_i^{\,n}$ are the selected nominal operating points,

$\partial\theta_i$, $\partial\dot{\theta}_i$, $\partial\ddot{\theta}_i$ are small perturbations about

the nominal points.

Details for linearizing the equations of motion are given in Appendix D. The resulting three linearized equations of motion are given by equations 3.11, 3.12 and 3.13. Disturbance terms are underlined.

$$
\begin{aligned}
(1)\ & (M_1 l_1^2+M_2 L_1^2+M_p L_1^2+J_1)\partial\ddot{\theta}_1 + \\
& [(M_2 L_1 l_2+M_p L_1 L_2)\cos(\theta_1^n-\theta_2^n)]\partial\ddot{\theta}_2 + \\
& [M_p L_1 l_p \cos(\theta_1^n-\theta_w^n)]\partial\ddot{\theta}_w = \\
& \{\underline{-(M_1 L_b l_1+M_2 L_b L_1+M_p L_b L_1)[\sin(\theta_b-\theta_1^n)\ddot{\theta}_b + \cos(\theta_b-\theta_1^n)\dot{\theta}_b^2]} \\
& + (M_2 L_1 l_2+M_p L_1 L_2)[\sin(\theta_1^n-\theta_2 n)\ddot{\theta}_2^n - \cos(\theta_1^n-\theta_2^n)(\dot{\theta}_2^n)^2] \\
& + M_p L_1 l_p[\sin(\theta_1^n-\theta_w^n)\ddot{\theta}_w^n - \cos(\theta_1^n-\theta_w^n)(\dot{\theta}_w^n)^2] + \\
& \underline{(M_1 l_1+M_2 L_1+M_p L_1)\sin\theta_1^n(\ddot{Y}+g)}\}\partial\theta_1 + \\
& \{(M_2 L_1 l_2+M_p L_1 L_2)[\cos(\theta_1^n-\theta_2^n)(\dot{\theta}_2^n)^2 - \\
& \sin(\theta_1^n-\theta_2^n)\ddot{\theta}_2^n]\}\partial\theta_2 - \\
& \{2(M_2 L_1 l_2+M_p L_1 L_2)[\sin(\theta_1^n-\theta_2^n)\dot{\theta}_2^n]\}\partial\dot{\theta}_2 + \\
& \{M_p L_1 l_p[\cos(\theta_1^n-\theta_w^n)(\dot{\theta}_w^n)^2 - \sin(\theta_1^n-\theta_w^n)\ddot{\theta}_w^n]\}\partial\theta_w - \\
& [2M_p L_1 l_p \sin(\theta_1^n-\theta_w^n)\dot{\theta}_w^n]\partial\dot{\theta}_w + T_1 - T_2 \qquad (3.11)
\end{aligned}
$$

$$
\begin{aligned}
(2)\ & [(M_2 L_1 l_2+M_p L_1 L_2)\cos(\theta_1^n-\theta_2^n)]\partial\ddot{\theta}_1 + (M_2 l_2^2+M_p L_2^2+J_2)\partial\ddot{\theta}_2 + \\
& [M_p L_2 l_p \cos(\theta_2^n-\theta_w^n)]\partial\ddot{\theta}_w = \\
& \{(M_2 L_1 l_2+M_p L_1 L_2)[\sin(\theta_1^n-\theta_2^n)\ddot{\theta}_1^n + \\
& \cos(\theta_1^n-\theta_2^n)(\dot{\theta}_1^n)^2]\}\partial\theta_1 + \\
& 2(M_2 L_1 l_2+M_p L_1 L_2)[\sin(\theta_1^n-\theta_2^n)\dot{\theta}_1^n]\partial\dot{\theta}_1 + \\
& \{M_p L_2 l_p[\sin(\theta_2-\theta_w^n)\ddot{\theta}_w^n - \cos(\theta_2^n-\theta_w^n)(\dot{\theta}_w^n)^2] + \\
& \underline{(M_2 l_2+M_p L_2)\sin\theta_2^n(\ddot{Y}+g)} -
\end{aligned}
$$

$$(M_2L_bl_2+M_pL_bL_2)[\sin(\theta_b-\theta_2^n)\ddot{\theta}_b + \cos(\theta_b-\theta_2^n)\dot{\theta}_b^2] -$$
$$(M_2L_1l_2+M_pL_1L_2)[\sin(\theta_1^n-\theta_2^n)\ddot{\theta}_1^n +$$
$$\cos(\theta_1^n-\theta_2^n)(\dot{\theta}_1^n)^2]\}\partial\theta^2 + M_pL_2l_p[\cos(\theta_2^n-\theta_w^n)(\dot{\theta}_w^n)^2 -$$
$$\sin(\theta_2^n-\theta_w^n)\ddot{\theta}_w^n]\partial\theta_w - [2M_pL_2l_p\sin(\theta_2^n-\theta_w^n)\dot{\theta}_w^n]\partial\dot{\theta}_w +$$
$$T_2 - T_w \qquad (3.12)$$

(3) $$[M_pL_1l_p\cos(\theta_1^n-\theta_w^n)]\partial\ddot{\theta}_1 + [M_pL_2l_p\cos(\theta_2^n-\theta_w^n)]\partial\ddot{\theta}_2 +$$
$$(M_pl_p^2+J_p)\partial\ddot{\theta}_w = \{M_pL_1l_p[\sin(\theta_1^n-\theta_w^n)\ddot{\theta}_1^n +$$
$$\cos(\theta_1^n-\theta_w^n)(\dot{\theta}_1^n)^2]\partial\theta_1 + [2M_pL_1l_p\sin(\theta_1^n-\theta_w^n)\dot{\theta}_1^n]\partial\dot{\theta}_1 +$$
$$\{M_pL_2l_p[\sin(\theta_2^n-\theta_w^n)\ddot{\theta}_2^n + \cos(\theta_2^n-\theta_w^n)(\dot{\theta}_2^n)^2]\}\partial\theta_2 +$$
$$[2M_pL_2l_p\sin(\theta_2^n-\theta_w^n)\dot{\theta}_2^n]\partial\dot{\theta}_2 + \{M_pl_p\sin\theta_w^n(\ddot{Y}+g) -$$
$$M_pL_bl_p[\sin(\theta_b-\theta_w^n)\ddot{\theta}_b + \cos(\theta_b-\theta_w^n)\dot{\theta}_b^2] -$$
$$M_pL_1l_p[\sin(\theta_1^n-\theta_w^n)\ddot{\theta}_1^n + \cos(\theta_1^n-\theta_w^n)(\dot{\theta}_1^n)^2] -$$
$$M_pL_1l_p[\sin(\theta_2^n-\theta_w^n)\ddot{\theta}_2^n + \cos(\theta_2^n-\theta_w^n)(\dot{\theta}_2^n)^2]\}\partial\theta_w + T_w$$
(3.13)

CHAPTER 4

CONTROLLER DESIGN

4.1 INTRODUCTION

A controller for a moving base robot must compensate for base disturbances and meet performance specifications, even during large base motions. Previous research by Rick Lynch demonstrated that conventional and even advanced control strategies had difficulty compensating for the vertical base accelerations of a mobile manipulator when only joint feedback signals were used [3]. The approach in this research is to design a controller using joint feedback signals which meets specifications without disturbances, test Rick Lynch's findings that this controller is inadequate to compensate for base disturbances, and then as shown to be required, design a feed forward compensator which uses measured base motion signals to successfully compensate for the base disturbances.

4.2 DECOUPLED CONTROLLER

Industrial robots utilize controllers that are relatively simple, reliable, and easy to implement. This is accomplished by utilizing local, decoupled, proportional-derivative (PD) or proportional-integral-derivative (PID) controllers at each joint. Integral control is often included to eliminate steady state error from constant disturbances, but can have a destabilizing effect on a

system. These controllers are typically represented by equation 4.1 for PD and equation 4.2 for PID.

$$T_i = -Kp_i(\theta_i - \theta_i^r) - Kd_i(\dot{\theta}_i) \qquad (4.1)$$

$$T_i = -Kp_i(\theta_i - \theta_i^r) - Kd_i(\dot{\theta}_i) - Ki_i\int(\theta_i - \theta_i^r)dt \qquad (4.2)$$

where: K_p = position gain

K_d = velocity gain

K_i = integral gain

θ^r = desired joint angle

T = generated actuator torque

θ = actual joint angle

$\dot{\theta}$ = actual joint angle velocity

These controllers ignore the coupling effects between joints and the nonlinear dynamics of the system. PID controllers accomplish position control, but can be sensitive to disturbances. Some advanced control strategies are more robust and less susceptible to parameteric uncertainty, but are currently still unsuccessful in compensating for base disturbances as demonstrated by Rick Lynch's work. A PID controller was chosen for this research since PID controllers are some what susceptible to base disturbances. A compensator designed for a PID controller, which can successfully compensate for base motion disturbances will also be transferable to other control

strategies. Therefore, this research is applicable to any control strategy.

The procedure used to design a controller for the nonlinear system, is to design a linear controller, using linearized equations. Linearized equations are only valid for small perturbations from the selected operating points. For the robotic loader system, the baseline load and return cycle causes all terms in the linearized equations to substanially change. Therefore, average values over the load and return cycle are used as the nominal values in the linearized equation, and all disturbance terms are set to zero. The designed linear controller's performance is then checked over the full range of operating values. The robot's configuration corresponding to the selected nominal values is depicted in figure 4-1.

where:

$\theta_1^n = .3541$	$\theta_2^n = 1.8473$	$\theta_w^n = 1.3811$	rad
$\dot{\theta}_1^n = -.1076$	$\dot{\theta}_2^n = .0499$	$\dot{\theta}_w^n = -.0317$	rad/sec
$\ddot{\theta}_1^n = .8036$	$\ddot{\theta}_2^n = .7914$	$\ddot{\theta}_w^n = -.0179$	rad/sec^2

$\theta_b = 0.0$ rad

$\dot{\theta}_b = 0.0$ rad/sec

$\ddot{\theta}_b = 0.0$ rad/sec^2

$\ddot{Y} = 0.0$ feet/sec^2

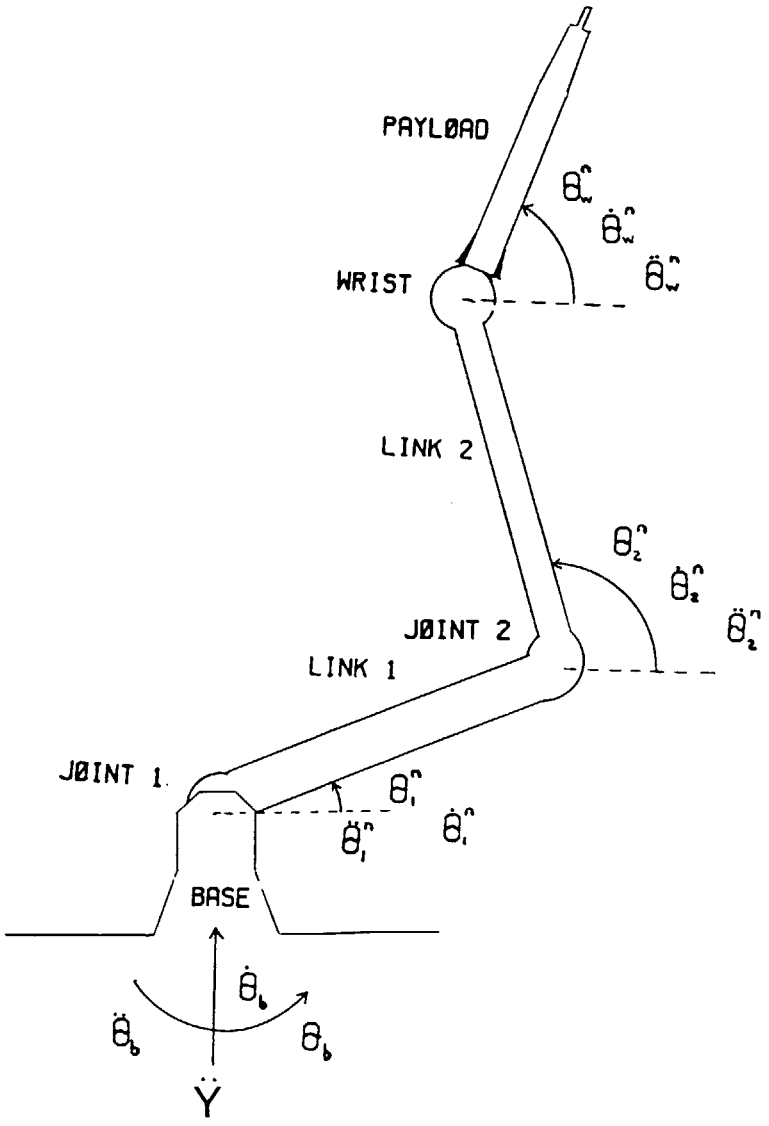

FIGURE 4-1 NOMINAL VALUES

These values, when substituted into the linearized equations, equations 3.11, 3.12 and 3.13, result in coefficients which can be expressed in standard state space matrix form, equations 4.3 and 4.4. The robotic loader system has three degrees of freedom and is second order,

therefore the linear system coefficient matrix, A, is (6x6).

$$X = AX + BU + LW \tag{4.3}$$

$$Y = CX \tag{4.4}$$

where: X is state variable vector

Y is output vector

U is input vector

A is coefficient matrix

B is input control matrix

C is output control matrix

L is disturbance matrix

W is disturbance input vector

The linearized equations can then be expressed in standard matrix form, equations 4.5 and 4.6, for the selected nominal values. A computer program, listed in Appendix E, computes these coefficients.

$$\begin{bmatrix} \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \\ \dot{\theta}_w \\ \ddot{\theta}_w \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -.3560 & -.0069 & .3762 & .0455 & .2916 & -.0028 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -1.0573 & .2794 & 1.1585 & -.0121 & 1.5460 & .0048 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ .8888 & -.1246 & -1.7529 & .0646 & -13.0232 & -.0057 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \\ \theta_w \\ \dot{\theta}_w \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .0090 & 0 & -.0086 & 0 & -.0103 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .0004 & 0 & .0287 & 0 & -.0817 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -.0099 & 0 & -.0427 & 0 & .4961 \end{bmatrix} \begin{bmatrix} 0 \\ T_1 \\ 0 \\ T_2 \\ 0 \\ T_w \end{bmatrix} \quad (4.5)$$

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \\ \theta_3 \\ \dot{\theta}_3 \end{bmatrix} \quad (4.6)$$

The open-loop linearized system, without feedback, is unstable. This is evidenced by the location of open-loop poles in the right half plane. The open-loop poles are:

$$\lambda = +0.8284$$

$$\lambda = +0.0198$$

$$\lambda = -0.005 + 3.5839j$$

$$\lambda = -0.005 - 3.5839j$$

$$\lambda = -0.1037$$

$$\lambda = -0.7591$$

The open-loop unstability of the linearized system, as evidenced by the positive poles, is due to the effect of gravity [14]. Without a controller, the robotic system will fall from the linearized position. To make the system stable, and to achieve the desired system response, state feedback is used. Full state feedback is described by equation 4.7.

$$U = r - KX \tag{4.7}$$

where: U is the input vector, torques

r is the command, desired joint angle positions

K is the controller gains

X is the state vector, joint angles and velocity

A PID controller with full state feedback is shown by the block diagram in figure 4-2.

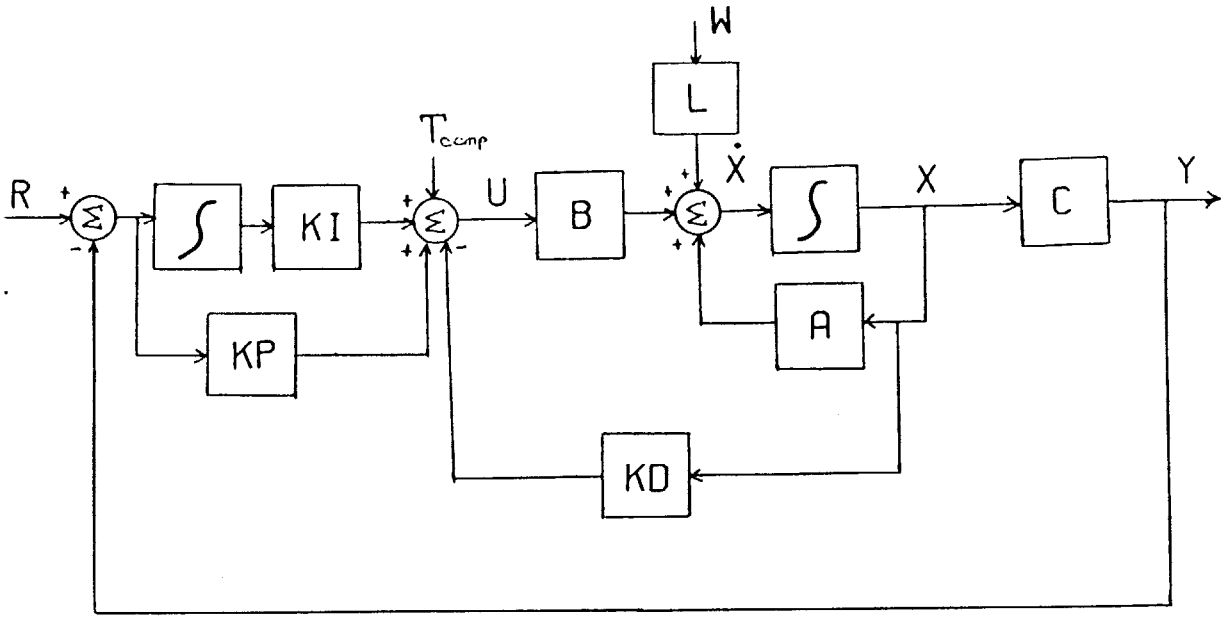

FIGURE 4-2 PID CONTROLLER BLOCK DIAGRAM

In standard matrix form, integral feedback is obtained by forming an augmented state vector, X_a, shown by equation 4.8 [15].

$$X_a = \begin{bmatrix} X \\ --- \\ a \end{bmatrix} \tag{4.8}$$

where: a is the intregral state vector

$$= \int (r - Y)dt$$

$$= \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_w \end{bmatrix}$$

Controller gains are determined by the selection of the eigenvalues (closed-loop poles). These controller eigenvalues are bounded by the mechanism's structural resonance frequency. Good design practice has the closed-loop poles at frequencies less than one third of the

lowest structural resonances, which is approximated in Appendix B to be 15 hertz or 94.25 rad/sec. Therefore, the imaginary component of the complex poles should be less than approximately one third of the resonance frequency or 30 rad/sec. Additionally, the controller response should be fast, and have minimal overshoot, so the damping coefficient is chosen to be 0.70. Hence, the desired dominate closed-loop poles should be in the vicinity of -20 - 20j and -20 + 20j and the other poles on the real axis around -50. Using a procedure described in reference 14, the coupling terms are ignored and the controller gains are obtained based on the selected dominate closed-loop poles. This results in the gain matrix, K, equation 4.9.

$$K = \begin{bmatrix} 68867 & 6886 & 0 & 0 & 0 & 0 & 826404 & 0 & 0 \\ 0 & 0 & 26000 & 2606 & 0 & 0 & 0 & 312000 & 0 \\ 0 & 0 & 0 & 0 & 1771 & 117 & 0 & 0 & 2152 \end{bmatrix} \tag{4.9}$$

where: $K = [K_t \quad K_i]$

$Kt = [Kp_1 \; Kd_1 \; Kp_2 \; Kd_2 \; Kp_3 \; Kd_3]$

Applying these gains, equation 4.9, to a simulation of the full non-linear system, without disturbances, demonstrates that the performance of PID controller fails to meet all of the performance specifications described in Chapter 2. This simulation, as with all simulations in this thesis, utilize a time increment of 0.001 seconds and a

fourth order Runga Kutta integration routine. The robotic loader system fails to successfully insert the ammunition into the main gun chamber without hitting the breach. Additionally, the end effector does not line up with the new ammunition round at the end of the return cycle. This performance is described in the following figures.

Figure 4-3 shows the motion of the manipulator with respect to the tank turret. A robot figure is plotted approximately every one third of a second during the robot motion. This motion, figure 4-3, shows the robotic loader extract an ammunition round from the storage rack and attempt to load the round into the main gun chamber. The robotic loader then returns to the ammunition storage area to grasp a new round. Figure 4-4 shows the actuator torques necessary for the described motion in figure 4-3. These required actuator torques are below the saturation levels, as specified in Chapter 2.

Figure 4-5 shows the comparison between the desired joint angles, as specified by the baseline trajectory, and the actual system joint angles. A better indication of the achieved accuracy is figure 4-6 which shows the joint angle errors, the differences between the desired and actual joint angles. As explained in Chapter 2, the specifications which determines the successful loading of ammunition, is the ability of the robotic loader to insert the ammunition payload through the breach into the chamber. The success of this operation is then indicated by whether the payload

joint angle error and payload vertical tip error are within the allowed tolerances during insertion. Figures 4-7 and 4-8 show that the errors are greater than allowed, demonstrating the failure of the PID controller.

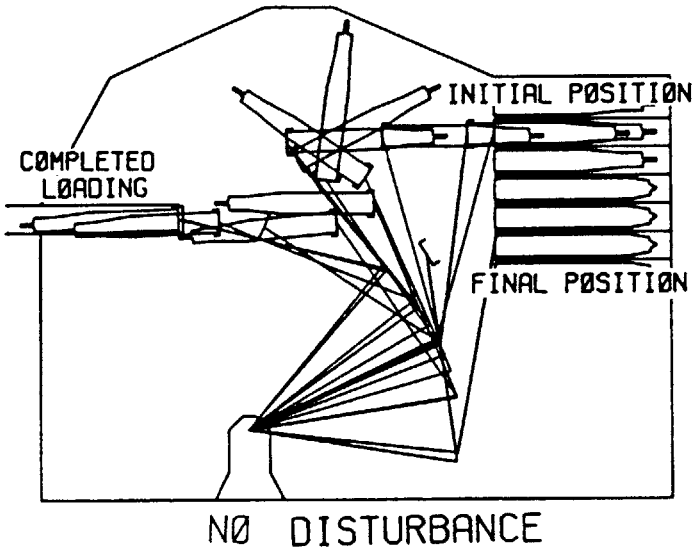

FIGURE 4-3 PID CONTROLLED ROBOT MOTION

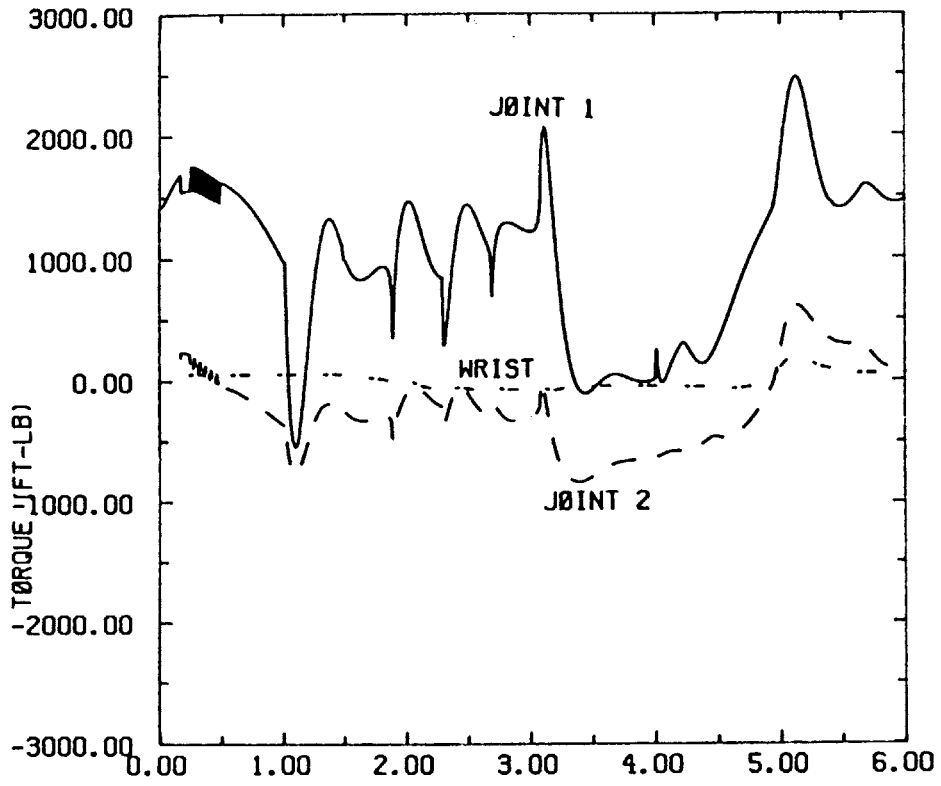

FIGURE 4-4 PID CONTROLLED MOTOR TORQUES

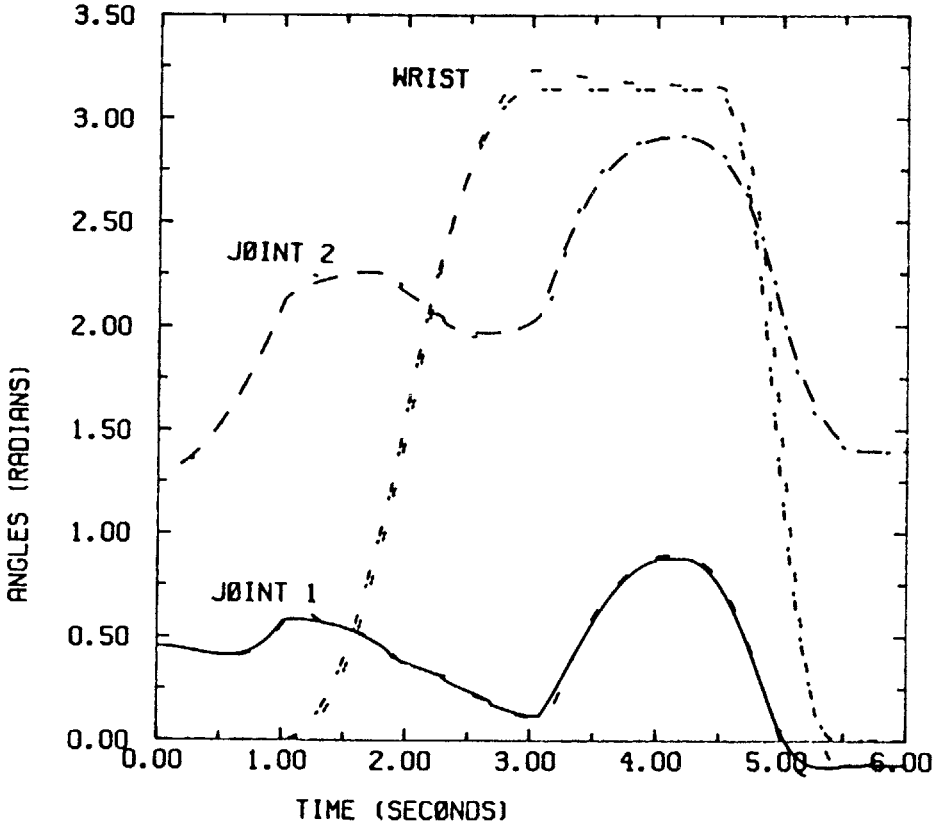

FIGURE 4-5 PID CONTROLLED JOINT ANGLE COMPARISON

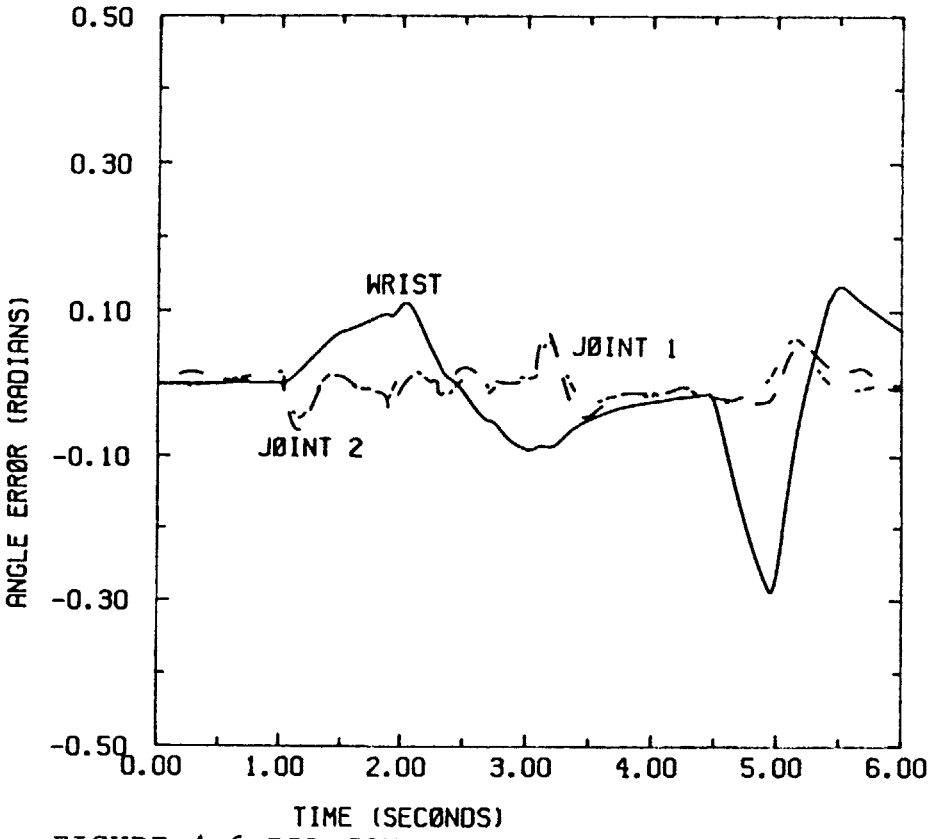

FIGURE 4-6 PID CONTROLLED JOINT ANGLE ERRORS

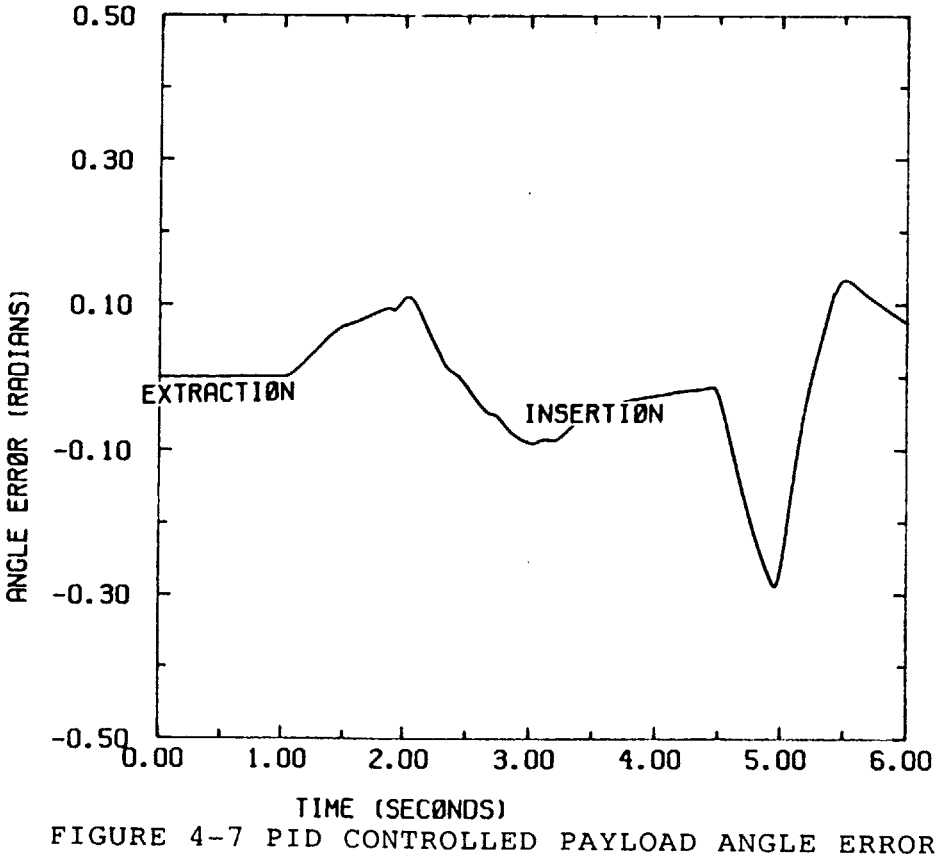

FIGURE 4-7 PID CONTROLLED PAYLOAD ANGLE ERROR

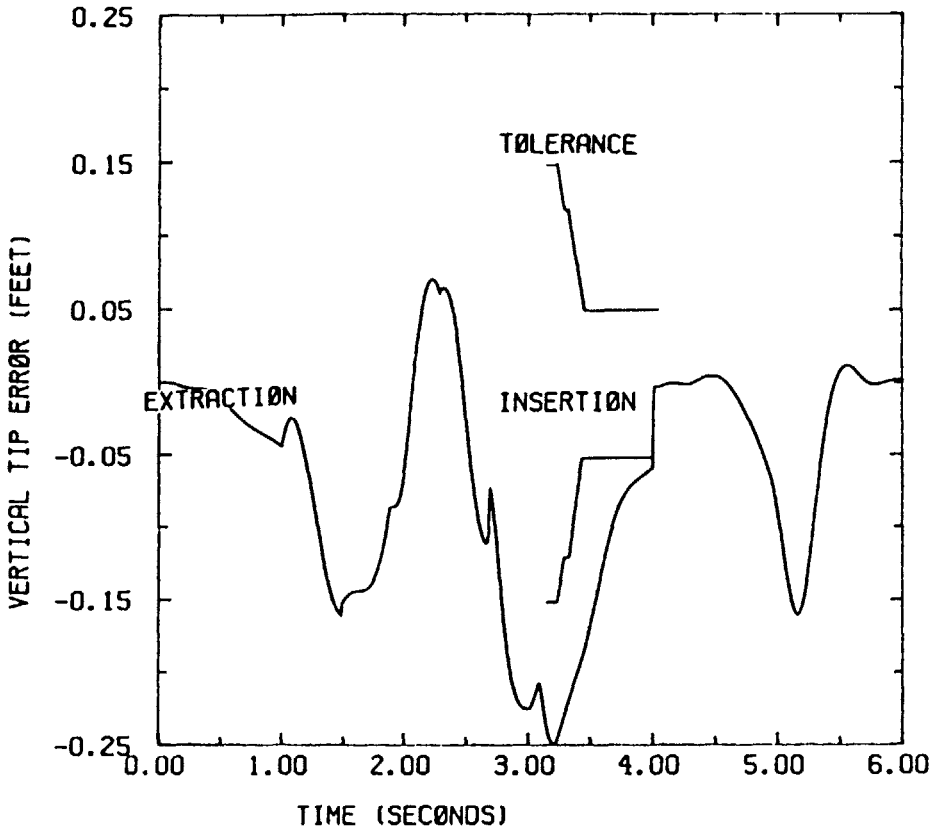

FIGURE 4-8 PID CONTROLLED PAYLOAD VERTICAL TIP ERROR

Simulations eliminating the integral controller element resulted in improved performance. This demonstrated that the integral controller element is not necessary for good performance in this system. Adjusting the gains, equation 4.9, to achieve the best PD linear controller performance resulted in the controller gains, K, equation 4.10.

$$K = \begin{bmatrix} 60000 & 6000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20000 & 2000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2000 & 100 \end{bmatrix} \qquad (4.10)$$

This linear PD system, with full state feedback, is described by the block diagram, shown in figure 4-9, which includes the base moton disturbance terms.

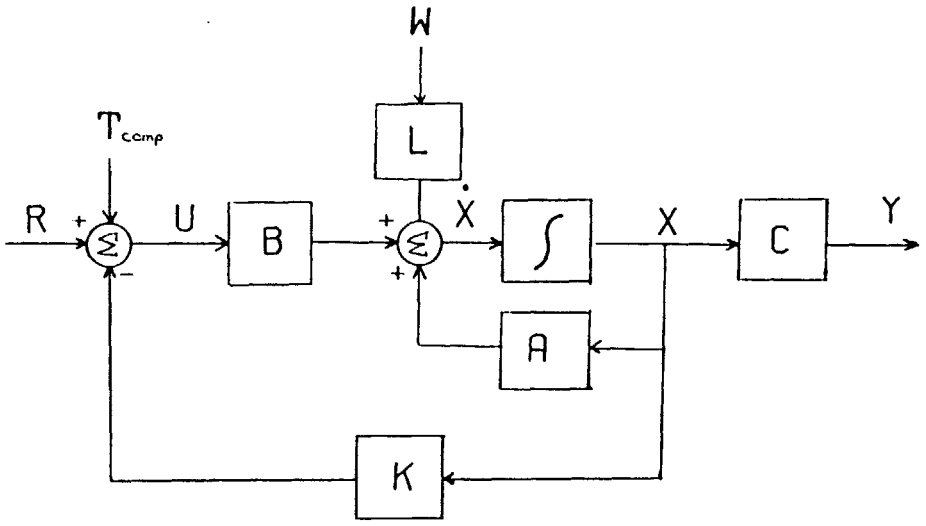

FIGURE 4-9 PD CONTROLLER BLOCK DIAGRAM

The performance of the PD uncoupled linear controller is tested against the performance specifications described in Chapter 2. Simulations of the full non-linear system are conducted using the computer program listed in Appendix E, with the calculated constant controller gains, K, equation 4.10. These simulations show that the performance of the linear PD controller over the entire range of operating values required by the baseline trajectory, Chapter 2, and without any base motion disturbances, meets all performance specifications. This performance is described in the following figures.

Figure 4-10 shows the motion of the robotic loader during the six second loading cycle. The robotic loader

extracts a modeled HEAT ammunition round from the ammunition 'ready rack' storage area, swings the ammunition around 180 degrees so that the round is oriented towards the main gun breach, and then inserts the round through the breach into the chamber. The loader then returns, flipping the wrist back to the initial orientation, to the ammunition storage area to grasp a new round. Figure 4-11 shows the torques required to generate the motion described in figure 4-10. These actuator torques, determined by equation 4.1, are below the saturation levels. Additionally, each joint retains over one third of the actuator torque potential for base motion disturbance rejection.

Figure 4-12 shows the comparison between the desired joint angles, as specified by the baseline trajectory, and the achieved actual system joint angles. This figure provides an indication of the system's ability to follow the prescribed path, and the obtained accuracy. The accuracy appears to be good, but a better indicator of joint angle accuracy is determined from the joint angle errors. The joint angle errors are shown in figure 4-13. The error is measured in radians, and the greatest joint is less than 0.15 radians, or 8.5 degrees. As explained in Chapter 2, the specifications which determine the loader's successful loading of ammunition, is the ability to insert the round, remaining within the allowed tolerances. Figures 4-14 and 4-15 show that the payload vertical tip error is within the allowable tolerance. The tolerance limit is shown on the

graph. Figure 4-16 shows the closed-loop system eigenvalues. These poles change as the robot's configuration changes. The poles are within the limits required by the structural frequencies.

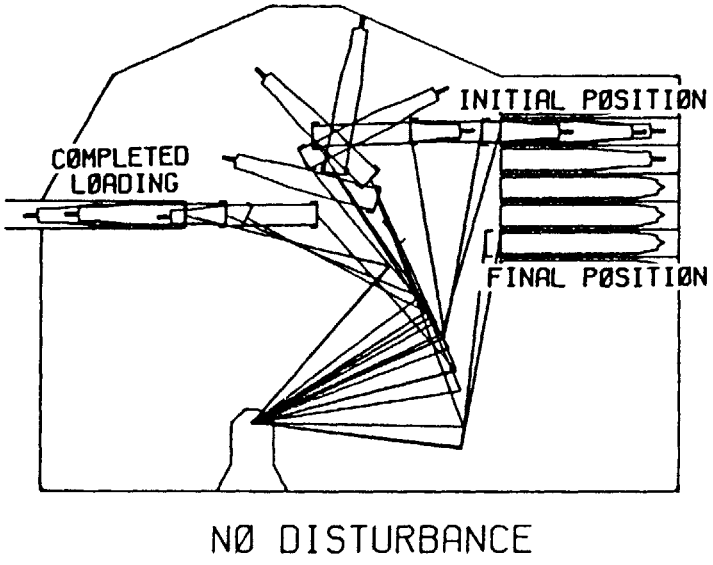

FIGURE 4-10 PD CONTROLLED ROBOT MOTION

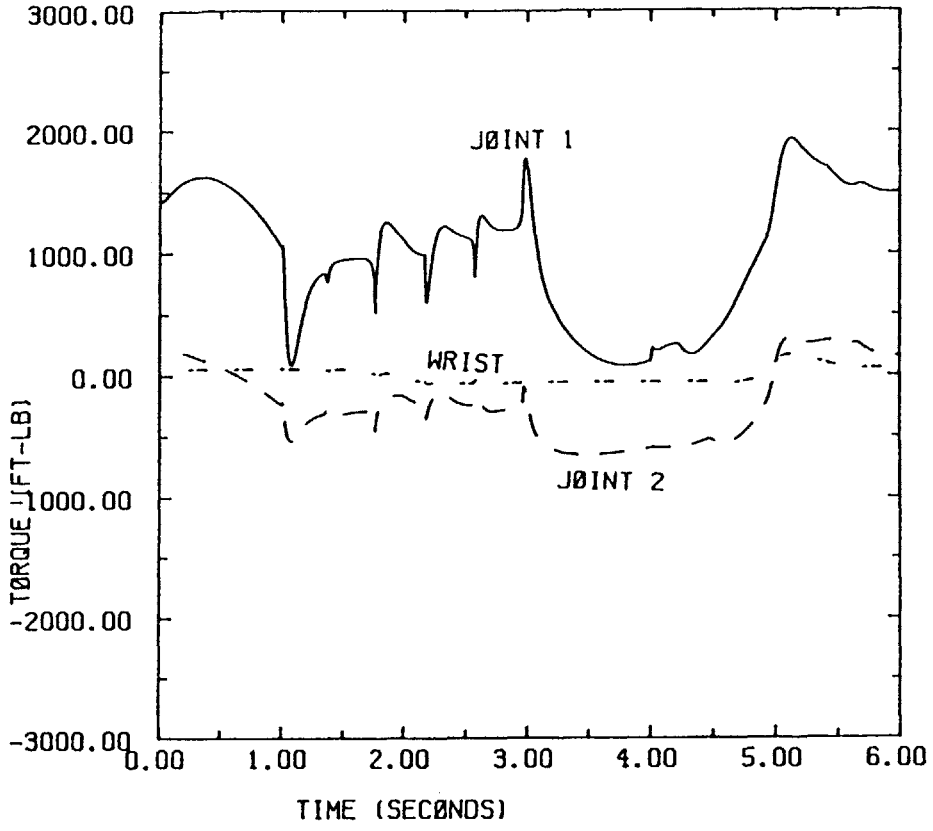

FIGURE 4-11 PD CONTROLLED MOTOR TORQUES

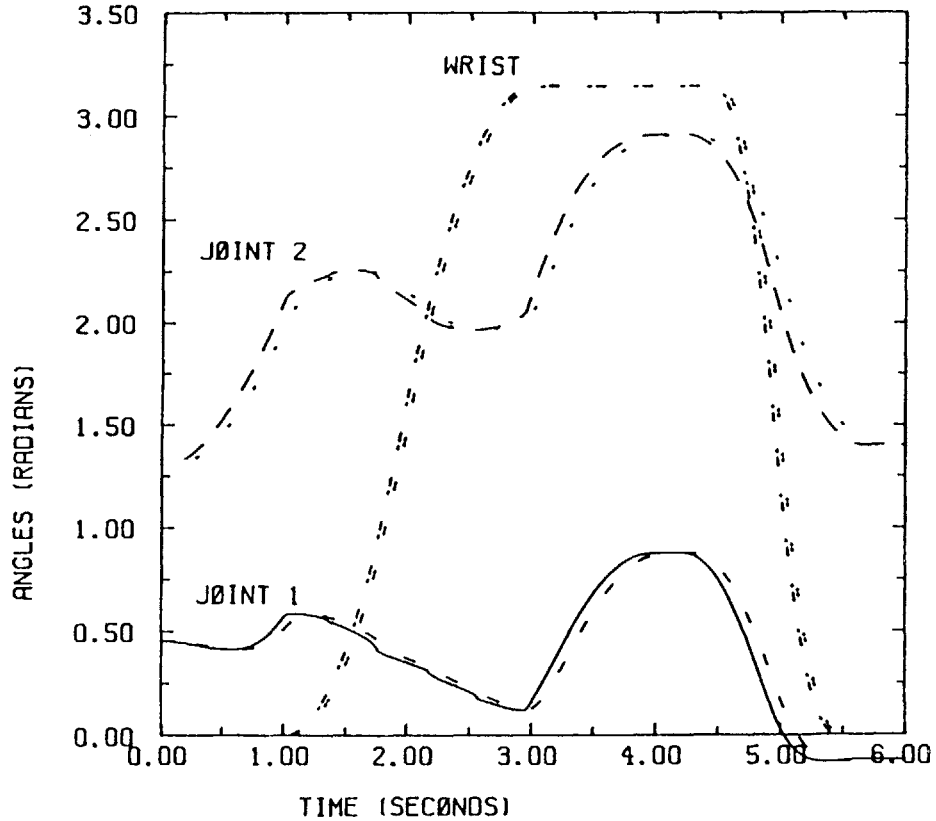

FIGURE 4-12 PD CONTROLLED JOINT ANGLE COMPARISON

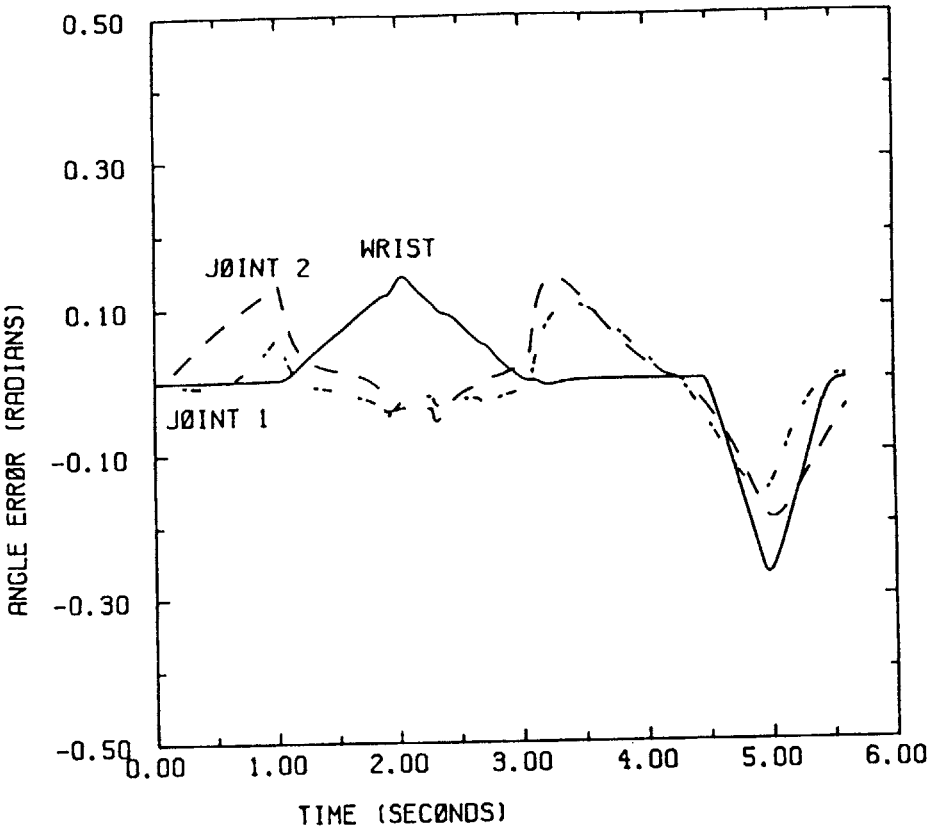

FIGURE 4-13 PD CONTROLLED JOINT ANGLE ERRORS

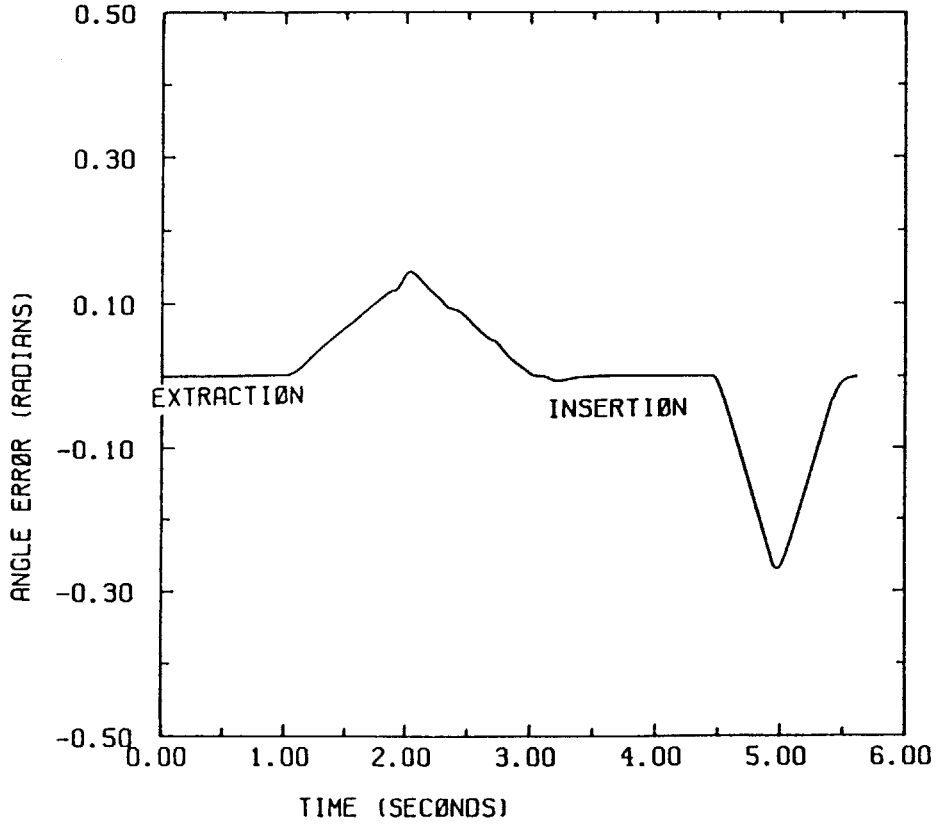

FIGURE 4-14 PD CONTROLLED PAYLOAD ANGLE ERROR

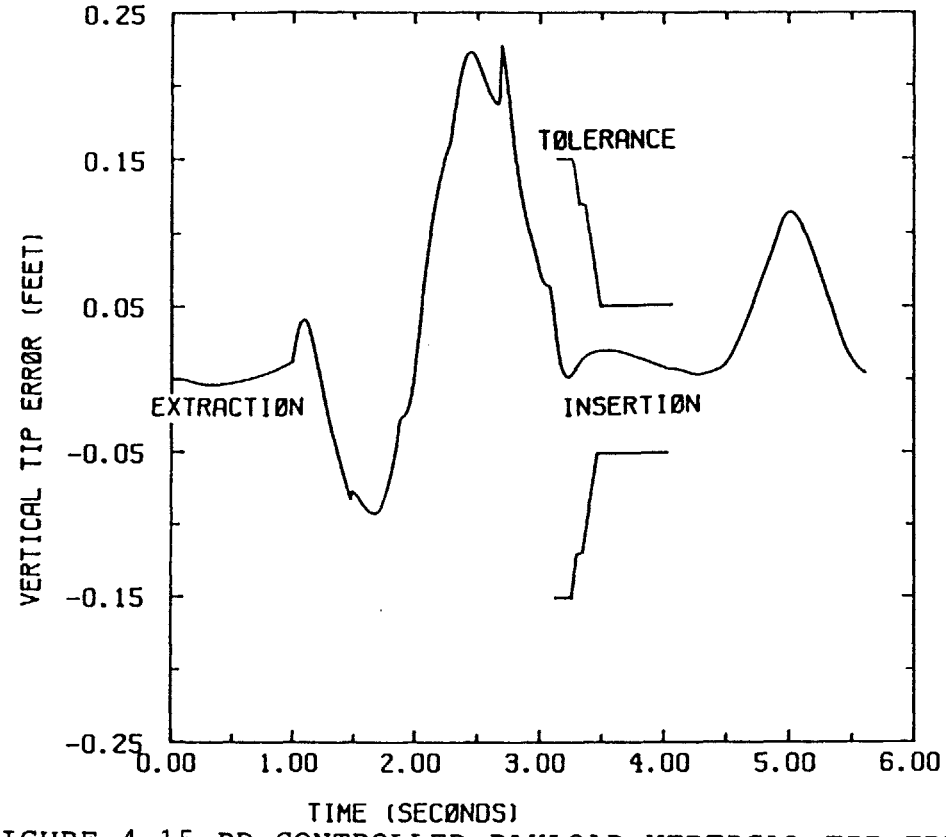

FIGURE 4-15 PD CONTROLLED PAYLOAD VERTICAL TIP ERROR

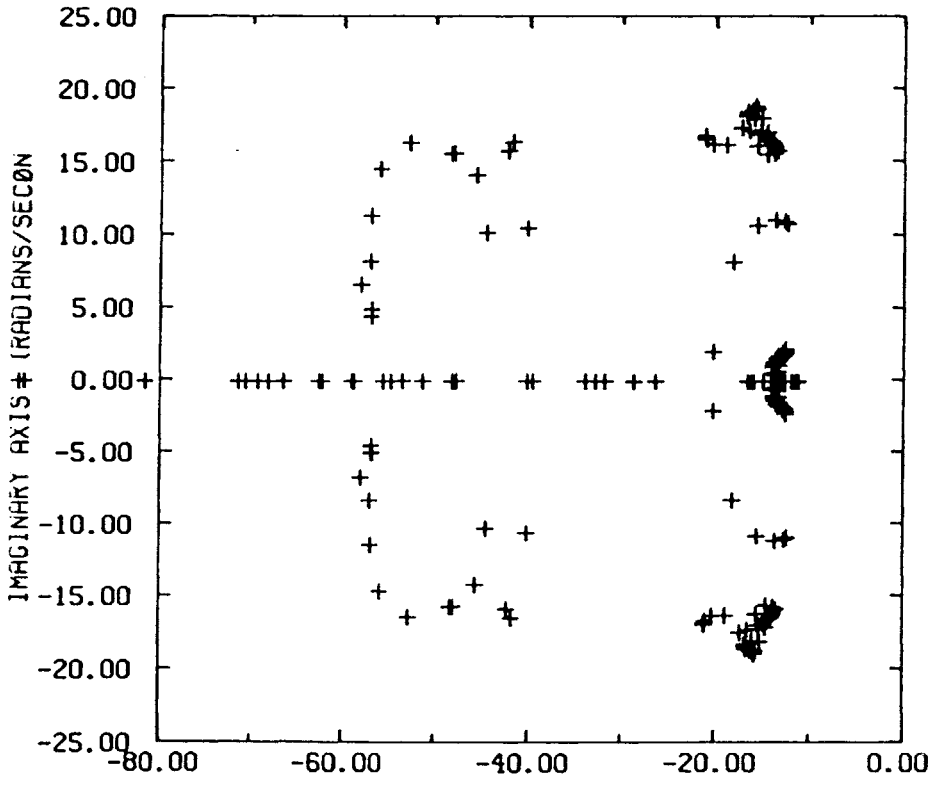

FIGURE 4-16 PD CONTROLLER CLOSED-LOOP POLES

The robotic loader needs to be capable of handling the various ammunition payloads. The robotic loader system uses a 50 pound average weight payload model. This robustness is tested through simulations by varying the payloads. Payloads of sixty and forty are used. The results of these simulations show negligible effect. The payload angle error and payload vertical tip error are shown for each payload. Figures 4-17 and 4-18 are the results of simulations using the heavest ammunition, sixty pounds. Figures 4-19 and 4-20 are the results for the forty pound payload.

