A SIX DEGREES-OF-FREEDOM, HYDRAULIC, ONE PERSON MOTION SIMULATOR

By

Peter A. W. Drexel

B. A. Sc., The University of British Columbia, 1990

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

DEPARTMENT OF ELECTRICAL ENGINEERING

We accept this thesis as conforming

to the required standard

....

....

....

....

THE UNIVERSITY OF BRITISH COLUMBIA

September 1992

© Peter A. W. Drexel, 1992

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering
The University of British Columbia
2075 Wesbrook Place
Vancouver, Canada
V6T 1W5

Date: October 5, 1992

Abstract

In order to simulate the motion of heavy equipment used in the forestry industry for the purpose of human factors and teleoperation research, we have designed a six degrees-of-freedom, hydraulic, one person motion simulator. In this thesis we derive the inverse kinematics and the jacobian of the platform. We also derive the forward kinematics of the platform using Newton's method and address workspace issues from the point of view of a "singularity boundary." In addition, we look at platform dynamics including the design of the hydraulic actuation system. Based on the results of dynamic simulations, we expect that this motion simulator will provide 9.8 m/s^2, $400°/s^2$ accelerations and 1 m/s, 30°/s speeds to a 250 kg payload with 1 m, 45° displacements from a nominal center.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank Tim Salcudean and Peter Lawrence for their ideas and encouragement during the course of my research. Thanks also to my family and friends for their support over the last two years.

Chapter 1

Introduction

In this first chapter we discuss the motivation for designing and building a six degrees-of-freedom, hydraulic, one person motion simulator which will be used to improve the human interface with heavy equipment in the forest industry. We present a literature review of several significant authors in which we extract points relevant to our project and finally, we give a general description of the desired simulator performance and its design.

1.1 Motivation

The motivation for *A Six Degrees-of-Freedom, Hydraulic, One Person Motion Simulator* comes from the desire to create a facility where we can perform human factors and teleoperation research for the forest industry. In addition to these two primary uses, the human factors and teleoperation research facility can be used to study virtual reality, motion simulation and parallel manipulators. In this thesis we concentrate on the design of the motion simulator which forms the main component in this facility.

Human factors research is used to improve the human interface with heavy equipment such as excavators, log-loaders and feller-bunchers. Currently, most heavy equipment used in the forest industry operates by using a joint-space control strategy. For example, an excavator has three actuators which control the position and orientation of the bucket: one for the boom, one for the stick and one for the bucket itself (See Figure 1.1). In order for the excavator to perform a particular task, the operator must determine which actuators to control and when to control them. For an experienced operator, this is a routine task. However, for new operators, determining which controls to move is difficult to learn and in some situations, making the

wrong choice can damage the machine or cause personal injury. In order to reduce these risks, the excavator can be modified by adding a computer and joystick so that the controls operate using a cartesian-space strategy (*i.e.* each control axis operates the appropriate actuators to move along the x, y, z axes). The cartesian-space control strategy is easier for new operators to learn because it is more intuitive: when the operator moves the joystick forward the bucket also moves forward. This control strategy can also be used on other more complicated machines such as feller-bunchers or log-loaders. However, the control designer requires each machine to prototype and test the controls which is impractical and costly. To overcome this problem, the human factors and teleoperation research facility will simulate each of the different machines using different hand controls and control strategies. By comparing operator performance under different control conditions we hope to develop better control systems to make forestry machines safer and easier to operate.

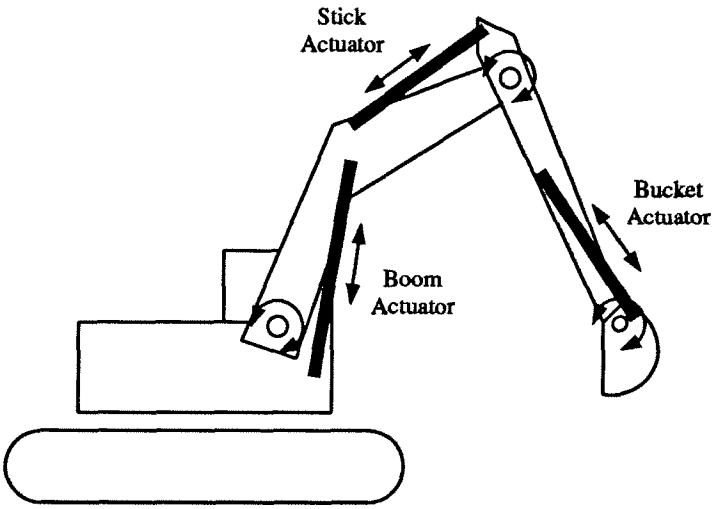


Figure 1.1: Excavator Actuators.

In addition to human factors research, the facility will be used for teleoperation research. This will provide an operator with sensory input such as motion, sight and sound cues needed to operate heavy equipment remotely from a safe location. In the forest industry workers are

constantly operating heavy equipment such as feller-bunchers, log-loaders and grapple-yarders in dangerous environments, for example, in British Columbia's mountainous terrain where these vehicles can easily tip over, possibly injuring the operator. By moving the operator off the machine into a safer environment these injuries will be avoided. However, one disadvantage is that the operator will lose all sense of sight, hearing and feel for what the machine is doing. To compensate for this, the human factors and teleoperation research facility will provide the operator with the appropriate motion, sight and sound cues in order for him/her to operate the machine as if he/she were still in it.

This facility is not simply restricted to human factors and teleoperation research. It may also be used in the study of virtual reality to create more realistic video games and movies in the entertainment industry. For motion simulation to simulate the motions of other vehicles such as boats and airplanes which will allow us to perform similar human factors studies. And finally for parallel manipulators to learn more about their dynamics, kinematics and singularities.

Figure 1.2 shows the proposed layout of this new human factors and teleoperation research facility. It is made up of three major components. The first is a six degrees-of-freedom, hydraulic motion simulator which will be programmed to respond to operator and environmental input and controlled by a Sparc 1E running the VxWorks Real-time operating system. The second component is a helmet-display driven by a Silicon Graphics Iris workstation which is worn by the operator and will provide a three dimensional view of a virtual workspace. As the motion simulator moves, workspace images will change, thus making the movement as realistic as possible for the operator. The third component is hand controls such as joysticks and force-reflecting controls. These controls will be mounted on the motion simulator which the operator will then use to command the virtual machine.

Motion simulators used as the basis for the human factors and teleoperation research facility are already available from several commercial manufacturers including Canadian Aerospace Electronics (CAE), Limited, Mannesmann-Rexroth/Hydraudyne, Limited [1], Limited and Kawasaki Heavy Industries, Limited [2]. Unfortunately, these simulators are too large and

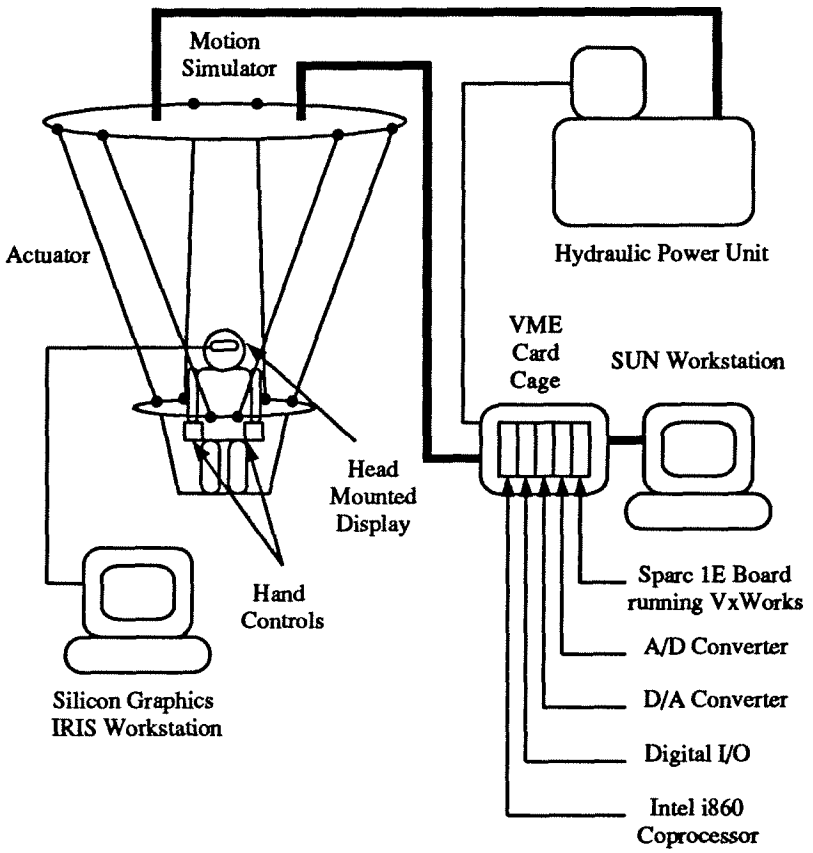


Figure 1.2: Human Factors/Teleoperation Research facility.

very expensive. They are typically designed for use as flight simulators supporting an entire airplane cockpit which usually includes several crew members and a large visual system. For this project, however, we only need a one person unit and because we are working with a limited budget, we are compelled to design and construct our own motion simulator.

1.2 Literature Review

Until recently there has not been a lot of literature published on Stewart platforms[1] or motion simulators (See Figure 1.3). Perhaps a reason for this delay in research lies in the fact that

[1] Although Stewart is usually credited with developing the Stewart platform [3], Gough *et al* devised a very similar device in 1947 for testing tires [4].

the forward kinematics and the dynamics of Stewart platform's are more complicated than those of serial type manipulators and only recently have we had the computing power to solve these problems numerically. The lack of published papers, however, has not prevented several companies, including Canadian Aerospace Electronics (CAE), Limited and Kawasaki Heavy Industries, Limited [2], from employing Stewart's design for flight simulators. Unfortunately their work is proprietary information. However, since 1980, several researchers including Hunt, Nguyen *et al*, Cleary *et al*, Fichter, Mohamed *et al*, Yang *et al*, Do *et al*, Nanua *et al*, Griffis *et al*, Geng *et al*, Innocenti *et al*, Merlet, Dieudonne *et al*, Koevermans *et al*, Koizumi *et al* and Duncan *et al* have examined Stewart's original ideas. The following is a review of their work.

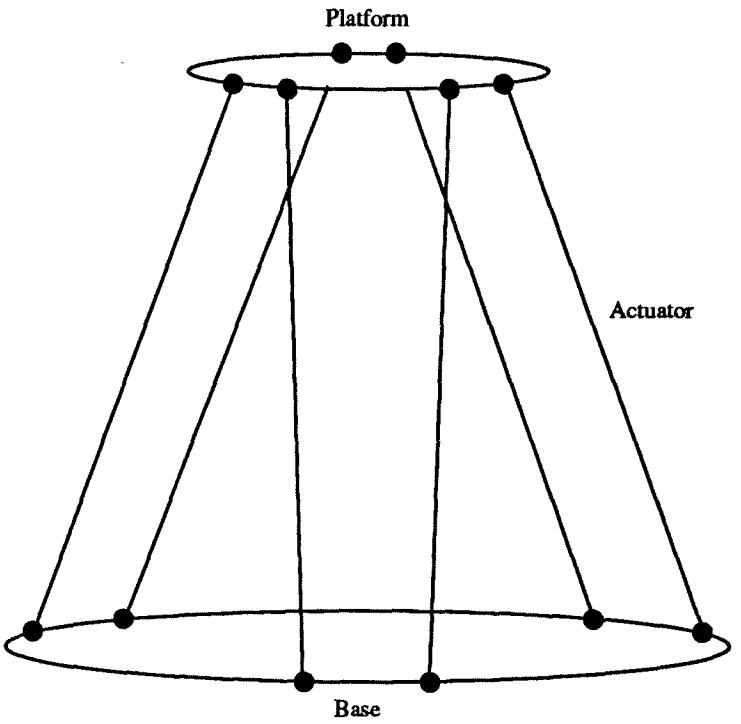


Figure 1.3: Stewart Platform.

Hunt gives a general survey of parallel mechanisms in [5]. He presents sample parallel mechanisms and tabulates all the possible combinations of connecting chains and their joints

for mechanisms with up to six degrees-of-freedom. He also considers the assembly modes of parallel devices, for a given set of actuator lengths there are several modes in which the actuators and end effector may be assembled. Hunt concludes, without proof, that there may be up to sixteen possible assembly modes for a Stewart platform.

The inverse kinematics (determining the actuator lengths given the platform position/orientation) of a Stewart platform is relatively easy to derive. These results are important because they form the basis for other derivations, workspace considerations and methods of control. This derivation is presented by Nguyen *et al*, Cleary *et al*, Fichter, Mohamed *et al*, Yang *et al*and Do *et al* in [6], [7], [8, 9], [10], [11] and [12, 13] respectively.

Yang *et al* use the inverse kinematics of the Stewart platform to plot its physical workspace. They assume that the actuators are attached to the base and platform via ball and socket joints. Using computer simulations, the platform is translated and rotated while monitoring the actuator lengths and joint angles. If the length of any one of the actuators becomes too short or too long, or the angle of any of the actuators exceeds the limit imposed by the joints, then the platform has reached a workspace limit. Using this exhaustive technique, the entire workspace of the platform is mapped out. Note however, that this algorithm does not consider platform singularities, therefore, the controllable workspace is smaller. Cleary *et al* and Fichter use similar techniques to plot cross-sections of the physical workspace of their prototype platforms in [7] and [9] respectively. Fichter determines that using gimbaled or Hooke joints instead of ball and socket joints will significantly increase the platform's physical workspace.

There is no known closed-form solution to the forward kinematics problem (determining the platform's position and orientation given the actuator lengths) for Stewart platforms. Nanua *et al* [14], Griffis *et al* [15], Geng *et al* [16], Innocenti *et al* [17] and Merlet [18] derive what we call the "polynomial method" for solving this forward kinematics problem. Their method reduces the problem to one of solving for the roots of a sixteenth order polynomial, thereby verifying Hunt's theory. The polynomial's variable is the tangent of the half angle from the base to the plane formed by two adjacent actuators. The order of the polynomial is reduced to eight if the

platform is restricted to be either above or below the base. The authors determine the correct root and position and orientation of the platform by back substitution and solution for the other angles. Unfortunately, the "polynomial method" only applies to platforms where the end points of the actuators intersect in either the base, platform or both as shown in Figure 1.4. Because of this restriction and significant computing time, this method is inappropriate for real-time control.

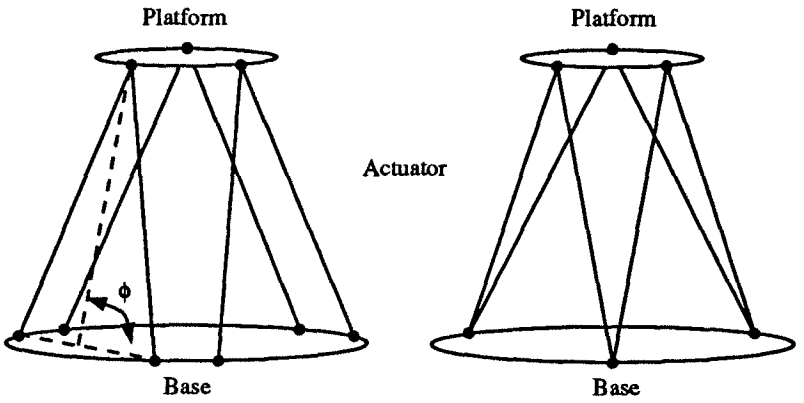


Figure 1.4: Special Geometries required for "Polynomial Method."

A second method of computing the forward kinematics, used by Nguyen *et al* in [6], is Powell's Direction Set Method. Powell's method minimizes the sum of the squares of the differences between the actual and computed actuator lengths. Similarly, Dieudonne *et al*, in [19], use Newton's method for multiple functions and variables to converge to a solution. This method makes use of the platform's jacobian matrix to update the estimates of the platform's position and orientation parameters. Unlike the "polynomial method," Powell and Newton's methods do not require special platform geometries. We will apply Newton's method to solve the forward kinematics problem in Chapter 2.

In [7], Cleary *et al* avoid the forward kinematics problem of the Stewart platform entirely by monitoring the position and orientation of their prototype platform with a passive serial linkage connected from the center of the base to the center of the platform. As the platform moves,

the six degrees-of-freedom serial manipulator follows. By reading the rotary position sensors on the serial manipulator and computing its forward kinematics, the position and orientation of the platform are determined.

Before controlling the position and orientation of a platform, we must have some idea about its singularities. At a singular position or orientation the platform gains one or more uncontrolled degrees-of-freedom. Unfortunately, the singularities of a parallel mechanism are not as evident as they are with most serial mechanisms. Cleary *et al* and Fichter look briefly at the singularities of the Stewart platform in [7] and [8, 9] respectively and find that when two or more of the actuators become linearly dependent, the platform is singular. Similarly, Cleary *et al* monitor the rank of the platform's jacobian matrix, when the rank is less than six, the platform is singular.

Fichter [9] and Do *et al* [12, 13] study the dynamics of parallel manipulators. Fichter addresses the platform's dynamics using screw theory and Plücker coordinates for each of the actuators. He assumes that the actuators have no mass, making the derivation of the actuator forces for a given platform acceleration relatively easy. Unlike Fichter, Do *et al* approach the inverse dynamics problem using Newton's laws of motion. In order to simplify their derivations, they assume that the actuators may be modeled as narrow rods, allowing them to ignore the torque about the actuator axes. Their algorithm consists of moving the platform and computing the angular velocity and acceleration of each actuator. Using these values and the algorithm, they compute the actuator reaction forces perpendicular to the actuator. By then applying these forces and the inertia forces to the platform, they determine the resulting forces parallel to the actuators. Do *et al*'s derivations and algorithm are used to determine the actuator performance requirements for the motion simulator in this thesis.

Nguyen *et al* provide an algorithm in [6] for computing straight line trajectories for a Stewart platform. In the algorithm they first compute the actuator lengths at the platform's start and finish positions. Second, they select the desired velocity and acceleration of the actuator with the largest absolute change in length. Third, they compute times for acceleration, deceleration

and constant velocity. And finally, they compute the velocity and acceleration for the other five actuators with the above times. Applying the algorithm results in the actuators having linearly dependent velocities. This algorithm may be of use because their experimental results indicate very small tracking errors.

Koevermans *et al* discuss the design and performance of a four degrees-of-freedom flight simulator in [20]. This simulator is capable of motion in the roll, pitch, yaw and heave directions. They use hydrostatic actuators to achieve a very low acceleration noise threshold of 0.1 m/s^2 which is imperceptible to humans. Their paper also provides a list of safety provisions which include position, velocity and acceleration limits on the actuators to protect the test pilot.

Koizumi *et al* discuss in [21] the performance and design of a six degrees-of-freedom, hydraulic motion base similar to the one discussed in this thesis which is now available from Kawasaki Heavy Industries Limited [2]. The design features hydrostatic cylinders for smooth operation and a large diameter platform (4.2 meters) to support wide loads. The step response and bandwidth for 5 centimeter platform displacements were found to be 0.19 seconds and 1 Hertz respectively which is the minimum we hope to achieve when running simulations with our platform.

In [22], Duncan *et al* describe a commercial human factors research laboratory for off-road vehicle design created at the Deere and Company Technical Center. Their system has several features including a six degrees-of-freedom, hydraulic motion simulator which can support a variety of tractor cabs, a noise generator for creating the sound cues which alert the operator to engine speed and load changes and finally, a large rear projection video screen to provide the operator with visual cues of a simulated work site. The researchers have five important goals designed to improve the performance of operators and the safety of heavy equipment. First, they wish to determine the effects of noise, vibration and task complexity on the performance of the operator. Second, they want to evaluate the performance of operators who are using advanced controls and displays. Third, they would like to improve the comfort of the operator by designing new suspension systems for the enclosure and the seat. Fourth, they wish to determine

the behaviour of operators to simulated emergencies and vehicle malfunctions. Finally, they want to compare the role of the operator with respect to different levels of vehicle automation.

1.3 Overall Design and Performance

In order to protect the operator as well as meet the demands of our simulations, we chose an inverted rather than the traditional design of most flight simulators. In this design we also include several safety features in the hydraulic and electrical systems to prevent injury during simulations. Table 1.1 summarizes the dimensions of the simulator. The figures in Table 1.2

Table 1.1: Simulator Dimensions.

Platform Radius, r_p	0.5 m
Platform Actuator Angle, γ_p	8°
Base radius, r_b	1.0 m
Base Actuator Angle, γ_b	51°
Nominal Height, h	2.5 m

reflect the performance of existing flight simulators [2, 23], human safety considerations [22, 24] and the ability to achieve the sensations of operating a real excavator. To achieve these

Table 1.2: Desired Simulator Performance.

	Displacement	Velocity	Acceleration
x-axis	±1 m	±1 m/s	±9.8 m/s^2
y-axis	±1 m	±1 m/s	±9.8 m/s^2
z-axis	±1 m	±1 m/s	±9.8 m/s^2
Yaw	±45°	±30°/s	±400°/s^2
Pitch	±45°	±30°/s	±400°/s^2
Roll	±45°	±30°/s	±400°/s^2

performance figures, the simulator uses six 1.5 inch bore, 54 inch stroke hydraulic actuators each controlled by a three-stage, proportional valve. The valves feature a near linear response curve and large bandwidth similar to that of more expensive servo valves making them easy

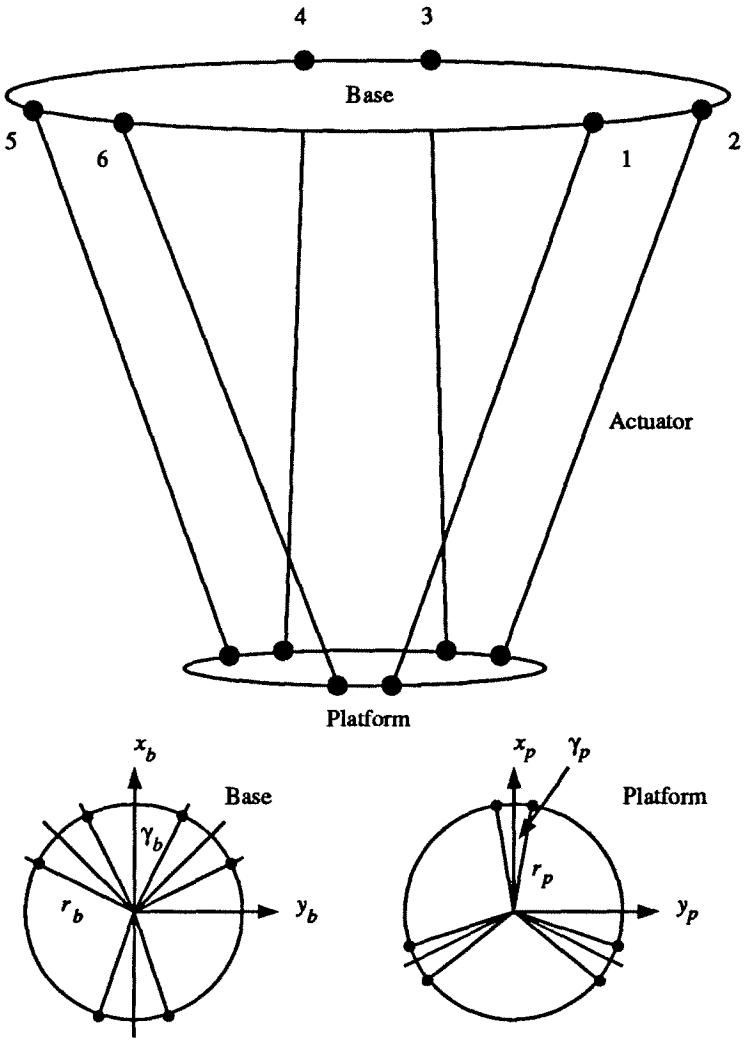


Figure 1.5: Inverted Simulator Geometry.

to model mathematically. The hydraulic fluid for the actuators is supplied by a 30 gallon per minute (GPM), 2500 pounds per square inch (PSI) power unit and two 10 gallon accumulators. With this design and expected performance, we anticipate a motion simulator which will form the basis of our human factors and teleoperation research facility.

The remainder of this thesis is divided into four chapters. In Chapter Two, we will derive the platform's kinematics and jacobian and look at singularity issues. We will explain relevant

issues concerning the design of the motion simulator, the actuators, the hydraulic, mechanical and electrical systems in Chapter Three. Chapter Four derives a simplified version of the motion simulator's dynamics as well as looking at issues of control. Finally, in the last chapter, we present some conclusions and suggestions for further work to help complete the human factors and teleoperation research facility.

Chapter 2

Kinematics

In Chapter Two we present the derivation of the platform's inverse kinematics which transform the platform's position and orientation to the lengths of the six actuators. We derive the platform's jacobian and use it to determine and avoid the platform's singularities so that we do not lose control of the platform during a simulation. We also use the jacobian matrix to determine expressions for the platform's velocity and acceleration kinematics. Then, using Newton's method for multiple equations and variables and the platform's jacobian, we derive the platform's forward kinematics which determine the platform's position and orientation given the lengths of the actuators.

2.1 Inverse Kinematics

In the case of a serial manipulator the inverse kinematics are usually quite difficult to derive and are generally non-unique. However, in this project, the platform's inverse kinematics are relatively easy to derive because the platform is a parallel-type manipulator. The platform's inverse kinematics map the position and orientation of the platform to the lengths of the platform's six actuators. Referring to Figure 2.6, we see that there is one coordinate frame fixed to the platform and one coordinate frame fixed to the base whose axes are labeled x_p, y_p, z_p and x, y, z respectively. We can write the equation for a vector, ${}^b\mathbf{a}_i$, representing actuator i in the base frame as

$$ {}^b\mathbf{a}_i = {}^b\mathbf{R}_p {}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i \tag{2.1} $$

where ${}^b\mathbf{R}_p$ (variable) is the rotation matrix whose columns are the coordinates of the platform basis vectors in the base frame, ${}^b\mathbf{d}_p = [x_p, y_p, z_p]^T$ (variable) is the position of the platform's

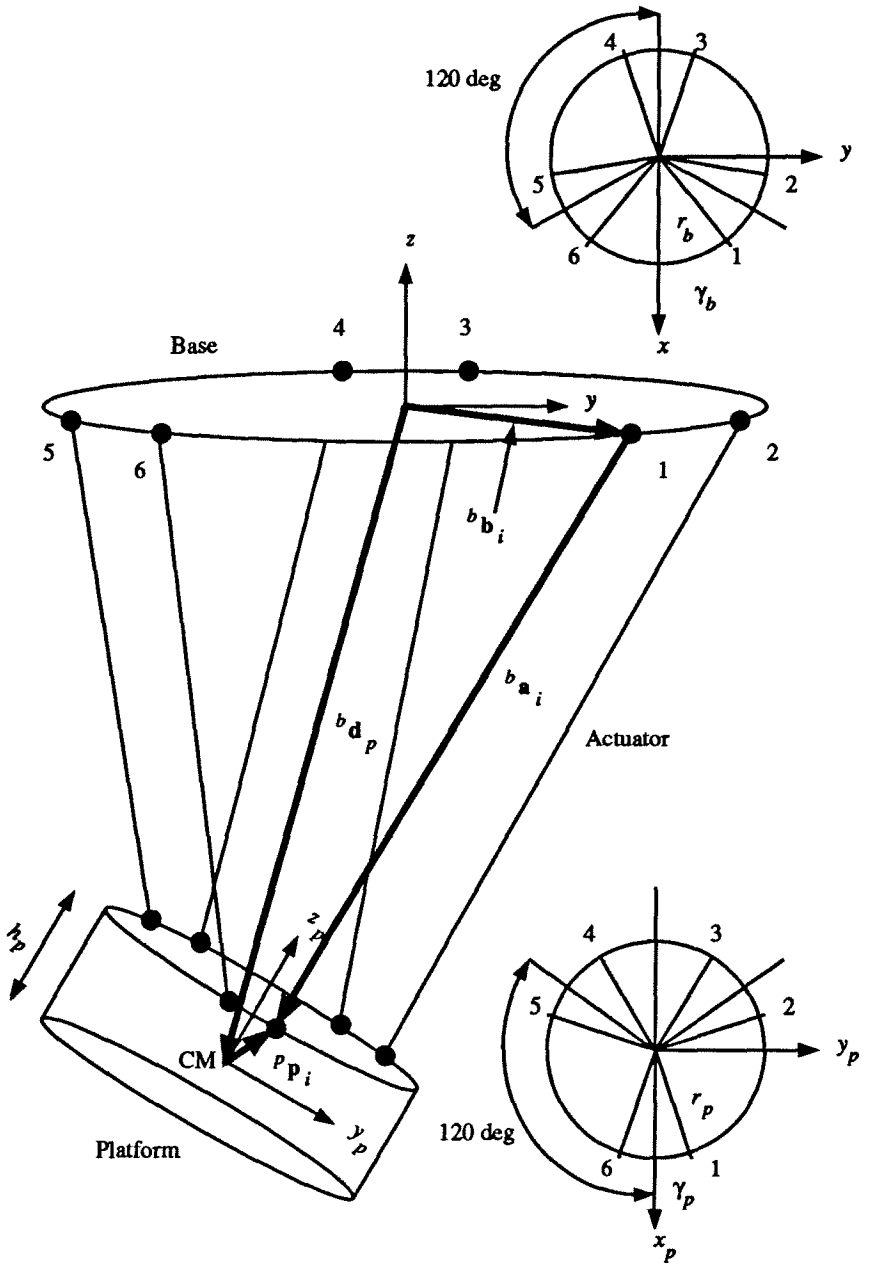


Figure 2.6: Vector Definitions for Inverse Kinematics.

center of mass with respect to the base frame, $^p\mathbf{p}_i$ (fixed) is the position of the platform end of actuator i with respect to the platform frame and $^b\mathbf{b}_i$ (fixed) is the position of the base end of

actuator i with respect to the base frame. If the actuator end points are arranged in pairs at 120 degree intervals around a circle, as shown in Figure 2.6, then we write vectors ${}^{p}\mathbf{p}_i$ and ${}^{b}\mathbf{b}_i$ as

$$ {}^{p}\mathbf{p}_i = \begin{bmatrix} r_p \cos\gamma_{pi} & r_p \sin\gamma_{pi} & \frac{h_p}{2} \end{bmatrix}^T \text{ and} \tag{2.2} $$

$$ {}^{b}\mathbf{b}_i = \begin{bmatrix} r_b \cos\gamma_{bi} & r_b \sin\gamma_{bi} & 0 \end{bmatrix}^T \tag{2.3} $$

where r_p is the radius of the platform circle, h_p is the nominal height of the platform, r_b is the radius of the base circle and γ_{pi} and γ_{bi} are defined for each actuator as shown in Table 2.3. Note that (2.2) assumes that the platform's center of mass is vertically centered and that the

Table 2.3: Platform and Base Actuator End Point Angles.

i	γ_{pi}	γ_{bi}
1	$\gamma_p = 8°$	$\gamma_b = 51°$
2	$120° - \gamma_p$	$120° - \gamma_b$
3	$120° + \gamma_p$	$120° + \gamma_b$
4	$-120° - \gamma_p$	$-120° - \gamma_b$
5	$-120° + \gamma_p$	$-120° + \gamma_b$
6	$-\gamma_p$	$-\gamma_b$

platform ends of the actuators attach to the top of the platform.

The platform's orientation matrix, ${}^{b}\mathbf{R}_p$, is defined using roll, pitch and yaw angles as

$$ \begin{aligned} {}^{b}\mathbf{R}_p(\psi_p, \theta_p, \phi_p) &= \begin{bmatrix} \mathrm{c}\phi_p & -\mathrm{s}\phi_p & 0 \\ \mathrm{s}\phi_p & \mathrm{c}\phi_p & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{c}\theta_p & 0 & \mathrm{s}\theta_p \\ 0 & 1 & 0 \\ -\mathrm{s}\theta_p & 0 & \mathrm{c}\theta_p \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}\psi_p & -\mathrm{s}\psi_p \\ 0 & \mathrm{s}\psi_p & \mathrm{c}\psi_p \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{c}\theta_p\mathrm{c}\phi_p & \mathrm{s}\psi_p\mathrm{s}\theta_p\mathrm{c}\phi_p - \mathrm{c}\psi_p\mathrm{s}\phi_p & \mathrm{c}\psi_p\mathrm{s}\theta_p\mathrm{c}\phi_p + \mathrm{s}\psi_p\mathrm{s}\phi_p \\ \mathrm{c}\theta_p\mathrm{s}\phi_p & \mathrm{s}\psi_p\mathrm{s}\theta_p\mathrm{s}\phi_p + \mathrm{c}\psi_p\mathrm{c}\phi_p & \mathrm{c}\psi_p\mathrm{s}\theta_p\mathrm{s}\phi_p - \mathrm{s}\psi_p\mathrm{c}\phi_p \\ -\mathrm{s}\theta_p & \mathrm{s}\psi_p\mathrm{c}\theta_p & \mathrm{c}\psi_p\mathrm{c}\theta_p \end{bmatrix} \end{aligned} \tag{2.4} $$

where ψ_p is the yaw angle, θ_p is the pitch angle, ϕ_p is the roll angle and $\mathrm{c}\psi_p = \cos\psi_p, \mathrm{s}\psi_p = \sin\psi_p$, etc. We chose roll, pitch and yaw angles rather than other possible angle combinations

because it is easier to relate them to the platform's actual physical orientation which is helpful when specifying trajectories.

The length of actuator i is easily obtained from (2.1) to give us,

$$l_i = \|{}^b\mathbf{a}_i\| = \sqrt{{}^b\mathbf{a}_i{}^T {}^b\mathbf{a}_i}. \tag{2.5}$$

For each actuator, $i = 1 \ldots 6$, (2.5) expresses its length given the platform's position and orientation and hence the inverse kinematics of the platform (See [25] for C source code).

2.2 Platform Jacobian

This section contains derivations of the platform's jacobian matrix and its derivatives as they are used to determine the platform's singularities, the actuator forces and the platform's position, velocity and acceleration. We use the first derivative of the jacobian matrix to determine the platform's acceleration and actuator forces and the second derivative to model the hydraulic actuator and platform dynamics.

For parallel-type manipulators, such as a Stewart platform, it is easier to define the jacobian matrix as a transformation from the platform's velocity to the actuator velocities. The platform's jacobian is defined as

$$\dot{\mathbf{l}} = \mathbf{J} \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix}, \tag{2.6}$$

where $\dot{\mathbf{l}}$ is the six vector of actuator velocities, $\mathbf{J}$ is the platform's six by six jacobian matrix, ${}^b\mathbf{v}_p = {}^b\dot{\mathbf{d}}_p$ (three vector) is the platform's translational velocity and ${}^b\omega_p$ (three vector) is the platform's angular velocity. Note that ${}^b\omega_p$ is written in terms of the derivatives of the platform's rotation angles as

$${}^b\omega_p = \mathbf{B} \begin{bmatrix} \dot{\psi}_p \\ \dot{\theta}_p \\ \dot{\phi}_p \end{bmatrix} \tag{2.7}$$

where for the Euler angle rotation ${}^b\mathbf{R}_p$ defined above

$$\mathbf{B} = \begin{bmatrix} c\theta_p c\phi_p & -s\phi_p & 0 \\ c\theta_p s\phi_p & c\phi_p & 0 \\ -s\theta_p & 0 & 1 \end{bmatrix}. \tag{2.8}$$

Because each row of the jacobian matrix corresponds to one of the platform's six actuators, and all the actuators are equivalent, we only need to derive row i. We can write an expression for the square of the magnitude (length) of actuator i, $l_i^2 = \|{}^b\mathbf{a}_i\|^2$ by multiplying the transpose of (2.1) with itself to obtain,

$$l_i^2 = {}^p\mathbf{p}_i{}^T{}^p\mathbf{p}_i + {}^b\mathbf{d}_p^T{}^b\mathbf{d}_p + {}^b\mathbf{b}_i{}^T{}^b\mathbf{b}_i - 2{}^b\mathbf{b}_i{}^T{}^b\mathbf{d}_p + 2\left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)^T\left({}^b\mathbf{d}_p - {}^b\mathbf{b}_i\right). \tag{2.9}$$

Taking the time derivative of (2.9) we obtain,

$$l_i\dot{l}_i = \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i\right)^T{}^b\mathbf{v}_p + \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)^T\mathbf{S}^T\left({}^b\omega_p\right)\left({}^b\mathbf{d}_p - {}^b\mathbf{b}_i\right). \tag{2.10}$$

where ${}^b\dot{\mathbf{R}}_p = \mathbf{S}\left({}^b\omega_p\right){}^b\mathbf{R}_p$ and $\mathbf{S}\left({}^b\omega_p\right)$ is a skew symmetric matrix, such that $\mathbf{S}(\mathbf{v})\mathbf{w} = \mathbf{v}\times\mathbf{w} =$ the cross product of $\mathbf{v}$ with $\mathbf{w}$. The elements of $\mathbf{S}(\mathbf{v})$ are

$$\mathbf{S}(\mathbf{v}) = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}. \tag{2.11}$$

Since

$$\mathbf{S}^T\left({}^b\omega_p\right)\left({}^b\mathbf{d}_p - {}^b\mathbf{b}_i\right) = \mathbf{S}\left({}^b\mathbf{d}_p - {}^b\mathbf{b}_i\right){}^b\omega_p \tag{2.12}$$

we rewrite (2.10) as

$$l_i\dot{l}_i = \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i\right)^T{}^b\mathbf{v}_p + \left(\left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right)\times\left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right)^T{}^b\omega_p. \tag{2.13}$$

or, in vector notation,

$$\dot{l}_i = \frac{1}{l_i}\begin{bmatrix} \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i\right)^T & \left(\left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right)\times\left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right)^T \end{bmatrix}\begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix}. \tag{2.14}$$

Comparing (2.6) to (2.14) we can write the i^{th} row of the jacobian matrix as (See [25] for C source code)

$$\mathbf{J}_i\left({}^b\mathbf{d}_p, {}^b\mathbf{R}_p\right) = \frac{1}{l_i}\left[\begin{array}{cc} \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i\right)^T & \left(\left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right)^T \end{array}\right]. \tag{2.15}$$

Equation (2.15) is written to show how the rows of the jacobian depend on the platform's position, ${}^b\mathbf{d}_p$, and orientation, ${}^b\mathbf{R}_p$.

As mentioned at the beginning of this section, the first and second derivatives of the jacobian matrix are useful for analyzing the platform's overall dynamics. The following are derivations of these derivatives.

We begin by taking the transpose of (2.15) to simplify further the mathematical operations

$$\mathbf{J}_i^T = \frac{1}{l_i}\left[\begin{array}{c} {}^b\mathbf{R}_p{}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i \\ \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) \end{array}\right]. \tag{2.16}$$

Using the quotient rule, we take the derivative of (2.16) with respect to time, giving

$$\begin{aligned} \dot{\mathbf{J}}_i^T = \frac{1}{l_i^2}\Bigg\{ l_i \left[\begin{array}{c} {}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\mathbf{v}_p \\ -{}^b\mathbf{v}_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) \end{array}\right] - \\ \left[\begin{array}{c} {}^b\mathbf{R}_p{}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i \\ \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) \end{array}\right] \dot{l}_i \Bigg\}. \end{aligned} \tag{2.17}$$

Equation (2.17) can be simplified using (2.16) to give

$$\dot{\mathbf{J}}_i^T = \frac{\left[\begin{array}{c} {}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\mathbf{v}_p \\ -{}^b\mathbf{v}_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) \end{array}\right] - \mathbf{J}^T\dot{l}_i}{l_i}. \tag{2.18}$$

Finally, taking the transpose of (2.18) we have an expression for each row of the first derivative of the platform's jacobian matrix (See [25] for C source code)

$$\dot{\mathbf{J}}_i\left({}^b\mathbf{d}_p, {}^b\mathbf{R}_p, {}^b\mathbf{v}_p, {}^b\omega_p, \dot{l}_i\right) = \frac{\left[\begin{array}{c} {}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\mathbf{v}_p \\ -{}^b\mathbf{v}_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) \end{array}\right]^T - \mathbf{J}_i\dot{l}_i}{l_i}. \tag{2.19}$$

To derive the second derivative of the jacobian matrix we start with (2.18) and use the quotient rule as before, giving

$$\ddot{\mathbf{J}}_i^T = \frac{1}{l_i^2}\left\{ l_i \left\{ \begin{bmatrix} {}^b\alpha_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right) + {}^b\omega_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right)\right) + {}^b\mathbf{a}_p \\ -{}^b\mathbf{a}_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right) - 2{}^b\mathbf{v}_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right)\right) + \\ \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\alpha_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right) + \\ {}^b\omega_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right)\right)\right) \end{bmatrix} - \dot{\mathbf{J}}_i^T \dot{l}_i - \mathbf{J}^T \ddot{l}_i \right\} \right.$$
$$\left. - \left\{ \begin{bmatrix} {}^b\omega_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right) + {}^b\mathbf{v}_p \\ -{}^b\mathbf{v}_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right) + \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right)\right) \end{bmatrix} - \mathbf{J}_i^T \dot{l}_i \right\} \dot{l}_i \right\} \quad (2.20)$$

where ${}^b\mathbf{a}_p$ (three vector) is the translational acceleration of the platform's center of mass, ${}^b\alpha_p$ (three vector) is the angular acceleration of the platform about its center of mass and $\ddot{l}_i$ is the acceleration of actuator i. Note that ${}^b\alpha_p$ is written in terms of the first and second derivatives of the platform's rotation angles as

$${}^b\alpha_p = \dot{\mathbf{B}} \begin{bmatrix} \dot{\psi}_p \\ \dot{\theta}_p \\ \dot{\phi}_p \end{bmatrix} + \mathbf{B} \begin{bmatrix} \ddot{\psi}_p \\ \ddot{\theta}_p \\ \ddot{\phi}_p \end{bmatrix} \quad (2.21)$$

where

$$\dot{\mathbf{B}} = \begin{bmatrix} -\dot{\theta}_p \mathrm{s}\theta_p \mathrm{c}\phi_p - \dot{\phi}_p \mathrm{c}\theta_p \mathrm{s}\phi_p & -\dot{\phi}_p \mathrm{c}\phi_p & 0 \\ -\dot{\theta}_p \mathrm{s}\theta_p \mathrm{s}\phi_p + \dot{\phi}_p \mathrm{c}\theta_p \mathrm{c}\phi_p & -\dot{\phi}_p \mathrm{s}\phi_p & 0 \\ -\dot{\theta}_p \mathrm{c}\theta_p & 0 & 0 \end{bmatrix}. \quad (2.22)$$

Equation (2.20) is simplified using (2.18) to give

$$\ddot{\mathbf{J}}_i^T = \frac{\begin{bmatrix} {}^b\alpha_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right) + {}^b\omega_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right)\right) + {}^b\mathbf{a}_p \\ -{}^b\mathbf{a}_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right) - 2{}^b\mathbf{v}_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right)\right) + \\ \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\alpha_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right) + {}^b\omega_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p {}^p\mathbf{p}_i\right)\right)\right) \end{bmatrix} - 2\dot{\mathbf{J}}_i^T \dot{l}_i - \mathbf{J}^T \ddot{l}_i}{l_i} \quad (2.23)$$

Finally, taking the transpose of (2.23) we have an expression for each row of the second derivative of the platform's jacobian matrix

$$\ddot{\mathbf{J}}_i \left({}^b\mathbf{d}_p, {}^b\mathbf{R}_p, {}^b\mathbf{v}_p, {}^b\omega_p, {}^b\mathbf{a}_p, {}^b\alpha_p, \dot{l}_i, \ddot{l}_i\right) =$$

$$\frac{\begin{bmatrix} {}^b\alpha_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\omega_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) + {}^b\mathbf{a}_p \\ -{}^b\mathbf{a}_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) - 2{}^b\mathbf{v}_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) + \\ \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\alpha_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\omega_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right)\right) \end{bmatrix}^T - 2\dot{\mathbf{J}}_i\dot{l}_i - \mathbf{J}_i\ddot{l}_i}{l_i} \tag{2.24}$$

In the sections that follow, the results derived above for the rows of the jacobian matrix and its first and second derivatives, namely (2.15), (2.19) and (2.24), are used to derive the following:

1. a method of determining when the platform is nearing a singular position or orientation so we do not lose control of the platform;

2. a transformation from the platform's velocity to the actuator velocities and the reverse transformation used to derive the combined dynamics of the platform and the hydraulics;

3. a transformation from the platform's acceleration to the actuator accelerations and the reverse transformation used to derive the combined dynamics of the platform and the hydraulics;

4. an iterative method used for estimating the platform's position and orientation given the actuator lengths and for controlling the position and orientation of the platform; and

5. a transformation from the actuator forces to the platform's force and torque and the reverse transformation used to derive the dynamics of the platform.