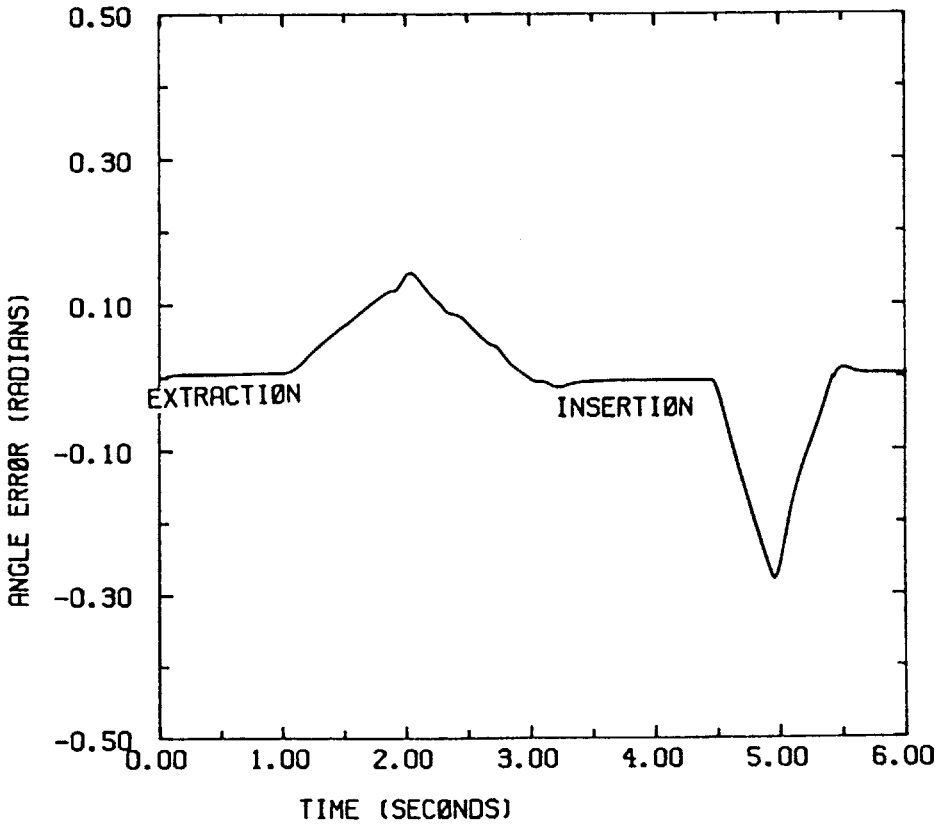

FIGURE 4-17 60 LB PAYLOAD ANGLE ERROR

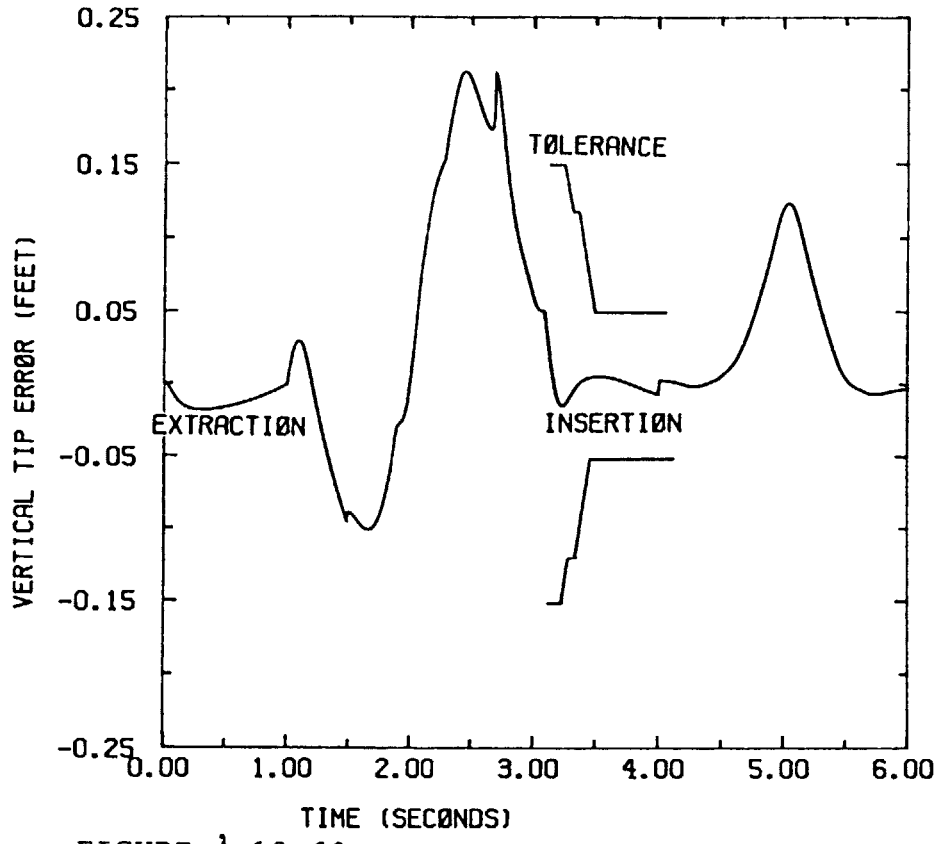

FIGURE 4-18 60 LB PAYLOAD VERTICAL TIP ERROR

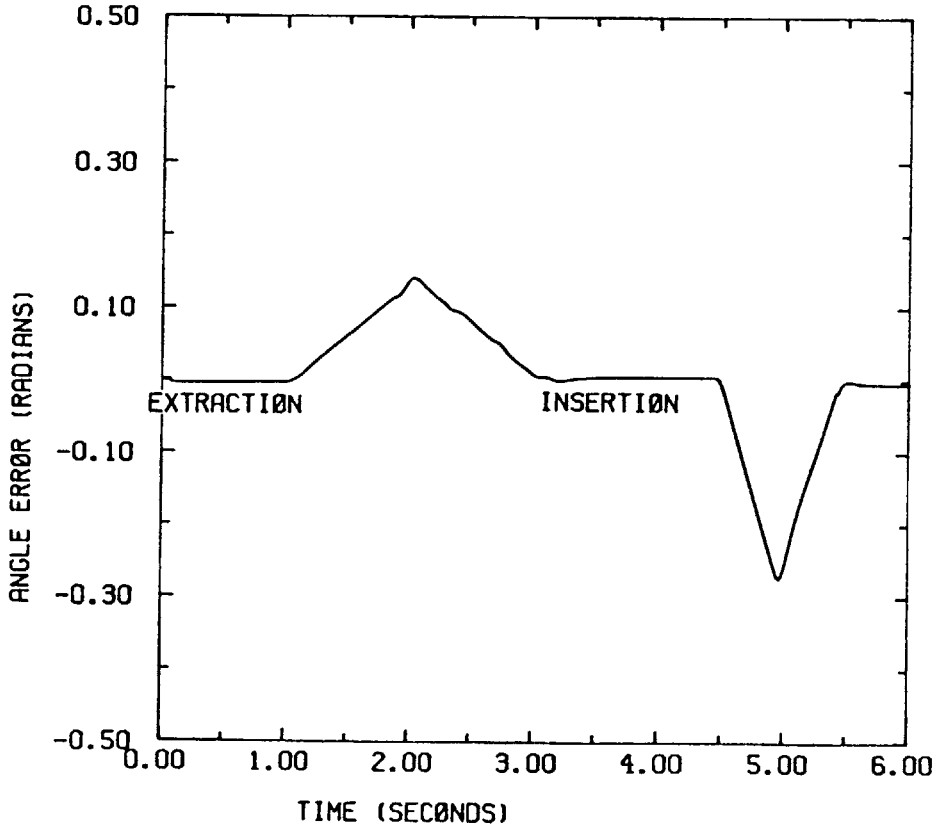

FIGURE 4-19 40 LB PAYLOAD ANGLE ERROR

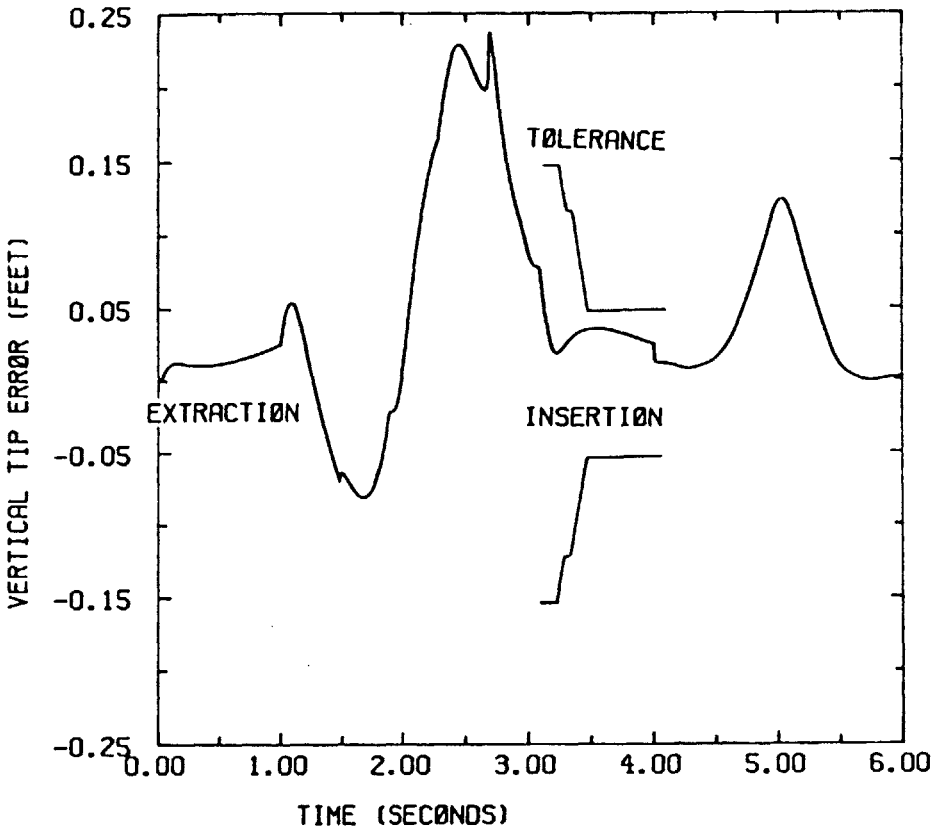

FIGURE 4-20 40 LB PAYLOAD VERTICAL TIP ERROR

The robotic loader system successfully loads ammunition when controlled with the linear PD uncoupled controller. The system meets all the performance specifications. The ammunition is successfully loaded into the main gun, the actuators retain over one third of their torque capabilities for disturbance rejection, the controller operates without exciting the structural resonances, and the system is robust enough to handle the various payloads. This controller will next be tested with the robot base experiencing motion disturbances. This is presented in Chapter 5.

4.4 COUPLED CONTROLLER

A coupled controller is more complex and difficult to design. A coupled controller is now designed and tested in an attempt to obtain better performance over the uncoupled controller. A coupled controller accounts for the natural coupling which occurs between joints, and can result in improved performance. For multiple input, multiple output (MIMO) systems, the selection of closed-loop eigenvalues does not uniquely define a closed-loop system [30]. The entire eigenstructure assignment method is used to assign both a closed-loop eigenvalue spectrum and an associated set of eigenvectors. These are then used to obtain a set of gains for a coupled controller [16].

Eigentructure Assignment

The linearized system is represented by the same matrix state equation 4.3. The control law for applying state variable feedback is given by equation 4.7. Substitution of equation 4.7 into equation 4.3 results in the closed-loop system equation, 4.12.

$$\dot{X} = Ax + B(r-KX) \tag{4.11}$$

$$= [A - BK]X + Br$$

$$= A_{cl}X + Br \tag{4.12}$$

where: A_{cl} is the closed-loop system matrix

The closed-loop eigenvalues and eigenvectors are related by equation 4.13, which results in the gain matrix, equation 4.17.

$$[A - Bk]v_i = \lambda_i v_i \tag{4.13}$$

where: λi is the ith eigenvalue

vi is the ith eigenvector

$$[A - \lambda_i I \mid -B] \begin{bmatrix} v_i \\ q_i \end{bmatrix} = 0 \tag{4.14}$$

where: q_i is defined as Kv_i

$= Kv_i$

Forming the augmented matrices V, equation 4.15, and Q, equation 4.16, with the vectors v_i and q_i, for i = 1 to 6 the system order, results in the gain matrix equation 4.17.

$$V = [v_1 \; v_2 \; v_3 \; v_4 \; v_5 \; v_6] \tag{4.15}$$

$$Q = [q_1 \; q_2 \; q_3 \; q_4 \; q_5 \; q_6] \tag{4.16}$$

$$K = QV^{-1} \tag{4.17}$$

Selection of closed-loop eigenvalues is dependent upon the desired system characteristics. The dominate poles are placed to achieve the desired system response and the remaining poles are placed far enough away from the dominate poles to prevent their interfering with the desired dominate pole response characteristics. Figure 4-21 illustrates the selection of closed-loop poles. The selected closed-loop eigenvalues follow the same rationale as for the uncoupled controller. Six eigenvalues and eigenvectors are necessary since the robotic system has three degrees of freedom and a PD controller which uses joint angle position and velocity.

The desired closed-loop eigenvalues are as follows:

$$\lambda_1 = -20 + 20*j$$

$$\lambda_2 = -20 - 20*j$$

$$\lambda_3 = -57.5$$

$$\lambda_4 = -60.0$$

$$\lambda_5 = -62.5$$

$$\lambda_6 = -65.0$$

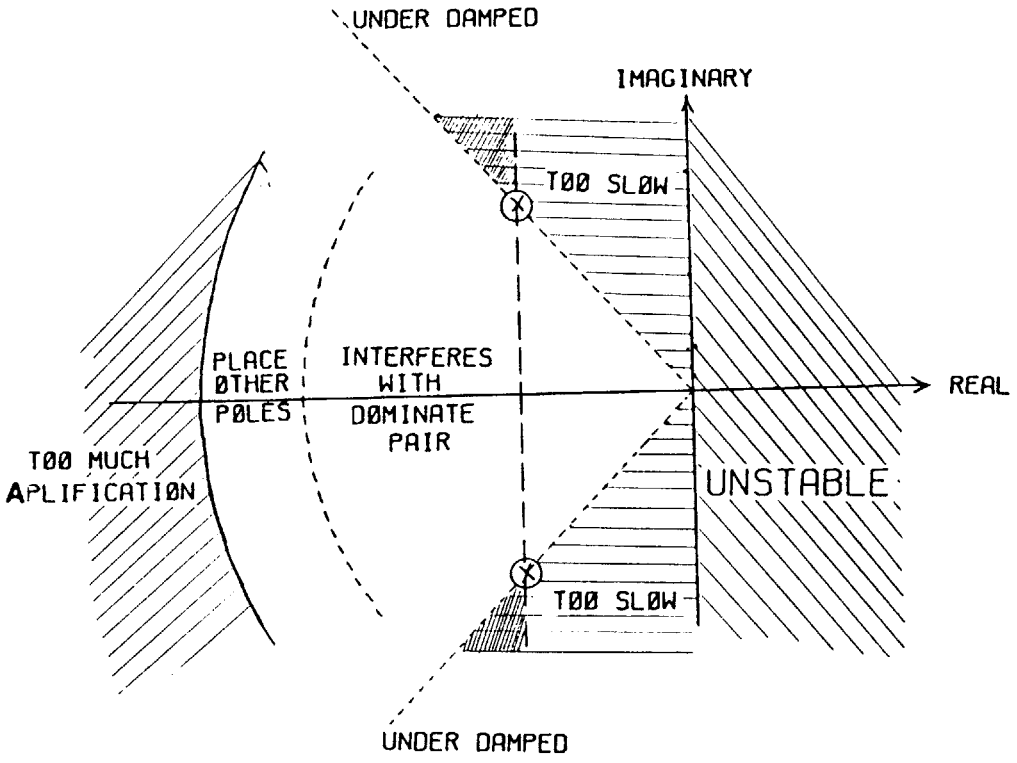

FIGURE 4-21 PLACEMENT OF POLES

The eigenvectors are selected to achieve the desired time response characteristics and the amount of coupling between the modes. Following the eigenstructure assignment procedure described in reference 16, the eigenvectors are selected. A augmented matrix, SS, is formed as shown by equation 4.18.

$$SS = \left[\begin{array}{c|c} -(\lambda_i I - A) & -B \\ \hline 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 & \\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 & \\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 & \end{array}\right] \qquad (4.18)$$

Then each selected eigenvalue is substituted for λ_i in the (9x9) matrix, SS. Elementry row operations are performed to achieve a form with zeros below the principle diagonal. The three resulting columns which have a -1 on the principle diagonal, c_1, c_2, and c_3, specify the null space, which the selected eigenvector must span. A linear combination of the three columns, c1, c2, and c3, shown by equation 4.20, becomes the augmented vector, VQ, equation 4.19. The top six elements of this vector form the obtained eigenvector, v_i.

$$VQ_i = \left[\begin{array}{c} v_i \\ \hline q_i \end{array}\right] \qquad (4.19)$$

where: $$VQ_i = a_1c_1 + a_2c_2 + a_3c_3 \qquad (4.20)$$

The factors, a1, a2, and a3, are arbitarily chosen so that the resulting six eigenvectors are linearly independent.

This procedure is repeated for each eigenvalue until six such vectors are obtained, one for each eigenvalue. The obtained six vectors are then separated into the Q and V matrices by eguation 4.21.

$$\begin{bmatrix} V \\ \hline Q \end{bmatrix} = \left[\begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\ \hline q_1 & q_2 & q_3 & q_4 & q_5 & q_6 \end{array}\right] \qquad (4.21)$$

The above procedure describes the method of obtaining the eigenvectors corresponding to a desired real eigenvalue. For complex closed-loop eigenvalues a procedure similiar to this is utilized [16]. MATRIXX, a computer software package for performing matrix operations [17], in conjunction with the macro files listed in Appendix E, is used for determining the eigenvectors for given eigenvalues.

The linear coupled controller is designed for a particular set of nominal values, just as the uncoupled controller is. The linearized terms which determine the system matrices, from which contoller gains are determined using the eigenstructure assignment described above, are functions of the selected system's operating values. These operating values change substantially with the different arm configurations which occur during the baseline loading trajectory. Correspondingly, the system matrices change, thus resulting in different gain matrices. Investigation of the resulting gain matrices, calculated for the various operating values, suggests a grouping based on the phase of the trajectory.

The chosen phases occur first, when the round is being extracted from the storage rack, called the extraction phase. The second, the flip phase, occurs when the ammunition payload is flipped 180 degrees, changing the rounds orientation to line up with the breach. The third phase, called the insertion phase, is when the round is inserted through the breach into the main gun chamber. The last phase is when the robot arm returns to the storage rack to grasp a new round. Selecting a controller gain matrix from the approximate middle of these phases results in the gains shown in figures 4-22, 4-24, 4-26, and 4-28. The robotic loader configurations which corresponds to these gain matrices are shown in figures 4-23, 4-25, 4-27, and 4-29.

$$K1 = \begin{bmatrix} -379200 & 2780 & 76130 & 5180 & 12870 & 770 \\ -611320 & -8110 & 52510 & 3510 & 5110 & 310 \\ 72240 & 1470 & -180 & 5 & 4010 & 230 \end{bmatrix}$$

FIGURE 4-22 EXTRACTION PHASE GAINS

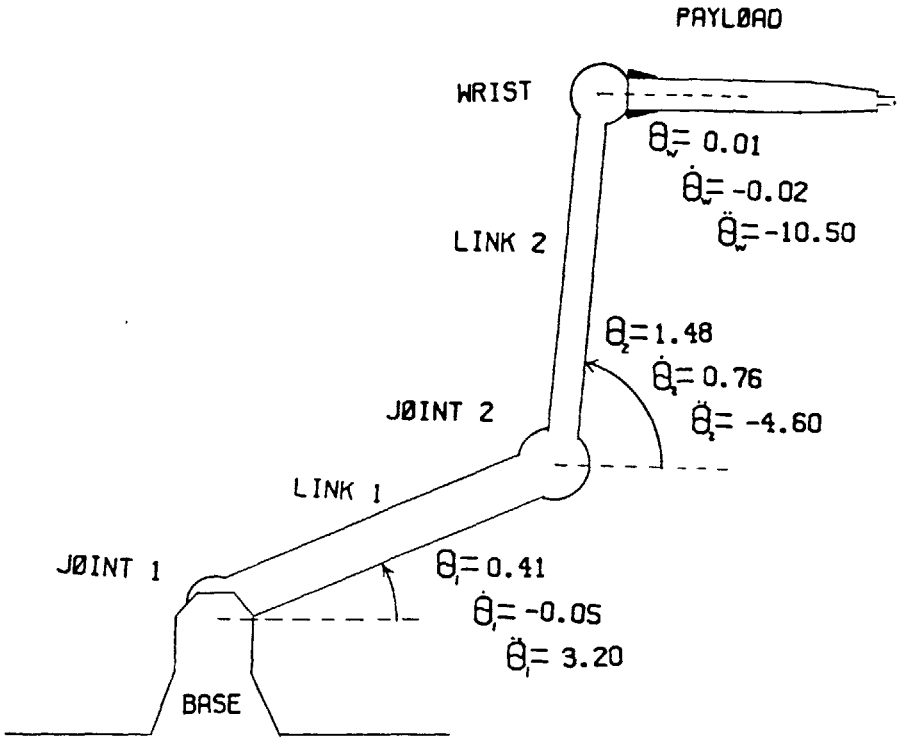

FIGURE 4-23 EXTRACTION PHASE CONFIGURATION

$$K2 = \begin{bmatrix} 731780 & 18200 & 53080 & 3540 & 2280 & 280 \\ 425560 & 6140 & 66660 & 4350 & 7580 & 580 \\ 57100 & 680 & 8240 & 470 & 3990 & 240 \end{bmatrix}$$

FIGURE 4-24 FLIP PHASE GAINS

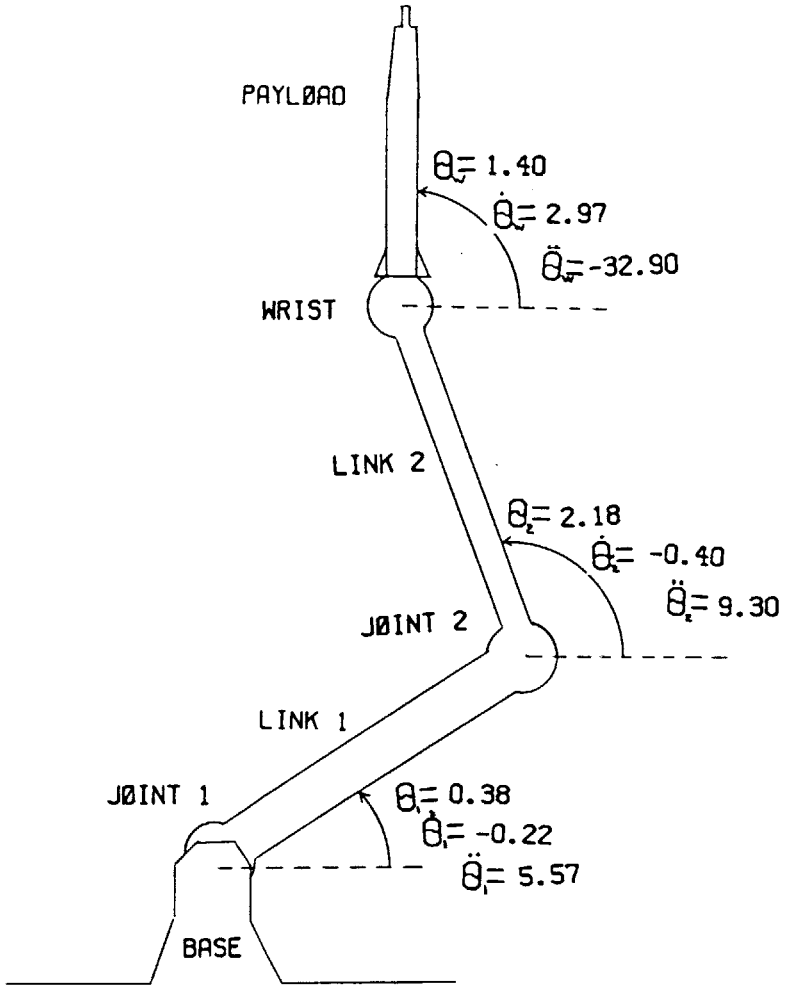

FIGURE 4-25 FLIP PHASE CONFIGURATION

$$K3 = \begin{bmatrix} 478310 & 13040 & 29380 & 1880 & 4590 & 320 \\ 239630 & 2070 & 63010 & 4280 & 7410 & 610 \\ 39840 & 410 & 9090 & 530 & 3960 & 240 \end{bmatrix}$$

FIGURE 4-26 INSERTION PHASE GAINS

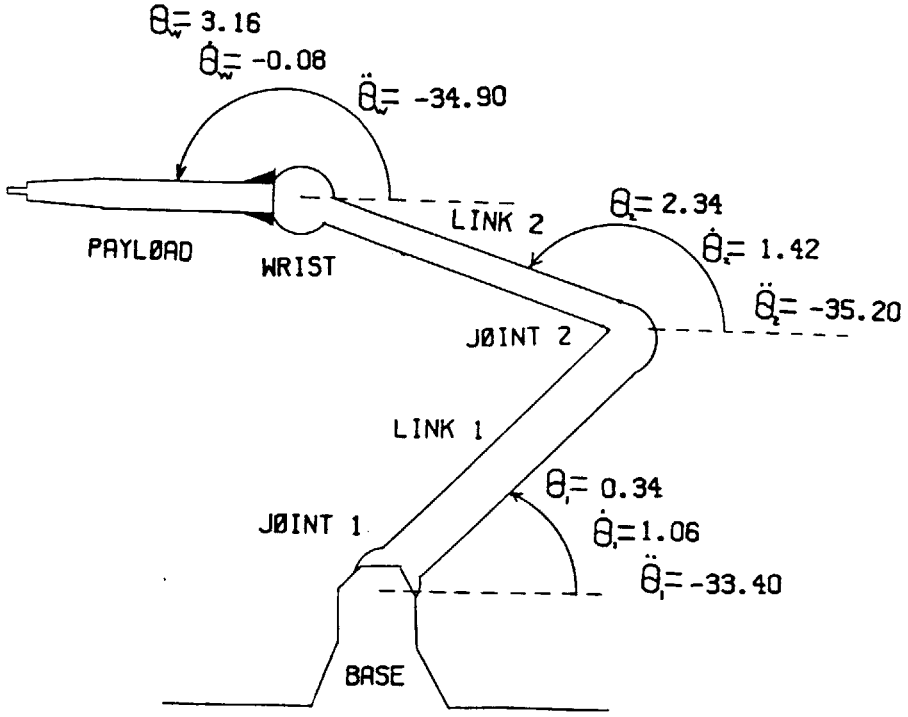

FIGURE 4-27 INSERTION PHASE CONFIGURATION

$$K4 = \begin{bmatrix} -67570 & 6520 & 55100 & 3630 & 13340 & 840 \\ -584070 & -8860 & 55330 & 3670 & 5500 & 380 \\ 36000 & 920 & 930 & 50 & 4070 & 240 \end{bmatrix}$$

FIGURE 4-28 RETURN PHASE GAINS

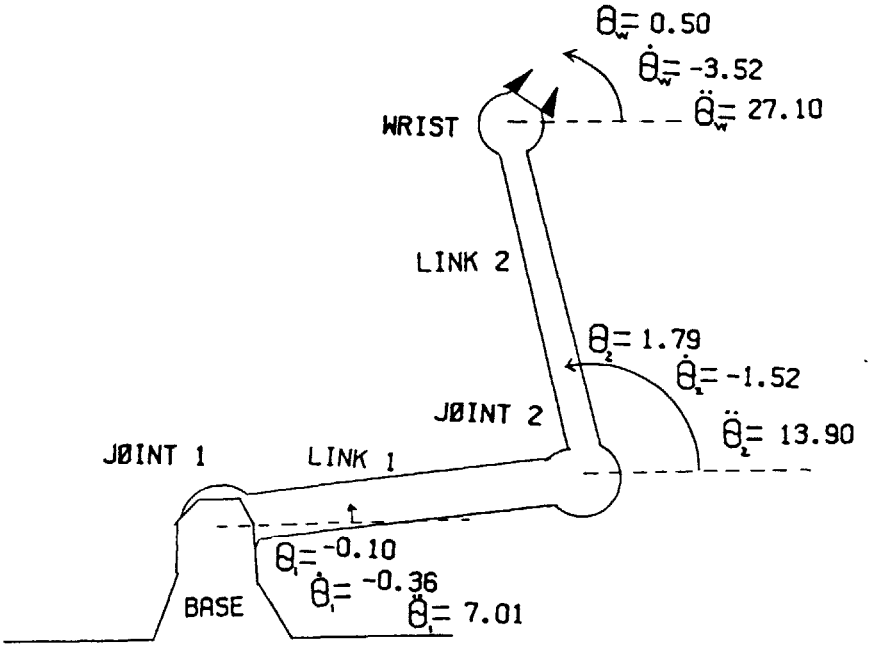

FIGURE 4-29 RETURN PHASE CONFIGURATION

Simulations, using the gain matrices obtained for the different phases, show that good performance is achieved in the portion of the trajectory for which the gains are designed. These results, shown in figures 4-30, 4-31, 4-32, and 4-33, suggest that gain scheduling can be used to obtain good performance over the entire trajectory. Gain scheduling is a control technique which adjusts the controller gains as the system's configuration changes. Simulations, though, using the four gain matrices corresponding to the four phases, resulted in poor performance. The system became unstable, shown in figure 4-34. This result is explained by the large switching transients between the different gain matrices. To reduce these large transcients, either the number of different gains needs to be increased or linear interpolation used between the gains. The resulting controller would be more complex than warranted for this research.

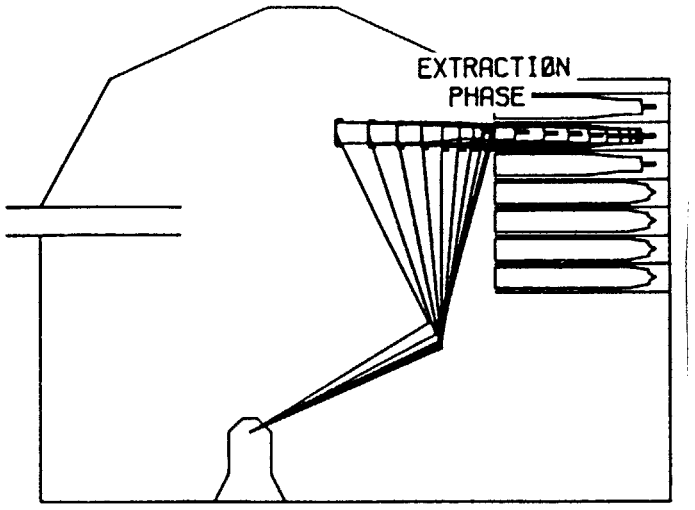

FIGURE 4-30 ROBOT MOTION FOR EXTRACTION PHASE GAINS

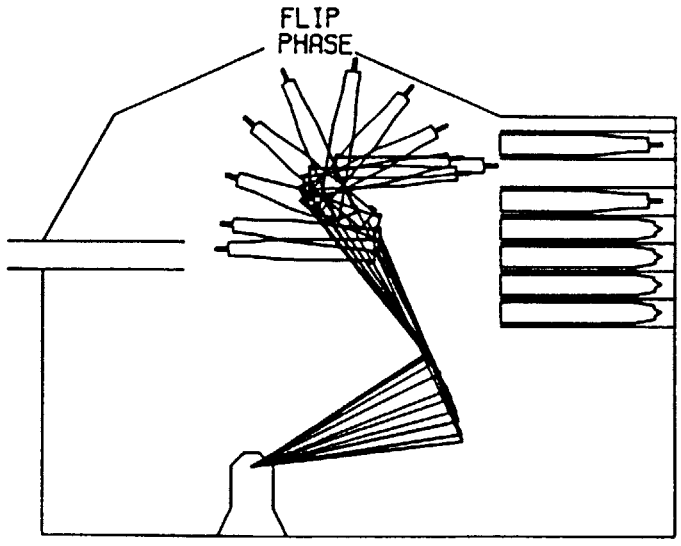

FIGURE 4-31 ROBOT MOTION FOR FLIP PHASE GAINS

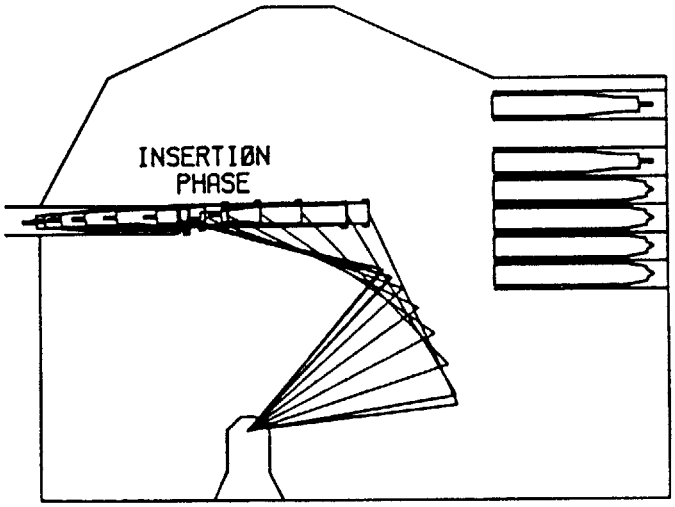

FIGURE 4-32 ROBOT MOTION FOR INSERTION PHASE GAINS

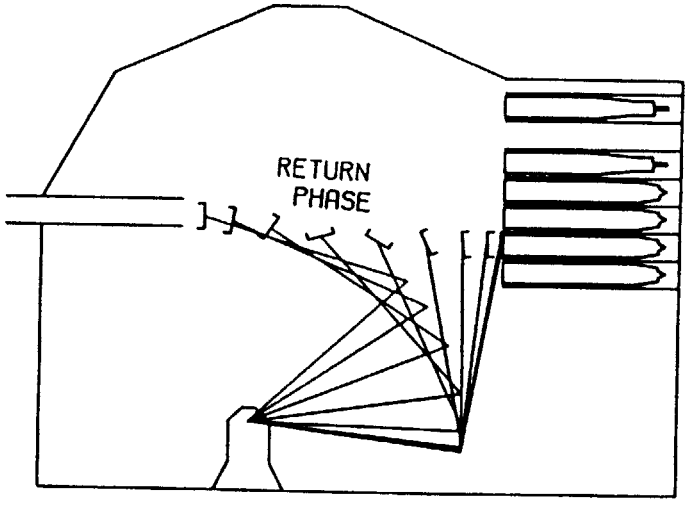

FIGURE 4-33 ROBOT MOTION FOR RETURN PHASE GAINS

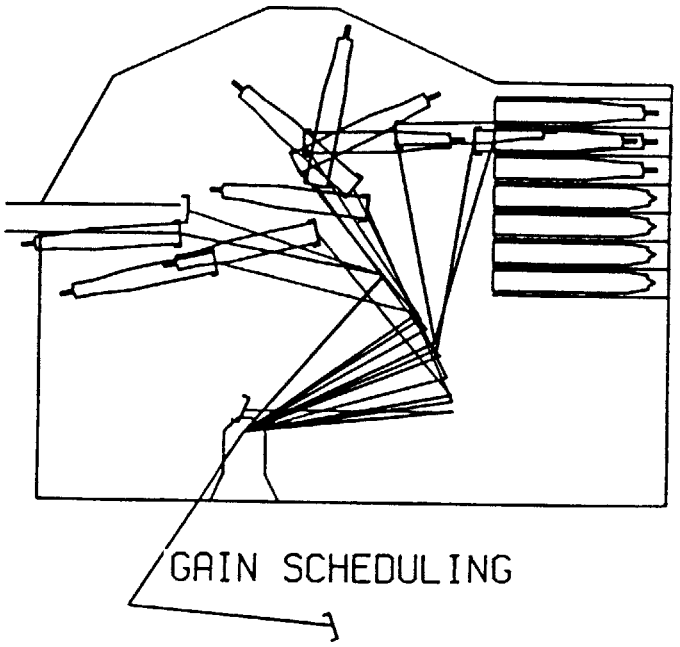

FIGURE 4-34 ROBOT MOTION FOR GAIN SCHEDULING

The gain matrix which resulted in the best performance over the entire trajectory, is the flip phase controller gains. Adjusting this set of gains, a constant gain coupled PD controller is obtained which completes the required trajectory and satisfactorily inserts the ammunition into the main gun, figure 4-35. Figure 4-36 shows that this controller's eigenvalues are higher than desirable, the largest imaginary pole is 42.0, over the desired maximum of 30.0, but still less than half of the structural resonances, 94.0. Figure 4-37, the actuator torques, are greater than those required by the uncoupled PD controller, figure 4-11. The coupled controller just reaches joint one's saturation limit, but still follows the

prescribed trajectory. Figure 4-38 and 4-39 show that the joint angle accuracy is comparable to that obtained by the uncoupled controller, figures 4-12 and 4-13. Figure 4-40 shows that the payload angle error is within allowable tolerances. Figure 4-41, the payload vertical tip error, shows that the payload is successfully inserted and the endeffector is aligned to grasp the new round at the end of the baseline trajectory. The error is greater than the uncoupled controller's error during extraction, but the controller will employ force control to extract the round from the storage rack.

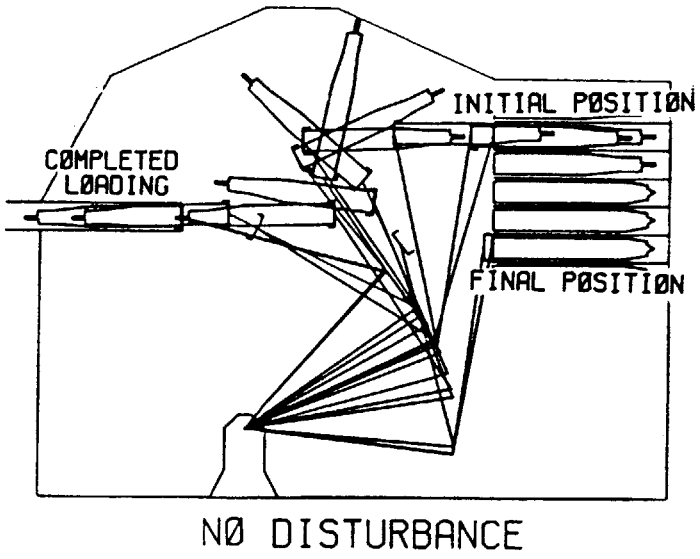

FIGURE 4-35 COUPLED PD CONTROLLED ROBOT MOTION

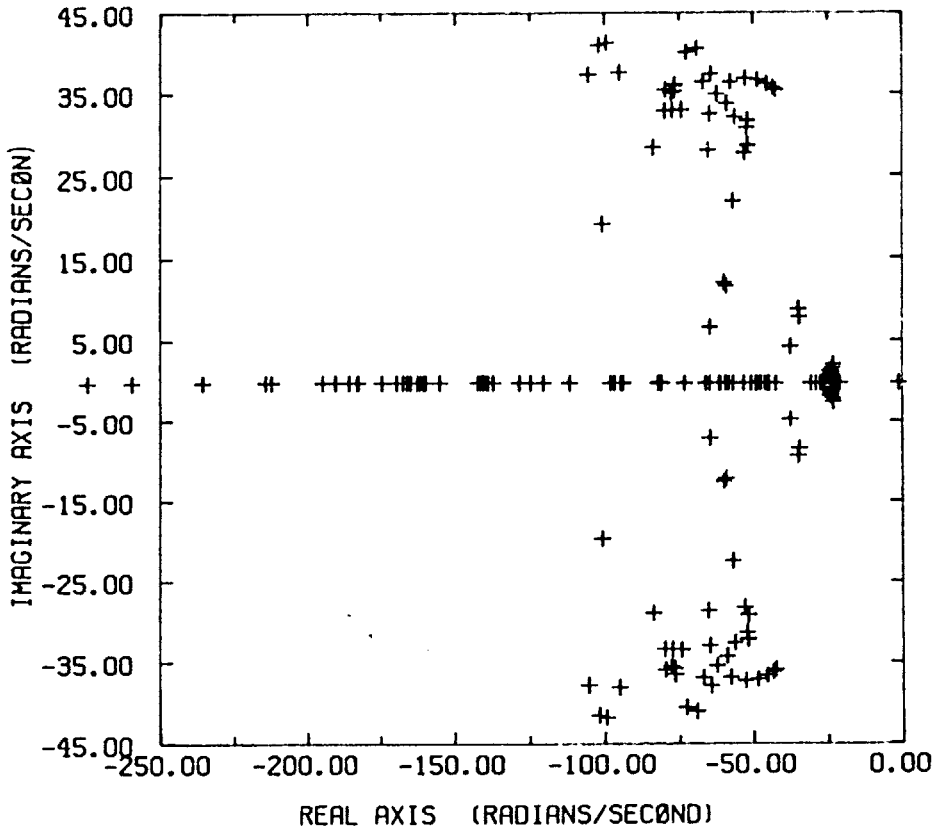

FIGURE 4-36 COUPLED PD CONTROLLER CLOSED-LOOP POLES

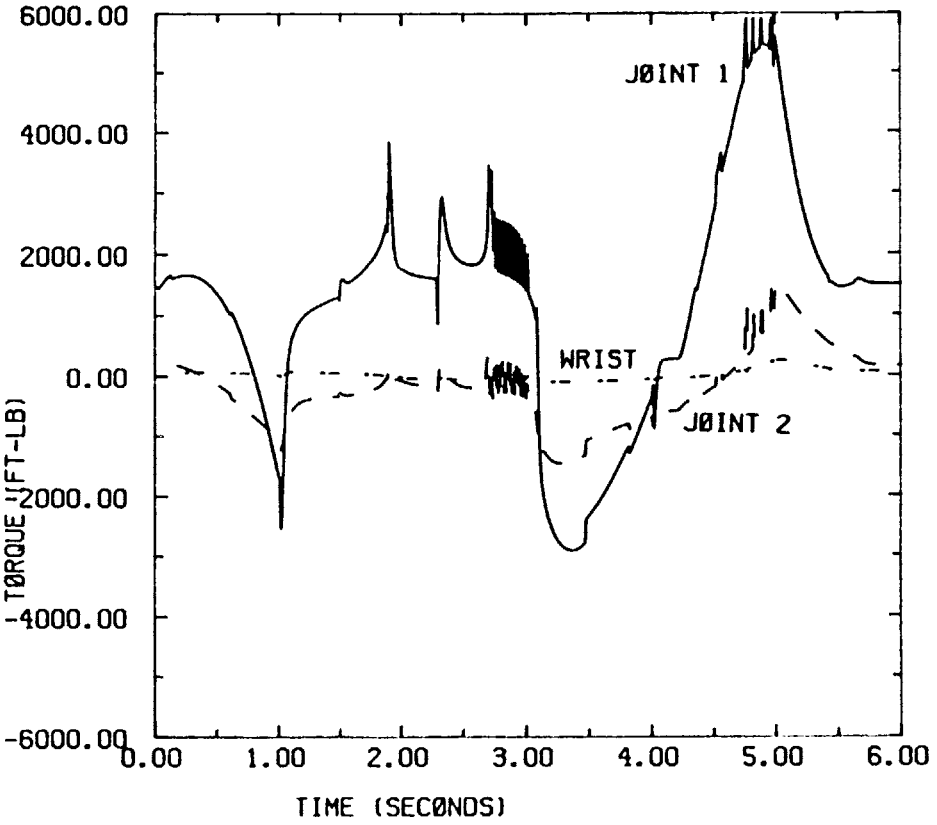

FIGURE 4-37 COUPLED PD CONTROLLED MOTOR TORQUES

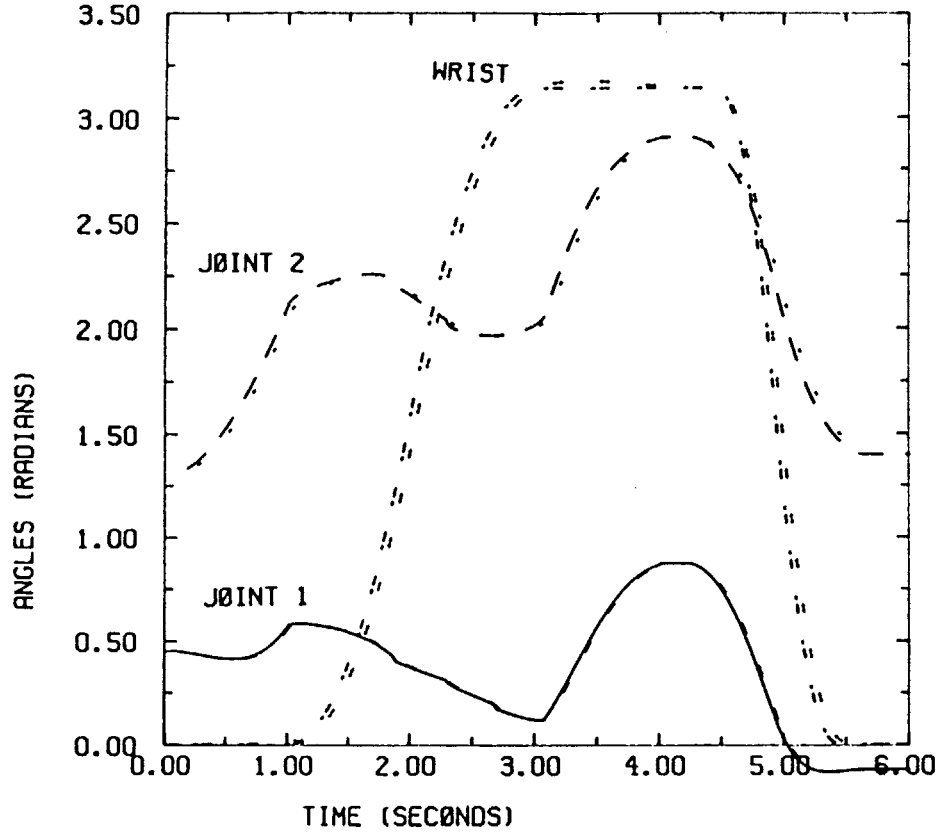

FIGURE 4-38 COUPLED PD CONTROLLED JOINT ANGLE COMPARISON

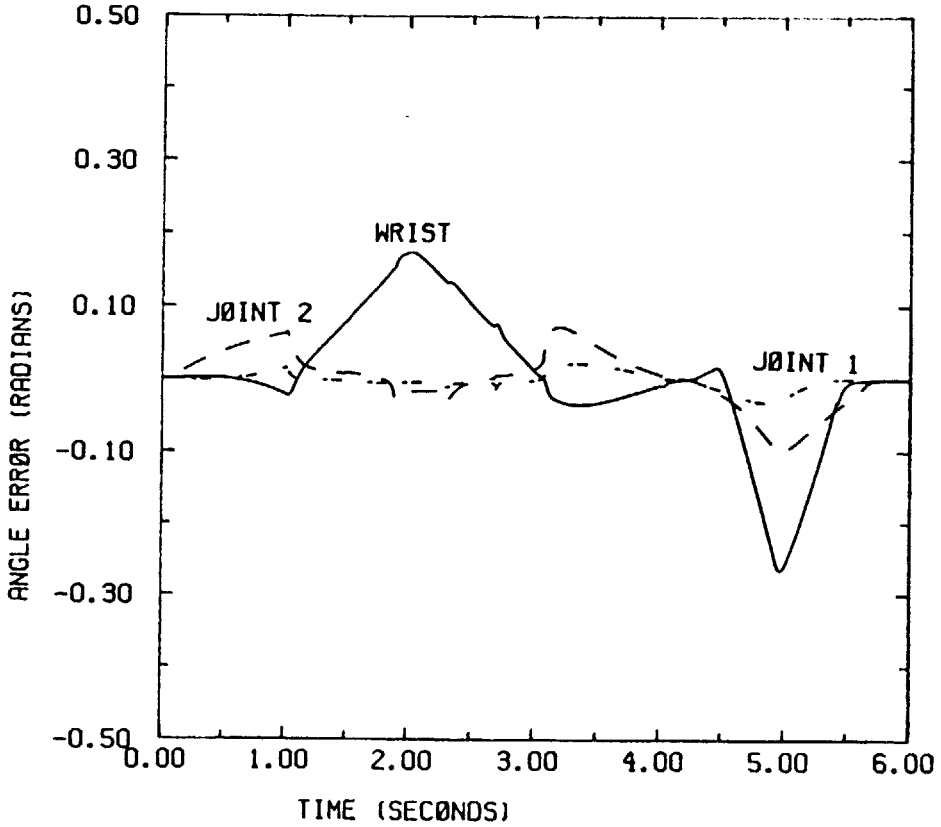

FIGURE 4-39 COUPLED PD CONTROLLED JOINT ANGLE ERRORS

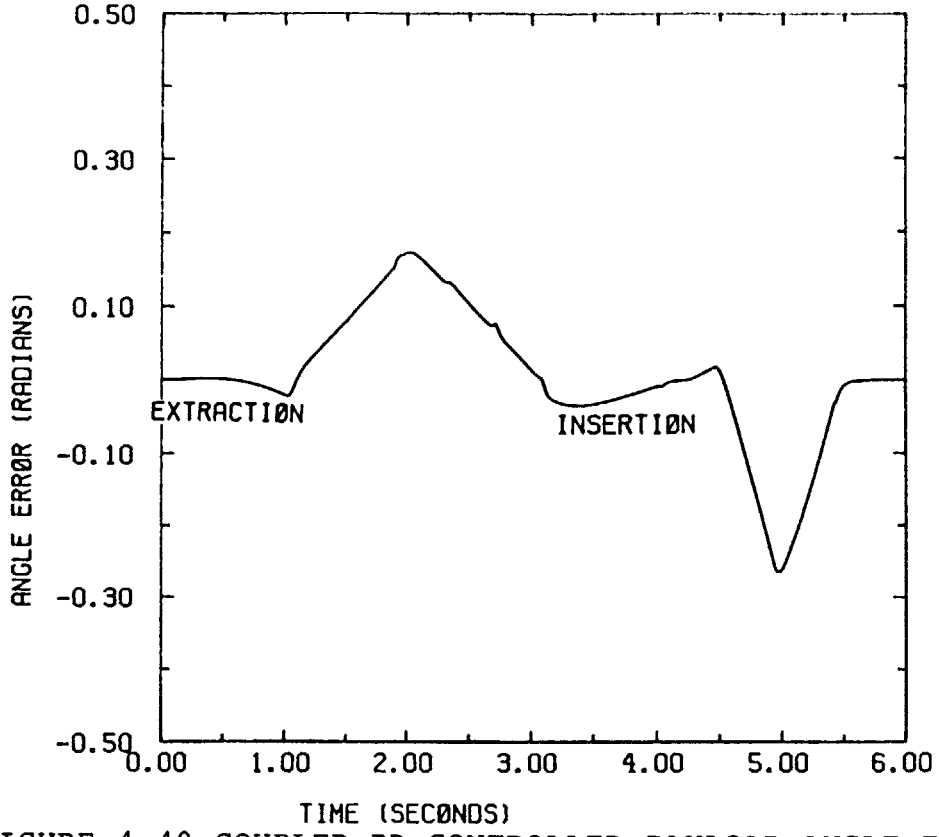

FIGURE 4-40 COUPLED PD CONTROLLED PAYLOAD ANGLE ERROR

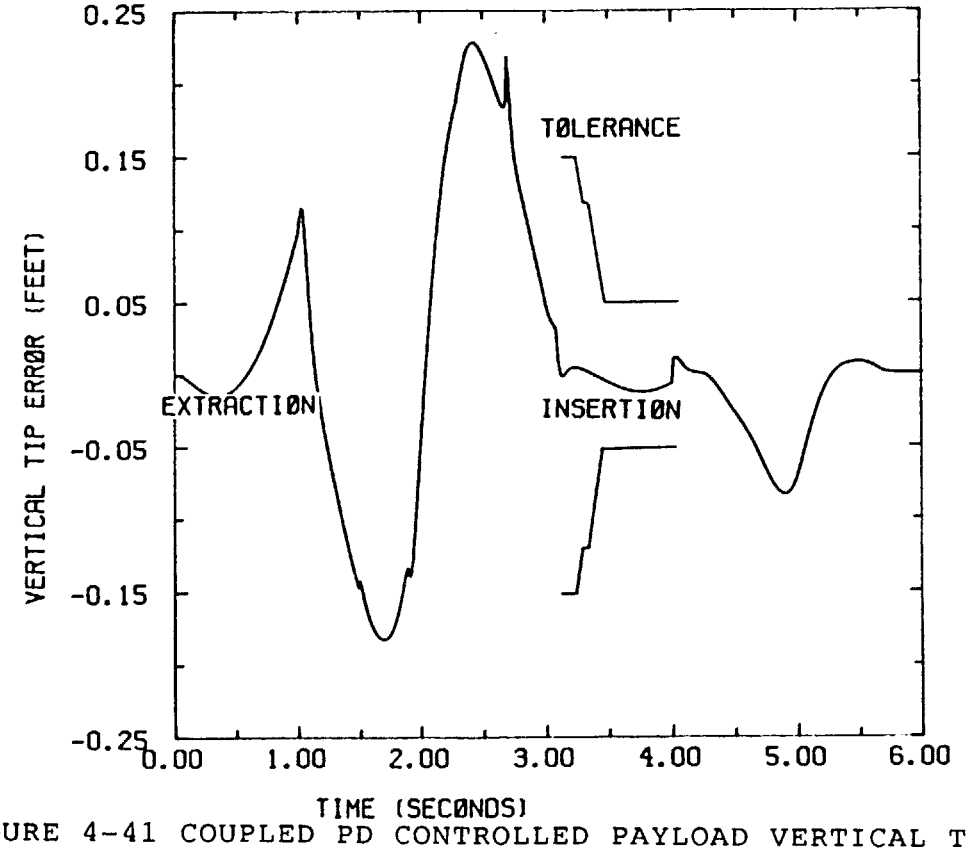

FIGURE 4-41 COUPLED PD CONTROLLED PAYLOAD VERTICAL TIP ERROR

The linear coupled PD controller, with constant gains, satisfactorily loads the ammunition payload into the main gun. The performance of the controller can probably be improved upon. The controller gains should be adjustable to reduce the eigenvalues, lower the actuator torques, and reduce the extraction error. However, for this research, the performance is sufficient to demonstrate the need for, and the use of, a compensator for base motion disturbances.

CHAPTER 5

SIMULATION RESULTS WITH DISTURBANCES

5.1 DECOUPLED CONTROLLER

The designed decoupled controller met all of the desired performance specifications. Next, simulations are conducted with vertical and rotational motions imposed on the base of the robot. All conditions are exactly the same as those imposed the stationary simulations, except that now base motion disturbances are subjected to the robotic system. First the modeled disturbances, figures 2-9 and 2-11, are used in the simulations, and then these results are confirmed using actual measurements for a tank moving across rough terrain, figures 2-10 and 2-12.

First, using the modeled disturbances, the resulting performance is unsatisfactory, in that the ammunition hits the chamber during insertion. Figure 5-1 shows the robotic loader conducting the baseline trajectory for loading ammunition. The diagrams on the right of the turret model show the motion of the turret relative to inertial space. The top figure, tank rotation, shows the pitch the tank experiences during the six second baseline trajectory. The indicator is initially vertical and then pitches back and forth. The middle figure, vertical displacement, shows the distance the tank moves vertically. This movement is the result of bumps the vehicle experiences during movement. The bottom figure, tank motion, is the combined motion,

rotational and vertical, of the tank.

Figure 5-2 shows that the torque required to perform the prescribed trajectory with base motion is greater than without, figure 4-11. The actuator torque requirements are greater, but the actuators are not completely saturated. Figures 5-3 and 5-4 show the accuracy obtained by each joint. Figure 5-4 clearly shows that the accuracy of the robot has been degraded by the base motion. The failure of the robotic loader is shown by the large payload vertical tip error during insertion. The ammunition payload fails to enter the breach, but hits the side of the main gun.