2.3 Singularity

When a parallel mechanism reaches a singular position or orientation it gains an uncontrolled degree of freedom and will rotate and/or translate without the lengths of the actuators changing. Therefore, in order to prevent this loss of control, we must avoid singularities. We use the platform's jacobian matrix to find the singularities because they are not geometrically obvious.

The singular value decomposition (SVD) [26] of the platform's jacobian matrix is:

$$\mathbf{J} = \mathbf{U}\Sigma\mathbf{V}^T \tag{2.25}$$

where $\mathbf{U}$ and $\mathbf{V}$ are row orthonormal matrices and Σ is a diagonal matrix of the singular values, $\sigma_1 \ldots \sigma_6$, of $\mathbf{J}$. By plotting the ratios of the largest to smallest singular value while varying two of the platform's six position and orientation parameters, we obtain a graphical description of the platform's singularities. For example, Figure 2.7 shows a plot of the singular value ratios versus varying yaw (ψ_p) and pitch (θ_p) for the platform at its nominal position (x = 0 m, y = 0 m, z = −2.5 m and ψ_p = 0°). We created the plot by computing the SVD of the platform's jacobian while varying the platform's yaw and pitch angles between 0° and 90° at 5° intervals (See Appendix C for MATLAB source code). This figure (2.7) shows that there

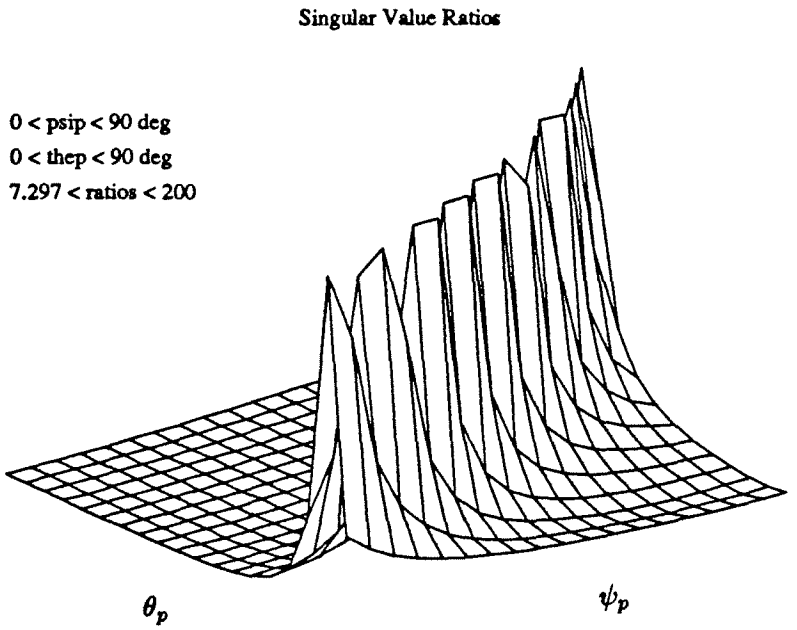


Figure 2.7: Singular Value Ratios Versus Pitch and Yaw

is a "singularity boundary" where the platform is nearing a singular position or orientation. To prevent losing control of the motion simulator we will monitor the ratios of the jacobian's singular values and stop the platform before it enters a singular position or orientation.

2.4 Velocity and Acceleration Kinematics

The jacobian matrix is used to solve for the actuator velocities given the platform's translational and angular velocities as shown in (2.26)

$$\dot{\mathbf{l}} = \mathbf{J} \begin{bmatrix} {}^{b}\mathbf{v}_p \\ {}^{b}\omega_p \end{bmatrix}. \tag{2.26}$$

If $\mathbf{J}$ is non-singular, then the inverse of the jacobian matrix can be used to solve for the platform's translational and angular velocities given the actuator velocities as shown in (2.27) (See [25] for C source code)

$$\begin{bmatrix} {}^{b}\mathbf{v}_p \\ {}^{b}\omega_p \end{bmatrix} = \mathbf{J}^{-1}\dot{\mathbf{l}}. \tag{2.27}$$

The jacobian matrix and its derivative can also be used to compute the actuator accelerations given the actuator velocities and the platform's velocity and acceleration. We see this by taking the derivative of (2.26) with respect to time to give

$$\ddot{\mathbf{l}} = \dot{\mathbf{J}} \begin{bmatrix} {}^{b}\mathbf{v}_p \\ {}^{b}\omega_p \end{bmatrix} + \mathbf{J} \begin{bmatrix} {}^{b}\mathbf{a}_p \\ {}^{b}\alpha_p \end{bmatrix}. \tag{2.28}$$

If the jacobian matrix is non-singular, (2.28) is rearranged allowing us to compute the platform's acceleration given its velocity and the actuator accelerations

$$\begin{bmatrix} {}^{b}\mathbf{a}_p \\ {}^{b}\alpha_p \end{bmatrix} = \mathbf{J}^{-1} \left(\ddot{\mathbf{l}} - \dot{\mathbf{J}} \begin{bmatrix} {}^{b}\mathbf{v}_p \\ {}^{b}\omega_p \end{bmatrix} \right). \tag{2.29}$$

(See [25] for C source code)

2.5 Forward Kinematics

To simulate and control the platform, we need to compute its forward kinematics. Although in practice, it is possible to measure the platform's position and orientation directly [7], it is mechanically easier to measure the lengths of the actuators. Unfortunately, as mentioned

earlier, for fully parallel six degrees-of-freedom manipulators, there is no known closed-form solution to the forward kinematics problem. Therefore, we propose to use Newton's method for multiple equations and variables [19, 26] to calculate the platform's forward kinematics iteratively. There are several advantages to this method. First, there is no geometry restriction which means that the actuators do not have to intersect. Second, the mathematics of this method is relatively uncomplicated compared to that of the "polynomial method." Finally, using a preplanned platform trajectory with small increments for each time step, we provide Newton's method with a "good" starting point. For this reason and the fact that Newton's method converges quadratically, this method will require only a few iterations per time step. For multiple equations and variables, Newton's method is

$$\mathbf{x}_{j+1} = \mathbf{x}_j - \left(\frac{\partial \mathbf{g}(\mathbf{x}_j)}{\partial \mathbf{x}_j}\right)^{-1} \mathbf{g}(\mathbf{x}_j) \tag{2.30}$$

where $\mathbf{x}$ is a vector of the variables we wish to estimate, $\mathbf{g}$ is a vector function which approaches zero as the estimate of $\mathbf{x}$ improves and j is the iteration count. For the motion simulator used in this project,

$$\mathbf{x}^T = \begin{bmatrix} {}^b\mathbf{d}_p^T & \psi_p & \theta_p & \phi_p \end{bmatrix}, \tag{2.31}$$

$$g_i(\mathbf{x}) = \| \left({}^b\mathbf{R}_p(\psi_p, \theta_p, \phi_p) {}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i \right) \| - l_i \text{ and} \tag{2.32}$$

$$\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{J}\left({}^b\mathbf{d}_p, {}^b\mathbf{R}_p(\psi_p, \theta_p, \phi_p) \right) \tag{2.33}$$

where g_i is the i^{th} row of $\mathbf{g}$ and l_i is the actual length of actuator i. Substituting into (2.30) gives the following iteration

$$\begin{bmatrix} {}^b\mathbf{d}_p \\ \psi_p \\ \theta_p \\ \phi_p \end{bmatrix}_{j+1} = \begin{bmatrix} {}^b\mathbf{d}_p \\ \psi_p \\ \theta_p \\ \phi_p \end{bmatrix}_{j} - \mathbf{J}^{-1} \begin{bmatrix} \|{}^b\mathbf{R}_p {}^p\mathbf{p}_1 + {}^b\mathbf{d}_p - {}^b\mathbf{b}_1\| - l_1 \\ \vdots \\ \|{}^b\mathbf{R}_p {}^p\mathbf{p}_6 + {}^b\mathbf{d}_p - {}^b\mathbf{b}_6\| - l_6 \end{bmatrix} \tag{2.34}$$

provided that $\mathbf{J}$ is non-singular. Note that it is not necessary to compute the inverse of $\mathbf{J}$ ($O(n^3)$) explicitly, thus saving time. We use the LU decomposition instead because it has a lower operation count, $O(n^2)$.

We applied Newton's method to several case studies (see Table 2.4) in order to test its convergence. In cases one and three, the initial estimate of the platform's position and orientation is (0 m, 0 m, -2.5 m, 0°, 0°, 0°). By improving our initial guess of the platform's position and orientation, we reduce the number of iterations required for convergence as shown in cases two and four respectively. Cases five and six show similar results, however, because the platform is closer to a singular position/orientation, the number of iterations is higher. Finally, in case seven, we see that even if we start with a "good" initial estimate of the platform's position and orientation, the number of iterations required for convergence is still high because we are very close to a singular position/orientation. To help improve the convergence, we are currently investigating the use of modified Newton methods (See [25] for C source code).

Table 2.4: Examples Using Newton's Method.

Case	Position	Results (m, m, m, °, °, °)	Time (s)	Number of Iterations
1	Guess	(0.0, 0.0, -2.5, 0, 0, 0)		
	Desired	(0.5, 0.0, -2.5, 0, 0, 0)		
	Actual	$(0.5, -1.04987 \times 10^{-17}$, -2.5, $-4.9643 \times 10^{-17}, 1.4814 \times 10^{-7}, 1.6039 \times 10^{-16})$	0.018	4
2	Guess	(0.49, 0.0, -2.5, 0, 0, 0)		
	Desired	(0.5, 0.0, -2.5, 0, 0, 0)		
	Actual	$(0.5, -9.36948 \times 10^{-21}$, -2.5, $1.03858 \times 10^{-19}, 8.1727 \times 10^{-10}, -2.2868 \times 10^{-19})$	0.014	3
3	Guess	(0.0, 0.0, -2.5, 0, 0, 0)		
	Desired	(0.0, 0.0, -2.5, 0, 45, 0)		
	Actual	$(1.07369 \times 10^{-11}, -1.48721 \times 10^{-16}$, -2.5, -6.1343×10^{-15}, 45, $-6.9181 \times 10^{-16})$	0.022	5
4	Guess	(0.0, 0.0, -2.5, 0, 43, 0)		
	Desired	(0.0, 0.0, -2.5, 0, 45, 0)		
	Actual	$(1.6586 \times 10^{-7}, -3.8722 \times 10^{-19}$, -2.5, 1.1119×10^{-16}, 45, 0)	0.013	3
5	Guess	(0.0, 0.0, -2.5, 0, 0, 0)		
	Desired	(0.0, 0.0, -2.5, 45, 45, 0)		
	Actual	$(-1.2139 \times 10^{-12}, 4.3792 \times 10^{-11}$, -2.5, 45, 45, $-3.2676 \times 10^{-4})$	0.06	13
6	Guess	(0.0, 0.0, -2.5, 43, 43, 0)		
	Desired	(0.0, 0.0, -2.5, 45, 45, 0)		
	Actual	$(-2.4712 \times 10^{-13}, 8.8653 \times 10^{-12}$, -2.5, 45, 45, $-1.47 \times 10^{-4})$	0.045	10
7	Guess	(0.0, 0.0, -2.5, 59, 59, 0)		
	Desired	(0.0, 0.0, -2.5, 60, 60, 0)		
	Actual	$(1.1188 \times 10^{-11}, 1.1347 \times 10^{-11}$, -2.5, 60, 60, $-3.643 \times 10^{-4})$	0.061	13

Chapter 3

Design

We discuss in Chapter Three the design of each of the motion simulator's components. We explain the advantages of our inverted design and justify its dimensions. We run several inverse dynamic simulations to determine the required actuator performance. We also discuss the design of the hydraulic system including the power unit, distribution and actuator subsystems. The motion simulator base, the universal joints, position transducer mounting and the distribution manifold frame are all parts of the mechanical design of the simulator. Finally, in the electrical design section we discuss the two subsystems involved, namely, safety and control.

3.1 Motion Simulator Design

We decided on an inverted design for our motion simulator because it provides several advantages over traditional flight simulator designs. First, the actuators are under tension, thereby allowing them to be narrower without buckling. Second, the platform conserves space by rising up toward the ceiling when it is not in use. Third, if the system fails for any reason, the operator can lower the platform to a level attitude allowing him/her to dismount safely. Finally, because the platform may be lowered, the operator can easily enter and exit without climbing stairs or crossing ramps. This inverted arrangement, however, does have one obvious disadvantage, that is, trouble in performing regular maintenance of the hydraulic system as it is difficult to reach. Yet, in spite of this undesired consequence, the alternative, mounting the actuators with their blind ends down, results in a more severe problem as the hoses and cables may become entangled.

To provide the operator with a comfortable environment, we chose a platform radius of

0.5 meters. This will provide enough space for a chair, hand controls and room for the operator to move without injury or discomfort during a simulation. To prevent the joints from interfering with one another, we chose a platform actuator angle of 8° which gives a center-to-center joint separation of 5.5 inches providing just enough room to mount the joints to the platform. In addition to the joints, the base ends of the actuators include hydraulic valves and manifolds which require a 12 inch separation, which results in a base actuator angle of 51°. We ran several inverse dynamic simulations, discussed later, with varying base radii in an attempt to minimize the actuator forces. We found that increasing the base radius reduces the actuator forces required for a given translational acceleration and decreasing the base radius reduces the actuator forces required for a given angular acceleration. Figures 3.8 and 3.9 show the differences in the actuator forces for two different base radii (0.75 m and 1 m) when the platform is accelerating horizontally in the x direction from its home position of (0 m, 0 m, -2.5 m, 0°, 0°, 0°). The width of the simulator room (3.6 m) was another factor which influenced the base radius. To maintain clearance between the walls and the actuators and to help reduce the actuator forces required for translational motion, the base radius is 1.0 meter. The platform's translational displacements are limited to ± 1 meter by the height (3.6 m) and width of the simulator room. The nominal height of the platform is 2.5 meters allowing the operator's legs to hang down.

3.2 Inverse Dynamic Simulations

To meet the desired platform performance as shown in Table 1.2, we had to determine the required performance of the actuators. To do this, we ran several inverse dynamic simulations as presented by Do *et al* in [12, 13]. Given the initial position, velocity and acceleration of the platform, the simulation program computes the actuator lengths, velocities, accelerations and forces at each time step using the following algorithm:

1. Initialize platform position and orientation, actuator lengths and angles.

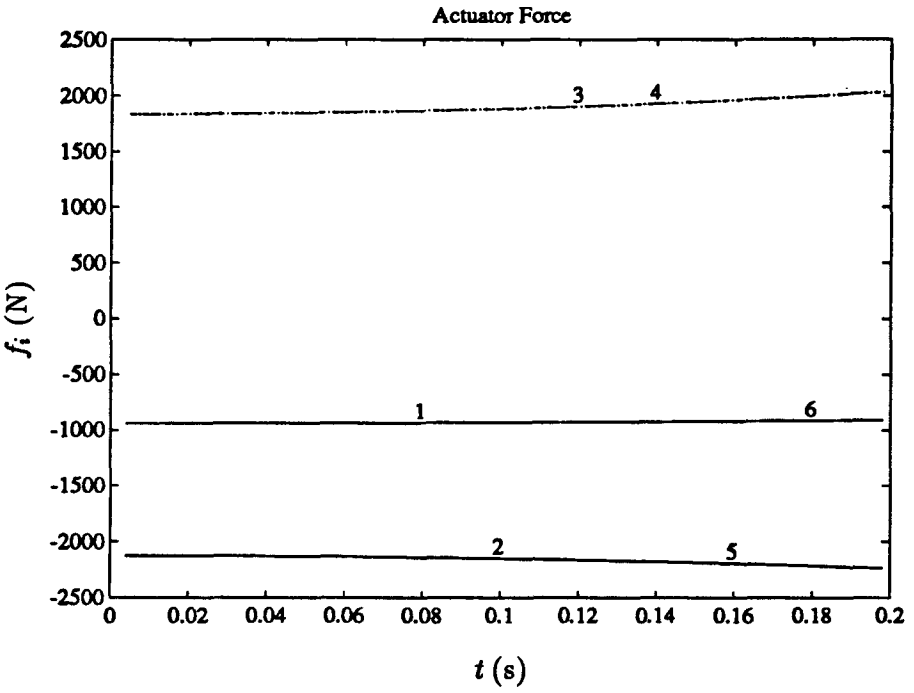


Figure 3.8: Actuator forces with a base radius of 0.75 m.

2. From the given acceleration compute the position, orientation and velocity of the platform at time $t + \Delta t$. For example,

$$v_{p_x} = a_{p_x}(t + \Delta t) + v_{p_{x0}} \text{ and} \tag{3.35}$$

$$x_p = x_{p0} + v_{p_{x0}}(t + \Delta t) + \frac{1}{2} a_{p_x}(t + \Delta t)^2 \tag{3.36}$$

3. For each actuator $i = 1 \ldots 6$:

   (a) Compute the coordinate frame attached to actuator i where $\mathbf{i}_i = {}^b\mathbf{a}_i / \|{}^b\mathbf{a}_i\|$ is the unit vector coincident with actuator i, $\mathbf{j}_i = (\mathbf{i}_i \times -\mathbf{k}) / \|\mathbf{i}_i \times -\mathbf{k}\|$ and $\mathbf{k}_i = \mathbf{i}_i \times \mathbf{j}_i$.

   (b) Compute the Euler angles of rotation, ψ_i, θ_i, ϕ_i, from the base frame to the frame attached to actuator i. Do *et al* use a 3-2-3 Euler rotation sequence so

$$\phi_i = \arctan\left(\frac{\mathbf{k}_{iy}}{\mathbf{k}_{ix}}\right), \tag{3.37}$$

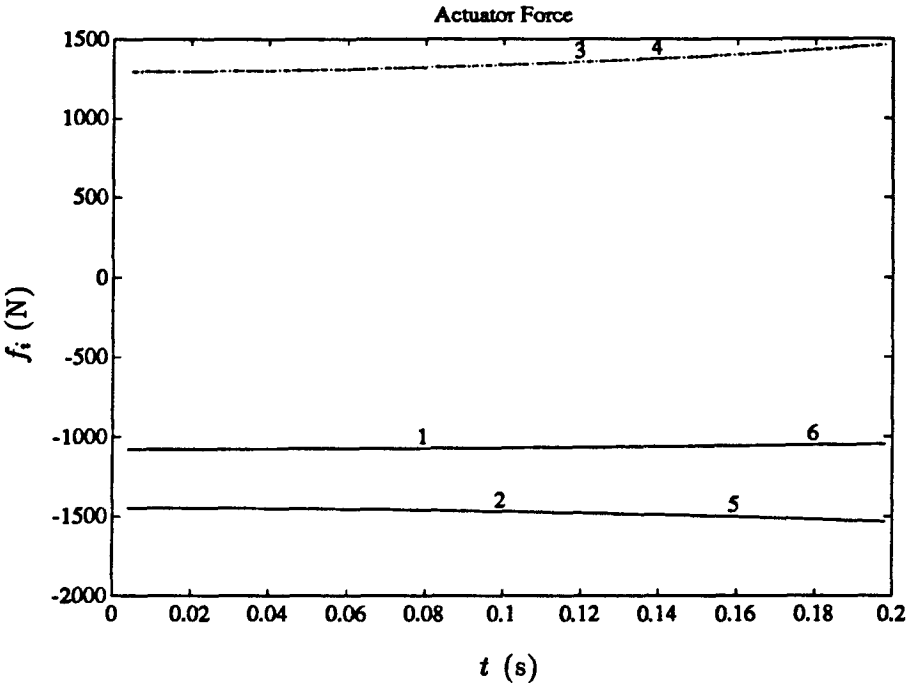


Figure 3.9: Actuator forces with a base radius of 1 m.

$$\theta_i = \arctan\left(\frac{\mathbf{k}_{ix}\cos\phi_i + \mathbf{k}_{iy}\sin\phi_i}{\mathbf{k}_{iz}}\right) \text{ and} \tag{3.38}$$

$$\psi_i = \arctan\left(\frac{-\mathbf{i}_{ix}\sin\phi_i + \mathbf{i}_{iy}\cos\phi_i}{-\mathbf{j}_{ix}\sin\phi_i + \mathbf{j}_{iy}\cos\phi_i}\right). \tag{3.39}$$

(c) Compute the first and second time derivatives of the Euler angles, for example, $\dot{\psi}_i = [\psi_i(t+\Delta t) - \psi_i(t)]/\Delta t$.

(d) Compute the angular velocity and acceleration of actuator i in the attached coordinate frame. For a 3-2-3 rotation

$${}^{i}\omega_i = \begin{bmatrix} -\dot{\phi}_i \mathrm{s}\theta_i \mathrm{c}\psi_i + \dot{\theta}_i \mathrm{s}\psi_i \\ \dot{\phi}_i \mathrm{s}\theta_i \mathrm{s}\psi_i + \dot{\theta}_i \mathrm{c}\psi_i \\ \dot{\phi}_i \mathrm{c}\theta_i + \dot{\psi}_i \end{bmatrix} \text{ and} \tag{3.40}$$

$$
{}^{i}\alpha_i = \begin{bmatrix} -\ddot{\phi}_i \mathrm{s}\theta_i \mathrm{c}\psi_i - \dot{\phi}_i \left(\dot{\theta}_i \mathrm{c}\theta_i \mathrm{c}\psi_i - \dot{\psi}_i \mathrm{s}\theta_i \mathrm{s}\psi_i\right) + \ddot{\theta}_i \mathrm{s}\psi_i + \dot{\theta}_i \dot{\psi}_i \mathrm{c}\psi_i \\ \ddot{\phi}_i \mathrm{s}\theta_i \mathrm{s}\psi_i + \dot{\phi}_i \left(\dot{\theta}_i \mathrm{c}\theta_i \mathrm{s}\psi_i + \dot{\psi}_i \mathrm{s}\theta_i \mathrm{c}\psi_i\right) + \ddot{\theta}_i \mathrm{c}\psi_i - \dot{\theta}_i \dot{\psi}_i \mathrm{s}\psi_i \\ \ddot{\phi}_i \mathrm{c}\theta_i - \dot{\phi}_i \dot{\theta}_i \mathrm{s}\theta_i + \ddot{\psi}_i \end{bmatrix} \tag{3.41}
$$

where $\mathrm{c}\psi_i = \cos\psi_i$ and $\mathrm{s}\psi_i = \sin\psi_i$, etc.

(e) Compute the reaction forces perpendicular to actuator i (*i.e.* along $\mathbf{j}_i$ and $\mathbf{k}_i$). Using conservation of angular momentum and assuming that the actuators are narrow and that their center of mass is positioned half way along their length, then the perpendicular forces are

$$
f_{iy} = -\frac{{}^{i}\dot{I}_{iz} {}^{i}\omega_{iz} + {}^{i}\mathbf{I}_{izz} {}^{i}\alpha_{iz} + {}^{i}\mathbf{I}_{iyy} {}^{i}\omega_{ix} {}^{i}\omega_{iy} + m_a g \frac{l_i}{2} \mathbf{j}_{iz}}{l_i} \tag{3.42}
$$

$$
f_{iz} = \frac{{}^{i}\dot{I}_{iy} {}^{i}\omega_{iy} + {}^{i}\mathbf{I}_{iyy} {}^{i}\alpha_{iy} - {}^{i}\mathbf{I}_{izz} {}^{i}\omega_{ix} {}^{i}\omega_{iz} - m_a g \frac{l_i}{2} \mathbf{k}_{iz}}{l_i} \tag{3.43}
$$

(f) Compute the i^{th} column of the six by six matrix $\mathbf{J}^T$ (similar to the transpose of the platform's jacobian derived earlier) such that

$$
\mathbf{J}^T = \begin{bmatrix} \mathbf{i}_{1x} & \cdots & \mathbf{i}_{6x} \\ \mathbf{i}_{1y} & \cdots & \mathbf{i}_{6y} \\ \mathbf{i}_{1z} & \cdots & \mathbf{i}_{6z} \\ {}^{p}\mathbf{p}_{1y} {}^{p}\mathbf{R}_{131} - {}^{p}\mathbf{p}_{1z} {}^{p}\mathbf{R}_{121} & \cdots & {}^{p}\mathbf{p}_{6y} {}^{p}\mathbf{R}_{631} - {}^{p}\mathbf{p}_{6z} {}^{p}\mathbf{R}_{621} \\ {}^{p}\mathbf{p}_{1z} {}^{p}\mathbf{R}_{111} - {}^{p}\mathbf{p}_{1x} {}^{p}\mathbf{R}_{131} & \cdots & {}^{p}\mathbf{p}_{6z} {}^{p}\mathbf{R}_{611} - {}^{p}\mathbf{p}_{6x} {}^{p}\mathbf{R}_{631} \\ {}^{p}\mathbf{p}_{1x} {}^{p}\mathbf{R}_{121} - {}^{p}\mathbf{p}_{1y} {}^{p}\mathbf{R}_{111} & \cdots & {}^{p}\mathbf{p}_{6x} {}^{p}\mathbf{R}_{621} - {}^{p}\mathbf{p}_{6y} {}^{p}\mathbf{R}_{611} \end{bmatrix} \tag{3.44}
$$

where ${}^{p}\mathbf{R}_i = {}^{b}\mathbf{R}_p{}^{T} {}^{b}\mathbf{R}_i$ is a transform from the frame attached to actuator i to the platform's frame and ${}^{p}\mathbf{R}_{ijk}$ is the element from the j^{th} row and k^{th} column of ${}^{p}\mathbf{R}_i$.

4. Sum the perpendicular reaction forces and their corresponding torques in the base and platform frames respectively.

$$
s_1 = \sum_{i=1}^{6} (f_{iy} \mathbf{j}_{ix} + f_{iz} \mathbf{k}_{ix}) \tag{3.45}
$$

$$s_2 = \sum_{i=1}^{6} \left(f_{iy} \mathbf{j}_{i_y} + f_{iz} \mathbf{k}_{i_y} \right) \tag{3.46}$$

$$s_3 = \sum_{i=1}^{6} \left(f_{iy} \mathbf{j}_{i_z} + f_{iz} \mathbf{k}_{i_z} \right) \tag{3.47}$$

$$s_4 = \sum_{i=1}^{6} \left({}^p\mathbf{p}_{i_y} \left({}^p\mathbf{R}_{i32} f_{iy} + {}^p\mathbf{R}_{i33} f_{iz} \right) - {}^p\mathbf{p}_{i_z} \left({}^p\mathbf{R}_{i22} f_{iy} + {}^p\mathbf{R}_{i23} f_{iz} \right) \right) \tag{3.48}$$

$$s_5 = \sum_{i=1}^{6} \left({}^p\mathbf{p}_{i_z} \left({}^p\mathbf{R}_{i12} f_{iy} + {}^p\mathbf{R}_{i13} f_{iz} \right) - {}^p\mathbf{p}_{i_x} \left({}^p\mathbf{R}_{i32} f_{iy} + {}^p\mathbf{R}_{i33} f_{iz} \right) \right) \tag{3.49}$$

$$s_6 = \sum_{i=1}^{6} \left({}^p\mathbf{p}_{i_x} \left({}^p\mathbf{R}_{i22} f_{iy} + {}^p\mathbf{R}_{i23} f_{iz} \right) - {}^p\mathbf{p}_{i_y} \left({}^p\mathbf{R}_{i12} f_{iy} + {}^p\mathbf{R}_{i13} f_{iz} \right) \right) \tag{3.50}$$

where the sums s_i correspond to the i^{th} row of $\mathbf{J}^T$.

5. Compute the total force and torque on the platform in the base and platform frames respectively. The total force includes the platform's inertia, gravity and the perpendicular reaction forces. The total torque includes the platform's rotational inertia and the torque due to the perpendicular reaction forces,

$$\begin{bmatrix} {}^b\mathbf{f}_p \\ {}^p\tau_p \end{bmatrix} = \begin{bmatrix} m_p {}^b\mathbf{a}_{p_x} - s_1 \\ m_p {}^b\mathbf{a}_{p_y} - s_2 \\ m_p {}^b\mathbf{a}_{p_z} + m_p g - s_3 \\ {}^p\mathbf{I}_{p_{xx}} {}^p\alpha_{p_x} + \left({}^p\mathbf{I}_{p_{zz}} - {}^p\mathbf{I}_{p_{yy}} \right) {}^p\omega_{p_y} {}^p\omega_{p_z} - s_4 \\ {}^p\mathbf{I}_{p_{yy}} {}^p\alpha_{p_y} + \left({}^p\mathbf{I}_{p_{xx}} - {}^p\mathbf{I}_{p_{zz}} \right) {}^p\omega_{p_x} {}^p\omega_{p_z} - s_5 \\ {}^p\mathbf{I}_{p_{zz}} {}^p\alpha_{p_z} - s_6 \end{bmatrix} \tag{3.51}$$

where the angular velocity and acceleration of the platform in the platform frame are

$${}^p\omega_p = \begin{bmatrix} \dot{\psi}_p - \dot{\phi}_p \mathrm{s}\theta_p \\ \dot{\theta}_p \mathrm{c}\psi_p - \dot{\phi}_p \mathrm{s}\psi_p \mathrm{c}\theta_p \\ \dot{\theta}_p \mathrm{s}\psi_p + \dot{\phi}_p \mathrm{c}\psi_p \mathrm{c}\theta_p \end{bmatrix}, \tag{3.52}$$

$${}^p\alpha_p = \begin{bmatrix} \ddot{\psi}_p - \ddot{\phi}_p \mathrm{s}\theta_p - \dot{\theta}_p \dot{\phi}_p \mathrm{c}\theta_p \\ \ddot{\theta}_p \mathrm{c}\psi_p - \dot{\psi}_p \dot{\theta}_p \mathrm{s}\psi_p - \ddot{\phi}_p \mathrm{s}\psi_p \mathrm{c}\theta_p - \dot{\psi}_p \dot{\phi}_p \mathrm{c}\psi_p \mathrm{c}\theta_p + \dot{\theta}_p \dot{\phi}_p \mathrm{s}\psi_p \mathrm{s}\theta_p \\ \ddot{\theta}_p \mathrm{s}\psi_p + \dot{\psi}_p \dot{\theta}_p \mathrm{c}\psi_p + \ddot{\phi}_p \mathrm{c}\psi_p \mathrm{c}\theta_p - \dot{\psi}_p \dot{\phi}_p \mathrm{s}\psi_p \mathrm{c}\theta_p - \dot{\theta}_p \dot{\phi}_p \mathrm{c}\psi_p \mathrm{s}\theta_p \end{bmatrix} \tag{3.53}$$

and we assume that the platform is symmetric about the z_p axis so that ${}^{p}\mathbf{I}_{p_{xx}} = {}^{p}\mathbf{I}_{p_{yy}}$.

6. The actuator forces are obtained by solving

$$\mathbf{J}^T \begin{bmatrix} \mathbf{f}_{1x} \\ \vdots \\ \mathbf{f}_{6x} \end{bmatrix} = \begin{bmatrix} {}^{b}\mathbf{f}_p \\ {}^{p}\tau_p \end{bmatrix} \tag{3.54}$$

using the LU decomposition.

For the purposes of the simulation, each actuator is represented as a narrow rod with the moment of inertia about its principal axes given by

$$ {}^{i}\mathbf{I}_{i_{xx}} = 0, \tag{3.55}$$

$$ {}^{i}\mathbf{I}_{i_{yy}} = \frac{m_a l_i^2}{3} \text{ and} \tag{3.56}$$

$$ {}^{i}\mathbf{I}_{i_{zz}} = \frac{m_a l_i^2}{3} \tag{3.57}$$

where m_a is the mass of the actuator and l_i is its length. Assuming Figure 2.6 is a "good" representation of the motion simulator, we model the platform as a solid cylinder of radius r_p, height h_p and mass m_p. The inertias about the platform's principal axes are

$$ {}^{p}\mathbf{I}_{p_{xx}} = \frac{m_p\left(3r_p^2 + h_p^2\right)}{12}, \tag{3.58}$$

$$ {}^{p}\mathbf{I}_{p_{yy}} = \frac{m_p\left(3r_p^2 + h_p^2\right)}{12} \text{ and} \tag{3.59}$$

$$ {}^{p}\mathbf{I}_{p_{zz}} = \frac{m_p r_p^2}{2}. \tag{3.60}$$

Using the above algorithm and inertia formulas we ran a set of six simulations (for the actuator numbering convention see Figure 1.5). In the simulations we assume that the mass of the platform is 250 kg (≈500 lbs) and that the mass of each actuator is 40 kg (≈80 lbs). We estimate the total mass of the platform by including the operator's mass ≈200 pounds maximum, the mass of the seat ≈100 pounds and the mass of the platform frame and miscellaneous equipment ≈200 pounds. The first set of three simulations start at the platform's home position

and orientation (0 m, 0 m, -2.5 m, 0°, 0°, 0°) and accelerate the platform at 1 g (9.8 m/s^2) in each of the cartesian directions, x, y, z, for 0.3 seconds. The first simulation, Figures 3.10 and 3.11, show the geometrical symmetry of the platform about the x axis. Actuators 1, 2, 5 and 6 are pulling the platform forward (out of the page) while actuators 3 and 4 are pushing it. Actuators 2 and 5 are pulling with more force than actuators 1 and 6 because their angle with the x axis is smaller. The second simulation, Figures 3.12 and 3.13, demonstrate the platform's lack of symmetry about the y axis. Actuators 1, 2 and 4 are pulling the platform while actuators 5 and 6 are pushing. At the beginning of the simulation actuator 3 is pulling however at the end, it is pushing. In the third simulation, the symmetry of the platform about the z axis is shown in Figures 3.14 and 3.15. All six actuators are pulling the platform upward with their force increasing as the angle between the actuators and the z axis increases.

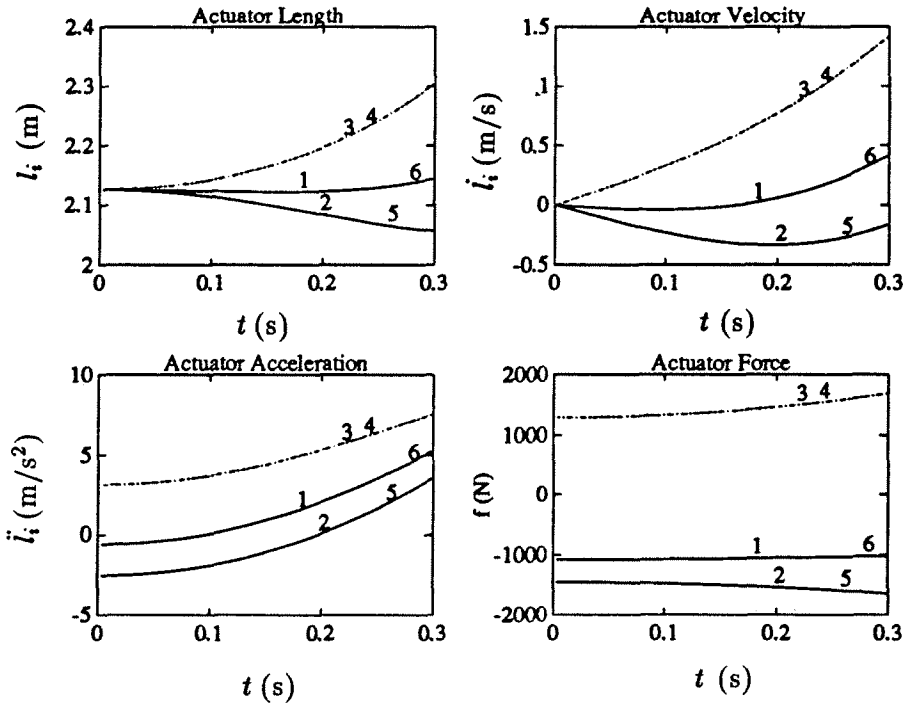


Figure 3.10: Acceleration in the x direction.

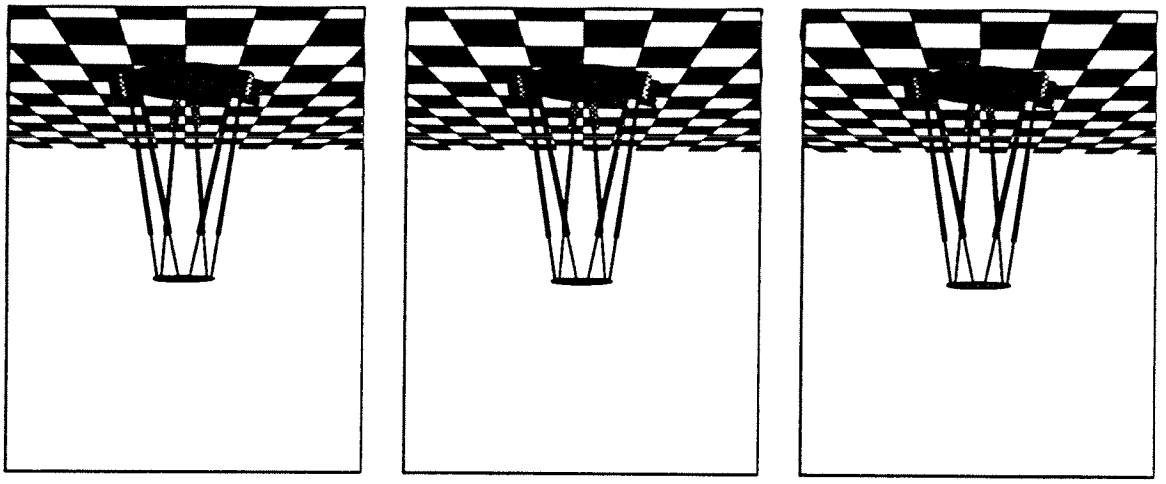


Figure 3.11: Frames showing acceleration in the x direction.

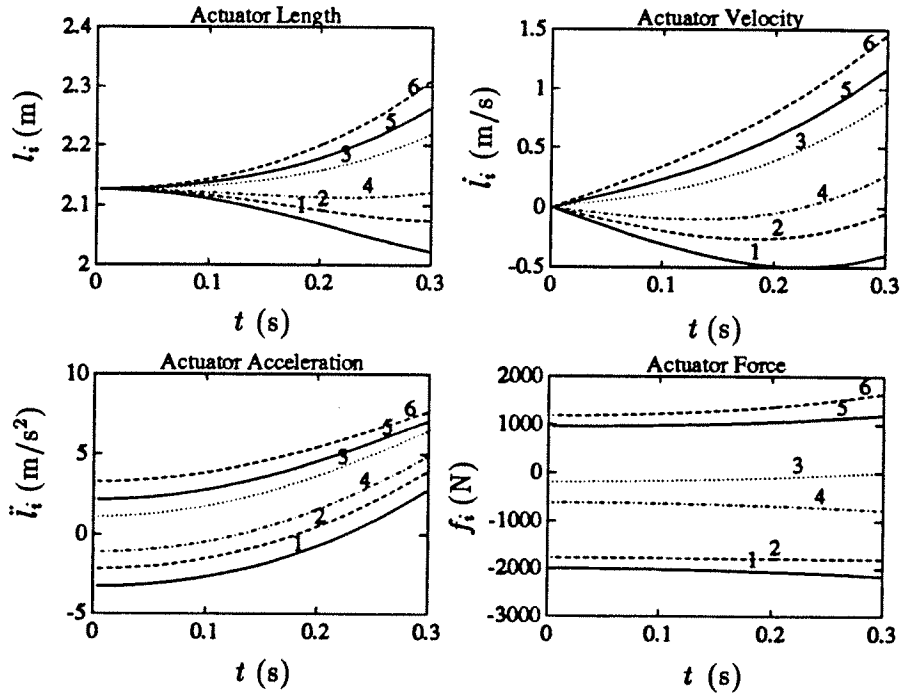


Figure 3.12: Acceleration in the y direction.

The second set of three simulations start at the platform's home position and orientation (0 m, 0 m, -2.5 m, 0°, 0°, 0°) and accelerate the platform at 400°/s^2 about the three cartesian

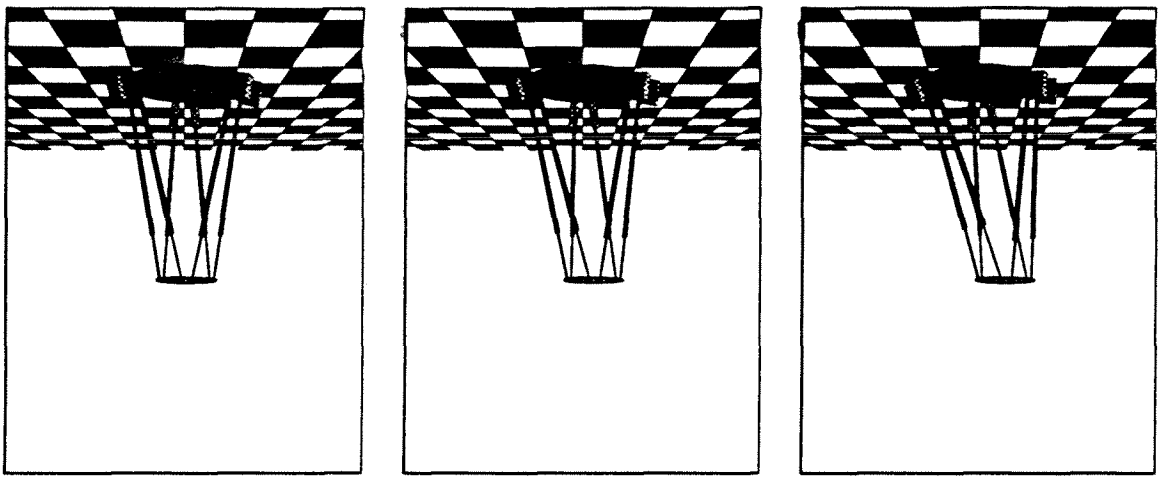


Figure 3.13: Frames showing acceleration in the y direction.