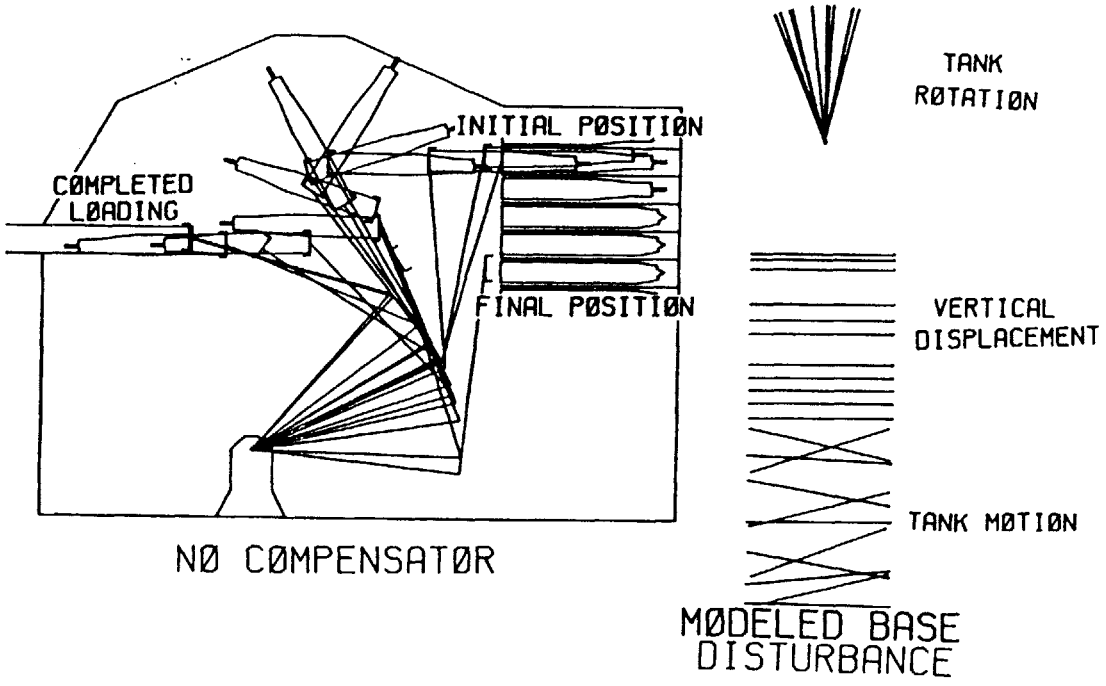

FIGURE 5-1 PD CONTROLLED ROBOT MOTION
WITH MODELED DISTURBANCES

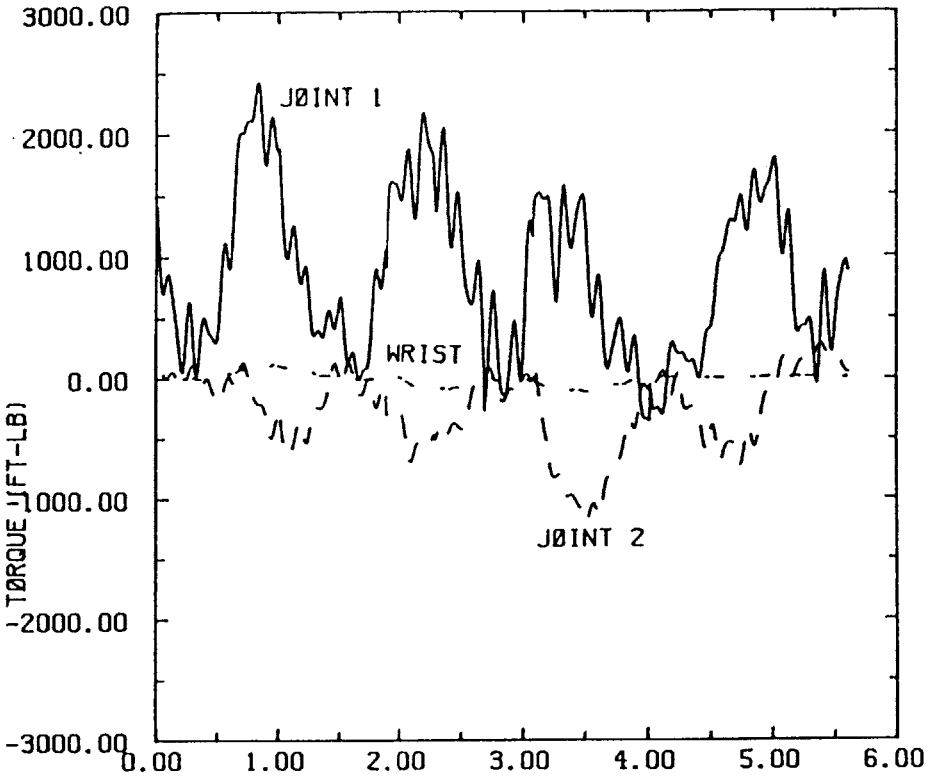

FIGURE 5-2 PD CONTROLLED MOTOR TORQUES
WITH MODELED DISTURBANCE

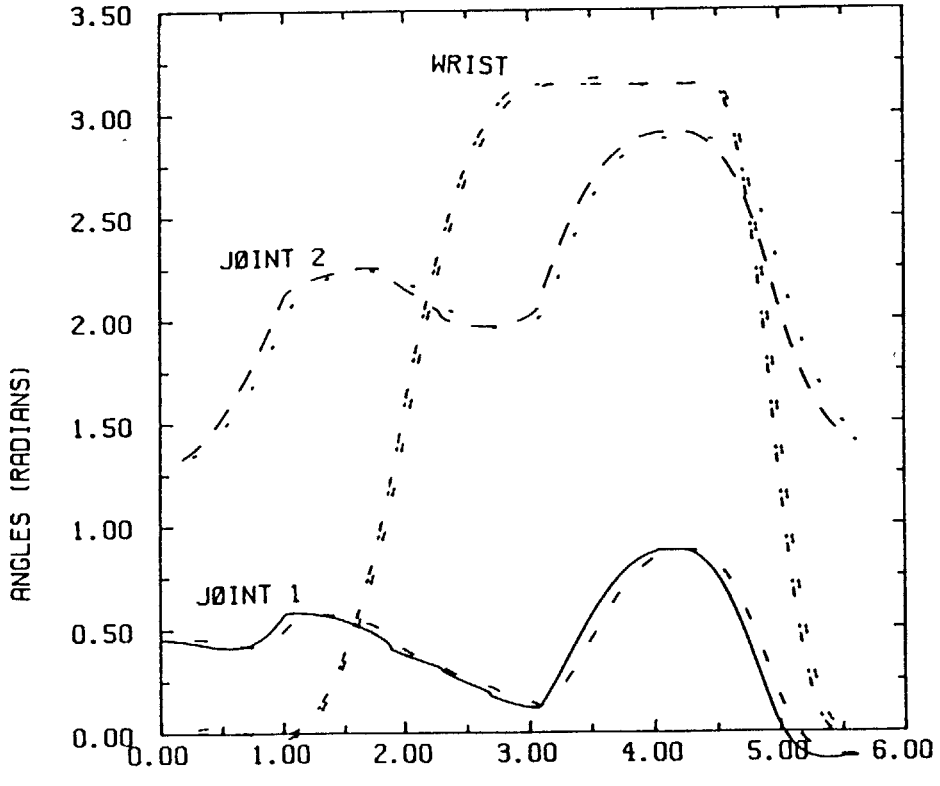

FIGURE 5-3 PD CONTROLLED JOINT ANGLE COMPARISON
WITH MODELED DISTURBANCES

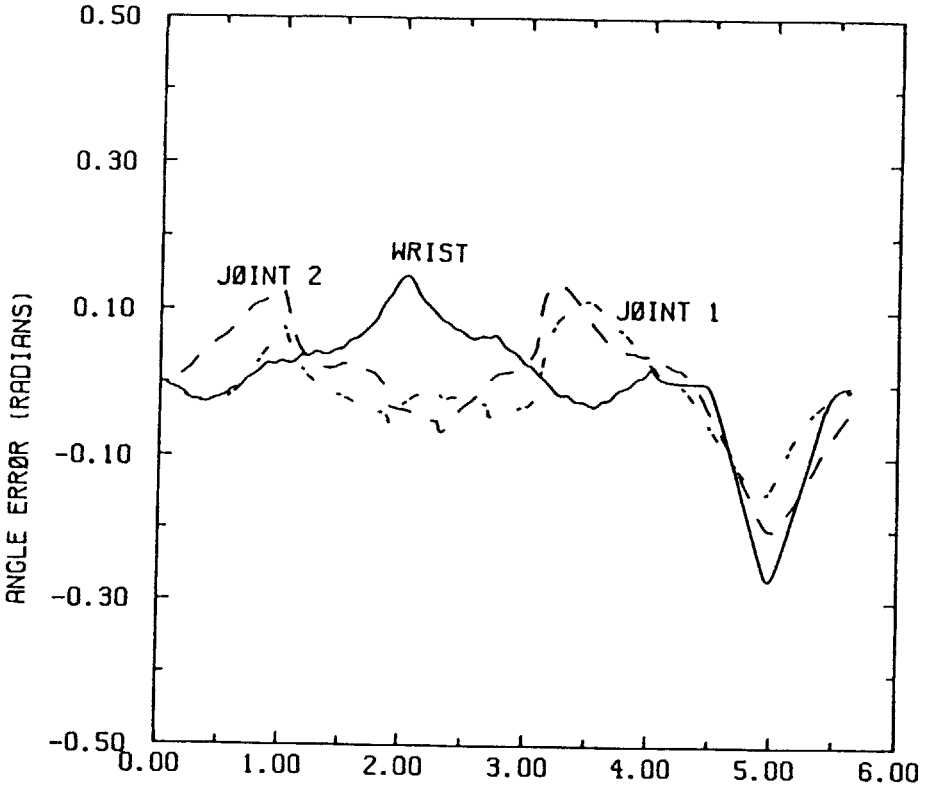

FIGURE 5-4 PD CONTROLLED JOINT ANGLE ERRORS WITH MODELED DISTURBANCES

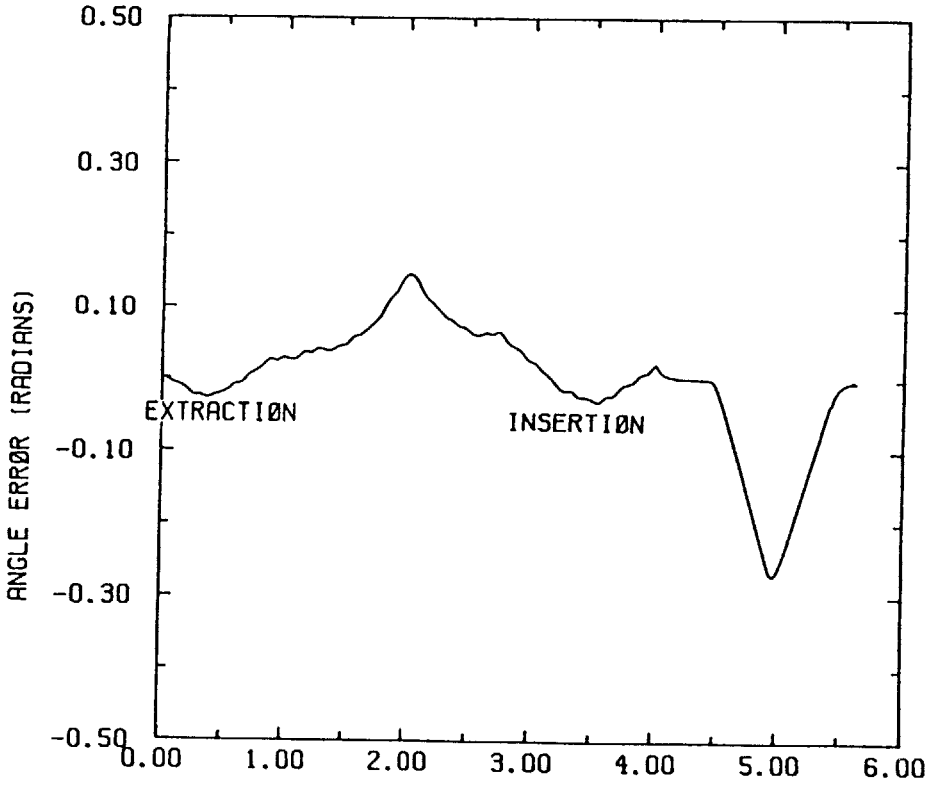

FIGURE 5-5 PD CONTROLLED PAYLOAD ANGLE ERROR WITH MODELED DISTURBANCES

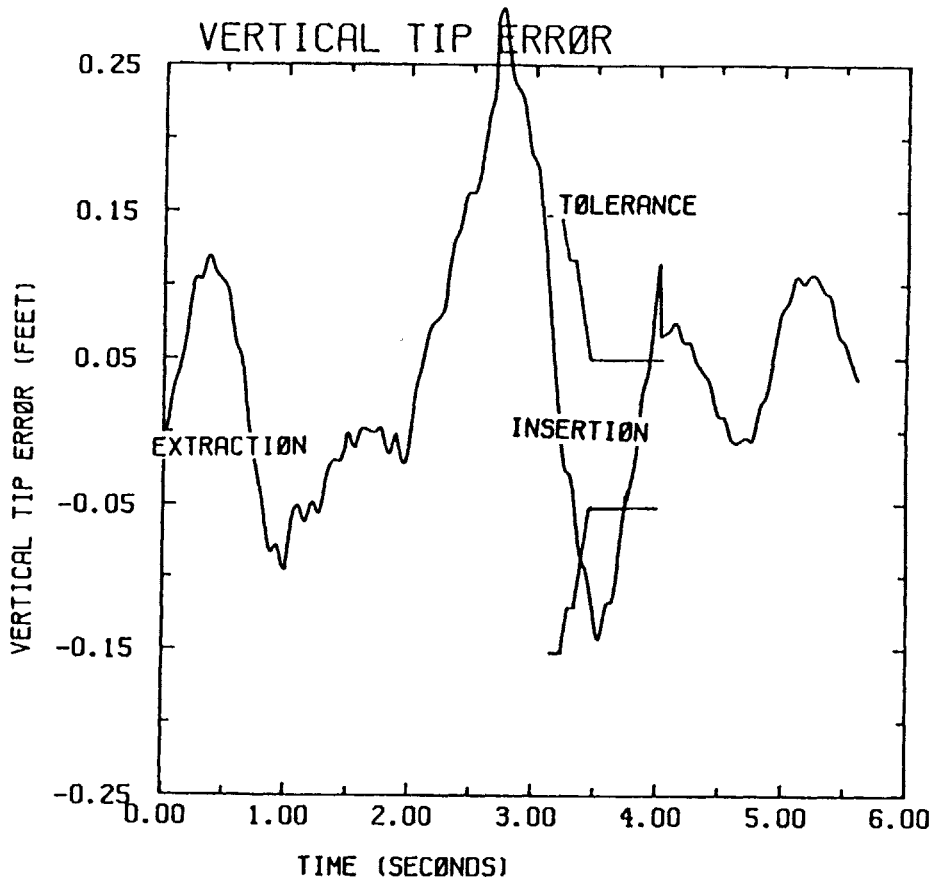

FIGURE 5-6 PD CONTROLLED PAYLOAD TIP ERROR WITH MODELED DISTURBANCES

Next, the uncoupled controller has the actual disturbances, figures 2-10 and 2-12, imposed on the robotic loader system. The results of these simulations are comparable to those obtained for the modeled disturbances. The performance of the robotic loader, figure 5-7, is unsatisfactory. The loader fails to insert the ammunition into the main gun without exceeding the allowed tolerances. This is best shown by figure 5-12, which shows the payload vertical tip error exceed the tolerances. The actuator torques, figure 5-8, are comparible to those for the modeled disturbances, figure 5-2. The joint angle accuracy, figures 5-9, 5-10 and 5-11, are also similar.

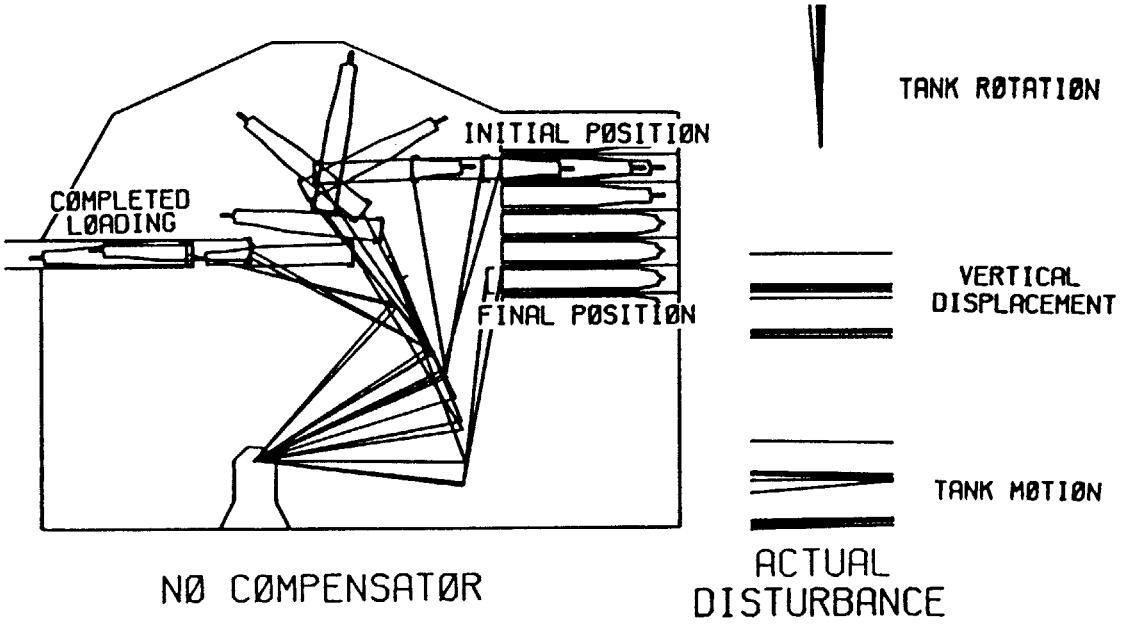

FIGURE 5-7 PD CONTROLLED ROBOT MOTION
WITH ACTUAL DISTURBANCES

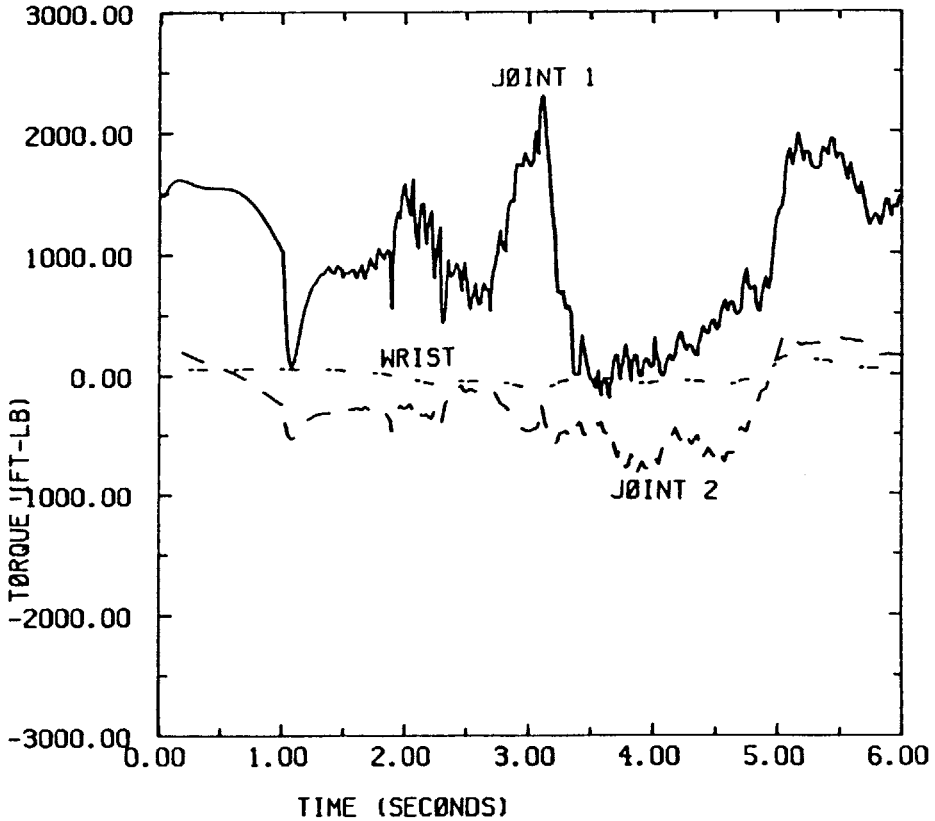

FIGURE 5-8 PD CONTROLLED MOTOR TORQUES
WITH ACTUAL DISTURBANCES

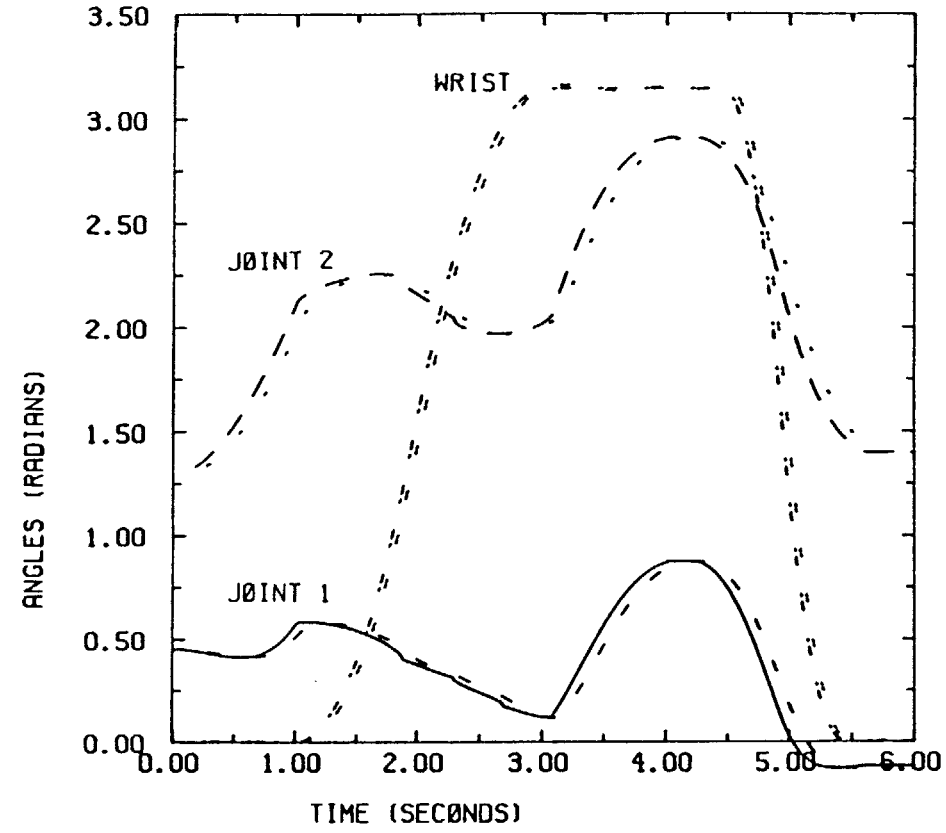

FIGURE 5-9 PD CONTROLLED JOINT ANGLE COMPARISON WITH ACTUAL DISTURBANCES

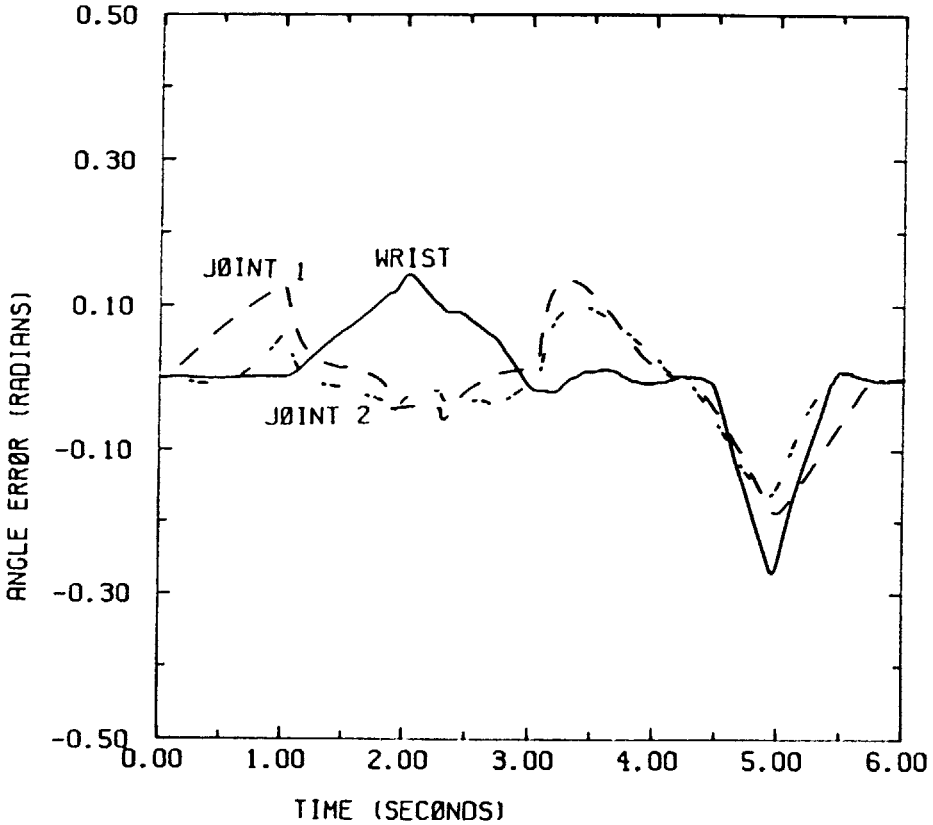

FIGURE 5-10 PD CONTROLLED JOINT ANGLE ERRORS WITH ACTUAL DISTURBANCES

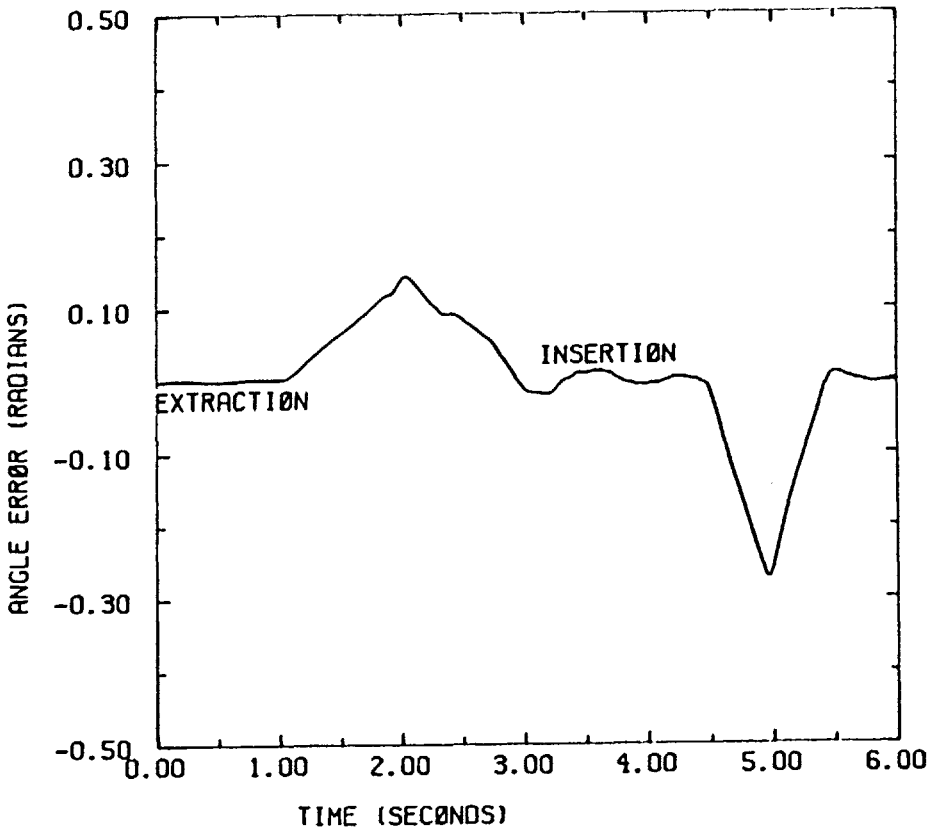

FIGURE 5-11 PD CONTROLLED PAYLOAD ANGLE ERROR WITH ACTUAL DISTURBANCES

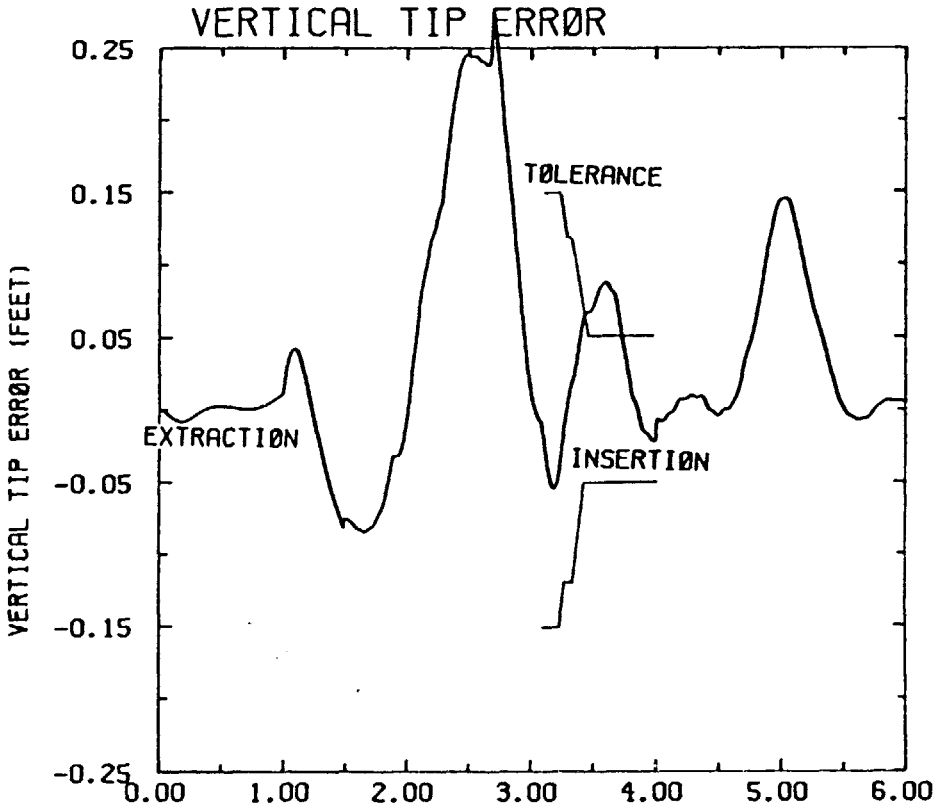

FIGURE 5-12 PD CONTROLLED PAYLOAD VERTICAL TIP ERROR WITH ACTUAL DISTURBANCES

5.2 COUPLED CONTROLLER

The coupled controller is more complex and possibly has a better chance for compensating for base motion disturbances. Simulations are conducted to confirm previous research that even advanced controllers alone can not provide good performance with the base moving. The resulting figures show that indeed the more complex coupled controller is unsuccessful in completing the prescribed motion when the robot is subjected to base disturbances. The following figures show the results which parallel those for the uncoupled controller.

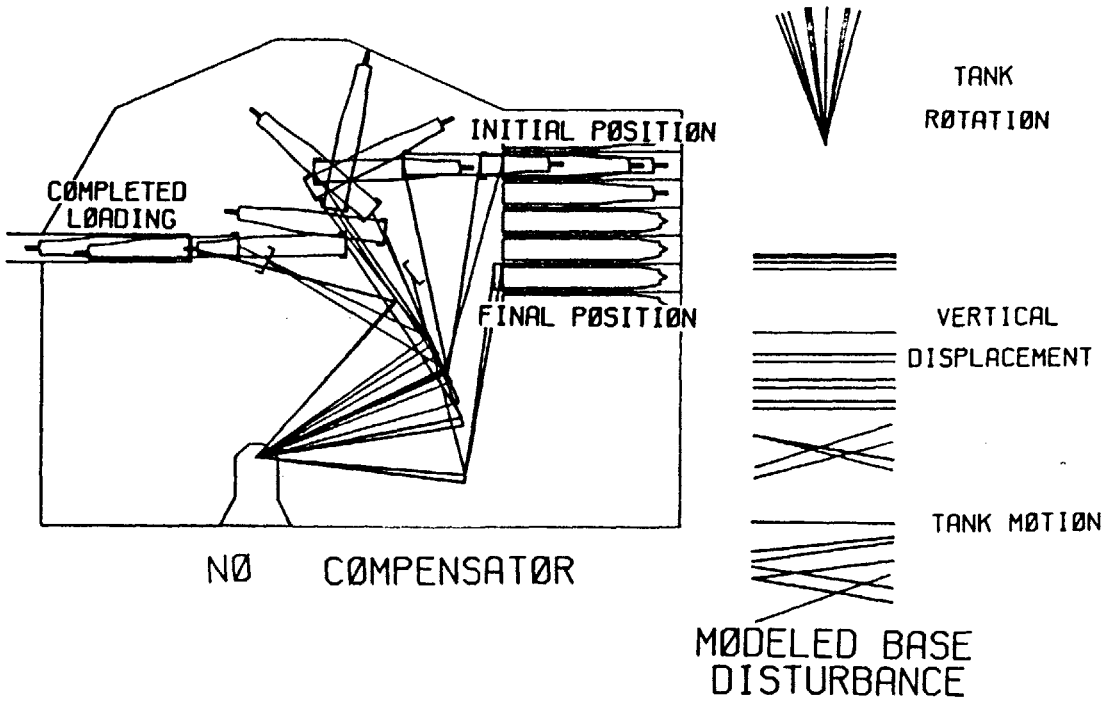

FIGURE 5-13 COUPLED PD CONTROLLED ROBOT MOTION WITH MODELED DISTURBANCES

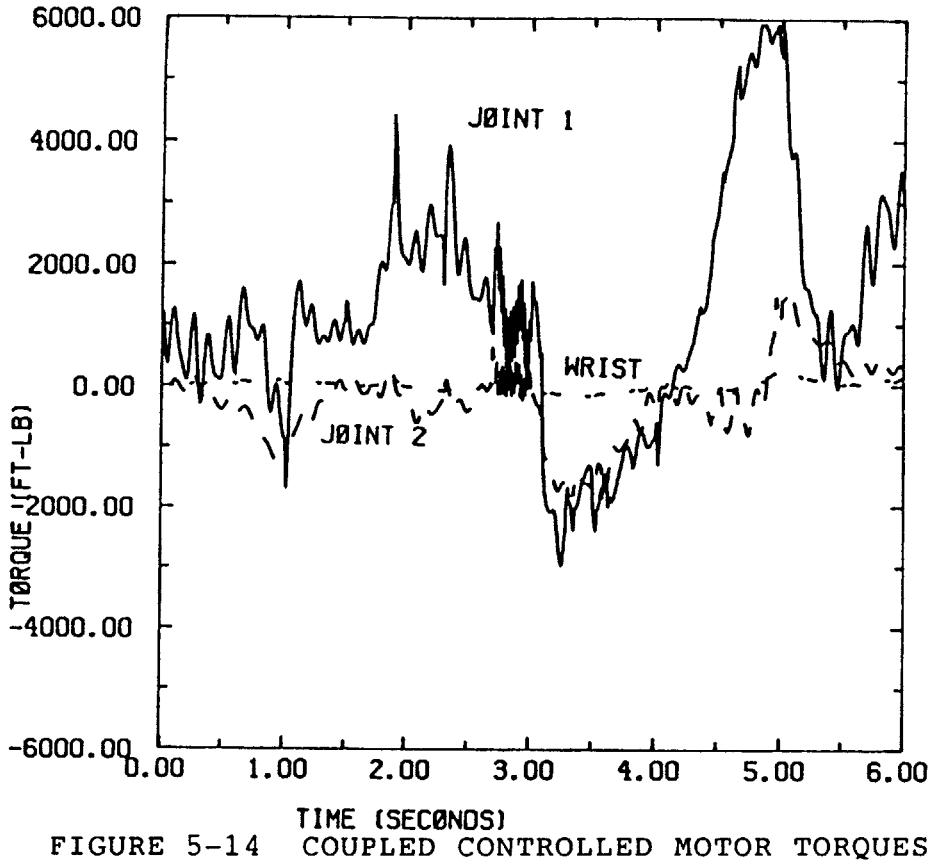

FIGURE 5-14 COUPLED CONTROLLED MOTOR TORQUES WITH MODELED DISTURBANCES

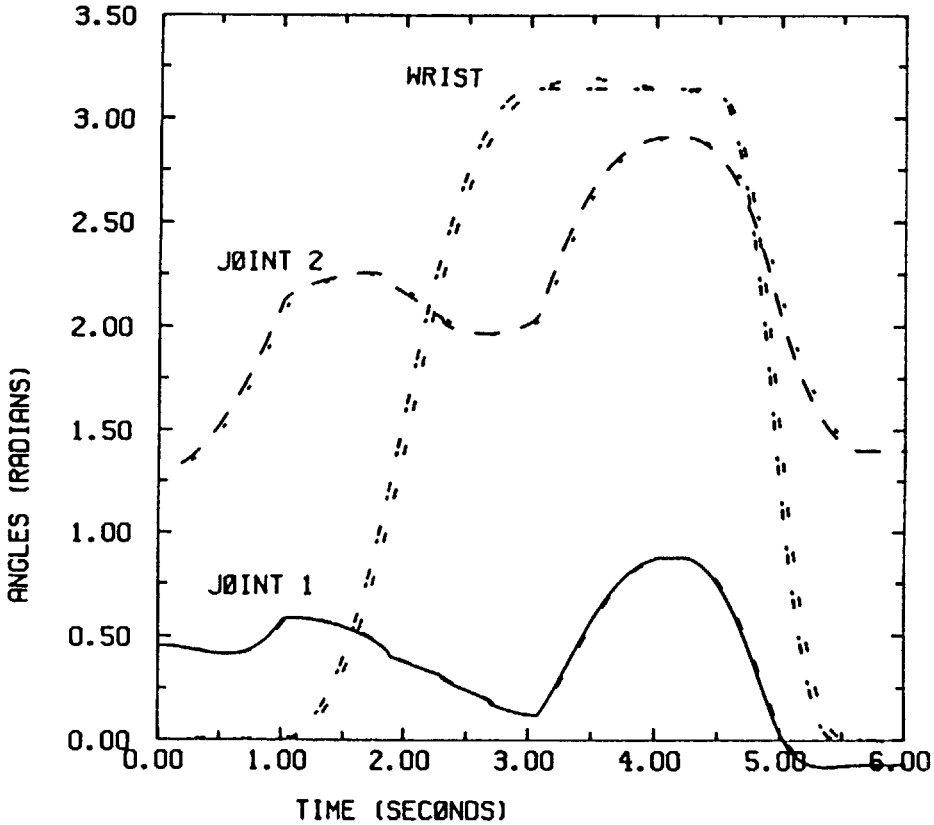

FIGURE 5-15 COUPLED PD CONTROLLED JOINT ANGLE COMPARISON WITH MODELED DISTURBANCES

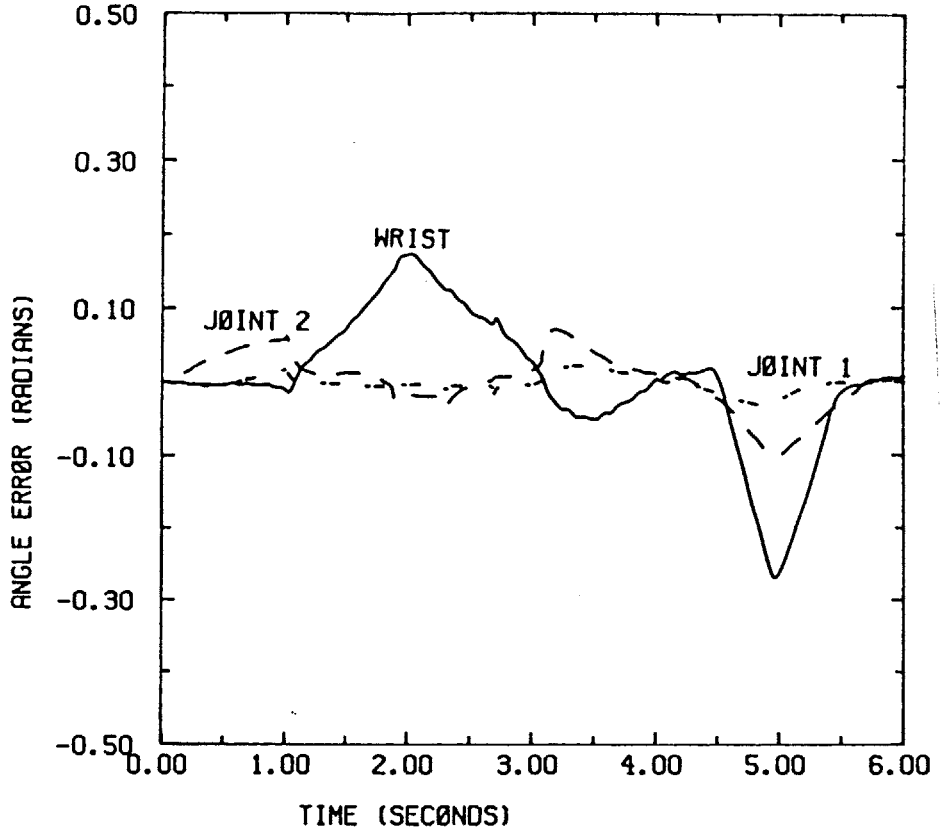

FIGURE 5-16 COUPLED PD CONTROLLED JOINT ANGLE ERRORS WITH MODELED DISTURBANCES

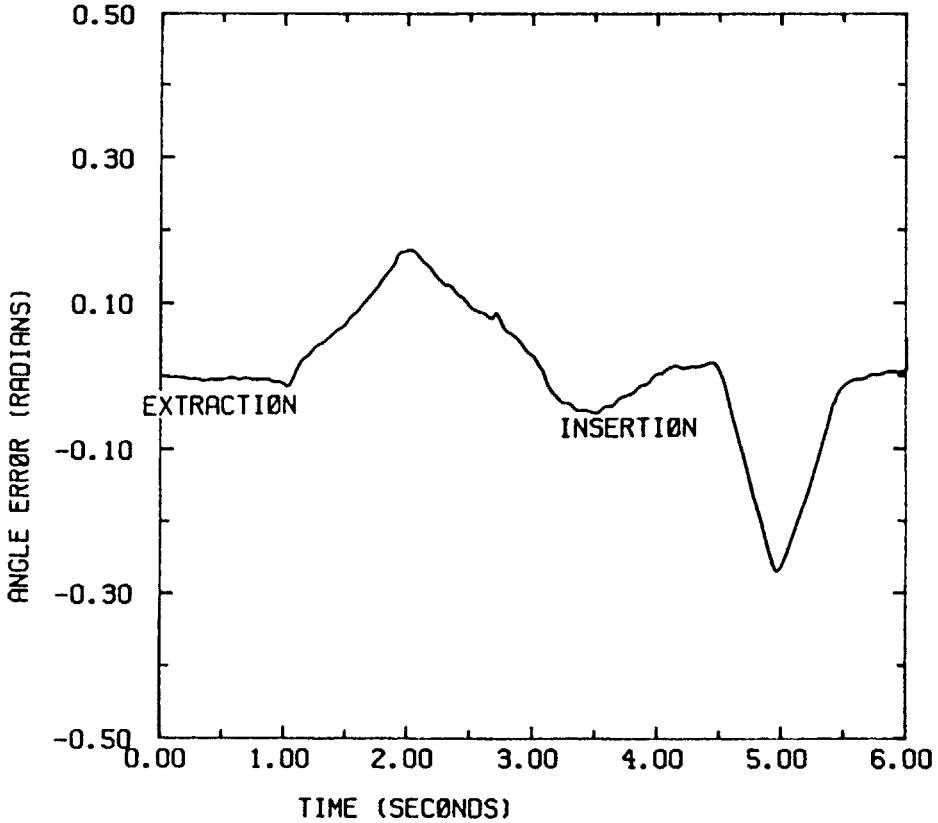

FIGURE 5-17 COUPLED PD CONTROLLED PAYLOAD ANGLE ERROR WITH MODELED DISTURBANCES

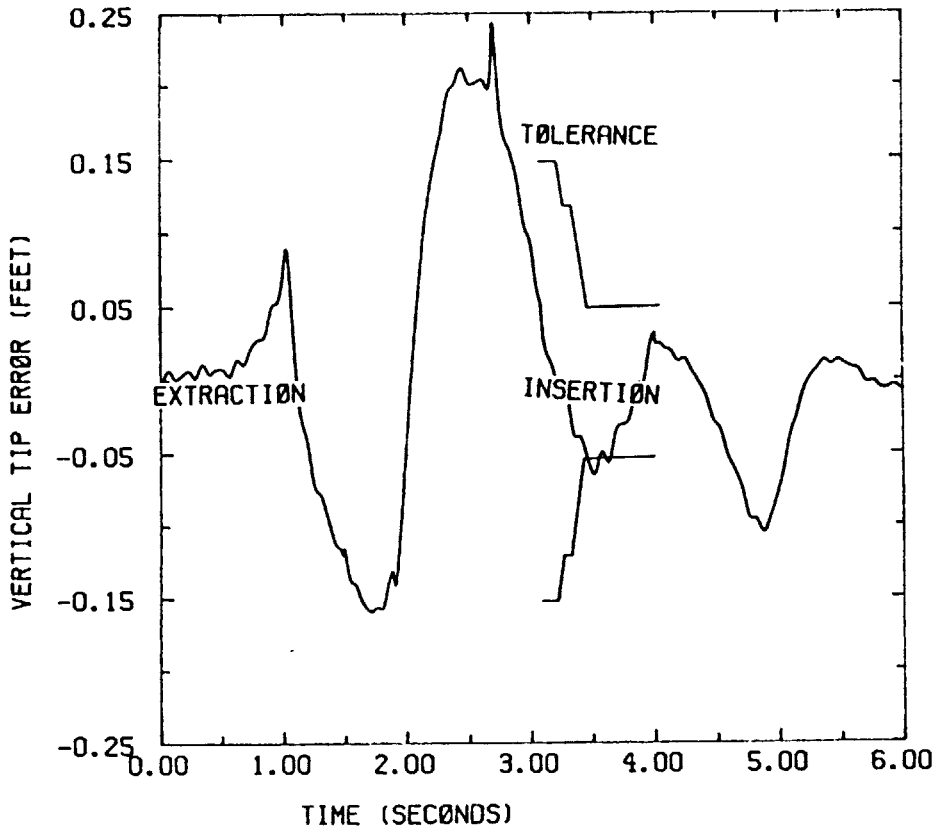

FIGURE 5-18 COUPLED PD CONTROLLED PAYLOAD VERTICAL TIP ERROR WITH MODELED DISTURBANCES

This chapter demonstrates the adverse effect that base motion disturbances have on a mobile robot. The designed controllers, both uncoupled and coupled PD, were tested with similiar results. The forces generated by base disturbances cause the robot to leave its prescribed path, resulting in reduced accuracy. Improved accuracy is necessary for mobile robots to be effective during movement. Hence, some form of compensator is necessary to overcome the adverse effects of base motion disturbances.

CHAPTER 6

COMPENSATOR DESIGN

6.1 NEED FOR COMPENSATOR

The results of simulations with disturbances, presented in Chapter 5, demonstrate that the PD controller is unable to effectively compensate for base accelerations. This result supports the conclusions obtained by Rick Lynch [3]. His research investigated the ability of conventional and advanced controllers to effectively compensate for vertical base accelerations. The conclusions from this work are that these control strategies are unable to provide good performance without raising the controller gains very high. Some form of compensator is then necessary which utilizes sensory information of the base motion. A compensator which uses sensory information of base motion can counter the effects of the dynamic forces generated by the disturbances. A controller, with this sensory compensator, should then have good performance with the base moving, and still have low enough gains to not operate in the vicinity of structural resonances.

6.2 COMPENSATOR DESIGN

Base motion disturbances create dynamic forces in the robotic loader system. Investigation of the equations of motion reveals the disturbance terms which cause these forces are Θ_b, $\dot{\Theta}_b$, $\ddot{\Theta}_b$, and $\ddot{Y}$. The disturbance terms for

each joint are shown in equations 6.1, 6.2, and 6.3.

JOINT 1:

$$[(M_1 l_1 + M_2 L_1 + M_p L_1)\cos\theta_1]\ddot{Y}$$
$$[(M_1 L_b l_1 + M_2 L_b L_1 + M_p L_b L_1)\cos(\theta_b-\theta_1)]\ddot{\theta}_b$$
$$[(M_1 L_b l_1 + M_2 L_b L_1 + M_p L_b L_1)\sin(\theta_b-\theta_1)]\dot{\theta}_b^2 \qquad (6.1)$$

JOINT 2:

$$[(M_2 l_2 + M_p L_2)\cos\theta_2]\ddot{Y}$$
$$[(M_2 L_b l_2 + M_p L_b L_2)\cos(\theta_b-\theta_2)]\ddot{\theta}_b$$
$$[(M_2 L_b l_2 + M_p L_b L_2)\sin(\theta_b-\theta_2)]\dot{\theta}_b^2 \qquad (6.2)$$

JOINT 3:

$$[M_p l_p \cos\theta_w]\ddot{Y}$$
$$[M_p L_b l_p \cos(\theta_b-\theta_w)]\ddot{\theta}_b$$
$$[M_p L_b l_p \sin(\theta_b-\theta_w)]\dot{\theta}_b^2 \qquad (6.3)$$

A sensory, feedforward compensator would receive measurements obtained from sensors on the robot's base. It is assumed that these sensors measure all disturbance terms, θ_b, $\dot{\theta}_b$, $\ddot{\theta}_b$, and $\ddot{Y}$, which are then used to calculate the disturbance generated forces using modeled parameteric information of the system. These calculated torques would counter the effects of the base disturbances by being added to the controller torques for the prescribed motion. This procedure should be effective within the actuator torque limits and the speed of the compensator to receive sensory

information and conduct the disturbance calculations.

6.3 COMPENSATOR RESULTS

Simulations of the full non-linear system are again conducted using the uncoupled PD controller designed in Chapter 4, which failed the performance specifications when base motion disturbances were added in Chapter 5. These next simulations assume full knowledge of the base motion terms, θ_b, $\dot{\theta}_b$, $\ddot{\theta}_b$, and $\ddot{Y}$, which are obtained through sensors. Additionally, the parameter model of the robotic loader system are assumed to be accurate. The results of this simulation indicate that good performance is obtained and that all performance specifications are achieved. This performance is comparable to that obtained by this controller when the base is stationary.

Figure 6-1 shows the robot motion successfully load the ammunition even while the tank is experiencing the indicated motion. Figure 6-2, shows that the actuator torques are greater than those required by the stationary base simulation, figure 4-11, but do not reach saturation. Figures 6-3 and 6-4 show that the joint accuracy is similiar to figures 4-12 and 4-13, when the tank is stationary. Figures 6-5 and 6-6 show that the payload is within the allowable tolerances during insertion and that the end effector is aligned to grasp a new round at the end of the baseline trajectory.

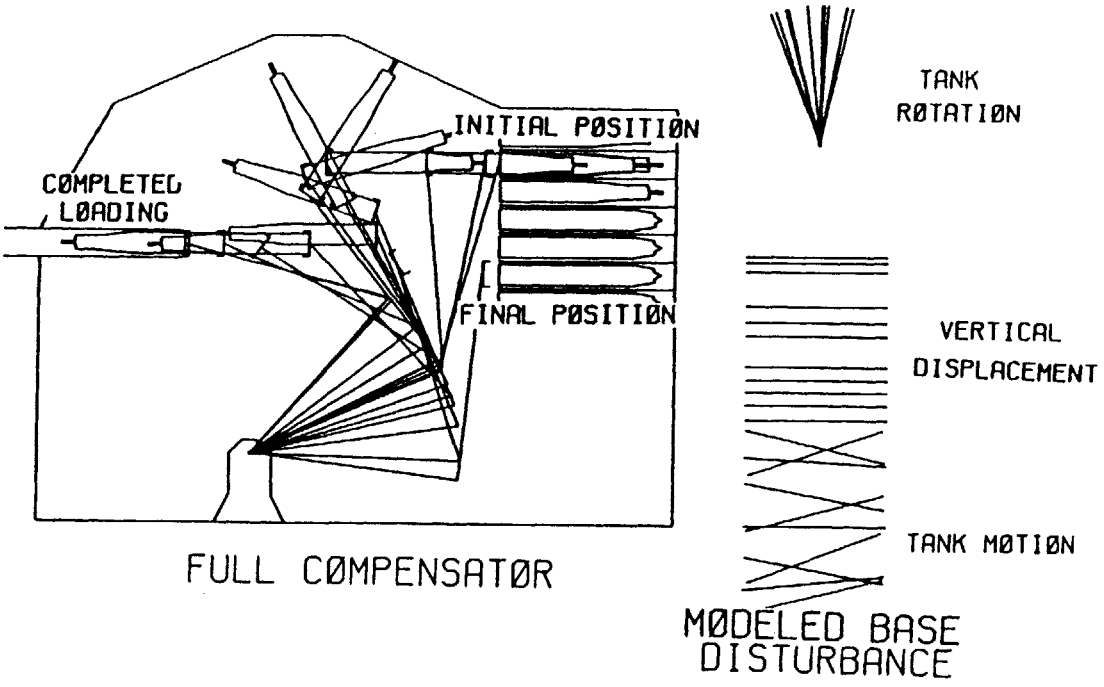

FIGURE 6-1 PD CONTROLLED ROBOT MOTION
WITH FULL COMPENSATOR

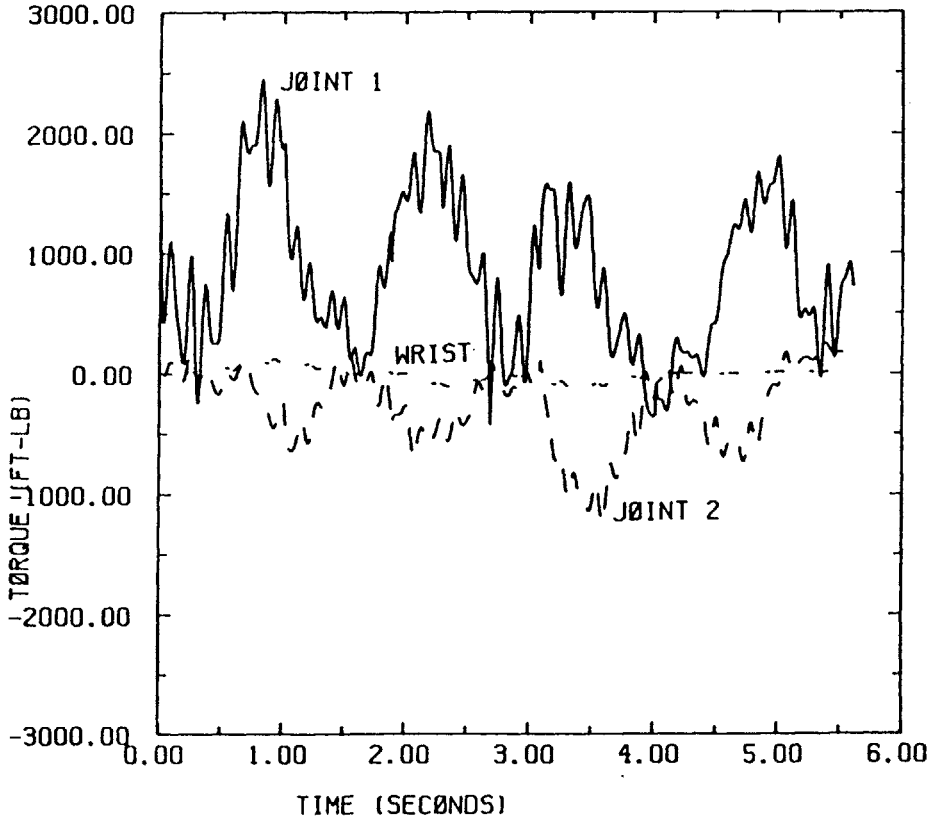

FIGURE 6-2 PD CONTROLLED MOTOR TORQUES
WITH FULL COMPENSATOR

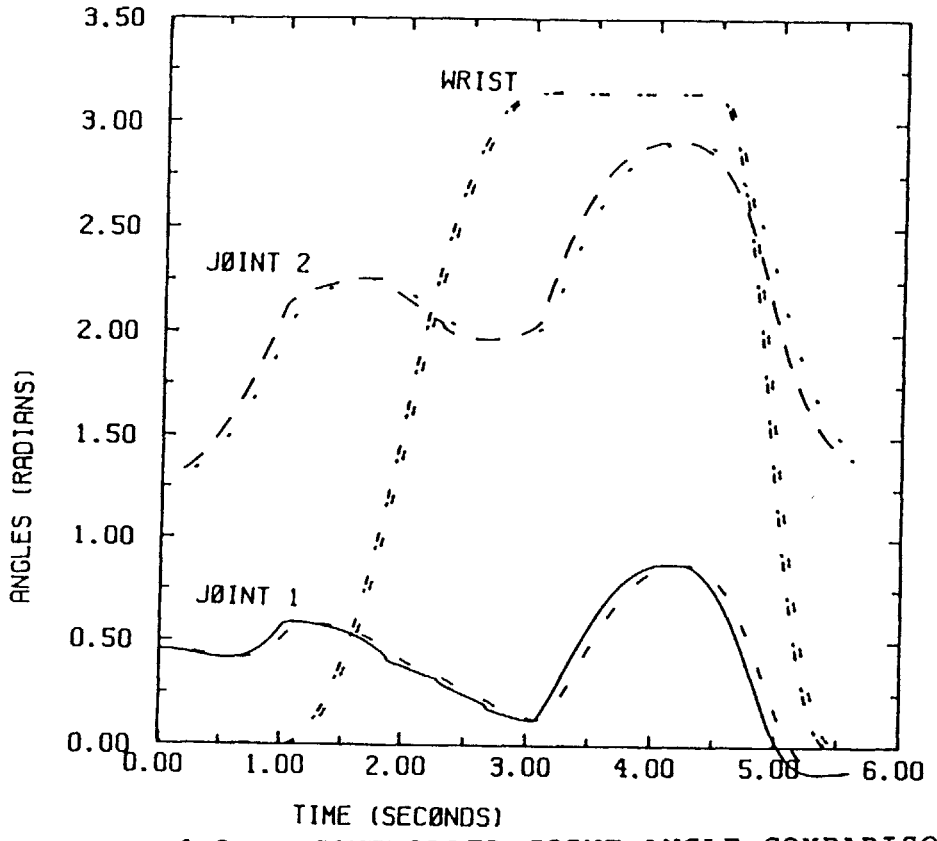

FIGURE 6-3 PD CONTROLLED JOINT ANGLE COMPARISON WITH FULL COMPENSATOR

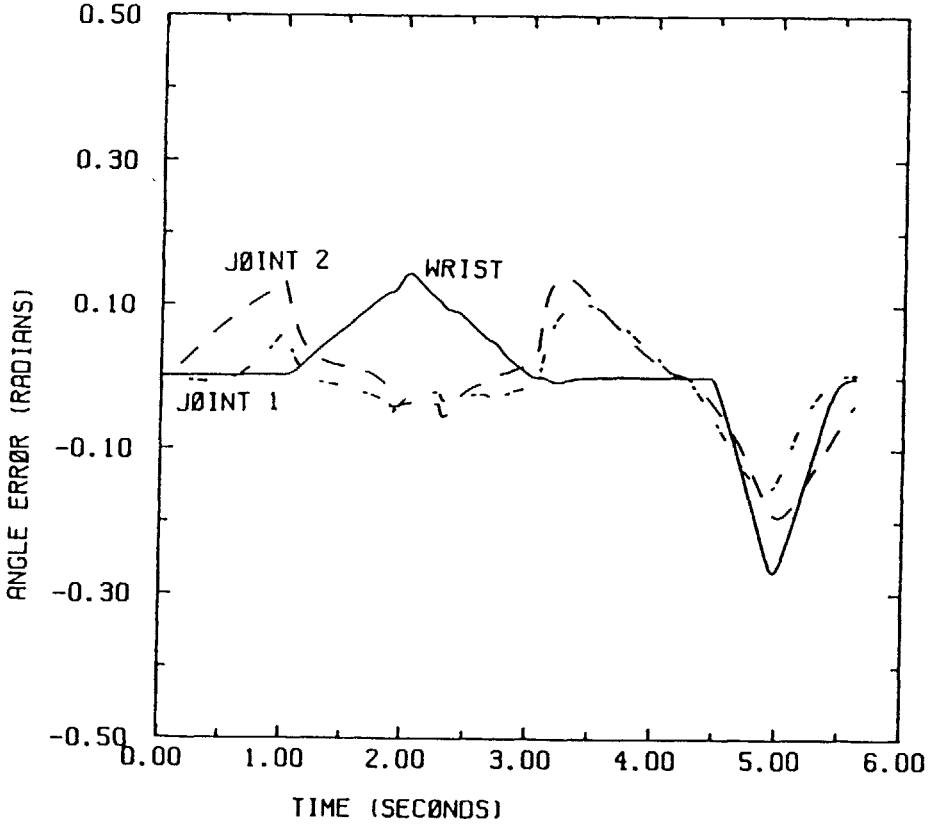

FIGURE 6-4 PD CONTROLLED JOINT ANGLE ERRORS WITH FULL COMPENSATOR

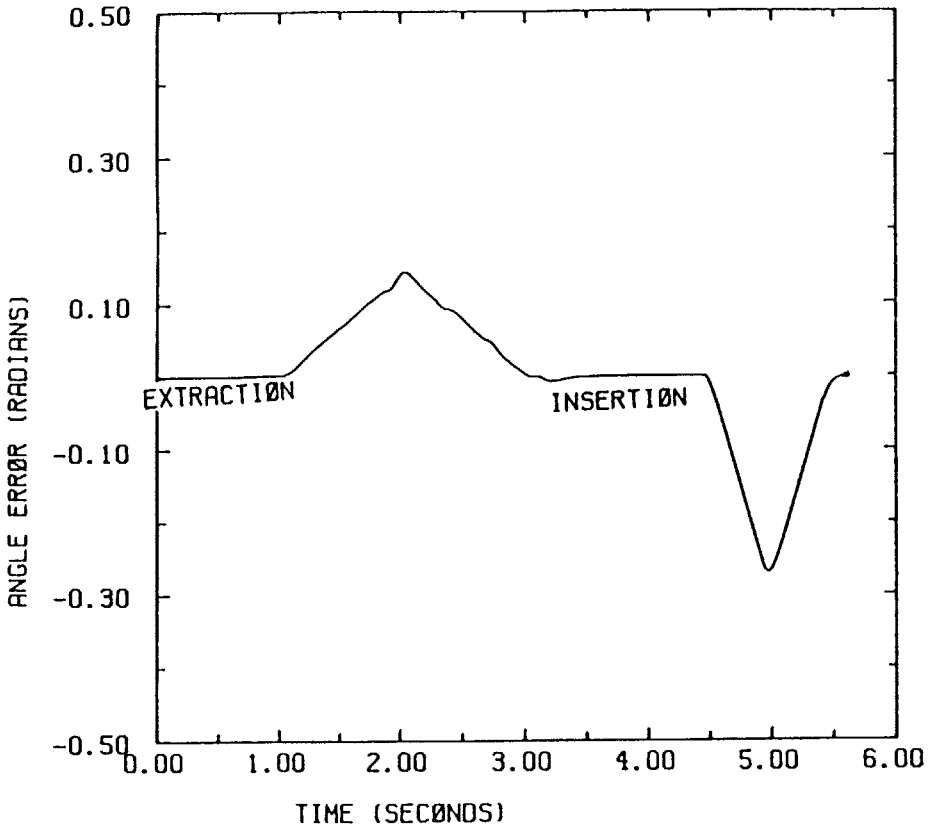

FIGURE 6-5 PD CONTROLLED PAYLOAD ANGLE ERROR WITH FULL COMPENSATOR

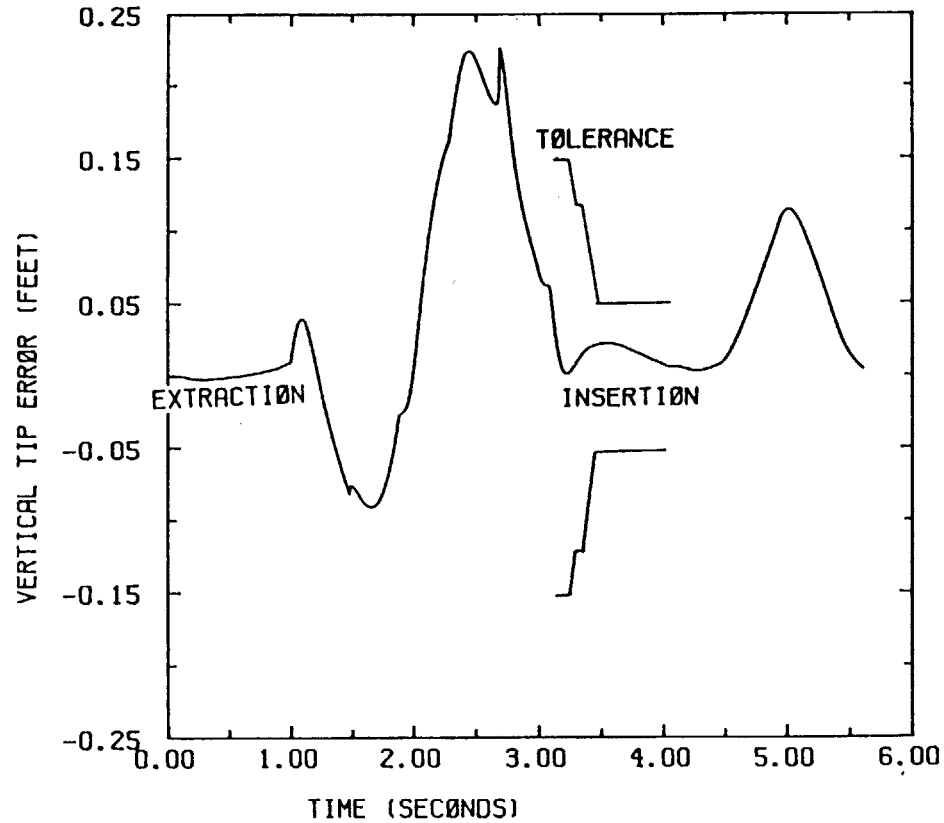

FIGURE 6-6 PD CONTROLLED PAYLOAD VERTICAL TIP ERROR WITH FULL COMPENSATOR

6.4 PARTIAL COMPENSATOR

The previous simulations assumed full knowledge of all disturbance terms, obtained through sensors on the robot's base. Actually obtaining these measurements is difficult. Angular acceleration are difficult to measure and are subject to noise. Therefore, having all disturbance terms available to the compensator is unrealistic. The disturbance terms, $\dot{\theta}_b$, $\ddot{\theta}_b$, and $\ddot{Y}$, create dynamic forces in the robotic system. The magnitude of these forces, generated by the disturbances, are calculated for the duration of the baseline trajectory and plotted in figure 6-7, 6-8, and 6-9 for joints 1, 2, and the wrist respectively. These figures show the relative magnitude of the disturbance forces generated by the disturbance terms for each joint. Figures 6-7, 6-8, and 6-9 reveals that the $\ddot{Y}$ terms are larger by a factor of four than either of the rotational disturbance terms, $\dot{\theta}_b$ or $\ddot{\theta}_b$. The magnitude of the vertical acceleration term, Y, indicates that this measurement is essential for good performance.

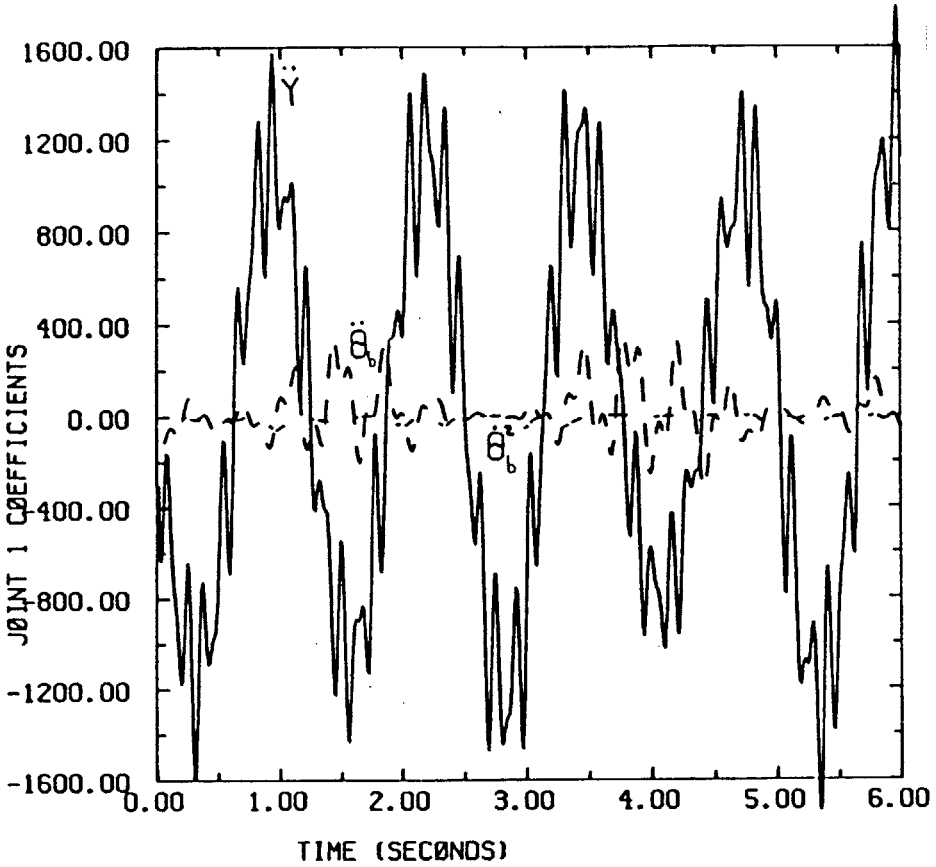

FIGURE 6-7 MAGNITUDE OF JOINT 1 DISTURBANCE TERMS

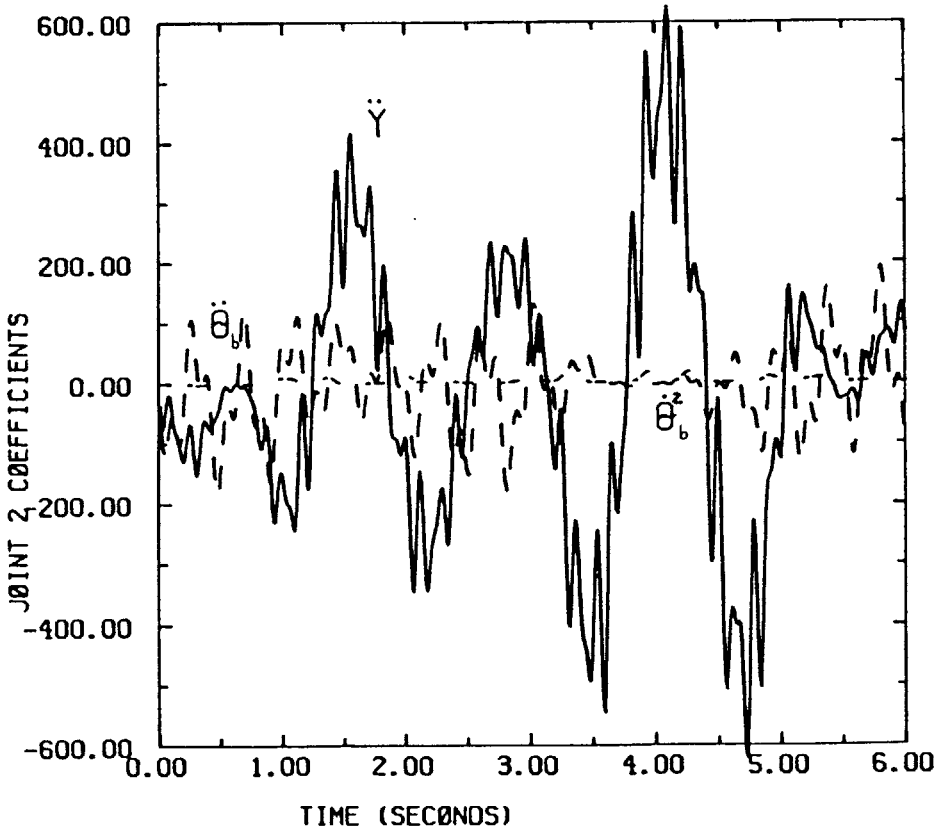

FIGURE 6-8 MAGNITUDE OF JOINT 2 DISTURBANCE TERMS

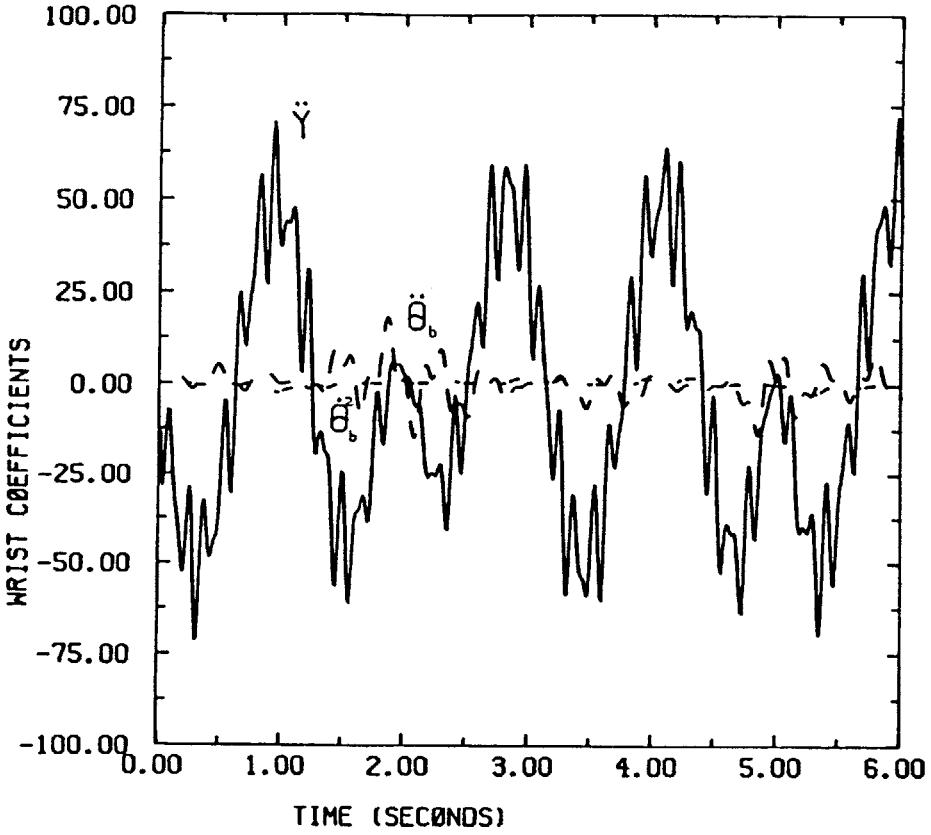

FIGURE 6-9 MAGNITUDE OF JOINT 3 DISTURBANCE TERMS

Tanks currently employ main gun stabilization systems to improve firing accuracy when the tank is moving. The stabilization system maintains the main gun stationary relative to inertial space during tank yaw rotations and turret pitch rotations. This main gun stabilization system obtains turret pitch velocity measurements, which is $\dot{\theta}_b$, from rate gyros and integrates these measurements to obtain the angle, which is θ_b, necessary to keep the gun elevation constant. Therefore, two of the disturbance terms, θ_b and $\dot{\theta}_b$, are available to the compensator from the stabilization system. The disturbance measurement which is unavailable and difficult to obtain is the rotational acceleration.

Simulations are conducted utilizing only some disturbance term measurements, $\ddot{Y}$, θ_b, and $\dot{\theta}_b$, and satisfactory performance is still achieved. This performance is shown in figures 6-10, 6-11, 6-12, 6-13, 6-14, and 6-15. The accuracy is only slightly less than that obtained with the full compensator, assuming complete base sensory information.

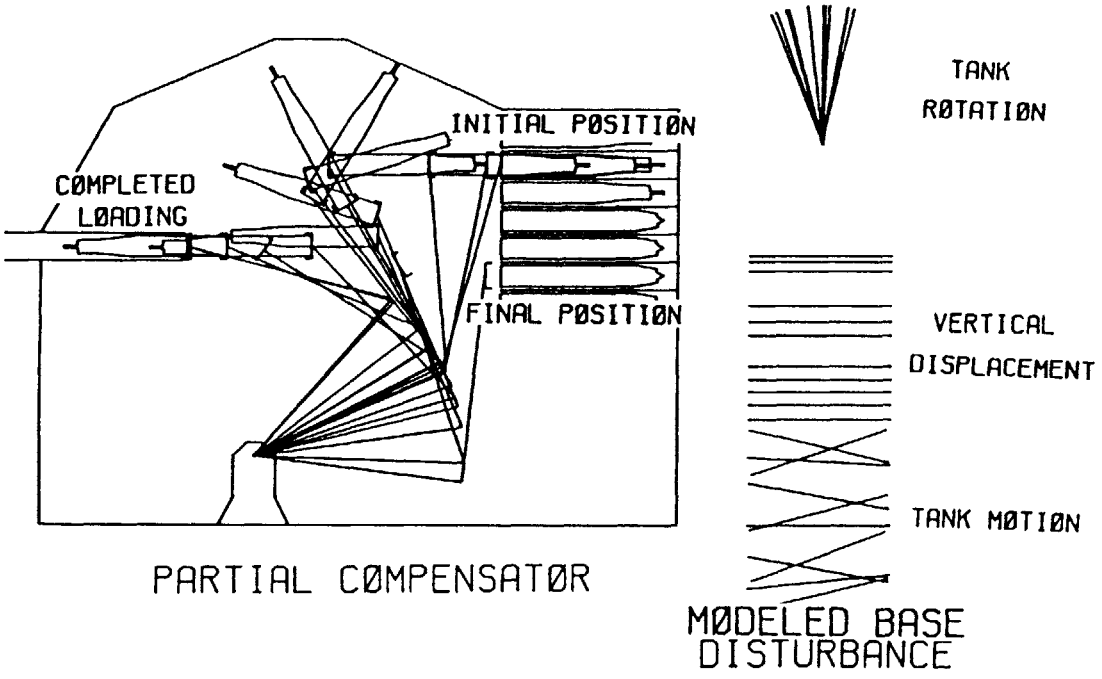

FIGURE 6-10 PD CONTROLLED ROBOT MOTION
WITH PARTIAL COMPENSATOR

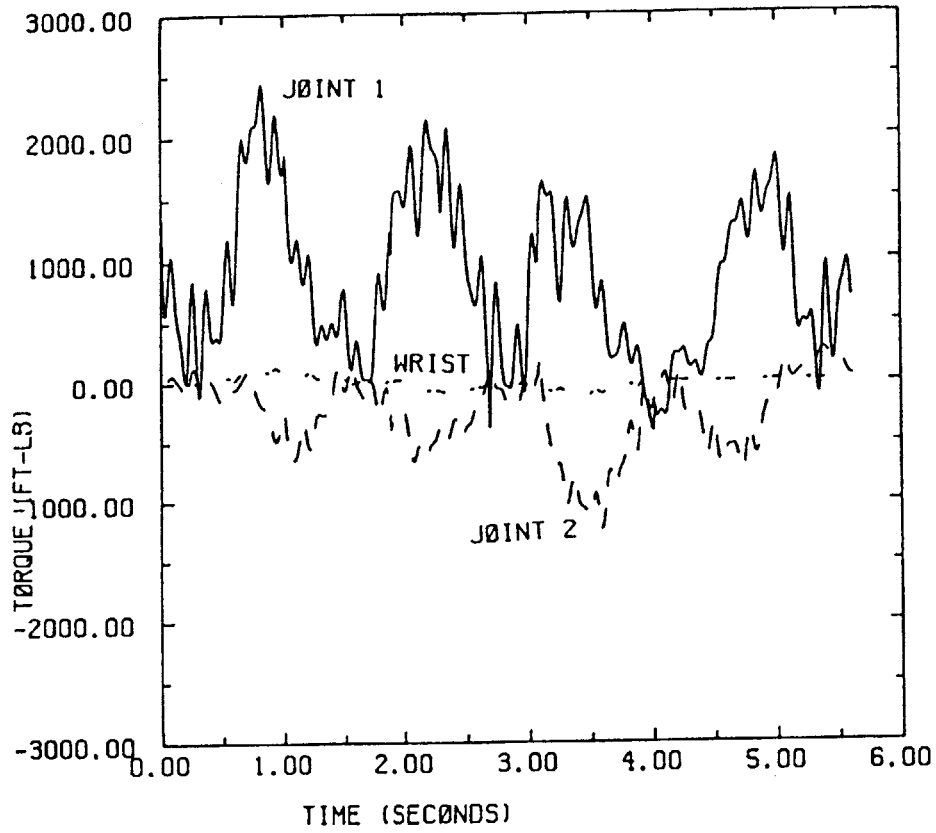

FIGURE 6-11 PD CONTROLLED MOTOR TORQUES
WITH PARTIAL COMPENSATOR

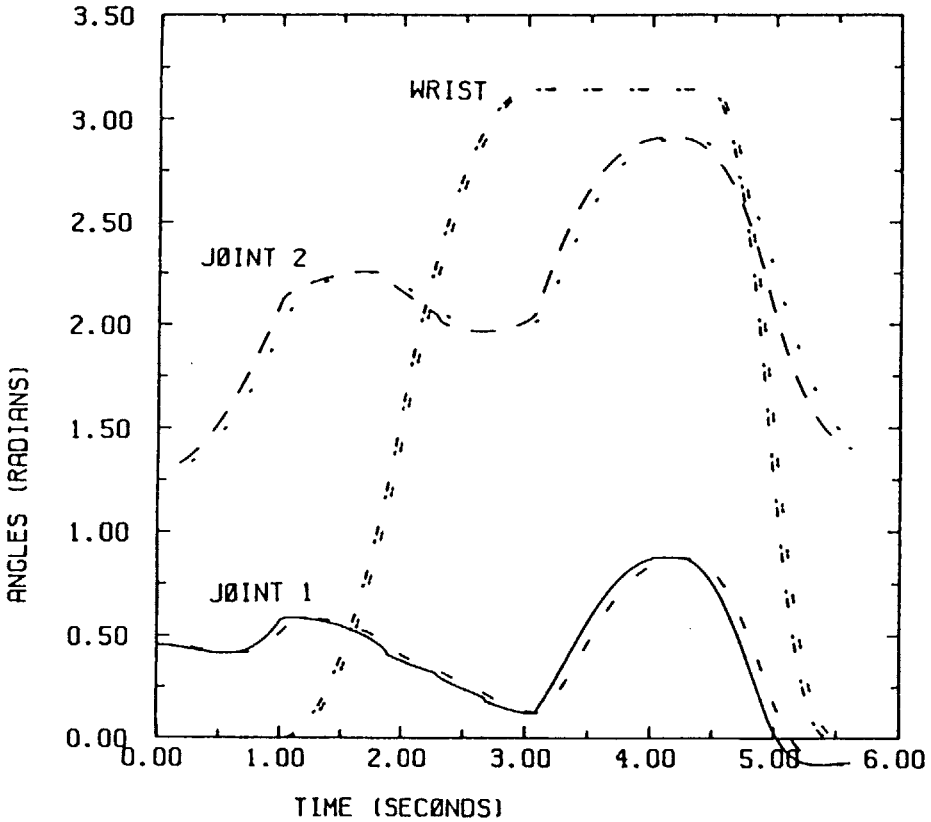

FIGURE 6-12 PD CONTROLLLED JOINT ANGLE COMPARISON
WITH PARTIAL COMPENSATOR

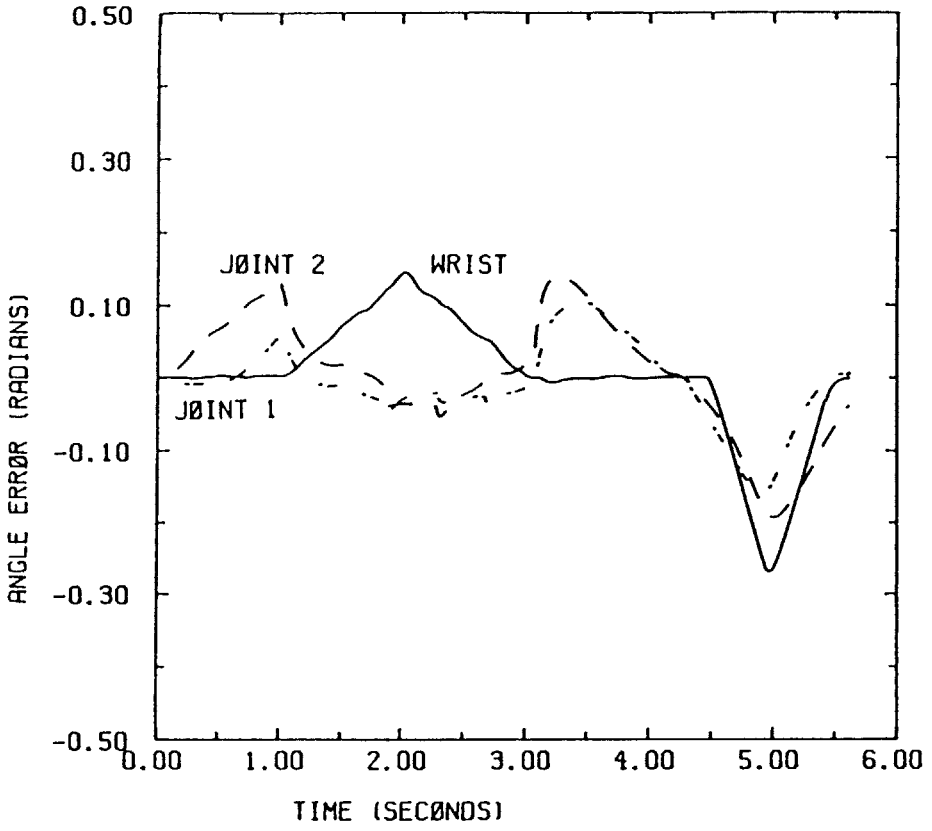

FIGURE 6-13 PD CONTROLLED JOINT ANGLE ERRORS WITH PARTIAL COMPENSATOR

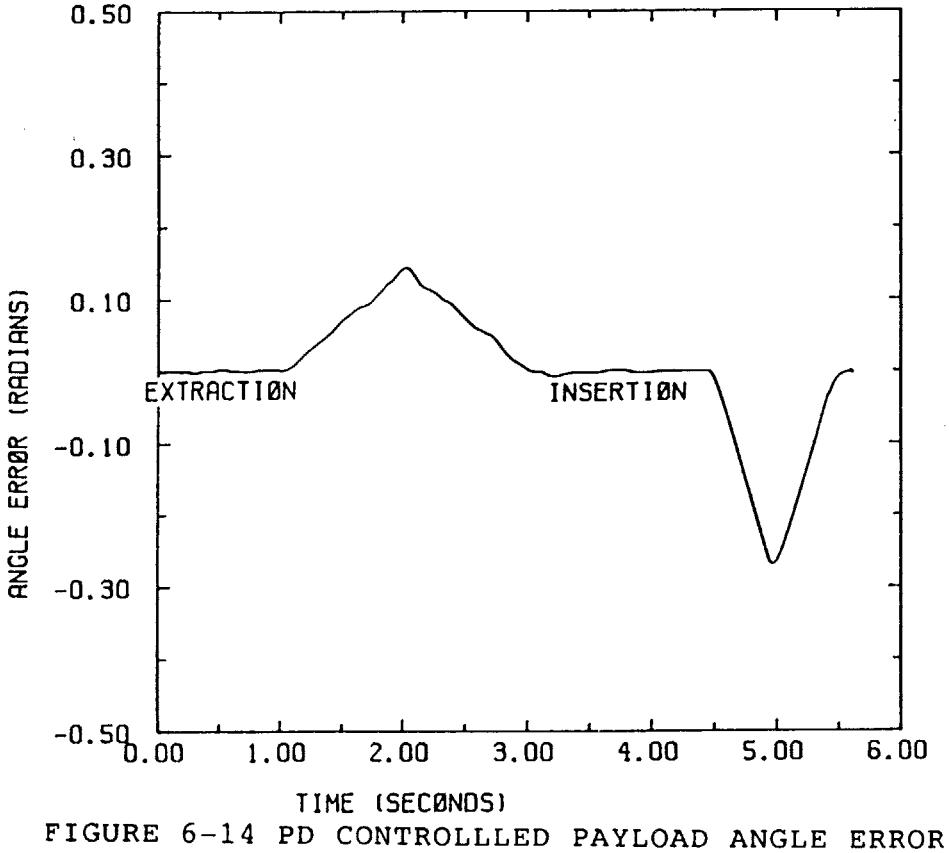

FIGURE 6-14 PD CONTROLLLED PAYLOAD ANGLE ERROR WITH PARTIAL COMPENSATOR

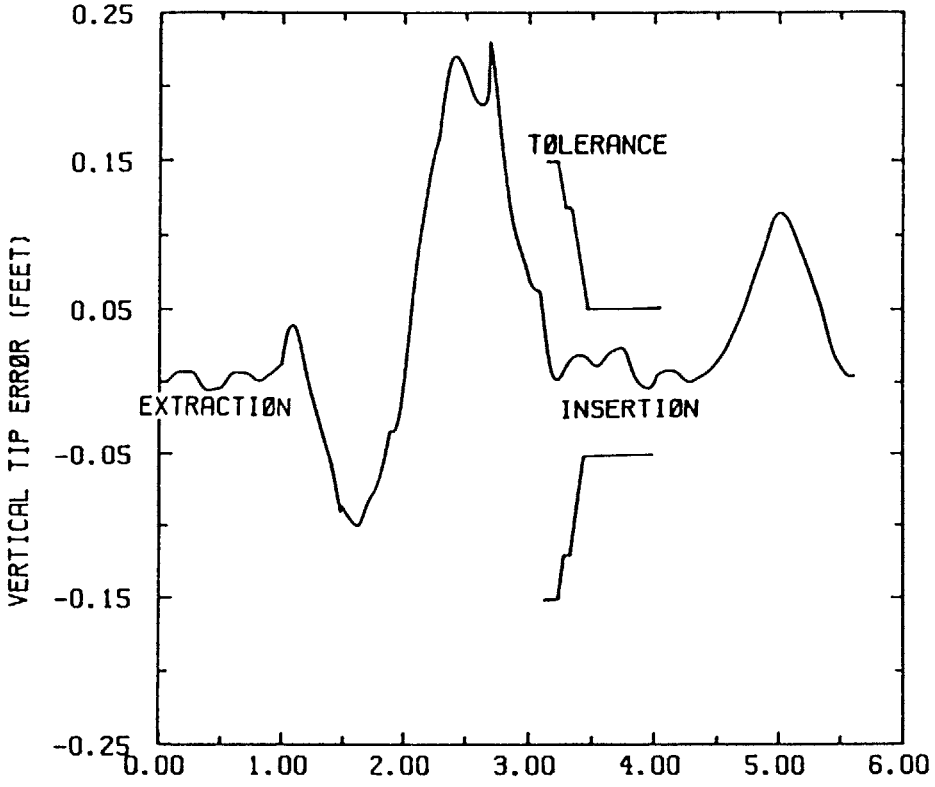

FIGURE 6-15 PD CONTROLLED PAYLOAD VERTICAL TIP ERROR WITH PARTIAL COMPENSATOR

This sensory compensator, which uses information of the base motion disturbances to calculate and feedforward a compensating torque, is effective in eliminating the joint accuracy errors caused by the modeled base motion disturbances. The achieved accuracy of the robotic system using the compensator, even when the base is subject to large disturbances, is as good as when the robotic system is stationary. The ability of the compensator to counter the actual tank disturbances, figures 2-10 and 2-12, is verified next. These results, shown in figures 6-16, 6-17, 6-18, 6-19, 6-20, and 6-21, confirm the compensator's ability to counter the disturbances generated by an actual tank moving over rough terrain.

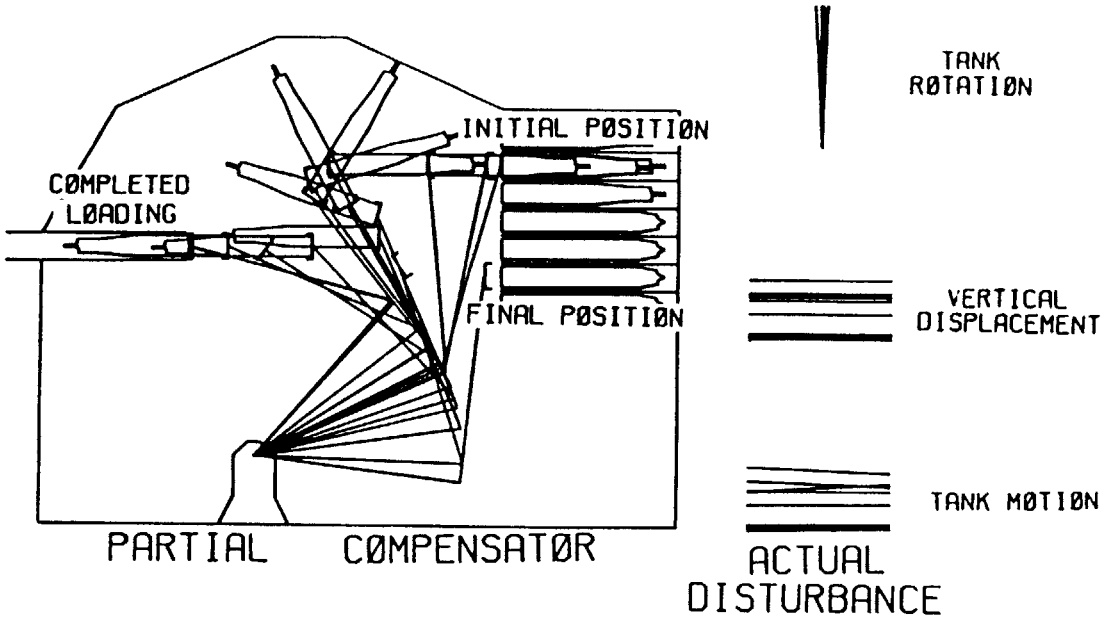

FIGURE 6-16 PD CONTROLLED ROBOT MOTION WITH PARTIAL COMPENSATOR DURING ACTUAL DISTURBANCES

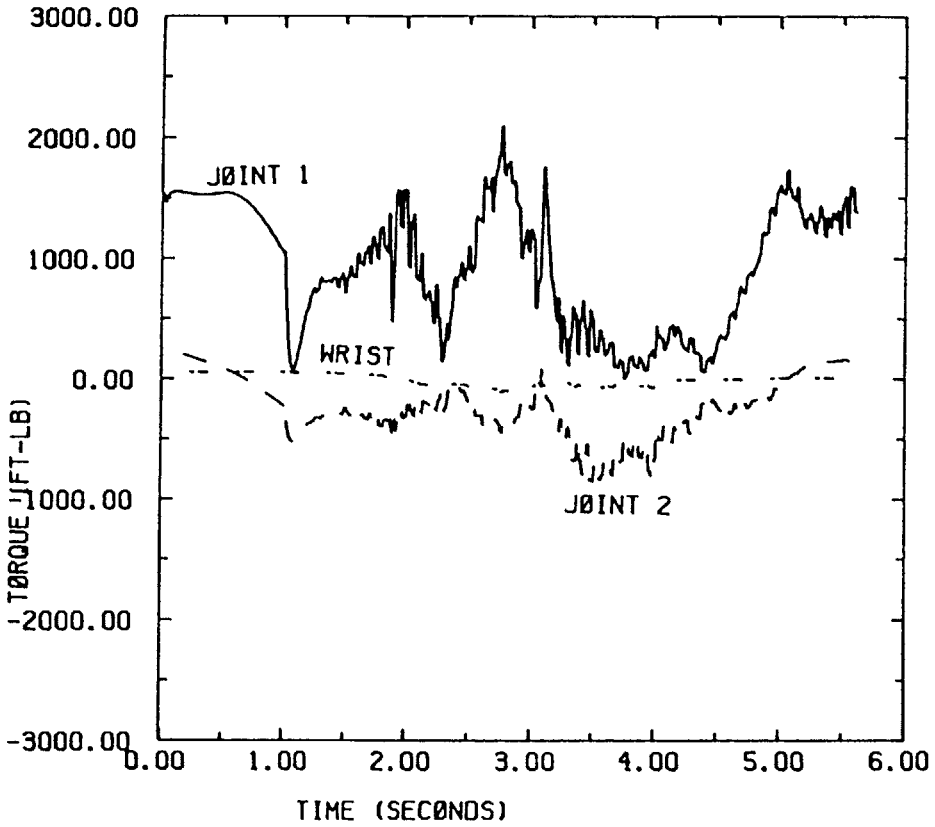

FIGURE 6-17 PD CONTROLLED MOTOR TORQUES WITH PARTIAL COMPENSATOR DURING ACTUAL DISTURBANCES

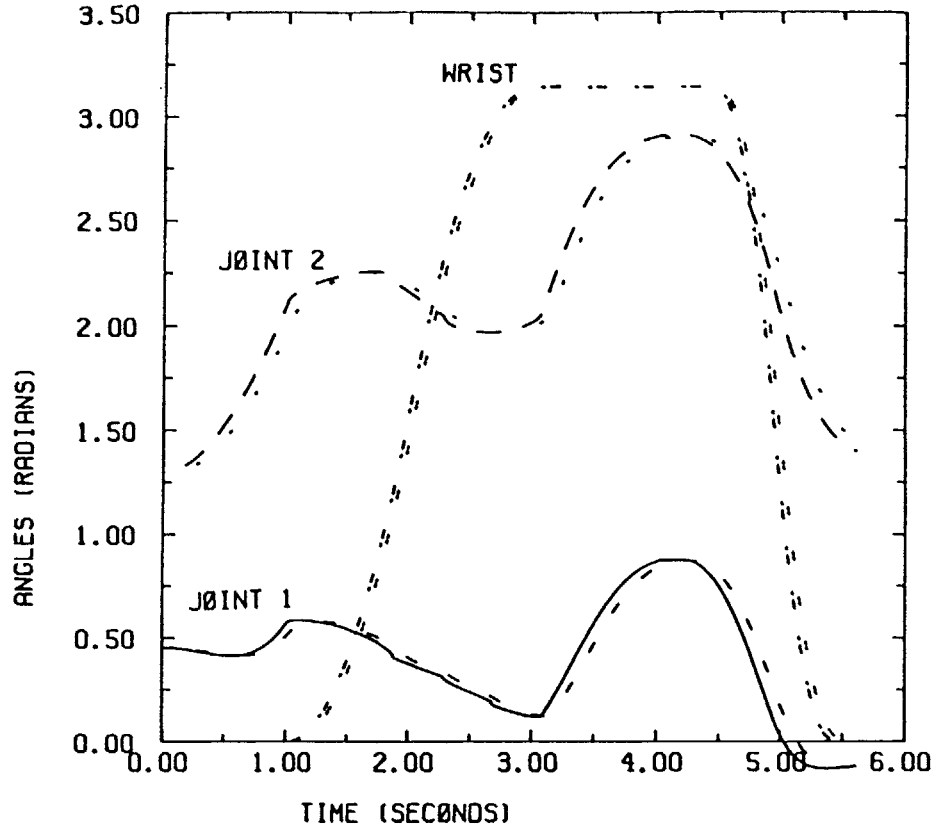

FIGURE 6-18 PD CONTROLLED JOINT ANGLE COMPARISON WITH PARTIAL COMPENSATOR DURING ACTUAL DISTURBANCES

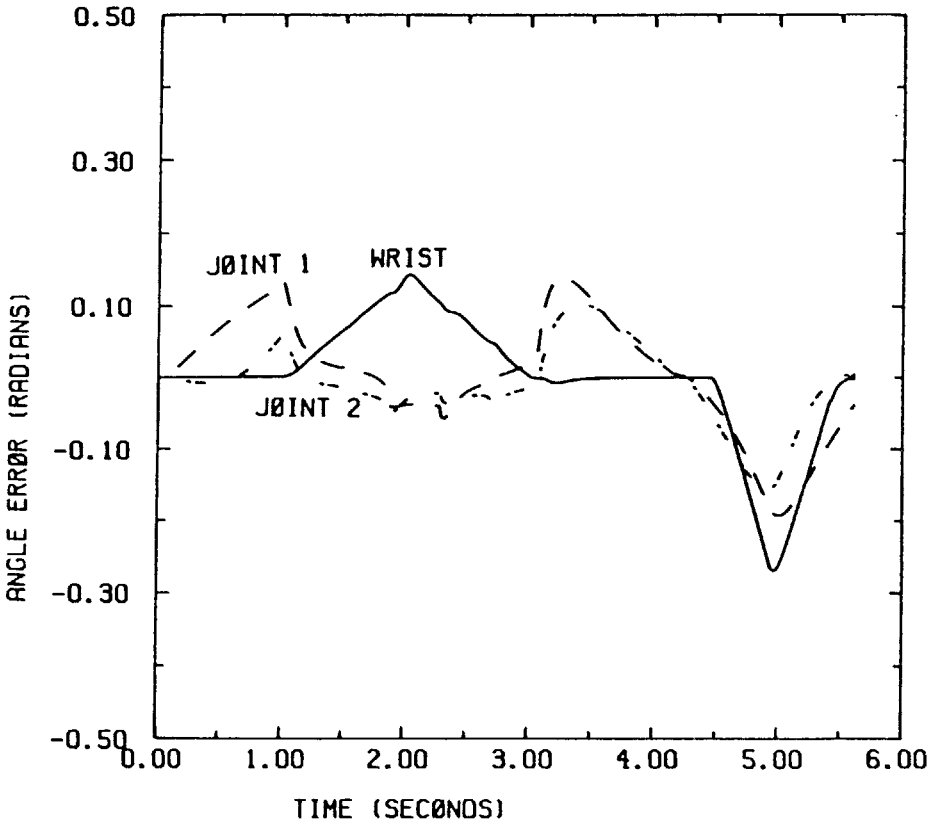

FIGURE 6-19 PD CONTROLLED JOINT ANGLE ERRORS WITH PARTIAL COMPENSATOR DURING ACTUAL DISTURBANCES

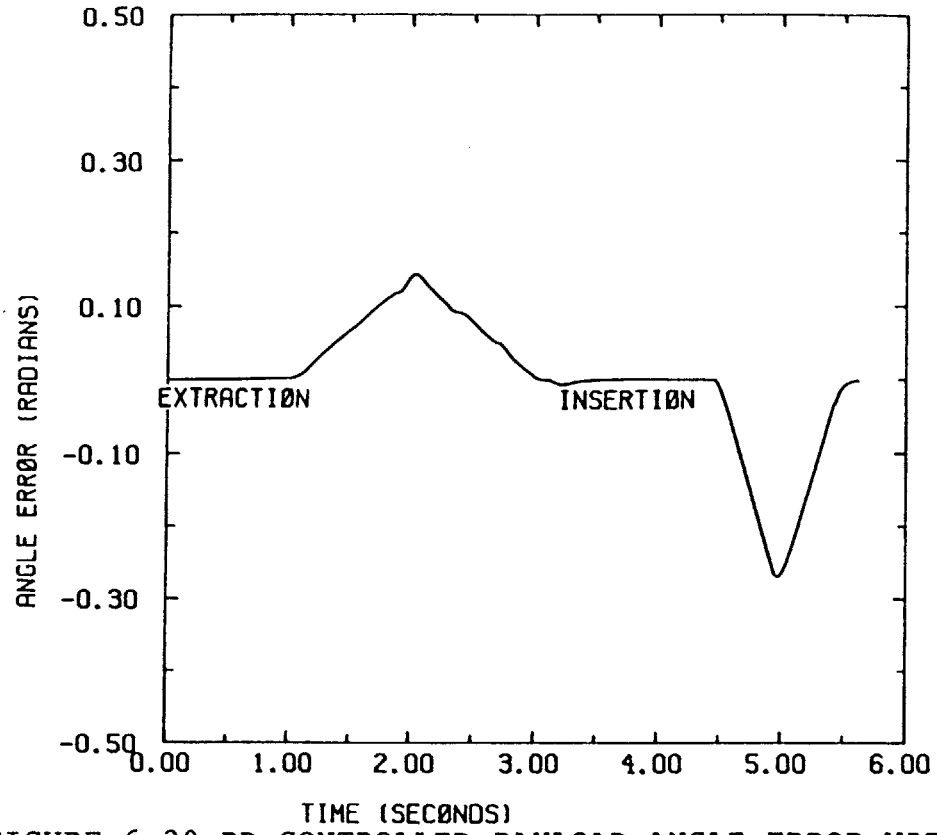

FIGURE 6-20 PD CONTROLLED PAYLOAD ANGLE ERROR WITH PARTIAL COMPENSATOR DURING ACTUAL DISTURBANCES

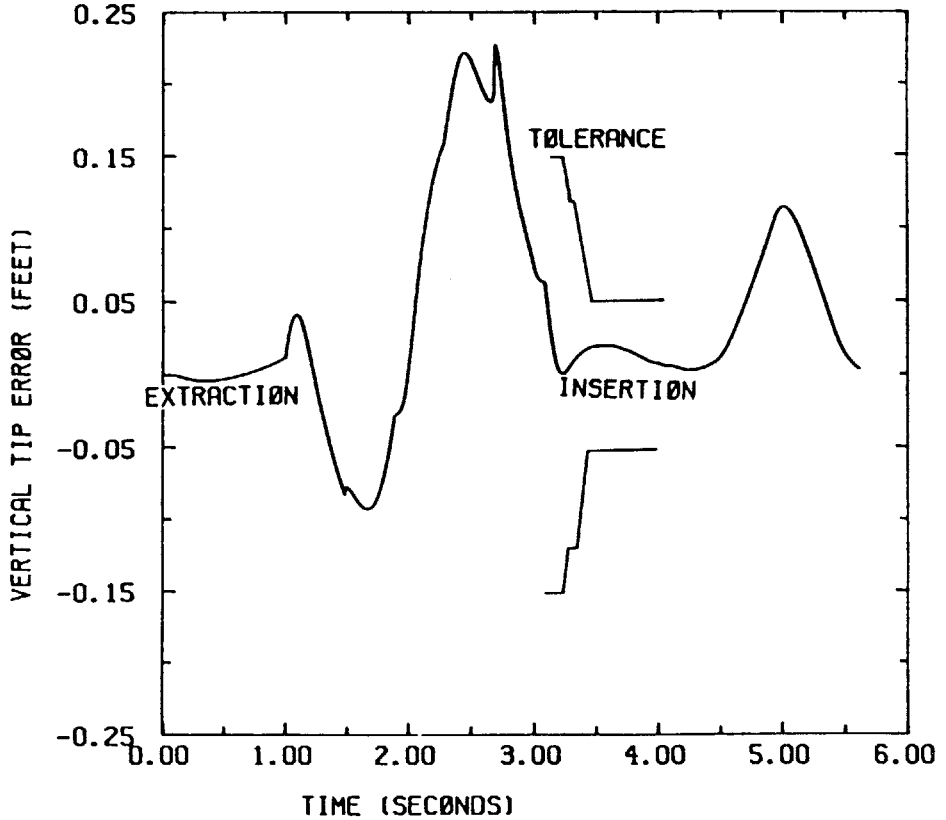

FIGURE 6-21 PD CONTROLLED PAYLOAD VERTICAL TIP ERROR WITH PARTIAL COMPENSATOR DURING ACTUAL DISTURBANCES

The previous simulations assumed an accurate model of the robotic loader system's parameteric values. This assumption is not always actually achieved. The calculated parameters of a system may be inaccurate since the actual values may vary slighty with different configurations. Additionally, some unmodeled characteristics may exist which influence the parameter model, causing inaccuracies. Therefore, the next simulation is conducted using the linear PD uncoupled controller with a partial compensator utilizing a parametric uncertainty of ten percent of the actual values. These results, shown in figures 6-22, 6-23, 6-24, 6-25, 6-26, and 6-27, demonstrate that the robotic loader still successfully loads the ammunition, despite the inaccuracies of the modeled parameters. This result shows the robustness of the system. The achieved accuracy is slightly less than that obtained for the accurate model, figures 6-13, 6-14, and 6-15, but the results are still within the allowable tolerances.

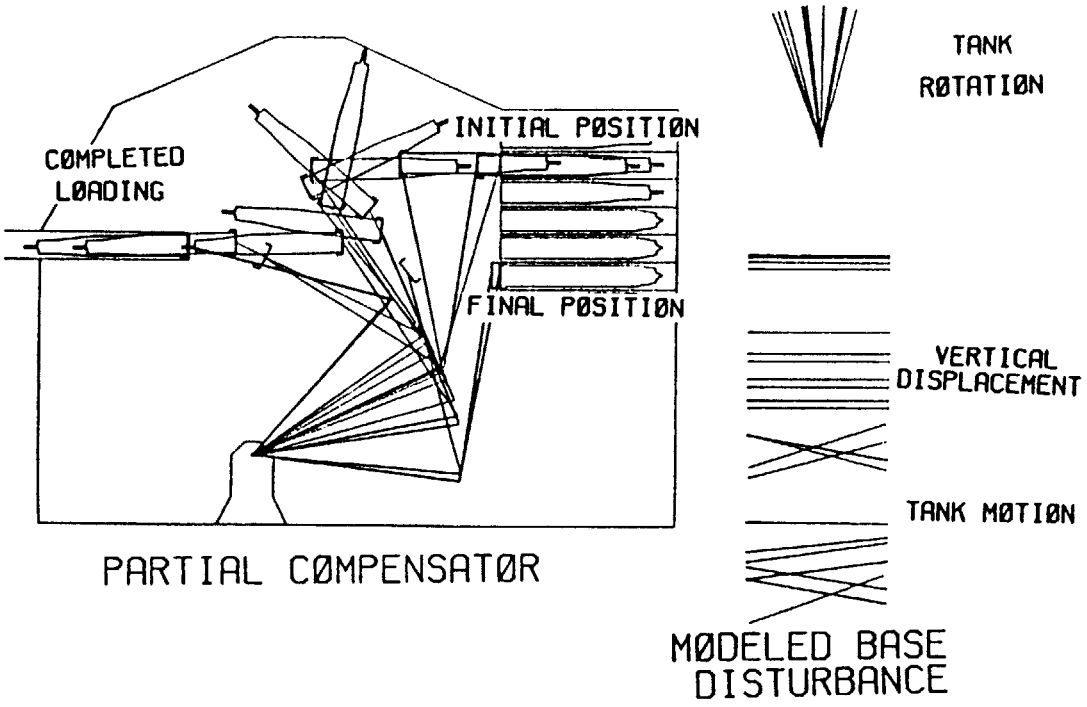

FIGURE 6-22 PD CONTROLLED ROBOT MOTION WITH PARTIAL COMPENSATOR USING PARAMETERIC UNCERTAINTY

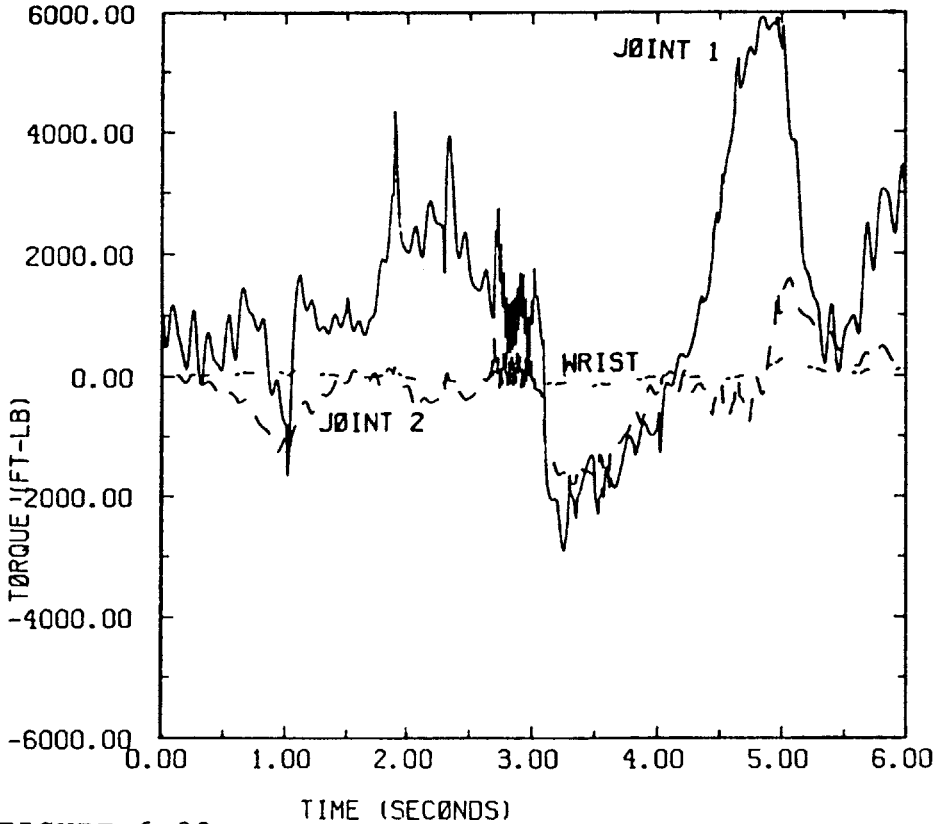

FIGURE 6-23 PD CONTROLLED MOTOR TORQUES WITH PARTIAL COMPENSATOR USING PARAMETRIC UNCERTAINTY

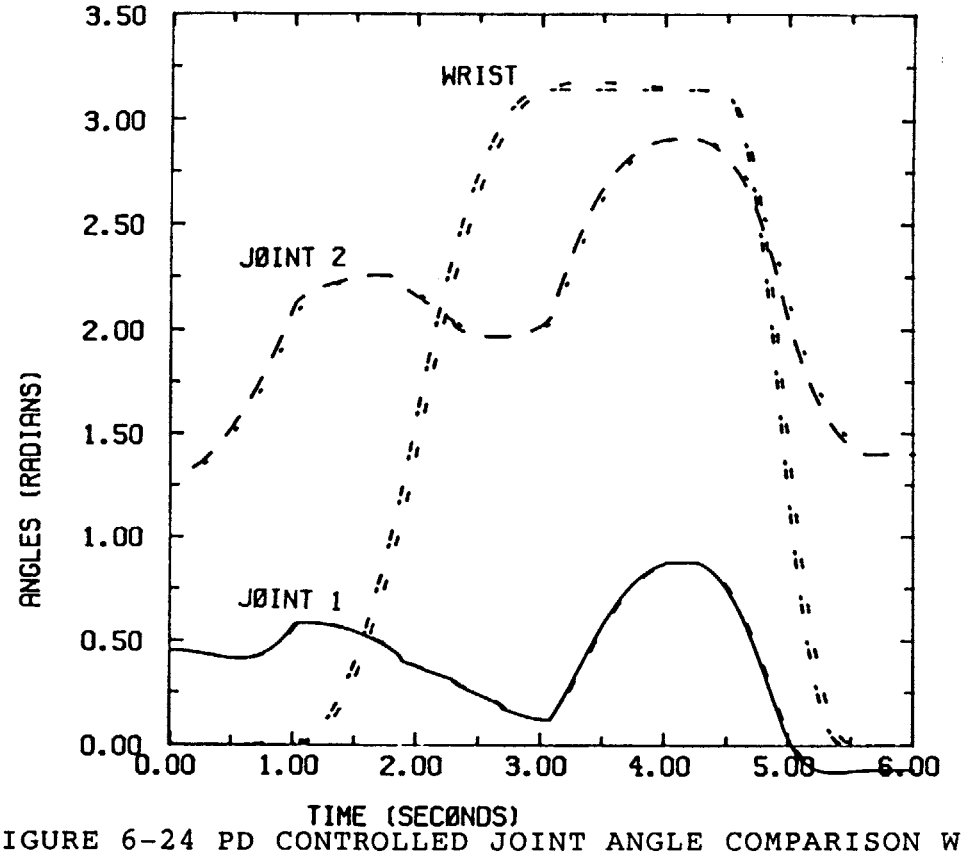

FIGURE 6-24 PD CONTROLLED JOINT ANGLE COMPARISON WITH PARTIAL COMPENSATOR USING PARAMETRIC UNCERTAINTY

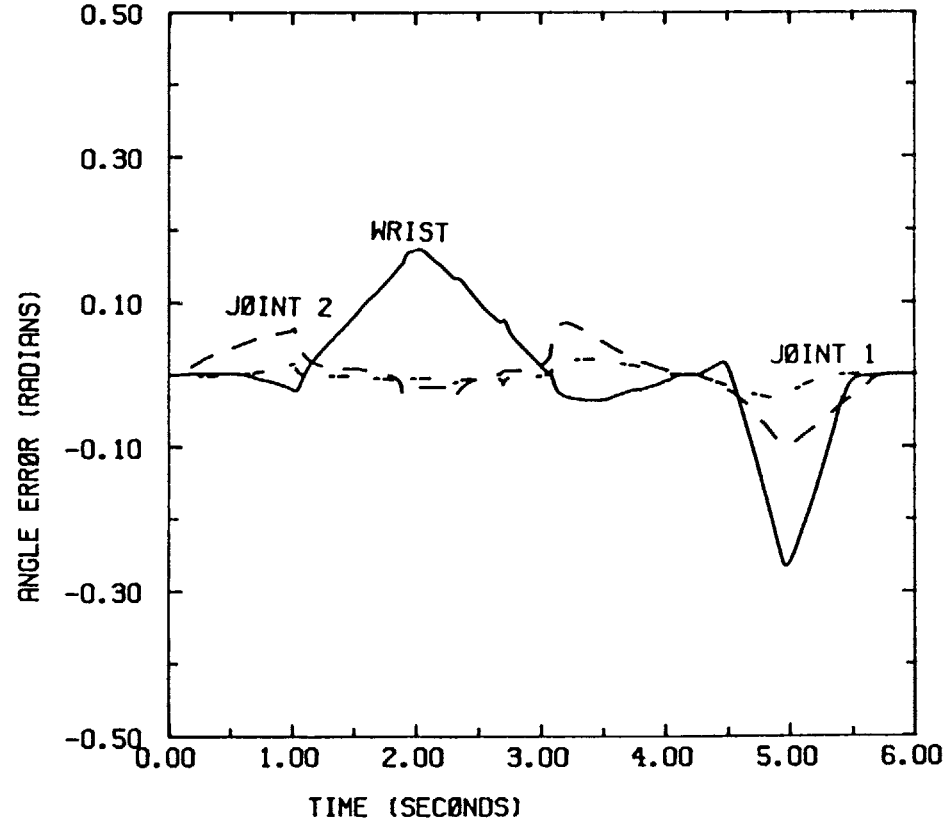

FIGURE 6-25 PD CONTROLLED JOINT ANGLE ERRORS WITH PARTIAL COMPENSATOR USING PARAMETRIC UNCERTAINTY

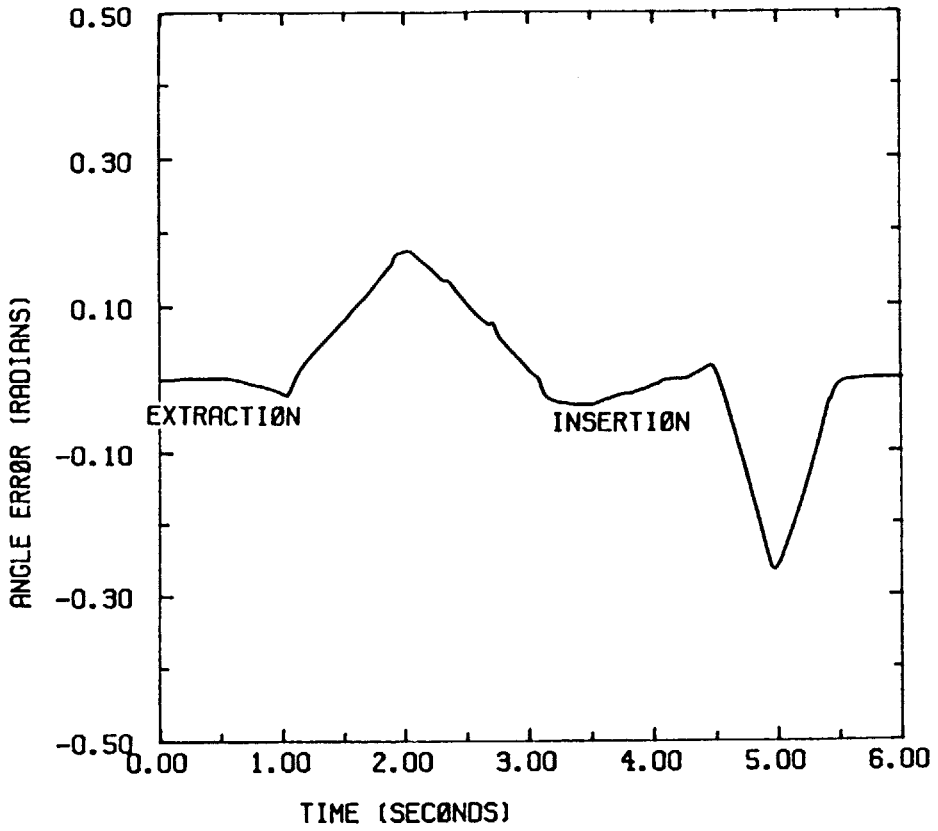

FIGURE 6-26 PD CONTROLLED PAYLOAD ANGLE ERROR WITH PARTIAL COMPENSATOR USING PARAMETRIC UNCERTAINTY

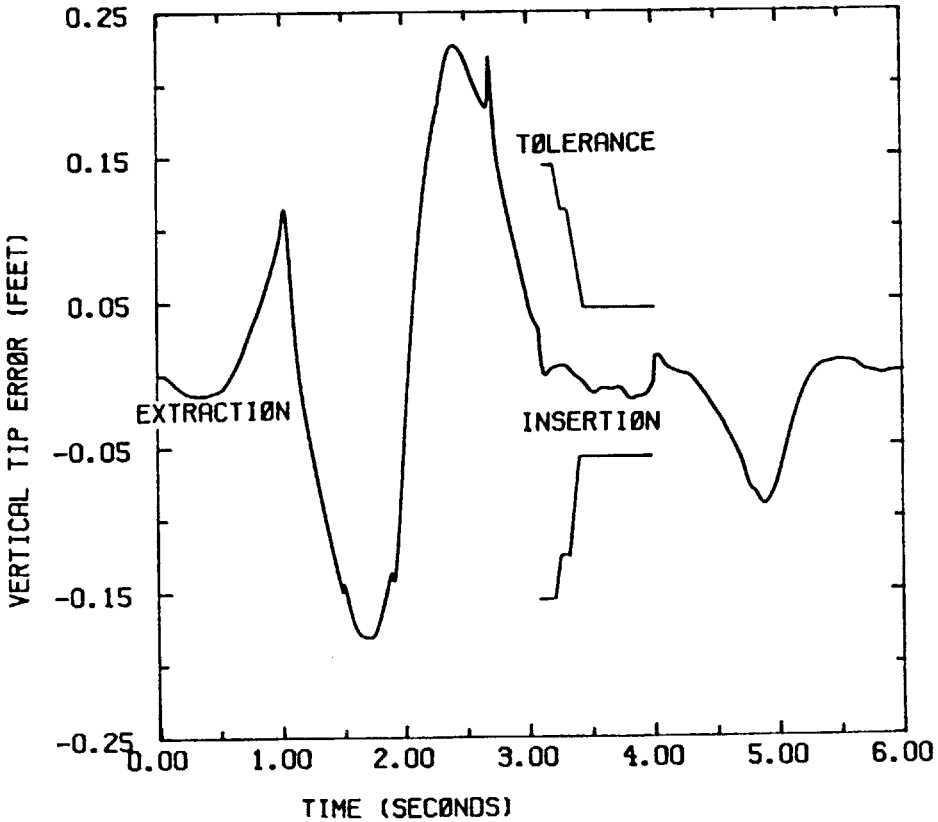

FIGURE 6-27 PD CONTROLLED PAYLOAD VERTICAL TIP ERROR WITH PARTIAL COMPENSATOR USING PARAMETRIC UNCERTAINTY

In Chapter 5, simulations show that the coupled controller is also ineffective in overcoming the dynamic disturbance forces. The compensator is now added to the coupled PD controller to test the combined performance. Simulations with the modeled disturbances imposed on the robotic system controlled with the PD coupled controller are conducted with the results shown in figures 6-22, 6-23, 6-24, 6-25, 6-26, and 6-27. These figures show the ability of the compensator to counter the disturbance forces.