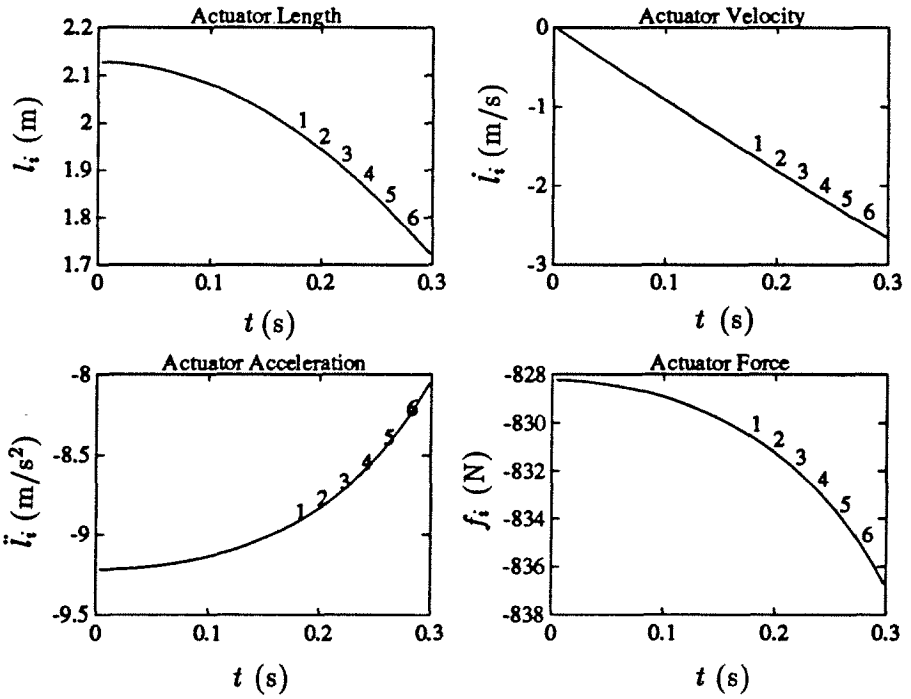


Figure 3.14: Acceleration in the z direction.

axes, x, y, z, for 0.3 seconds. For the rotation about the x axis, Figures 3.16 and 3.17, all the

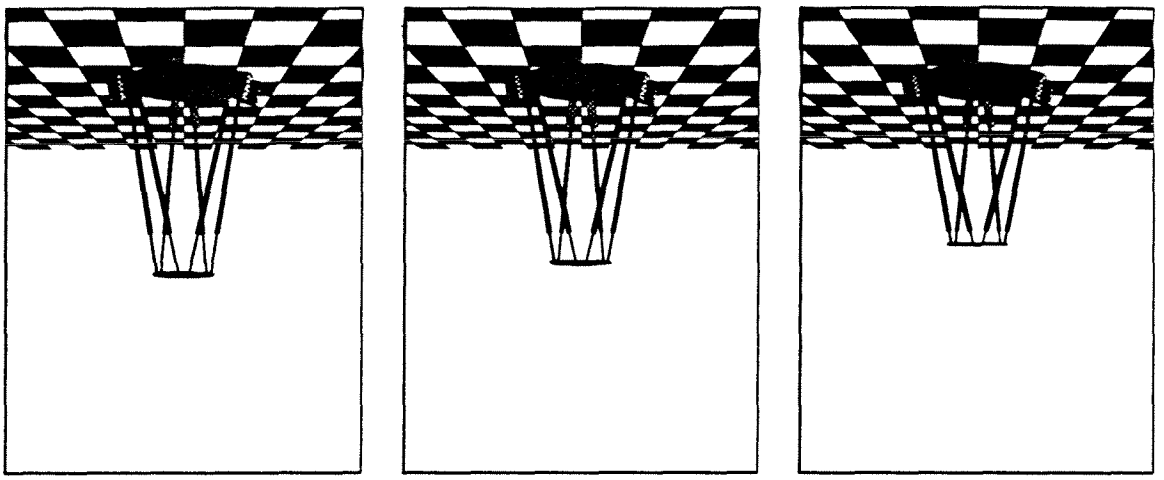


Figure 3.15: Frames showing acceleration in the z direction.

actuators are pulling to support the weight of the platform. Because the distance from the x axis to the end points of actuators 2, 3, 4 and 5 is much larger than the distance to the end points of actuators 1 and 6, actuators 2 and 3 are doing the most work in pulling their side of the platform up, while actuators 4 and 5 do very little work allowing their side of the platform to fall. Figures 3.18 and 3.19 show the result of accelerating the platform about the y axis. The actuators work in pairs to rotate the platform. Actuators 1 and 6 allow their side of the platform to fall while actuators 2, 3, 4 and 5 are pulling their side up. Because there are four actuators pulling up instead of just two as in the previous simulation, the actuator forces are smaller. Finally, Figures 3.20 and 3.21 show the platform's symmetry about the z axis. All actuators are pulling to support the weight of the platform, however, the odd numbered actuators are pulling more than the even numbered actuators because they have a positive component of force in the direction of rotation. The even numbered actuators, on the other hand, have a negative component of force in the direction of rotation so that they are pulling with less force.

In addition to the six simulations described above which start the platform from its home position and orientation, we ran some simulations which start the platform from other positions

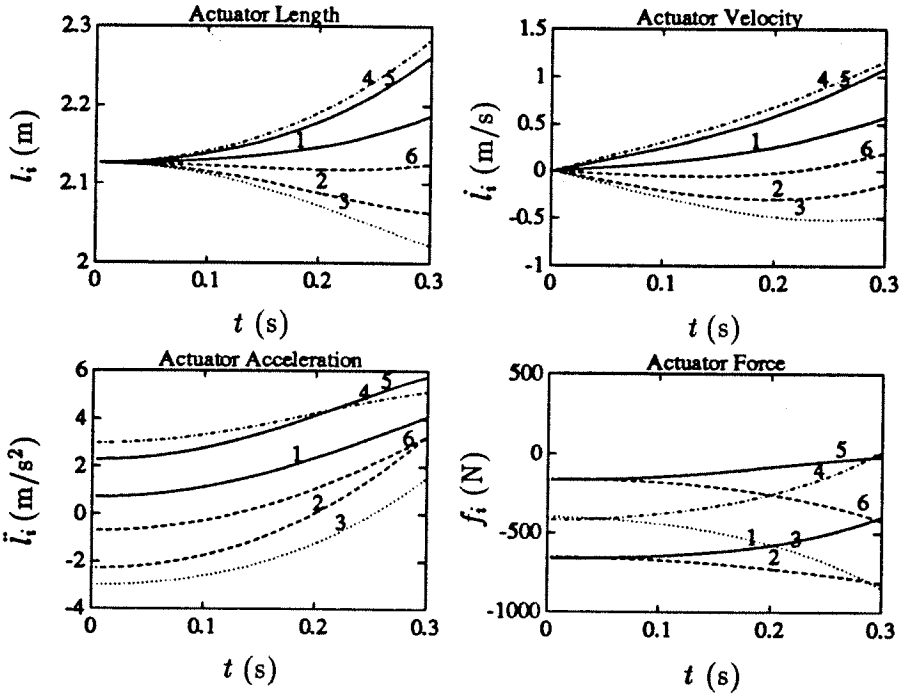


Figure 3.16: Acceleration about the x axis.

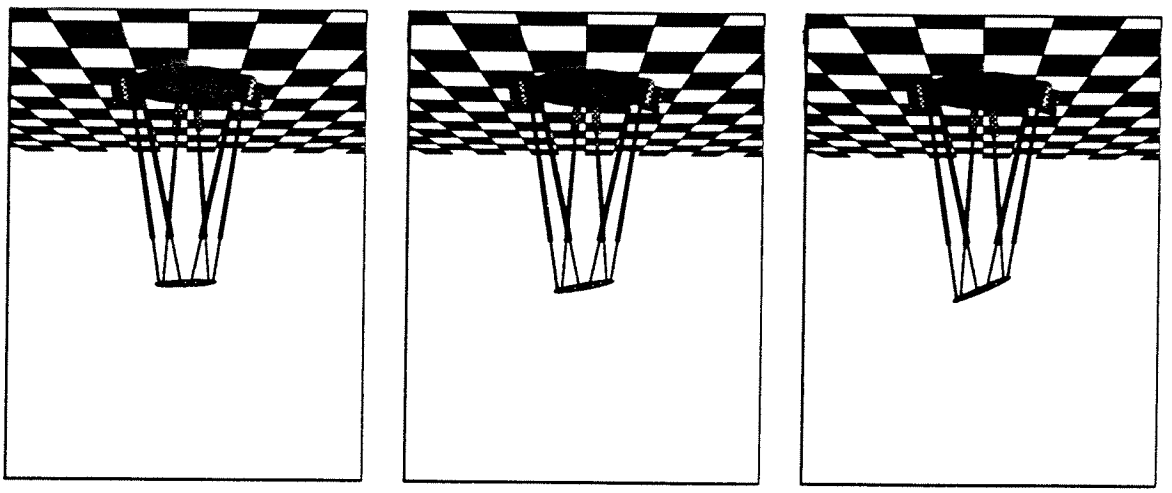


Figure 3.17: Frames showing acceleration about the x axis.

to see if the actuator forces changed significantly. As an example, Figures 3.22 and 3.23 show

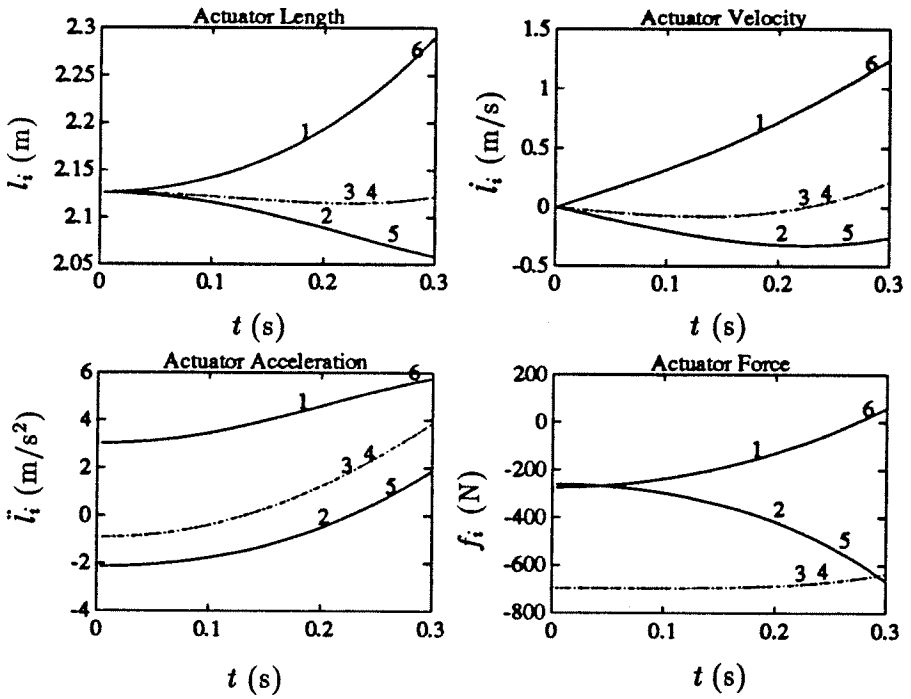


Figure 3.18: Acceleration about the y axis.

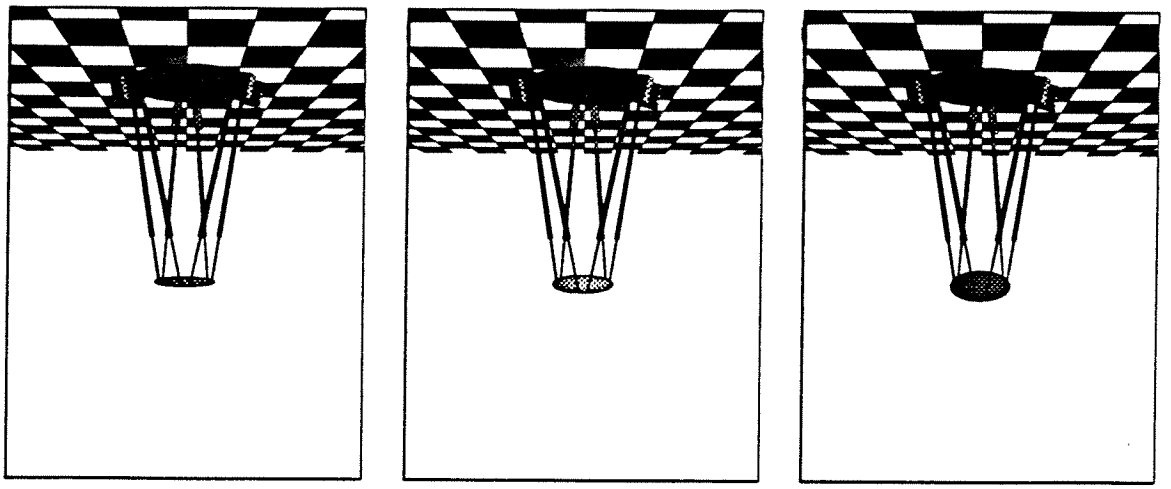


Figure 3.19: Frames showing acceleration about the y axis.

the results of accelerating the platform diagonally at 1 g from (0.75 m, 0.75 m, -3.0 m, 0°, 0°,

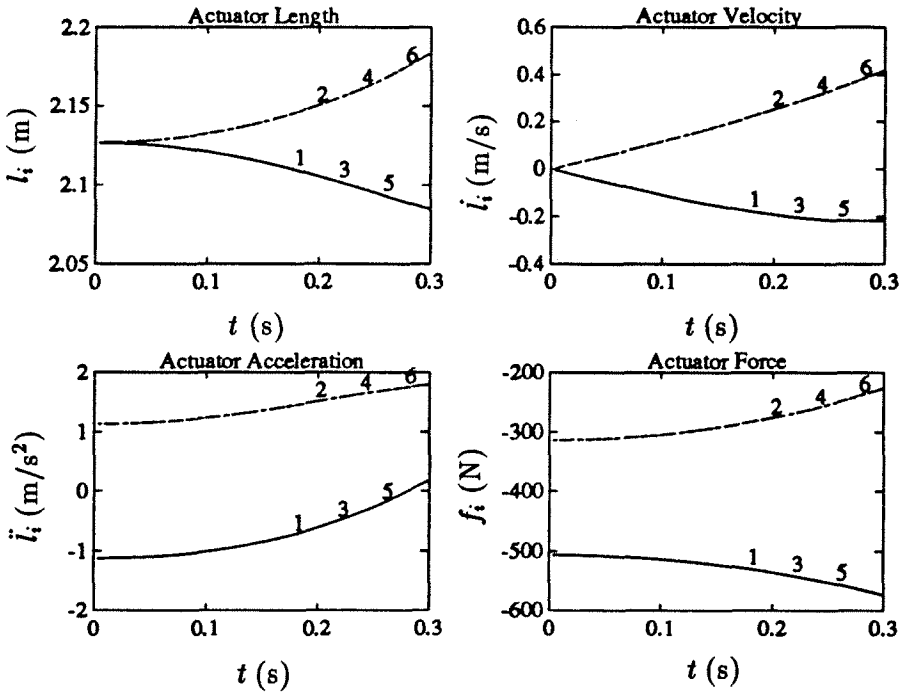


Figure 3.20: Acceleration about the z axis.

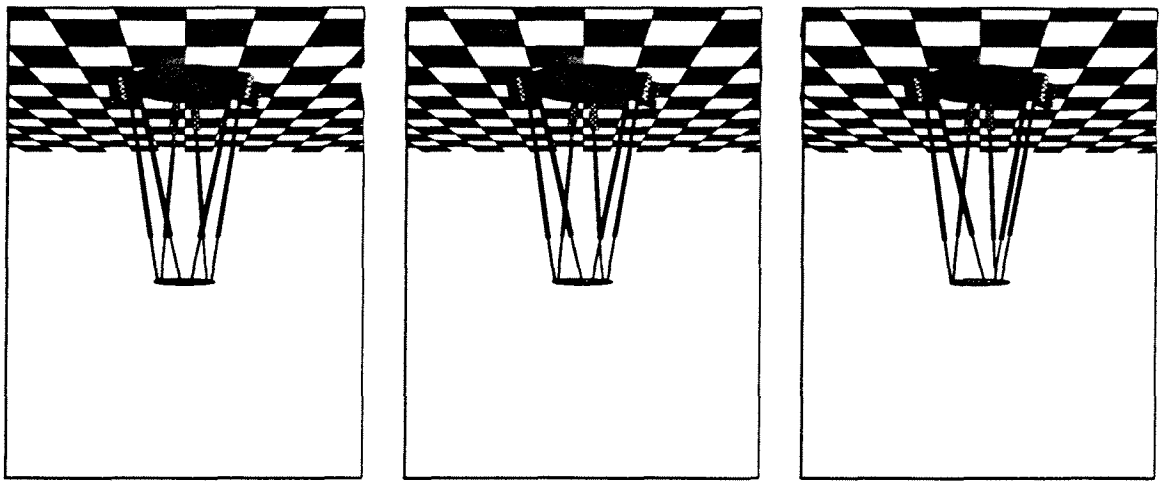


Figure 3.21: Frames showing acceleration about the z axis.

0°) for 0.3 seconds. These results indicate that the actuators should be capable of forces which

are somewhat higher than those in the previous simulations.

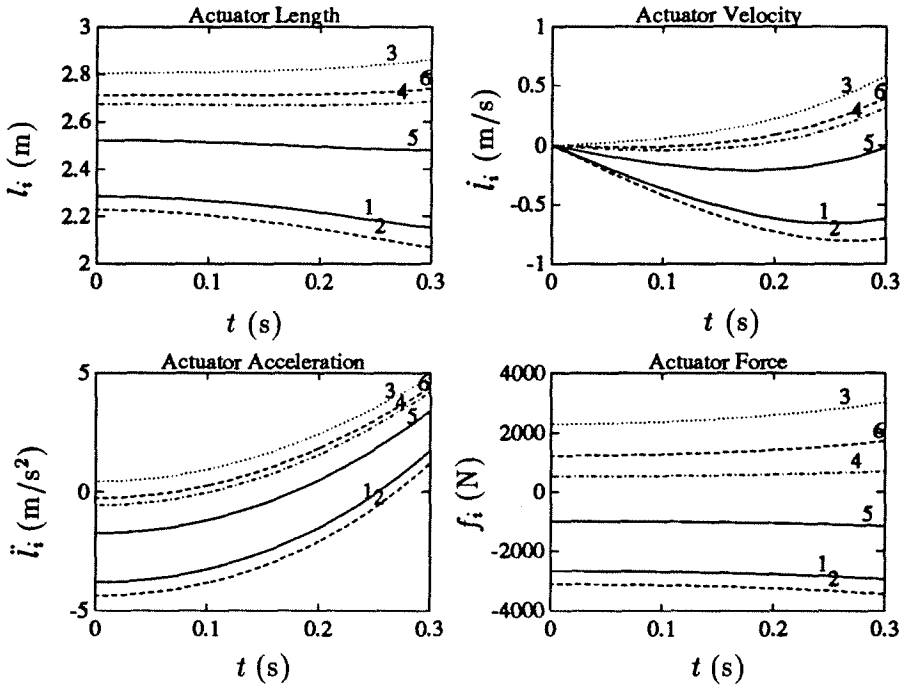


Figure 3.22: Diagonal Acceleration.

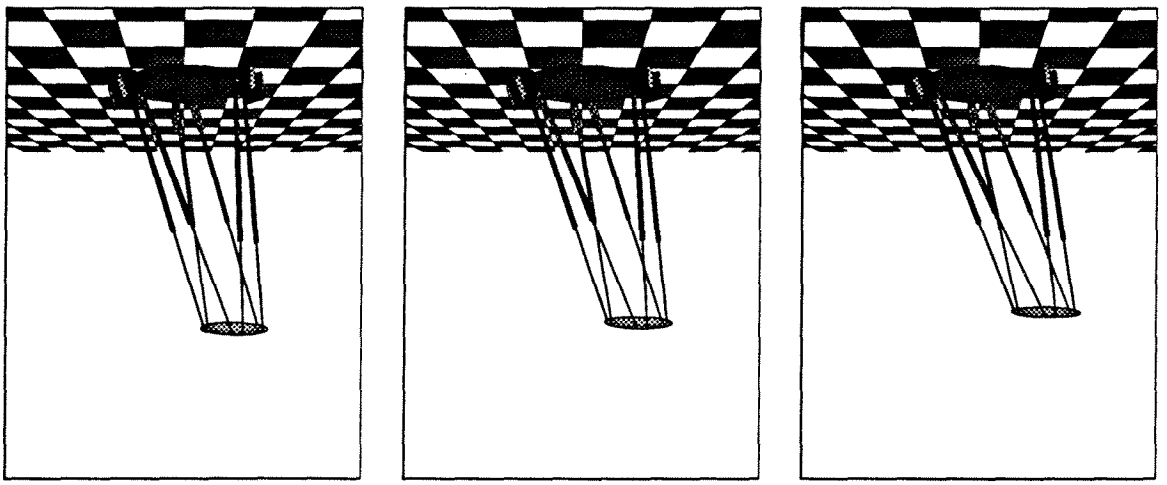


Figure 3.23: Frames Showing Diagonal Acceleration.

By combining the simulation results and the desired platform performance in Table 1.2 we arrive at the desired actuator performance shown in Table 3.5. We now proceed to determine the type, size and configuration of actuator that will meet these performance criteria.

Table 3.5: Desired Actuator Performance.

Force	±4000 N
Stroke	1.5 m (60 inches)
Velocity	±1.5 m/s
Acceleration	±11 m/s^2

3.3 Actuator Design

Table 3.5 indicates that the actuators must be capable of providing a force of 4000 Newtons at a speed of 1.5 meters per second. With this in mind, the obvious choice of actuator type is hydraulic because of its high force to weight ratio [27]. Initially, we wanted to use hydrostatic cylinders typically found in flight simulators because they operate with virtually no friction thus no noticeable "stick-slip" motion (or turn-around bump) is felt by the operator. Manufacturers achieve this frictionless motion because the piston and piston rod bushings are tapered which allows fluid to flow between the piston and cylinder and between the piston rod and rod seal keeping the piston and piston rod centered [28]. See Figure 3.24. Unfortunately, the machin-

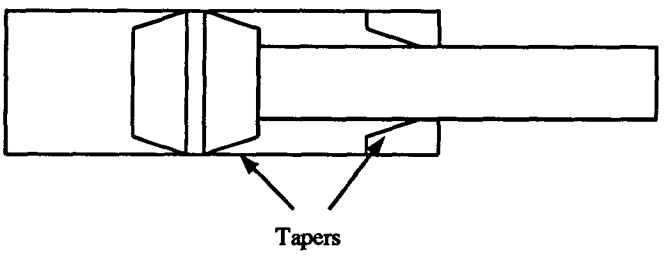


Figure 3.24: Hydrostatic Cylinder.

ing required to make hydrostatic cylinders is very precise which in turn makes the actuators

very expensive. We obtained quotes from Basic Hydraulics (Rexroth, Germany) and Wainbee (Parker Fluid Power, USA) which ranged from $5000 to $6000 per actuator (not including valves). As an alternative to buying hydrostatic cylinders, we considered machining standard cylinders to the required tolerances, however, local machines shops were unable to meet these tolerances for small bore cylinders (1 to 2 inches). Due to financial limits we were forced to use standard industrial actuators with low-friction, Teflon[1] based seals. Once we determined the actuator type, we had to determine its size. We chose the bore, rod diameter, stroke and operating pressure of the platform's actuators in order to satisfy several criteria. Firstly, the actuator had to be sufficiently rigid to prevent buckling. Secondly, to keep power unit and valve costs down, the fluid flow into the actuator, at a speed of 1.5 meters per second, had to be minimized. Thirdly, to maximize the platform's workspace, the actuator's stroke had to be maximized. Finally, we wanted an actuator that would meet our performance requirements of 4000 Newtons at 1.5 meters per second. To satisfy this last requirement, we had to consider the type of valve and its connection to the actuator as these determine the actuator's force and speed. In the discussion that follows we limit the actuator choices to asymmetrical 1, 1.5 and 2 inch bores (standard sizes) with 0.625, 1, and 1.375 inch rod diameters respectively. One inch bore actuators are the narrowest standard available while 2 inch bore actuators require the highest flow. We are not considering symmetrical actuators because they reduce the platform's workspace as they are twice as long as asymmetrical actuators when fully retracted.

To prevent injury to the operator and damage to the platform, we ensured that the actuators would not buckle when subjected to the maximum force requirement of 4000 Newtons. We used Euler's column formula [29], (3.61), to conservatively estimate the critical buckling force of the actuator. Euler's formula gives a conservative estimate because it assumes that we are considering a single, solid column of uniform diameter whose ends are free to pivot,

$$f_{cr} = \frac{\pi^2 EI}{L^2} \tag{3.61}$$

where E is the modulus of elasticity of the material (200 GPa for steel), I is the moment of

[1]Teflon is a trademark of the Du Pont Company.

area of the column and L is the column length. To simplify the calculation of the moment of area we assume that the diameter of the actuator is the same as the rod along the entire length. The moment of area is then given by

$$I = \frac{\pi R^4}{4} \tag{3.62}$$

where R is the column's radius. Figure 3.25 shows a plot of the critical buckling force versus length (when fully extended) for the three actuator sizes (See Appendix D for MATLAB source code). The results are as expected: the longer the actuator the lower the buckling force for a given bore. Therefore, 1.5 and 2 inch bore actuators are suitable for driving the platform while 1 inch bore actuators are not.

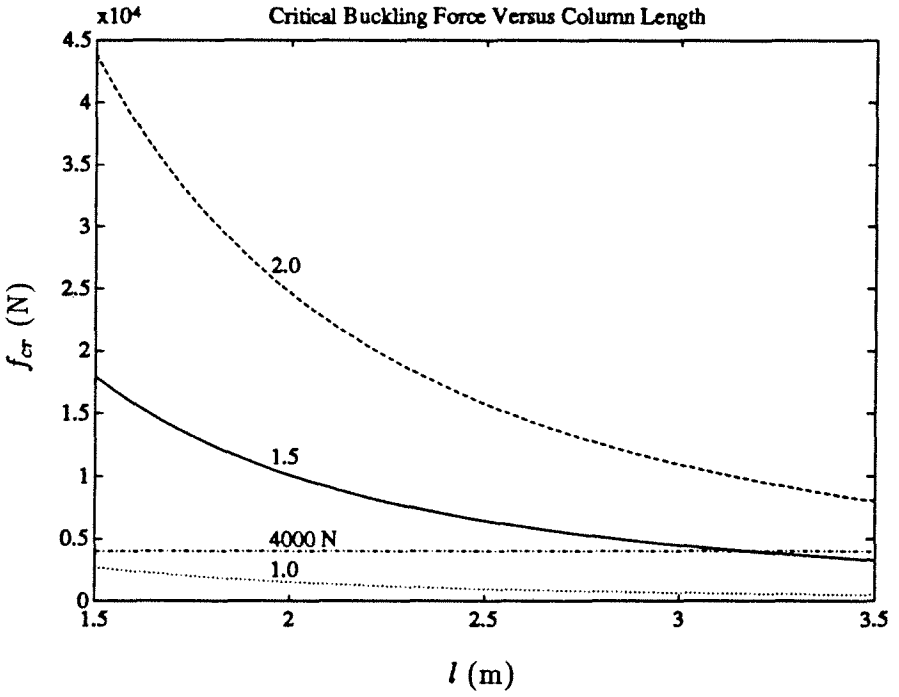


Figure 3.25: Actuator Critical Buckling Force.

After considering the problem of buckling we had to minimize the size of the power unit as it represents a large portion of the total system cost. The higher the flow for a given pressure, the higher the cost of the power unit. Also, as the flow increases the cost of valves increases

while their bandwidth decreases. When the platform is moving downward at 1.5 meters per second the total flow is six times the flow into each actuator. To calculate the flow into each actuator, we multiply the actuator's speed times its piston area. Table 3.6 shows the individual and total flows required for each of the three actuator sizes at a speed of 1.5 meters per second. To keep costs down we will use the smallest bore possible.

Table 3.6: Fluid Flow versus Bore for 1.5 m/s Actuator Speed.

Actuator Bore (inches)	Actuator Flow (GPM)	Total Flow (GPM)
1.0	12	72
1.5	27	162
2.0	48	298

Our final concerns, over and above preventing buckling and minimizing flow, are the type of valve and its connection to the cylinder. We looked at several servo and proportional valves manufactured by Moog Controls and Mannesmann-Rexroth. Generally speaking, servo valves are more linear and have a higher bandwidth than proportional valves. However, as the size of the valve increases (above 20 GPM), the bandwidth falls. Also, servo valves require a minimum pressure drop across both ports of 1000 PSI to operate properly while proportional valves can operate at around 500 PSI so the supply pressure is lower. After considering the Moog 78 series servo valves (55 Hz bandwidth at $\pm 40\%$ signal), 641 series proportional valves (30 Hz bandwidth at $\pm 40\%$ signal) and the small (100 Hz bandwidth at $\pm 25\%$ signal) and the medium (70 Hz at $\pm 25\%$ signal) size Rexroth 4WRDE three-stage proportional valves we chose the small Rexroth three-stage proportional valves. We found many advantages to the Rexroth valves. First, they feature a near linear response curve and large bandwidth (150 Hz at $\pm 5\%$, 100 Hz at $\pm 25\%$ and 35 Hz at $\pm 100\%$ of maximum valve opening) similar to that of servo valves making them easy to model mathematically. The valves achieve this performance by using a built-in control circuit that monitors the main stage spool position using a linear variable differential transducer

(LVDT). Second, the valves have a built-in isolation valve which closes in the case of system failure and stops the actuator, thereby preventing injury to the operator and damage to the motion simulator. Third, as described below, the valves are capable of providing the platform with the required forces (±4000 N) over most of the desired range of velocities (±1.5 m/s). Finally, the valves have a larger flow capacity for less cost than comparable servo valves.

When working with hydraulics there are two principal methods of connecting a cylinder and its control valve: four-way and three-way. In the four-way connection, the valve controls the flow into both sides of the actuator. This increases the actuator's stiffness and bandwidth because there are two small fluid chambers, one on either side of the piston. With asymmetric actuators, however, the four-way connection leads to pressure jumps when the actuator's speed changes sign because the unequal areas on either side of the piston are incompatible with the equal port areas of the valve [28]. These jumps will detract from the quality of the simulation and may damage the hydraulic system. Therefore, we will not use the four-way connection, instead, we will use the three-way connection shown in Figure 4.41. In the three-way connection, the valve controls the flow into the blind end of the actuator while the rod end remains at the constant supply pressure. Because there is only one fluid chamber, the actuator's stiffness (and bandwidth) is lower than in the four-way connection. An asymmetric, three-way connected actuator's output force is given by

$$f = P_c A - P_s a \tag{3.63}$$

where P_c is the cylinder pressure at the blind end, A is the piston area, P_s is the supply pressure and a is the annulus area between the cylinder and the piston rod. Furthermore, P_c is determined by the pressure drop across the valve for a given flow. If the actuator is extending, the flow q, into the blind end is given by

$$q = C_v \sqrt{P_s - P_c} \tag{3.64}$$

where C_v is the valve's flow/load coefficient when it is fully open [27]. If the actuator is

retracting, the flow out of the blind end is given by

$$q = C_v\sqrt{P_c}. \tag{3.65}$$

Solving (3.64) and (3.65) for P_c and substituting each result into (3.63) gives the following force versus flow equations

$$f = \left(P_s - \frac{q^2}{C_v^2}\right) A - P_s a, \quad q > 0 \tag{3.66}$$

$$f = \frac{q^2}{C_v^2} A - P_s a, \quad q < 0 \tag{3.67}$$

We use (3.66) and (3.67) to plot the force-velocity characteristic curves of the four valve and three cylinder combinations as shown in Figures 3.26, 3.27, 3.28 and 3.29 (See Appendix E for MATLAB source code). We created the plots with the highest possible supply pressure of 2500 PSI and use the valve specifications provided by Moog and Rexroth. All four plots indicate that the 2 inch bore cylinder can produce very high forces (>12,000 N). In three cases (Figure 3.28 excluded), however, the 2 inch bore cylinder is unable to achieve the desired velocity of 1.5 m/s. Also, the possibility of high forces in all four cases is undesirable as the resulting accelerations may harm the operator. In all four cases, the 1 inch bore cylinder is capable of relatively high speeds (>1.5 m/s), but as we saw earlier, it is not strong enough for our application. With the 1.5 inch bore cylinder the two Moog valves (Figures 3.26 and 3.27) and the small Rexroth proportional valve (Figure 3.29) produce similar performance characteristics. However, none of these valves are capable of the desired force over the entire velocity range because their flow/load coefficients are too small. The combination of the medium size Rexroth proportional valve and the 1.5 inch bore cylinder (Figure 3.28) is capable of producing the desired forces over a wide velocity range (>2 m/s) and would be the best choice for our application if it were not for our concerns about operator safety. If the actuator accidentally attains a high velocity and must then stop suddenly, the high acceleration could cause injury to the operator. With this in mind the 1.5 inch bore cylinder with the small Rexroth proportional

valve (Figure 3.29), although not able to deliver all the performance we would like, is the best choice for our application. Therefore, using the information obtained from the buckling, flow and valve studies, we chose to use 1.5 inch bore, 60 inch (≈1.5 m) stroke actuators with a 6 inch stop-tube. The stop-tube, although reducing the actuator's effective stroke to 54 inches, increases its strength when it is near full extension. These actuators provide the platform with the desired forces (±4000 N) over a velocity range of -1.3 to 1.1 m/s.

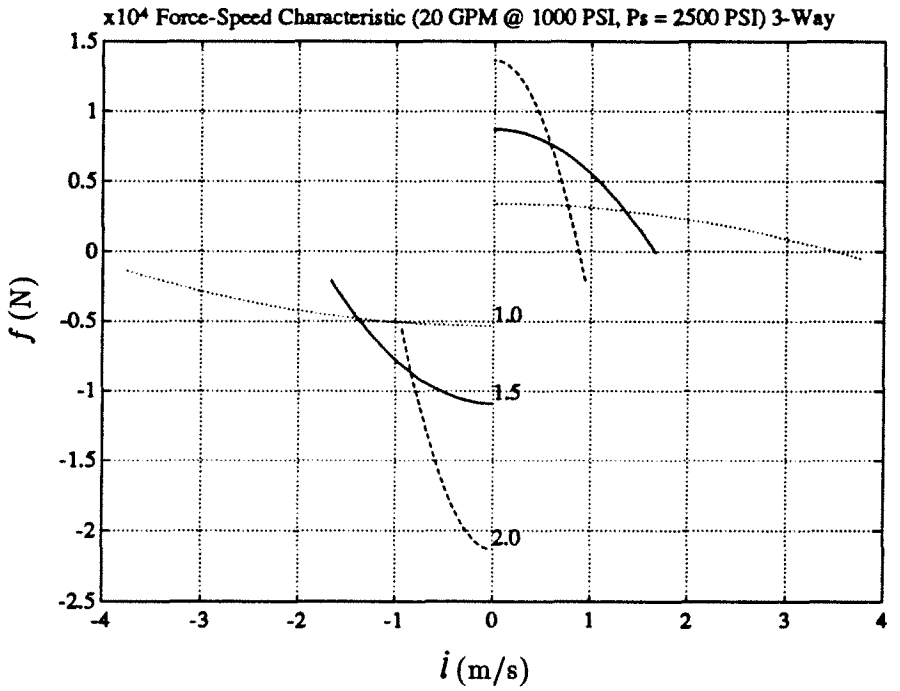


Figure 3.26: Force/Speed Characteristics: Moog Servo Valve & Varying Cylinder Bores.

3.4 Hydraulic System Design

The platform's hydraulic actuation system can be logically divided into three main subsystems. The first is the power unit subsystem which supplies and receives fluid to and from the rest of the system. The second is the distribution subsystem which filters, accumulates and distributes fluid to the actuators. And finally, the six actuator subsystems which control the fluid flow into

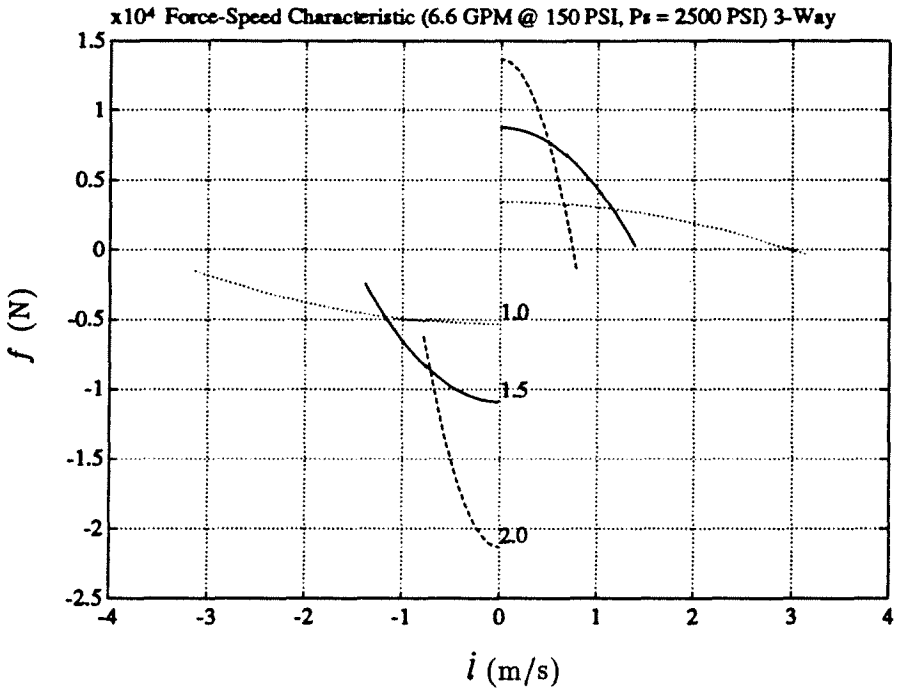


Figure 3.27: Force/Speed Characteristics: Moog Prop. Valve & Varying Cylinder Bores.

and out of each actuator.

3.4.1 Power Unit Subsystem

The power unit is a standard item designed by Basic Hydraulics meeting our pressure and flow requirements of 2500 PSI at 30 GPM. A schematic of the power unit is shown in Figure 3.30. At the heart of the unit is a compensated, swash plate type piston pump driven by a 40 horse power, three phase motor. After the user sets the desired system pressure, the compensator adjusts the pump flow according to the system's demand for fluid. To keep the noise level down, the pump is submerged in a 90 gallon reservoir. In addition to the pump, the power unit has a water cooled heat exchanger and a low pressure filter on the fluid return line. The heat exchanger controls the flow of water by monitoring the temperature of the fluid in the tank. When the temperature exceeds a predefined limit a valve opens allowing water to cool

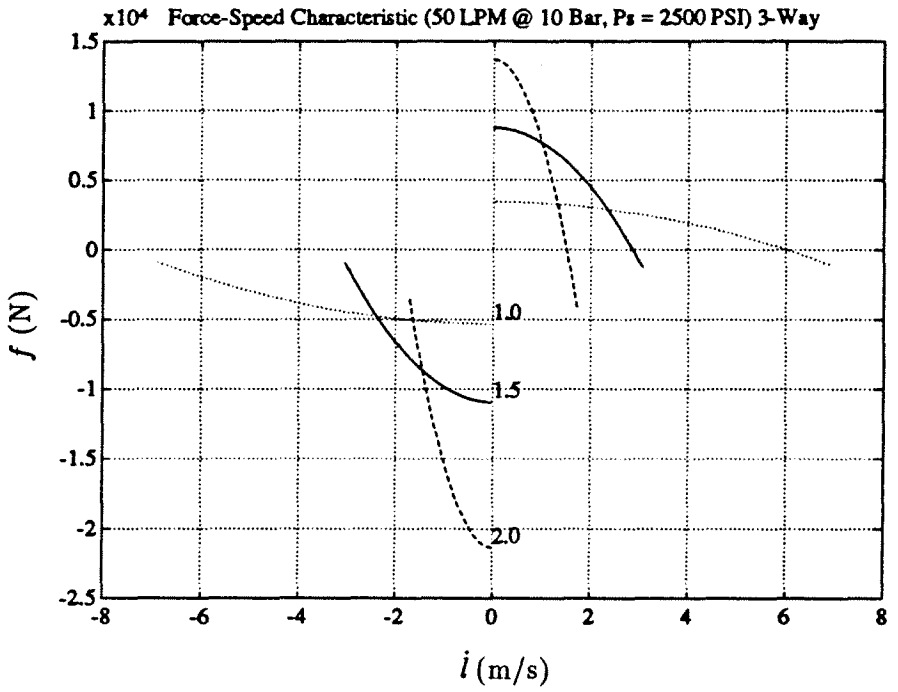


Figure 3.28: Force/Speed Characteristics: Medium Rexroth Valve & Varying Cylinder Bores.

the returning fluid. The filter removes dirt picked up by the fluid after leaving the actuators. When the filter is full, a mechanical, pop-up indicator tells the user to replace the filter element. The power unit subsystem also includes several other features including a pressure gauge which allows the user to accurately adjust the system pressure, a breather which filters air entering at the top of the tank, a fluid level indicator with a fluid level switch and a fluid temperature switch which are monitored by the computer control system. If the level is too low or the temperature is too high, the control system will shut the power unit down.