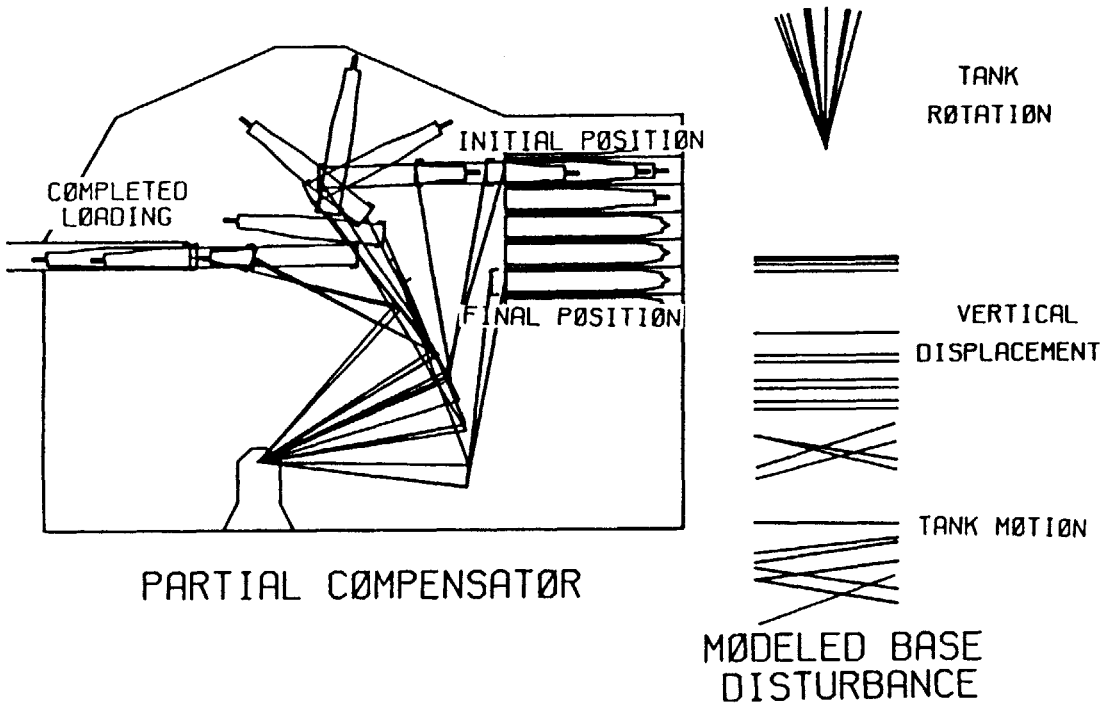

FIGURE 6-28 COUPLED PD CONTROLLED ROBOT MOTION WITH PARTIAL COMPENSATOR

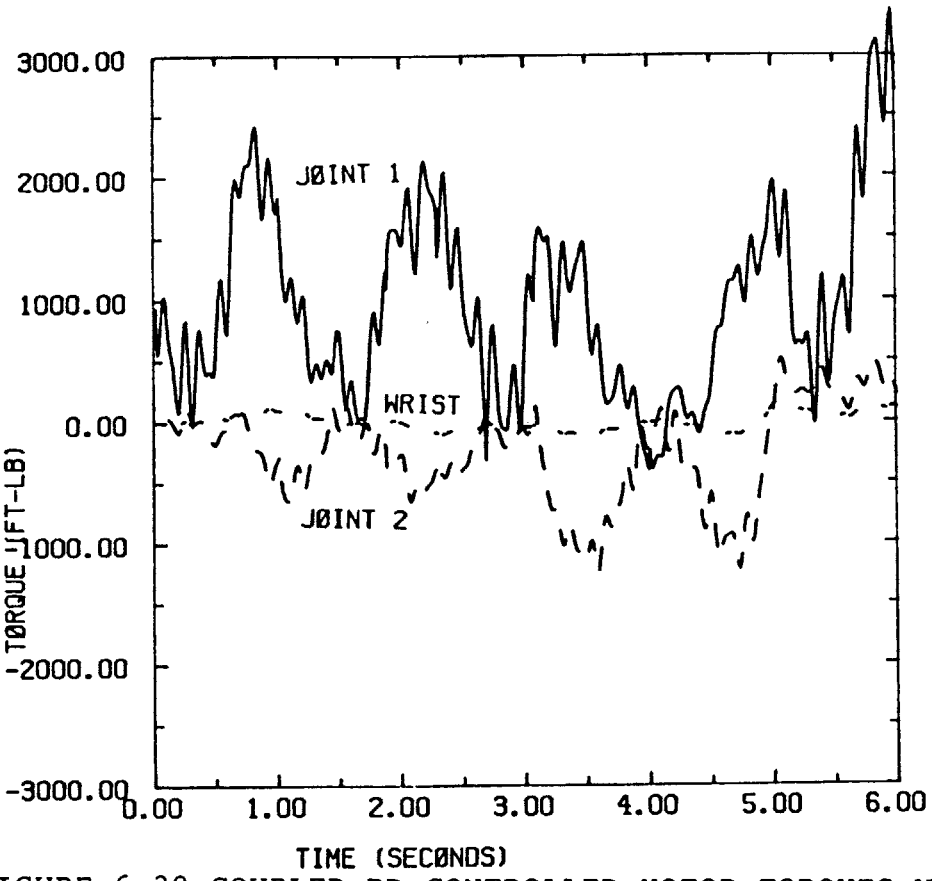

FIGURE 6-29 COUPLED PD CONTROLLED MOTOR TORQUES WITH PARTIAL COMPENSATOR

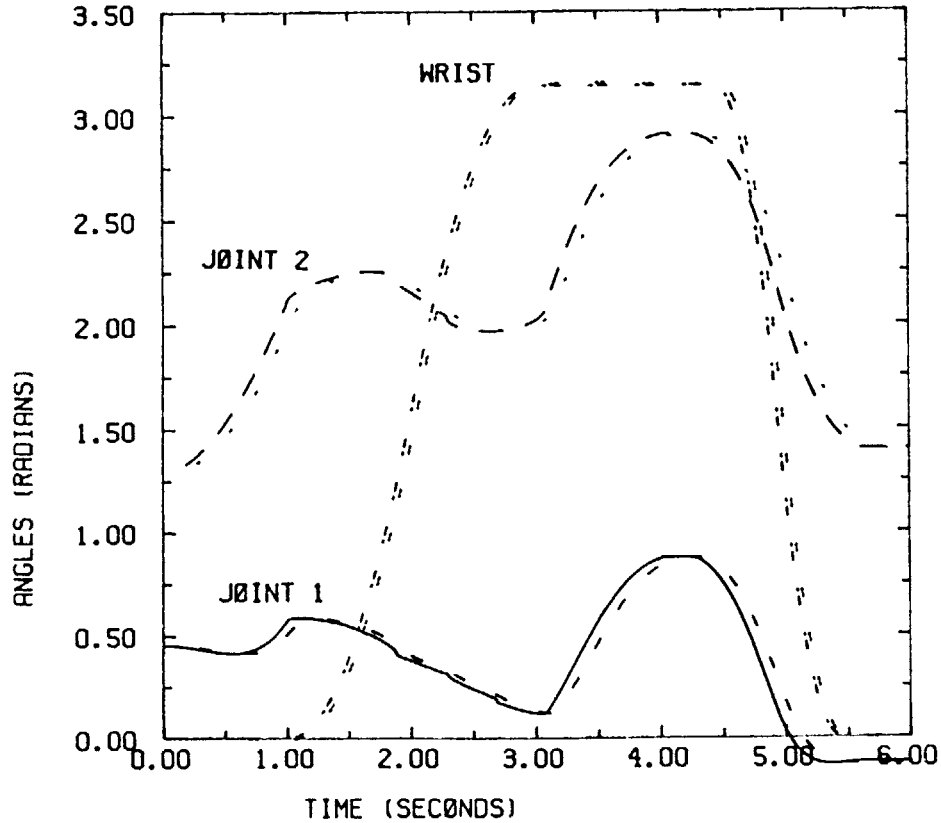

FIGURE 6-30 COUPLED PD CONTROLLED JOINT ANGLE COMPARISON WITH PARTIAL COMPENSATOR

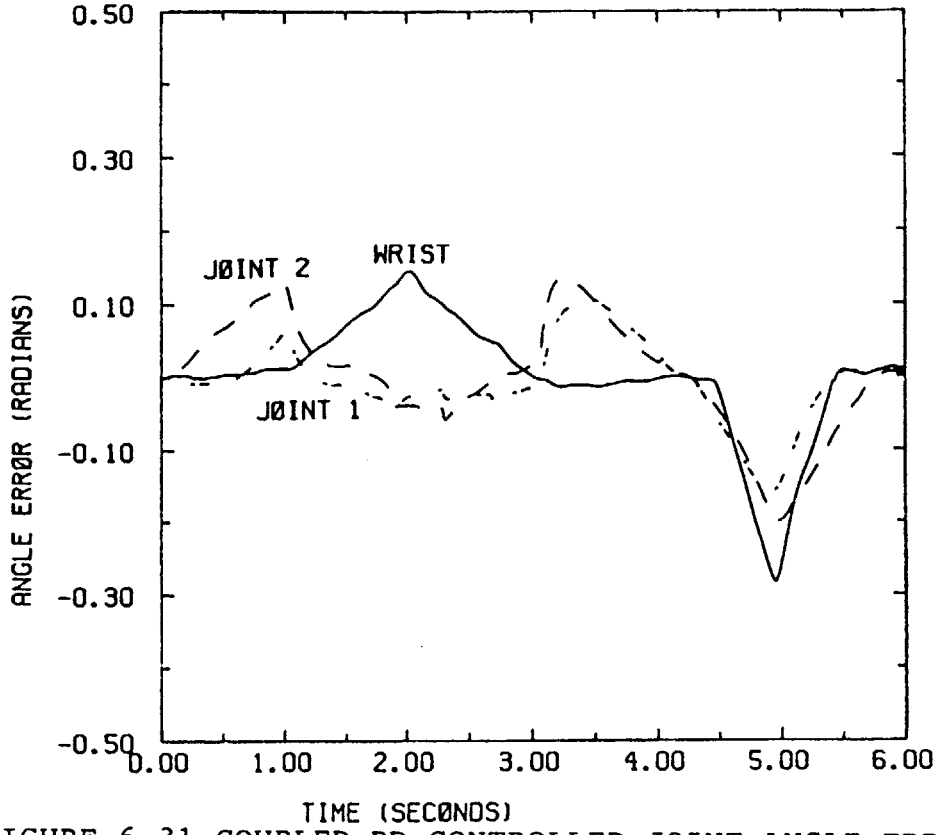

FIGURE 6-31 COUPLED PD CONTROLLED JOINT ANGLE ERRORS WITH PARTIAL COMPENSATOR

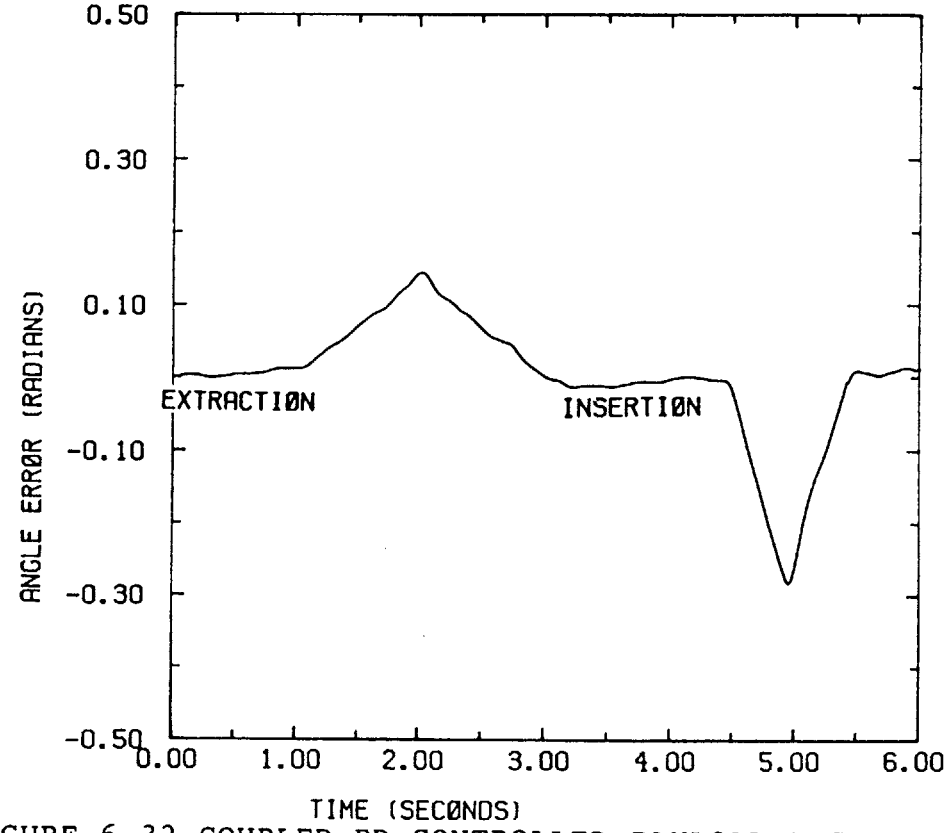

FIGURE 6-32 COUPLED PD CONTROLLED PAYLOAD ANGLE ERROR WITH PARTIAL COMPENSATOR

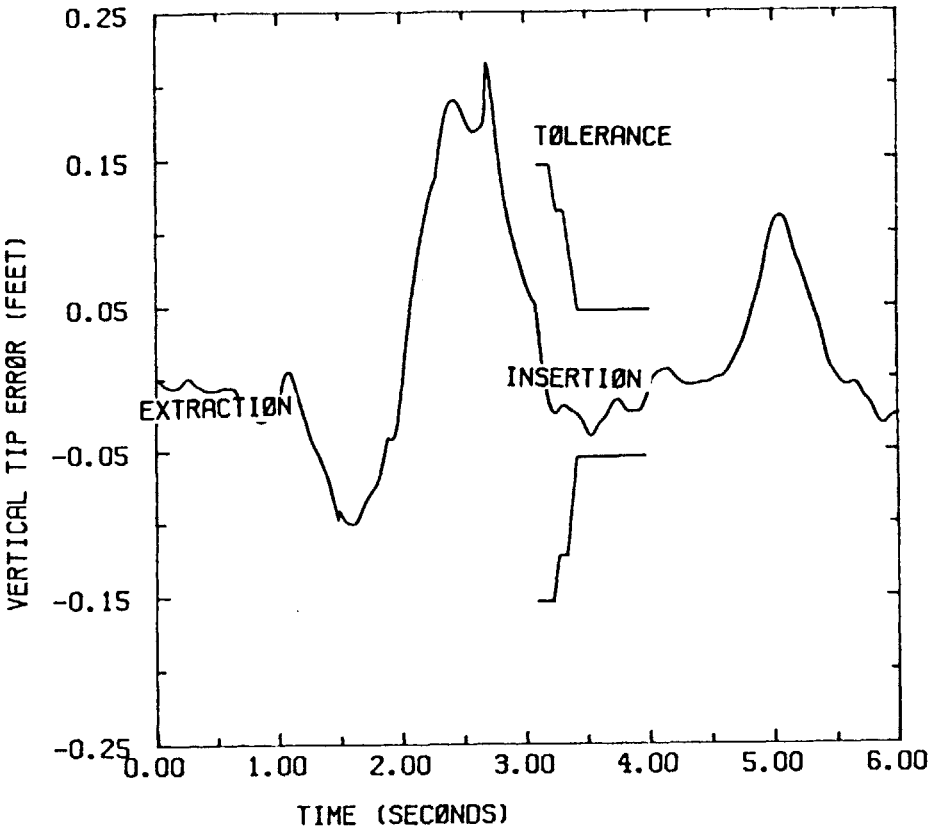

FIGURE 6-33 COUPLED PD CONTROLLED PAYLOAD VERTICAL TIP ERROR WITH PARTIAL COMPENSATOR

6.5 STABILIZED OPERATIONS

Tanks need the ability to fire accurately during movement. To help achieve this, a main gun stabilization system is used which maintains the gun barrel constant during turret pitch rotations. The gun elevation is maintained constant relative to world coordinates, such as a target even when the turret experiences a pitch rotation. A stabilized gun, though creates an even greater difficulty for a robotic or manual loader. During tank movement, an elevation stabilized gun moves relative to the turret. This results in a moving main gun breach that the loader must insert the sixty pound round into. The main gun breach

motion can not be anticipated by the loader, which makes the loading process nearly impossible and dangerous for the human loader. Currently, the tank must stop or turn off the stabilization system and index the gun breach to a specified elevation before the human loader attempts to load ammunition. Ideally, the main gun should be able to be loaded in all elevations while moving in a stabilized mode of operation [5]. During stabilized operation, the breach is moving, hence the loader must must properly align the ammunition with the gun breach prior to starting insertion of the round. The first simulations demonstrate the inability of the uncoupled controller, without compensator, to satisfactory load ammunition. Figure 6-34 and 6-35 clearly show that the payload fails to meet the tolerance limits.

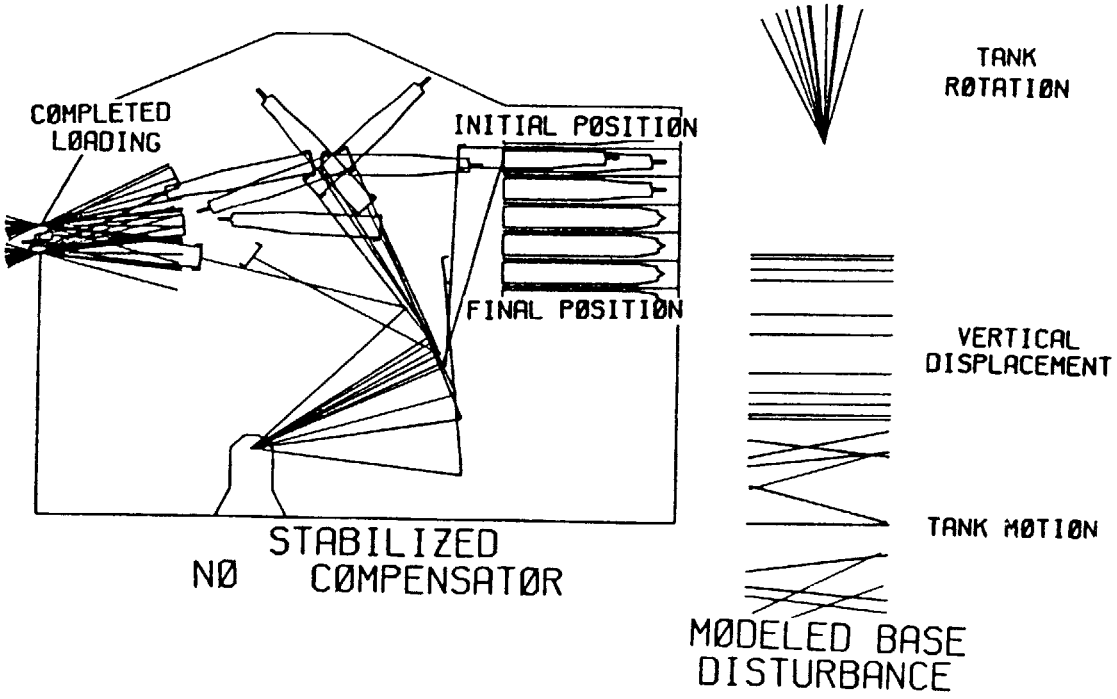

FIGURE 6-34 PD CONTROLLED ROBOT MOTION
DURING STABILIZED OPERATIONS

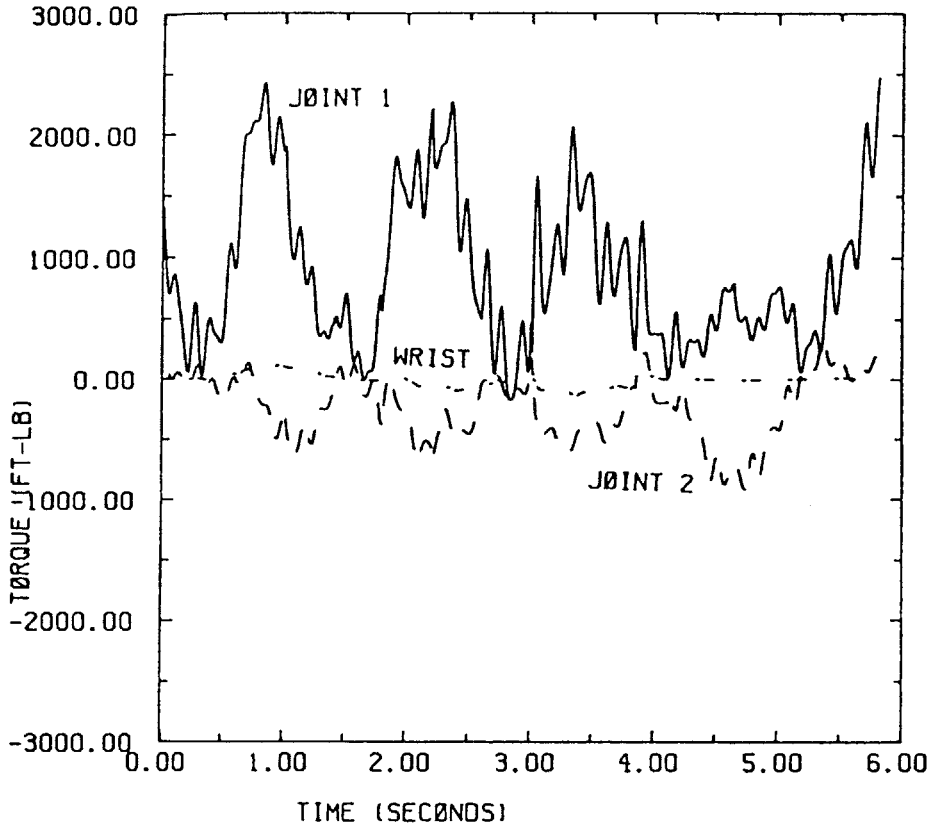

FIGURE 6-35 PD CONTROLLED MOTOR TORQUES DURING STABILIZED OPERATIONS

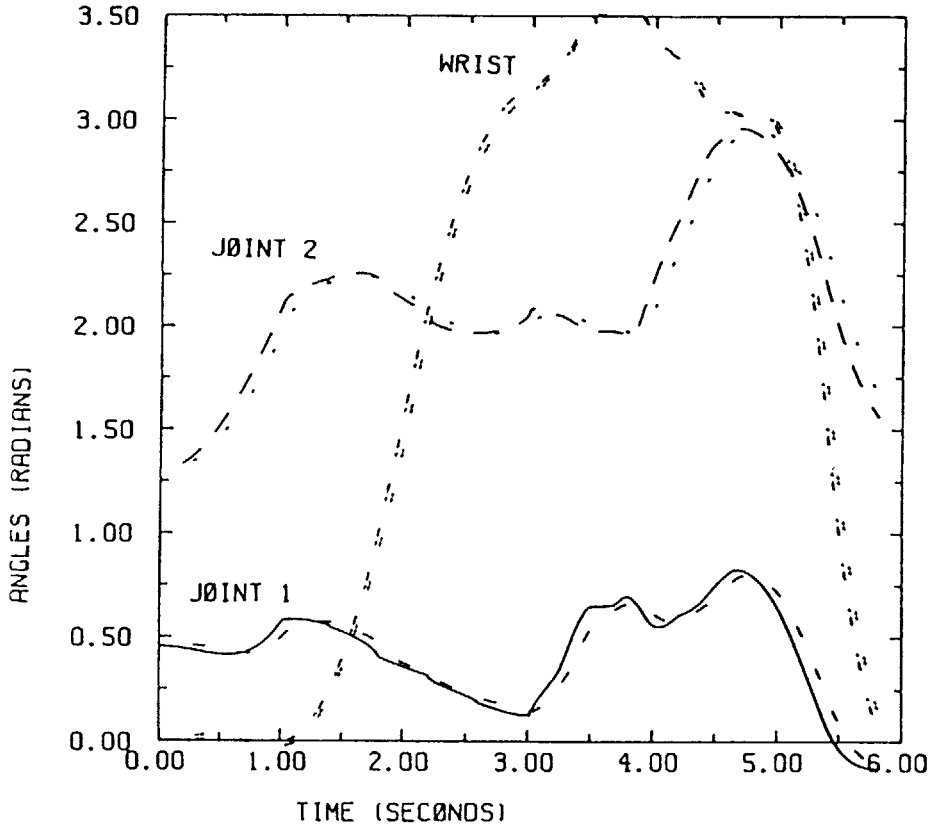

FIGURE 6-36 PD CONTROLLED JOINT ANGLES COMPARISON DURING STABILIZED OPERATIONS

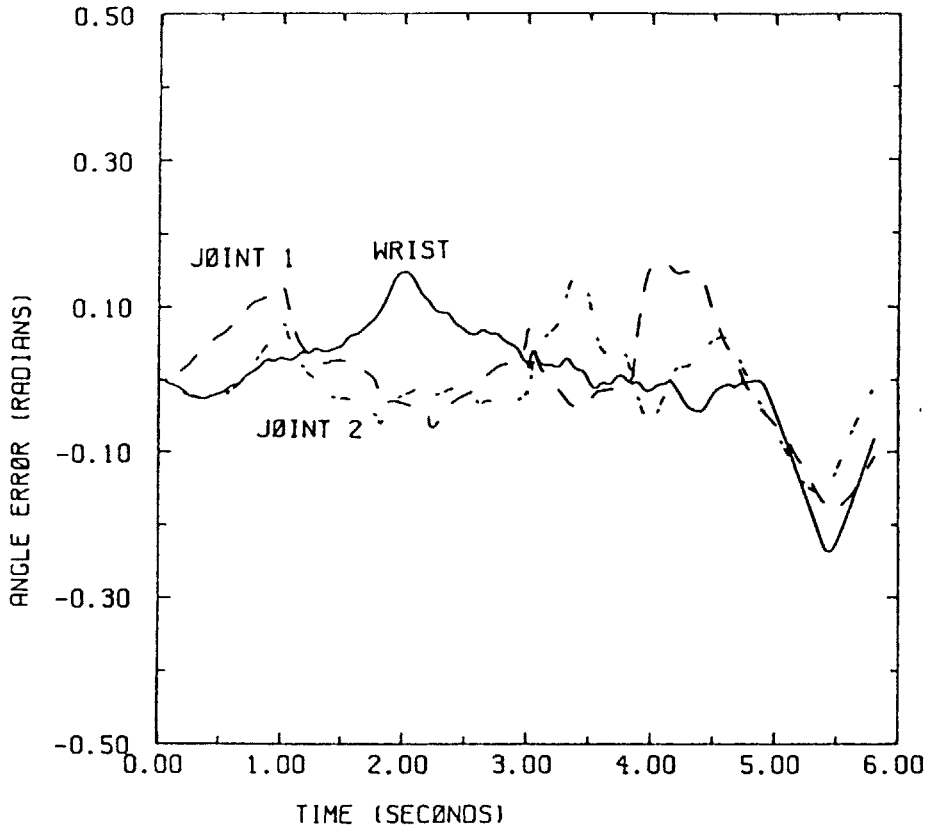

FIGURE 6-37 PD CONTROLLED JOINT ANGLE ERRORS DURING STABILIZED OPERATIONS

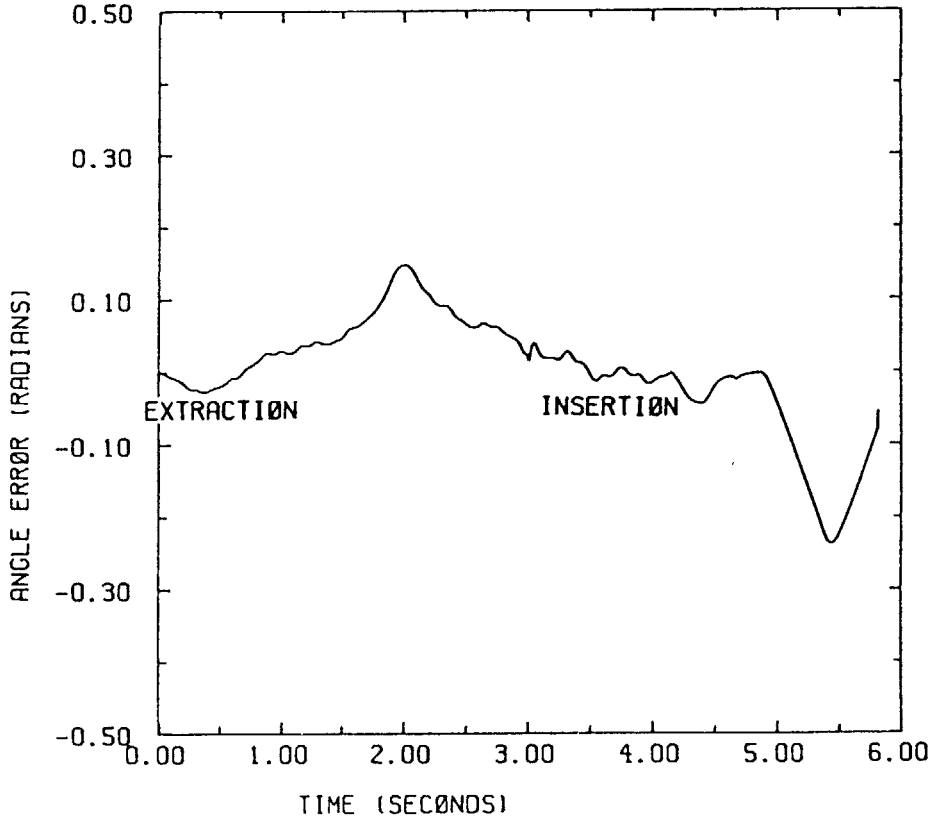

FIGURE 6-38 PD CONTROLLED PAYLOAD ANGLE ERROR DURING STABILIZED OPERATIONS

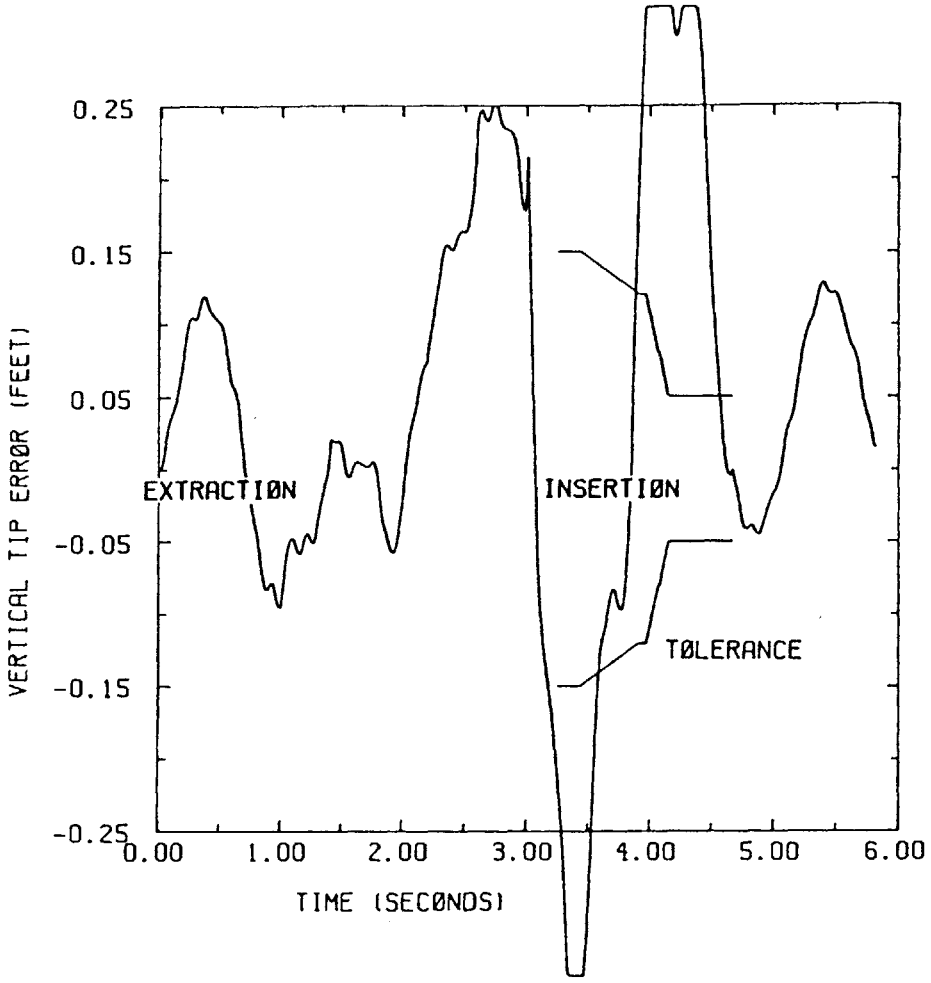

FIGURE 6-39 PD CONTROLLED PAYLOAD VERTICAL TIP ERROR DURING STABILIZED OPERATIONS

The next simulations are conducted with the main gun still stabilized. The performance above is unsatisfactory, as expected, so now the designed partial compensator is added to the PD controller. The performance is better, shown in figures 6-40, 6-41, 6-42, 6-43, 6-44, and 6-45, but even with the compensator not satisfactory.

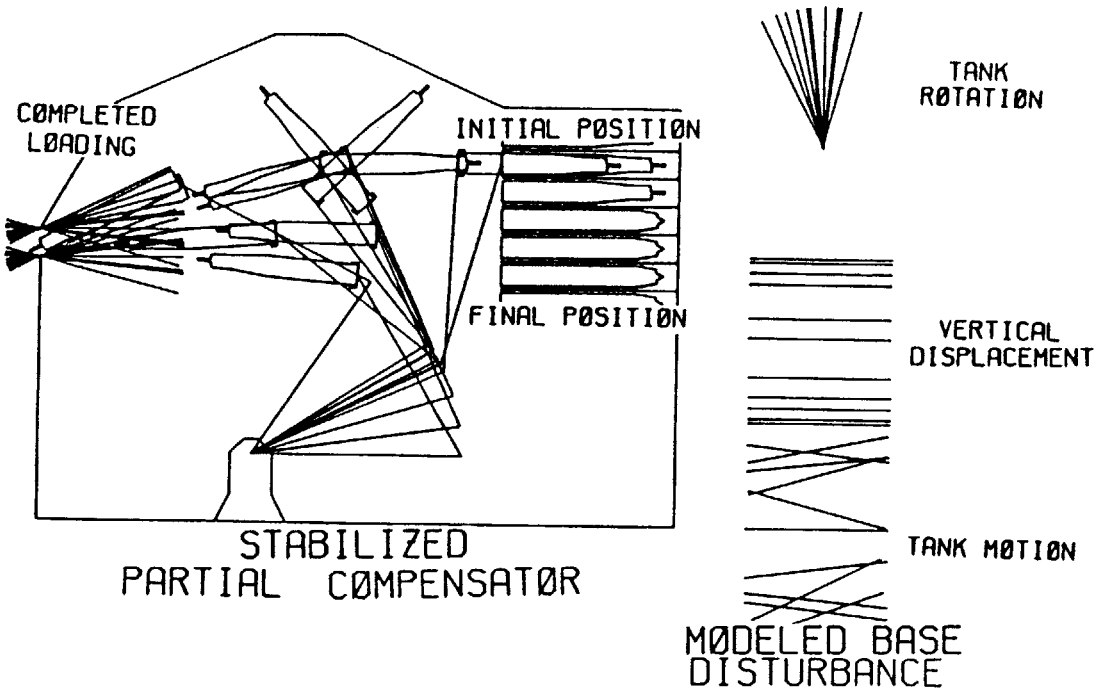

FIGURE 6-40 PD CONTROLLED ROBOT MOTION WITH PARTIAL COMPENSATOR DURING STABILIZED OPERATIONS

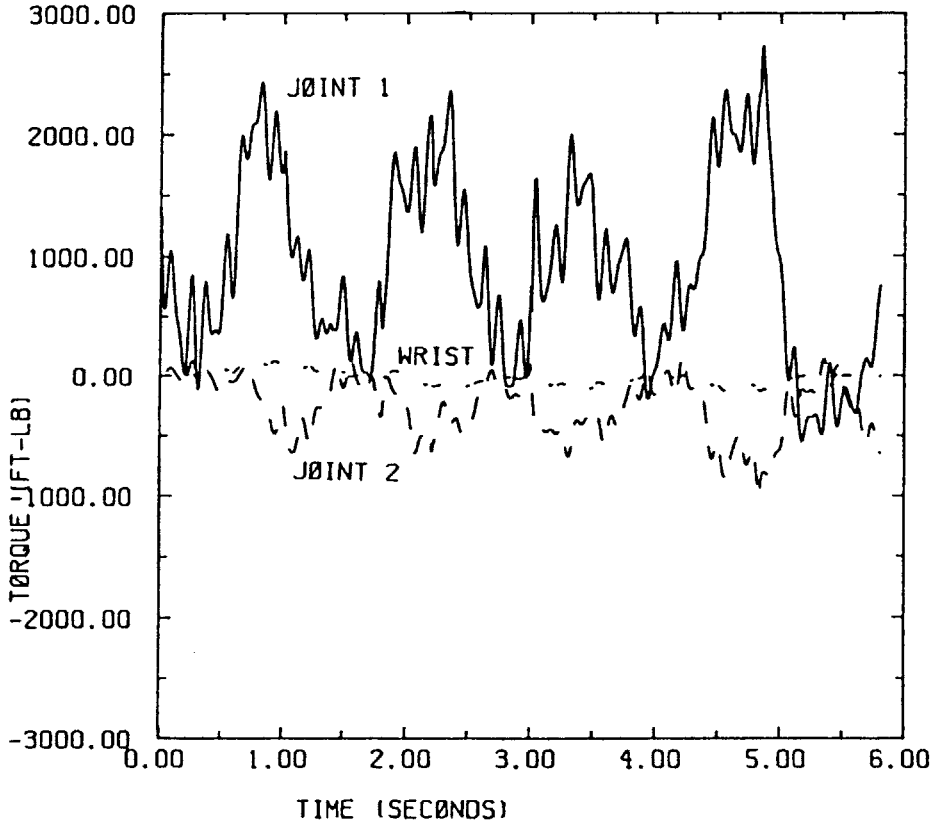

FIGURE 6-41 PD CONTROLLED MOTOR TORQUES WITH PARTIAL COMPENSATOR DURING STABILIZED OPERATIONS

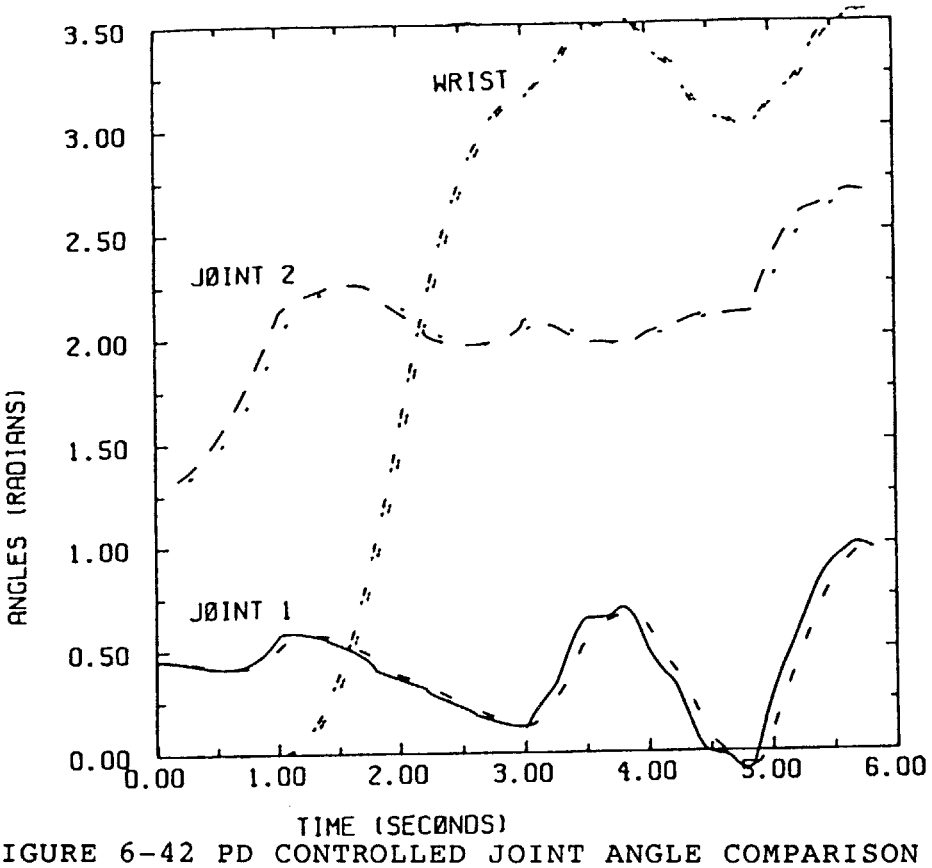

FIGURE 6-42 PD CONTROLLED JOINT ANGLE COMPARISON WITH PARTIAL COMPENSATOR DURING STABILIZED OPERATIONS

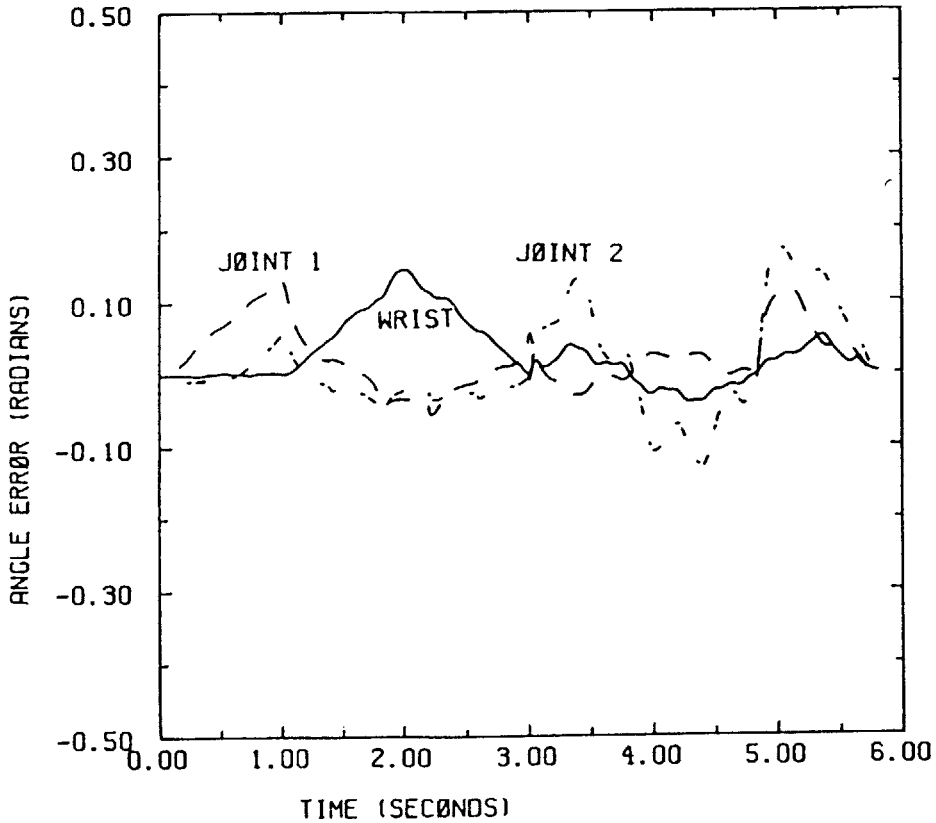

FIGURE 6-43 PD CONTROLLED JOINT ANGLE ERRORS WITH PARTIAL COMPENSATOR DURING STABILIZED OPERATIONS

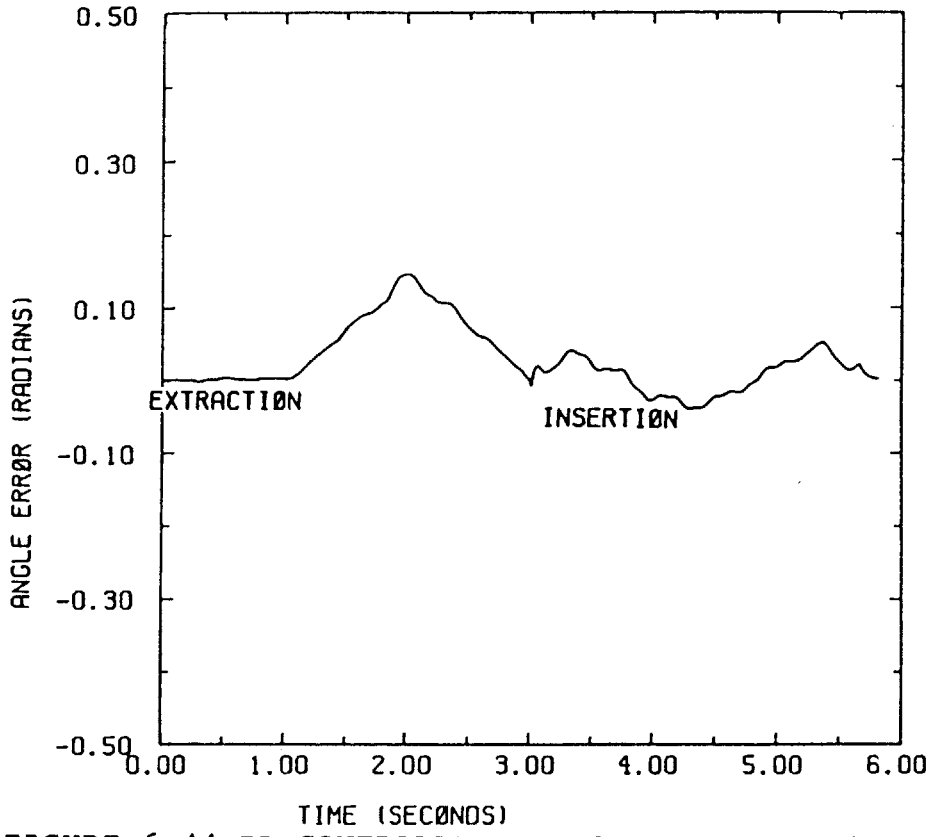

FIGURE 6-44 PD CONTROLLED PAYLOAD ANGLE ERROR WITH PARTIAL COMPENSATOR DURING STABILIZED OPERATIONS

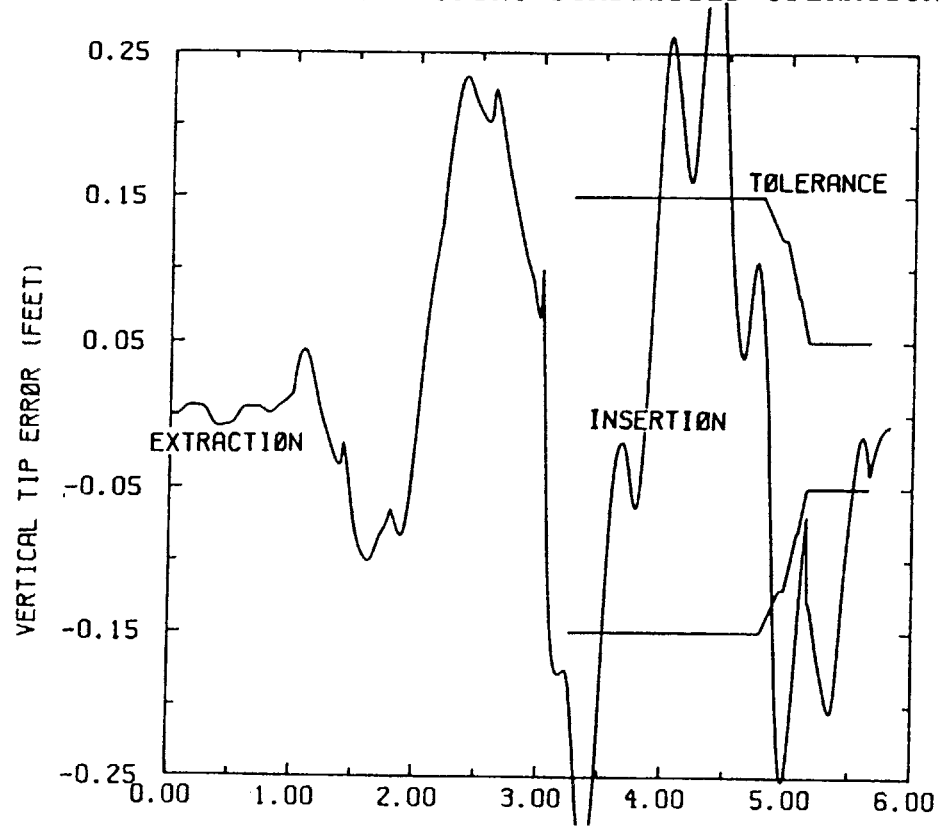

FIGURE 6-45 PD CONTROLLED PAYLOAD VERTICAL TIP ERROR WITH PARTIAL COMPENSATOR DURING STABILIZED OPERATIONS

These results demonstrate that the performance does not meet objectives. This failure can be explained by figures 6-46 and 6-47. Figure 6-46 shows a typical non-stabilized case. The vehicle is initially shown in figure 6-46 with a solid line. The vehicle then experiences a pitch rotation which rotates the vehicle to a new position, shown in figure 4-46 with a dashed line. The gun position, which is not stabilized, experiences the same rotation as the vehicle and stays constant relative to the tank turret. The robotic loader's position remains the same, which is accomplished using the compensator. Figure 4-47 shows the same situation, but now the gun is stabilized, and the gun has maintained its elevation and the breach has risen in reference to the turret. The rotational disturbance force causes the round to want to drop relative to the turret. The compensator overcomes this force, which keeps the round in the dotted position, but an additional force is necessary to lift the round to the gun breach's new position. This is achieved by feeding joint velocity terms forward. The results of this simulation are shown in figures 6-48, 6-49, 6-50, 6-51, 6-52, and 6-53. The additional feedforward torques are only sent during the insertion phase.

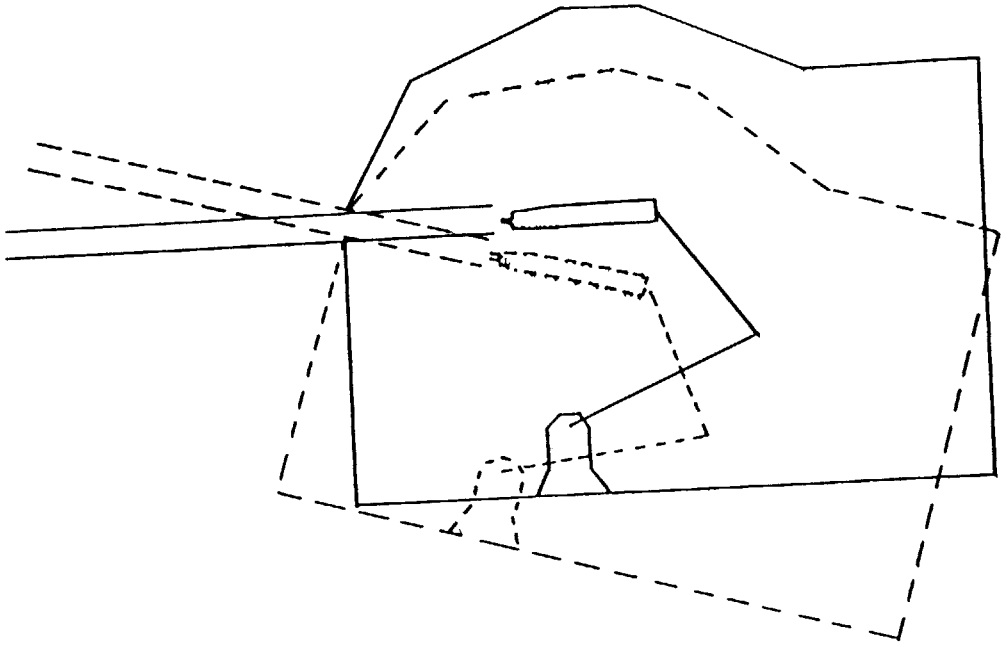

FIGURE 6-46 TANK NON-STABILIZED OPERATION

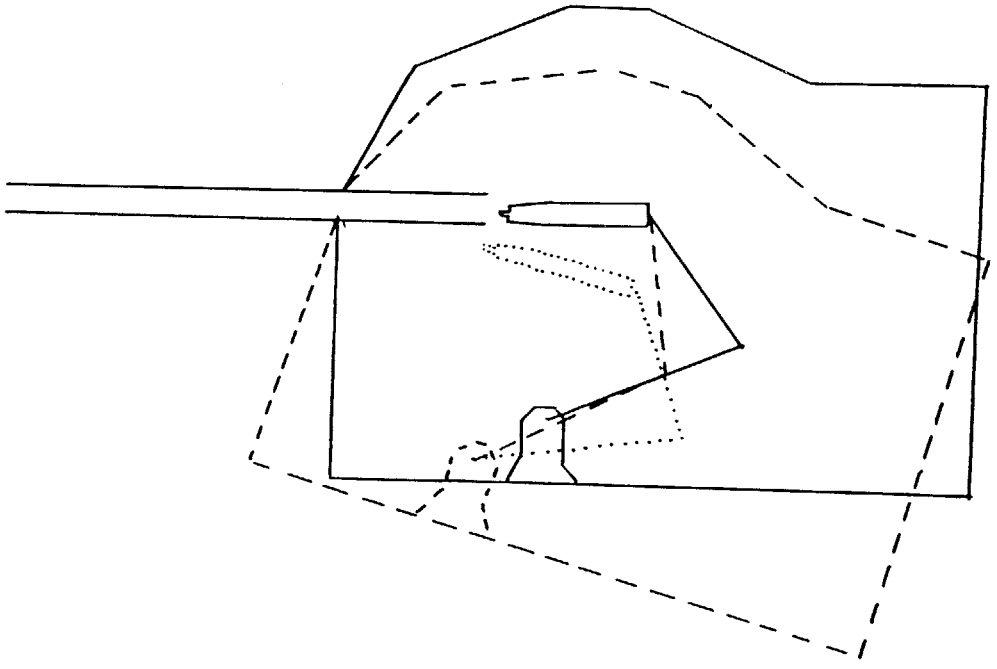

FIGURE 6-47 TANK STABILIZED OPERATION

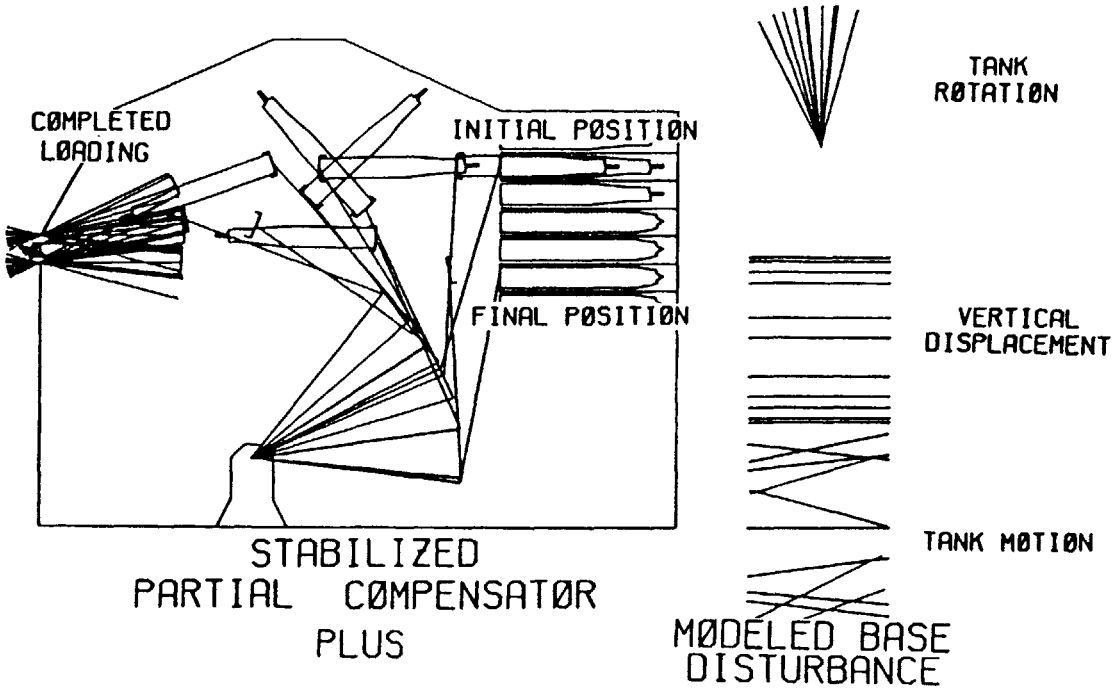

FIGURE 6-48 PD CONTROLLED ROBOT MOTION WITH PARTIAL COMPENSATOR PLUS DURING STABILIZED OPERATIONS

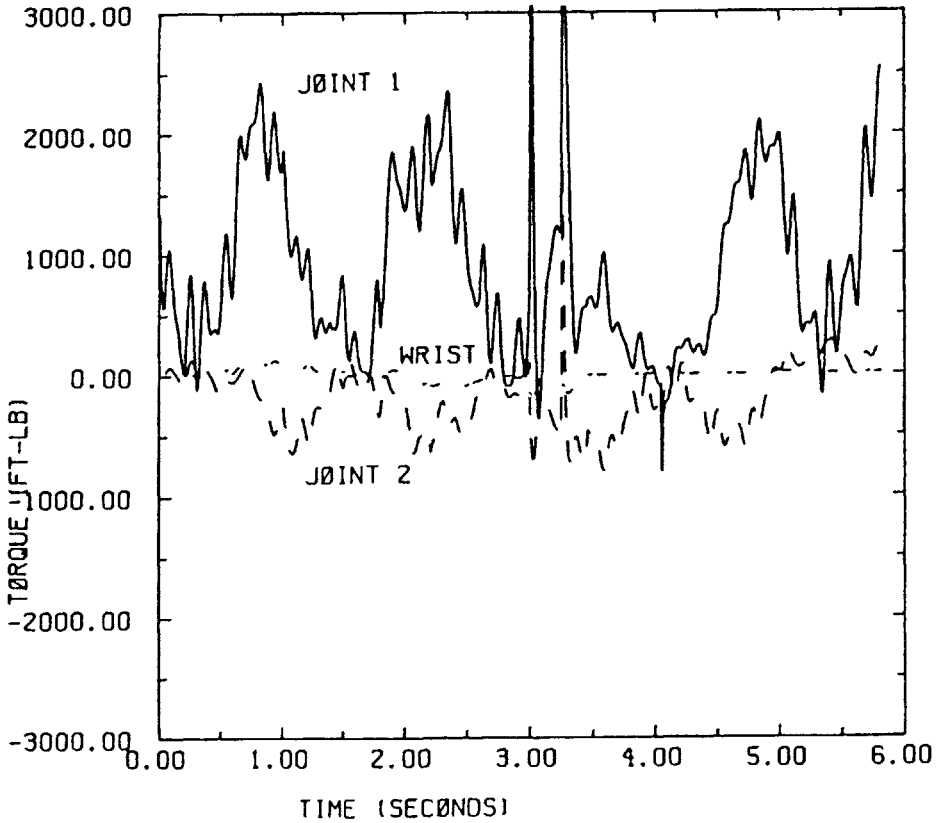

FIGURE 6-49 PD CONTROLLED MOTOR TORQUES WITH PARTIAL COMPENSATOR PLUS DURING STABILIZED OPERATIONS

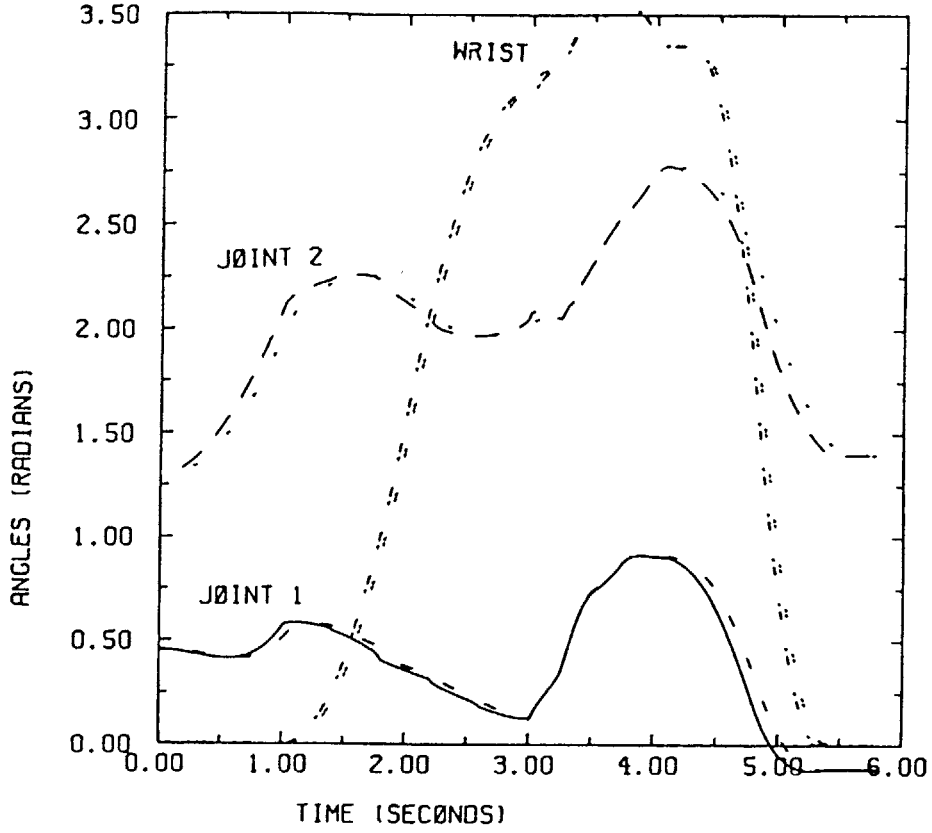

FIGURE 6-50 PD CONTROLLED JOINT ANGLE COMPARISON WITH PARTIAL COMPENSATOR PLUS DURING STABILIZED OPERATIONS

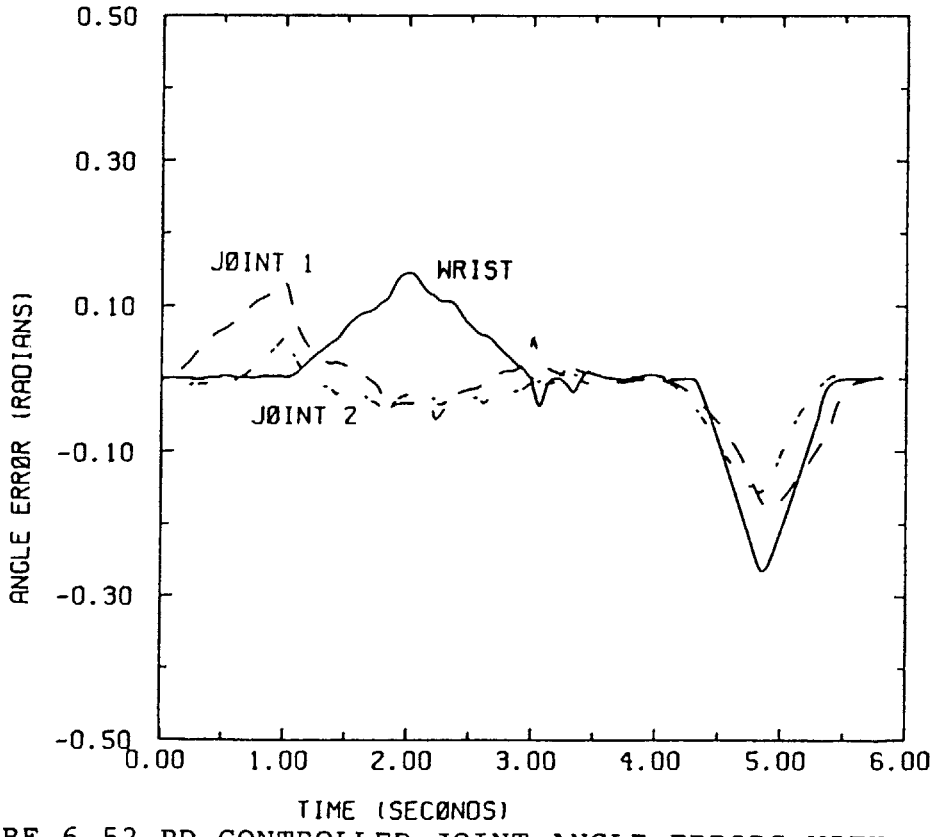

FIGURE 6-52 PD CONTROLLED JOINT ANGLE ERRORS WITH PARTIAL COMPENSATOR PLUS DURING STABILIZED OPERATIONS

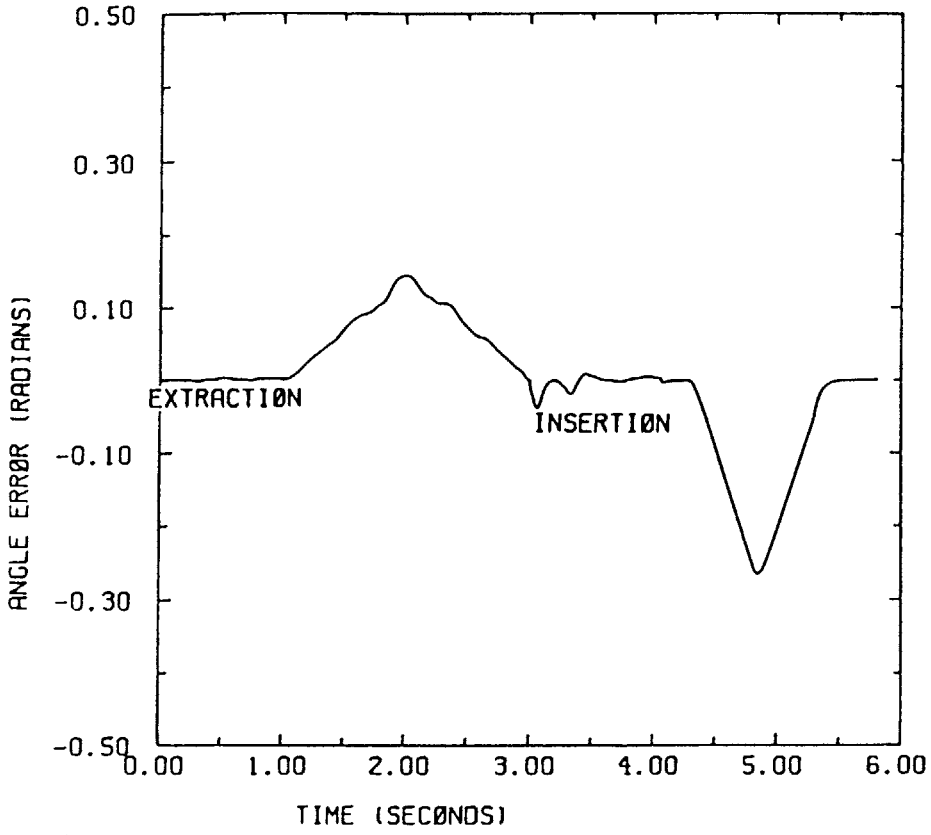

FIGURE 6-53 PD CONTROLLED PAYLOAD ANGLE ERROR WITH PARTIAL COMPENSATOR PLUS DURING STABILIZED OPERATIONS

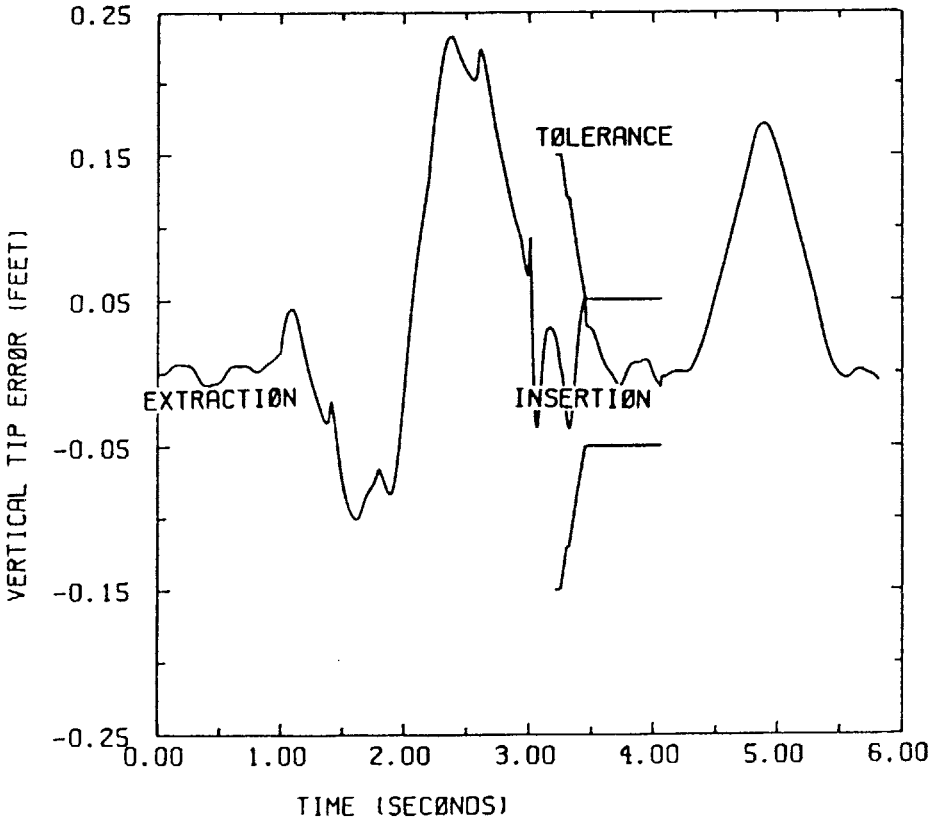

FIGURE 6-54 PD CONTROLLED PAYLOAD VERTICAL TIP ERROR WITH PARTIAL COMPENSATOR PLUS DURING STABILIZED OPERATIONS

This chapter demonstrates the successful design and application of a compensator for overcoming the adverse effects generated by base motion disturbances on the accuracy of a mobile robot. This compensator utilizes sensory information from the base to calculate the forces created by the disturbances and then feed forward a corresponding torque to each joint actuator to counter the disturbance effect. The base motion vertical acceleration term is determined to have the greatest adverse effect on robot accuracy, and hence must be included in a compensator to achieve good performance. This compensator is then successfully applied to both the uncoupled and coupled controllers.

CHAPTER 7

EXPERIMENTAL DEMONSTRATION

7.1 EXPERIMENTAL VERIFCATION

Experimental implementation of the sensory compensator approach is desirable to demonstrate its validity and practicality. It is important to show that this compensator can practically be included with a mobile robot controller. To be practical, this implementation should demonstrate that the necessary sensory base measurements can be accurately obtained and the computations completed fast enough for the compensating torques to be useful. Additionally, the simplifying assumptions made to develop the actual robotic loader system can result in misleading conclusions. The assumptions made in this research are considered valid and should provide accurate results, but unmodelled characteristics may influence the behavior of an actual system. Therefore, to give credibility to the model used and the conclusions made, the designed compensator is demonstrated experimentally. The demonstration shows the necessity for, and the ability of, the compensator to overcome the base disturbances imposed on a mobile robot.

7.2 EXPERIMENTAL HARDWARE

The experiment is conducted in the laboratories of the Mechanical Engineering Department at the Massachuetts

Institute of Technology. A schematic of the experimental hardware is shown in figure 7-1. This figure shows a PUMA 250 robot mounted on a vertical motion platform. Two micro-computers are utilized, one to control the platform motion and one to control the robot. Measured acceleration signals from a accelerometer mounted on the platform are received by the robot computer from the platform computer for use by the disturbance compensator. Each of the individual components of the experimental hardware are discussed next.

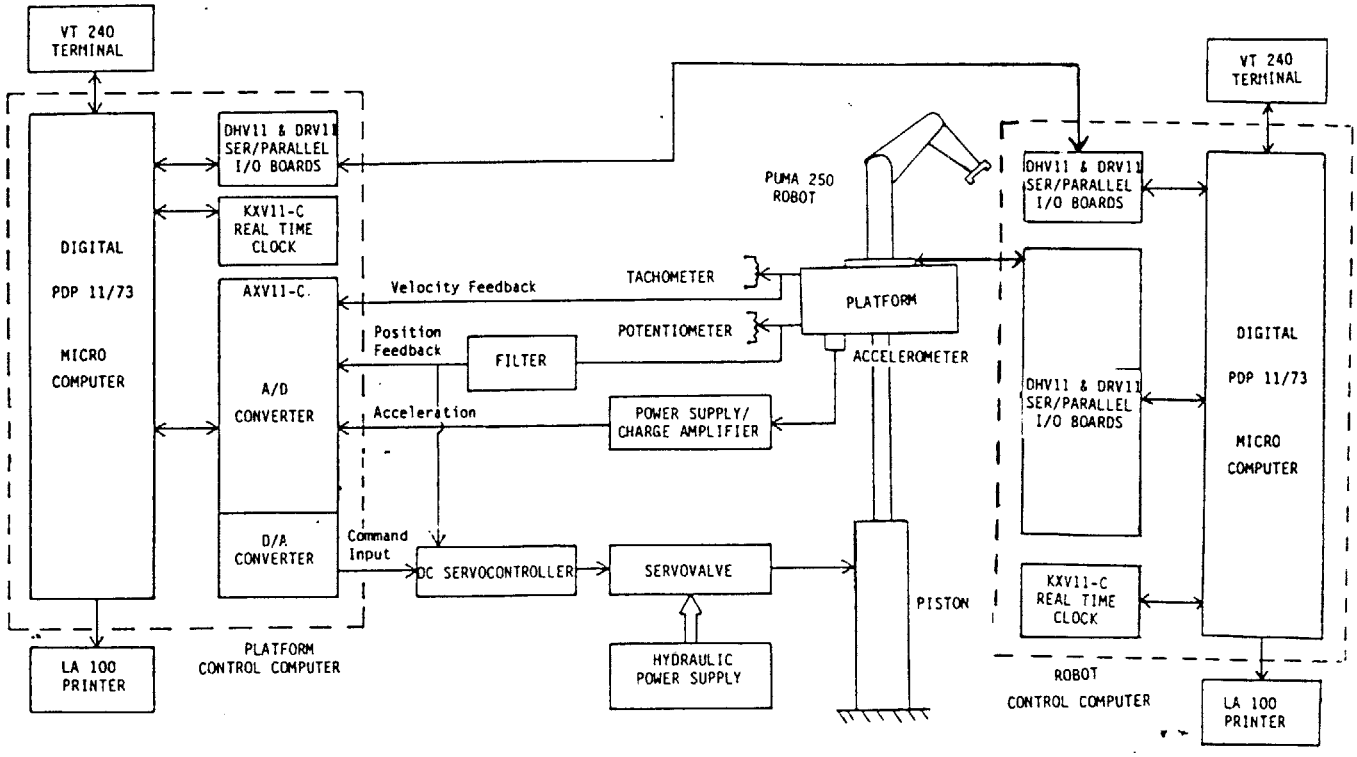

FIGURE 7-1 EXPERIMENTAL HARDWARE

ROBOT:

A Unimate PUMA 250 computer controlled robot arm system is used to demonstrate the mobile robot. The robot, shown in figure 7-2, has six revolute axes. Each joint is driven by a permanent-magnet servomotor through a gear train. Optical incremental encoders provide positioning information for each joint [18]. The demonstration of planar motion requires three degrees of freedom, so robot joints 2, 3, and 5 are utilized.

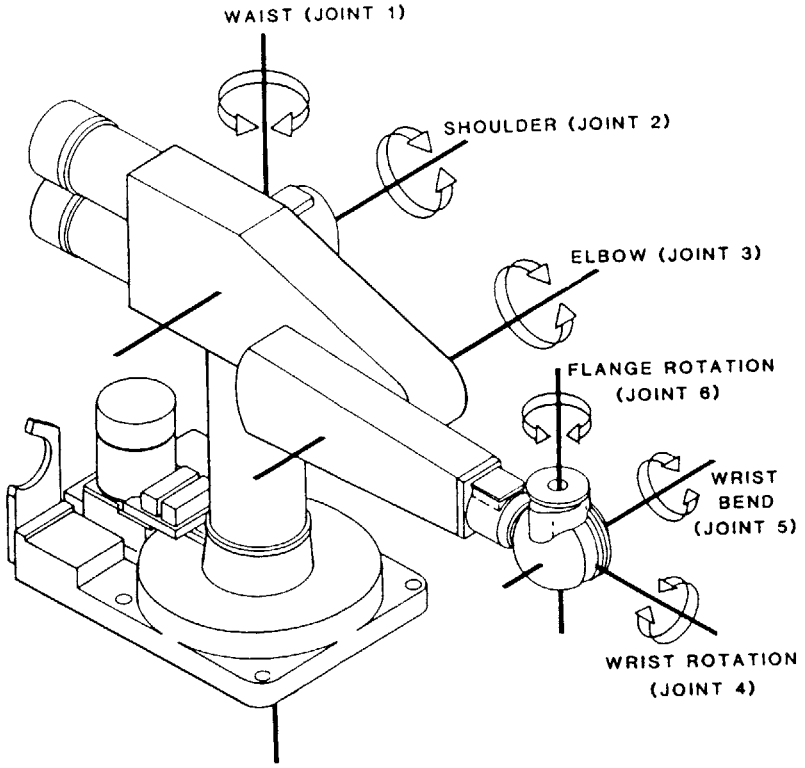

FIGURE 7-2 PUMA 250

The disturbances generate forces which must be countered by joint torques applied by the compensator. The calculation of these compensating torques require information of the robot link parameters. The link parameters for the PUMA 250 robot are determined experimentally by balancing the gravity forces on the robot arm held horizontally with a gravity compensator. By placing different known payloads in the endeffector, the necessary torques at each joint to balance the links is determined. The link parameters are estimated from this information, and tested in all positions. The parameters used are listed below:

M_2 = 2.30 kilograms

l_2 = 18 mm

L_2 = 203 mm

M_3 = 1.30 kilograms

l_3 = 38 mm

L_3 = 203 mm

M_w = 0.35 kilograms

l_w = 57 mm

L_w = 117.5 mm

M_p = 0.198 kilograms

l_p = 67 mm

PLATFORM:

A one degree of freedom platform, which only allows vertical motion, is used to create the robot base motion disturbance. The platform consists of an aluminum mounting plate driven by a hydraulic piston, which is mounted centrally under the plate. The plate is maintained parallel to the horizontal by three shafts sliding in linear ball bearings [19].

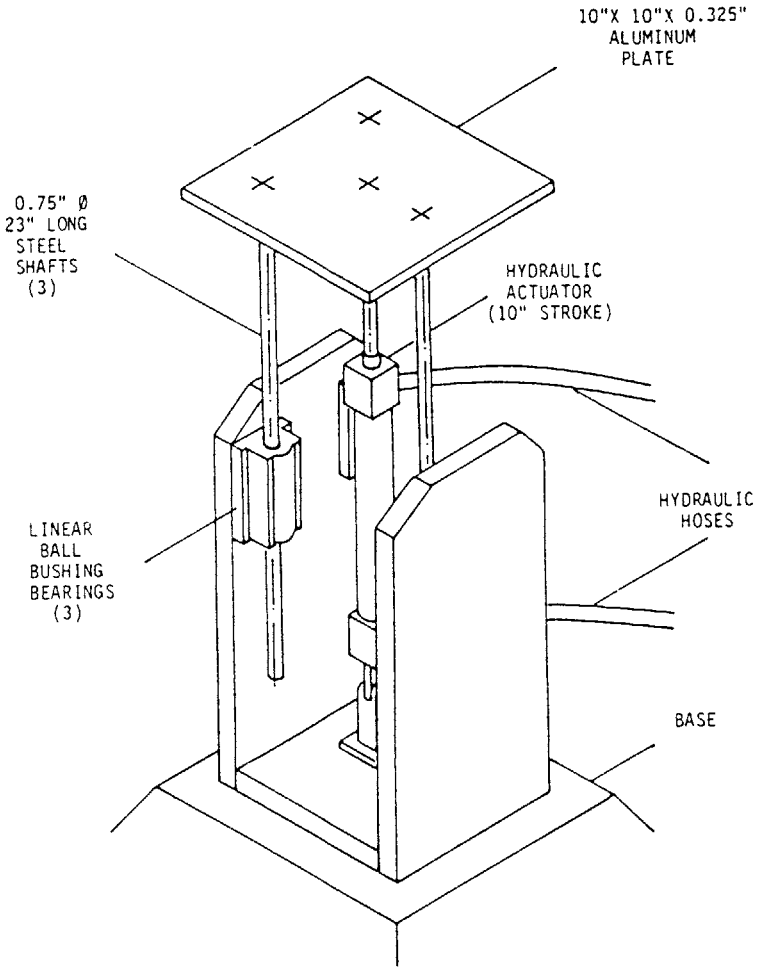

FIGURE 7-3 PLATFORM

The platform is driven hydraulically by a servovalve controlled by a Digital Equipment Corporation PDP 11/73

micro computer. The desired platform motion is specified in the platform controller and includes any sinusoidal motion whose amplitude of acceleration does not exceed two times the acceleration of gravity. The platform is equiped with a PCB model number 308B09 accelerometer with 100.9 mvolts per gravity sensitivity [20]. The acceleration measurements made by the accelerometer contain high frequency noise. The acceleration signal is shown in figure 7-4, which shows the acceleration, in g's, for the platform operating sinusoidally with 4.5 inches amplitude and a frequency of 0.7 hertz. This figure shows the high frequency noise in the acceleration signal which has to be filtered to achieve a smooth acceleration measurement.

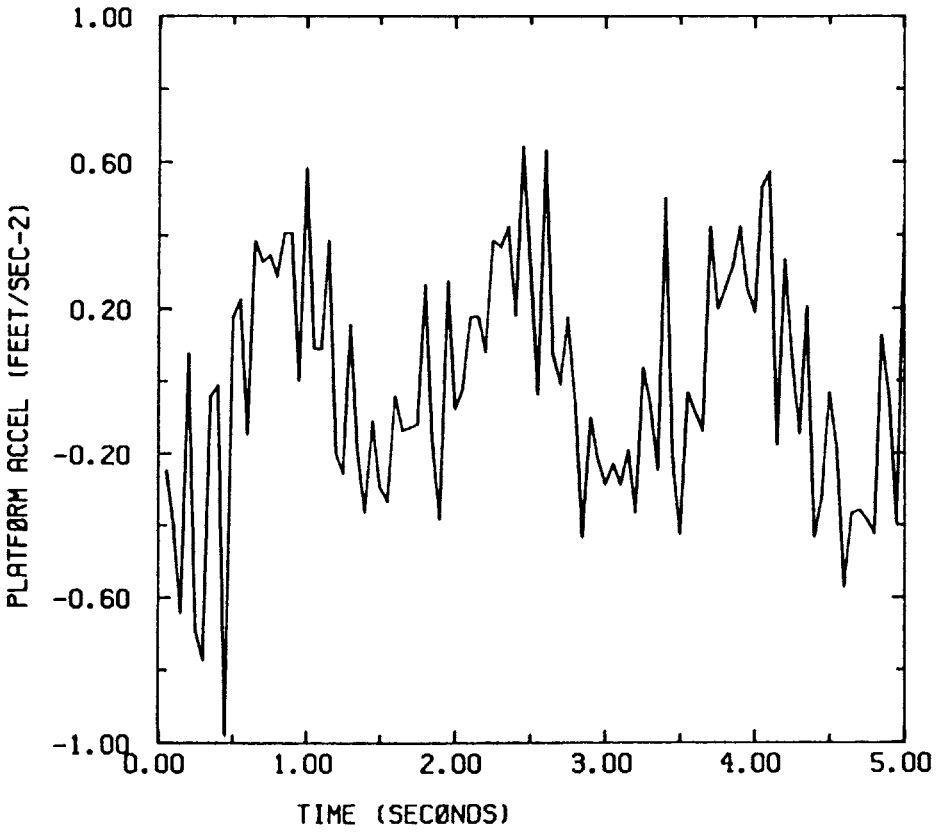

FIGURE 7-4 UNFILTERED ACCELERATION SIGNALS

With a PCB model number 474N06 50 hertz filter installed, many of the higher frequency signals are filtered and the signal is better, but still contains high frequency noise and is not suitable for the compensator. This filtered signal is shown in figure 7-5.

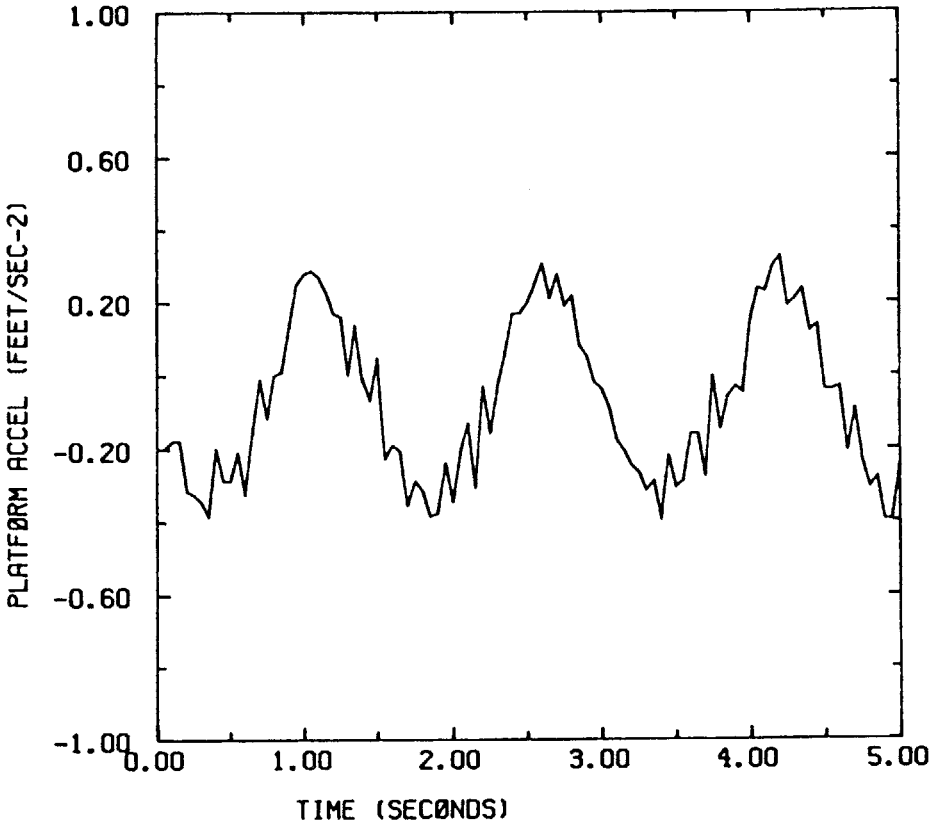

FIGURE 7-5 FILTERED ACCELERATION SIGNALS

A digital filter is then designed to smooth out the acceleration signals. A Butterworth filter is used to pass the low frequency signals and to filter the high frequency signals. The platform operates at frequencies up to one hertz, so signals above one hertz need to be filtered. Additionally, the filter can not introduce a substantial phase lag, which would make the acceleration signal

unsuitable. A fourth order digital filter is designed which filters ten percent of the signals below 3.68 hertz and ninety-five percent above 7.0 hertz [21]. The result of this filter is shown in figure 7-6, which provides a smooth acceleration signal.

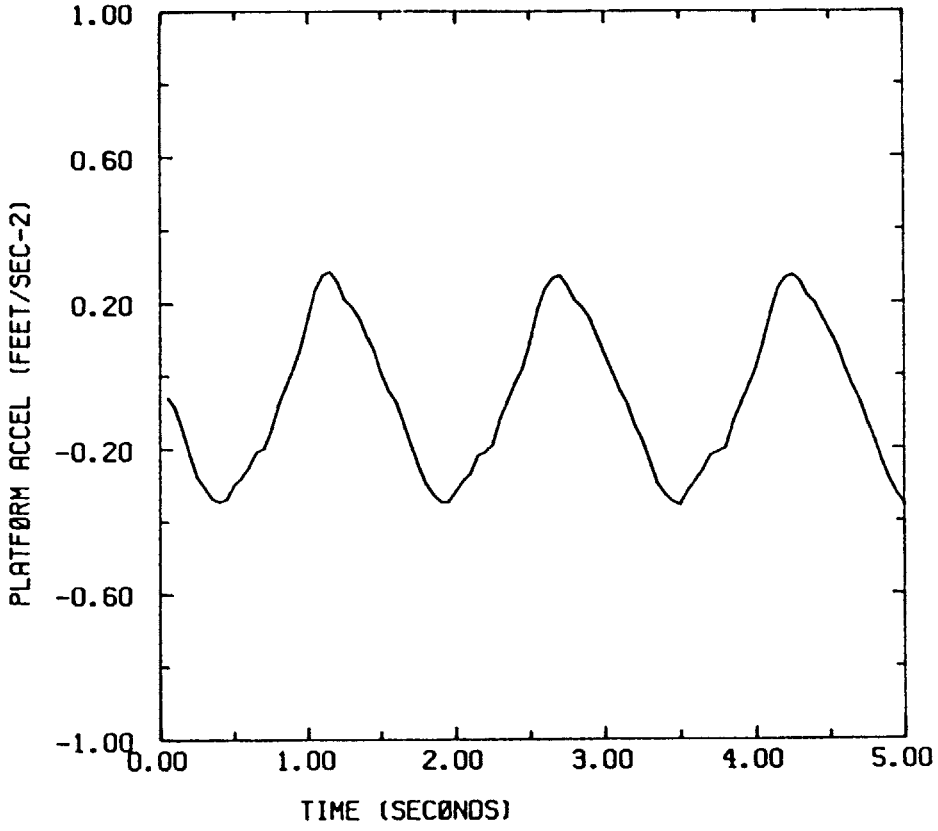

FIGURE 7-6 DIGITALLY FILTERED ACCELERATION SIGNAL

The platform PDP 11/73 computer receives the acceleration signals from the platform, filters the signals, and makes these signals available to the robot controller through parallel interface boards.

ROBOT CONTROLLER:

The robot controller is accomplished in a PDP 11/73 Digital micro-computer. The computer is connected to the robot through custom designed electronic boards which receive position information from the joints and sends motor voltages to the joints. This information passes through the Unimation VAL controller's digital interface boards and the analog servo boards respectively. A PID controller, designed by Jeff Whaley, is used to control the PUMA robot [22]. This controller is easily adapted to meet any desired controller design. The controller sample rate time interval, the distance the joints are commanded to move during each sample rate period, and the controller gains are the adjustable variables within the controller. To obtain smooth robot motion, two modifications are made to the controller. The desired angles necessary to achieve the baseline trajectory, developed in Appendix C, are used for this experimental demonstration. The distance between successive joint angle commands are further apart than the robot can move in one sample rate period, since the robot's motion is limited by the motor saturations. Therefore, the number of iterations that the controller spends on each set of joint angle commands is made variable. Additionally, the distance, measured in counts, moved by each joint during each sample rate period is made variable and synchronized with each other. This value insures each joint approaches the commanded position simultaneously and is constantly

adjusted for every set of joint angle commands.

The trajectory developed to conduct a sample load and return cycle for the model, and used for simulations, is now used for the demonstration. The required absolute joint angles, measured in radians, are transformed to relative joint angles, measured in counts. The transformed joint angles are compatible with the robot controller.

$$\theta_2^r = \phi_1^r \text{ [RADTOCOUNT(2)] [FACTOR(2)]}$$

$$\theta_3^r = -\theta_2^r + \phi_2^r \text{ [RADTOCOUNT(3)] [FACTOR(3)]}$$

$$\theta_5^r = (\theta_3^r - \theta_2^r) + \phi_3^r \text{ [RADTOCOUNT(5)] [FACTOR(5)]}$$

where: ϕ^r is the desired absolute joint angles(rads)

θ^r is the transformed relative angles(counts)

RADTOCOUNT are conversions dependent upon the joint's gear train. Values are determined by moving the joints through measured angles(degrees)

FACTOR are the number of counts each joint is from a reference, horizontal.

COMPENSATOR:

A compensator, utilizing the disturbance terms from the equations of motions, equations 6.1, 6.2, and 6.3, is included into the controller program. The calculated robot parameters are included to compute the compensating torques necessary to counter the vertical base motion disturbance imposed on the robot by the platform. These torques are added to the generated torques from the robot controller and sent through digital to analog boards and amplifiers to the robot's joint motors. The PASCAL computer programs are listed in Appendix E.

7.3 EXPERIMENTAL RESULTS

The first test conducted with the robot, is a demonstration of the robot arm held horizontally with a payload in the gripper of the end effector. The controller gains are set to zero and a gravity compensator is used, utilizing the developed parametric model of the arm, to hold the robot in position, horizontally. The platform is operated with an amplitude of 4.5 inches and frequencies of 0.7 and 0.8 hertz. Figure 7-5 and 7-6 show the joint positions, in radians, while the platform is moving. The vertical base motion disturbance create forces which cause the arm position to change. Next, the compensator is turned on, and figures 7-7 and 7-8 show that the compensator is effective in reducing the joint errors.

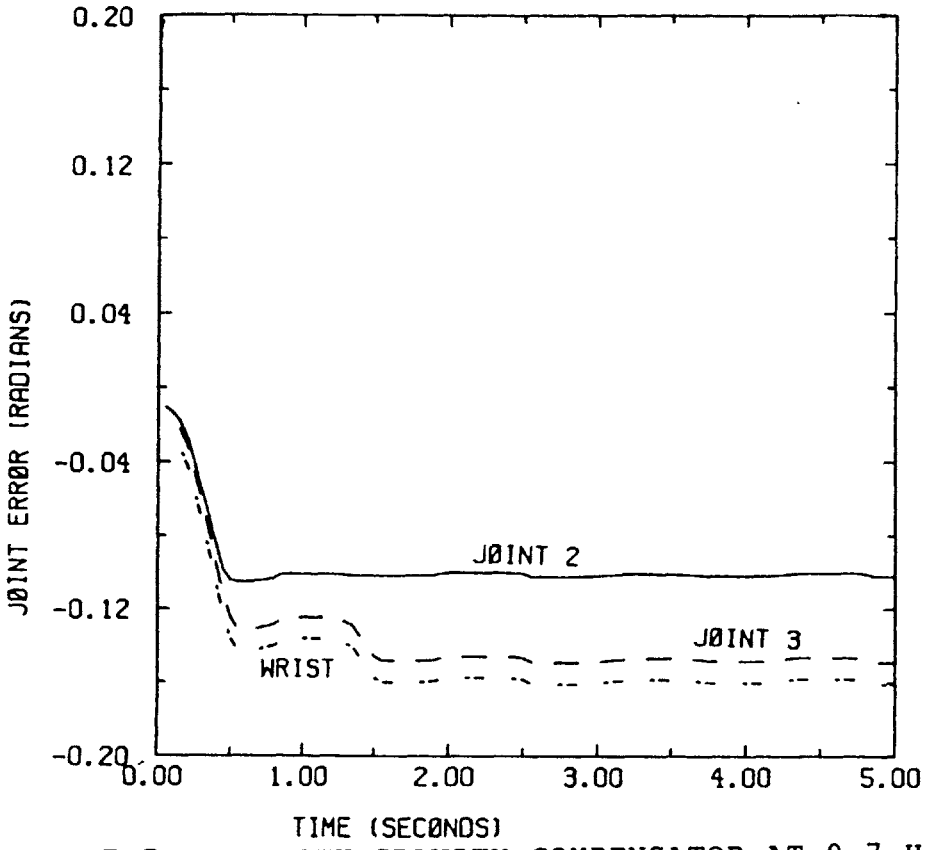

FIGURE 7-7 PUMA WITH GRAVITY COMPENSATOR AT 0.7 HERTZ

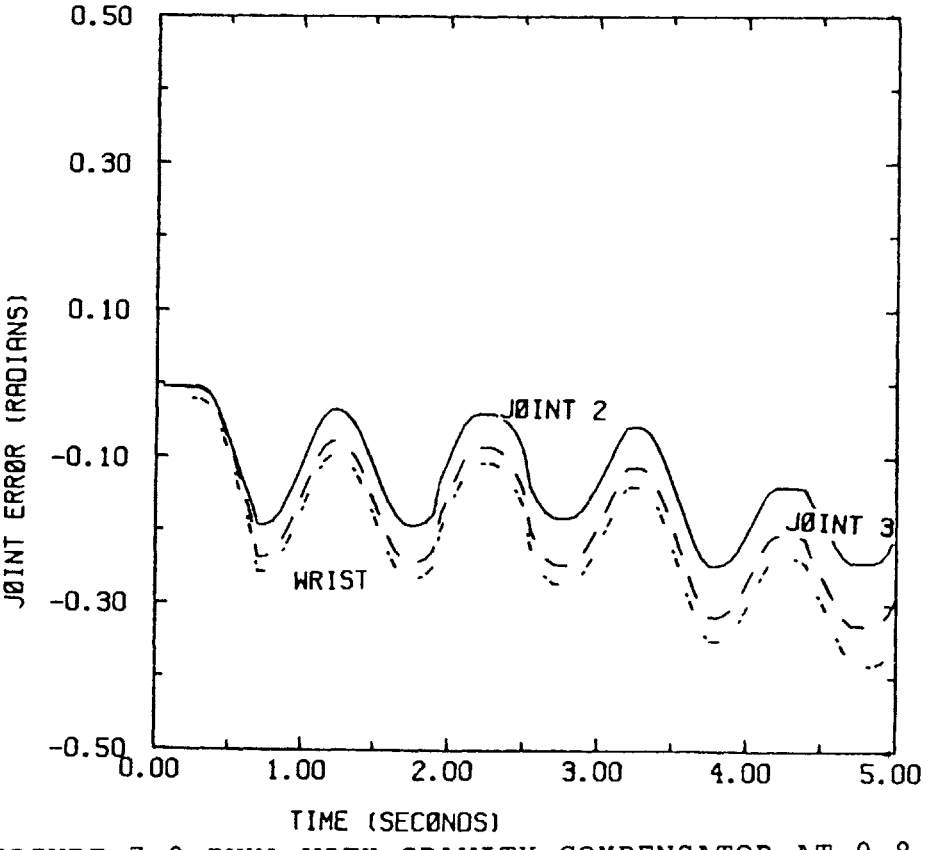

FIGURE 7-8 PUMA WITH GRAVITY COMPENSATOR AT 0.8 HERTZ

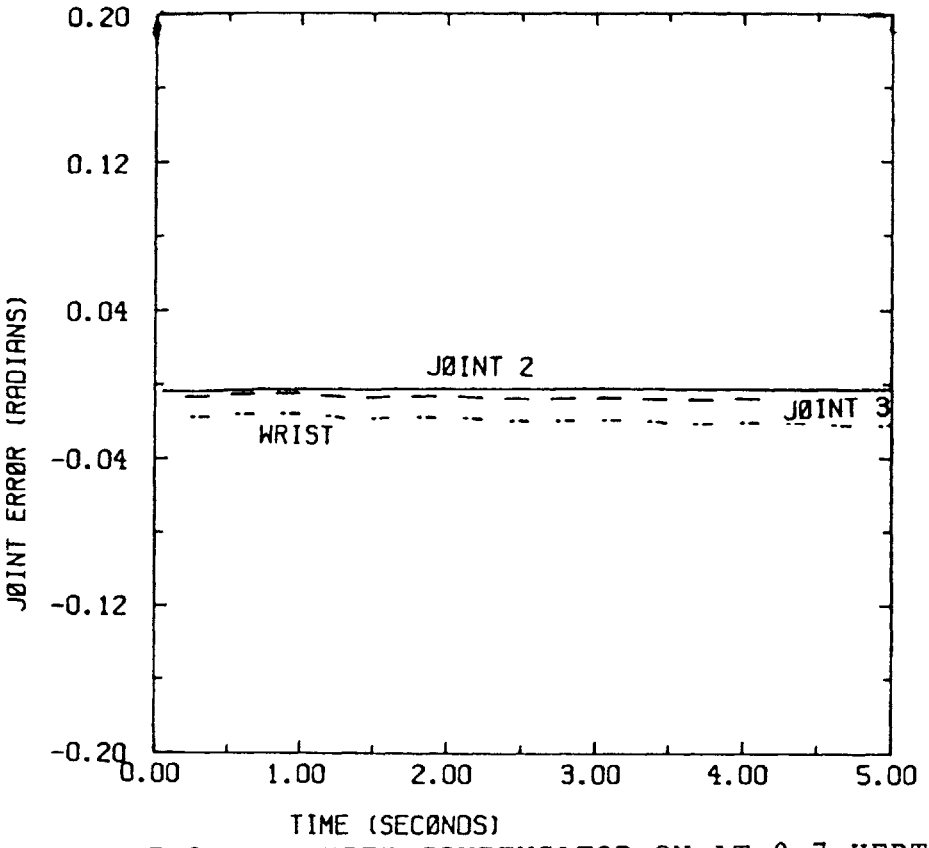

FIGURE 7-9 PUMA WITH COMPENSATOR ON AT 0.7 HERTZ

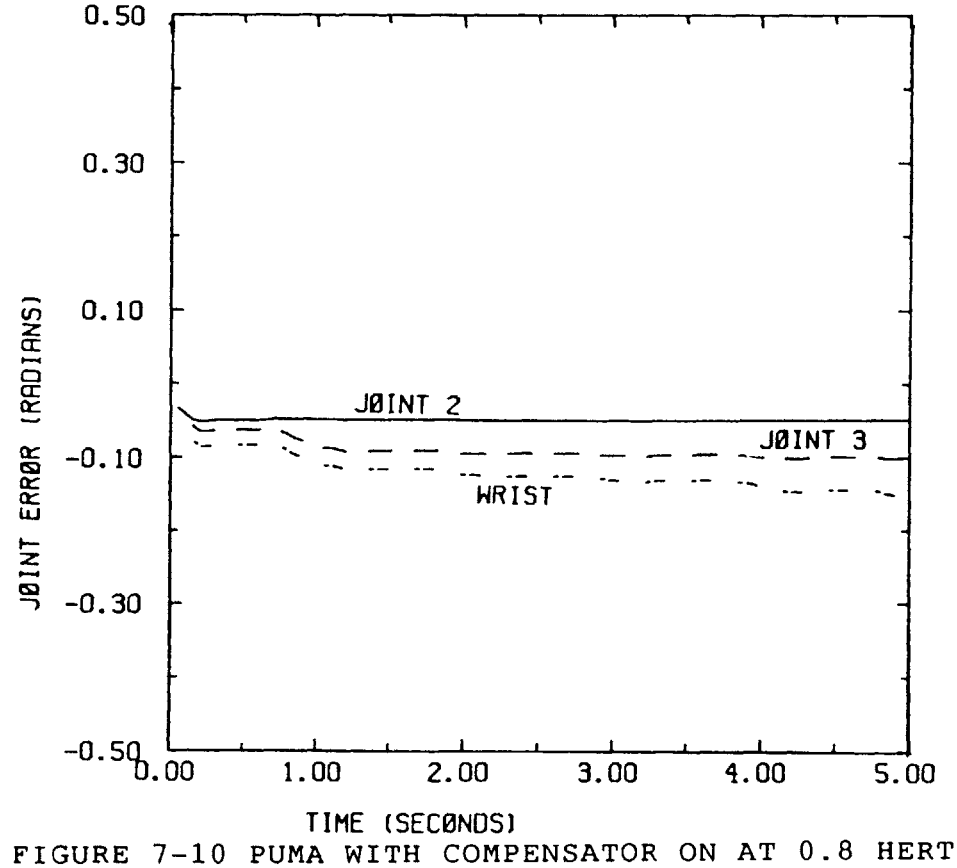

FIGURE 7-10 PUMA WITH COMPENSATOR ON AT 0.8 HERTZ

The next test utilizes the same baseline trajectory used in the digital simulations. The platform is operated with a frequency of 0.7 hertz and an amplitude of 4.5 inches. Figure 7-11 shows the vertical tip error for the uncompensated case. Figure 7-12 shows the same for the compensated case. The compensated error is substantially less during the extraction and insertion phases. This is when the wrist is most susecptible to vertical base motion disturbances, when the wrist is horizontal with a payload. This experimental demonstration shows the benefit of the sensory base motion disturbance compensator.

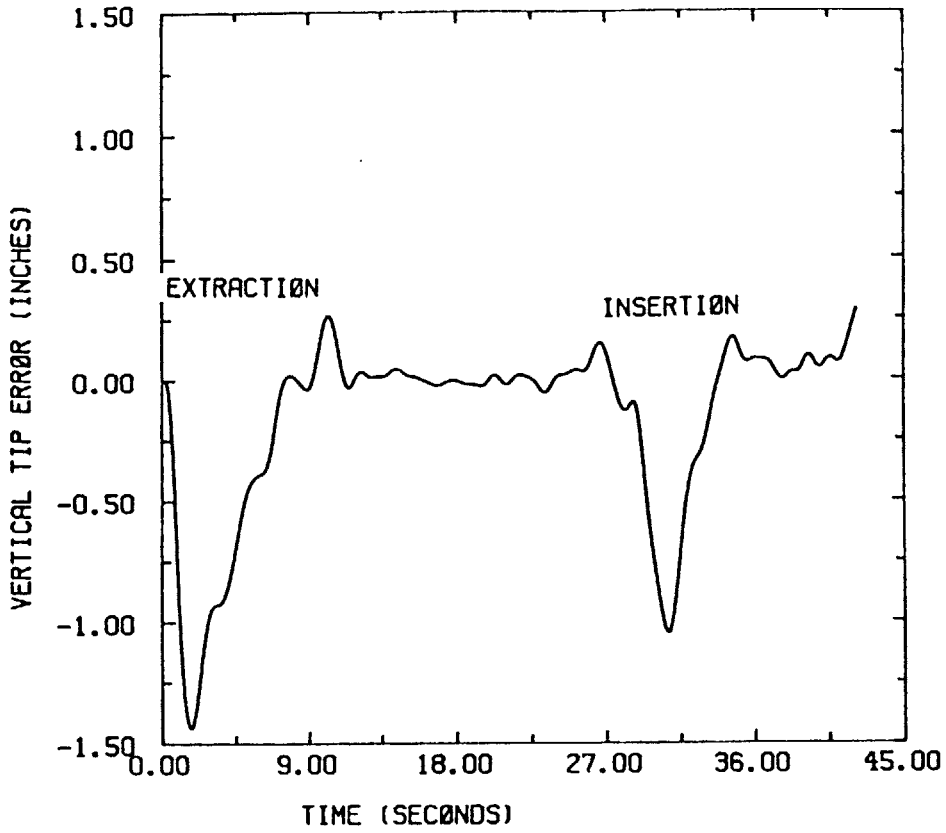

FIGURE 7-11 UNCOMPENSATED PUMA VERTICAL TIP ERROR

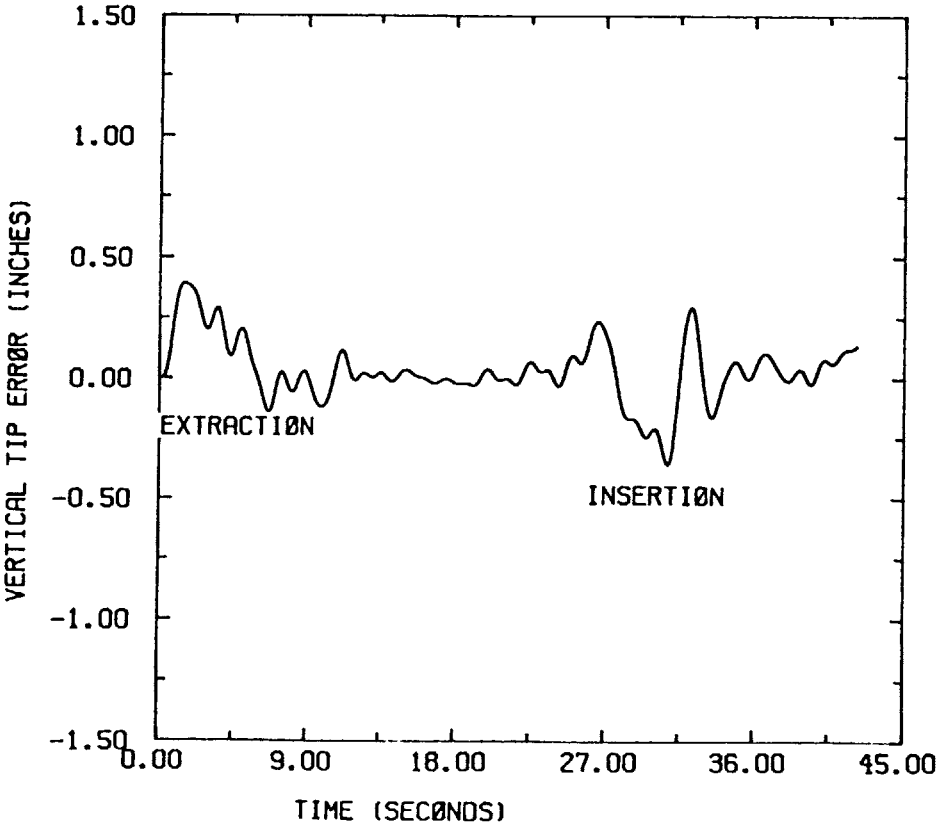

FIGURE 7-12 COMPENSATED PUMA VERTICAL TIP ERROR

CHAPTER 8

CONCLUSIONS

8.1 SUMMARY

This thesis has shown that control problems arise from mounting a robot on a mobile platform. The base motion disturbances create dynamic forces in the robot which degrades the accuracy of the robot joints and the robot's ability to accurately follow a prescribed path. Linear PD controllers, both uncoupled and coupled, are designed which satisfactorily moves a stationary robotic loader over a range of operating values. During base motion disturbances, both controllers have equally reduced accuracy. This reinforces previous research that conventional and advanced controllers are unable to adequately compensate for these disturbances. A compensator is designed, which utilizes measured sensory base motion information to calculate the actuator torques necessary to compensate for the adverse effects of the disturbances. Both controllers, with the compensator included, provided practically the same accuracy achieved when the base was stationary. These results are also verified with hardware.

Mobile robots can achieve accuracy and perform precision tasks even when subjected to base motion disturbances. This is accomplished with a compensator utilizing sensory information of the base motion. This compensator is practical and implemented with existing

sensors and computers.

LIST OF REFERENCES

[1] U.S. Department of the Army, Operations, Field Manual 100-5, U.S. Government Printing Office, Washington D.C., 1982.

[2] "Applications of Robotics and Artificial Intelligence to Reduce Risk and Improve Effectiveness: A Study for the United States Army", Published by the National Academy Press, Washington, D.C. 1983.

[3] Lynch, Ricky "Analysis of the Dynamics and Control of a Two Degree of Freedom Robotic Manipulator Mounted on a Moving Base", M.S. Thesis, MIT Cambridge, MA., OCT 1985.

[4] Joshi, J. and Desrochers, A., "Modeling and Control of a Mobile Robot", Rensselaer Polytechnic Institute, N.Y., 1985.

[5] Holly, J. W., "Automatic Ammunition Loading Systems for Large Caliber Weapons Mounted in Close Combat Armored Vehicles", Tank Automotive Command, Warren, Michigan, 1983.

[6] Campbell, R. C., "Design and Development of Diagnostic Measures for Armor Crewman Performance - XM1", U.S. Army Research Institute for Behavioral and Social Sciences, Washington D.C., 1980.

[7] Kruczynski, D. L., "Final Report for Product Improvement Test of Cartridge, 105mm", U.S. Army Test and Evaluation Command, Aberdeen Proving Ground, Maryland, 1983.

[8] Bird-Johnson Company Fluid Power Division, Bird-Johnson Rotary Hydraulic Actuator Catalog, Walpole, MA., 1986.

[9] Ogata, K., Modern Control Engineering, Prentice-Hall, Englewood Cliffs, NJ., 1970

[10] Cannon, R.A. and Schmitz E., "Initial Experiments on the End-point Control of a Flexible One Link Robot", Published in the International Journal of Robotics Research, Fall 1984.

[11] Lance, G.M., Liang, C., and McCleary, M.A., "Integrated Simulation of Vehicular Systems with Stabilization," Proceedings of the Meeting of the Coordinating Group on Modern Control Theory (4th), Rochester, Michigan, 27-28 October 1982.

[12] Musell, R., "Actual Disturbances Measured on the Abrams Tank", United States Army Tank Automotive Command, Warren, Michigan, 1986.

[13] Crandall, S., Karnopp, D., Kurtz, E., and Pridmore-Brown, D., Dynamics of Mechanical and Electromechanical Systems, Robert E. Krieger Publishing Company, Malabar, Florida, 1982.