3.4.2 Distribution Subsystem

The distribution subsystem is made up of several components (See Figure 3.31). Starting where the fluid enters the distribution subsystem there is a ball valve. This valve is used to disconnect the power unit from the rest of the hydraulic system for maintenance purposes. Following

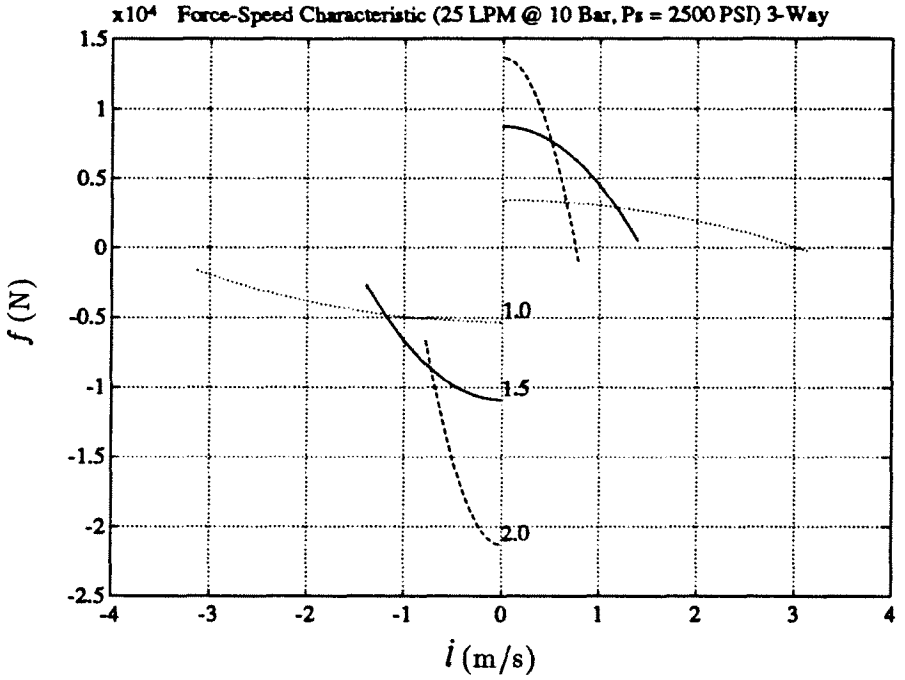


Figure 3.29: Force/Speed Characteristics: Small Rexroth Valve & Varying Cylinder Bores.

the ball valve there is a pressure filter which prevents fine particles of dirt from damaging the proportional valve spools and sleeves. The pressure filter has an electrical clogging indicator which is monitored by the computer system. As the filter fills with dirt, a pressure drop develops across it. When the pressure drop becomes too high, a switch closes interrupting the computer system which in turn performs a hydraulic system shutdown. Next to the pressure filter there are two 10 gallon, gas filled accumulators. These accumulators act like capacitors in electric circuits providing storage for extra fluid. When the system demands more fluid than the power unit can deliver, the accumulators make up the difference. If the platform requires a sudden, short acceleration burst, the power unit and the accumulators together provide the necessary fluid. After the burst, however, the accumulators must recharge before another one can take place. There are two valves that direct the fluid from the accumulators. The first valve, which is between the pressure filter and the accumulators, prevents a backwash of fluid through the

filter while the second valve allows the accumulators to drain back to the tank when the power unit shuts down, thereby reducing the fluid pressure to zero. Beyond the accumulators there is a distribution manifold and pilot filter assembly which distributes and collects fluid to and from the six actuators via eighteen flexible hoses (6 pressure hoses, 6 return hoses and 6 pilot hoses). The first stage of each proportional valve is separately piloted through the pilot filter which provides extra protection for the dirt sensitive first stage of the proportional valves. An electric pressure transducer on the distribution manifold allows the computer system to monitor the system pressure. If the pressure falls below the normal operating range then the control system initiates a power unit shutdown.

3.4.3 Actuator Subsystem

Each actuator subsystem is made up of six components (See Figure 3.32). First, there is a 1.5 inch bore, 54 inch stroke cylinder which drives the platform. If the control system fails and an actuator tries to fully retract, the cylinders have hydraulic "cushions" on their blind ends to slow the pistons down. Second, an actuator manifold provides a connection network and mounting base for the other components. The manifold mounts directly to the cylinder avoiding the use of hoses which would increase the compliance of the system. Third, there is a proportional valve which controls the fluid flow to the blind end of the cylinder including a built-in isolation valve which is normally energized and open allowing proper operation of the proportional valve. If there is a control system failure and an actuator tries to extend past a predetermined length, a switch at the rod end of the cylinder opens and deactivates all of the isolation valves. To prevent the platform from moving beyond the boundaries of the simulator room, the platform will connect to limit switches which are fixed to the wall and attached to the platform with chains of a predetermined length. If the platform pulls on one of the chains the switch opens and the isolation valves are deactivated, thereby stopping the platform. The fourth component in the actuator subsystem is a "home" valve which fully extends the actuator regardless of the condition of the proportional and isolation valves. The "home" valves

are operated manually and contain a velocity limiting orifice allowing us to move the platform downward and remove the operator safely. The fifth component is an adjustable pressure relief valve which regulates the pressure at the blind end of the cylinder. With this valve we can limit the maximum force and acceleration produced by the actuator. Finally, there is a provision for a differential pressure transducer which measures the pressure difference between the rod end and the blind end of the cylinder. We use the measurements from this transducer to determine the load on the actuator and adjust the controller response.

3.5 Mechanical Design

The mechanical design of the motion simulator consists of the base frame, universal joints, position transducer mounting and the distribution manifold frame. The following subsections contain an overview of the design of each of these components.

3.5.1 Motion Simulator Base

In order to support the motion simulator including the twelve universal joints, the six actuators, the platform frame, the seat and the operator, we decided to use a frame attached to the ceiling of the room. We contracted Jim Mandelli of Yolles Partnership Limited to design such a frame with several important specifications. First, the frame must support a maximum load of 4000 Newtons at each joint attachment point as well as allowing the load to be directed up to 45 degrees from vertical. Second, we wanted the radius from the center of the frame to the joint attachment points to be one meter (See Table 1.1) to minimize the actuator forces. Third, the frame must be as close as possible to the ceiling so that is does not interfere with the actuators and joints. Fourth, the center of the frame must lie near the center of the room in order to provide maximum clearance between the platform and the walls. Finally, the joint attachment points must be compatible with the joint base design and they must be at a 45 degree angle with the vertical to prevent the joints from exceeding their 90 degree bending limit. Figures 3.33 and 3.34 show the final design submitted by Jim Mandelli which meets all our requirements and

exceeds the requested loading capacity by more than 2.5 times. We then contracted Continental Steel to manufacture and install the three support frames.

3.5.2 Universal Joints

Because the platform has six degrees-of-freedom, each of the six actuators must have five passive degrees-of-freedom and one active degree-of-freedom. Each hydraulic cylinder has one active (prismatic) and one passive (rotary) degree-of-freedom in its cylindrical joint. The other four passive degrees-of-freedom are provided by two joints, one at either end of each cylinder. We considered using ball and socket joints because they have no singularities to avoid, however, they significantly restrict the platform's workspace. We also considered locally developed joints which would allow adjacent actuator endpoints to intersect. However, these joints would also restrict the platform's workspace and they do not provide enough room between adjacent actuators for the valves and length transducers. After looking at the joint designs of several flight simulators, we decided to use universal-type joints. These joints provide a large range of motion and are relatively easy to manufacture. Their one drawback, however, is that they have a singularity which we will avoid while operating the motion simulator.

Figure 3.35 shows the design of the motion simulator's joints. Each joint is made up of four main parts. The first is the base which bolts to either the platform or base frames and holds the cups of two tapered roller bearings (first degree-of-freedom) which share a common axis. We use tapered bearings rather than ball bearings because they withstand radial and longitudinal loads (in one direction) better. The second is the base shaft which rotates on the bearings inside the base and is held in by a nut and lock washer that thread onto the shaft. At the other end of the shaft there is a loop which holds two more bearing cups (second degree-of-freedom) whose common axis is perpendicular to the previous one. The third component changes depending on which end of the cylinder the joint attaches to. On the rod end there is the rod fork which threads on to the actuator's piston rod and is held in place by a pin. The rod fork has a bearing surface and circlip groove to attach the tube clamp plate (not shown) of the position

transducer mounting (described in a later section). On the blind end there is a cylinder fork which is held at the base of the cylinder by the cylinder's tie rods. The final component is the short shaft which connects the rod (or cylinder) fork to the base shaft. The short shaft is held by a shoulder at one end and a nut and lock washer at the other. See Appendix A for detailed design drawings of the joint components.

3.5.3 Position Transducer Mounting

In order for the control system to measure the length of each actuator, and hence determine the platform's position and orientation, we need to mount a position transducer on each actuator. We chose magneto-sonic transducers because they have been used successfully in many hydraulic applications, including flight simulators. They are not adversely affected by oil and dust and we obtained surplus units for $500 each (half the typical price). Normally, in hydraulic applications, the piston rod is bored out so that the transducer rod can fit inside. However, the piston rods in our actuators are relatively narrow so we had to mount the transducers along side the cylinders.

Figure 3.36 shows our first mounting idea. The transducer body is fixed at the blind end of the cylinder. A tube with the transducer magnet on one end slides over the transducer tube and through a guide, containing low friction bushings, which is fixed to the rod end of the cylinder. The other end of the tube is clamped to the piston rod so that when the actuator extends and retracts the magnet follows. Because each actuator must have six degrees-of-freedom: four degrees in the revolute joints at each end and two degrees in the cylindrical rod/cylinder joint, the piston rod must rotate relative to the cylinder and tube. Therefore, the piston rod is connected to the tube clamp with a bearing. Although this first idea seems reasonable and would probably work satisfactorily, we are concerned that the transducer tube will rattle inside the tube causing unwanted noise and possibly damaging the transducer. We modified the design as shown in Figure 3.37 so that the transducer tube is held firmly at both ends. We also made the tube more rigid by lengthening the tube guide. A prototype of this design was manufactured and tested on a single cylinder with favourable results. See Appendix B for

detailed design drawings of the transducer mounting components.

3.5.4 Distribution Manifold Frame

As mentioned in the hydraulic distribution system design subsection, there is a manifold which connects the six actuators via hoses to the pressure, pilot and tank lines of the power unit subsystem. To keep the hoses as short as possible we will position the manifold close to the center of the base of the motion simulator. To accomplish this we designed and manufactured a frame which bolts to the existing beam in the simulator room. Figure 3.38 shows the design of the distribution manifold frame.

3.6 Electrical Design

The motion simulator's electrical system is divided into two subsystems: the safety subsystem and the control subsystem. The safety subsystem includes components which do not affect the performance of the simulator. These include "home" and isolation valves, "chicken", actuator limit, platform limit, filter, fluid temperature and fluid level switches and a supply pressure transducer (See Figure 3.39). The "home" valves (V1–6) are connected in parallel and are activated by a momentary push button switch (S17). Operating the switch causes all the actuators to extend simultaneously, thereby, lowering the platform and allowing the operator to exit. The isolation valves (V7–12) are connected in parallel to a relay switch (S20) and are normally energized (open) allowing their corresponding proportional valves to function. If the simulator operator becomes disoriented or the computer operator discovers a malfunction, they can press their "chicken" switches (S18–19) which will de-energize the relay and the isolation valves which stops the platform. In addition, to prevent the actuators from fully extending and bottoming-out, possibly causing damage, and to prevent the platform from exceeding the physical limits of the simulator room, there are actuator and platform limit switches. These switches are normally closed and are connected in series with the "chicken" switches. If any one of these switches opens, all the actuators will stop. The platform limit switches are actually

plugs which connect to receptacles on the walls, floor and ceiling. The plugs are connected to the platform with light chains of a predetermined length. If the platform moves too far, the plug pulls out and opens the circuit. The two filter switches (S1–2, one for the main pressure filter and one for the pilot filter), the fluid temperature switch (S3), the fluid level switch (S4) and the supply pressure transducer are monitored by the computer control system. When the filters are nearing a full condition their switches close which tells the computer system to warn the operator to change the filter elements. Similarly, the fluid temperature and level switches close when the temperature or level exceed predefined limits. This closure may possibly indicate a cooling system failure or a leak. The supply pressure transducer returns a voltage proportional to the supply pressure to the computer via an analog to digital converter. If the pressure rises above or falls below predefined limits, possibly indicating a pump failure or leak, then the computer can warn the operator and shut down the power unit.

The second subsystem of the motion simulator's electrical system are six control subsystems, one per actuator. Each of these consist of three components: a three-stage proportional valve (V13–18), a length transducer and a differential pressure transducer (See Figure 3.40). First, the main-stage spool position of the three-stage proportional valve, and hence the flow into the blind end of the actuator, is controlled by the computer system through a digital to analog converter. The valve provides the computer system with the actual spool position through an analog to digital converter which allows the computer to verify whether the valve is responding to the commands it is sending. Second, the length transducer on each actuator provides a feedback signal proportional to its length. The transducer consists of two components: the transducer itself and the amplifier recirculation circuit. This circuit generates pulses which are converted to sound waves and sent down a wire inside the transducer tube. Magnets outside the tube constrict the wire making the sound wave return. The time required for the wave to return is directly proportional to the position of the magnets along the tube. Finally, the differential pressure transducer returns a voltage proportional to the difference in pressure between the cylinder's blind and rod ends to the computer through an analog to digital converter. These

pressures are proportional to the load on the actuator and may be used to improve the control system's response by increasing the damping.

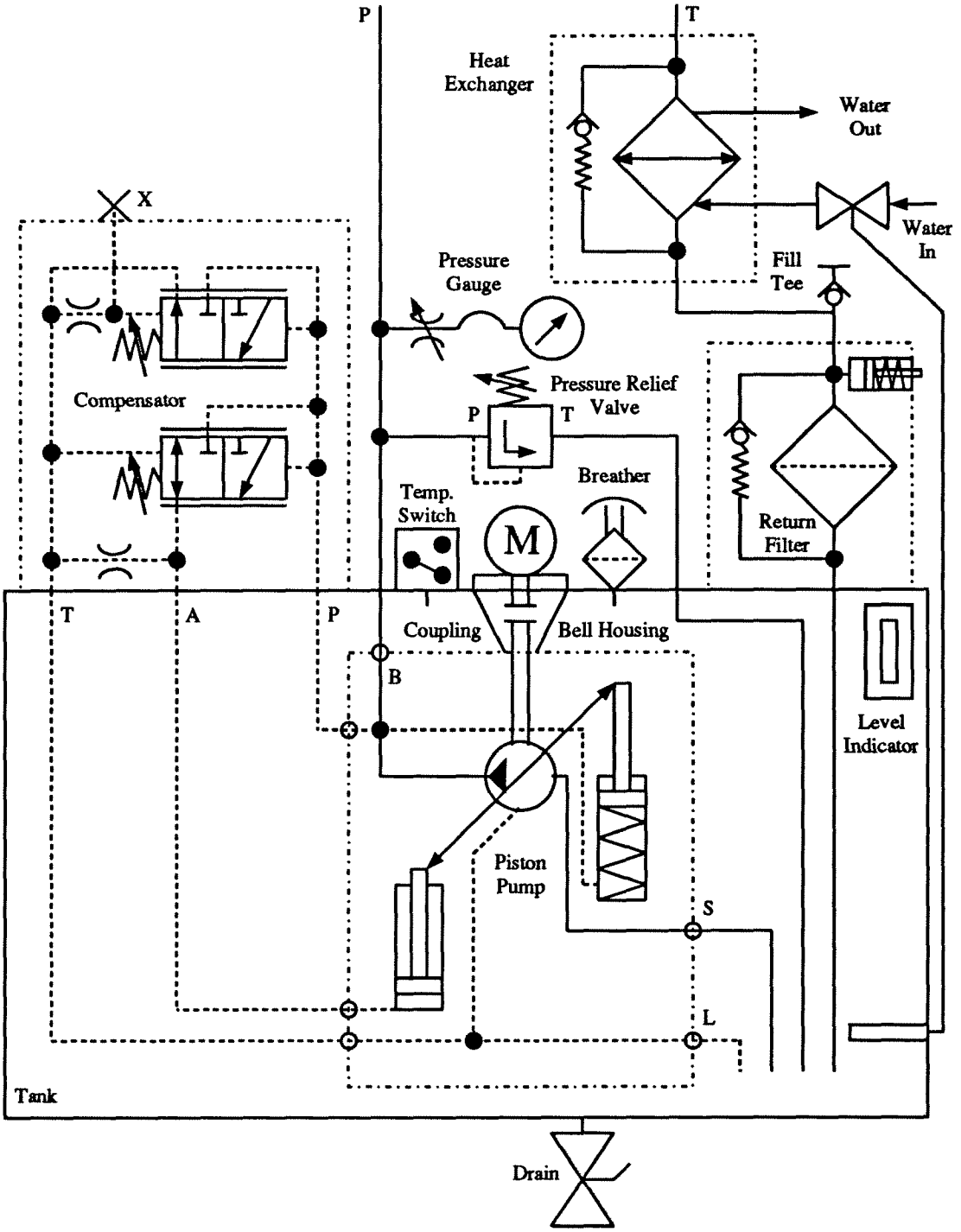


Figure 3.30: Hydraulic Power Unit Schematic.

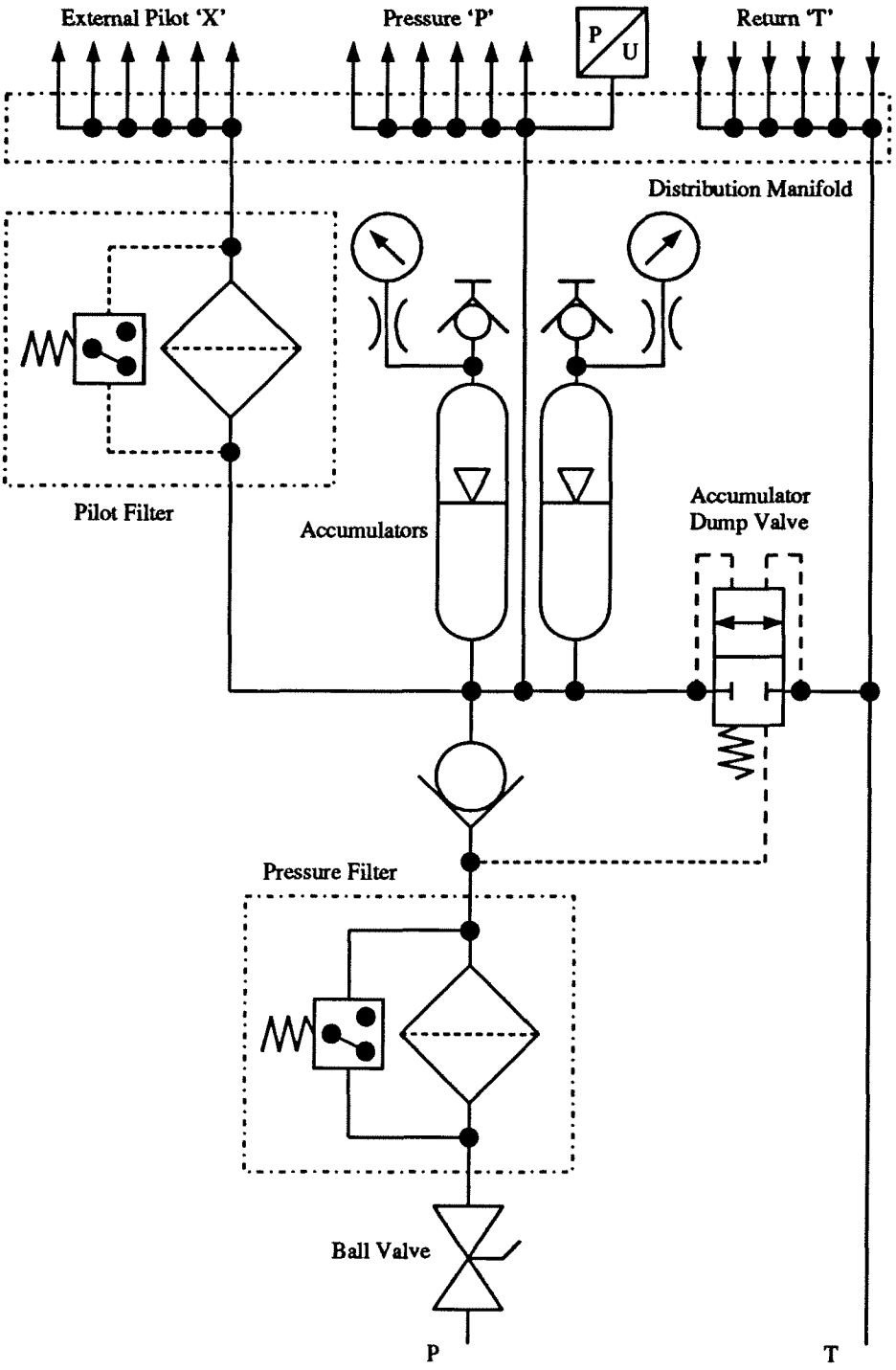


Figure 3.31: Distribution Subsystem Schematic.

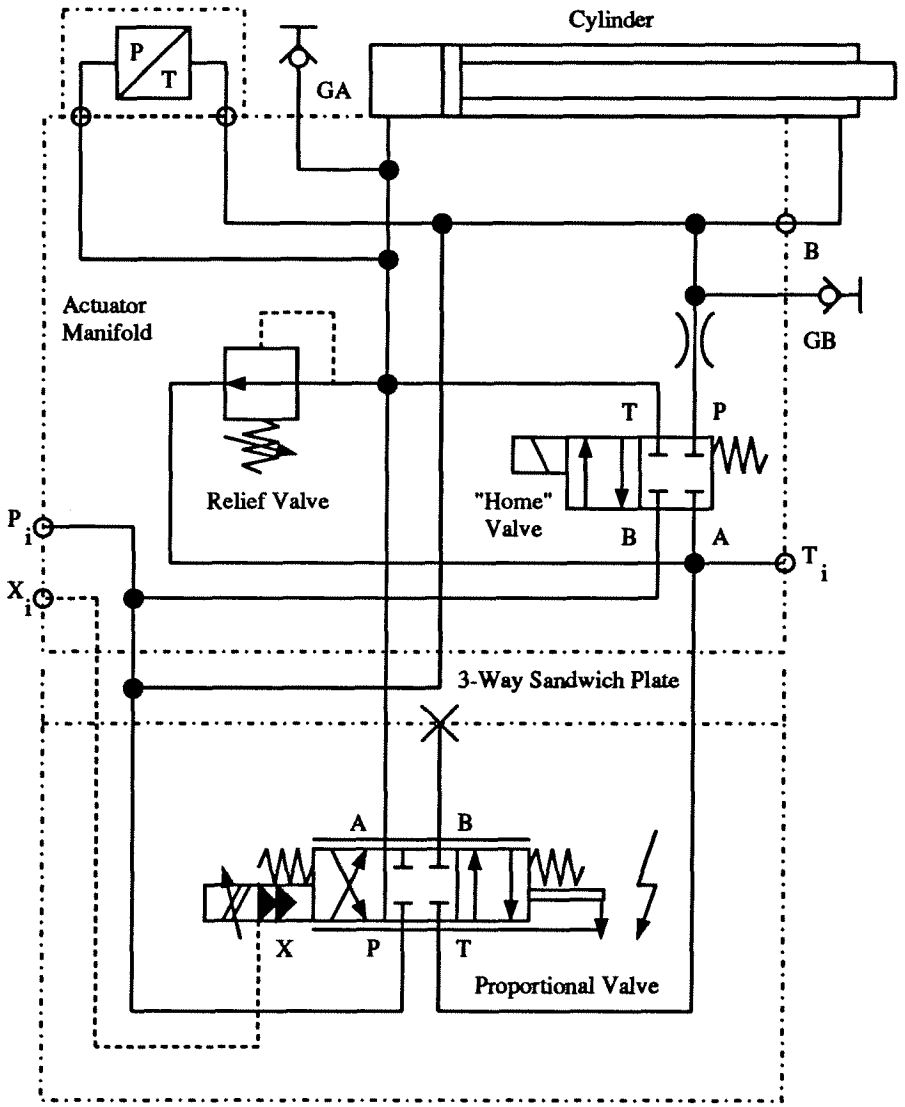


Figure 3.32: Actuator Subsystem Schematic.

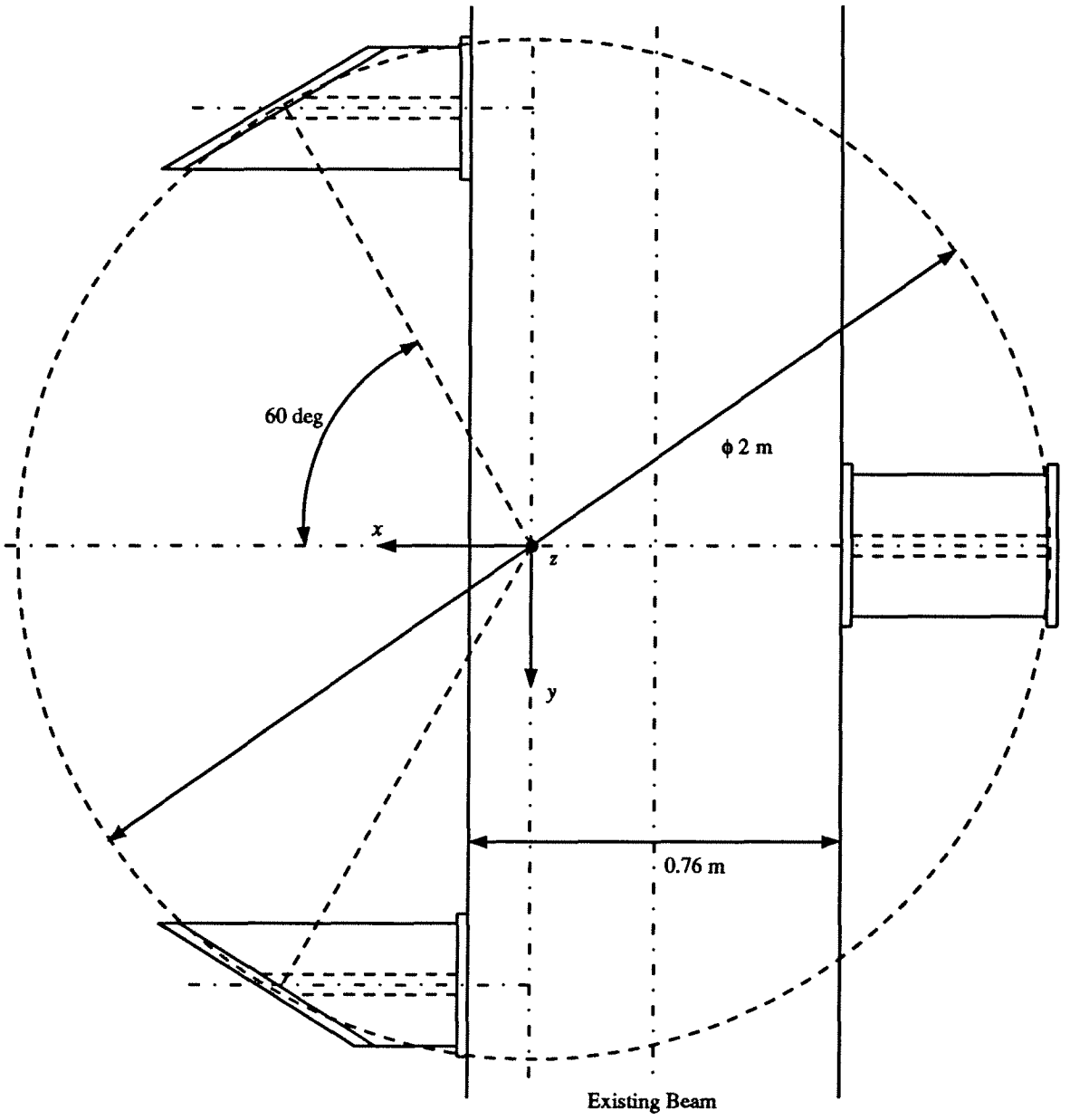


Figure 3.33: Motion Simulator Base Design—Plan View.

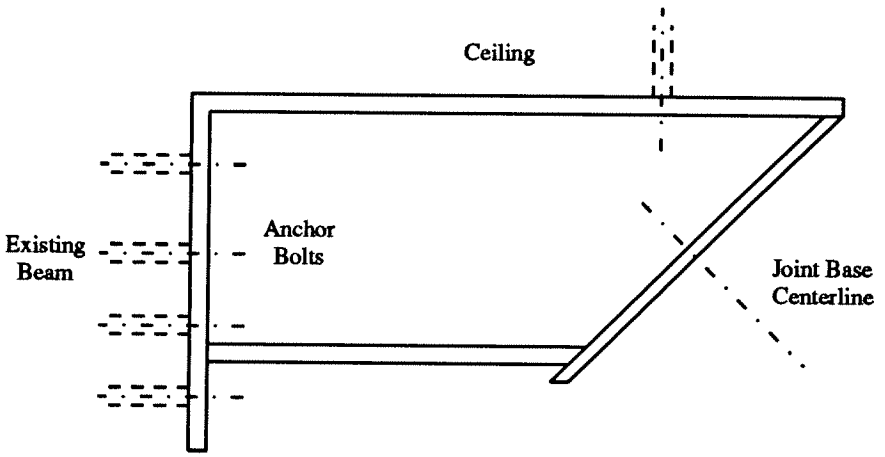


Figure 3.34: Motion Simulator Base Design—Profile View.

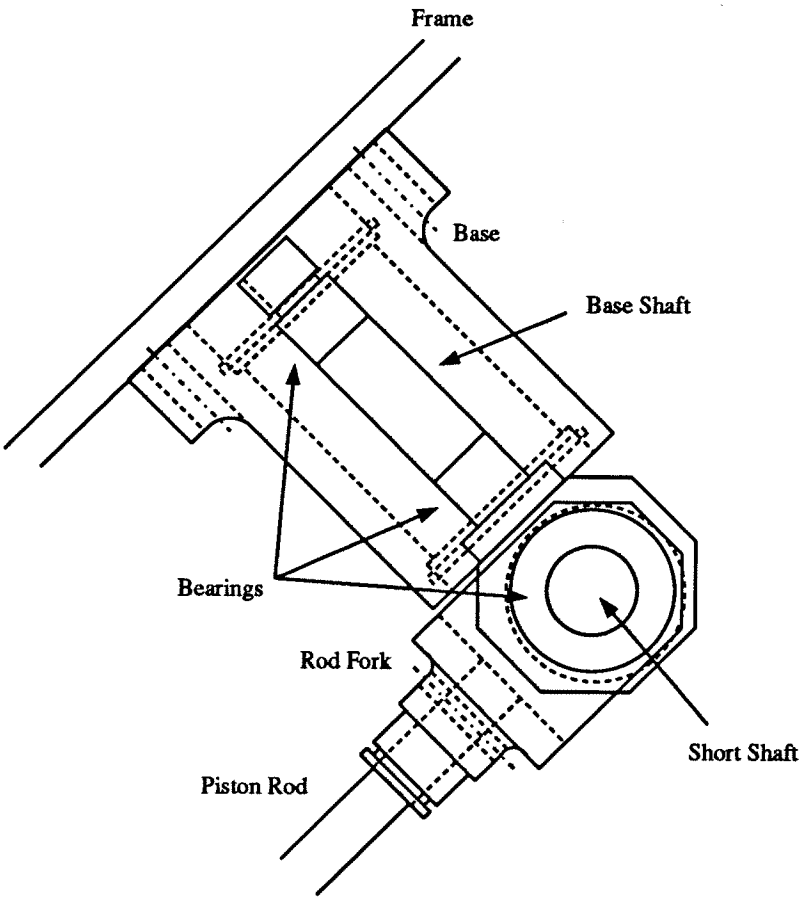


Figure 3.35: Universal Joint.

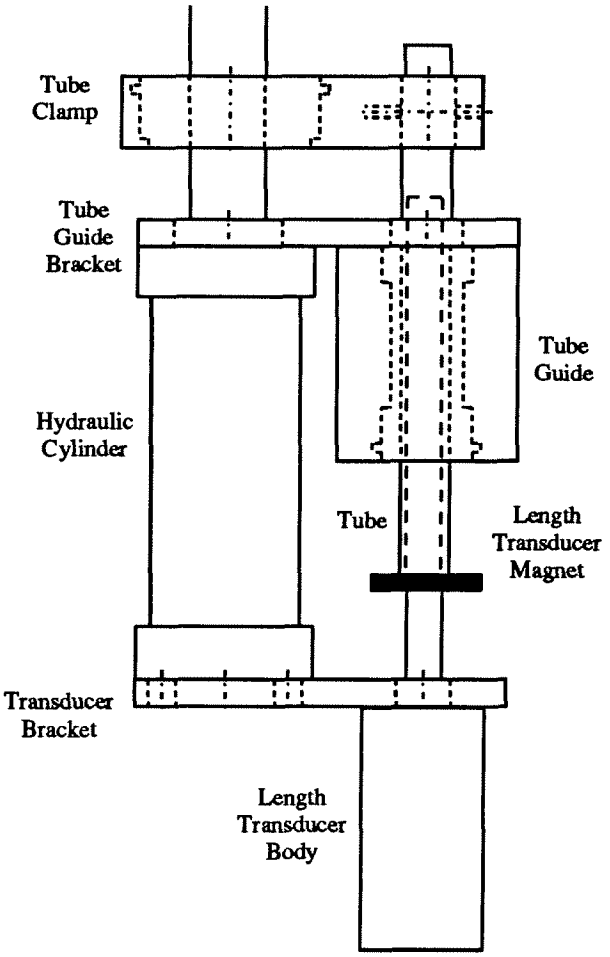


Figure 3.36: Initial Transducer Mounting Design.

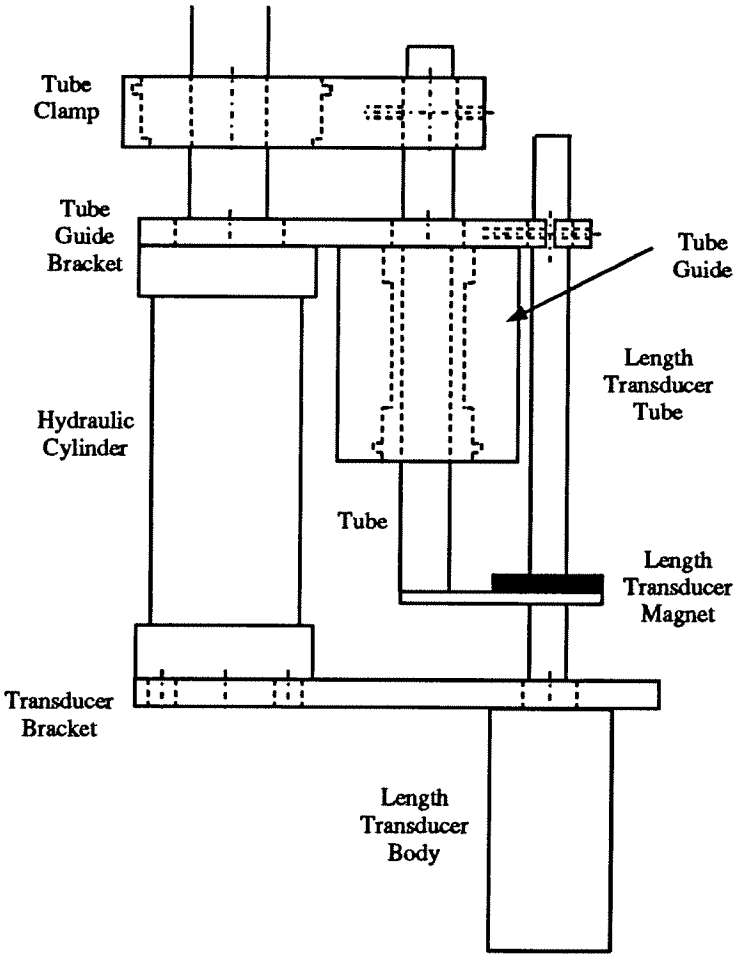


Figure 3.37: Final Transducer Mounting Design.

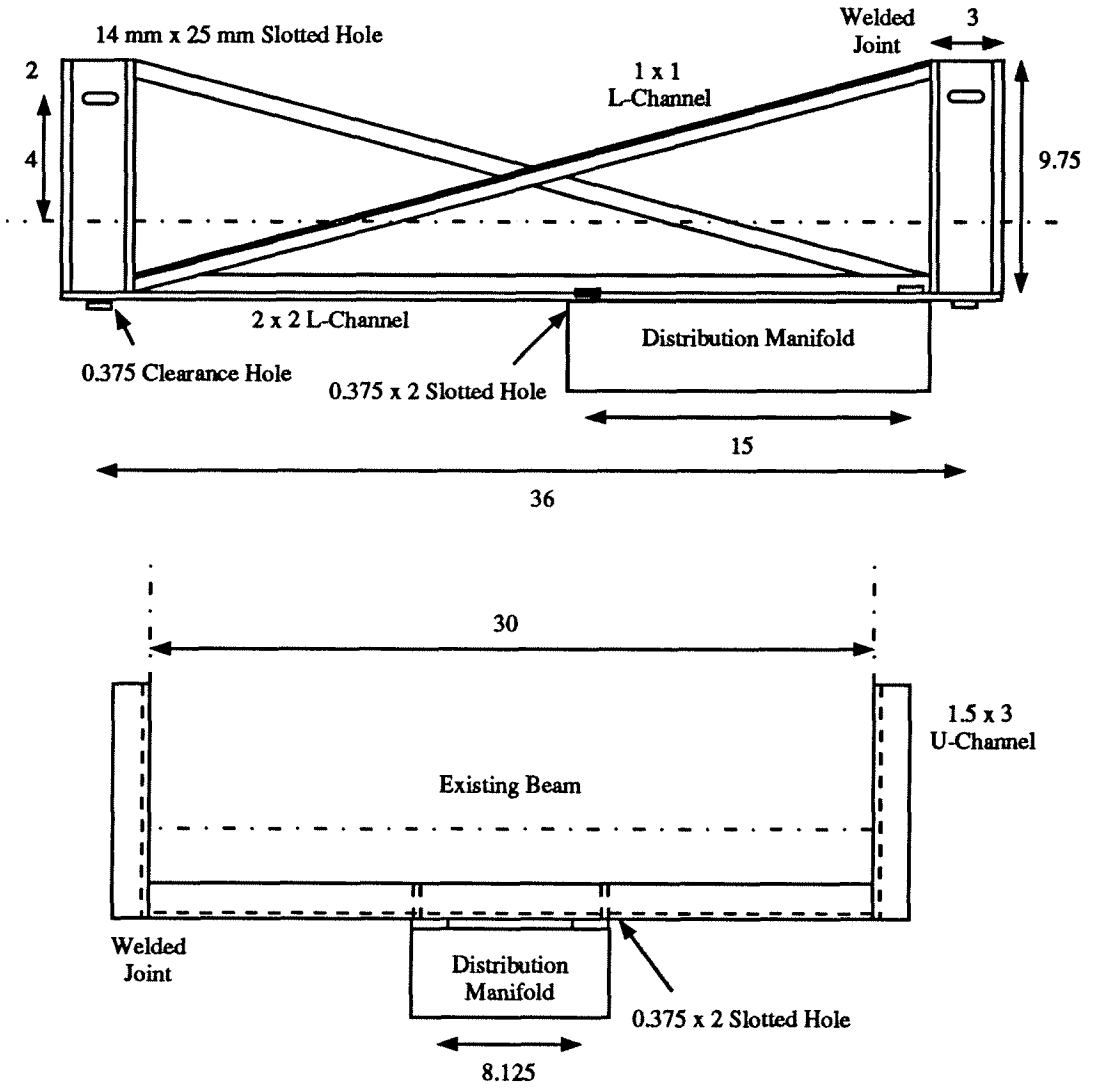


Figure 3.38: Distribution Manifold Frame.

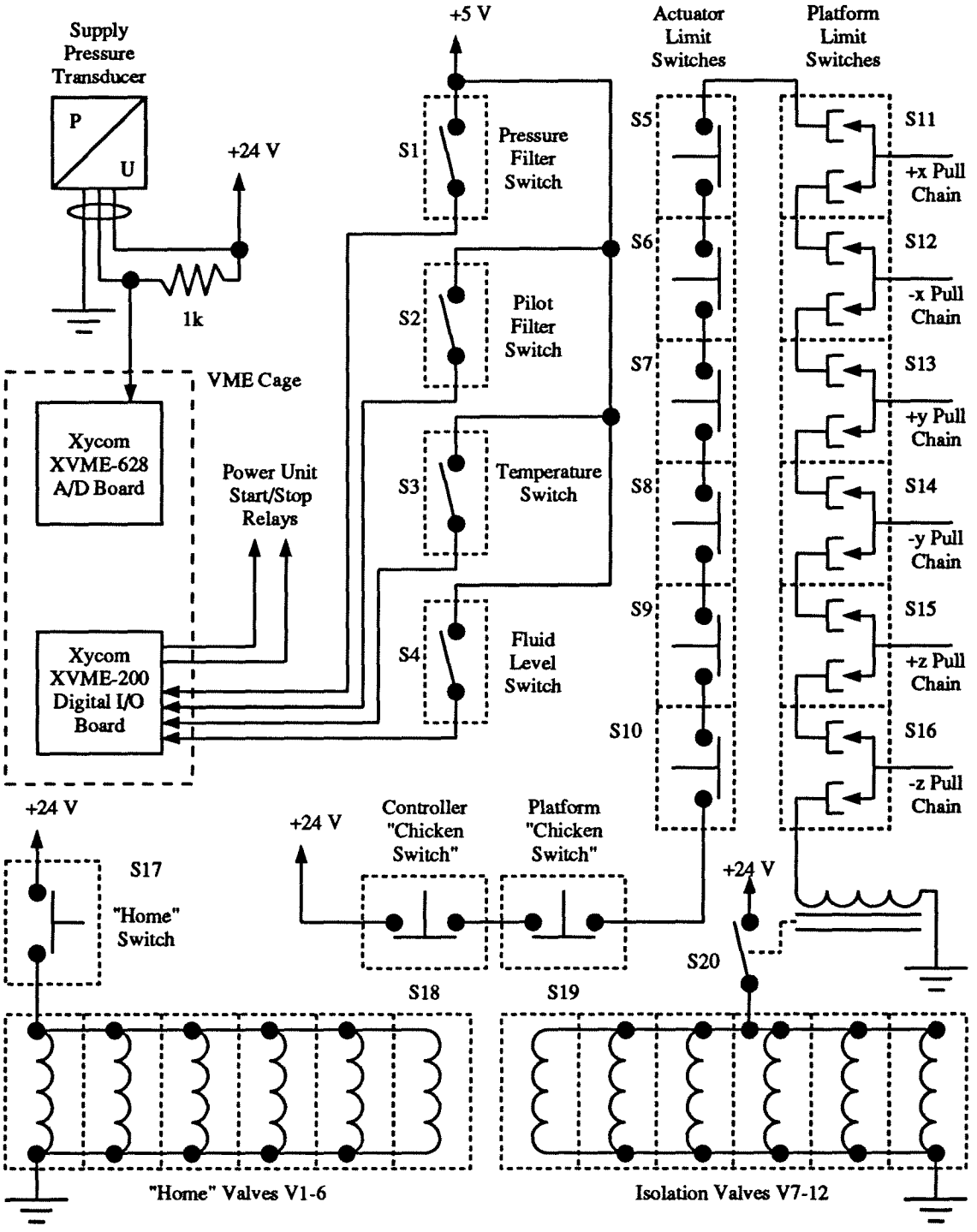


Figure 3.39: Electrical Schematic—Safety System.