[14] Asada, H. and Slotine, J.J.E., Robot Analysis and Control, John Wiley and Sons, N.Y., 1986.

[15] Kailath, Thomas, Linear Systems, Prentice-Hall, Englewood Cliffs, NJ., 1980.

[16] D'Azzo, J.J. and Houpis, H.H., Linear Control System Analysis and Design, Conventional and Modern, McGraw-Hill New York, NY., 1981.

[17] MATRIXX User's Guide, Version 4.0, Integrated Systems, Inc., Palo Alto, CA., 1982.

[18] PUMA Robot Series 200 Equipment and Programing Manual, Unimation, Danbury, CT., 1983.

[19] Fresko, Moris, "The Design and Implementation of a Computer Controlled Platform With Variable Admittance", M.S. Thesis, MIT Cambridge, MA., JAN 1986.

[20] Piezontronics, Shock and Vibration Motion Sensor Catalog, PCB Piezontronics Inc, New York, 1985.

[21] Tretter, Steven A., Introduction to Discrete-time Signal Processing, John Wiley and Sons, New York, NY., 1986.

[22] Whaley, J.L., "An Experimental System for Testing Robotic Manipulator Control Alogrithms: A Demonstration of Adaptive Control", M.S. Thesis, MIT Cambridge, MA., NOV 1985.

[24] CRC Standard Mathematical Tables, 22nd Edition, CRC Press, 1974.

[25] Brady, M. et al., Robot Motion Planning and Control, MIT Press, Cambridge, MA., 1982.

[26] Sander, George N. and Erdman, Arthur G., Advanced Mechanism Design: Analysis and Synthesis, Prentice-Hall, Englewood Cliffs, NJ., 1984.

[27] Higdon, Archie, et. al., Mechanics of Materials, John Wiley and Sons, New York, NY., 1976.

APPENDIX A

DETERMINATION OF SYSTEM PARAMETERS

A.1 LINKS

The work space dimensions dictates that the manipulator links each be 3.0 feet long. This length includes the length of the actuators. To obtain an estimate of the link parameters, from which to begin the inerative process of selecting actuators, the actuators are assumed to be included as a section of the link. The links are designed as thin walled hollow steel tubes.

Using an estimation to begin with that a typical payload to manipulator weight ratio is 1 : 10, indicates that each link weigh approximately 250.0 pounds. Exact link parameters are determined iteratively to achieve the desired natural frequency.

Link parameters are calculated based on the dimensions indicated in figures A-1 and A-2.

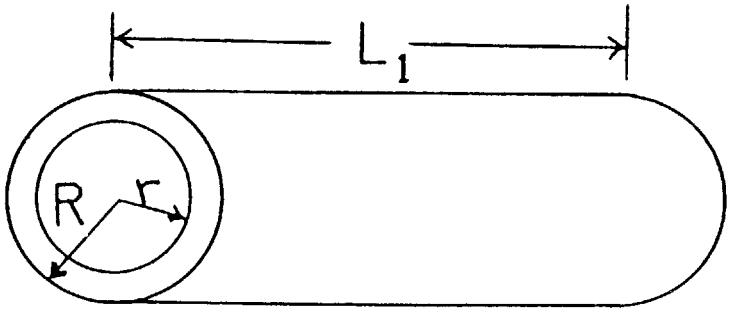

FIGURE A-1 LINK 1

where: R is the outside radius of the link
= 4.5 inches

r is the inside radius of the link
= 3.5 inches

L is the total link length (includes actuators)
= 36.0 inches

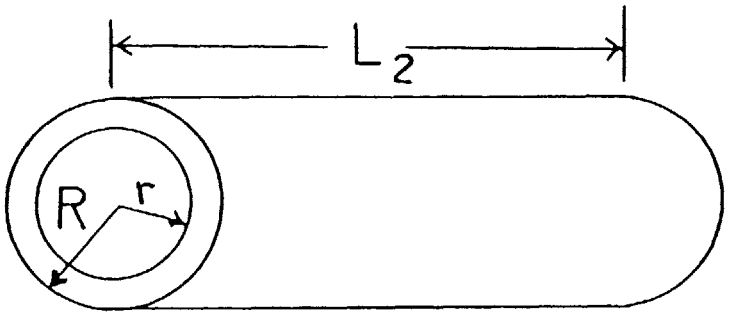

FIGURE A-2 LINK 2

where: R = 4.0 inches

r = 3.3 inches

L = 36.0 inches

Volume calculations are conducted using equation A.1.

$$V = (R^2L - r^2L)\pi \qquad (A.1)$$

which result in

$$V_1 = 882.459 \text{ inches}^3$$

$$V2 = 702.334 \text{ inches}^3$$

Weight calculations are made using equation A.2.

$$W = pV \qquad (A.2)$$

where : p is the density of steel

$$= .284 \text{ lb/in}^3$$

which results in

$$W_1 = 250.5 \text{ pounds}$$

$$W_2 = 199.5 \text{ pounds}$$

These values are used as estimates to determine the required torques for selection of actuators and calculation of natural frequencies, after which the exact link parameters are determined.

A.2 SELECTION OF ACTUATORS

The models of the actuator motors are required to obtain an accurate model of the system. Due to the relatively heavy payloads, hydraulic rotary actuators are used. Hydraulic actuators have a high torque to weight ratio and do not require troublesome transmissions. Additionally, a hydraulic power system already exists in tanks, so selection of hydraulic actuators is realistic. To select actuators which meet this system's requirements, an assumption is made that one third of the availabe actuator torque is used to support the link under gravity, one third for joint accelerations needed to follow a trajectory, and one third is for disturbance rejection. Torque requirements

for the manipulator arm held horizontal are calculated and the appropriate actuator selected.

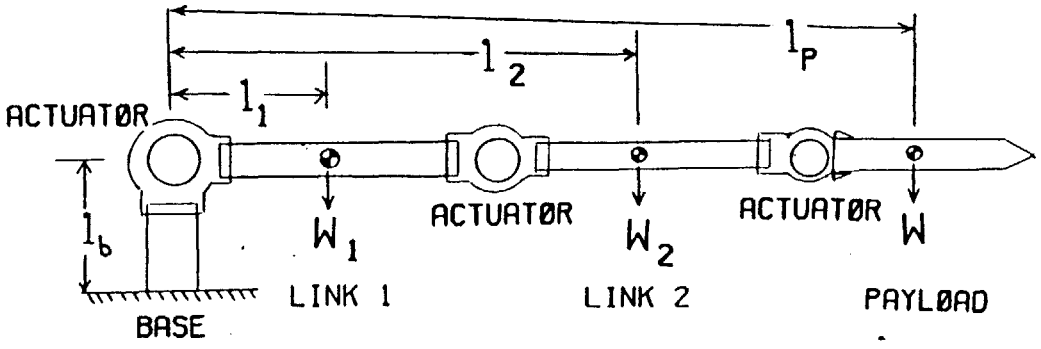

FIGURE A-3 HORIZONTAL ARM

Center of masses are assumed to be centered for the torque calculations. The resulting torques needed to support the arm are calcuated in equations A.3, A.4 and A.5.

Joint 3 : 12(50) = 600 in lb (A.3)

Joint 2 : [48(60) + 18(199.5)] = 6471 in lb (A.4)

Joint 1 : [84(60)+54(199.5)+18(250.5)]= 20322 in lb (A.5)

These torques required to support the robot's arm under gravity provide an estimate to select the actuators. The actuator requirements are found by multipling by a factor of three, equations A.6, A.7 and A.8.

Joint 3 : 600(3) = 1800 in lb (A.6)

Joint 2 : 6471(3) = 19413 in lb (A.7)

Joint 1 : 20322(3) = 60966 in lb (A.8)

These torque requirements necessitate the following actuators, selected from commerially available Bird-Johnson Hydraulic Rotary Actuators [8]:

Joint 3 : Model SS 3/3 a 3 degree of freedom wrist and end effector with maximum torque capability of 6840 inch pounds in all directions and weight of 43 pounds.

Joint 2 : Model SS 4/8 with maximum torque capability of 21700 inch pounds and weight of 40 pounds.

Joint 1 : Model SS 12/25 with maximum torque capability of 70965 inch pounds and weight of 100 pounds.

The assumption regarding dynamic loads and disturbance rejection is checked by digital simulations, presented in section 5.2. At no time did the actuators saturate their capabilities, even when subject to disturbances. The following installed weights for actuators, mounting hardware, and hoses with fluid are used in the model with these weight distributions:

Joint 1: Total weight 150 lbs

Distribution .75 on base and .25 on link 1

Joint 2: Total weight 80 lbs

Distribution .67 on link 1 and .33 on link 2

Joint 3: Total weight 50 lbs

Distribution .5 on link 2 and .5 on payload

A.3 MODEL PARAMETERS

After the actual actuator information is obtained, the actual link parameters is determined. Maintaining the necessary robotic loader reach, the links are now reduced in length the amount of the actuators. This results in the following calculations:

Link volume: $V = (R^2 - r^2)\pi L$ (A.9)

$$V_1 = (4.4^2 - 3.4^2)\pi 30.0 = 735.13 \text{ inches}^3$$
$$V_2 = (3.9^2 - 3.0^2)\pi 30.0 = 585.28 \text{ inches}^3$$

Link weight: $W = pV$ $(p = .284 \text{ lbs/in}^3)$ (A.10)

$$W_1 = .284(735.13) = 208.78 \text{ lbs}$$
$$W_2 = .284(585.28) = 166.22 \text{ lbs}$$

Link mass: $M = W/g$ $(g = 32.174 \text{ ft/sec}^2)$ (A.11)

$$M_1 = 208.78/32.174 = 6.49 \text{ slugs}$$

$M2 = 166.22/32.174 = 5.17$ slugs

$M_p = 60.0/32.174 = 1.86$ slugs

Link mass: (including actuators)

$M_1 = 6.49 + 37.5/g + 53.6/g = 9.32$ slugs

$M_2 = 5.17 + 26.4/g + 25/g = 6.77$ slugs

$M_p = 1.86 + 25/g = 2.64$ slugs (with payload)

$= 0.78$ slugs (without payload)

Center of mass: $l_i = SM_j l_j / M_i$ (A.12)

where: M_i = total link mass

M_j = jth link mass

l_j = jth mass length

$l_1 = [2(1.17)+18(6.49)+34(1.67)]/9.32 = 18.86$ inches

$l_2 = [2(.82)+18(5.17)+34(.78)]/6.77 = 17.89$ inches

$l_p = [2(.78) + 12(1.86)]/2.67 = 8.94$ inches (with pay)

$= [2(.78)]/.78 = 2.0$ inches (without payload)

Moments of inertia: $I = 1/12\ M\ [3(R^2 - r^2) + L^2]$ (A.13)

$I_1 = 1/12\ (9.32)[3(4.4^2 - 3.4^2) + 36^2] = 1024.73$

$= 7.116$ lbf sec^2 ft

$I_2 = 1/12\ (6.77)[3(3.9^2 - 3.0^2) + 36^2] = 741.67$

$= 5.150$ lbf sec^2 ft

$$I_p = 1/12\ (2.64)[3(4.5^2) + 30^2] = 211.365$$

$$= 1.468 \text{ lbf sec}^2 \text{ ft}$$

$$= 1/12\ (0.78)[3(4.5^2) + 3^2] = 4.534$$

$$= 0.031 \text{ lbf sec}^2 \text{ ft}$$

With these actuator weights, weight distributions, and sizes, the model link parameters are determined and shown in figure A-4.

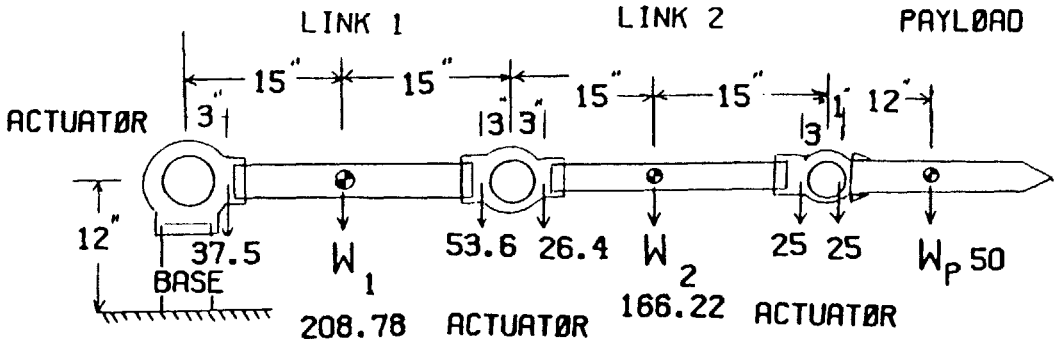

FIGURE A-4 FINAL SYSTEM WEIGHTS/DISTRIBUTIONS

APPENDIX B

STRUCTURAL FREQUENCY

To properly construct a controller, the mechanism's structural frequency must be known, or at least bounded. The controller frequency needs to be lower than the structural frequency to prevent exciting the mechanism's structural resonances.

A lumped parameter method, with the arm held horizontal is used to get an estimate of the natural structural resonances. The following figure shows the model with the weights of each element at their center of gravity:

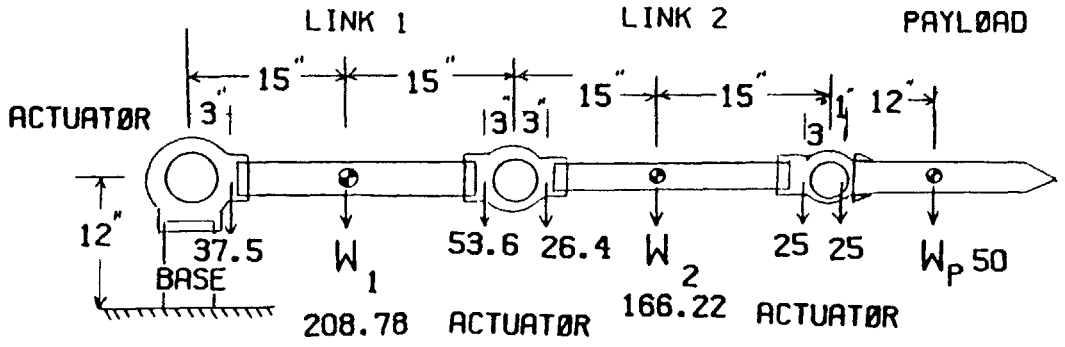

FIGURE B-1 MODEL OF ROBOT ARM

Beam theory, assuming locked actuators, and lumped modeling is used to determine effective weights to use in the equation B.1.

$$w_n = \sqrt{\frac{3EIg}{PL^3}} \qquad (B.1)$$

where: $E = 29 \text{ ksi} \times 10^3$

$g = 32.174 \text{ ft/sec}^2$

P = effective weight

L = extended length

I = moment of inertia

Effective weights, lumped to the end of the arm are calculated using equation B.2 [24,25].

$$W_i(\text{effective}) = n^6 W_i \qquad (B.2)$$

n = ratio of cg length to total length

This results in a natural structural frequency of

$w_n = 14.73$ hertz (with payload)

$w_n = 23.51$ hertz (without payload)

APPENDIX C

TRAJECTORY

To generate the joint angle commands required to produce the desired trajectory, the motion in workspace coordinates (X,Y), is converted to joint coordinates (Θ_1,Θ_2). This is accomplished using trigometric functions.

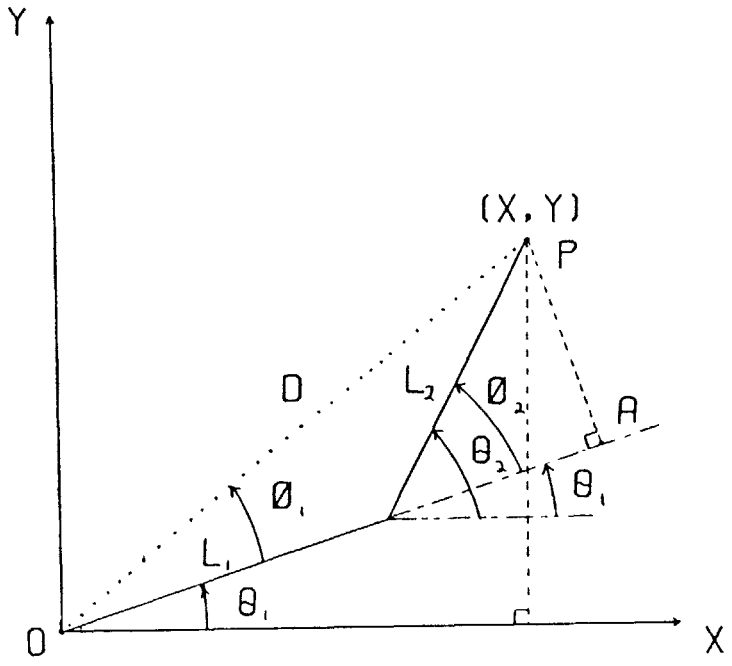

FIGURE C-1 INVERSE KINEMATICS

The distance D can be used to relate X and Y with $\emptyset_2$, which is a function of Θ_1 and Θ_2. The right triangle (O,x,y) gives the relationship

$$D^2 = X^2 + Y^2 \qquad (C.1)$$

By forming the right triangle (O,P,A), the relationship involving $\emptyset_2$ is found

$$D^2 = (L_1 + L_2\cos\emptyset_2)^2 + (L_2\sin\emptyset_2)^2 \qquad (C.2)$$

Solving for $\emptyset_2$ gives

$$\emptyset_2 = \cos^{-1}\left[\frac{D^2 - L_1^2 - L_2^2}{2L_1L_2}\right] \qquad (C.3)$$

$$= \cos^{-1}\left[\frac{X^2 + Y^2 - L_1^2 - L_2^2}{2L_1L_2}\right] \qquad (C.4)$$

To determine a relationship for θ_1

$$\theta_1 = (\emptyset_1 + \theta_1) - \emptyset_1 \qquad (C.5)$$

where:

$$\tan(\emptyset_1 + \theta_1) = Y/X \qquad (C.6)$$

and

$$\tan\emptyset_1 = \frac{L_2\sin\emptyset_2}{L_1 + L_2\cos\emptyset_2} \qquad (C.7)$$

Combining gives

$$\theta_1 = \tan^{-1}\left[\frac{Y}{X}\right] - \tan^{-1}\left[\frac{L_2\sin\emptyset_2}{L_1 + L_2\cos\emptyset_2}\right] \qquad (C.8)$$

and

$$\theta_2 = \emptyset_2 - \theta_1 \qquad (C.9)$$

Similarly the joint velocities are found

$$\dot{\theta}_1 = \frac{-L_1\cos\theta_2}{L_1L_2\sin\phi_2}X + \frac{L_2\sin\theta_2}{L_1L_2\sin\phi_2}Y \qquad (C.10)$$

$$\dot{\phi}_2 = \frac{-L_1\cos\theta_1 - L_2\cos\theta_2}{L_1L_2\sin\phi_2}X - \frac{L_1\sin\theta_1 + L_2\sin\theta_2}{L_1L_2\sin\phi_2}Y \qquad (C.11)$$

$$\dot{\theta}_2 = \dot{\theta}_1 + \dot{\phi}_2 \qquad (C.12)$$

The trajectory is separated into path segments. Then equations for each segment are developed which relate the workspace coordinates (X,Y) with time. A FORTRAN program was written to conduct the calculations. This program, TRAJ, is listed in Appendix E.

APPENDIX D

EQUATIONS OF MOTION

D.1 NONLINEAR EQUATIONS

Generalized forces:

$$Q_1 = -T_1(\partial\theta_b/\partial\theta_1) + T_1(\partial\theta_1/\partial\theta_1) - T_2(\partial\theta_1/\partial\theta_1) + T_2(\partial\theta_2/\partial\theta_1) - T_3(\partial\theta_2/\partial\theta_1) + T_3(\partial\theta_w/\partial\theta_1) \quad (D.1)$$

$$Q_2 = -T_1(\partial\theta_b/\partial\theta_2) + T_1(\partial\theta_1/\partial\theta_2) - T_2(\partial\theta_1/\partial\theta_2) + T_2(\partial\theta_2/\partial\theta_2) - T_3(\partial\theta_2/\partial\theta_2) + T_3(\partial\theta_w/\partial\theta_2) \quad (D.2)$$

$$Q_3 = -T_1(\partial\theta_b/\partial\theta_w) + T_1(\partial\theta_1/\partial\theta_w) - T_2(\partial\theta_1/\partial\theta_w) + T_2(\partial\theta_2/\partial\theta_w) - T_3(\partial\theta_2/\partial\theta_w) + T_3(\partial\theta_w/\partial\theta_w) \quad (D.3)$$

which results in:

$$Q_1 = T_1 - T_2 \quad (D.4)$$

$$Q_2 = T_2 - T_3 \quad (D.5)$$

$$Q_3 = T_3 \quad (D.6)$$

Velocity components:

link 1; X: $-L_b\dot{\theta}_b\sin\theta_b - l_1\dot{\theta}_1\sin\theta_1$

Y: $\dot{Y} + L_b\dot{\theta}_b\cos\theta_b + l_1\dot{\theta}_1\cos\theta_1$

link 2; X: $-L_b\dot{\theta}_b\sin\theta_b - L_1\dot{\theta}_1\sin\theta_1 - l_2\dot{\theta}_2\sin\theta_2$

Y: $\dot{Y} + L_b\dot{\theta}_b\cos\theta_b + L_1\dot{\theta}_1\cos\theta_1 + l_2\dot{\theta}_2\cos\theta_2$

link p; X: $-L_b\dot{\theta}_b\sin\theta_b - L_1\dot{\theta}_1\sin\theta_1 - L_2\dot{\theta}_2\sin\theta_2 - l_p\dot{\theta}_w\sin\theta_w$

Y: $\dot{Y} + L_b\dot{\theta}_b\cos\theta_b + L_1\dot{\theta}_1\cos\theta_1 + L_2\dot{\theta}_2\cos\theta_2 + l_p\dot{\theta}_w\cos\theta_w$

Substitution into kinetic energy equation and expanding:

$$T = 1/2\ M_1[L_b^2\dot{\theta}_b^2\sin^2\theta_b + l_1^2\dot{\theta}_1^2\sin^2\theta_1 + 2L_bl_1\dot{\theta}_b\dot{\theta}_1\sin\theta_b\sin\theta_1 + \dot{Y}^2 + L_b^2\dot{\theta}_b^2\cos^2\theta_b + l_1^2\dot{\theta}_1^2\cos^2\theta_1 + 2\dot{Y}L_b\dot{\theta}_b\cos\theta_b + 2\dot{Y}l_1\dot{\theta}_1\cos\theta_1 + 2L_bl_1\dot{\theta}_b\dot{\theta}_1\cos\theta_b\cos\theta_1] + 1/2\ I_1\dot{\theta}_1^2$$
$$+ 1/2\ M_2[l_b^2\dot{\theta}_b^2\sin^2\theta_b + L_1^2\dot{\theta}_1^2\sin^2\theta_1 + l_2^2\dot{\theta}_2^2\sin^2\theta_2 + 2L_bl_1\dot{\theta}_b\dot{\theta}_1\sin\theta_b\sin\theta_1 + 2L_bl_2\dot{\theta}_b\dot{\theta}_2\sin\theta_b\sin\theta_2 + 2L_1l_2\dot{\theta}_1\dot{\theta}_2\sin\theta_1\sin\theta_2 + \dot{Y}^2 + L_b^2\dot{\theta}_b^2\cos^2\theta_b + L_1^2\dot{\theta}_1^2\cos^2\theta_1 + l_2^2\dot{\theta}_2^2\cos^2\theta_2 + 2\dot{Y}L_b\dot{\theta}_b\cos\theta_b + 2\dot{Y}L_1\dot{\theta}_1\cos\theta_1 + 2\dot{Y}l_2\dot{\theta}_2\cos\theta_2 + 2L_bL_1\dot{\theta}_b\dot{\theta}_1\cos\theta_b\cos\theta_1 + 2L_bl_2\dot{\theta}_b\dot{\theta}_2\cos\theta_b\cos\theta_2 + 2L_1l_2\dot{\theta}_1\dot{\theta}_2\cos\theta_1\cos\theta_2] + 1/2\ I_2\dot{\theta}_2^2 + 1/2\ M_p[L_b^2\dot{\theta}_b^2\sin^2\theta_b + L_1^2\dot{\theta}_1^2\sin^2\theta_1 + L_2^2\dot{\theta}_2^2\sin^2\theta_2 + l_p^2\dot{\theta}_w^2\sin^2\theta_w + 2L_bL_1\dot{\theta}_b\dot{\theta}_1\sin\theta_b\sin\theta_1 + 2L_bL_2\dot{\theta}_b\dot{\theta}_2\sin\theta_b\sin\theta_2 + 2L_bl_p\dot{\theta}_b\dot{\theta}_w\sin\theta_b\sin\theta_w + 2L_1L_2\dot{\theta}_1\dot{\theta}_2\sin\theta_1\sin\theta_2 + 2L_1l_p\dot{\theta}_1\dot{\theta}_w\sin\theta_1\sin\theta_w + 2L_2l_p\dot{\theta}_2\dot{\theta}_w\sin\theta_2\sin\theta_w + \dot{Y}^2 + L_b^2\dot{\theta}_b^2\cos^2\theta_b + L_1^2\dot{\theta}_1^2\cos^2\theta_1 + L_2^2\dot{\theta}_2^2\cos^2\theta_2 + l_p^2\dot{\theta}_w^2\cos^2\theta_w + 2\dot{Y}L_b\dot{\theta}_b\cos\theta_b + 2\dot{Y}L_1\dot{\theta}_1\cos\theta_1 + 2\dot{Y}L_2\dot{\theta}_2\cos\theta_2 + 2\dot{Y}l_p\dot{\theta}_w\cos\theta_w + 2L_bL_1\dot{\theta}_b\dot{\theta}_1\cos\theta_b\cos\theta_1 + 2L_bL_2\dot{\theta}_b\dot{\theta}_2\cos\theta_b\cos\theta_2 + 2L_bl_p\dot{\theta}_b\dot{\theta}_w\cos\theta_b\cos\theta_w + 2L_1L_2\dot{\theta}_1\dot{\theta}_2\cos\theta_1\cos\theta_2 + 2L_1l_p\dot{\theta}_1\dot{\theta}_w\cos\theta_1\cos\theta_w + 2L_2l_p\dot{\theta}_2\dot{\theta}_w\cos\theta_2\cos\theta_w] + 1/2\ I_w\dot{\theta}_w^2 \tag{D.7}$$

Simplifying using trigonometry identities [23]:

$$A\cos^2\theta + A\sin^2\theta = A \tag{D.8}$$

$$\cos\theta\cos\phi + \sin\theta\sin\phi = \cos(\theta-\phi) \qquad (D.9)$$

$$T = 1/2\ M_1[L_b^2\dot{\theta}_b^2 + l_1^2\dot{\theta}_1^2 + \dot{Y}^2 + 2L_b l_1\cos(\theta_b-\theta_1)\dot{\theta}_b\dot{\theta}_1 + 2\dot{Y}L_b\cos\theta_b\dot{\theta}_b + 2\dot{Y}l_1\cos\theta_1\dot{\theta}_1] + 1/2\ I_1\dot{\theta}_1^2 + 1/2\ M_2[L_b^2\dot{\theta}_b^2 + L_1^2\dot{\theta}_1^2 + l_2^2\dot{\theta}_2^2 + \dot{Y}^2 + 2L_bL_1\cos(\theta_b-\theta_1)\dot{\theta}_b\dot{\theta}_1 + 2L_b l_2\cos(\theta_b-\theta_2)\dot{\theta}_b\dot{\theta}_2 + 2L_1 l_2\cos(\theta_1-\theta_2)\dot{\theta}_1\dot{\theta}_2 + 2\dot{Y}L_b\cos\theta_b\dot{\theta}_b + 2\dot{Y}L_1\cos\theta_1\dot{\theta}_1 + 2\dot{Y}l_2\cos\theta_2\dot{\theta}_2] + 1/2\ I_2\dot{\theta}_2^2 + 1/2\ M_p[L_b^2\dot{\theta}_b^2 + L_1^2\dot{\theta}_1^2 + L_2^2\dot{\theta}_2^2 + l_p^2\dot{\theta}_w^2 + \dot{Y}^2 + 2L_bL_1\cos(\theta_b-\theta_1)\dot{\theta}_b\dot{\theta}_1 + 2L_bL_2\cos(\theta_b-\theta_2)\dot{\theta}_b\dot{\theta}_2 + 2L_b l_p\cos(\theta_b-\theta_w)\dot{\theta}_b\dot{\theta}_w + 2L_1L_2\cos(\theta_1-\theta_2)\dot{\theta}_1\dot{\theta}_2 + 2L_1 l_p\cos(\theta_1-\theta_w)\dot{\theta}_1\dot{\theta}_w + 2L_2 l_p\cos(\theta_2-\theta_w)\dot{\theta}_2\dot{\theta}_w + 2L_b\cos\theta_b\dot{Y}\dot{\theta}_b + 2L_1\cos\theta_1\dot{Y}\dot{\theta}_1 + 2L_2\cos\theta_2\dot{Y}\dot{\theta}_2 + 2l_p\cos\theta_w\dot{Y}\dot{\theta}_w] + 1/2\ I_p\dot{\theta}_w^2$$

(D.10)

Taking the partial derivatives to substitute into Lagranges equation:

$$\partial T/\partial\theta_1 = 1/2\ M_1[2L_b l_1\sin(\theta_b-\theta_1)\dot{\theta}_b\dot{\theta}_1 - 2l_1\sin\theta_1\dot{Y}\dot{\theta}_1] + 1/2\ M_2[2L_bL_1\sin(\theta_b-\theta_1)\dot{\theta}_b\dot{\theta}_1 - 2L_1 l_2\sin(\theta_1-\theta_2)\dot{\theta}_1\dot{\theta}_2 - 2L_1\sin\theta_1\dot{Y}\dot{\theta}_1] + 1/2\ M_p[2L_bL_1\sin(\theta_b-\theta_1)\dot{\theta}_b\dot{\theta}_1 - 2L_1L_2\sin(\theta_1-\theta_2)\dot{\theta}_1\dot{\theta}_2 - 2L_1 l_p\sin(\theta_1-\theta_w)\dot{\theta}_1\dot{\theta}_w - 2L_1\sin\theta_1\dot{Y}\dot{\theta}_1]$$

(D.11)

$$\partial T/\partial\theta_2 = 1/2\ M_2[2L_b l_2\sin(\theta_b-\theta_2)\dot{\theta}_b\dot{\theta}_2 + 2L_1 l_2\sin(\theta_1-\theta_2)\dot{\theta}_1\dot{\theta}_2 - 2l_2\sin\theta_2\dot{Y}\dot{\theta}_2] + 1/2\ M_p[2L_bL_2\sin(\theta_b-\theta_2)\dot{\theta}_b\dot{\theta}_2 + 2L_1 l_2\sin(\theta_1-\theta_2)\dot{\theta}_1\dot{\theta}_2 - 2L_2 l_p\sin(\theta_2-\theta_w)\dot{\theta}_2\dot{\theta}_w - 2L_2\sin\theta_2\dot{Y}\dot{\theta}_2]$$

(D.12)

$$\partial T/\partial \theta_w = 1/2\ M_p[2L_b l_p \sin(\theta_b-\theta_w)\dot{\theta}_b\dot{\theta}_w + 2L_1 l_p \sin(\theta_1-\theta_w)\dot{\theta}_1\dot{\theta}_w - 2l_p \sin\theta_w \dot{Y}\dot{\theta}_w + 2L_2 l_p \sin(\theta_2-\theta_w)\dot{\theta}_2\dot{\theta}_w]$$

(D.13)

$$\partial T/\partial \dot{\theta}_1 = 1/2\ M_1[2l_1^2\dot{\theta}_1 + 2L_b l_1 \cos(\theta_b-\theta_1)\dot{\theta}_b + 2l_1 \cos\theta_1 \dot{Y}] + I_1\dot{\theta}_1 + 1/2\ M_2[2L_1^2\dot{\theta}_1 + 2L_b L_1 \cos(\theta_b-\theta_1)\dot{\theta}_b + 2L_1 l_2 \cos(\theta_1-\theta_2)\dot{\theta}_2 + 2L_1 \cos\theta_1 \dot{Y}] + 1/2M_p[2L_1^2\dot{\theta}_1 + 2L_b L_1 \cos(\theta_b-\theta_1)\dot{\theta}_b + 2L_1 L_2 \cos(\theta_1-\theta_2)\dot{\theta}_2 + 2L_1 \cos\theta_1 \dot{Y} + 2L_1 l_p \cos(\theta_1-\theta_w)\dot{\theta}_w]$$

(D.14)

$$\partial T/\partial \dot{\theta}_2 = 1/2\ M_2[2l_2^2\dot{\theta}_2 + 2L_b l_2 \cos(\theta_b-\theta_2)\dot{\theta}_b + 2L_1 l_2 \cos(\theta_1-\theta_2)\dot{\theta}_1 + 2l_2 \cos\theta_2 \dot{Y}] + I_2\dot{\theta}_2 + 1/2M_p[2L_2^2\dot{\theta}_2 + 2L_b l_2 \cos(\theta_b-\theta_2)\dot{\theta}_b + 2L_1 l_2 \cos(\theta_1-\theta_2)\dot{\theta}_1 + 2L_2 \cos\theta_2 \dot{Y} + 2L_2 l_p \cos(\theta_2-\theta_w)\dot{\theta}_w]$$

(D.15)

$$\partial T/\partial \dot{\theta}_w = 1/2\ M_p[2l_p^2\dot{\theta}_w + 2L_b l_p \cos(\theta_b-\theta_w)\dot{\theta}_b + 2L_1 l_p \cos(\theta_1-\theta_w)\dot{\theta}_1 + 2L_2 l_p \cos(\theta_2-\theta_w)\dot{\theta}_2 + 2l_p \cos\theta_w \dot{Y}] + I_p\dot{\theta}_w$$

(D.16)

$$d/dt[\partial T/\partial \dot{\theta}_1] = 1/2\ M_1[2l_1^2\ddot{\theta}_1 + 2L_b l_1 \cos(\theta_b-\theta_1)\ddot{\theta}_b - 2L_b l_1 \sin(\theta_b-\theta_1)(\dot{\theta}_b-\dot{\theta}_1)\dot{\theta}_b + 2l_1 \cos\theta_1 \ddot{Y} - 2l_1 \sin\theta_1 \dot{Y}\dot{\theta}_1] + I_1\ddot{\theta}_1 + 1/2M_2[2L_1^2\ddot{\theta}_1 + 2L_b L_1 \cos(\theta_b-\theta_1)\ddot{\theta}_b - 2L_b L_1 \sin(\theta_b-\theta_1)(\dot{\theta}_b-\dot{\theta}_1)\dot{\theta}_b + 2L_1 l_2 \cos(\theta_1-\theta_2)\ddot{\theta}_2 - 2L_1 l_2 \sin(\theta_1-\theta_2)(\dot{\theta}_1-\dot{\theta}_2)\dot{\theta}_2 + 2L_1 \cos\theta_1 \ddot{Y} -$$

$$2L_1l_2\cos(\theta_1-\theta_2)\ddot{\theta}_2 - 2L_1l_2\sin(\theta_1-\theta_2)(\dot{\theta}_1-\dot{\theta}_2)\dot{\theta}_2 + 2L_1\cos\theta_1\ddot{Y} - 2L_1\sin\theta_1\dot{Y}\dot{\theta}_1] + 1/2M_p[2L_1^2\ddot{\theta}_1 + 2L_bL_1\cos(\theta_b-\theta_1)\ddot{\theta}_b - 2L_bL_1\sin(\theta_b-\theta_1)(\dot{\theta}_b-\dot{\theta}_1)\dot{\theta}_b + 2L_1L_2\cos(\theta_1-\theta_2)\ddot{\theta}_2 - 2L_1L_2\sin(\theta_1-\theta_2)(\dot{\theta}_1-\dot{\theta}_2)\dot{\theta}_2 + 2L_1l_p\cos(\theta_1-\theta_w)\ddot{\theta}_w - 2l_1l_p\sin(\theta_1-\theta_w)(\dot{\theta}_1-\dot{\theta}_w)\dot{\theta}_w + 2L_1\cos\theta_1\ddot{Y} - 2L_1\sin\theta_1\dot{Y}\dot{\theta}_1]$$

(D.17)

$$d/dt[\partial T/\partial\dot{\theta}_2] = 1/2M_2[2l_2^2\ddot{\theta}_2 + 2L_bl_2\cos(\theta_b-\theta_2)\ddot{\theta}_b - 2L_bl_2\sin(\theta_b-\theta_2)(\dot{\theta}_b-\dot{\theta}_2)\dot{\theta}_b + 2L_1l_2\cos(\theta_1-\theta_2)\ddot{\theta}_1 - 2L_1l_2\sin(\theta_1-\theta_2)(\dot{\theta}_1-\dot{\theta}_2)\dot{\theta}_1 + 2l_2\cos\theta_2\ddot{Y} - 2l_2\sin\theta_2\dot{Y}\dot{\theta}_2] + I_2\ddot{\theta}_2 + 1/2M_p[2L_2^2\ddot{\theta}_2 + 2L_bL_2\cos(\theta_b-\theta_2)\ddot{\theta}_b - 2L_bL_2\sin(\theta_b-\theta_2)(\dot{\theta}_b-\dot{\theta}_2)\dot{\theta}_b + 2L_1L_2\cos(\theta_1-\theta_2)\ddot{\theta}_1 - 2L_1L_2\sin(\theta_1-\theta_2)(\dot{\theta}_1-\dot{\theta}_2)\dot{\theta}_1 + 2L_2l_p\cos(\theta_2-\theta_w)\ddot{\theta}_w - 2L_2l_p\sin(\theta_2-\theta_w)(\dot{\theta}_2-\dot{\theta}_w)\dot{\theta}_w + 2L_2\cos\theta_2\ddot{Y} - 2L_2\sin\theta_2\dot{Y}\dot{\theta}_2]$$

(D.18)

$$d/dt[\partial T/\partial\dot{\theta}_w] = 1/2M_p[2l_p^2\ddot{\theta}_w + 2L_bl_p\cos(\theta_b-\theta_w)\ddot{\theta}_b - 2L_bl_p\sin(\theta_b-\theta_w)(\dot{\theta}_b-\dot{\theta}_w)\dot{\theta}_b + 2L_1l_p\cos(\theta_1-\theta_w)\ddot{\theta}_1 - 2L_1l_p\sin(\theta_1-\theta_w)(\dot{\theta}_1-\dot{\theta}_w)\dot{\theta}_1 + 2l_p\cos\theta_w\ddot{Y} + 2L_2l_p\cos(\theta_2-\theta_w)\ddot{\theta}_2 - 2L_2l_p\sin(\theta_2-\theta_w)(\dot{\theta}_2-\dot{\theta}_w)\dot{\theta}_2 - 2l_p\sin\theta_w\dot{Y}\dot{\theta}_w] + I_p\ddot{\theta}_w$$

(D.19)

The potential energy term is:

$$V = M_1gl_1\sin\theta_1 + M_2gL_1\sin\theta_1 + M_pgL_1\sin\theta_1 + M_2gl_2\sin\theta_2 + M_pgL_2\sin\theta_2 + M_pgl_p\sin\theta_w \quad (D.20)$$

Taking the required partial derivatives results in:

$$\partial V/\partial\theta_1 = M_1 g l_1 \cos\theta_1 + M_2 g L_1 \cos\theta_1 + M_p g L_1 \cos\theta_1 \quad (D.21)$$

$$\partial V/\partial\theta_2 = M_2 g l_2 \cos\theta_2 + M_p g L_2 \cos\theta_2 \quad (D.22)$$

$$\partial V/\partial\theta_w = M_p g l_p \cos\theta_w \quad (D.23)$$

Substitution of these quantities into Lagranges equation (3.1) results in the equation of motions (3.5), (3.6), and (3.7).

D.2 LINEARIZED EQUATIONS

Linearizing the non-linear equations of motion is accomplished by Taylor series expansion and ignoring higher order terms. This is completed by substituting the following into the non-linear equations:

$$\theta_1 = \theta_1^{\,n} + \partial\theta_1 \quad (D.24)$$
$$\dot{\theta}_1 = \dot{\theta}_1^{\,n} + \partial\dot{\theta}_1 \quad (D.25)$$
$$\ddot{\theta}_1 = \ddot{\theta}_1^{\,n} + \partial\ddot{\theta}_1 \quad (D.26)$$

$$\theta_2 = \theta_2^{\,n} + \partial\theta_2 \quad (D.27)$$
$$\dot{\theta}_2 = \dot{\theta}_2^{\,n} + \partial\dot{\theta}_2 \quad (D.28)$$
$$\ddot{\theta}_2 = \ddot{\theta}_2^{\,n} + \partial\ddot{\theta}_2 \quad (D.29)$$

$$\theta_w = \theta_w^{\,n} + \partial\theta_w \quad (D.30)$$
$$\dot{\theta}_w = \dot{\theta}_w^{\,n} + \partial\dot{\theta}_w \quad (D.31)$$

$$\ddot{\theta}_w = \ddot{\theta}_w^{\,n} + \partial\ddot{\theta}_w \tag{D.32}$$

This results in the transformation of the following terms:

$$\cos(\theta_1) = \cos\theta_1^{\,n} - \sin\theta_1^{\,n}\partial\theta_1 \tag{D.33}$$
$$\cos(\theta_2) = \cos\theta_2^{\,n} - \sin\theta_2^{\,n}\partial\theta_2 \tag{D.34}$$
$$\cos(\theta_w) = \cos\theta_w^{\,n} - \sin\theta_w^{\,n}\partial\theta_w \tag{D.35}$$

$$\sin(\theta_1) = \sin\theta_1^{\,n} + \cos\theta_1^{\,n}\partial\theta_1 \tag{D.36}$$
$$\sin(\theta_2) = \sin\theta_2^{\,n} + \cos\theta_2^{\,n}\partial\theta_2 \tag{D.37}$$
$$\sin(\theta_w) = \sin\theta_w^{\,n} + \cos\theta_w^{\,n}\partial\theta_w \tag{D.38}$$

$$\cos(\theta_b-\theta_1) = \cos(\theta_b-\theta_1^{\,n}) + \sin(\theta_b-\theta_1^{\,n})\partial\theta_1 \tag{D.39}$$
$$\cos(\theta_b-\theta_2) = \cos(\theta_b-\theta_2^{\,n}) + \sin(\theta_b-\theta_2^{\,n})\partial\theta_2 \tag{D.40}$$
$$\cos(\theta_b-\theta_w) = \cos(\theta_b-\theta_w^{\,n}) + \sin(\theta_b-\theta_w^{\,n})\partial\theta_w \tag{D.41}$$

$$\cos(\theta_1-\theta_2) = \cos(\theta_1^{\,n}-\theta_2^{\,n}) + \sin(\theta_1^{\,n}-\theta_2^{\,n})(\partial\theta_2-\partial\theta_1) \tag{D.42}$$
$$\cos(\theta_1-\theta_w) = \cos(\theta_1^{\,n}-\theta_w^{\,n}) + \sin(\theta_1^{\,n}-\theta_w^{\,n})(\partial\theta_w-\partial\theta_1) \tag{D.43}$$
$$\cos(\theta_2-\theta_w) = \cos(\theta_2^{\,n}-\theta_w^{\,n}) + \sin(\theta_2^{\,n}-\theta_w^{\,n})(\partial\theta_w-\partial\theta_2) \tag{D.44}$$

$$\sin(\theta_b-\theta_1) = \sin(\theta_b-\theta_1^{\,n}) - \cos(\theta_b-\theta_1^{\,n})\partial\theta_1 \tag{D.45}$$
$$\sin(\theta_b-\theta_2) = \sin(\theta_b-\theta_2^{\,n}) - \cos(\theta_b-\theta_2^{\,n})\partial\theta_2 \tag{D.46}$$
$$\sin(\theta_b-\theta_w) = \sin(\theta_b-\theta_w^{\,n}) - \cos(\theta_b-\theta_w^{\,n})\partial\theta_w \tag{D.47}$$

$$\sin(\theta_1-\theta_2) = \sin(\theta_1^{\,n}-\theta_2^{\,n}) + \cos(\theta_1^{\,n}-\theta_2^{\,n})(\partial\theta_1-\partial\theta_2) \tag{D.48}$$
$$\sin(\theta_1-\theta_w) = \sin(\theta_1^{\,n}-\theta_w^{\,n}) + \cos(\theta_1^{\,n}-\theta_w^{\,n})(\partial\theta_1-\partial\theta_w) \tag{D.49}$$
$$\sin(\theta_2-\theta_w) = \sin(\theta_2^{\,n}-\theta_w^{\,n}) + \cos(\theta_2^{\,n}-\theta_w^{\,n})(\partial\theta_2-\partial\theta_w) \tag{D.50}$$

$$\dot{\theta}_1^2 = (\dot{\theta}_1^n)^2 + 2\dot{\theta}_1^n \partial\dot{\theta}_1 \qquad (D.51)$$

$$\dot{\theta}_2^2 = (\dot{\theta}_2^n)^2 + 2\dot{\theta}_2^n \partial\dot{\theta}_2 \qquad (D.52)$$

$$\dot{\theta}_w^2 = (\dot{\theta}_w^n)^2 + 2\dot{\theta}_w^n \partial\dot{\theta}_w \qquad (D.53)$$

Substitution of these terms into the non-linear equations (3.5), (3.6), and (3.7) results in linear equations of motion equations (3.11), (3.12), and (3.13).