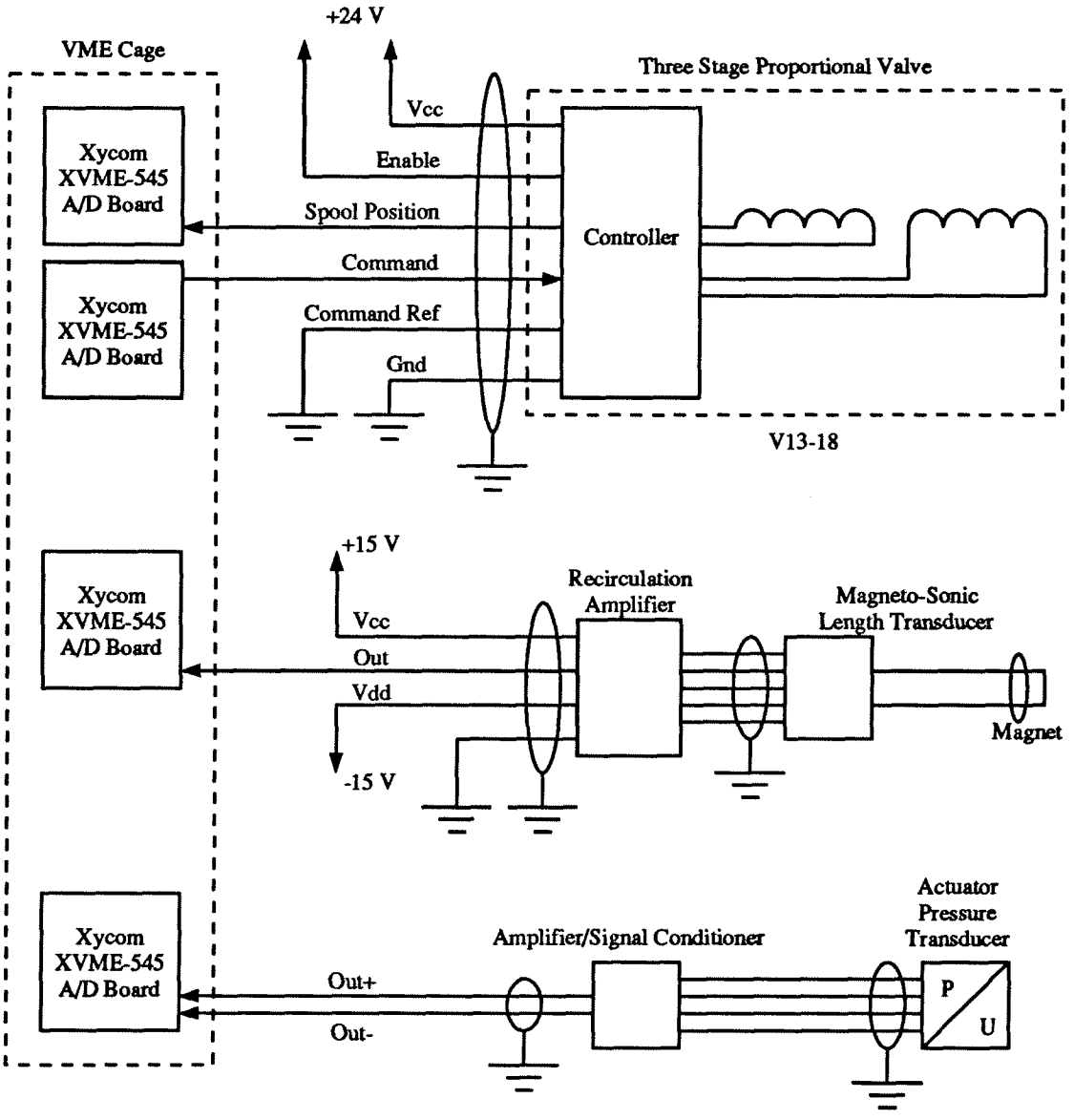


Figure 3.40: Electrical Schematic—Control System.

Chapter 4

Dynamics and Control

Chapter Four gives an overview of the dynamics and the control of the simulator. We provide a simplified model of the platform's dynamics as well as the dynamics of the hydraulic actuators as they play an important role in the simulator's performance. We then combine these to simulate the dynamics of the motion simulator. In the final section, we discuss the control of the simulator including both joint-space and cartesian-space strategies.

4.1 Platform Dynamics

Before we can test and evaluate methods of controlling the platform's position and orientation, velocity and acceleration, we must adequately model its dynamics. Using the principle of virtual work, we can relate the actuator forces to the force and torque acting on the platform. The virtual work, δw, done in translating the platform by a virtual distance, $[\delta x, \delta y, \delta z]^T$, and rotating the platform by the virtual angle $[\delta\psi, \delta\theta, \delta\phi]^T$ is

$$\delta w = \mathbf{f}^T \delta \mathbf{l} - \begin{bmatrix} {}^b\mathbf{f}_p \\ {}^b\tau_p \end{bmatrix}^T \begin{bmatrix} \delta \mathbf{d} \\ \delta\psi \\ \delta\theta \\ \delta\phi \end{bmatrix} \tag{4.68}$$

where $\mathbf{f}$ is a six-vector of the actuator forces, $\delta\mathbf{l}$ is a six-vector of virtual actuator displacements, ${}^b\mathbf{f}_p$ is the total force acting on the center of mass of the platform, ${}^b\tau_p$ is the total torque acting about the center of mass of the platform, $\delta\mathbf{d}$ is the virtual translational displacement of the platform's center of mass and $[\delta\psi, \delta\theta, \delta\phi]^T$ is the virtual rotational displacement of the platform

about its center of mass. We can substitute $\mathbf{J}\,[\delta\mathbf{d}, \delta\psi, \delta\theta, \delta\phi]^T$ for $\delta\mathbf{l}$ in (4.68), giving

$$\delta w = \left(\mathbf{J}^T\mathbf{f} - \begin{bmatrix} {}^b\mathbf{f}_p \\ {}^b\tau_p \end{bmatrix} \right)^T \delta\mathbf{x} \tag{4.69}$$

The principle of virtual work says that the work done by external forces $(\mathbf{f}; {}^b\mathbf{f}_p, {}^b\tau_p)$ corresponding to any virtual displacements $(\delta\mathbf{l}; \delta\mathbf{d}, \delta\psi, \delta\theta, \delta\phi)$ is zero $(\delta w = 0)$ [30]. Therefore, we can write

$$\begin{bmatrix} {}^b\mathbf{f}_p \\ {}^b\tau_p \end{bmatrix} = \mathbf{J}^T\mathbf{f} \tag{4.70}$$

If $\mathbf{J}$ is non-singular then we calculate the actuator forces given the force and torque on the platform using

$$\mathbf{f} = \left(\mathbf{J}^T\right)^{-1} \begin{bmatrix} {}^b\mathbf{f}_p \\ {}^b\tau_p \end{bmatrix}. \tag{4.71}$$

If we neglect the actuator dynamics, then we can easily derive the force and torque on the platform using Newton's second law, $\mathbf{F} = m\mathbf{a}$. Summing the forces on the platform's center of mass gives

$$\sum_{i=1}^{6} {}^b\mathbf{f}_i - m_p g\mathbf{k} = m_p {}^b\mathbf{a}_p \tag{4.72}$$

where ${}^b\mathbf{f}_i$ is the force of actuator i in the base frame, m_p is the mass of the platform, $g \approx 9.8$ m/s^2 is the acceleration due to gravity and $\mathbf{k}$ is the base frame's unit vector in the z direction. The force delivered by the actuators is

$${}^b\mathbf{f}_p = \begin{bmatrix} m_p {}^b\mathbf{a}_{p_x} \\ m_p {}^b\mathbf{a}_{p_y} \\ m_p \left({}^b\mathbf{a}_{p_z} + g \right) \end{bmatrix}. \tag{4.73}$$

The angular momentum of the platform with respect to the base frame is, by definition,

$${}^b\mathbf{L}_p = {}^b\mathbf{I}_p {}^b\omega_p = {}^b\mathbf{R}_p {}^p\mathbf{I}_p {}^b\mathbf{R}_p{}^T {}^b\omega_p \tag{4.74}$$

where ${}^{p}\mathbf{I}_p$ is the platform's inertia matrix expressed with respect to the platform frame. If we compute the moments of inertia along the principle axes x_p, y_p, z_p then ${}^{p}\mathbf{I}_p$ takes the form

$$ {}^{p}\mathbf{I}_p = \begin{bmatrix} {}^{p}\mathbf{I}_{p_{xx}} & 0 & 0 \\ 0 & {}^{p}\mathbf{I}_{p_{yy}} & 0 \\ 0 & 0 & {}^{p}\mathbf{I}_{p_{zz}} \end{bmatrix}. \tag{4.75} $$

Taking the derivative of (4.74) gives the torque on the platform as

$$ {}^{b}\tau_p = {}^{b}\mathbf{I}_p {}^{b}\alpha_p + {}^{b}\omega_p \times \left({}^{b}\mathbf{I}_p {}^{b}\omega_p \right) \tag{4.76} $$

where we assume that the inertia of the platform does not change with time in the platform frame. Rewriting (4.76) in terms of the base frame coordinates and combining with (4.73) gives

$$ \begin{bmatrix} {}^{b}\mathbf{f}_p \\ {}^{b}\tau_p \end{bmatrix} = \begin{bmatrix} m_p {}^{b}\mathbf{a}_p + m_p \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T \\ {}^{b}\mathbf{I}_p {}^{b}\alpha_p + {}^{b}\omega_p \times \left({}^{b}\mathbf{I}_p {}^{b}\omega_p \right) \end{bmatrix}. \tag{4.77} $$

Using (4.71), we can write a expression for the actuator forces given the platform's acceleration as

$$ \mathbf{f} = \left(\mathbf{J}^T \right)^{-1} \left(\mathbf{D} \begin{bmatrix} {}^{b}\mathbf{a}_p \\ {}^{b}\alpha_p \end{bmatrix} + \mathbf{E} \right) \tag{4.78} $$

where

$$ \mathbf{D} = \begin{bmatrix} m_p & 0 & 0 & \\ 0 & m_p & 0 & 0 \\ 0 & 0 & m_p & \\ & 0 & & {}^{b}\mathbf{I}_p \end{bmatrix} \tag{4.79} $$

and

$$ \mathbf{E} = \begin{bmatrix} 0 \\ 0 \\ m_p g \\ {}^{b}\omega_p \times \left({}^{b}\mathbf{I}_p {}^{b}\omega_p \right) \end{bmatrix}. \tag{4.80} $$

(See [25] for C source code)

4.2 Hydraulic Dynamics

To model the overall dynamics of the platform and devise a satisfactory control scheme, we must include the dynamics of the actuator hydraulics as they play a large part in determining the platform's performance including its velocity, acceleration and bandwidth. Unfortunately, hydraulic systems are difficult to model accurately for several reasons. First, fluid systems are inherently non-linear so we cannot analyze them over their entire operating range with standard linear techniques, although they can be linearized around a given operating point. Second, some of the system parameters, such as the damping and compliance of the load, are hard to measure for many actuators. Third, the effects of viscous and coulomb friction between the cylinder and the piston are highly non-linear and are therefore difficult to estimate. Fourth, any air trapped in the hydraulic fluid can greatly reduce its bulk modulus, thereby decreasing the actuators stiffness and performance. Finally, we assume in hydraulic analysis that we have turbulent flow (versus laminar flow) at any orifice in the system. However, this assumption is only valid for sharp edged orifices with adequately large flows. Over time, these orifice edges become rounded and some laminar flow will occur which is very difficult to model.

In spite of all these difficulties we can still model the hydraulic actuators for somewhat ideal circumstances. To model the actuators we use a standard mathematical approach [27]. Figure 4.41 shows a typical three-way valve/cylinder connection. There are three equations which describe the valve, cylinder and load dynamics respectively. First, for the valve, assuming that there is no leakage, the flow Q_c from the valve to the blind end of the cylinder is

$$Q_c = Q_1 - Q_2 \tag{4.81}$$

where Q_1 is the flow from the pressure line to the cylinder and Q_2 is the flow from the cylinder to the tank. Using the orifice equation [27] we can write expressions for Q_1 and Q_2 as

$$Q_1 = \begin{cases} C_d w (U + x)\sqrt{\frac{2}{\rho}(P_s - P_c)}, & x > -U \\ 0, & x < -U \end{cases} \tag{4.82}$$

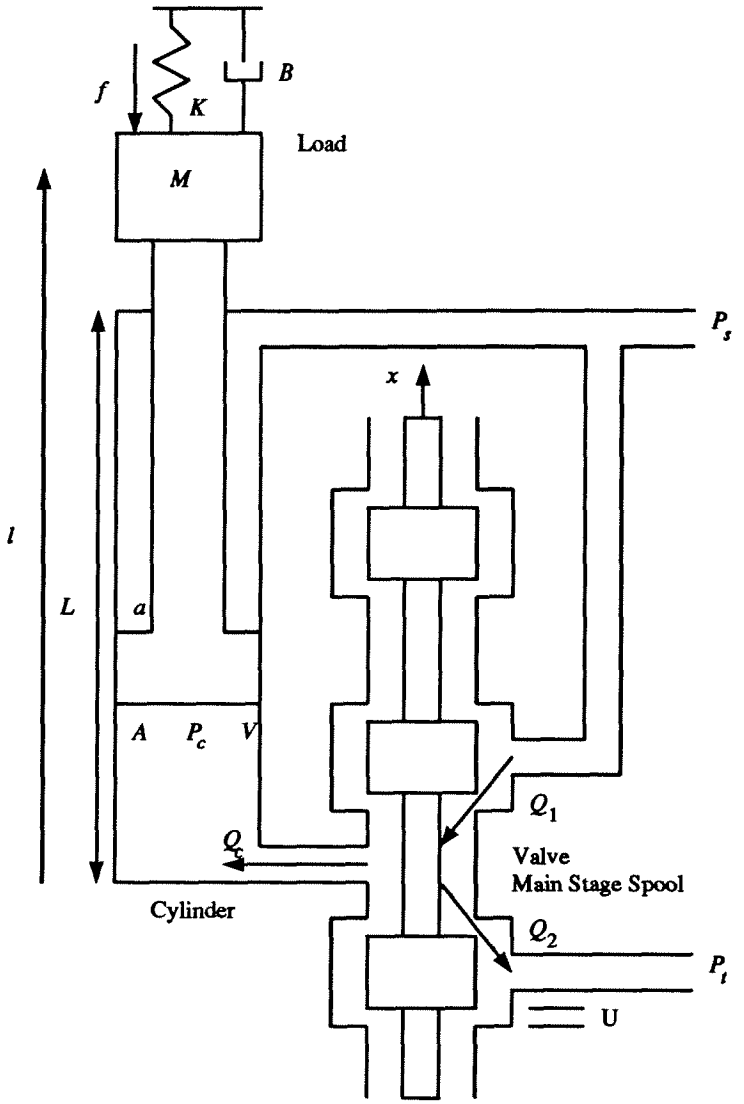


Figure 4.41: Definition of Three-way Connection Parameters.

and

$$Q_2 = \begin{cases} C_d w (U - x)\sqrt{\frac{2}{\rho}(P_c - P_t)}, & x < U \\ 0, & x > U \end{cases} \tag{4.83}$$

where $C_d = \frac{\pi}{\pi+2} \approx 0.611$ is the discharge coefficient, w is the port width of the valve, U is the valve's underlap, x is the spool position of the valve relative to center, ρ is the density of the

fluid, P_s is the supply pressure, P_c is the controlled pressure at the blind end of the cylinder and $P_t \approx 0$ is the tank (or return line) pressure. Second, for the cylinder, the flow Q_c, is the result of two processes: movement of the piston (change in volume) and compression of the fluid. These two processes are described by the continuity equation [27] as

$$Q_c = \dot{V} + \frac{V}{\beta}\dot{P}_c \tag{4.84}$$

where V is the total volume contained between the piston and the blind end of the cylinder and β is the bulk modulus of the fluid. Finally, applying Newton's second law to the cylinder and the load gives

$$P_c A - P_s a = M\ddot{l} + B\dot{l} + Kl + f \tag{4.85}$$

where A is the area of the piston, a is the annulus area between the piston rod and the cylinder wall, M is the mass of the load, B is the load's viscous damping coefficient, K is the spring constant of the load and f is an external force. Equations (4.81), (4.82), (4.83), (4.84) and (4.85) describe the dynamics of the valve, the cylinder and the load.

For the purposes of numerical integration, (4.81), (4.82), (4.83), (4.84) and (4.85) are solved for $\dddot{l}$ giving the following

$$\dddot{l} = \frac{1}{M}\left[\frac{\beta A}{V}\left(C_d w\sqrt{\frac{2}{\rho}}h(x, P_c) - \dot{V}\right) - B\ddot{l} - K\dot{l} - \dot{f}\right] \tag{4.86}$$

where

$$V \approx A(l - L), \tag{4.87}$$

$$\dot{V} = A\dot{l}, \tag{4.88}$$

$$P_c = \frac{M\ddot{l} + B\dot{l} + Kl + f + P_s a}{A}, \tag{4.89}$$

$$h(x, P_c) = \begin{cases} (U + x)\sqrt{P_s - P_c}, & x > U \\ (U + x)\sqrt{P_s - P_c} - (U - x)\sqrt{P_c - P_t}, & -U < x < U \\ -(U - x)\sqrt{P_c - P_t}, & x < -U \end{cases} \tag{4.90}$$

and L is the stroke length of the cylinder.

4.3 Combined Dynamics

We have looked at the actuator and platform dynamics separately. Now we will combine them in order to derive and simulate the dynamics of the motion simulator as a whole. To simplify this derivation we will not include the dynamic effects corresponding to the mass of the actuators themselves. This simplification is valid if the platform is larger than the actuators and the mass of each actuator is concentrated near the fixed end so that the actuator's moment of inertia is relatively small. We meet these two criteria with the design of our motion simulator because the mass of the platform is more than six times the mass of an actuator and the hydraulic valves and manifolds are mounted at the fixed ends of the actuators.

The hydraulic dynamics of each of the platform's six actuators is described by (4.86). Now we can combine them into a single vector equation to give

$$m_a \ddot{\mathbf{l}} = \begin{bmatrix} \frac{\beta A}{V_1}\left(C_d w \sqrt{\frac{2}{\rho}} h\left(x_1, P_{c1}\right) - \dot{V}_1\right) \\ \vdots \\ \frac{\beta A}{V_6}\left(C_d w \sqrt{\frac{2}{\rho}} h\left(x_6, P_{c6}\right) - \dot{V}_6\right) \end{bmatrix} - B\ddot{\mathbf{l}} - K\dot{\mathbf{l}} - \dot{\mathbf{f}} \tag{4.91}$$

where $M = m_a$ and m_a is the combined mass of the piston and piston rod of each actuator, $\mathbf{l}$ is a six-vector of the actuator lengths and $\mathbf{f}$ is a six-vector of the actuator forces. Recall that the actuator forces can be expressed in terms of the platform's acceleration as

$$\mathbf{f} = \left(\mathbf{J}^T\right)^{-1}\left(\mathbf{D}\begin{bmatrix} {}^b\mathbf{a}_p \\ {}^b\alpha_p \end{bmatrix} + \mathbf{E}\right). \tag{4.92}$$

Taking the derivative of (4.92) gives

$$\dot{\mathbf{f}} = -\left(\mathbf{J}^T\right)^{-1}\dot{\mathbf{J}}^T\left(\mathbf{J}^T\right)^{-1}\left(\mathbf{D}\begin{bmatrix} {}^b\mathbf{a}_p \\ {}^b\alpha_p \end{bmatrix} + \mathbf{E}\right) + \left(\mathbf{J}^T\right)^{-1}\left(\dot{\mathbf{D}}\begin{bmatrix} {}^b\mathbf{a}_p \\ {}^b\alpha_p \end{bmatrix} + \mathbf{D}\begin{bmatrix} {}^b\dot{\mathbf{a}}_p \\ {}^b\dot{\alpha}_p \end{bmatrix} + \dot{\mathbf{E}}\right) \tag{4.93}$$

where

$$\dot{\mathbf{D}} = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}\left({}^b\omega_p\right) \end{bmatrix}\mathbf{D} + \mathbf{D}\begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}^T\left({}^b\omega_p\right) \end{bmatrix} \tag{4.94}$$

and

$$\dot{\mathbf{E}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ {}^b\alpha_p \times \left({}^b\mathbf{I}_p {}^b\omega_p\right) + {}^b\omega_p \times \left({}^b\omega_p \times \left({}^b\mathbf{I}_p {}^b\omega_p\right) + {}^b\mathbf{I}_p {}^b\alpha_p\right) \end{bmatrix}. \tag{4.95}$$

From the (2.27) and (2.29) we know that the platform's velocity and acceleration can be expressed in terms of the actuator velocities and accelerations. Equation (4.93), however, contains terms in ${}^b\dot{\mathbf{a}}_p$ and ${}^b\dot{\alpha}_p$ so we must take the derivative of (2.29) giving, after some simplification,

$$\begin{bmatrix} {}^b\dot{\mathbf{a}}_p \\ {}^b\dot{\alpha}_p \end{bmatrix} = \mathbf{J}^{-1}\left(\dddot{\mathbf{l}} - \ddot{\mathbf{J}}\begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} - 2\dot{\mathbf{J}}\begin{bmatrix} {}^b\mathbf{a}_p \\ {}^b\alpha_p \end{bmatrix}\right). \tag{4.96}$$

Substituting (4.96) into (4.93) and the result into (4.91) gives

$$\begin{aligned} \dddot{\mathbf{l}} \;=\; & \left(m_a\mathbf{I} + \left(\mathbf{J}^T\right)^{-1}\mathbf{D}\mathbf{J}^{-1}\right)^{-1}\left\{\begin{bmatrix} \frac{\beta A}{V_1}\left(C_d w\sqrt{\frac{2}{\rho}}h\left(x_1, P_{c1}\right) - \dot{V}_1\right) \\ \vdots \\ \frac{\beta A}{V_6}\left(C_d w\sqrt{\frac{2}{\rho}}h\left(x_6, P_{c6}\right) - \dot{V}_6\right) \end{bmatrix} - \right. \\ & B\ddot{\mathbf{l}} - K\dot{\mathbf{l}} + \left(\mathbf{J}^T\right)^{-1}\dot{\mathbf{J}}^T\left(\mathbf{J}^T\right)^{-1}\left(\mathbf{D}\begin{bmatrix} {}^b\mathbf{a}_p \\ {}^b\alpha_p \end{bmatrix} + \mathbf{E}\right) - \\ & \left.\left(\mathbf{J}^T\right)^{-1}\left(\dot{\mathbf{D}}\begin{bmatrix} {}^b\mathbf{a}_p \\ {}^b\alpha_p \end{bmatrix} - \mathbf{D}\mathbf{J}^{-1}\left(\ddot{\mathbf{J}}\begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} + 2\dot{\mathbf{J}}\begin{bmatrix} {}^b\mathbf{a}_p \\ {}^b\alpha_p \end{bmatrix}\right) + \dot{\mathbf{E}}\right)\right\} \end{aligned} \tag{4.97}$$

where $\mathbf{I}$ is a six by six identity matrix. Note that all the terms on the right side of (4.97) can be expressed in terms of the actuator lengths, velocities and accelerations so it expresses the combined dynamics of the platform and the actuator hydraulics as a set of six coupled, third order differential equations where the inputs are the six valve spool positions $x_1 \ldots x_6$.

4.4 Control

For the motion simulator to follow trajectories and respond to operator commands, it needs a feedback control system. This system takes a desired platform position and orientation

and compares it to the current platform position and orientation and drives the hydraulic actuators. Typically, there are two common control strategies which achieve this goal: joint-space and cartesian-space (see Figures 4.42 and 4.43). Using the joint-space strategy, each actuator is driven independently, making it computationally very fast because the feedback occurs at the joint variable (actuator length) level. This strategy does, however, ignore the coupling between the actuators and the platform. On the other hand, the cartesian-space strategy includes coupling between the actuators, however, it requires more computational time because the feedback (platform position and orientation) must be computed from the actuator lengths (forward kinematics). In the block diagrams (Figures 4.42 and 4.43), we see that both control strategies require compensators, one for each actuator, which transform the difference between the actual and desired actuator lengths, $\Delta \mathbf{l}$, to the six valve spool position inputs, $\mathbf{x}$. Determining the type of controller and it's performance parameters is easily accomplished

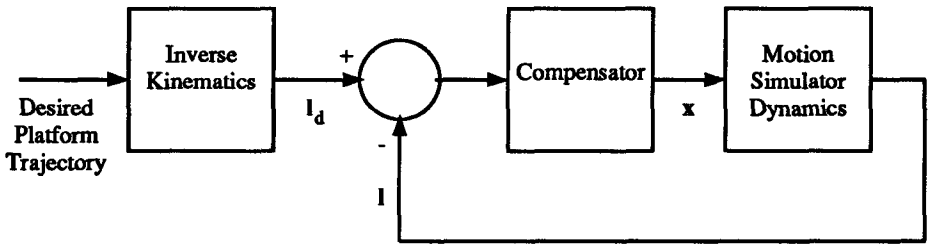


Figure 4.42: Joint-Space Control Block Diagram.

using classic control theory which is based on linear, time-invariant (LTI) systems. We can see from (4.90) and (4.97) that the platform's hydraulics and dynamics form a complex, non-linear system. Therefore, in order to use classic control theory we must linearize the system. We do this by performing a Taylor series expansion of (4.81) about an operating point as described below.

From (4.81), (4.82) and (4.83) we see that Q_c is a non-linear function of the valve spool position x and the pressure at the cylinder's blind end P_c. As mentioned earlier, we linearize

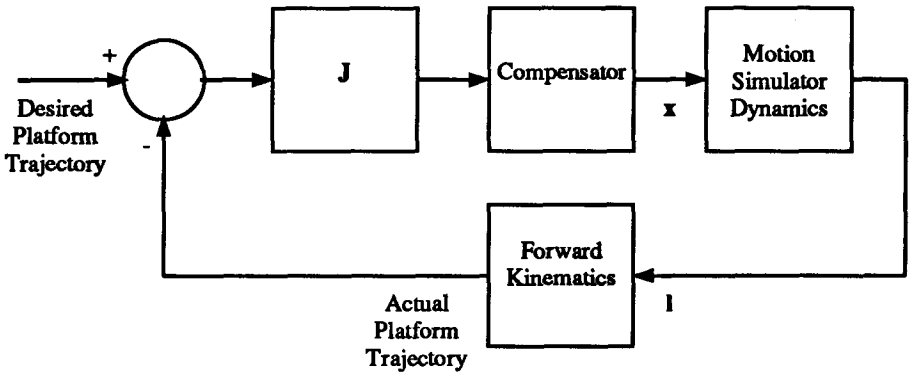


Figure 4.43: Cartesian-Space Control Block Diagram.

Q_c about a particular operating point. Q_c is a differentiable function of x and P_c:

$$Q_c = Q_c(x, P_c). \tag{4.98}$$

Thus, we perform a Taylor series expansion about the operating point $Q_c = Q_{c0}$ giving

$$Q_c = Q_{c0} + \left.\frac{\partial Q_c}{\partial x}\right|_{Q_{c0}} \Delta x + \left.\frac{\partial Q_c}{\partial P_c}\right|_{Q_{c0}} \Delta P_c + \cdots \tag{4.99}$$

Note that at $x = \pm U$ (4.98) is not differentiable so (4.99) does not exist. If we confine ourselves to a small region around the operating point then we can neglect higher order terms so

$$\Delta Q_c = K_q \Delta x - K_c \Delta P_c \tag{4.100}$$

where $K_q = \frac{\partial Q_c}{\partial x}$ is the flow gain, $K_c = -\frac{\partial Q_c}{\partial P_c}$ is the flow-pressure coefficient and $\Delta P_c = \left(M\Delta\ddot{l} + B\Delta\dot{l} + K\Delta l + \Delta f\right)/A$. Taking the derivative of (4.81) with respect to x and P_c we write expressions for K_q and K_c as

$$K_q = \begin{cases} C_d w\sqrt{\frac{2}{\rho}(P_s - P_c)}, & x > U \\ C_d w\sqrt{\frac{2}{\rho}}\left(\sqrt{P_s - P_c} + \sqrt{P_c - P_t}\right), & -U < x < U \\ C_d w\sqrt{\frac{2}{\rho}(P_c - P_t)}, & x < -U \end{cases} \tag{4.101}$$

and

$$K_c = \begin{cases} \frac{C_d w}{2}(U+x)\frac{\sqrt{\frac{2}{\rho}(P_s-P_c)}}{P_s-P_c}, & x > U \\ \frac{C_d w}{2}\sqrt{\frac{2}{\rho}}\left[(U+x)\frac{\sqrt{P_s-P_c}}{P_s-P_c} + (U-x)\frac{\sqrt{P_c-P_t}}{P_c-P_t}\right], & -U < x < U \\ \frac{C_d w}{2}(U-x)\frac{\sqrt{\frac{2}{\rho}(P_c-P_t)}}{P_c-P_t}, & x < -U \end{cases} \tag{4.102}$$

We combine (4.100), (4.84) and (4.85) as before with one additional modification. As part of the linearization we write $V = V_0 + A\Delta l$ where V_0 is the initial volume between the piston and the blind end of the cylinder and $A\Delta l$ is the change in volume such that $|A\Delta l| \ll V_0$. Now we can write the differential equation relating l and x as

$$\frac{K_q}{A}x - \frac{K_c}{A^2}f - \frac{V_0}{\beta A^2}\dot{f} = \frac{V_0 M}{\beta A^2}\dddot{l} + \left(\frac{V_0 B}{\beta A^2} + \frac{K_c M}{A^2}\right)\ddot{l} + \left(\frac{V_0 K}{\beta A^2} + \frac{K_c B}{A^2} + 1\right)\dot{l} + \frac{K_c K}{A^2}l \tag{4.103}$$

where Δx, ΔP_c and Δl are replaced by x, l and P_c for convenience in notation. By taking the Laplace transform of (4.103) we get the transfer function

$$L(s) = \frac{\frac{K_q}{A}X(s) - \left(\frac{K_c}{A^2} + \frac{V_0}{\beta A^2}s\right)F(s)}{\frac{1}{\omega_h^2}s^3 + \frac{2\xi_h}{\omega_h}s^2 + \left(\frac{V_0 K}{\beta A^2} + \frac{K_c B}{A^2} + 1\right)s + \frac{K_c K}{A^2}} \tag{4.104}$$

where $\omega_h = 2\pi f_h = A\sqrt{\frac{\beta}{V_0 M}}$ is the hydraulic undamped natural frequency and $\xi_h = \frac{B}{2A}\sqrt{\frac{V_0}{\beta M}} + \frac{K_c}{2A}\sqrt{\frac{\beta M}{V_0}}$ is the hydraulic damping ratio. Because the frequency and damping depend on the volume of fluid contained in the actuator and hence on the length of the actuator, it is unlikely that a fixed gain controller will provide acceptable performance. To compensate for changes in the frequency and damping in the actuators, we adjust the gains of the controller while the platform is moving.

Equation (4.104) is made up of two terms. The first is the transfer function from the valve spool position $X(s)$ to the actuator length $L(s)$. The second is the transfer function from the reaction force $F(s)$ of the platform on the actuator to the actuator length. This term is treated as a disturbance input in a simple compensator, however, it's effect on the length may be lessened by using a more advanced feedforward compensator design [31]. At this time, we are using a simple proportional gain, joint-space controller to simulate the motion simulator's

combined dynamics (Equation 4.97). The compensator's transfer function is

$$G_c(s) = \frac{X(s)}{E(s)} = K_p \tag{4.105}$$

where $E(s)$ is the Laplace transform of the error between the desired and the actual actuator lengths and K_p is the proportional gain. We choose a value for K_p which prevents the system from becoming unstable while meeting a desired steady-state error. If we assume the load on the actuator is a simple, rigid mass ($K = 0$), then the plant transfer function from the valve input to the actuator length is

$$G_p(s) = \frac{L(s)}{X(s)} = \frac{\frac{K_q}{A}}{\frac{1}{\omega_h^2}s^3 + \frac{2\xi_h}{\omega_h}s^2 + \left(\frac{K_c B}{A^2} + 1\right)s}. \tag{4.106}$$

Figures 4.44 and 4.45 show the frequency response of $G_p(s)$ for three different actuator lengths: 70, 100 and 130 inches. Adding a proportional gain of 0.02 to 0.03 will lower the magnitude curve and make the system stable. The closed-loop system characteristic equation is

$$1 + G_c(s)G_p(s) = 1 + \frac{\frac{K_p K_q}{A}}{\frac{1}{\omega_h^2}s^3 + \frac{2\xi_h}{\omega_h}s^2 + \left(\frac{K_c B}{A^2} + 1\right)s} = 0 \tag{4.107}$$

or

$$\frac{1}{\omega_h^2}s^3 + \frac{2\xi_h}{\omega_h}s^2 + \left(\frac{K_c B}{A^2} + 1\right)s + \frac{K_p K_q}{A} = 0. \tag{4.108}$$

By applying the Routh-Hurwitz stability criterion [31], we find that the proportional gain must satisfy

$$\frac{2\xi_h \omega_h A}{K_q}\left(\frac{K_c B}{A^2} + 1\right) > K_p > 0 \tag{4.109}$$

so that the closed-loop system remains stable. If the system is stable then the steady-state system error is found using the final value theorem of the Laplace transform, namely,

$$e_{ss} = \lim_{s \to 0} sE(s). \tag{4.110}$$

For the closed-loop system

$$e_{ss} = \lim_{s \to 0} \frac{sR(s)}{1 + G_c(s)G_p(s)}. \tag{4.111}$$

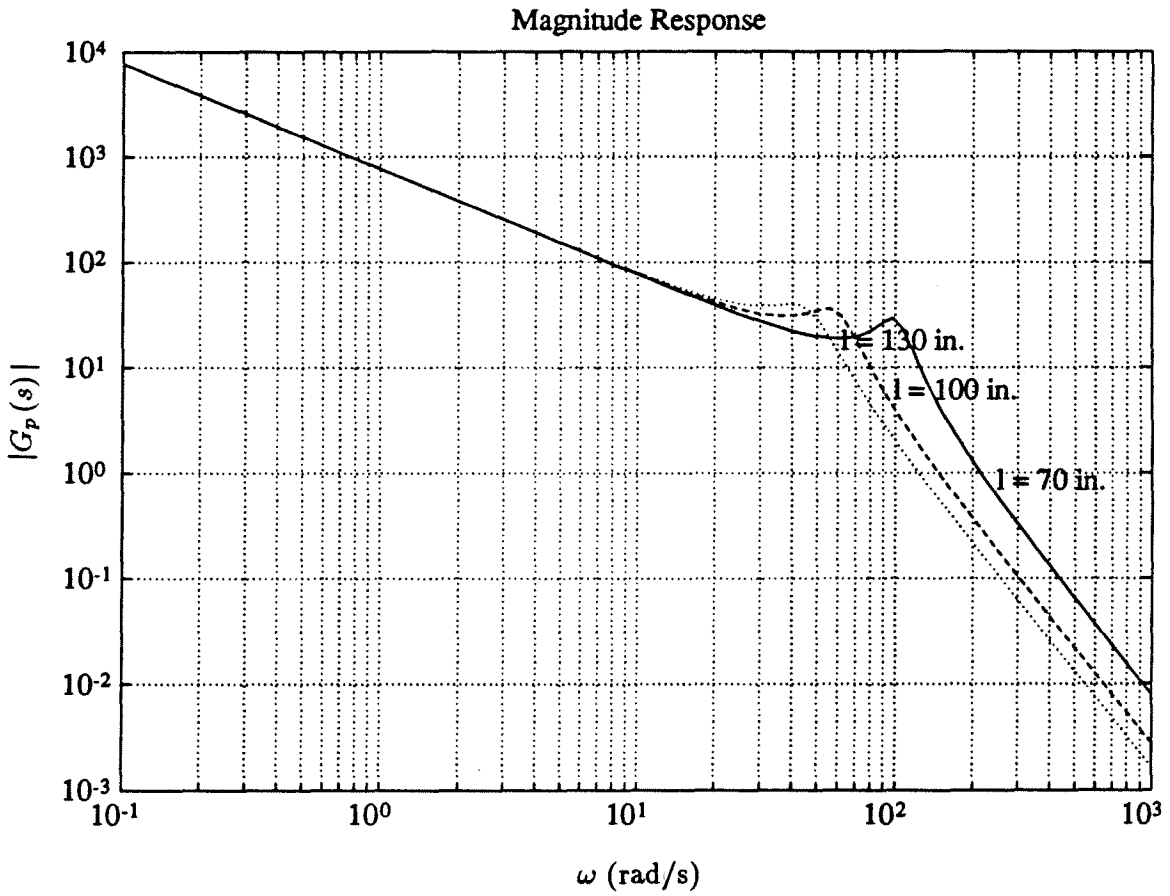


Figure 4.44: Open-loop Magnitude Response.

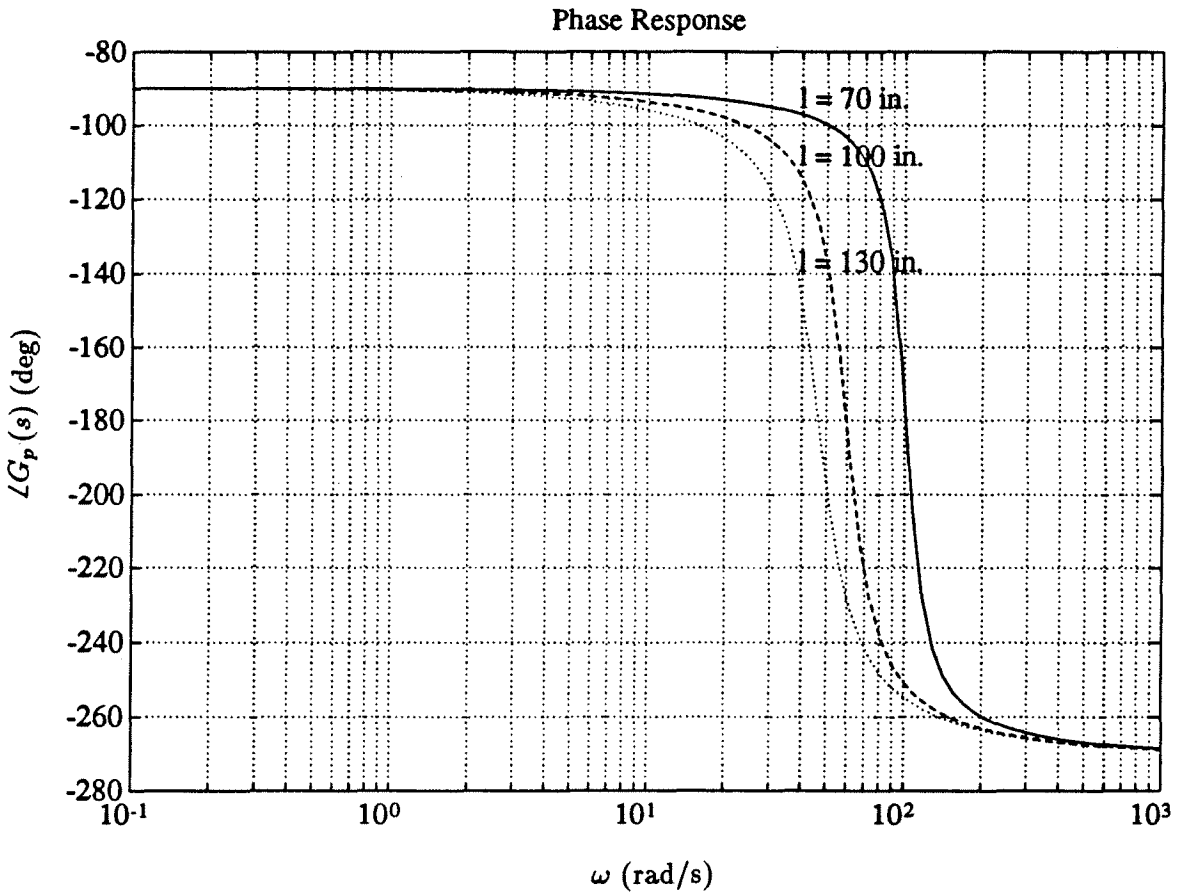


Figure 4.45: Open-loop Phase Response.

As the system is type 1, the steady-state error for a ramp input $R(s) = \frac{1}{s^2}$ is finite and non-zero. Substituting for $R(s)$, $G_c(s)$ and $G_p(s)$ into (4.111) and taking the limit as s goes to zero gives

$$e_{ss} = \frac{\frac{K_c B}{A^2} + 1}{\frac{K_p K_q}{A}}. \tag{4.112}$$

Solving (4.112) for K_p gives

$$K_p = \frac{\frac{K_c B}{A^2} + 1}{\frac{e_{ss} K_q}{A}}. \tag{4.113}$$

Equations (4.109) and (4.113) provide a method of choosing the compensator's proportional gain. Using this method, parameters in Table 4.7 and the joint-space control strategy, we ran some closed-looped simulations of the motion simulator's combined dynamics to learn more about the dynamic response. Figures 4.46 to 4.68 show the results of four of these simulations where in each simulation there are nineteen plots (six figures) showing the actuator and platform positions (lengths), velocities and accelerations as well as the actuator forces, pressures and valve spool positions. In the first two simulations (Figures 4.46 to 4.49 and Figures 4.52 to 4.55), we oscillated the platform along the x axis and about the z axis with an amplitude

Table 4.7: Control Simulation Parameters.

Parameter	Value	Definition
β	900 MPa	Bulk Modulus of the Fluid
A	$(1.5)^2\pi/4$ in^2	Piston Area
a	$A - (1.0)^2\pi/4$ in^2	Annulus Area
C_d	0.611	Discharge Coefficient
w	5.7 mm	Port Width of the Valve
ρ	858.2 kg/m^3	Density of the Fluid
L	54 in	Stroke Length of the Cylinder
m_p	250 kg	Mass of the Platform
B[1]	5000 N s/m	Viscous Damping of the Load
K	0 N s^2/m	Spring Constant of the Load
P_s	2000 PSI	Supply Pressure
P_t	0 PSI	Tank Pressure
U	55.4×10^{-6} m	Underlap of the Valve

[1]The viscous damping, was determined experimentally to give a typical hydraulic damping ratio ξ_h in the range 0.1 to 0.3 [27].

of 0.15 meters and 10° at 1 Hertz respectively. In addition, Figures 4.51 and 4.57 show the closed-loop magnitude response of the platform for oscillations in the x direction and about the z axis. From these figures we see that the platform has a 2.3 Hz bandwidth along x and a 2 Hz bandwidth about z which is higher than that of the Kawasaki motion base. In the last two simulations (Figures 4.58 to 4.61 and Figures 4.64 to 4.67) we applied a 0.1 meter step input to the platform's x_p coordinate and a 10° step input to the platform's ϕ_p coordinate. Figures 4.63 and 4.69 show a more detailed view of the motion simulator's step response and indicate that the platform has a rise time of $\approx$ 0.15 seconds which is 0.04 seconds less than the Kawasaki motion base. Also, the oscillation present in the x step response may indicate a need to increase the damping using pressure feedback [27]. The preliminary results of the simulations indicate that the motion simulator and hydraulics are capable of producing the desired performance as shown in Table 1.2. However, the "rough" response shown in the simulations indicate that a simple proportional gain compensator is not adequate for controlling the platform's actuators. This "rough" response may be due in part because the actuators are not completely synchronized. As each actuator responds independently, some of them may reach their desired position sooner than others, causing the platform to wander from the desired trajectory. Because these results are preliminary, further investigation of the control system design is required.

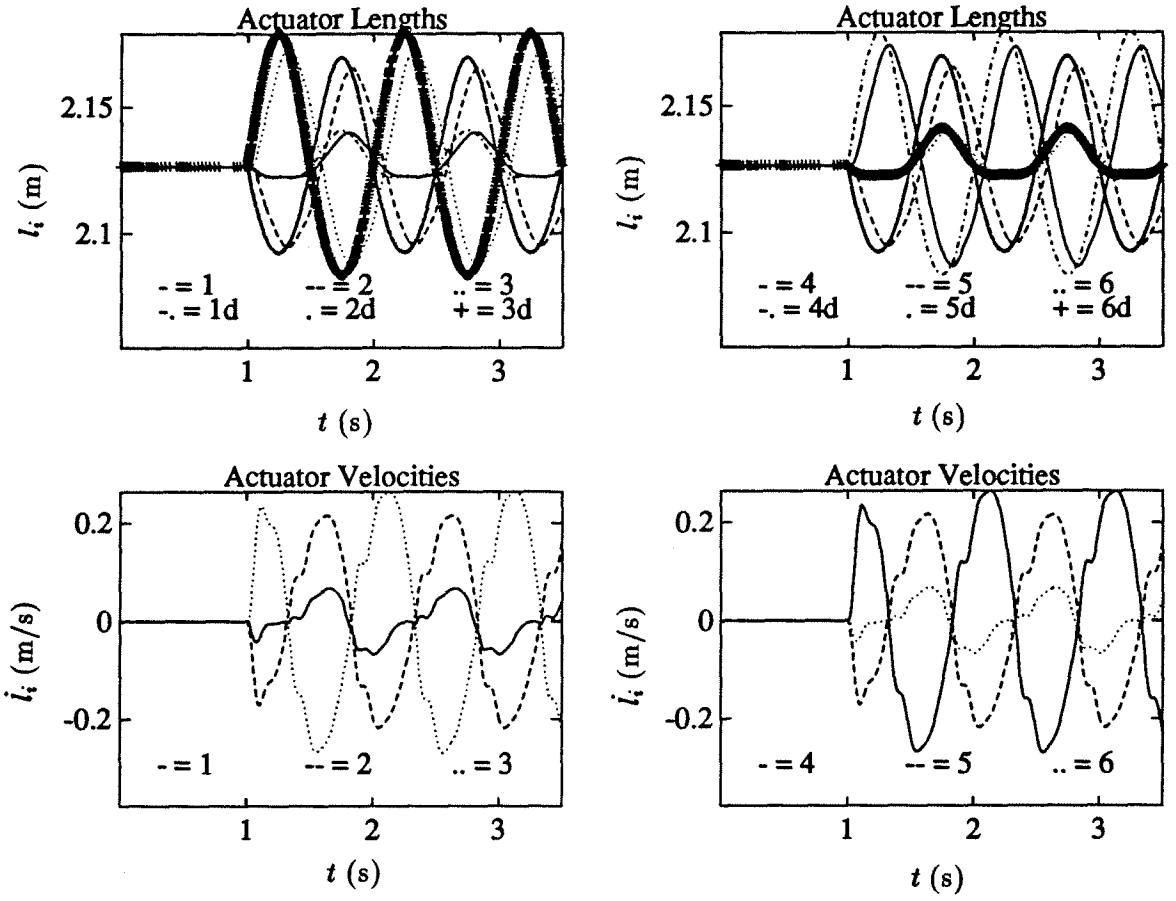


Figure 4.46: Proportional Control Simulation Results for Oscillation in the x direction.

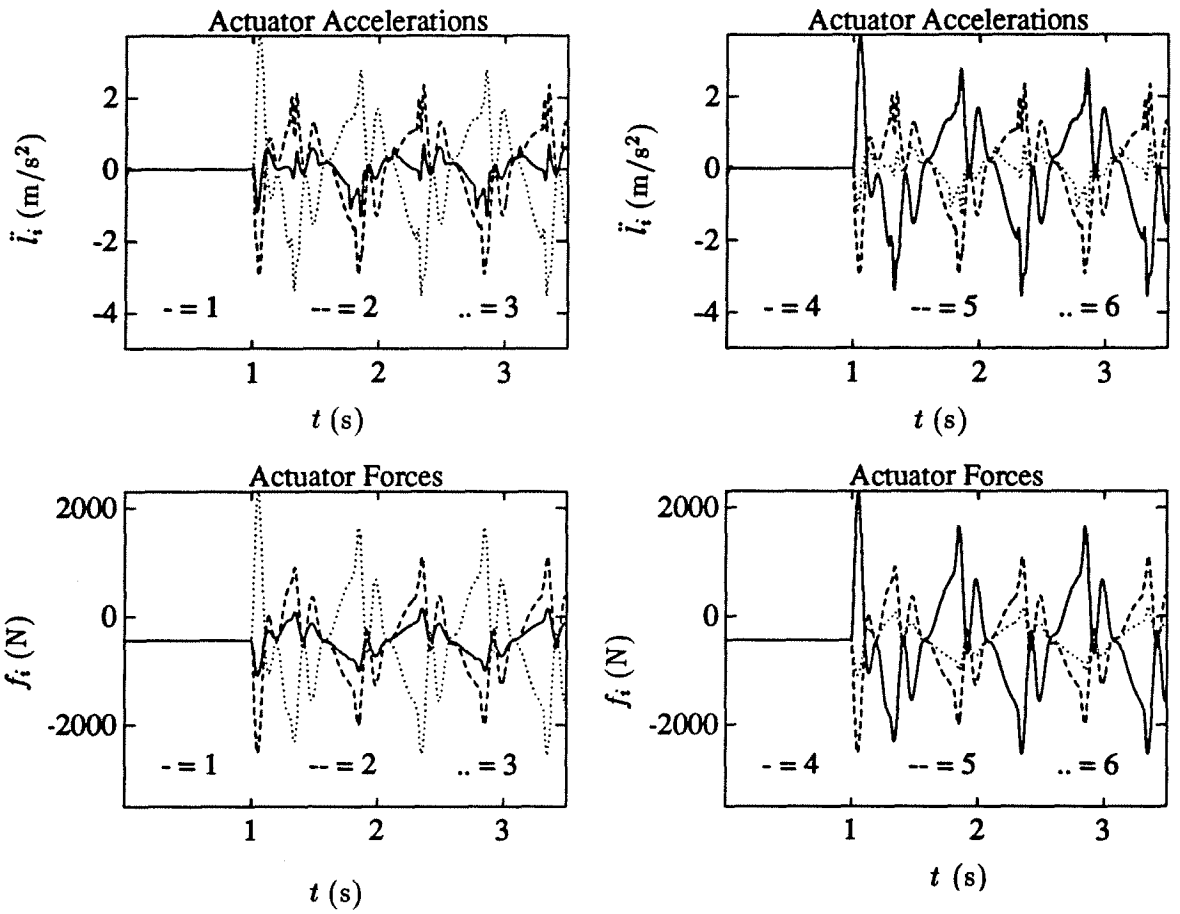


Figure 4.47: Proportional Control Simulation Results for Oscillation in the x direction.

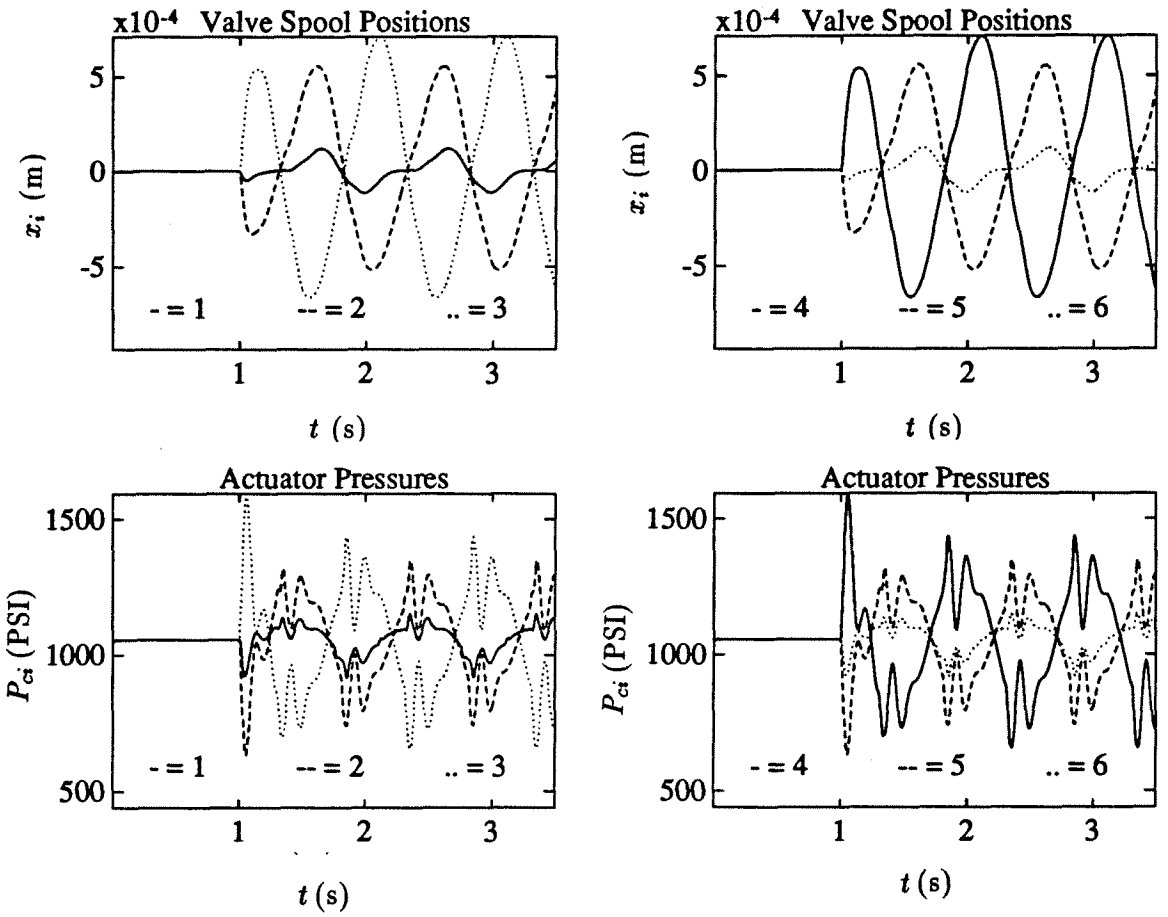


Figure 4.48: Proportional Control Simulation Results for Oscillation in the x direction.

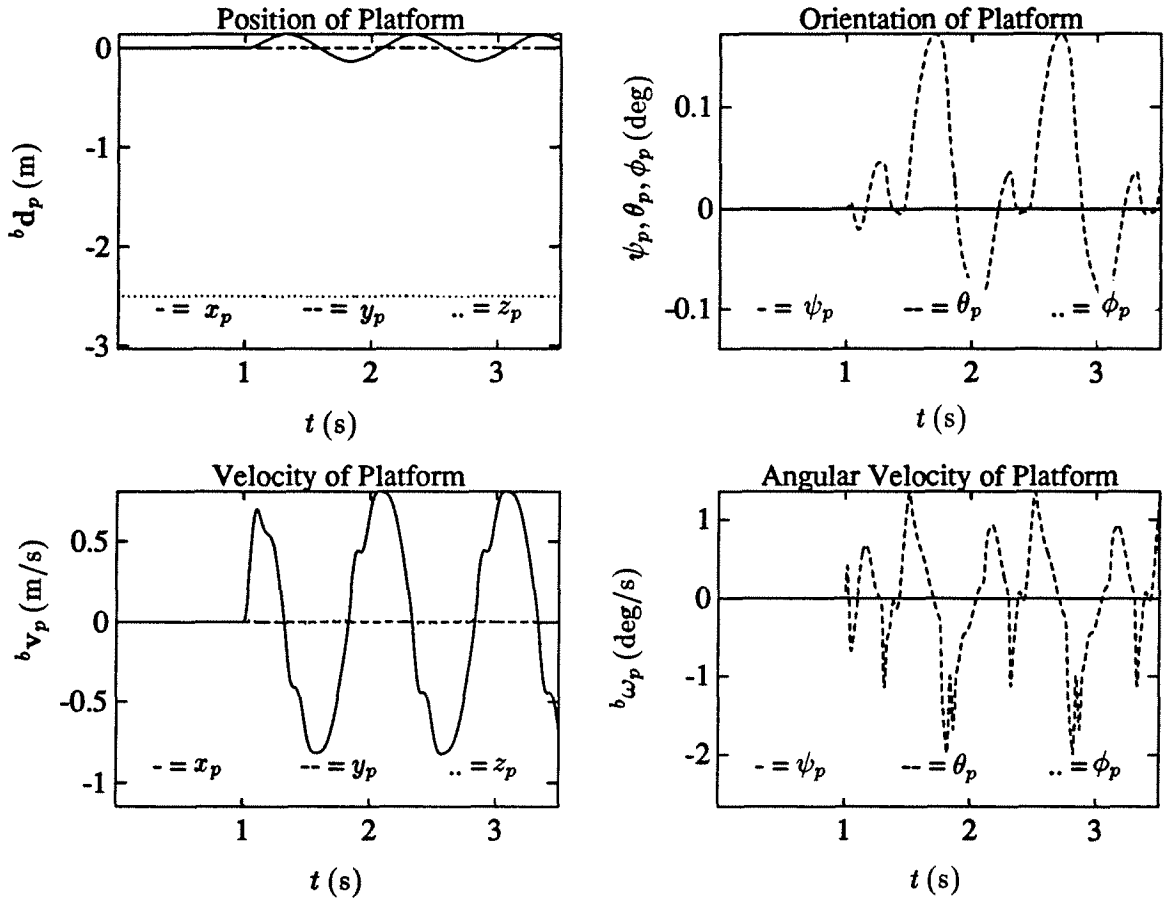


Figure 4.49: Proportional Control Simulation Results for Oscillation in the x direction.

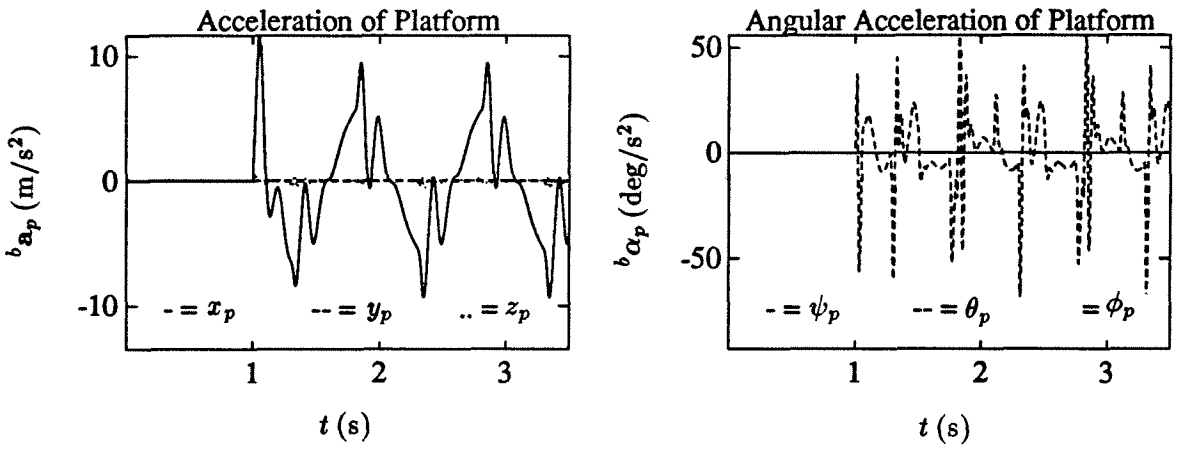


Figure 4.50: Proportional Control Simulation Results for Oscillation in the x direction.

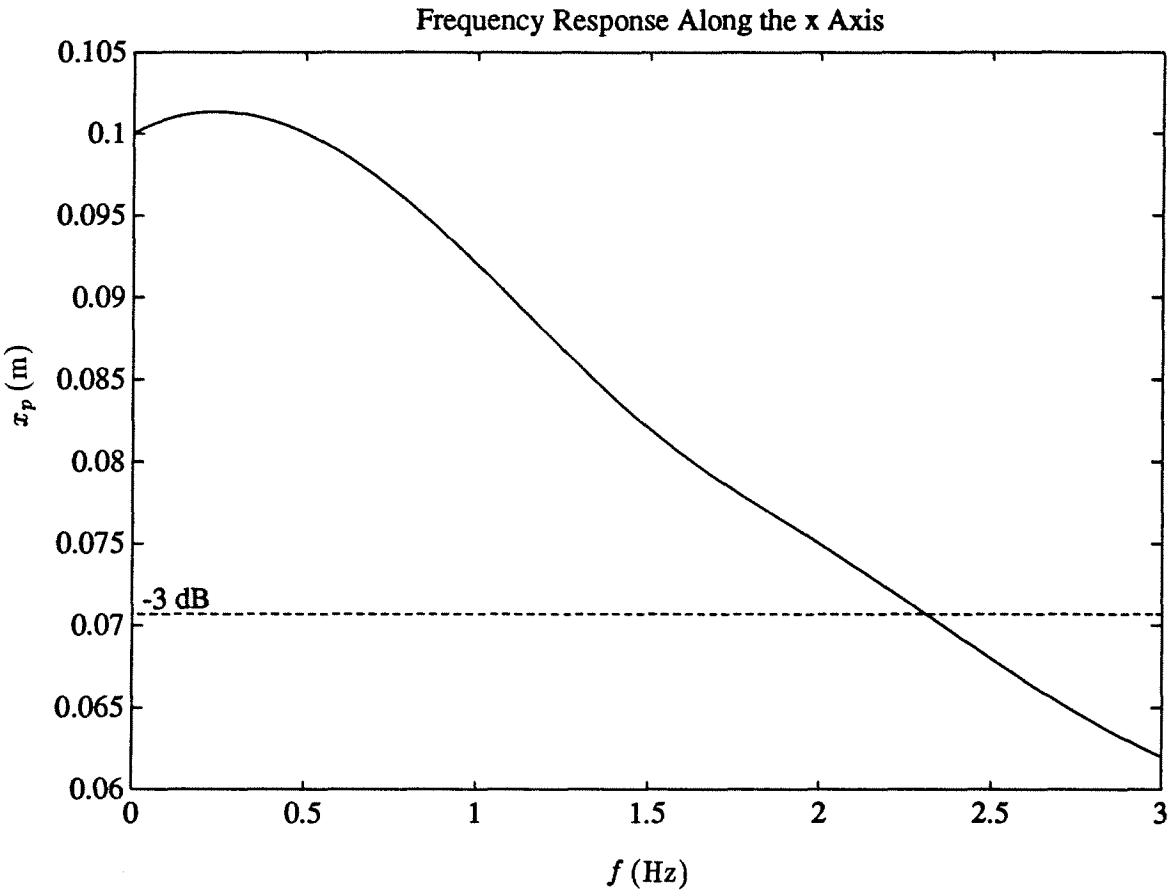


Figure 4.51: Frequency Response for Oscillation in the x direction.

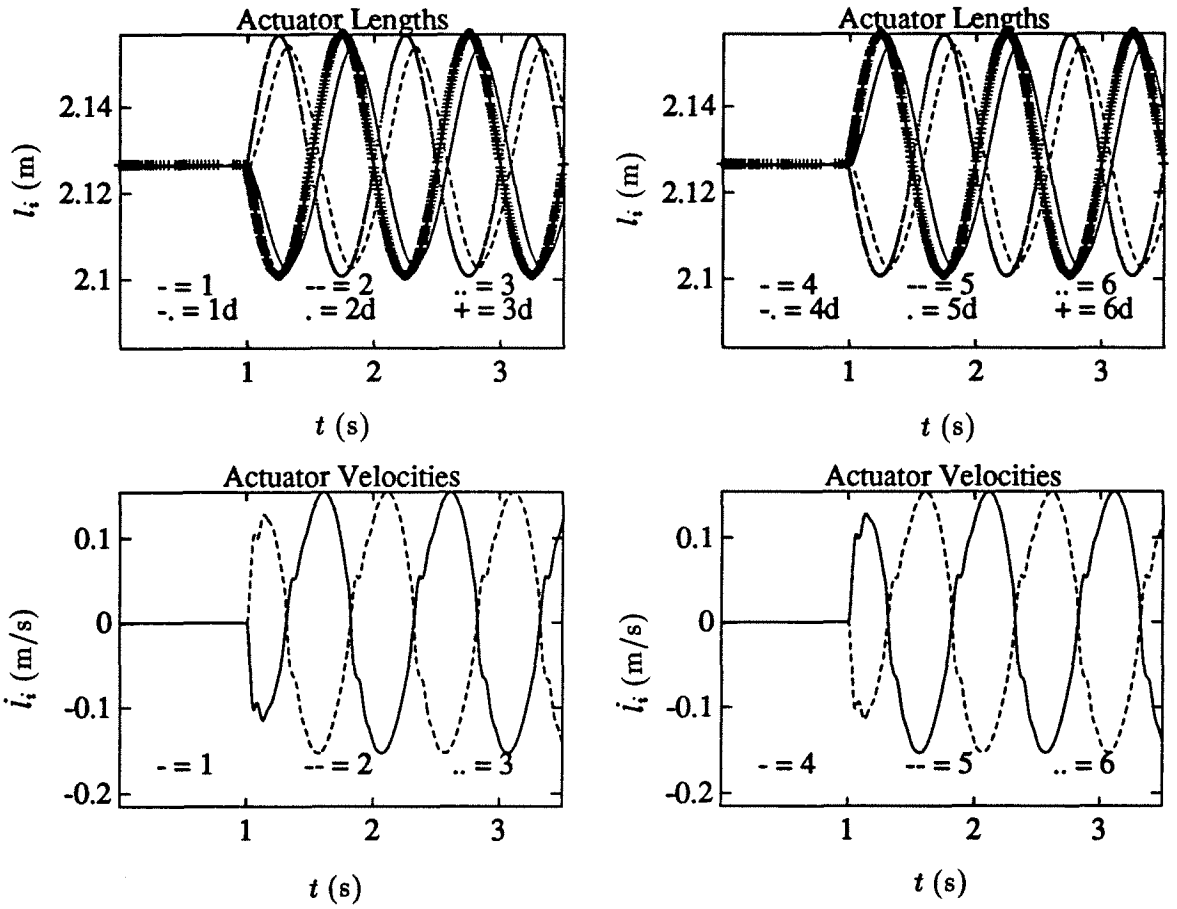


Figure 4.52: Proportional Control Simulation Results for Oscillation about the z axis.

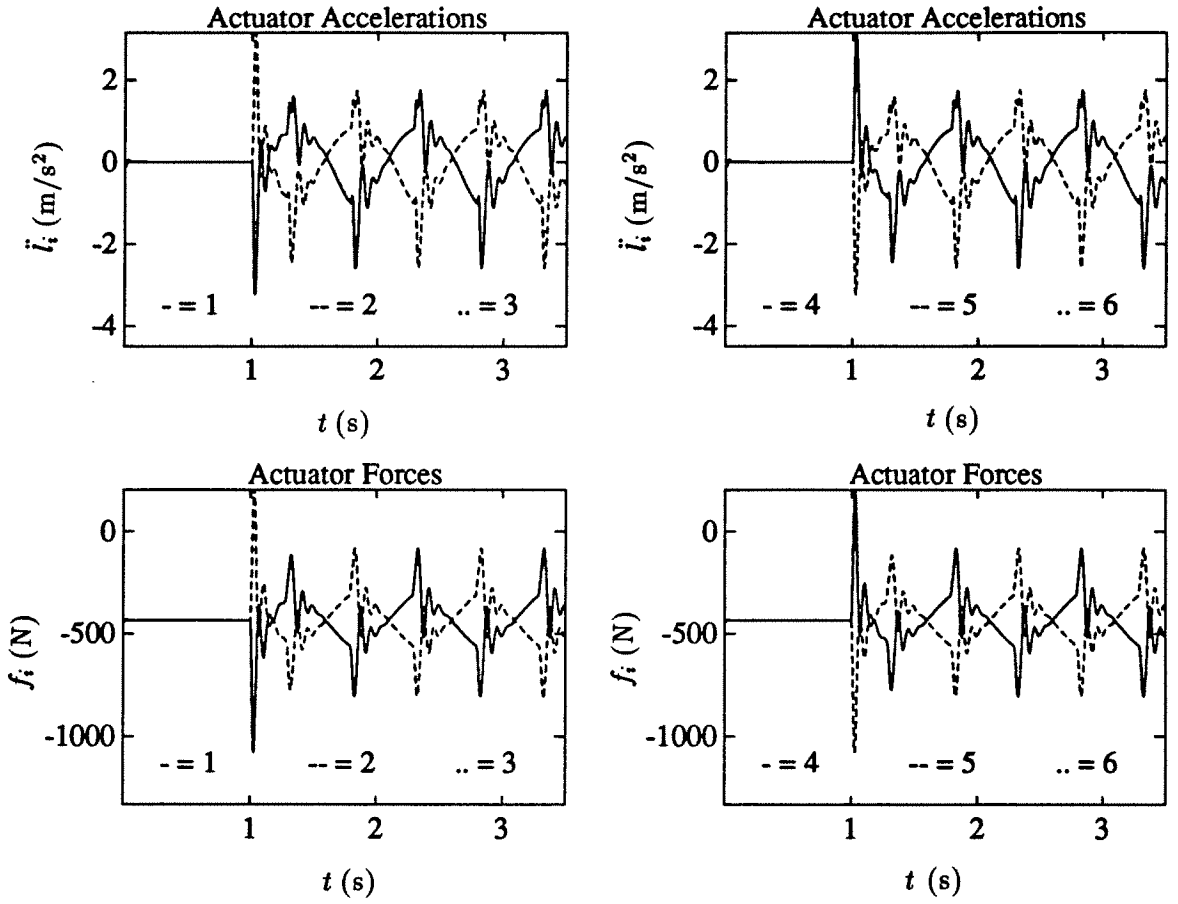


Figure 4.53: Proportional Control Simulation Results for Oscillation about the z axis.

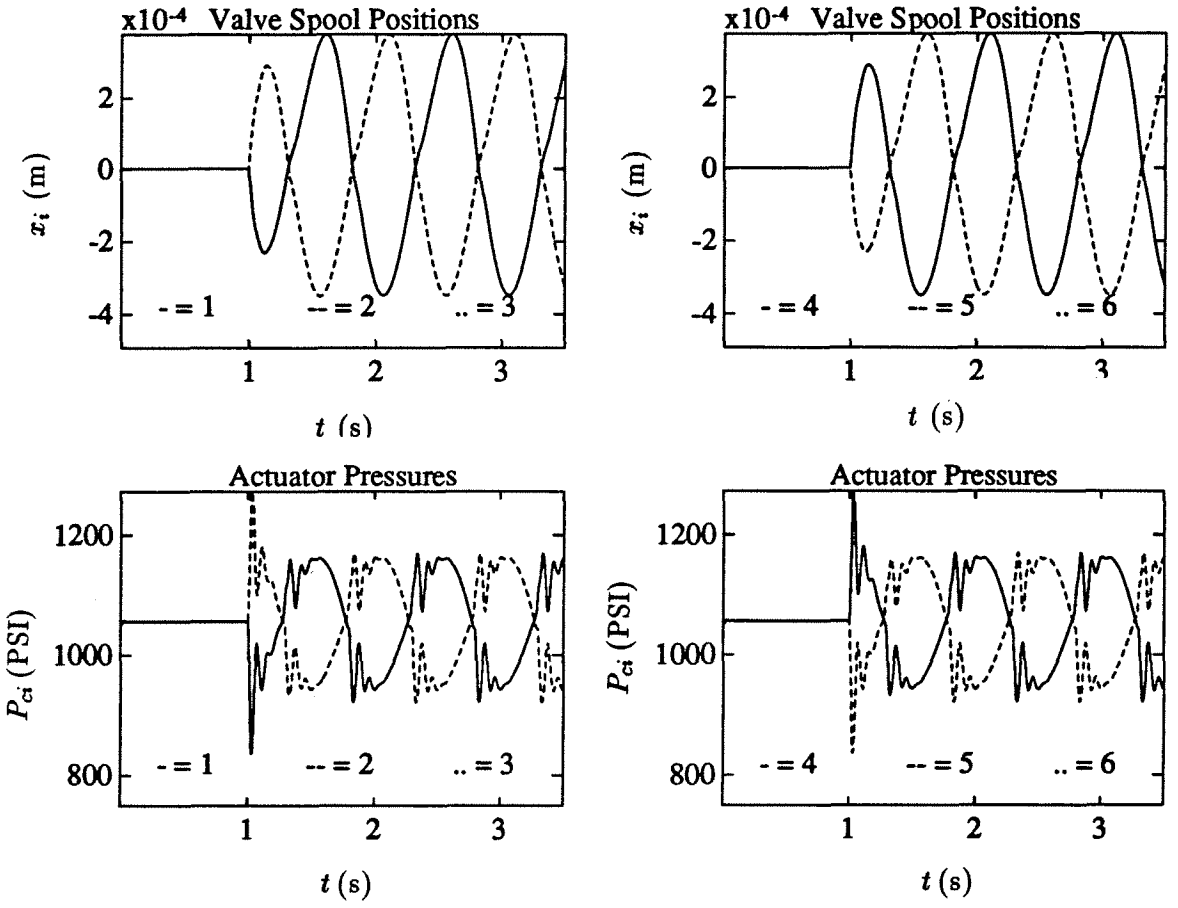


Figure 4.54: Proportional Control Simulation Results for Oscillation about the z axis.

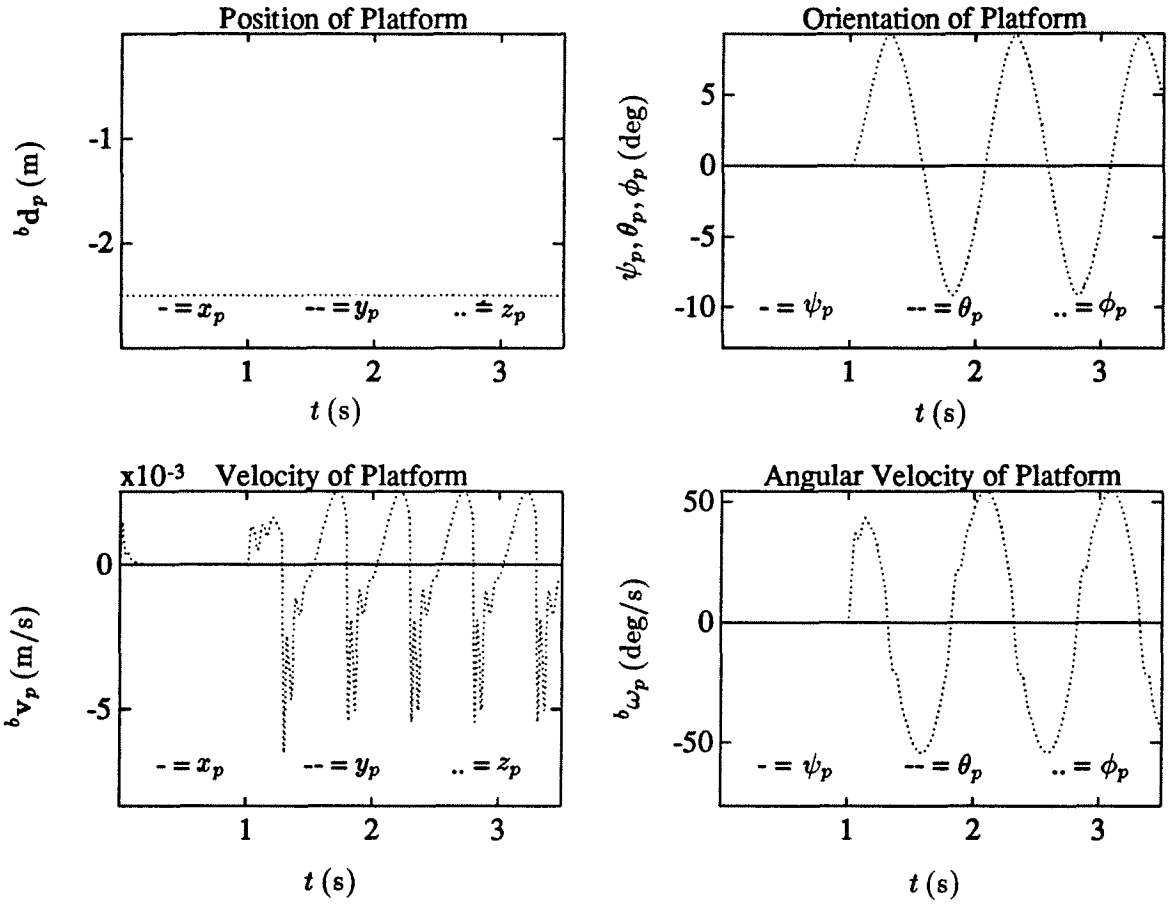


Figure 4.55: Proportional Control Simulation Results for Oscillation about the z axis.

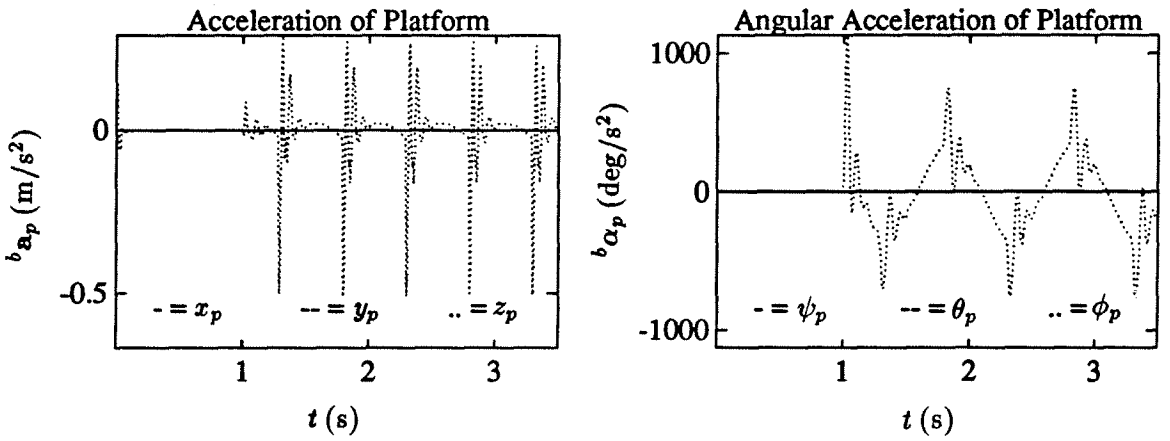


Figure 4.56: Proportional Control Simulation Results for Oscillation about the z axis.

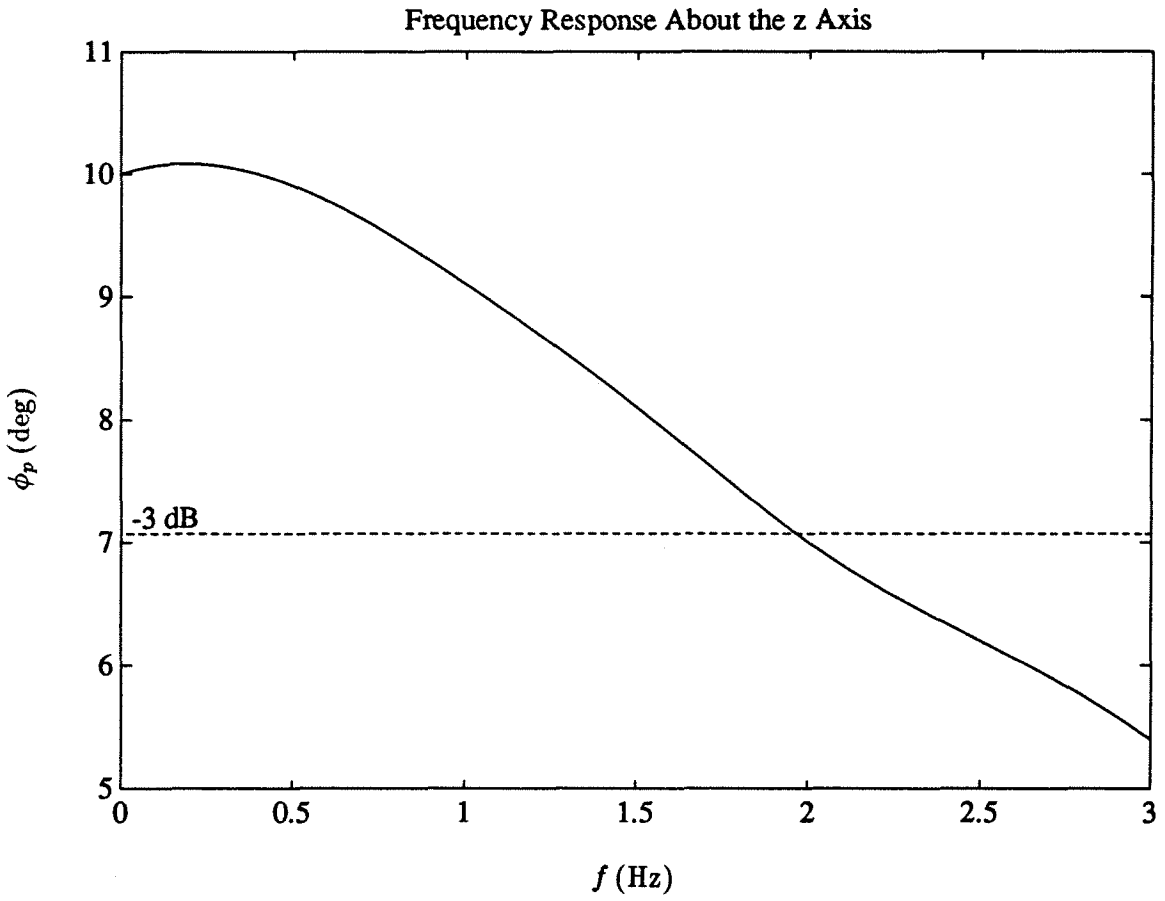


Figure 4.57: Frequency Response for Oscillation about the z axis.

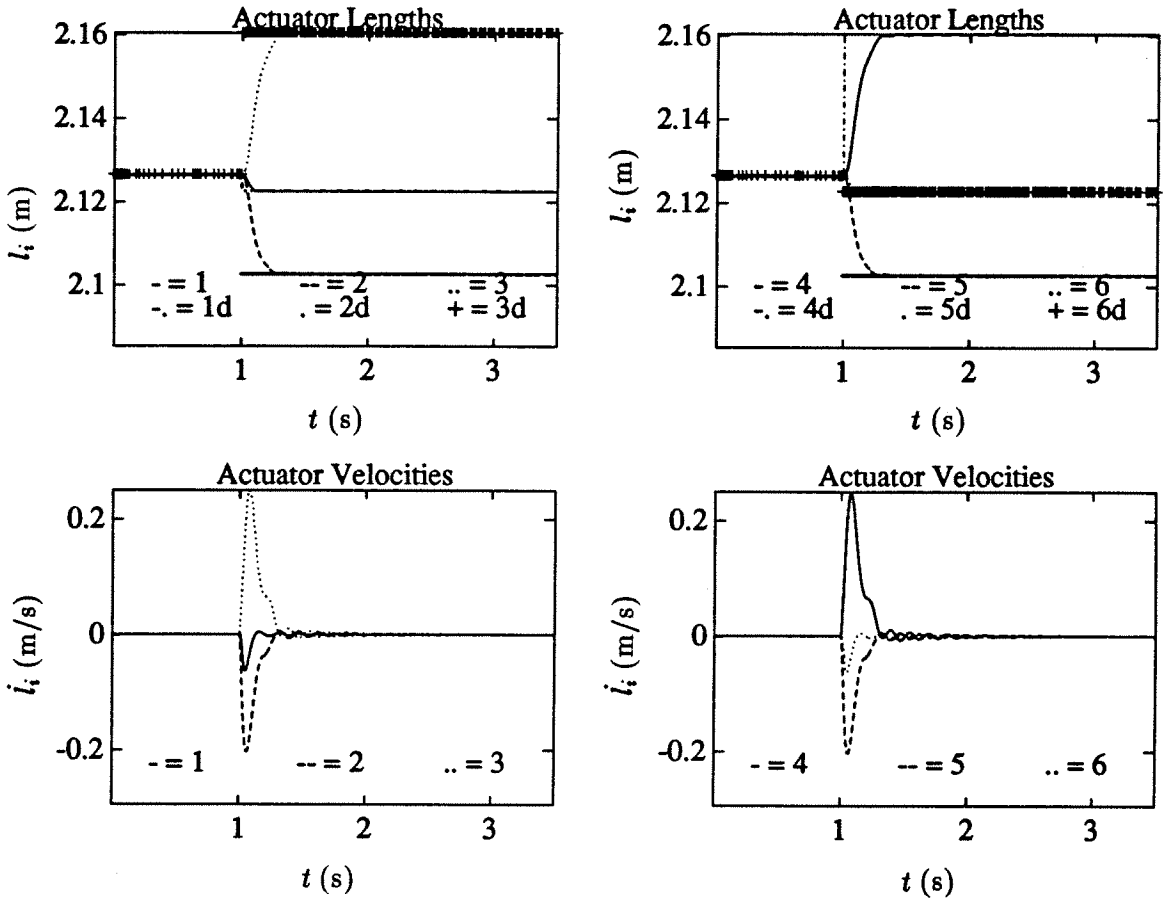


Figure 4.58: Proportional Control Simulation Results for Step Input in the x direction.

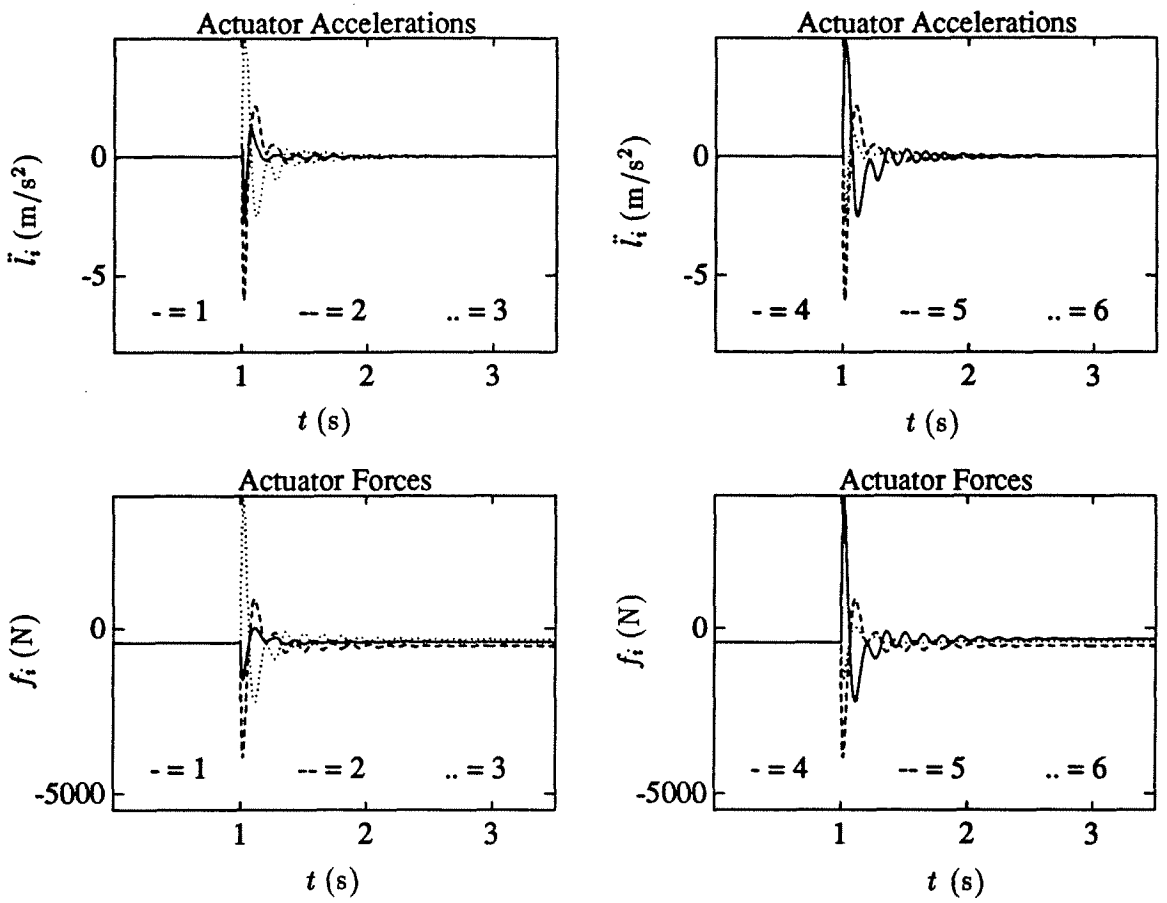


Figure 4.59: Proportional Control Simulation Results for Step Input in the x direction.