APPENDIX E

COMPUTER PROGRAMS

E.1 NON-LINEAR SIMULATIONS

```
      REAL Y(13),C(24),W(13,14),QTQ(27)
      COMMON TR1,TR2,TR3,T1,T2,T3,VP1,VD1,VP2,VD2,VP3,VD3,TB
     1  XX,YY,ZZ,X1,X2,X3,X5,X6,X7,X9,X10,X11,TNT,TVT,TRD1,TRD
      EXTERNAL DERS
      INTEGER NTN
C*************************************************************
C                        DATA OUTPUT FILES
C-------------------------------------------------------------
      OPEN(UNIT=1,NAME='SNORC.PLT',STATUS='NEW')
      OPEN(UNIT=2,NAME='SNORC.PTT',STATUS='NEW')
      OPEN(UNIT=3,NAME='SNORC.COP',STATUS='NEW')
      OPEN(UNIT=4,NAME='SNORC.TOR',STATUS='NEW')
      OPEN(UNIT=5,NAME='SNORC.ERR',STATUS='NEW')
      OPEN(UNIT=6,NAME='SNORC.YER',STATUS='NEW')
      OPEN(UNIT=7,NAME='SNORC.END',STATUS='NEW')
      OPEN(UNIT=8,NAME='SNORC.YDD',STATUS='NEW')
      OPEN(UNIT=9,NAME='SNORC.TBD',STATUS='NEW')
      OPEN(UNIT=10,NAME='SNORC.ROT',STATUS='NEW')
      OPEN(UNIT=11,NAME='SNORC.KKK',STATUS='NEW')
C     OPEN(UNIT=12,NAME='SNORC.A1J',STATUS='NEW')
C     OPEN(UNIT=13,NAME='SNORC.A2J',STATUS='NEW')
C     OPEN(UNIT=14,NAME='SNORC.A3J',STATUS='NEW')
C     OPEN(UNIT=15,NAME='SNORC.CM1',STATUS='NEW')
C     OPEN(UNIT=16,NAME='SNORC.CM2',STATUS='NEW')
C     OPEN(UNIT=17,NAME='SNORC.CM3',STATUS='NEW')
      OPEN(UNIT=18,NAME='SNORC.3ER',STATUS='NEW')
      OPEN(UNIT=19,NAME='SNORC.TRA',STATUS='NEW')
      OPEN(UNIT=20,NAME='SNORC.2ND',STATUS='NEW')
C*************************************************************
C                        INITIALIZE VALUES
C          FINALT = SIMULATION TIME
C              DT = TIME INTERVAL
C             IND =
C               N =
C              NW =
C             TOL =
C               T = SUMATION OF TOTAL TIME
C             TVT =
C            TEND = TIME SENT TO TRAJECTORY SUBROUTINE
C-------------------------------------------------------------
      FINALT = 5.5
      DT = .001
      IND = 1
      N = 13
      NW = N
      TOL = .01
      T = 0.0
      TVT = 0.0
      JJJ = 50
      TEND = -DT
```

```
C-----------------------------------------------------------------
        Y(2) = .4514
        Y(5) = 1.3007
        Y(8) = 0.0
        Y(10) = 1.57079
        Y(11) = 0.0
        Y(13) = 0.0
C*****************************************************************

C                              CONTROLLER GAIN MATRIX
C-----------------------------------------------------------------
C       TYPE *,'ENTER KP1,KD1'
C       ACCEPT *,VP1,VD1
        VP1 = 60000.0
        VD1 = 6000.0
C       TYPE *,'ENTER KP2,KD2'
C       ACCEPT *,VP2,VD2
        VP2 = 20000.0
        VD2 = 2000.0
C       TYPE *,'ENTER KP3,KD3'
C       ACCEPT *,VP3,VD3
        VP3 = 2000.0
        VD3 = 100.0
C-----------------------------------------------------------------
C*****************************************************************
C                           DISTURBANCE FLAG
C-----------------------------------------------------------------
        TYPE *,'ENTER A=1 FOR YDD, B=1 FOR TBDD'
        ACCEPT *,A,B
        IF(A .EQ. 1.0)Y(13) = 6.24467
        IF(B .EQ. 1.0)Y(11) = 1.260869
C-----------------------------------------------------------------
C*****************************************************************
C                           GAIN MATRIX SENT TO FILE
C-----------------------------------------------------------------
        WRITE(11,330)VP1,VD1,ZZ,ZZ,ZZ,ZZ
        WRITE(11,331)ZZ,ZZ,VP2,VD2,ZZ,ZZ
        WRITE(11,332)ZZ,ZZ,ZZ,ZZ,VP3,VD3
330     FORMAT(3X,4HKK=[,6(F10.2,1X),1H;)
331     FORMAT(6(F10.2,1X),1H;)
332     FORMAT(6(F10.2,1X)2H];)
C-----------------------------------------------------------------
        TR1 = .4514
        TR2 = 1.3007
        TR3 = 0.0
        TEND = TEND + DT
        GO TO 141
C*****************************************************************
C                        TRAJECTORY SUBROUTINE CALLED
C-----------------------------------------------------------------
140     CALL TRAJS(TEND,TR1,TR2,TR3,TRD1,TRD2,TNT,Y(10),Y(11),TVT
C       IF(TNT .LT. 0.4 .OR. TNT .GT. 1.2)THEN
           TRD1 = 0.0
           TRD2 = 0.0
```

```
            TRD3 = 0.0
C          ELSE
C             TRD3 = -Y(11)
C          ENDIF
C*************************************************************
C                      INTEGRATION ROUTINE
C-------------------------------------------------------------
141       CALL DVERK(N,DERS,T,Y,TEND,TOL,IND,C,NW,W,IER)
C*************************************************************
C                              SCREEN OUTPUT
C-------------------------------------------------------------
C         TYPE *,'J1= ',Y(2),'J2= ',Y(5),'J3= ',Y(8),'T= ',TEND
C*************************************************************
C                               DATA TO FILES
C-------------------------------------------------------------

          Y(10) = Y(10) - 1.5708
          WRITE(1,380)Y(10),Y(2),Y(5),Y(8),Y(12),TNT
          WRITE(2,380)Y(10),TR1,TR2,TR3,Y(12),TNT
380       FORMAT(6(F10.4,1X))
          Y(10) = Y(10) + 1.5708
          WRITE(4,385)TEND,T1,T2,T3,T,T,T
          WRITE(3,385)TEND,TR1,TR2,TR3,Y(2),Y(5),Y(8)
          WRITE(19,385)T,TRX,TRY,TR3,T,T,T
385       FORMAT(7(F10.4,1X))
          ER1=TR1-Y(2)
          ER2=TR2-Y(5)
          ER3=TR3-Y(8)
          WRITE(5,385)T,ER3,ER2,ER1,T,T,T
C*************************************************************
C                         EXPLANATION OF ERRORS

C     TX,TY ARE DESIRED TIP LOCATIONS DURING LOAD CYCLE
          TX=COS(TR1)*3.0+COS(TR2)*3.0+COS(TR3)*2.3
          TY=SIN(TR1)*3.0+SIN(TR2)*3.0+SIN(TR3)*2.3
C     AX,AY ARE ACTUAL TIP LOCATIONS DURING LOAD CYCLE
          AX=COS(Y(2))*3.0+COS(Y(5))*3.0+COS(Y(8))*2.3
          AY=SIN(Y(2))*3.0+SIN(Y(5))*3.0+SIN(Y(8))*2.3
C     TTX,TTY ARE DESIRED END EFFECTOR LOCATIONS DURING RETURN CYCL
          TTX = COS(TR1)*3.0+COS(TR2)*3.0
          TTY = SIN(TR1)*3.0+SIN(TR2)*3.0
C     AAX,AAY ARE ACTUAL END EFFECTOR LOCATIONS DURING RETURN CYCLE
          AAX = COS(Y(2))*3.0+COS(Y(5))*3.0
          AAY = SIN(Y(2))*3.0+SIN(Y(5))*3.0
C     VER3 IS THE VERTICAL ERROR
          VER3 = AY-TY
C     TNT = 0.5 INDICATES READY TO START TRYING INSERTION
C     TVT = 2.0 INDICATES ROUND IS UNDER FORCE CONTROL
C     TNT = 1.0 INDICATES ROUND HAS BEEN INSERTED
C     TNT = 1.5 INDICATES BEGINNING RETURN CYCLE
C     TVT = 1.0 INDICATES THE ROUND IS LINED UP ENOUGH TO START INS
          IF(ABS(VER3).LT..01.AND.ABS(TR3-Y(8)).LT..007.AND.TNT.EQ..
          IF(TVT .GE. 1.5)VER3=AAY-TTY
          IF(AX .GE. -.82 .AND. AX .LE. -.80 .AND. TNT .LT. 1.2)THEN
```

```
        TN = T
        TOLV = .15
        TOL3 = .2
        GO TO 216
  ELSE
        TN = 0.0
        TOLV = 0.0
        TOL3 = 0.0
  ENDIF
  IF(AX .GE. -1.15 .AND. AX .LE. -1.0 .AND. TNT .LT. 1.2)THEN
        TN = T
        TOLV = .12
        TOL3 = .1745
        GO TO 216
  ELSE
        TN = 0.0
        TOLV = 0.0
        TOL3 = 0.0
  ENDIF
  IF(AX .GE. -1.53 .AND. AX .LE. -1.47 .AND. TNT .LT. 1.2)THEN
        TN = T
        TOLV = .08
        TOL3 = .1745

              GO TO 216
        ELSE
              TN = 0.0
              TOLV = 0.0
              TOL3 = 0.0
        ENDIF
        IF(AX .GE. -3.3 .AND. AX .LE. -1.75 .AND. TNT .LT. 1.2)THEN
              TN = T
              TOLV = 0.05
              TOL3 = 0.0
              TVT = 2.0
              GO TO 216
        ELSE
              TN = 0.0
              TOLV = 0.0
              TOL3 = 0.0
        ENDIF
216     WRITE(6,388)T,VER3,TN,TOLV
        WRITE(18,388)T,ER3,TN,TOL3
        IF(TVT .LT. 1.5)GO TO 217
        TX=TTX
        TY=TTY
        AY=AAY
        AX=AAX
217     WRITE(7,388)TX,TY,AX,AY
        WRITE(20,388)TTX,TTY,AAX,AAY
        WRITE(8,385)T,YDD,Y(13),Y(12),T,T,T
        WRITE(9,385)T,TBDD,Y(11),Y(10),T,T,T
388     FORMAT(4(F10.4,1X))
C       WRITE(15,385)T,X1,X2,X3,T,T,T
```

```
C       WRITE(16,385)T,X5,X6,X7,T,T,T
C       WRITE(17,385)T,X9,X10,X11,T,T,T
        JJJ=JJJ+1
        IF(JJJ .LT. 15)GO TO 590
        JJJ = 0.0
        NTN=NTN+1
C       WRITE(12,388)Y(2),Y(3),XX,T
C       WRITE(13,388)Y(5),Y(6),YY,T
C       WRITE(14,388)Y(8),Y(9),ZZ,T
        CALL LINECS(Y(2),Y(3),XX,Y(5),Y(6),YY,Y(8),Y(9),ZZ,NTN,QTQ)
        ZZ = 0.0
        OO = 1.0
C
        IF(NTN .GE. 10)GO TO 500
        IF(NTN .GE. 99)GO TO 590
        WRITE(10,400)NTN,ZZ,OO,ZZ,ZZ,ZZ,ZZ
400     FORMAT(3X,2HAA,I1,2H=[,6(F12.6,1X),1H;)
        GO TO 505
500     WRITE(10,501)NTN,ZZ,OO,ZZ,ZZ,ZZ,ZZ
501     FORMAT(3X,2HAA,I2,2H=[,6(F12.6,1X),1H;)
505     WRITE(10,401)QTQ(1),QTQ(2),QTQ(3),QTQ(4),QTQ(5),QTQ(6)
401     FORMAT(6(F12.6,1X),1H;)
        WRITE(10,401)ZZ,ZZ,ZZ,OO,ZZ,ZZ
        WRITE(10,401)QTQ(7),QTQ(8),QTQ(9),QTQ(10),QTQ(11),QTQ(12)
        WRITE(10,401)ZZ,ZZ,ZZ,ZZ,ZZ,OO
        WRITE(10,402)QTQ(13),QTQ(14),QTQ(15),QTQ(16),QTQ(17),QTQ(18)
402     FORMAT(6(F12.6,1X),2H];)
        IF(NTN .GE. 10)GO TO 550
        WRITE(10,403)NTN,ZZ,ZZ,ZZ
403     FORMAT(3X,2HBB,I1,2H=[,3(F12.6,1X),1H;)
        GO TO 555

550     WRITE(10,551)NTN,ZZ,ZZ,ZZ
551     FORMAT(3X,2HBB,I2,2H=[,3(F12.6,1X),1H;)
555     WRITE(10,404)QTQ(19),QTQ(20),QTQ(21)
        WRITE(10,404)ZZ,ZZ,ZZ
        WRITE(10,404)QTQ(22),QTQ(23),QTQ(24)
        WRITE(10,404)ZZ,ZZ,ZZ
404     FORMAT(3(F12.6,1X),1H;)
        WRITE(10,405)QTQ(25),QTQ(26),QTQ(27)
405     FORMAT(3(F12.6,1X),2H];)
C
590     IF(TEND .LT. FINALT)GO TO 140
        CLOSE(UNIT=1)
        CLOSE(UNIT=2)
        CLOSE(UNIT=3)
        CLOSE(UNIT=4)
        CLOSE(UNIT=5)
        CLOSE(UNIT=6)
        CLOSE(UNIT=7)
        CLOSE(UNIT=8)
        CLOSE(UNIT=9)
        CLOSE(UNIT=10)
        CLOSE(UNIT=11)
```

```
C        CLOSE(UNIT=12)
C        CLOSE(UNIT=13)
C        CLOSE(UNIT=14)
C        CLOSE(UNIT=15)
C        CLOSE(UNIT=16)
C        CLOSE(UNIT=17)
         CLOSE(UNIT=18)
         CLOSE(UNIT=19)
         CLOSE(UNIT=20)
         STOP
         END
C
C
C
         SUBROUTINE DERS(N,T,Y,YPRIME)
         REAL Y(13),YPRIME(13)
         COMMON AK1,AK2,AK3,T1,T2,T3,VP1,VD1,VP2,VD2,VP3,VD3,TBDD,
     1   XX,YY,ZZ,X1,X2,X3,X5,X6,X7,X9,X10,X11,TNT,TVT,TD1,TD2,TD3
C
         AM1=9.32
         AM2=6.77
         AM3=2.33
         IF(TVT .GE. 1.5)AM3 = .78
C        60 LB PAYLOAD AM3=2.64
C        50 LB PAYLOAD AM3=2.33
C        40 LB PAYLOAD AM3=2.02
C        NO    PAYLOAD AM3=.78
         AL1=3.0
         AL2=3.0
         ALB=1.0
         AL1C=1.57
         AL2C=1.49
         AL3C=.723
         IF(TVT .GE. 1.5)AL3C = .17
C        WITH 60 LB PAYLOAD AL3C=.75
C        WITH 50 LB PAYLOAD AL3C=.723
C        WITH 40 LB PAYLOAD AL3C=.679
C        WITHOUT    PAYLOAD AL3C=.17
         AJ1=7.25

         AJ2=5.22
         AJ3=1.47
         IF(TVT .GE. 1.5)AJ3 = .031
C        WITH PAYLOAD    AJ3=1.468
C        WITHOUT PAYLOAD AJ3=.031
C
         G=32.174
C
         VI1=0.0
C
         VI2=0.0
C
         VI3=0.0
C
```

```
      YDD1 = 30.0*SIN(5.0*T+3.14159)
      YDD2 = 8.0*SIN(45.0*T+3.14159)
      YDD3 = 5.0*SIN(66.0*T+3.14159)
      YDD = AC*(YDD1+YDD2+YDD3)
C
      TBDD1 = 6.0*SIN(16.0*T+3.14159)
      TBDD2 = 3.5*SIN(43.0*T+3.14159)
      TBDD3 = 2.5*SIN(3.0*T+3.14159)
      TBDD = B*(TBDD1+TBDD2+TBDD3)
C
C     COMPENSATOR
C
      CRR1 = (AM1*AL1C+AM2*AL1+AM3*AL1)*COS(Y(2))*G
      CRR2 = (AM2*AL2C+AM3*AL2)*COS(Y(5))*G
      CRR3 = AM3*AL3C*COS(Y(8))*G
C
      CFFD1A = (AM1*AL1C+AM2*AL1+AM3*AL1)*COS(Y(2))*YDD
      CFFD1B = (AM1*ALB*AL1C+AM2*ALB*AL1+AM3*ALB*AL1)*COS(Y(10)-
      CFFD1C = -(AM1*ALB*AL1C+AM2*ALB*AL1+AM3*ALB*AL1)*SIN(Y(10)
      CFFD1 = CFFD1A+CFFD1B+CFFD1C*Y(11)*Y(11)
C
      X1 = CFFD1A
      X2 = CFFD1B
      X3 = CFFD1C*Y(11)*Y(11)
C
      CFFD2A = (AM2*AL2C+AM3*AL2)*COS(Y(5))*YDD
      CFFD2B = (AM2*ALB*AL2C+AM3*ALB*AL2)*COS(Y(10)-Y(5))*TBDD
      CFFD2C = -(AM2*ALB*AL2C+AM3*ALB*AL2)*SIN(Y(10)-Y(5))*Y(11)
      CFFD2 = CFFD2A+CFFD2B+CFFD2C
C
      X5 = CFFD2A
      X6 = CFFD2B
      X7 = CFFD2C
C
      CFFD3A = AM3*AL3C*COS(Y(8))*YDD
      CFFD3B = AM3*ALB*AL3C*COS(Y(10)-Y(8))*TBDD
      CFFD3C = -AM3*ALB*AL3C*SIN(Y(10)-Y(8))*Y(11)*Y(11)
      CFFD3 = CFFD3A+CFFD3B+CFFD3C
C
      X9 = CFFD3A
      X10 = CFFD3B
      X11 = CFFD3C
C
CC                EQUATIONS OF MOTION
C
      TC1 = CRR1+CRR2+CRR3+CFFD1+CFFD2+CFFD3+CFFJ1+CFFJ2+CFFJ3

      T1 = VP1*(AK1-Y(2))+VD1*(TD1-Y(3))+VI1*Y(1)+TC1
      TC2 = CRR2+CRR3+CFFD2+CFFD3+CFFJ2+CFFJ3
      T2 = VP2*(AK2-Y(5))+VD2*(TD2-Y(6))+VI2*Y(4)+TC2
      TC3 = CRR3+CFFD3+CFFJ3
      T3 = VP3*(AK3-Y(8))+VD3*(TD3-Y(9))+VI3*Y(7)+TC3
C
C     TORQUE SATURATION
```

```
C
      IF(T1 .GT. 5913.75)T1=5913.75
      IF(T2 .GT. 1808.33)T2=1808.33
      IF(T3 .GT. 570.00)T3=570.00
      IF(T1 .LT. -5913.75)T1=-5913.75
      IF(T2 .LT. -1808.33)T2=-1808.33
      IF(T3 .LT. -570.00)T3=-570.00
C
      A1 = AM1*AL1C*AL1C+AJ1+AM2*AL1*AL1+AM3*AL1*AL1
      A2 = (AM2*AL1*AL2C+AM3*AL1*AL2)*COS(Y(2)-Y(5))
      A3 = AM3*AL1*AL3C*COS(Y(2)-Y(8))
C
      B1 = (AM2*AL1*AL2C+AM3*AL1*AL2)*COS(Y(2)-Y(5))
      B2 = AM2*AL2C*AL2C+AJ2+AM3*AL2*AL2
      B3 = AM3*AL2*AL3C*COS(Y(5)-Y(8))
C
      C1 = AM3*AL1*AL3C*COS(Y(2)-Y(8))
      C2 = AM3*AL2*AL3C*COS(Y(5)-Y(8))
      C3 = AM3*AL3C*AL3C+AJ3
C
      D1 = 0.0
      D2 = (AM2*AL1*AL2C+AM3*AL1*AL2)*SIN(Y(2)-Y(5))
      D3 = AM3*AL1*AL3C*SIN(Y(2)-Y(8))
C
      E1 = -(AM2*AL1*AL2C+AM3*AL1*AL2)*SIN(Y(2)-Y(5))
      E2 = 0.0
      E3 = AM3*AL2*AL3C*SIN(Y(2)-Y(8))
C
      F1 = -AM3*AL1*AL3C*SIN(Y(2)-Y(8))
      F2 = -AM3*AL2*AL3C*SIN(Y(5)-Y(8))
      F3 = 0.0
C
      O1 = -(AM1*ALB*AL1C+AM2*ALB*AL1+AM3*ALB*AL1)*COS(Y(10)-Y(2))
      O2 = -(AM2*ALB*AL2C+AM3*ALB*AL2)*COS(Y(10)-Y(5))
      O3 = -AM3*ALB*AL3C*COS(Y(10)-Y(8))
C
      P1 = (AM1*ALB*AL1C+AM2*ALB*AL1+AM3*ALB*AL1)*SIN(Y(10)-Y(2))
      P2 = (AM2*ALB*AL2C+AM3*ALB*AL2)*SIN(Y(10)-Y(5))
      P3 = AM3*ALB*AL3C*SIN(Y(10)-Y(8))
C
      Q1 = -(AM1*AL1C+AM2*AL1+AM3*AL1)*COS(Y(2))
      Q2 = -(AM2*AL2C+AM3*AL2)*COS(Y(5))
      Q3 = -AM3*AL3C*COS(Y(8))
C
      R1 = 1.0
      R2 = 0.0
      R3 = 0.0
C
      S1 = -1.0
      S2 = 1.0
      S3 = 0.0
C
      U1 = 0.0
```

```
      U2 = -1.0
      U3 = 1.0
C
      DA1 = B2*C3-B3*C2
      DA2 = B1*C3-B3*C1
      DA3 = B1*C2-B2*C1
C
      DB1 = A2*C3-A3*C2
      DB2 = A1*C3-A3*C1
      DB3 = A1*C2-A2*C1
C
      DC1 = A2*B3-A3*B2
      DC2 = A1*B3-A3*B1
      DC3 = A1*B2-A2*B1
C
      AA = A1*B2*C3+A2*B3*C1+A3*B1*C2-A3*B2*C1-A2*B1*C3-A1
C
      DD1 = (DA1*D1-DB1*D2+DC1*D3)/AA
      DD2 = (-DA2*D1+DB2*D2-DC2*D3)/AA
      DD3 = (DA3*D1-DB3*D2+DC3*D3)/AA
C
      EE1 = (DA1*E1-DB1*E2+DC1*E3)/AA
      EE2 = (-DA2*E1+DB2*E2-DC2*E3)/AA
      EE3 = (DA3*E1-DB3*E2+DC3*E3)/AA
C
      FF1 = (DA1*F1-DB1*F2+DC1*F3)/AA
      FF2 = (-DA2*F1+DB2*F2-DC2*F3)/AA
      FF3 = (DA3*F1-DB3*F2+DC3*F3)/AA
C
      OO1 = (DA1*O1-DB1*O2+DC1*O3)/AA
      OO2 = (-DA2*O1+DB2*O2-DC2*O3)/AA
      OO3 = (DA3*O1-DB3*O2+DC3*O3)/AA
C
      PP1 = (DA1*P1-DB1*P2+DC1*P3)/AA
      PP2 = (-DA2*P1+DB2*P2-DC2*P3)/AA
      PP3 = (DA3*P1-DB3*P2+DC3*P3)/AA
C
      QQ1 = (DA1*Q1-DB1*Q2+DC1*Q3)/AA
      QQ2 = (-DA2*Q1+DB2*Q2-DC2*Q3)/AA
      QQ3 = (DA3*Q1-DB3*Q2+DC3*Q3)/AA
C
      RR1 = (DA1*R1-DB1*R2+DC1*R3)/AA
      RR2 = (-DA2*R1+DB2*R2-DC2*R3)/AA
      RR3 = (DA3*R1-DB3*R2+DC3*R3)/AA
C
      SS1 = (DA1*S1-DB1*S2+DC1*S3)/AA
      SS2 = (-DA2*S1+DB2*S2-DC2*S3)/AA
      SS3 = (DA3*S1-DB3*S2+DC3*S3)/AA
C
      UU1 = (DA1*U1-DB1*U2+DC1*U3)/AA
      UU2 = (-DA2*U1+DB2*U2-DC2*U3)/AA
      UU3 = (DA3*U1-DB3*U2+DC3*U3)/AA
C
```

```
      XX1 = DD1*Y(3)*Y(3)+EE1*Y(6)*Y(6)+FF1*Y(9)*Y(9)
      XX2 = OO1*TBDD+PP1*Y(11)*Y(11)+QQ1*(G+YDD)
      XX3 = RR1*T1+SS1*T2+UU1*T3
      XX = XX1+XX2+XX3
C
      YY1 = DD2*Y(3)*Y(3)+EE2*Y(6)*Y(6)+FF2*Y(9)*Y(9)
      YY2 = OO2*TBDD+PP2*Y(11)*Y(11)+QQ2*(G+YDD)
      YY3 = RR2*T1+SS2*T2+UU2*T3
      YY = YY1+YY2+YY3
C
      ZZ1 = DD3*Y(3)*Y(3)+EE3*Y(6)*Y(6)+FF3*Y(9)*Y(9)
      ZZ2 = OO3*TBDD+PP3*Y(11)*Y(11)+QQ3*(G+YDD)
      ZZ3 = RR3*T1+SS3*T2+UU3*T3
      ZZ = ZZ1+ZZ2+ZZ3
C
      YPRIME(1) = AK1-Y(2)
      YPRIME(2) = Y(3)
      YPRIME(3) = XX
      YPRIME(4) = AK2-Y(5)
      YPRIME(5) = Y(6)
      YPRIME(6) = YY
      YPRIME(7) = AK3-Y(8)
      YPRIME(8) = Y(9)
      YPRIME(9) = ZZ
      YPRIME(10) = Y(11)
      YPRIME(11) = TBDD
      YPRIME(12) = Y(13)
      YPRIME(13) = YDD
C
      RETURN
      END
```

E.2 LINEARIZATION

```
C*****************************************************************
C       THIS PROGRAM LINEARIZES MY EQUATIONS OF MOTION (3-DOF)
C*****************************************************************
C
        REAL X(15),Y(8),Y1(13),Y2(13),Y3(13),QTQ(27)
C       OPEN(UNIT=2,NAME='CMAT.ABM',STATUS='NEW')
C
        X(1) = X1
        X(2) = X2
        X(3) = X3
        X(4) = X4
        X(5) = X5
        X(6) = X6
        X(7) = X7
        X(8) = X8
        X(9) = X9
C
        AM1 = 9.32
        AM2 = 6.77
        AM3 = 2.64
        AL1 = 3.0
        AL2 = 3.0
        ALB = 1.0
        AL1C = 1.57
        AL2C = 1.49
        AL3C = .75
        AJ1 = 7.25
        AJ2 = 5.22
        AJ3 = 1.47
C
        C1 = COS(X(1)-X(4))
        S1 = SIN(X(1)-X(4))
        C2 = COS(X(1)-X(7))
        S2 = SIN(X(1)-X(7))
        C3 = COS(Y(7)-X(1))
        S3 = SIN(Y(7)-X(1))
        C4 = COS(X(1))
        S4 = SIN(X(1))
        C5 = COS(Y(7)-X(4))
        S5 = SIN(Y(7)-X(4))
        C6 = COS(X(4)-X(7))
        S6 = SIN(X(4)-X(7))
        C7 = COS(X(4))
        S7 = SIN(X(4))
        C8 = COS(Y(7)-X(7))
        S8 = SIN(Y(7)-X(7))
        C9 = COS(X(7))
        S9 = SIN(X(7))
```

```
      X(12) = 0.0
      X(15) = 0.0
      Y(8) = 0.0
C
      A1 = AM1*AL1C*AL1C+AJ1+AM2*AL1*AL1+AM3*AL1*AL1
      A2 = (AM2*AL1*AL2C+AM3*AL1*AL2)*C1
      A3 = AM3*AL1*AL3C*C2
C
      B1 = (AM2*AL1*AL2C+AM3*AL1*AL2)*C1
      B2 = AM2*AL2C*AL2C+AJ2+AM3*AL2*AL2
      B3 = AM3*AL2*AL3C*C6
C
      PC1 = AM3*AL1*AL3C*C2
      PC2 = AM3*AL2*AL3C*C6
      PC3 = AM3*AL3C*AL3C+AJ3


      AD11 = -(AM1*ALB*AL1C+AM2*ALB*AL1+AM3*ALB*AL1)*(S3*X(12)
      AD12 = (AM2*AL1*AL2C+AM3*AL1*AL2)*(S1*X(6)-C1*X(5)*X(5))
      AD13 = AM3*AL1*AL3C*(S2*X(9)-C2*X(8)*X(8))
C     AD14 = (AM1*AL1C+AM2*AL1+AM3*AL1)*S4*(G+X(15))
      AD1 = AD11+AD12+AD13+AD14
      AD2 = (AM2*AL1*AL2C+AM3*AL1*AL2)*(S1*X(3)+C1*X(2)*X(2))
      AD3 = AM3*AL1*AL3C*(S2*X(3)+C1*X(2)*X(2))
C
      D1 = 0.0
      D2 = (2.0*(AM2*AL1*AL2C+AM3*AL1*AL2))*S1*X(2)
      D3 = 2.0*AM3*AL1*AL3C*S2*X(2)
C
      AE1 = (AM2*AL1*AL2C+AM3*AL1*AL2)*(C1*X(5)*X(5)-S1*X(6))
      AE21 = -(AM2*ALB*AL2C+AM3*ALB*AL2)*(S5*X(12)+C5*Y(8)*Y(8))
      AE22 = -(AM2*AL1*AL2C+AM3*AL1*AL2)*(S1*X(3)+C1*X(2)*X(2))
      AE23 = AM3*AL2*AL3C*(S6*X(9)-C6*X(8)*X(8))
C     AE24 = (AM2*AL2C+AM3*AL2)*S7*(G+X(15))
      AE2 = AE21+AE22+AE23+AE24
      AE3 = AM3*AL2*AL3C*(S6*X(6)+C6*X(5)*X(5))
C
      E1 = -2.0*(AM2*AL1*AL2C+AM3*AL1*AL2)*S1*X(5)
      E2 = 0.0
      E3 = 2.0*AM3*AL2*AL3C*S6*X(5)
C
      AF1 = AM3*AL1*AL3C*(C2*X(8)*X(8)-S2*X(9))
      AF2 = AM3*AL2*AL3C*(C6*X(8)*X(8)-S6*X(9))
      AF31 = -AM3*ALB*AL3C*(S8*X(12)+C8*Y(8)*Y(8))
      AF32 = -AM3*AL1*AL3C*(S2*X(3)+C2*X(2)*X(2))
      AF33 = -AM3*AL2*AL3C*(S6*X(6)+C6*X(5)*X(5))
      AF34 = -AM3*AL3C*S9*(G+X(15))*.5
      AF3 = AF31+AF32+AF33+AF34
C
      F1 = -2.0*AM3*AL1*AL3C*S2*X(8)
      F2 = -2.0*AM3*AL2*AL3C*S6*X(8)
      F3 = 0.0
C
      O1 = -(AM1*ALB*AL1C+AM2*ALB*AL1+AM3*ALB*AL1)*C3
      O2 = -(AM2*ALB*AL2C+AM3*ALB*AL2)*C5
      O3 = -AM3*ALB*AL3C*C8
```

```
      P1 = (AM1*ALB*AL1C+AM2*ALB*AL1+AM3*ALB*AL1)*S3
      P2 = (AM2*ALB*AL2C+AM3*ALB*AL2C)*S5
      P3 = AM3*ALB*AL3C*S8
C
      Q1 = -(AM1*AL1C+AM2*AL1+AM3*AL1)*C4
      Q2 = -(AM2*AL2C+AM3*AL2)*C7
      Q3 = -AM3*AL3C*C9
C
      AN1 = -(AM1*AL1C+AM2*AL1+AM3*AL1)*C4*G
      AN2 = -(AM2*AL2C+AM3*AL2)*C7*G
      AN3 = -AM3*AL3C*C9*G
C
      AO1 = -(AM1*AL1C*AL1C+AM2*AL1*AL1+AM3*AL1*AL1+AJ1)*X(3)
      AO2 = -(AM2*AL1*AL2C+AM3*AL1*AL2)*C1*X(3)
      AO3 = -AM3*AL1*AL3C*C2*X(3)
C
      AP1 = -(AM2*AL1*AL2C+AM3*AL1*AL2)*C1*X(6)
      AP2 = -(AM2*AL2C*AL2C+AM3*AL2*AL2+AJ2)*X(6)
      AP3 = -AM3*AL2*AL3C*C6*X(6)
C
      AQ1 = -AM3*AL1*AL3C*C2*X(9)
      AQ2 = -AM3*AL2*AL3C*C6*X(9)
      AQ3 = -(AM3*AL3C*AL3C+AJ3)*X(9)
C
      AR1 = 0.0
      AR2 = (AM2*AL1*AL2C+AM3*AL1*AL2)*S1*X(2)*X(2)
      AR3 = AM3*AL1*AL3C*S2*X(2)*X(2)
C
      AS1 = -(AM2*AL1*AL2C+AM3*AL1*AL2)*S1*X(5)*X(5)
      AS2 = 0.0
      AS3 = AM3*AL2*AL3C*S6*X(5)*X(5)
C
      AU1 = -AM3*AL1*AL3C*S2*X(8)*X(8)
      AU2 = -AM3*AL2*AL3C*S6*X(8)*X(8)
      AU3 = 0.0
C
      V1 = AN1+AO1+AP1+AQ1+AR1+AS1+AU1
      V2 = AN2+AO2+AP2+AQ2+AR2+AS2+AU2
      V3 = AN3+AO3+AP3+AQ3+AR3+AS3+AU3
C
      R1 = 1.0
      R2 = 0.0
      R3 = 0.0
C
      S1 = -1.0
      S2 = 1.0
      S3 = 0.0
C
      U1 = 0.0
      U2 = -1.0
      U3 = 1.0
C
      DA1 = B2*PC3-B3*PC2
      DA2 = B1*PC3-B3*PC1
      DA3 = B1*PC2-B2*PC1
```

```
      DB1 = A2*PC3-A3*PC2
      DB2 = A1*PC3-A3*PC1
      DB3 = A1*PC2-A2*PC1
C
      DC1 = A2*B3-A3*B2
      DC2 = A1*B3-A3*B1
      DC3 = A1*B2-A2*B1
C
      A = A1*B2*PC3+A2*B3*PC1+A3*B1*PC2-A3*B2*PC1-A2*B1
C
      DD1 = (DA1*D1-DB1*D2+DC1*D3)/A
      DD2 = (-DA2*D1+DB2*D2-DC2*D3)/A
      DD3 = (DA3*D1-DB3*D2+DC3*D3)/A
C
      ADD1 = (DA1*AD1-DB1*AD2+DC1*AD3)/A
      ADD2 = (-DA2*AD1+DB2*AD2-DC2*AD3)/A
      ADD3 = (DA3*AD1-DB3*AD2+DC3*AD3)/A
C
      EE1 = (DA1*E1-DB1*E2+DC1*E3)/A
      EE2 = (-DA2*E1+DB2*E2-DC2*E3)/A
      EE3 = (DA3*E1-DB3*E2+DC3*E3)/A
C
      AEE1 = (DA1*AE1-DB1*AE2+DC1*AE3)/A
      AEE2 = (-DA2*AE1+DB2*AE2-DC2*AE3)/A
      AEE3 = (DA3*AE1-DB3*AE2+DC3*AE3)/A
C
      FF1 = (DA1*F1-DB1*F2+DC1*F3)/A
      FF2 = (-DA2*F1+DB2*F2-DC2*F3)/A
      FF3 = (DA3*F1-DB3*F2+DC3*F3)/A
C
      AFF1 = (DA1*AF1-DB1*AF2+DC1*AF3)/A
      AFF2 = (-DA2*AF1+DB2*AF2-DC2*AF3)/A
      AFF3 = (DA3*AF1-DB3*AF2+DC3*AF3)/A
C
      OO1 = (DA1*O1-DB1*O2+DC1*O3)/A
      OO2 = (-DA2*O1+DB2*O2-DC2*O3)/A
      OO3 = (DA3*O1-DB3*O2+DC3*O3)/A
C
      PP1 = (DA1*P1-DB1*P2+DC1*P3)/A
      PP2 = (-DA2*P1+DB2*P2-DC2*P3)/A
      PP3 = (DA3*P1-DB3*P2+DC3*P3)/A
C
      QQ1 = (DA1*Q1-DB1*Q2+DC1*Q3)/A
      QQ2 = (-DA2*Q1+DB2*Q2-DC2*Q3)/A
      QQ3 = (DA3*Q1-DB3*Q2+DC3*Q3)/A
C
      VV1 = (DA1*V1-DB1*V2+DC1*V3)/A
      VV2 = (-DA2*V1+DB2*V2-DC2*V3)/A
      VV3 = (DA3*V1-DB3*V2+DC3*V3)/A
C
      RR1 = (DA1*R1-DB1*R2+DC1*R3)/A
      RR2 = (-DA2*R1+DB2*R2-DC2*R3)/A
      RR3 = (DA3*R1-DB3*R2+DC3*R3)/A
C
```

```
      SS1 = (DA1*S1-DB1*S2+DC1*S3)/A
      SS2 = (-DA2*S1+DB2*S2-DC2*S3)/A
      SS3 = (DA3*S1-DB3*S2+DC3*S3)/A


      UU1 = (DA1*U1-DB1*U2+DC1*U3)/A
      UU2 = (-DA2*U1+DB2*U2-DC2*U3)/A
      UU3 = (DA3*U1-DB3*U2+DC3*U3)/A
C
      ZZ = 0.0
      OO = 1.0
C
      QTQ(1)=ADD1
      QTQ(2)=DD1
      QTQ(3)=AEE1
      QTQ(4)=EE1
      QTQ(5)=AFF1
      QTQ(6)=FF1
      QTQ(7)=ADD2
      QTQ(8)=DD2
      QTQ(9)=AEE2
      QTQ(10)=EE2
      QTQ(11)=AFF2
      QTQ(12)=FF2
      QTQ(13)=ADD3
      QTQ(14)=DD3
      QTQ(15)=AEE3
      QTQ(16)=EE3
      QTQ(17)=AFF3
      QTQ(18)=FF3
      QTQ(19)=RR1
      QTQ(20)=SS1
      QTQ(21)=UU1
      QTQ(22)=RR2
      QTQ(23)=SS2
      QTQ(24)=UU2
      QTQ(25)=RR3
      QTQ(26)=SS3
      QTQ(27)=UU3
C     WRITE(2,100)N,ZZ,OO,ZZ,ZZ,ZZ,ZZ
C100  FORMAT(3X,3HAA(,I2,3H)=[,6(F12.6,1X),1H;)
C     WRITE(2,101)ADD1,DD1,AEE1,EE1,AFF1,FF1
C101  FORMAT(6(F12.6,1X),1H;)
C     WRITE(2,101)ZZ,ZZ,ZZ,OO,ZZ,ZZ
C     WRITE(2,101)ADD2,DD2,AEE2,EE2,AFF2,FF2
C     WRITE(2,101)ZZ,ZZ,ZZ,ZZ,ZZ,OO
C     WRITE(2,102)ADD3,DD3,AEE3,EE3,AFF3,FF3
C102  FORMAT(6(F12.6,1X),2H];)
```

```
C       WRITE(2,103)N,ZZ,ZZ,ZZ
C103    FORMAT(3X,3HBB(,I2,3H)=[,3(F12.6,1X),1H;)
C       WRITE(2,104)RR1,SS1,UU1
C       WRITE(2,104)ZZ,ZZ,ZZ
C       WRITE(2,104)RR2,SS2,UU2
C       WRITE(2,104)ZZ,ZZ,ZZ
C104    FORMAT(3(F12.6,1X),1H;)
C       WRITE(2,105)RR3,SS3,UU3
C105    FORMAT(3(F12.6,1X),2H];)
C       CLOSE(UNIT=2)
        RETURN
        END
```

E.3 TRAJECTORY

```
      SUBROUTINE TRAJS(T,T1,T2,T3,TD1,TD2,TNT,TBB,TDB,
C*************************************************************
C     THIS ROUTINE SOLVES THE INVERSE KINEMATICS FOR A
C     TRAJECTORY FOR THE TURRET MOUNTED (3 DOF) ROBOT
C*************************************************************
C
C     VARIABLES
C
C     OPEN(UNIT=2,NAME='TRAJ.1DT',STATUS='NEW')
C     OPEN(UNIT=3,NAME='TRAJ.2DT',STATUS='NEW')
C     OPEN(UNIT=24,NAME='TRAJ.3DT',STATUS='NEW')
C     OPEN(UNIT=25,NAME='TRAJ.XDT',STATUS='NEW')
C     OPEN(UNIT=26,NAME='TRAJ.YDT',STATUS='NEW')
C     OPEN(UNIT=7,NAME='TRAJ.PLT',STATUS='NEW')
C     OPEN(UNIT=8,NAME='TRAJ.GRF',STATUS='NEW')
      SDD = 5.0
      SDD1 = 8.0
C     SDD2 = 1.7
      SD = 5.0
      XO = 3.5
      YO = 4.2
      YO2 = 2.6
      XF = -1.0
      YF = 3.0
      DT = .001
      R = (YO-YF)/4.0
      RL = 4.5
      RY1 = YF+3.0*R
      RY2 = YF+R
      RX1 = 1.0
      RX2 = 1.5
      PI = 3.1415927
      AL1 = 3.0
      AL2 = 3.0
      TB = TBB-(PI*.5)
C     T = -DT
C
      IF(T .GT. 0.0)GO TO 10
      X = XO
      CC = 0.0
      TNT = 0.0
C
10    DO 113 J=1,10
      T = T+DT
C
      S1 = .5*SDD*T*T
      IF(X .LE. 1.0 .OR. CC .GE. 1.0)GO TO 20
```

```
      TA = T
C
      X = XO-S1
      XD = -SDD*T
      XDD = -SDD
C
      Y = YO
      YD = 0.0
      YDD = 0.0
C
      T3 = 0.0
      TD3 = 0.0
      TDD3 = 0.0
C
      X1 = X
      XD1 = XD
      XDD1 = XDD
C




      GO TO 100
C
C
20    IF(X .LE. .70015 .OR. CC .GE. 2.0)GO TO 25
      CC = 1.0
C
      TTS = T-TA
C
      X = X1-SQRT(.09-(.3-.3*TTS/.49)**2.0)
      XDD = -2.5*XDD1
      XD = XD1+XDD*TTS
C
      Y = RY1+SQRT(R*R-(X-RX1)**2.0)
      YD = -(2.5*TTS)/.4
      YDD = -XDD
C
      TDD3 = PI
      TD3 = PI*(T-TA)
      T3 = .5*PI*(T-TA)**2.0
C
      X2 = X
      XD2 = XD
      XDD2 = XDD
C
      GO TO 100
C
C
```

```
25      IF(DC .EQ. 2.0)GO TO 26
C
        DC = 2.0
C
        TATA = T
C
        X = .7
        XD = 0.0
        XDD = 0.0
C
        Y = 3.9
        YD = -2.5
        YDD = 0.0
C
        TDD3 = PI
        TD3 = PI*(T-TA)
        T3 = .5*PI*(T-TA)**2.0
C
        GO TO 100
C
26      IF(X .GE. .97 .OR. CC .GE. 3.0)GO TO 30
        CC = 2.0
C
        TTA = T-TATA
C
        X = 1.0-SQRT(.09-(.3-.3*(.4-TTA)/.4)**2.0)
        XDD = XDD2
        XD = XDD2*TTA
C
        Y = RY1-SQRT(R*R-(X-RX1)**2.0)
        YD = -2.5+2.5*TTA/.4
        YDD = XDD2
C
        TDD3 = PI
        TD3 = PI*(T-TA)
        T3 = .5*PI*(T-TA)**2.0
C
        XH = X
C

        GO TO 100
C
C
30      IF(EC .EQ. 2.0)GO TO 31
C
        EC = 2.0
C
        TBTB=T
C
        X = .985
        XD = 2.5
        XDD = 0.0
C
        Y = 3.6008
        YD = 0.0
        YDD = 0.0
```

```
          TDD3 = PI
          TD3 = (T-TA)*PI
          T3 = .5*PI*(T-TA)**2.0
C
          GO TO 100
C
31        IF(X .GE. 1.5 .OR. CC .GE. 4.0)GO TO 40
          CC = 3.0
C
          TBT = T-TBTB
          TC = T
C
          X = 1.0+.5*TBT/.4
          XD = 2.5
          XDD = 0.0
C
          Y = 3.6
          YD = 0.0
          YDD = 0.0
C
          IF(TBT .GE. .202)GO TO 35
C
          THT = T
C
          TDD3 = PI
          TD3 = PI*(T-TA)
          T3 = .5*PI*(T-TA)**2.0
C
          TA3 = T3
          TAD3 = TD3
C
          GO TO 36
C
35        TDD3 = -PI
          TD3 = TAD3+TDD3*(T-THT)
          T3 = TA3+(TAD3*(T-THT)-.5*PI*(T-THT)**2.0)
C
36        X3 = X
          XD3 = XD
          XDD3 = XDD
C
          GO TO 100
C
C
40        IF(X .GE. 1.798 .OR. CC .GE. 5.0)GO TO 45
          CC = 4.0
C
          TD = T
          TTC = T-TC-DT/2.0
```

```
C
      X = 1.511803+SQRT(.09-(.3-.3*TTC/.567)**2.0)
      XDD = -XDD2
      XD = 2.5+XDD*TTC
C
      Y = RY2+SQRT(R*R-(X-RX2+.0016)**2.0)
      YD = -2.5*TTC/.4
      YDD = XDD
C
      TDD3 = -PI
      TD3 = TAD3+TDD3*(T-THT)
      T3 = TA3+(TAD3*(T-THT)-.5*PI*(T-THT)**2.0)
C
      X4 = X
      XD4 = XD
      XDD4 = XDD
C
      GO TO 100
C
C
45    IF(X .LE. 1.53 .OR. CC .GE. 6.0)GO TO 50
      CC = 5.0
C
      TE = T
      TTE = T-TD-DT
C
      X = 1.5+SQRT(.09-(.3-.3*(.4-TTE)/.42)**2.0)
      XDD = XDD4
      XD = XDD*TTE
C
      Y = RY2+.022-SQRT(R*R-(X-RX2)**2.0)
      YD = -2.5+2.5*TTE/.4
      YDD = -XDD
C
      IF(T3 .GE. 3.1414)GO TO 47
C
      TDD3 = -PI
      TD3 = TAD3+TDD3*(T-THT)
      T3 = TA3+(TAD3*(T-THT)-.5*PI*(T-THT)**2.0)
      GO TO 48
C
47    TDD3 = 0.0
      TD3 = 0.0
      T3 = PI
C
48    X = X
C
      GO TO 100
C
C
50    IF(CGF .EQ. 3.0)GO TO 51
C
      CGF = 3.0
```

```
C
      X = 1.514
      XD = -2.5
      XDD = 0.0
C
      Y = 3.011
      YD = 0.1
      YDD = 0.0
C
      XE = X
      XDE = XD
C
      GO TO 100



C
51    IF(RL .LE. 2.1 .OR. CC .GE. 7.0)GO TO 60
C
      XE = 1.5
      YE = 3.0
      CC = 6.0
      TNT = 0.5
C
C     RL = 4.5
C
      X = RL*COS(-TB)+XE-4.5
      XD = RL*TDB*SIN(-TB)
C
      Y = RL*SIN(-TB)+YE
      YD = -RL*TDB*COS(-TB)
C
      T3 = -TB+PI
C
      IF(TVT .GE. 1.0)GO TO 52
      TF=T
      GO TO 100
C
52    TTF = T-TF
C
      RL = 4.5-(5.0*TTF-.5*5.0*TTF**2.0)
      RDL = -5.0+5.0*TTF
C
      X = RL*COS(-TB)+XE-4.5
      XD = RL*TDB*SIN(-TB)+RDL*COS(-TB)
C
      Y = RL*SIN(-TB)+YF
      YD = -RL*TDB*COS(-TB)+RDL*SIN(-TB)
C
      T3 = PI-TB
C
      RRL = RL
      TT5 = T
      GO TO 100
```

```
60      IF(TPAUSE .GE. 0.01 .OR. CC .GE. 8.0)GO TO 70
        CC = 7.0
        TPAUSE = T-TT5
        T6 = T
C
        XDD = 0.0
        XD = RRL*TDB*SIN(-TB)
        X = XE+RRL*COS(-TB)-4.5
C
        YDD = 0.0
        YD = -RRL*TBD*COS(-TB)
        Y = YF+RRL*SIN(-TB)
C
        TDD3 = 0.0
        TD = 0.0
        T3 = PI-TB
C
        TNT = 1.0
C
        XX6 = X
        YY6 = Y
        TT3 = T3
C
        SDD2 = (YY6-2.6)*1.4075/(YF-2.6)
        TBDD3 = TT3*3.523
        AHALFX = (3.25-.75)/2.0
        AHALFY = ((YY6+2.6)/2.0)


C
        GO TO 100
C
C
70      IF(X .GE. -.75 .OR. CC .GE. 9.0)GO TO 80
        CC = 8.0
        TT6 = T-T6
C
        XDD = SDD1
        XD = XDD*TT6
        X = XX6+.5*XDD*TT6*TT6
C
        YDD = 0.0
        YD = 0.0
        Y = YY6
C
        T3DD = 0.0
        T3D = 0.0
        T3 = TT3
C
        T7 = T
C
        TNT = 1.5
C
        GO TO 100
```

```
80      IF(X .GE. 1.4 .OR. Y .LE. AHALFY .OR. CC .GE. 10.0)
        CC = 9.0
C
        TT7 = T-T7
        TT6 = T-T6
C
        XDD = SDD1
        XD = XDD*TT6
        X = XX6+.5*XDD*TT6*TT6
C
        YDD = -SDD2
        YD = YDD*TT7
        Y = YY6+.5*YDD*TT7*TT7
C
        TDD3 = -TBDD3
        TD3 = TDD3*TT7
        T3 = TT3+.5*TDD3*TT7*TT7
C
        T8=T
C
        XD8 = XD
        X8 = X
        YD8 = YD
        Y8 = Y
        TAD8 = TD3
        TA8 = T3
C
        TNT = 1.5
C
        GO TO 100
C
C
90      IF(X .GE. 3.36 .OR. Y .LE. 2.6 .OR. CC .GE. 11.0)
        CC = 10.0
C
        TT8 = T-T8
C
        XDD = -SDD1
        XD = XD8+XDD*TT8

        X = X8+(XD8*TT8)+.5*XDD*TT8*TT8
C
        YDD = SDD2
        YD = YD8+YDD*TT8
        Y = Y8+(YD8*TT8)+.5*YDD*TT8*TT8
C
        TDD3 = TBDD3
        TD3 = TAD8+TDD3*TT8
        T3 = TA8+(TAD8*TT8)+.5*TDD3*TT8*TT8
C
        TNT = 1.5
C
        GO TO 100
```

```
95      IF(X .GE. XO .OR. CC .GE. 12.0)GO TO 113
        CC = 11.0
C
        TT8 = T-T8
C
        XDD = -SDD1
        XD = XD8+XDD*TT8
        X = X8+(XD8*TT8)+.5*XDD*TT8*TT8
C
        YDD = 0.0
        YD = 0.0
        Y = YO2
C
        TDD3 = 0.0
        TD3 = 0.0
        T3 = 0.0
C
        TNT = 1.5
C
C
100     YY = Y**2.0
C       IF(T .GT. 5.0)TYPE *,X,Y,T,CC,T3
        XX = X**2.0
        ALL1 = AL1**2.0
        ALL2 = AL2**2.0
        AAL = 2.0*AL1*AL2
        XXX = (YY+XX-ALL1-ALL2)/AAL
        T12 = ACOS(XXX)
        T1 = ATAN(Y/X)-ATAN((AL2*SIN(T12))/(AL1+AL2*COS(T12)))
        T2 = T12+T1
C
        IF(X .LT. 0.0)GO TO 108
        GO TO 110
108     T1=T1+PI
        T2=T2+PI
        T12=T12+PI
C
110     DD = AL1*AL2*SIN(T12)
C
        TD1 = (AL2*COS(T2)/DD)*XD+(AL2*SIN(T2)/DD)*YD
        TTD12 = (-(AL1*COS(T1)+AL2*COS(T2))/DD)*XD
        TTTD12 = -((AL1*SIN(T1)+AL2*SIN(T2))/DD)*YD
        TD12 = TTD12+TTTD12
        TD2 = TD1+TD12
C       IF(T .GE. 3.0)TYPE *,T,T1,TD1,T2,TD2
C
        IF(X .LT. 0.0)GO TO 111
        GO TO 112
111     TD1 = -TD1
        TD2 = -TD2
C
```

```
112     TTDD1 = (AL2*COS(T2)/DD)*XDD+(AL2*SIN(T2)/DD)*YDD
        TTTDD1 = (AL1*AL2*COS(T12)/DD)*TD1*TD1+(AL2*AL2/DD)
        TDD1 = TTDD1+TTTDD1
        TTDD2 = -(AL1*COS(T1)/DD)*XDD-(AL1*SIN(T1)/DD)*YDD
        TTTDD2 = -(AL1*AL1/DD)*TD1*TD1-(AL1*AL2*COS(T12)/DD)
        TDD2 = TTDD2+TTTDD2
C
C       VT1 = VT1+T1
C       VTD1 = VTD1+TD1
C       VTDD1 = VTDD1+TDD1
C       VT2 = VT2+T2
C       VTD2 = VTD2+TD2
C       VTDD2 = VTDD2+TDD2
C       VT3 = VT3+T3
C       VTD3 = VTD3+TD3
C       VTDD3 = VTDD3+TDD3
C       D1 = T1*180.0/PI
C       D2 = T2*180.0/PI
C       D3 = T3*180.0/PI
C
113     CONTINUE
C       WRITE(2,115)T1,TD1,TDD1,T
C       WRITE(3,115)T2,TD2,TDD2,T
C       WRITE(24,115)T3,TD3,TDD3,T
C       WRITE(25,115)X,XD,XDD,T
C       WRITE(26,115)Y,YD,YDD,T
C       WRITE(7,117)T1,T2,T3
C       WRITE(8,118)T1,T2,T3,T
C115    FORMAT(' ',4(F8.4,2X))
C117    FORMAT(3(F10.4,1X))
C118    FORMAT(4(F10.4,1X))
C       GO TO 10
C120    CLOSE(UNIT=2)
C       CLOSE(UNIT=3)
C       CLOSE(UNIT=24)
C       CLOSE(UNIT=25)
C       CLOSE(UNIT=26)
C       CLOSE(UNIT=7)
C       CC = T/DT
C       CT1 = VT1/CC
C       CTD1 = VTD1/CC
C       CTDD1 = VTDD1/CC
C       CT2 = VT2/CC
C       CTD2 = VTD2/CC
C       CTDD2 = VTDD2/CC
C       CT3 = VT3/CC
C       CTD3 = VTD3/CC
C       CTDD3 = VTDD3/CC
C
C       TYPE *,' '
C       TYPE *,' '
C       TYPE *,CT1,CTD1,CTDD1
C       TYPE *,CT2,CTD2,CTDD2
C       TYPE *,CT3,CTD3,CTDD3
C
        RETURN
        END
```

E.4 MATRIXX MACROS

```
Z=[0 0 0 0 0 0 0 0 0];
II=EYE(6);
SS=-[S1*II-A B;Z;Z;Z];
[L,U]=LU(SS);
MAT=EYE(9);
MAT(1,1)=1.0/U(1,1);
MAT(2,2)=1.0/U(2,2);
MAT(3,3)=1.0/U(3,3);
MAT(4,4)=1.0/U(4,4);
MAT(5,5)=1.0/U(5,5);
MAT(6,6)=1.0/U(6,6);
UU=MAT*U;
UU(5,:)=-UU(5,6)*UU(6,:)+UU(5,:);
UU(4,:)=-UU(4,5)*UU(5,:)+UU(4,:);
UU(4,:)=-UU(4,6)*UU(6,:)+UU(4,:);
UU(3,:)=-UU(3,4)*UU(4,:)+UU(3,:);
UU(3,:)=-UU(3,5)*UU(5,:)+UU(3,:);
UU(3,:)=-UU(3,6)*UU(6,:)+UU(3,:);
UU(2,:)=-UU(2,3)*UU(3,:)+UU(2,:);
UU(2,:)=-UU(2,4)*UU(4,:)+UU(2,:);
UU(2,:)=-UU(2,5)*UU(5,:)+UU(2,:);
UU(2,:)=-UU(2,6)*UU(6,:)+UU(2,:);
UU(1,:)=-UU(1,2)*UU(2,:)+UU(1,:);
UU(1,:)=-UU(1,3)*UU(3,:)+UU(1,:);
UU(1,:)=-UU(1,4)*UU(4,:)+UU(1,:);
UU(1,:)=-UU(1,5)*UU(5,:)+UU(1,:);
UU(1,:)=-UU(1,6)*UU(6,:)+UU(1,:);
UU(7,7)=-1. ;
UU(8,8)=-1. ;
UU(9,9)=-1. ;
C1=UU(:,7);
C2=UU(:,8);
C3=UU(:,9);
V=-C1(1)*C2+C2(1)*C1;
SS=-[S2*II-A B;Z;Z;Z];
[L,U]=LU(SS);
MAT=EYE(9);
MAT(1,1)=1.0/U(1,1);
MAT(2,2)=1.0/U(2,2);
MAT(3,3)=1.0/U(3,3);
MAT(4,4)=1.0/U(4,4);
MAT(5,5)=1.0/U(5,5);
MAT(6,6)=1.0/U(6,6);
UU=MAT*U;
```

```
UU(5,:)=-UU(5,6)*UU(6,:)+UU(5,:);
UU(4,:)=-UU(4,5)*UU(5,:)+UU(4,:);
UU(4,:)=-UU(4,6)*UU(6,:)+UU(4,:);
UU(3,:)=-UU(3,4)*UU(4,:)+UU(3,:);
UU(3,:)=-UU(3,5)*UU(5,:)+UU(3,:);
UU(3,:)=-UU(3,6)*UU(6,:)+UU(3,:);
UU(2,:)=-UU(2,3)*UU(3,:)+UU(2,:);
UU(2,:)=-UU(2,4)*UU(4,:)+UU(2,:);
UU(2,:)=-UU(2,5)*UU(5,:)+UU(2,:);
UU(2,:)=-UU(2,6)*UU(6,:)+UU(2,:);
UU(1,:)=-UU(1,2)*UU(2,:)+UU(1,:);
UU(1,:)=-UU(1,3)*UU(3,:)+UU(1,:);
UU(1,:)=-UU(1,4)*UU(4,:)+UU(1,:);
UU(1,:)=-UU(1,5)*UU(5,:)+UU(1,:);
UU(1,:)=-UU(1,6)*UU(6,:)+UU(1,:);
UU(7,7)=-1. ;
UU(8,8)=-1. ;
UU(9,9)=-1. ;
C1=UU(:,7);
C2=UU(:,8);
C3=UU(:,9);
VV=-C3(1)*C2+C2(1)*C3;
```

```
I=EYE(6);
II=EYE(9);
SS=[A-S1*I -B;II];
YY=SS(:,2);
SS(:,2)=SS(:,9);
SS(:,9)=YY;
YY=SS(:,4);
SS(:,4)=SS(:,7);
SS(:,7)=YY;
YY=SS(:,6);
SS(:,6)=SS(:,5);
SS(:,5)=YY;
SS(:,1)=-SS(1,1)*SS(:,9)+SS(:,1);
SS(:,3)=-SS(3,3)*SS(:,7)+SS(:,3);
SS(:,6)=-SS(5,6)*SS(:,5)+SS(:,6);
SS(:,8)=SS(:,8)/SS(2,8);
SS(:,6)=-SS(2,6)*SS(:,8)+SS(:,6);
SS(:,4)=-SS(2,4)*SS(:,8)+SS(:,4);
SS(:,3)=-SS(2,3)*SS(:,8)+SS(:,3);
SS(:,2)=-SS(2,2)*SS(:,8)+SS(:,2);
SS(:,1)=-SS(2,1)*SS(:,8)+SS(:,1);
SS(:,6)=SS(:,6)/SS(4,6);
SS(:,4)=-SS(4,4)*SS(:,6)+SS(:,4);
SS(:,3)=-SS(4,3)*SS(:,6)+SS(:,3);
SS(:,2)=-SS(4,2)*SS(:,6)+SS(:,2);
SS(:,1)=-SS(4,1)*SS(:,6)+SS(:,1);
SS(:,4)=SS(:,4)/SS(6,4);
SS(:,3)=-SS(6,3)*SS(:,4)+SS(:,3);
SS(:,2)=-SS(6,2)*SS(:,4)+SS(:,2);
SS(:,1)=-SS(6,1)*SS(:,4)+SS(:,1);
Z=[0 0 0 0 0 0];
ZZZZ=[Z;Z;Z;Z;Z;Z;Z;Z;Z];
ZZI=[ZZZZ II];
C1=SS(:,1);
C1=ZZI*C1;
C2=SS(:,2);
C2=ZZI*C2;
C3=SS(:,3);
C3=ZZI*C3;
CC1=-C1(6)*C2+C2(6)*C1;
R1=REAL(CC1);
I1=IMAG(CC1);
R2=REAL(C2);
I2=IMAG(C2);
R3=REAL(C3);
I3=IMAG(C3);
V=R1;
VV=I1;
```

```
T=[V1 V2 V3 V4 V5 V6];
V=[T(1,:);T(2,:);T(3,:);T(4,:);T(5,:);T(6,:)];
Q=[T(7,:);T(8,:);T(9,:)];
KK=Q*INV(V)
ACL=A-B*KK;
E=EIG(ACL)
```

E.5 PUMA CONTROL

```
{[a+]}
program mbmain;
 %include 'mbtype.h';

{*************************************************************

  This is the main program to test a compensator for robots subj
  to base motion.  The various procedures are defined as follows

        INPUTMENU        - provides the different program options

        INPUTPARAMETERS - allows parameters to be varied

        MBCLEAR          - clears encoder counts

        START            - moves PUMA from nest

        RETURN           - returns PUMA to the nest

        TRAJECTORY       - provides command angles for sample tra

        HORIZONTAL       - provides command angles for horizontal

        DISPLAY          - provides program instructions

        SLOWMOVE         - controller used by 'START and RETURN'

        CONTROLLER       - controller used by 'TRAJECTORY and HOR

        INPUTANDCHANGE   - Jeff's menu program, defined externall

        DEVHAN           - Device handler procedures
*************************************************************

procedure display;

  external;

procedure info;

  external;
```

```
  procedure inputmenu;
    external;

  procedure inputparameters;
    external;

  procedure mbclear;
    external;

  procedure start;
    external;

  procedure return;
   external;

 procedure horizontal;
   external;

 procedure trajectory;
   external;

 {*************************************************************

   This is the main program section.

 begin {mbmain}
   display;
   mstop := false;
repeat
   inputmenu;
   if oinfo > 0.5 then
     begin
     info;
     end;
   if oclear > 0.5 then
     begin
     mbclear;
     end;
   if ostart > 0.5 then
     begin
     samplerate := 0.005;
     start;
     end;
```

```
    if ohorizontal > 0.5 then
      begin
      inputparameters;
      horizontal;
      mstop := false;
      end;
    if otrajectory > 0.5 then
      begin
      inputparameters;
      trajectory;
      end;
    if oreturn > 0.5 then
      begin
      samplerate := 0.005;
      return;
      end;
    if ostop > 0.5 then
      begin
      mstop := true;
      end;
until mstop;
 end. {mbmain}
```

```
{The sample trajectory is used by this program}
{$nomain}
{$nocheck}
{[a+]}
%include 'mbtype.h';
%include 'mbsrvo.h';


procedure controller(k: integer);
  external;


procedure xslowmoves(home5, home2, home3: real;
                     rate, firstdelay, lastdelay: integer);
  external;

procedure holdposition(home5, home2, home3: real);
  external;

procedure trajectory;
  external;


procedure trajectory;


  var
    k, size: integer;
    temp,initpos1,initpos2,initpos3: real;
    initpos: real3type;


  begin {trajectory}
    size := 575;
    reset(hh,'sample.tra', ,size);
    read(hh,temp,initpos2,initpos3,initpos1);
    initpos[2] := initpos2*RADTOCONT[2]+FACTOR[2];
    initpos[3] := (initpos3-initpos2)*RADTOCONT[3]+FACTOR[3];
    initpos[1] := -(initpos1-initpos3)*RADTOCONT[1]+FACTOR[1];
    k := 0;
    writeln('Put wrist into configuration, then hit panic for
   xslowmoves(homepos[1],homepos[2],homepos[3],20,0,0);
   holdposition(homepos[1],homepos[2],homepos[3]);
    writeln('Hit panic to start traj');
   xslowmoves(initpos[1],initpos[2],initpos[3],20,0,0);
   holdposition(initpos[1],initpos[2],initpos[3]);
  controller(k);
    writeln('Hit panic to return arm to homeposition');
   xslowmoves(14879.58,2594.87,25063.30,20,0,0);
   holdposition(14879.58,2594.87,25063.30);
    writeln('Hit panic to return to menu');
   xslowmoves(14879.58,2594.87,28000,20,0,0);
   xslowmoves(homepos[1],homepos[2],homepos[3],20,0,0);
   holdposition(homepos[1],homepos[2],homepos[3]);
    writeln(' ');
    end;
```

```
{$nomain}  {PD controller joints 2,3,5 with compensator}
{$nocheck}
{[a+]}
%include 'mbtype.h';
%include 'mbsrvo.h';


procedure controller(var j: integer);
  external;



procedure controller;

  const
    NMTOCON = real3type(63.59389, -9.052489, -14.55563);
    l2 = 0.2032;
    l3 = 0.2032;
    yfactor = -0.99108;
    integralsaturation = real3type(100.0, 100.0, 100.0); {maximum val
                                                          integrator}
    integralrange = real3type(100.0, 200.0, 200.0); {range where inte
                                                     on}

  var
    kkk, kk, k, jj, i: integer;
    reallydone, dummy, error: boolean;
    prate, comp, integral, oldposition, home, setpoint: real3type;
    cf2, cf3, cf4, tcomp, ydd, outputmotor: real;
    s, c: real4type;
    position, velocity, positionerror, tg, rphi, phi: real3type;
    dat1, dat2, dat3, temp, err1, err2, err3, accel, acc, vel: real;
    numpoints, size, snumerr: integer;
    ddat1,ddat2,ddat3: packed array [1..565] of real;



function slowtraj(homeposition: real;
                  rate: real;
                  var setpoint: real):boolean;


  function realinrange(value, top, bottom: real): boolean;


     begin {realinrange}
       if (value > top) or (value < bottom) then
         realinrange := false
       else
         realinrange := true;
     end; {realinrange}
```

```
 begin {slowtraj}
   slowtraj := FALSE;
   if realinrange(setpoint, homeposition + rate, homeposition
     begin {if}
      setpoint := homeposition;
      slowtraj := TRUE;
     end {if}
   else if setpoint < homeposition then
     setpoint := setpoint + rate
   else
     setpoint := setpoint - rate;
 end; {slowtraj}


begin {controller}
    stop := FALSE;
    size := 575;
    k := 0;
    jj := 3;
    j := 0;
    kk := 55;
    kkk := 1;
    snumerr := 0;
    comp[1] := 0.0;
    comp[2] := 0.0;
    comp[3] := 0.0;
   reset(hh,'sample.tra', ,size);
    cf2 := lc2*m2+l2*m3+l2*m4+l2*mp;
    cf3 := lc3*m3+l3*m4+l3*mp;
    cf4 := lc4*m4+lp*mp;
     for i := 1 to 565 do
       begin
         read(hh,temp,dat2,dat3,dat1);
         ddat2[i] := dat2*RADTOCONT[2]+FACTOR[2];
         ddat3[i] := (dat3-dat2)*RADTOCONT[3]+FACTOR[3];
         ddat1[i] := -(dat1-dat3)*RADTOCONT[1]+FACTOR[1];
       end;
  for i := 1 to NUMAXES do
    begin {for}
     setpoint[i] := xreadpumas(i);
     oldposition[i] := setpoint[i];
     integral[i] := 0.0;
    end; {for}
   xclockstarts(round(samplerate*10)); {start for samplerate ms.
   error := xwaitonclocks;
 repeat
     if kk > numiterate then
       begin
        home[1] := ddat1[kkk];
        home[2] := ddat2[kkk];
        home[3] := ddat3[kkk];
        prate[1] := abs((ddat1[kkk]-ddat1[kkk+1])/(numiterate+1));
        prate[2] := abs((ddat2[kkk]-ddat2[kkk+1])/(numiterate+1));
        prate[3] := abs((ddat3[kkk]-ddat3[kkk+1])/(numiterate+1));
        kkk := kkk + 1;
        kk := 1;
       end
```

```
        else
          kk := kk + 1;
         if kkk = 563 then stop := TRUE;
            tcomp := 0.0;
       if xbuttons = TRUE then
         begin
           receive(accel, vel, acc);
           tcomp := g*accel*yfactor;
       for i := 1 to NUMAXES do
          begin
            rphi[i] := (xreadpumas(i) - FACTOR[i]) * CONTORAD[i];
          end;
         phi[1] := -rphi[1] + rphi[2] + rphi[3];
         phi[3] := rphi[2] + rphi[3];
         phi[2] := rphi[2];
      for i := 1 to NUMAXES do
        begin
         s[i] := sin(phi[i]);
         c[i] := cos(phi[i]);
        end;
       comp[2] := (c[2]*cf2+c[3]*cf3+c[1]*cf4)*NMTOCON[2]*tcomp;
       comp[3] := (c[3]*cf3+c[1]*cf4)*NMTOCON[3]*tcomp;
       comp[1] := c[1]*cf4*NMTOCON[1]*tcomp;

  end;
  for i := 1 to NUMAXES do
    begin {for}
      reallydone := slowtraj(home[i],prate[i],setpoint[i]);
      position[i] := xreadpumas(i);
      positionerror[i] := position[i] - setpoint[i];
      velocity[i] := (position[i] - oldposition[i])/samplerate;
      outputmotor := gp[i]*positionerror[i]+gv[i]*velocity[i]+comp[i];
      if abs(positionerror[i]) < integralrange[i] then
        begin {if}
         integral[i] := integral[i] + positionerror[i];
         if integral[i] > integralsaturation[i] then
           integral[i] := integralsaturation[i];
         if integral[i] < - integralsaturation[i] then
           integral[i] := - integralsaturation[i];
         outputmotor := outputmotor + integral[i] * gi[i];
        end {if}
      else
        integral[i] := 0.0;
      xmotoroutputs(i, outputmotor, dummy);
      oldposition[i] := position[i];
   end; {for}
    error := xwaitonclocks;
    if error then
      begin
        error := FALSE;
        snumerr := snumerr + 1;
      end; {if}
until stop;
```

```
 j := numiterate*kkk;
 writeln('There were ',snumerr:4,' errors more than ',samplerate:3:1);
 writeln(' ');
 writeln('The controller made ',j:4 ,' cycles');
 writeln(' ');

end;
```