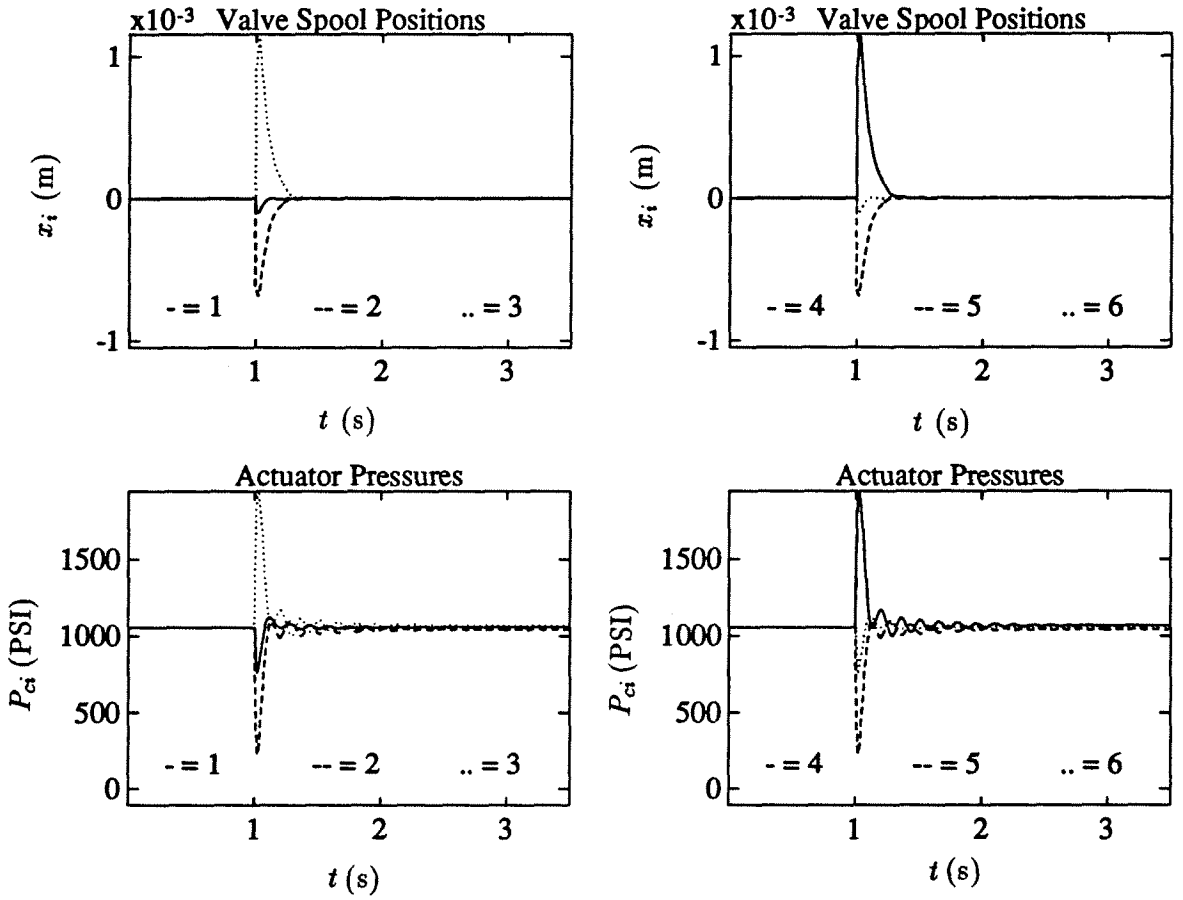


Figure 4.60: Proportional Control Simulation Results for Step Input in the x direction.

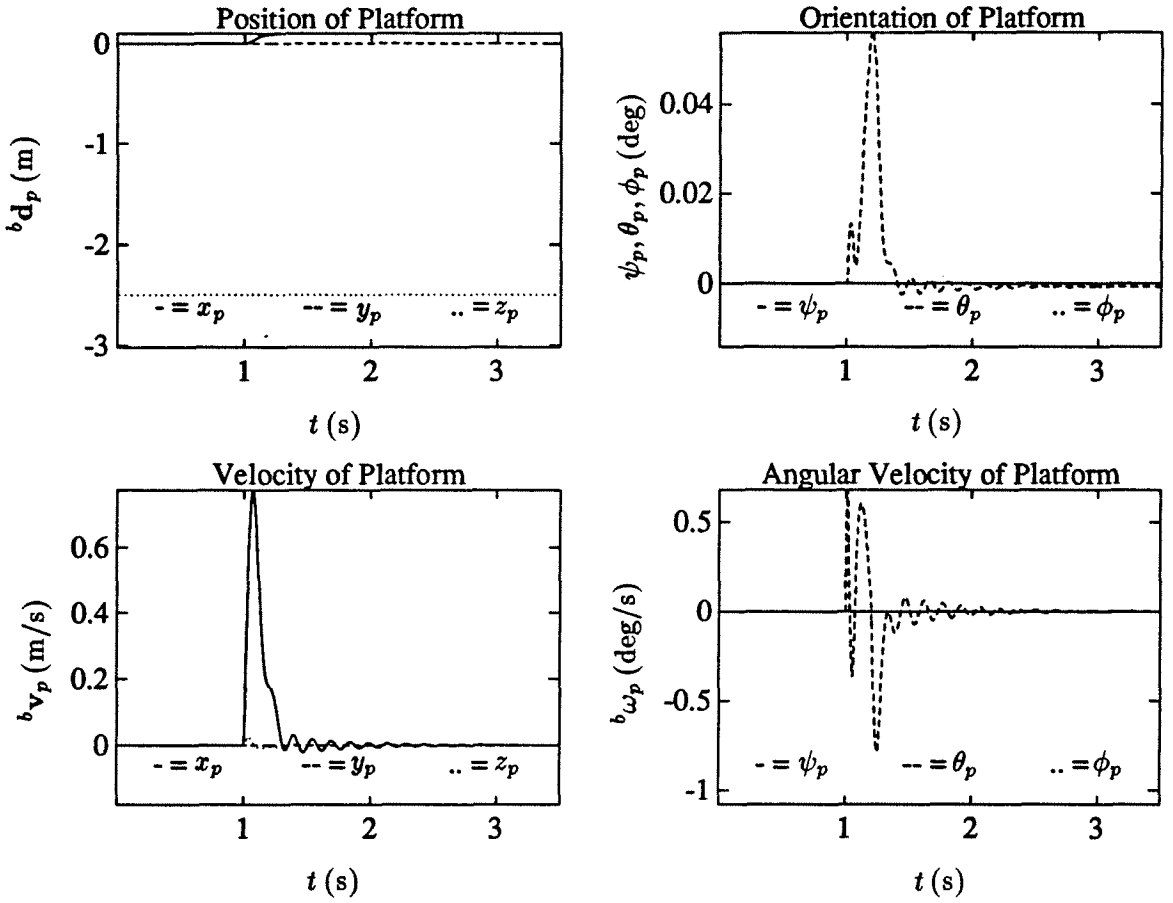


Figure 4.61: Proportional Control Simulation Results for Step Input in the x direction.

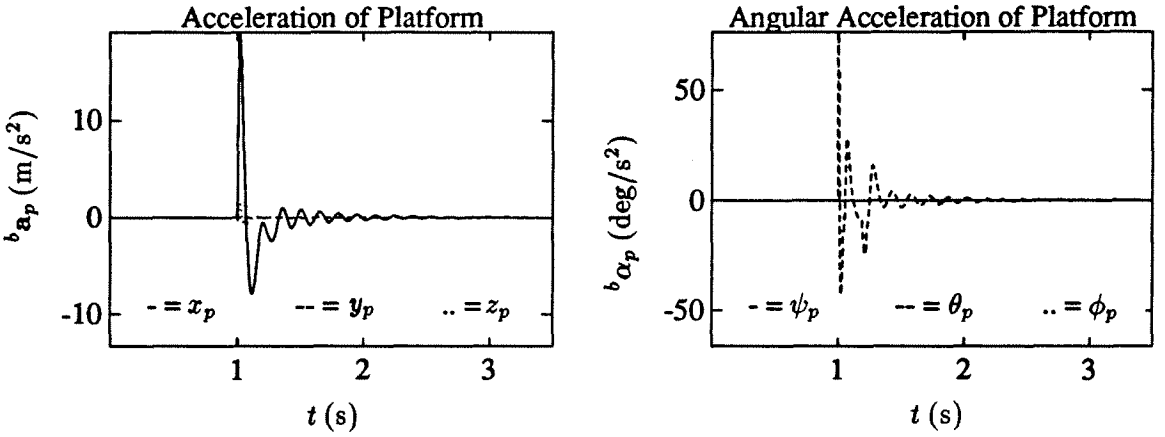


Figure 4.62: Proportional Control Simulation Results for Step Input in the x direction.

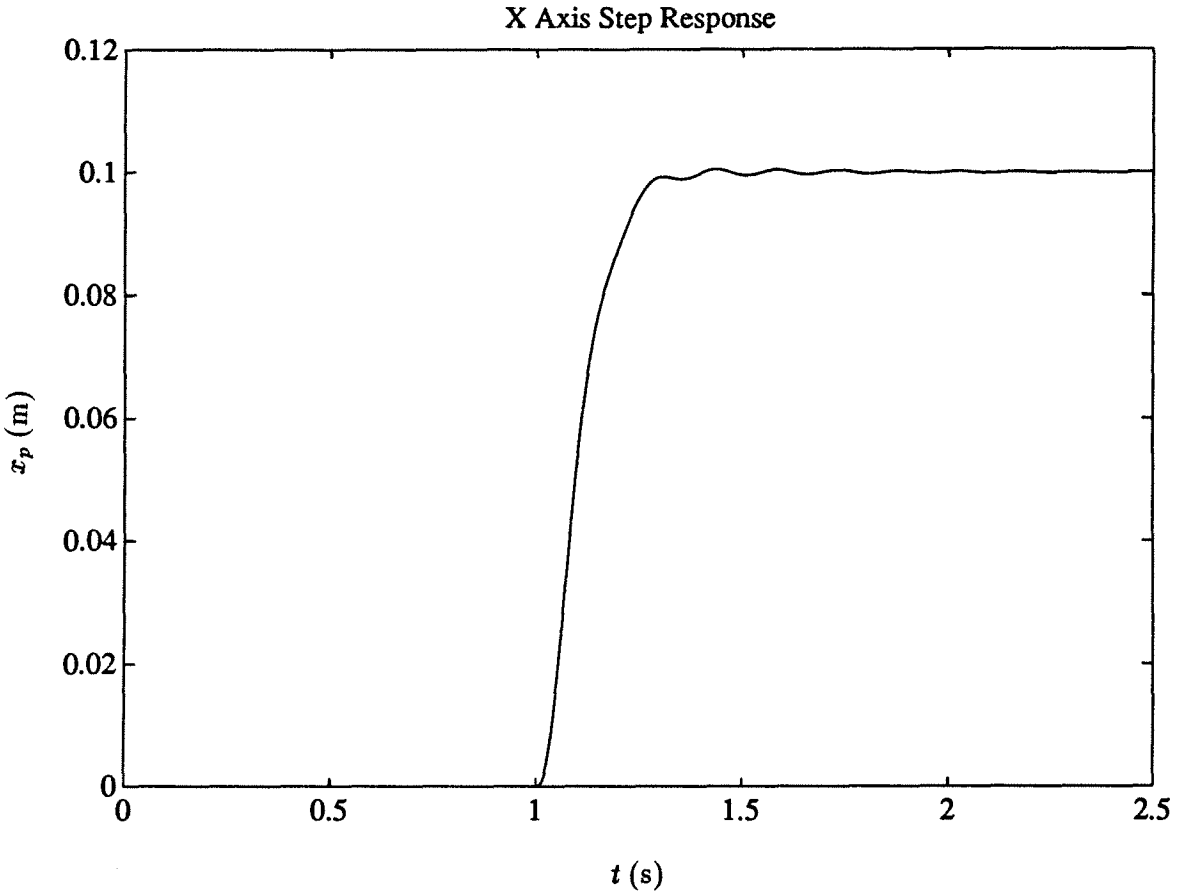


Figure 4.63: Proportional Control Simulation Results for Step Input in the x direction.

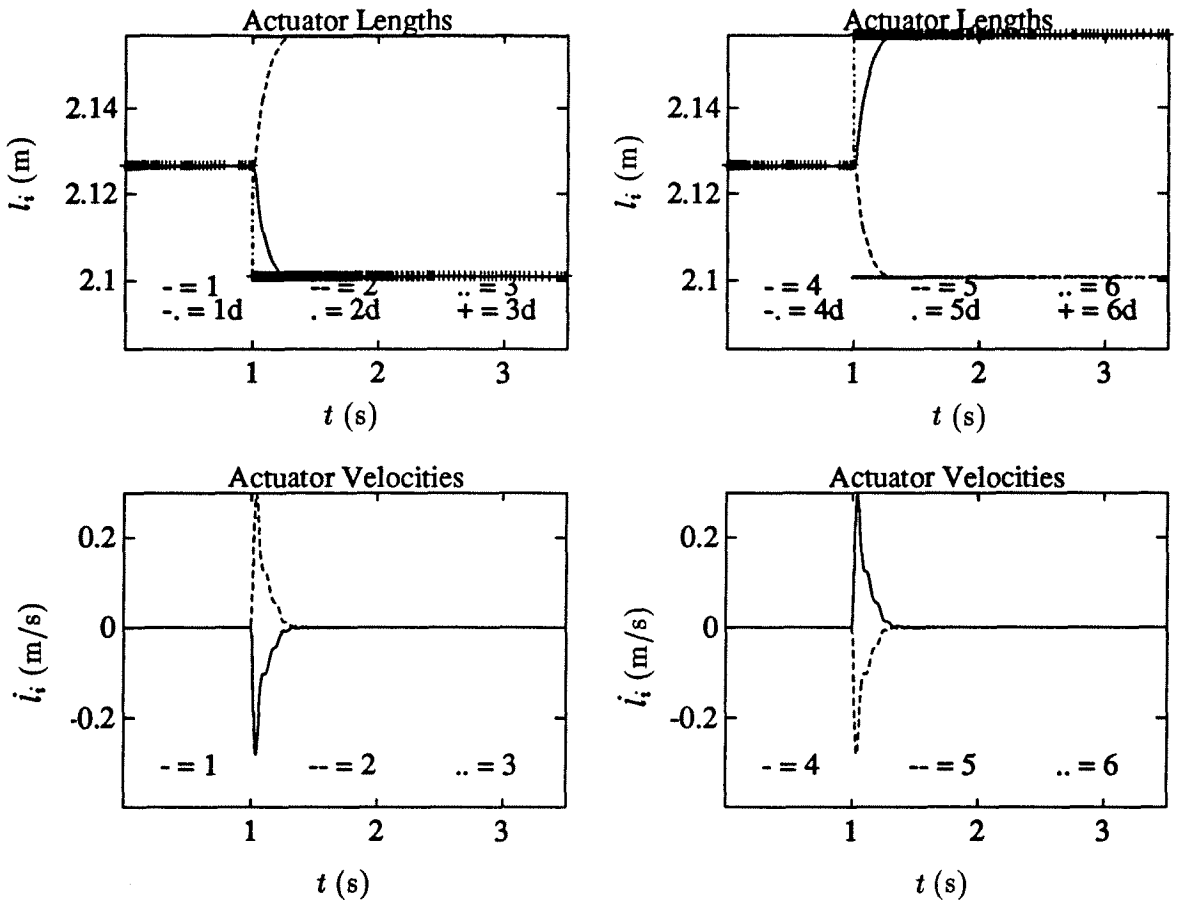


Figure 4.64: Proportional Control Simulation Results for Step Input about the z axis.

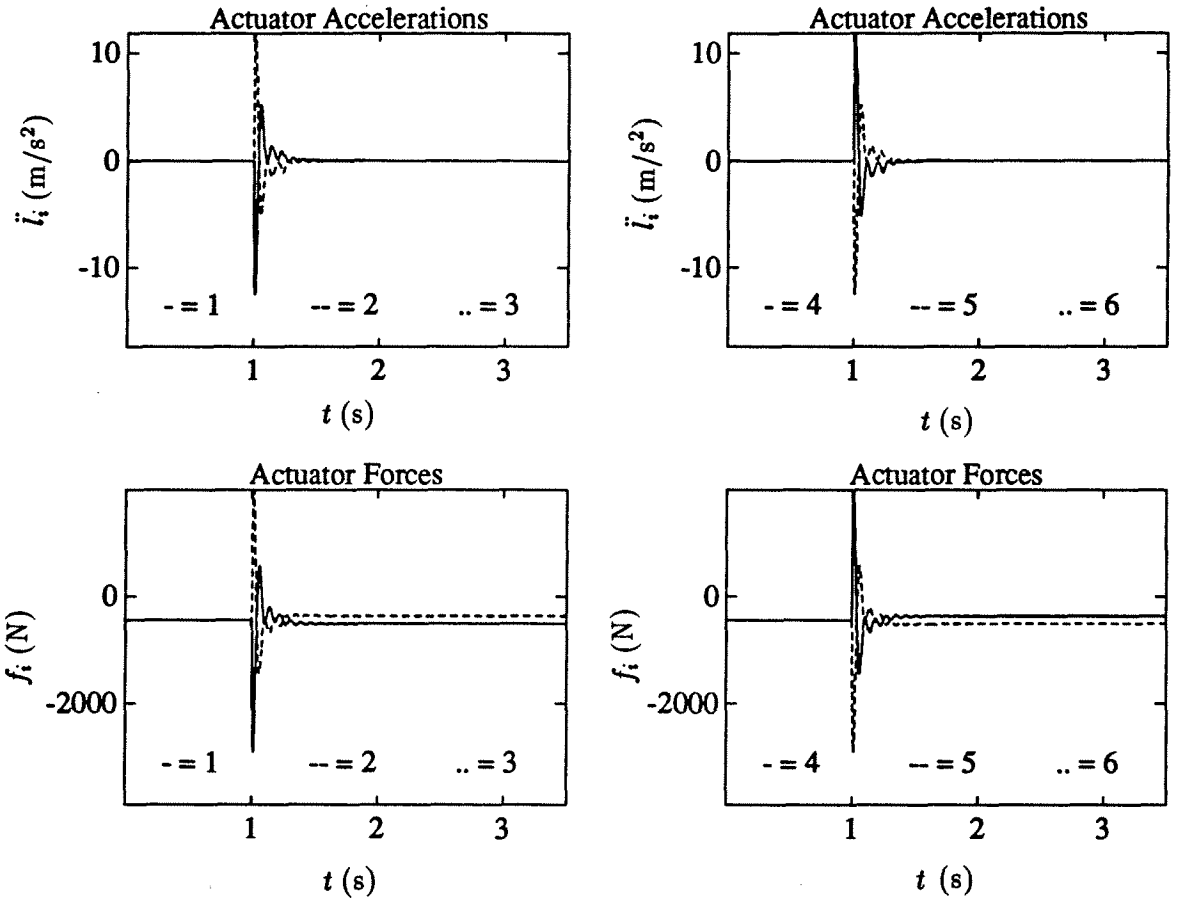


Figure 4.65: Proportional Control Simulation Results for Step Input about the z axis.

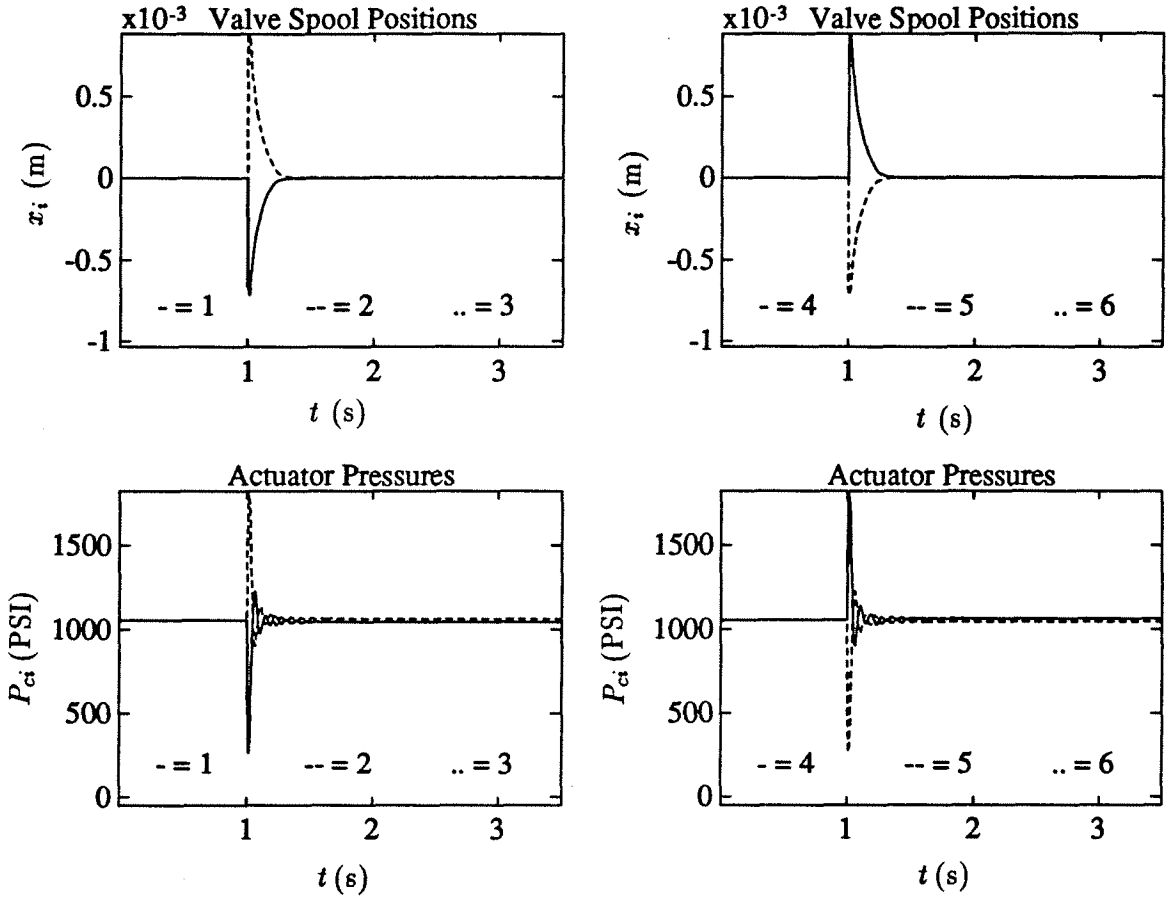


Figure 4.66: Proportional Control Simulation Results for Step Input about the z axis.

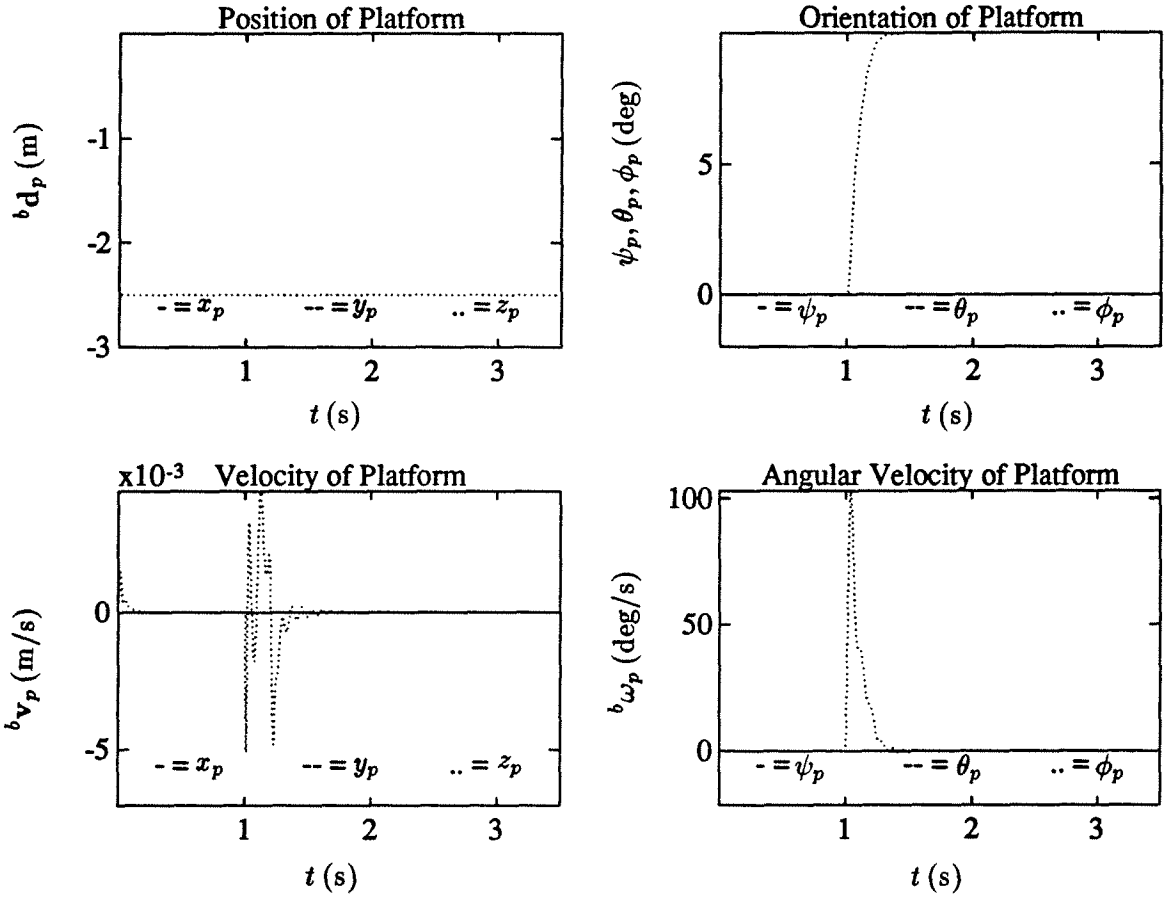


Figure 4.67: Proportional Control Simulation Results for Step Input about the z axis.

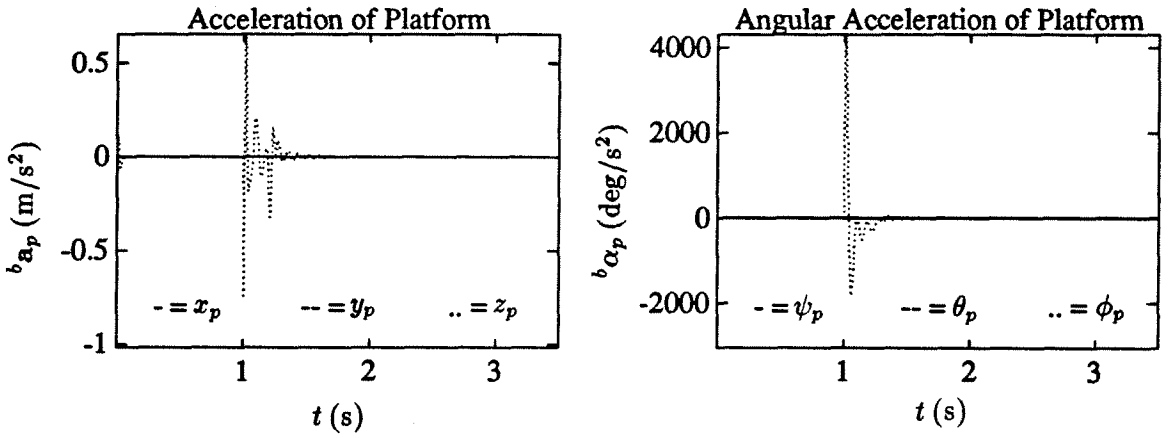


Figure 4.68: Proportional Control Simulation Results for Step Input about the z axis.

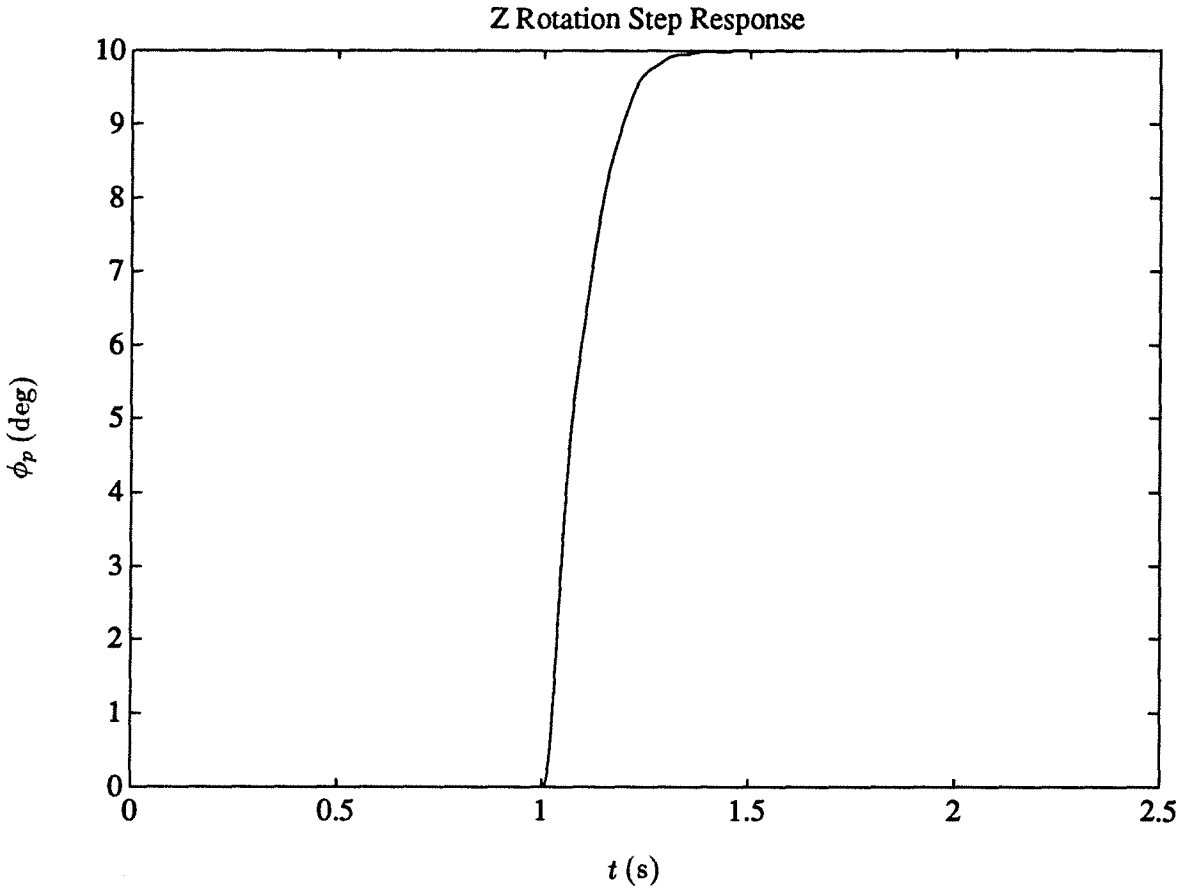


Figure 4.69: Proportional Control Simulation Results for Step Input about the z axis.

Chapter 5

Conclusions

In this thesis we have presented the design of an inverted six degrees-of-freedom, hydraulic, one person motion simulator capable of providing 9.8 m/s^2, 400°/s^2 accelerations and 1 m/s, 30°/s speeds to a 250 kg payload with 1 m, 45° displacements from a nominal center. This simulator allows us to perform human factors and teleoperation research to improve the human interface to heavy equipment used in the forest industry.

We derived the motion simulator's inverse kinematics and jacobian and used them to determine the platform's forward kinematics and controllable workspace. Using Newton's method and the jacobian, we presented an iterative method to solve the platform's forward kinematics. We also plotted the ratios of the singular values of the jacobian to determine where the platform's position and orientation becomes singular. With this information, we will be able to design a computer control system which avoids the motion simulator's singularities, thereby maintaining control of the platform.

The inverted design of our motion simulator has several advantages over those of traditional flight simulators. First, we were able to use narrower hydraulic actuators which require less fluid, making the hydraulic system less expensive. Second, the operator may enter and exit easily without any steps, ramps or ladders because the platform can be lowered to the floor. Finally, in case of system failure, the platform lowers to a level attitude allowing the operator to dismount safely. In order to do this we used isolation and "home" valves in the hydraulic system as well as platform and actuator limit switches in the electrical system. These switches deactivate the isolation valves if the platform or the actuators move too far. Activating the "home" valves lowers the platform to the floor allowing the operator to exit safely.

Using the results of inverse dynamic simulations, we determined the dimensions of the motion simulator as well as the type and size of the hydraulic cylinders and valves. We found that a 0.5 meter platform radius has enough room for an operator, hand controls and a chair. We decided on a base radius of 1.0 meter to prevent the actuators from colliding with the walls and to minimize the forces needed in translational motion. We chose a nominal platform height of 2.5 meters to accommodate the operator's legs during a simulation. As well as the platform itself, we also designed actuators using 1.5 inch bore, 54 inch stroke cylinders and Rexroth three-stage proportional valves. The force-velocity characteristics of this valve and cylinder combination indicate that the actuators will provide the maximum desired forces ($\pm$4000 N) over a -1.3 to 1.1 meter per second velocity range. The three-stage valves feature near linear response, large bandwidth and higher flow capacity than more expensive servo valves.

We also presented a derivation of the motion simulator's combined dynamics and included the results of several closed-loop, proportional gain simulations using a joint-space control strategy. The preliminary results indicated that the platform has a bandwidth of $\approx$ 2 Hertz and a step response time of $\approx$ 0.15 seconds. Further investigation is needed to design a more satisfactory control system and improve the "smoothness" of the platform's motion.

5.1 Contributors

To accomplish a project of this magnitude requires contributions from many people and organizations. One person cannot possibly have the knowledge or expertise in every area required, therefore, we would like to acknowledge and thank the following people for their fine work:

1. Al Mackenzie for his assistance in every phase of this project;

2. Captain Wood and Canadian Airlines for giving us a tour of their flight simulator facility and a ride in their DC-10 simulator;

3. Daniel Ben-Dov for designing the motion simulator's hydraulic system;

4. Jon Finlay and Basic Hydraulics Limited for providing and installing the hydraulic system components;

5. Tim Salcudean for designing the motion simulator's universal joints;

6. Dave Fletcher and the Electrical Engineering Machine Shop for manufacturing the universal joints, length transducer mounts and distribution manifold frame;

7. David Grigg and UBC's Plant Operations for:

   (a) designing the power unit frame;

   (b) designing and installing the electrical system for the power unit; and

   (c) designing and installing the cooling system for the power unit;

8. Jim Mandelli and Yolles Partnership Limited, Consulting Structural Engineers, for designing the motion simulator's base frame;

9. John Calderella and Continental Steel Limited for manufacturing and installing the base and power unit frames;

10. Ben Xiao for wiring the electrical system;

11. Pardeep Sangha for designing and coding a graphical representation of the motion simulator using HOOPS[1];

12. Simon Bachmann for designing the platform frame; and

13. Alison Taylor for designing and coding the motion simulator's control software.

5.2 Suggestions for Further Work

Some suggestions for further work in order to complete this project include the following:

1. Finish manufacturing the universal joints;

[1] HOOPS is a graphics software library and a trademark of Ithaca Software.

2. Manufacture the platform frame and seat;
3. Assemble the platform frame, universal joints and actuators;
4. Finish assembling the hydraulic system, including open-loop tests;
5. Assemble and test the safety systems;
6. Complete design and implementation of the computer control system and perform closed-loop tests;
7. Incorporate operator hand controls with the platform frame; and
8. Program excavator motion simulation.

Bibliography

[1] Mannesmann-Rexroth and Hydraudyne Systems and Engineering B.V., Kruisbroeksestraat 1a, P.O. Box 32, 5280 AA Boxtel, The Netherlands. *The Refined Precision in Motion.* Company Brochure.

[2] Kawasaki Heavy Industries, Limited Precision Machinery Division, World Trade Center Building,. 4-1 Hamamatsu-cho 2-chome, Minato-ku, Tokyo, 105 Japan. *Six-Degree-of-Freedom Motion Base*, 1991.

[3] D. Stewart. A platform with six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers*, 180(15):371–386, 1966.

[4] V. E. Gough and S. G. Whitehall. Universal tyre test machine. In *Proceedings, Ninth International Technical Congress FISITA*, page 117, May 1962. Institution of Mechanical Engineers.

[5] K.H. Hunt. Structural kinematics of in-parallel-actuated robot arms. *Journal of Mechanisms, Transmissions and Automation in Design*, 105:705–712, 1983.

[6] C.C. Nguyen, S.S. Antrazi, Z-L. Zhou, and C.E. Campbell, Jr. Experimental study of motion control and trajectory planning for a Stewart platform robot manipulator. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 1873–1878, 1991.

[7] Kevin Cleary and Tatsuo Arai. A prototype parallel manipulator: Kinematics, construction, software, workspace results, and singularity analysis. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 566–571, 1991.

[8] E.F. Fichter. Kinematics of a parallel connection manipulator. In *ASME Design Engineering Technical Conference*, 1984. Paper Number 84-DET-45.

[9] E.F. Fichter. A Stewart platform based manipulator: General theory and practical construction. *International Journal of Robotics Research*, 5(2):157–182, 1986.

[10] M.G. Mohamed and J. Duffy. A direct determination of the instantaneous kinematics of fully parallel robot manipulators. *Transactions of the ASME*, 107:226–229, 1985.

[11] D.C.H. Yang and T.W. Lee. Feasibility study of a platform type of robotic manipulators from a kinematic viewpoint. *Journal of Mechanisms, Transmissions and Automation in Design*, 106:191–198, 1984.

[12] W.Q.D. Do and D.C.H. Yang. Inverse dynamic analysis and simulation of a platform type of robot. In *ASME Design Engineering Technical Conference*, 1986. Paper Number 86-DET-94.

[13] W.Q.D. Do and D.C.H. Yang. Inverse dynamic analysis and simulation of a platform type of robot. *Journal of Robotic Systems*, 5(3):209–207, 1988.

[14] Prabjot Nanua, Kenneth J. Waldron, and Vasudeva Murthy. Direct kinematic solution of a Stewart platform. *IEEE Transactions on Robotics and Automation*, 6(4):438–444, 1990.

[15] M. Griffis and J. Duffy. A forward displacement analysis of a class of Stewart platforms. *Journal of Robotic Systems*, 6(6):703–720, 1989.

[16] Zheng Geng, Leonard S. Haynes, and Robert L. Carroll. Direct forward kinematic solution for general Stewart platforms. *ASME Press Series Robotics and Manufacturing: Recent Trends in Research, Education, and Applications*, 3:11–16, 1990.

[17] Carlo Innocenti and Vincenzo Parenti-Castelli. Direct position analysis of the Stewart platform mechanism. *Journal of Mechanisms and Machine Theory*, 25(6):611–621, 1990.

[18] J-P. Merlet. Assembly modes and direct kinematics of parallel manipulators. *ASME Press Series Robotics and Manufacturing: Recent Trends in Research, Education and Applications*, 3:43–48, 1990.

[19] J.E. Dieudonne et al. An actuator extension transformation for a motion simulator and an inverse transformation applying Newton-Raphson's method. Technical Report D-7067, NASA, 1972.

[20] W.P. Koevermans and C.J. Jansen. Design and performance of the four-degrees-of-freedom motion system of the NLR Research Flight Simulator. In *Proceedings of the Flight Simulation/Guidance Systems Simulation, AGARD Symposium*, The Hague, 1975.

[21] Yutaka Koizumi, Masao Ohashi, Tōru Yoshinari, Hiroshi Muramoto, and Tomoyuki Takahashi. Development of six-degree-of-freedom motion base. Technical report, Kagoshima University, Faculty of Engineering, Department of Mechanical Engineering, 1991.

[22] J.R. Duncan and E.L. Wegscheid. Development of a human factors research laboratory for off-road vehicle operator workstation design. In *Proceedings of the Human Factors Society-26th Annual Meeting*, pages 896–900, 1982.

[23] Lloyd D. Reid and Meyer A. Nahon. Response of airline pilots to variations in flight simulator motion algorithms. *Journal of Aircraft*, 25(7):639–646, 1988.

[24] Clifford T. Morgan, editor. *Human Engineering Guide to Equipment Design.* McGraw-Hill, New York, NewYork, 1963.

[25] P. A. Drexel. *A Six Degrees-of-Freedom, Hydraulic, One Person Motion Simulator Users Manual.* Department of Electrical Engineering, The University of British Columbia. In Progress.

[26] William H. Press, Saul A. Teukolsky, Brian P. Flannery, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1988.

[27] Herbert E. Merritt. *Hydraulic Control Systems.* Wiley and Sons, New York, New York, 1967.

[28] Taco J. Viersma. *Analysis, Synthesis, and Design of Hydraulic Servo Systems and Pipelines.* Elsevier Publishing Company, Amsterdam, New York, 1980.

[29] Joseph E. Shigley. *Mechanical Engineering Design.* McGraw-Hill, Tokyo, Japan, 1963.

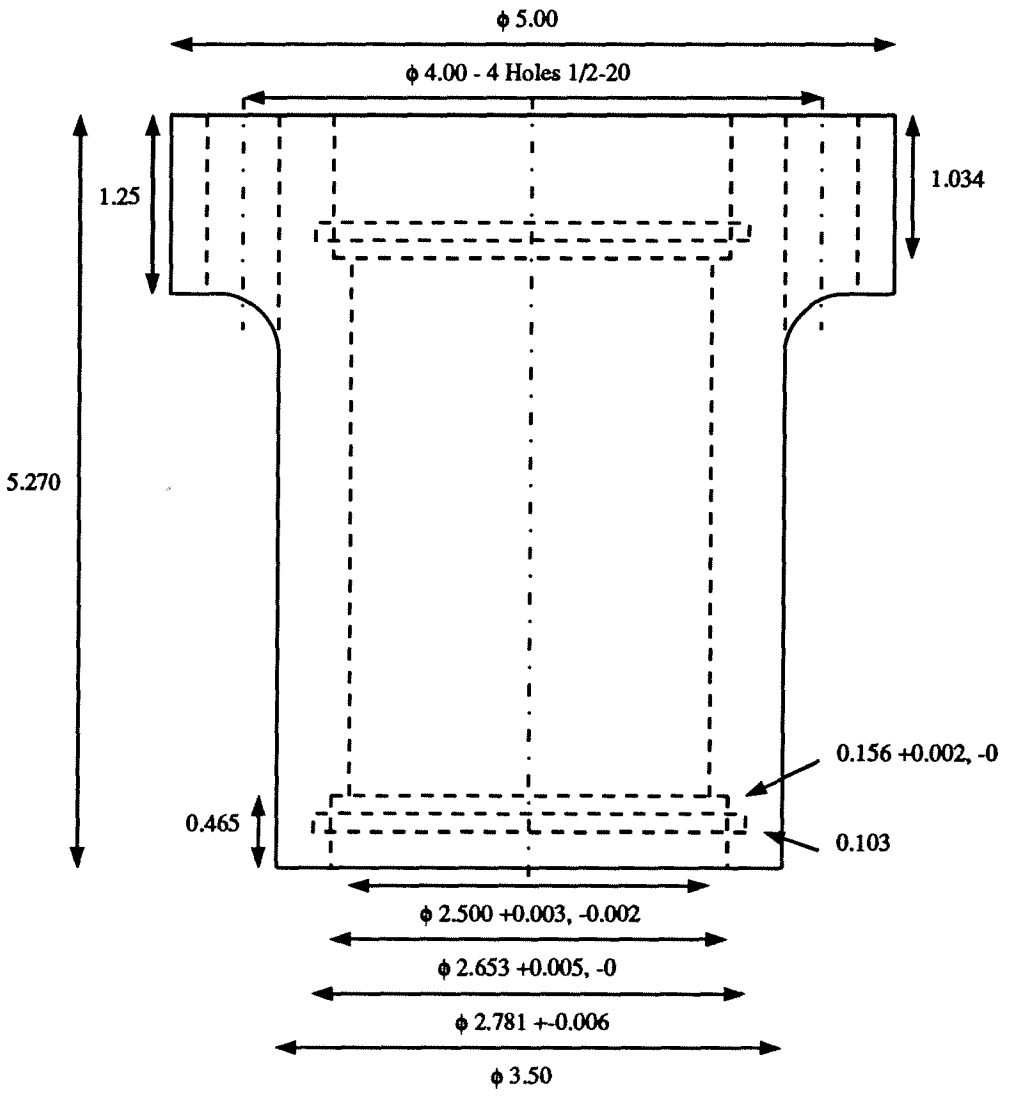
[30] Mark W. Spong and M. Vidyasagar. *Robot Dynamics and Control.* John Wiley and Sons, New York, New York, 1989.

[31] Charles L. Phillips and Royce D. Harbor. *Feedback Control Systems.* Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

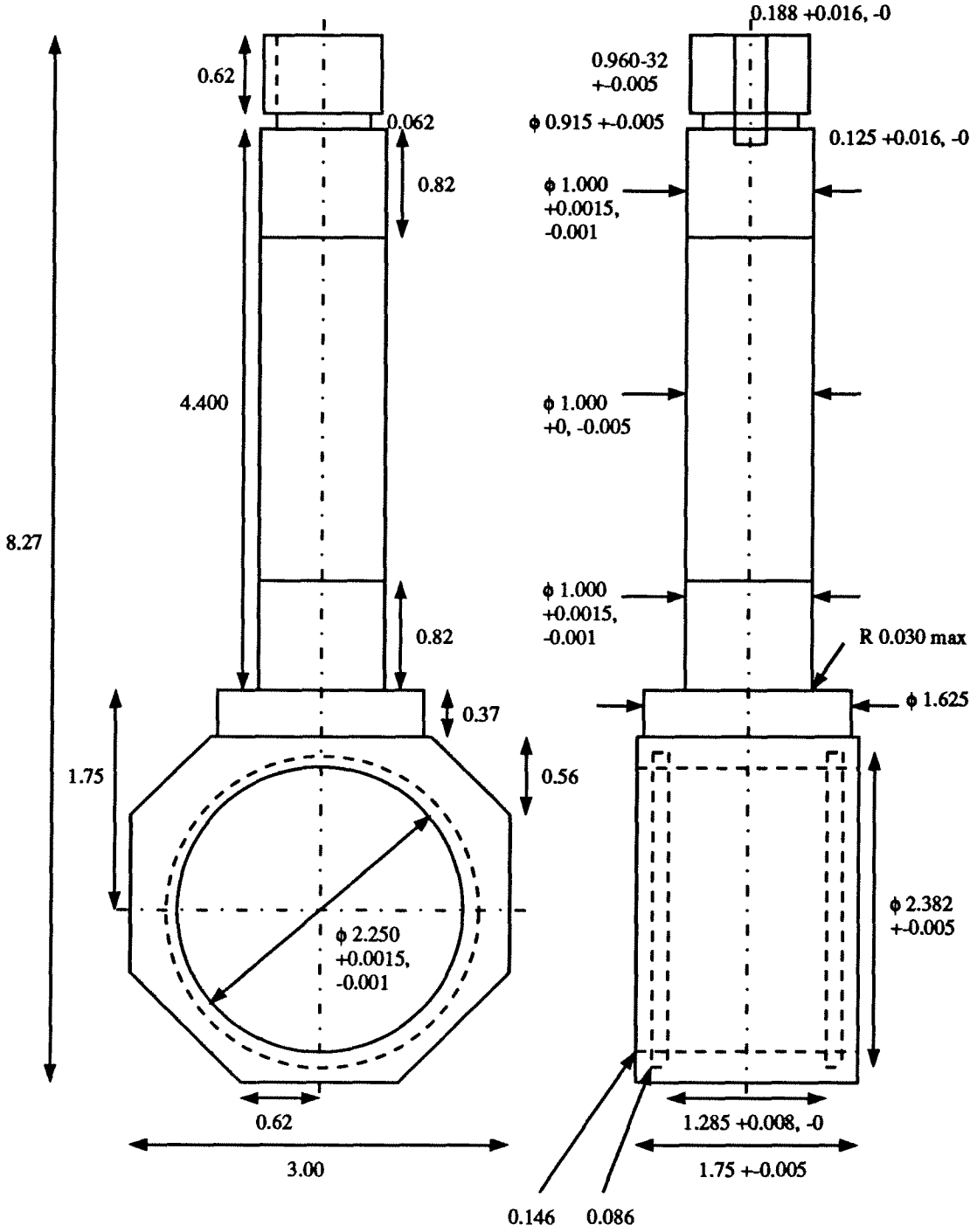
Appendix A

Joint Components

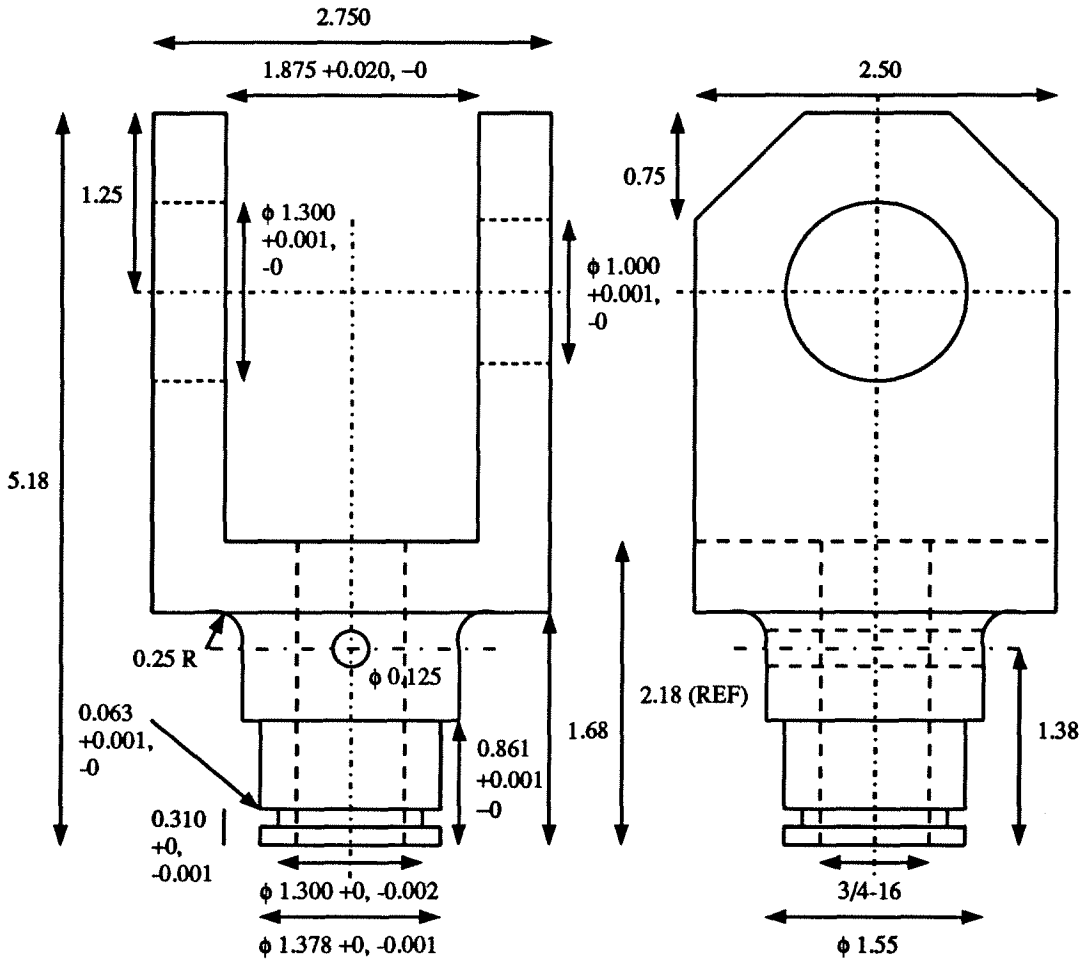
A.1 Base



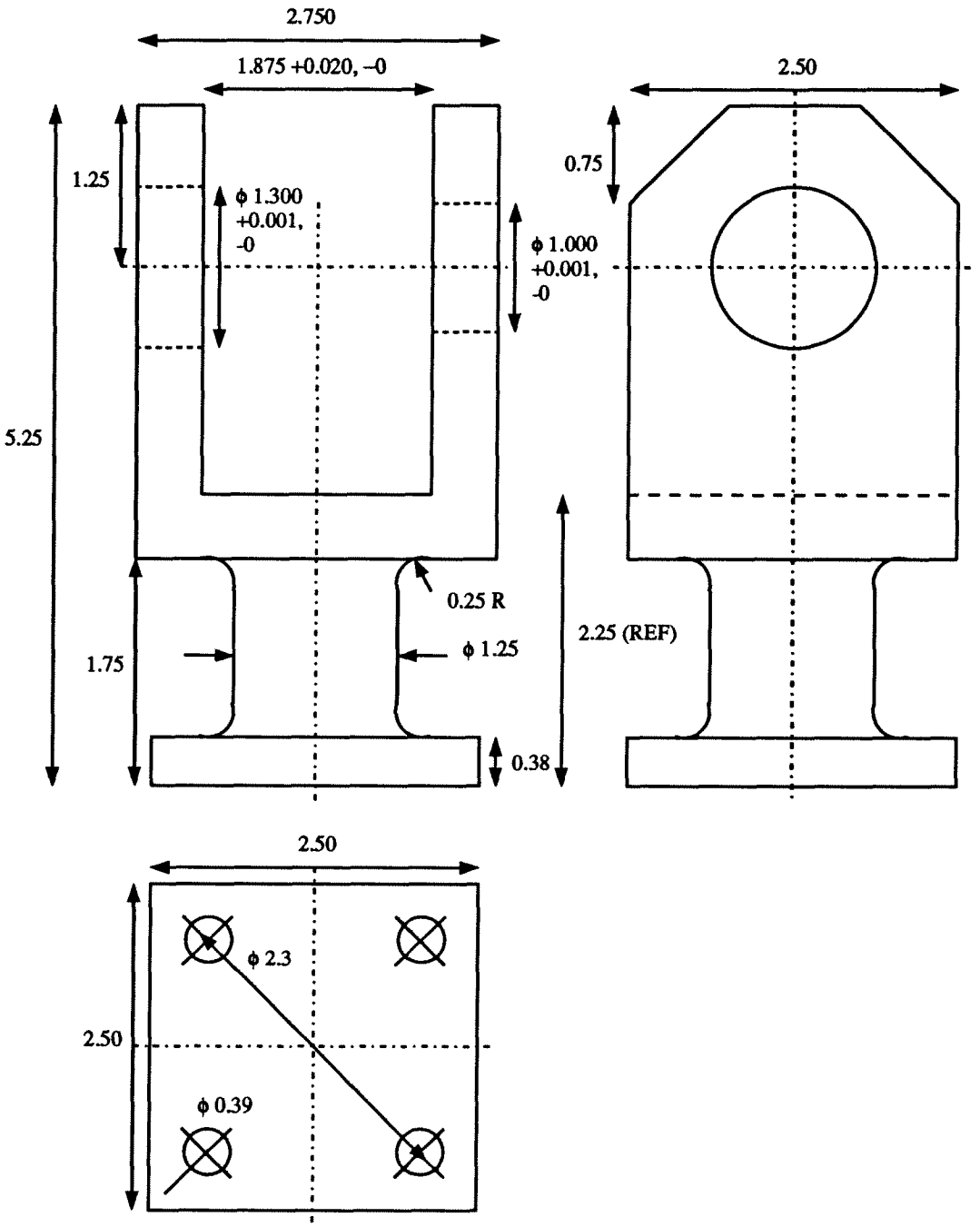
A.2 Base Shaft



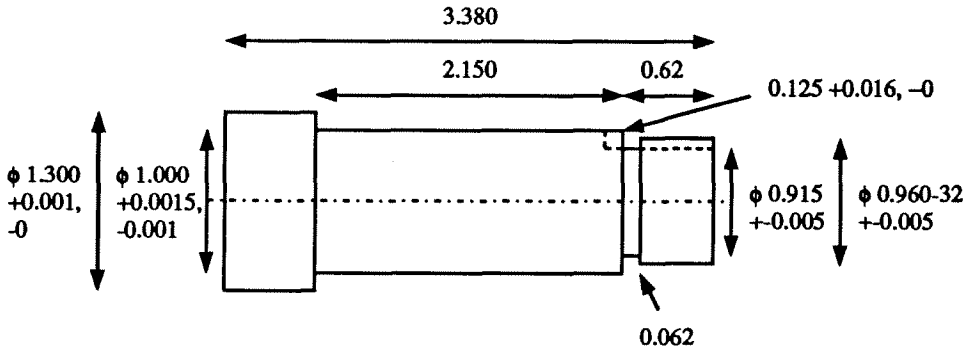
A.3 Rod Fork


2.750
1.875 +0.020, –0
2.50
0.75
1.25
ϕ 1.300
+0.001,
-0
ϕ 1.000
+0.001,
-0
5.18
0.25 R
ϕ 0.125
2.18 (REF)
0.063
+0.001,
-0
1.68
0.861
+0.001
–0
1.38
0.310
+0,
-0.001
ϕ 1.300 +0, -0.002
3/4-16
ϕ 1.378 +0, -0.001
ϕ 1.55

A.4 Cylinder Fork



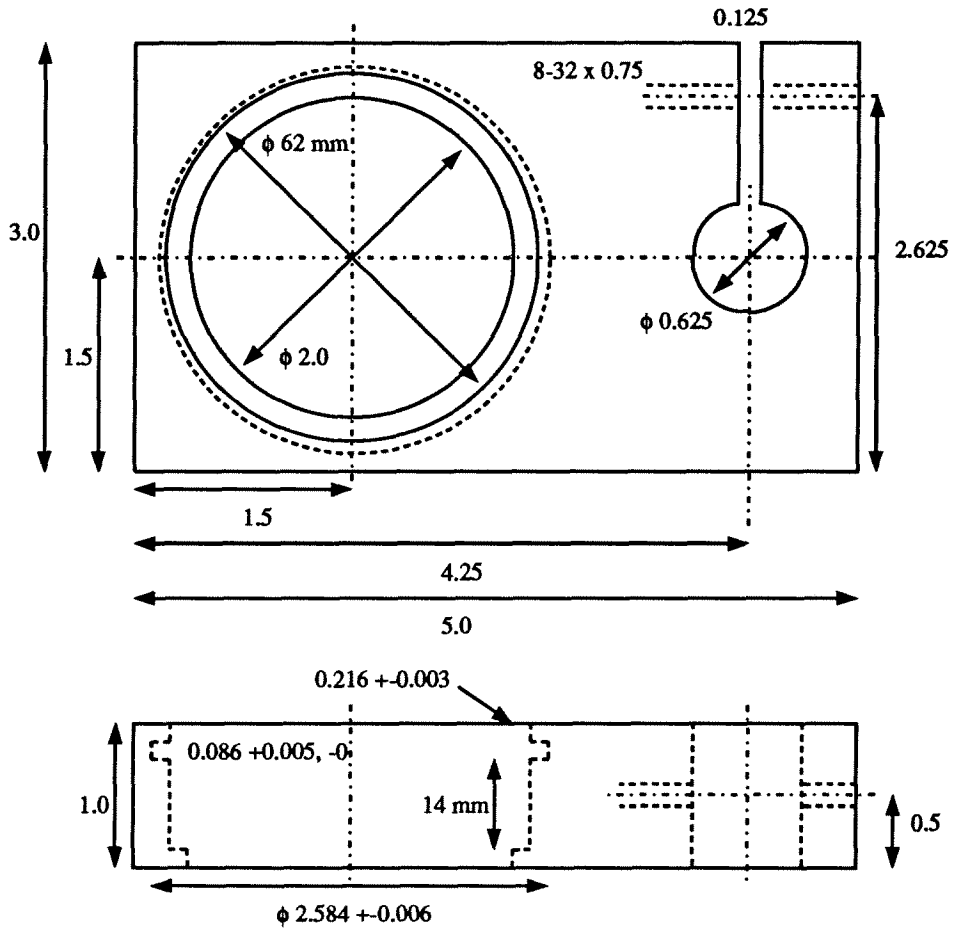
A.5 Short Shaft



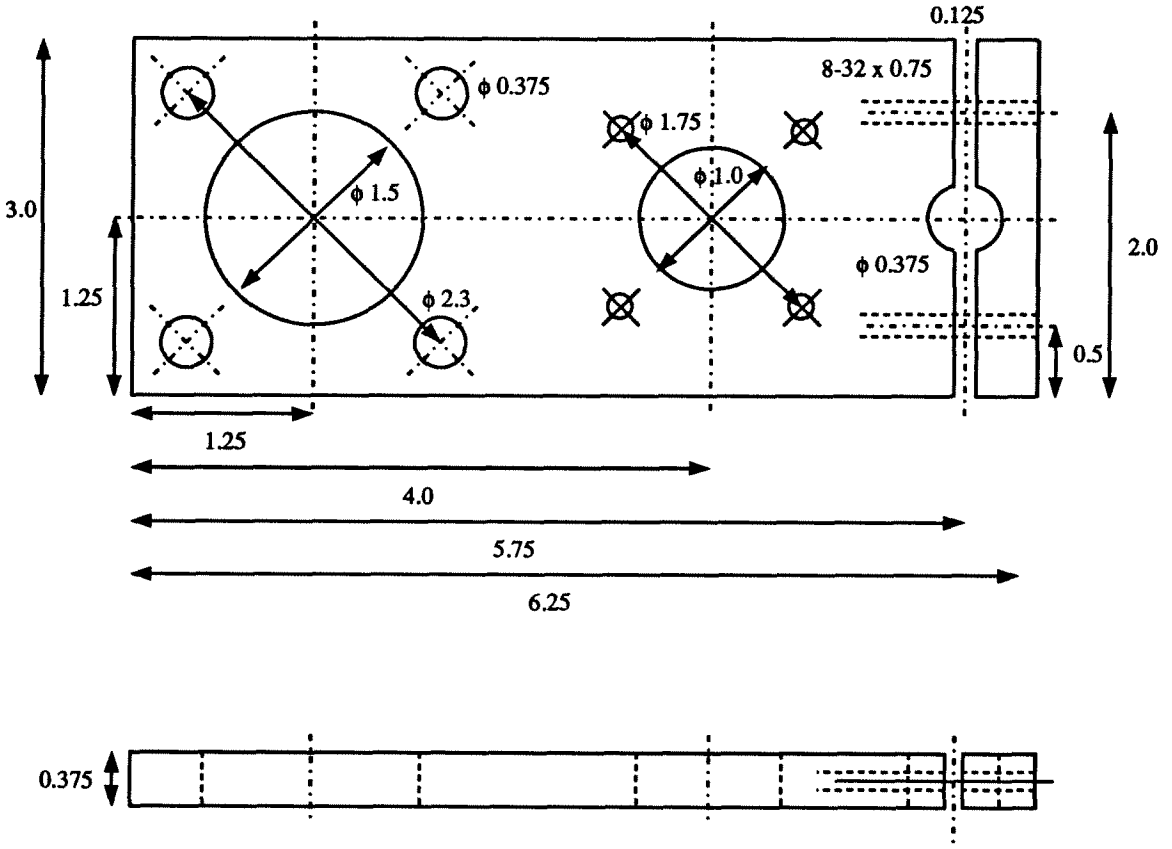
Appendix B

Position Transducer Mounting System Components

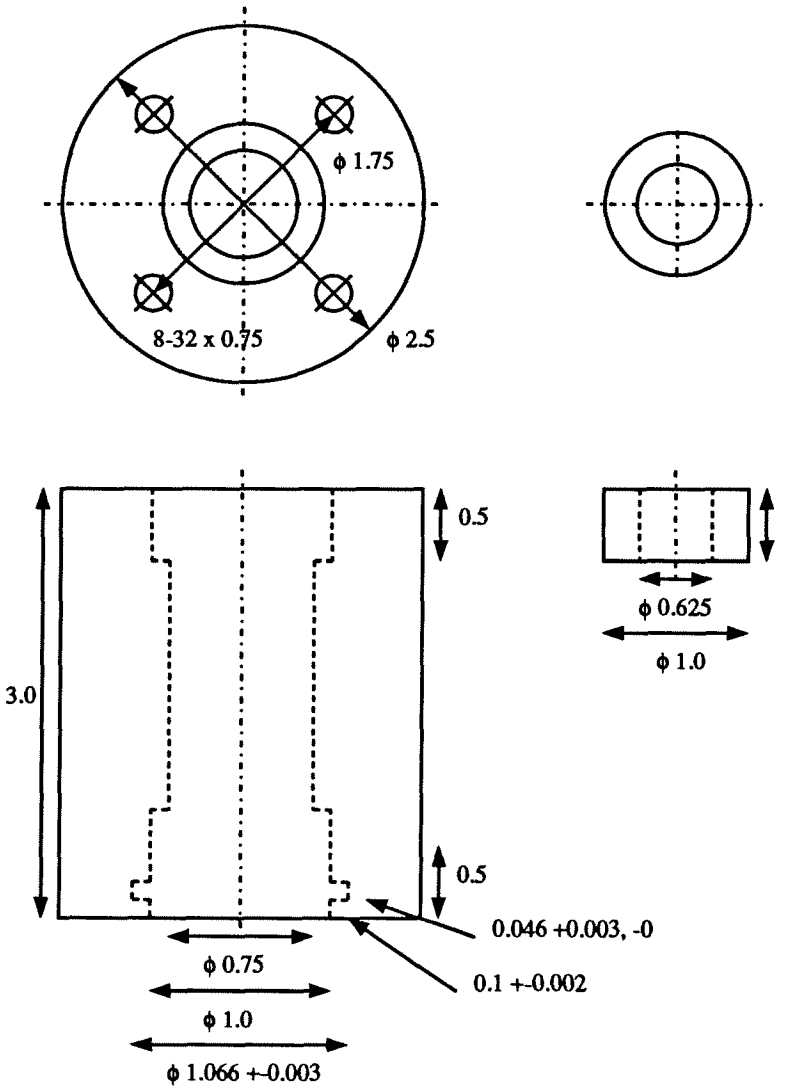
B.1 Clamp Plate



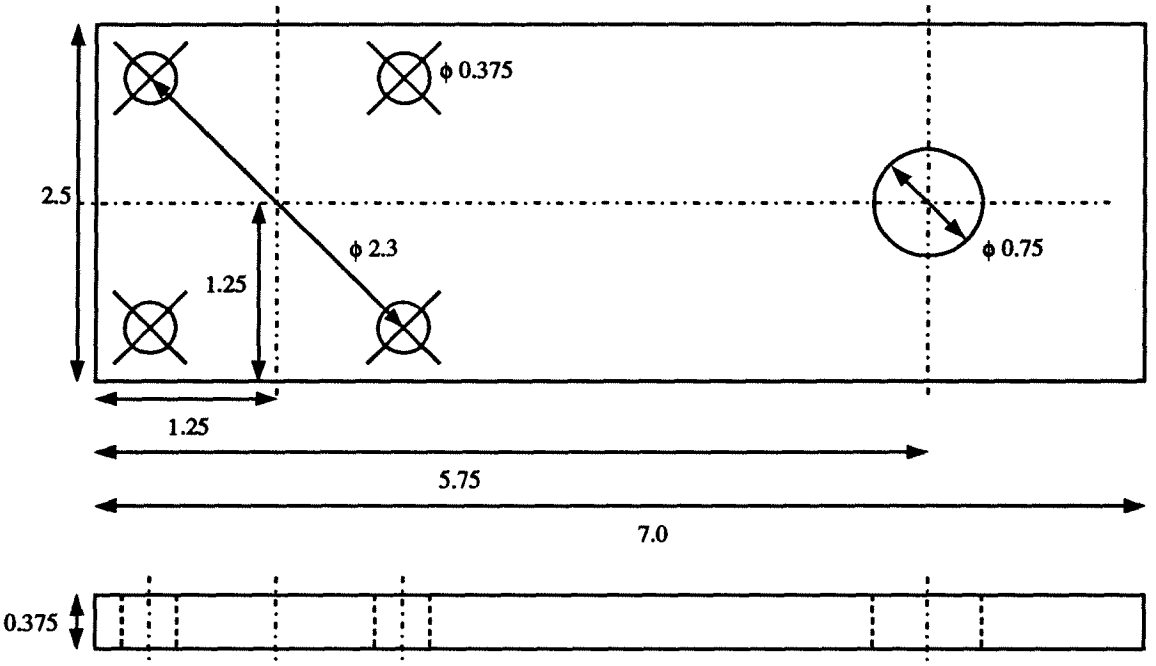
B.2 Tube Guide Plate



B.3 Tube Guide



B.4 Transducer Bracket



Appendix C

Singularity Source Code

C.1 plotsingratios.m

```
function plotsingratios( p1, p1s, p1e, p1inc, p2, p2s, p2e, ...
         p2inc, xp, yp, zp, psip, thep, phip )

% File: plotsingratios.m
% Project: A Six Degrees-of-Freedom, Hydraulic, One Person Motion Simulator
% For: Dr. Tim Salcudean and Dr. Peter Lawrence
% By: Peter Drexel
% Copyright University of British, 1992

% plot the max to min singular value ratios of the platform's jacobian
% while varying two of the platform's position/orientation parameters

% p1 = (1..6) first platform parameter to vary, for example, xp position
% p1s = starting value of first parameter
% p1e = ending value of first parameter
% p1inc = increment value of first parameter
% p2, p2s, p2e, p2inc same as above but second parameter, say yp
% xp, yp, zp, psip, thep, phip = initial position/orientation of platform
% NOTE: for p1 and p2 (xp = 1, yp = 2, zp = 3, etc.)

% conversion factors and offset to actuator pairs

degrees = pi / 180.0;
offset = 120.0 * degrees;

rp = 0.5; % platform radius
gammap = 8.0 * degrees; % angle to platform end of first actuator
rb = 1.00; % base radius
gammab = 51.25 * degrees; % angle to base end of first actuator

% actuator end points

pp = [  [ rp * cos( gammap ), rp * sin( gammap ), 0 ];
   [ rp * cos( offset - gammap ), rp * sin( offset - gammap ), 0 ];
   [ rp * cos( offset + gammap ), rp * sin( offset + gammap ), 0 ];
   [ rp * cos( -offset - gammap ), rp * sin( -offset - gammap ), 0 ];
```

```
    [ rp * cos( -offset + gammap ), rp * sin( -offset + gammap ), 0 ];
    [ rp * cos( -gammap ), rp * sin( -gammap ), 0 ] ]';

bb = [  [ rb * cos( gammab ), rb * sin( gammab ), 0 ];
    [ rb * cos( offset - gammab ), rb * sin( offset - gammab ), 0 ];
    [ rb * cos( offset + gammab ), rb * sin( offset + gammab ), 0 ];
    [ rb * cos( -offset - gammab ), rb * sin( -offset - gammab ), 0 ];
    [ rb * cos( -offset + gammab ), rb * sin( -offset + gammab ), 0 ];
    [ rb * cos( -gammab ), rb * sin( -gammab ), 0 ] ]';

% if parameter 1 is one of the platform's orientation angles
% then convert degrees to radians

if ( p1 == 4 ) | ( p1 == 5 ) | ( p1 == 6 ),
    p1s = p1s * pi / 180;
    p1e = p1e * pi / 180;
    p1inc = p1inc * pi / 180;
end

% same as above for parameter 2

if ( p2 == 4 ) | ( p2 == 5 ) | ( p2 == 6 ),
    p2s = p2s * pi / 180;
    p2e = p2e * pi / 180;
    p2inc = p2inc * pi / 180;
end

% allocate space for 2D array of singular value ratios

ratios = ones( ( p1e - p1s ) / p1inc, ( p2e - p2s ) / p2inc );

% for each element in the array

l = 1;
for p1temp = p1s:p1inc:p1e,
    m = 1;
    for p2temp = p2s:p2inc:p2e,

        % set the appropriate parameters

        if p1 == 1,
            xp = p1temp;
        elseif p1 == 2,
            yp = p1temp;
        elseif p1 == 3,
            zp = p1temp;
        elseif p1 == 4,
            psip = p1temp;
        elseif p1 == 5,
```

```
            thep = p1temp;
         elseif p1 == 6,
            phip = p1temp;
         end

         if p2 == 1,
            xp = p2temp;
         elseif p2 == 2,
            yp = p2temp;
         elseif p2 == 3,
            zp = p2temp;
         elseif p2 == 4,
            psip = p2temp;
         elseif p2 == 5,
            thep = p2temp;
         elseif p2 == 6,
            phip = p2temp;
         end

         % compute platform jacobian

         j = jacobian( xp, yp, zp, phip, thep, psip, pp, bb );

         % singular value decomposition

         svs = svd( j );

         % save ratio of max to min SV ratios

         ratios( l, m ) = max( svs ) / min( svs );

         % limit maximum ratio because they get really big

         if ratios( l, m ) > 200,
            ratios( l, m ) = 200;
         end

         m = m + 1;
      end
      l = l + 1;
   end

   % set axis label names and units
   % NOTE: MATLAB has no way of specifying labels for "mesh" plots
   % so this is an adequate fix

   if p1 == 1,
      name1 = 'xp';
   elseif p1 == 2
      name1 = 'yp';
   elseif p1 == 3
```

```
   name1 = 'zp';
elseif p1 == 4
   name1 = 'psip';
elseif p1 == 5
   name1 = 'thep';
elseif p1 == 6
   name1 = 'phip';
end

if ( p1 >= 4 ) & ( p1 <= 6 )
   p1e = p1e * 180 / pi;
   units1 = ' deg';
else
   units1 = ' m';
end

if p2 == 1,
   name2 = 'xp';
elseif p2 == 2
   name2 = 'yp';
elseif p2 == 3
   name2 = 'zp';
elseif p2 == 4
   name2 = 'psip';
elseif p2 == 5
   name2 = 'thep';
elseif p2 == 6
   name2 = 'phip';
end

if ( p2 >= 4 ) & ( p2 <= 6 )
   p2e = p2e * 180 / pi;
   units2 = ' deg';
else
   units2 = ' m';
end

clg;

mesh( ratios );
title('Singular Value Ratios');
text( 0.75, 0.12, name1, 'sc' );
text( 0.25, 0.10, name2, 'sc' );
text( 0.1, 0.8, [ num2str( p1s ), ' < ', name1, ' < ', ...
      num2str( p1e ), units1 ], 'sc' );
text( 0.1, 0.75, [ num2str( p2s ), ' < ', name2, ' < ', ...
      num2str( p2e ), units2 ], 'sc' );
```

```
text( 0.1, 0.7, [ num2str( min( min( ratios ) ) ), ' < ratios < ', ...
      num2str( max( max( ratios ) ) ) ], 'sc' );
```

C.2 jacobian.m

```
function j = jacobian( x, y, z, phi, the, psi, pp, bb )

% File: jacobian.m
% Project: A Six Degrees-of-Freedom, Hydraulic, One Person Motion Simulator
% For: Dr. Tim Salcudean and Dr. Peter Lawrence
% By: Peter Drexel
% Copyright University of British, 1992

% compute platform's jacobian for "plotsingratios.m"

% cos and sin of platform orientation angles

cphi = cos( phi ); sphi = sin( phi );
cthe = cos( the ); sthe = sin( the );
cpsi = cos( psi ); spsi = sin( psi );

% orientation matrix

bRp = [ [ cphi*cthe, -sphi*cpsi+cphi*sthe*spsi, sphi*spsi+cphi*sthe*cpsi ];
   [ sphi*cthe, cphi*cpsi+sphi*sthe*spsi, -cphi*spsi+sphi*sthe*cpsi ];
   [ -sthe, cthe*spsi, cthe*cpsi ] ];

j = zeros( 6 );

for i = 1:6,
   bRpTimesppi = bRp * pp( :, i );
   bbiMinusbdp = bb( :, i ) - [ x, y, z ]';
   j( i, 1:3 ) = bRpTimesppi' - bbiMinusbdp';
   j( i, 4:6 ) = CrossProduct( bbiMinusbdp, bRpTimesppi )';
end
```

C.3 CrossProduct.m

```
function c = CrossProduct( a, b )

c = [ a( 2 )*b( 3 )-a( 3 )*b( 2 );
      a( 3 )*b( 1 )-a( 1 )*b( 3 );
      a( 1 )*b( 2 )-a( 2 )*b( 1 ) ];
```

Appendix D

Buckling Source Code

D.1 buckling.m

```
function buckling( minlen, maxlen, minforce )

% File: buckling.m
% Project: A Six Degrees-of-Freedom, Hydraulic, One Person Motion Simulator
% For: Dr. Tim Salcudean and Dr. Peter Lawrence
% By: Peter Drexel
% Copyright University of British Columbia, 1992

% plot buckling force versus column length using Euler's column formula
% for actuator bores of 2, 1.5 and 1 inch

% minlen = minimum column length to consider
% maxlen = maximum column length to consider
% minforce = minimum acceptable force (draws a threshold line)

% unit convesrion factors

inch_to_meter = 0.0254;    % 1 inch = 0.0254 meters

% set piston rod diameters

rodDia10 = 5/8 * inch_to_meter;
rodDia15 = 1 * inch_to_meter;
rodDia20 = 1.25 * inch_to_meter;

% moment of area

I10 = pi * rodDia10^4 / 64;
I15 = pi * rodDia15^4 / 64;
I20 = pi * rodDia20^4 / 64;

E = 200e9; % modulus of elasticity of steel

% set plot range

l = linspace( minlen, maxlen, 50 );

threshold = linspace( minforce, minforce, 50 );

fcr10 = pi^2 * E * I10 ./ l.^2;
```

```
fcr15 = pi^2 * E * I15 ./ l.^2;
fcr20 = pi^2 * E * I20 ./ l.^2;

plot( l, fcr10, ':', l, fcr15, '-', l, fcr20, '--', l, threshold, '-.' );
title( 'Critical Buckling Force Versus Column Length' );
xlabel( 'l (m)' );
ylabel( 'fcr (N)' );
text( l( length( l ) / 4 ), fcr10( length( l ) / 4 ), '1.0' );
text( l( length( l ) / 4 ), fcr15( length( l ) / 4 ), '1.5' );
text( l( length( l ) / 4 ), fcr20( length( l ) / 4 ), '2.0' );
text( l( length( l ) / 4 ), threshold( length( l ) / 4 ), '4000 N' );
```

Appendix E

Force Versus Flow Source Code

E.1 fvsq.m

```
function fvsq( qMax, deltaP, Ps, maxFlow, metricValve, fourway )

% File: fvsq.m
% Project: A Six Degrees-of-Freedom, Hydraulic, One Person Motion Simulator
% For: Dr. Tim Salcudean and Dr. Peter Lawrence
% By: Peter Drexel
% Copyright University of British, 1992

% plot the force versus flow/speed characteristics for 2, 1.5 and 1 inch bore
% cylinders and the valve specified in the input parameters

% qMax = flow through valve for a given pressure drop, deltaP, across BOTH
% valve ports
% deltaP = pressure drop across BOTH valve ports at a flow of qMax
% Ps = supply pressure
% metricValve: if =1 then qMax must be given in liter/min and deltaP
% must be given in Bar,
%  if =0 then qMax   "   "   "    " gallons/min " deltaP   "   "   "    " PSI
%  NOTE: Ps and maxFlow are always given in PSI and GPM respectively
% fourway: if =1 then the fluid into both ends of the cylinder is controlled
%          if =0 then the fluid into the blind end only is controlled

% unit convesrion factors

PSI_to_Pascal = 6895;      % 1 PSI = 6895 Pascal
bar_to_Pascal = 100e3;     % 1 Bar = 100000 Pascal
inch_to_meter = 0.0254;       % 1 inch = 0.0254 meters
min_to_sec = 60;    % 1 minute = 60 seconds
liter_to_m3 = 0.001;       % 1 L = 0.001 m^3
gal_to_m3 = 0.0038;    % 1 gal = 0.0038 m^3
LPM_to_m3perS = liter_to_m3 / min_to_sec;
GPM_to_m3perS = gal_to_m3 / min_to_sec;

% convert input to SI
```

```
if metricValve == 1,
   qMax_SI = qMax * LPM_to_m3perS;
   Ps_SI = Ps * PSI_to_Pascal;
   deltaP_SI = deltaP * bar_to_Pascal;
else
   qMax_SI = qMax * GPM_to_m3perS;
   Ps_SI = Ps * PSI_to_Pascal;
   deltaP_SI = deltaP * PSI_to_Pascal;
end

% set piston and rod diameters

pistonDia10 = 1 * inch_to_meter;
rodDia10 = 5/8 * inch_to_meter;

pistonDia15 = 1.5 * inch_to_meter;
rodDia15 = 1 * inch_to_meter;

pistonDia20 = 2 * inch_to_meter;
rodDia20 = 1.25 * inch_to_meter;

% piston and annulus areas

A10 = pi * pistonDia10^2 / 4.0;
a10 = A10 - pi * rodDia10^2 / 4.0;

A15 = pi * pistonDia15^2 / 4.0;
a15 = A15 - pi * rodDia15^2 / 4.0;

A20 = pi * pistonDia20^2 / 4.0;
a20 = A20 - pi * rodDia20^2 / 4.0;

% valve coefficient, divide by two because data sheet pressure
% drop is across both ports

Cv = qMax_SI / sqrt( deltaP_SI / 2 );

% force versus flow

% for positive flows into the blind end

qp = linspace( 0, maxFlow, 50 );

if fourway == 1,
   fp10 = ( Ps_SI - ( qp * GPM_to_m3perS / Cv ).^2 ) * A10 - ...
          ( qp * GPM_to_m3perS * a10 / Cv / A10 ).^2 * a10;
   fp15 = ( Ps_SI - ( qp * GPM_to_m3perS / Cv ).^2 ) * A15 - ...
          ( qp * GPM_to_m3perS * a15 / Cv / A15 ).^2 * a15;
   fp20 = ( Ps_SI - ( qp * GPM_to_m3perS / Cv ).^2 ) * A20 - ...
          ( qp * GPM_to_m3perS * a20 / Cv / A20 ).^2 * a20;
else
```

```
    fp10 = ( Ps_SI - ( qp * GPM_to_m3perS / Cv ).^2 ) * A10 - Ps_SI * a10;
    fp15 = ( Ps_SI - ( qp * GPM_to_m3perS / Cv ).^2 ) * A15 - Ps_SI * a15;
    fp20 = ( Ps_SI - ( qp * GPM_to_m3perS / Cv ).^2 ) * A20 - Ps_SI * a20;
end

% for negative flows out of the blind end

qn = -qp;

if fourway == 1,
    fn10 = ( qn * GPM_to_m3perS / Cv ).^2 * A10 - ( Ps_SI - ...
           ( qn * GPM_to_m3perS * a10 / Cv / A10 ).^2 ) * a10;
    fn15 = ( qn * GPM_to_m3perS / Cv ).^2 * A15 - ( Ps_SI - ...
           ( qn * GPM_to_m3perS * a15 / Cv / A15 ).^2 ) * a15;
    fn20 = ( qn * GPM_to_m3perS / Cv ).^2 * A20 - ( Ps_SI - ...
           ( qn * GPM_to_m3perS * a20 / Cv / A20 ).^2 ) * a20;
else
    fn10 = ( qn * GPM_to_m3perS / Cv ).^2 * A10 - Ps_SI * a10;
    fn15 = ( qn * GPM_to_m3perS / Cv ).^2 * A15 - Ps_SI * a15;
    fn20 = ( qn * GPM_to_m3perS / Cv ).^2 * A20 - Ps_SI * a20;
end

% set unit label for plots

if metricValve == 1,
    flowunits = 'LPM';
    pressunits = 'Bar';
else,
    flowunits = 'GPM';
    pressunits = 'PSI';
end

% set connection label for plots

if fourway == 1,
    connection = '4-Way';
else
    connection = '3-Way';
end

% plot force versus flow

plot( qp, fp10, ':', qn, fn10, ':', ...
    qp, fp15, '-', qn, fn15, '-', ...
    qp, fp20, '--', qn, fn20, '--' );
grid;
title( [ 'Force-Flow Characteristic (', num2str(qMax), ' ', flowunits, ' @ ', ...
    num2str(deltaP), ' ', pressunits, ', ', 'Ps = ', num2str(Ps), ...
    ' PSI) ', connection ] );
```

```
xlabel( 'q (GPM)' );
ylabel( 'f (N)' );
text( qn(1), fn10(1), '1.0' );
text( qn(1), fn15(1), '1.5' );
text( qn(1), fn20(1), '2.0' );

% wait for "return"

keyboard;

% plot force versus speed

plot( qp * GPM_to_m3perS / A10, fp10, ':', qn * GPM_to_m3perS / A10, fn10, ...
   ':', qp * GPM_to_m3perS / A15, fp15, '-', qn * GPM_to_m3perS / A15, ...
   fn15, '-', qp * GPM_to_m3perS / A20, fp20, '--', ...
   qn * GPM_to_m3perS / A20, fn20, '--' );
grid;
title( [ 'Force-Speed Characteristic (', num2str(qMax), ' ', flowunits, ...
   ' @ ', num2str(deltaP), ' ', pressunits, ', ', 'Ps = ', ...
   num2str(Ps), ' PSI) ', connection ] );
xlabel( 'dl/dt (m/s)' );
ylabel( 'f (N)' );
text( qn(1) * GPM_to_m3perS / A10, fn10(1), '1.0' );
text( qn(1) * GPM_to_m3perS / A15, fn15(1), '1.5' );
text( qn(1) * GPM_to_m3perS / A20, fn20(1), '2.0' );
```