MODELING, SIMULATION, AND CONTROL OF A STEWART PLATFORM

By

Daming Li

B. Eng., Tsinghua University, China, 1991

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

November 1996

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
Vancouver, Canada

Date April 29, 1997

Abstract

This thesis describes the modeling, simulation, and control of an inverted, ceiling-mounted Stewart platform, which is designed to be a motion simulator. This hydraulically actuated Stewart platform is capable of providing 10 m/s^2, 400 $degree/s^2$ accelerations and 1 m/s, 30 $degree/s$ speeds to a 250 kg payload.

The issues of modeling and control of such a platform are addressed here. The inverse kinematics and forward kinematics are studied first. The platform rigid-body dynamics are derived based on the virtual work principle and then combined with the actuator dynamics to simulate the response of the Stewart platform given a pre-planned motion path. Design and implementation of the link-space controller are discussed and also validated using experimental data. Cartesian-space controllers are also addressed. Motion drive algorithms are finally addressed to complete the system's function as a motion simulator.

When the controller is well tuned, the bandwidth of the system can reach about 9Hz along the vertical axis for a payload of about 140 kg.

Table of Contents

List of Tables

List of Figures

Acknowledgments

My greatest debt of gratitude goes to my supervisor, Dr. Tim Salcudean, who has introduced and guided me into this exciting project, and provided sound advice and invaluable critical feedback. Without his help and encouragement, this thesis could not have been written.

Thanks are also to Alison Taylor, Simon Bachmann, and Icarus Chau, research engineers in the Robotics and Control Laboratory at UBC, for their contributions to the project.

I would also thank professor Peter Lawrenece, Professor Farrokh Sassani, and my colleagues in the lab for their advice and discussions.

I would like to acknowledge that this project was supported by the Canadian IRIS/PRECARN Network of Centers of Excellence.

I dedicate this work to my parents.

Chapter 1

Introduction

The Stewart platform is a type of parallel manipulator, which consists of a mobile platform and a stationary base, connected to each other by six linear actuators (legs). A photo of the Stewart platform developed in the Robotics and Control Laboratory at UBC is presented in Figure 1.1. Since proposed by D. Stewart [1] in 1965, the platform has been used in many applications such as aircraft simulators, assembly workstations, mills, and robot wrists; it has also attracted considerable attention of researchers in different areas.

In this project, an inverted, ceiling-mounted Stewart platform design was employed to build a one-person motion simulator. This Stewart platform is used to simulate the motion of heavy hydraulic equipment for the purpose of human factors and teleoperation work for the forest industry [2]. In addition, it can also be used to study virtual reality, motion algorithm, and control of parallel manipulators. The initial design of this project was described in [3]. It is our goal to study the kinematics and dynamics of the Stewart platform in details and successfully accomplish the control basing on a effective understanding of the actuator dynamics. With a successful control, we hope to achieve a good high-frequency response and high stability margin, so that this Stewart platform can have similar performance to that of commercial systems costing much more.

In this first chapter we present a literature review of current research on Stewart platforms, and then we give a general description of the design and performance of our Stewart platform.

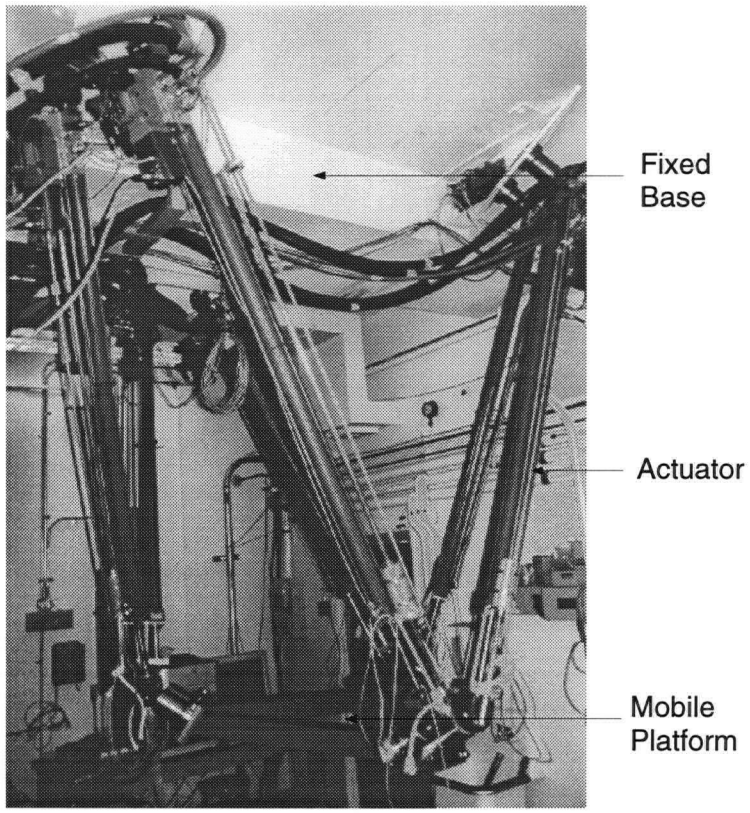

Figure 1.1: Stewart platform motion simulator

1.1 Literature Review

To the author's best knowledge, the 6 degrees-of-freedom (DOF) parallel manipulator was first described and built by Gough [4, 5] as part of a tire-testing machine. Stewart proposed a similar parallel architecture for use as a flight simulator, and 6 DOF parallel manipulators are generally called Stewart platforms. Several companies, including Canadian Aerospace Electronics (CAE) Ltd. and Moog Inc., have been using Stewart's design for flight simulators and entertaining motion simulators. However, there was no significant progress on the research of Stewart platform and other types of parallel manipulators until 1980's. A number of papers have been published on the kinematics, dynamics, and control of parallel manipulators since then.

The inverse kinematic equations (to determine the leg lengths given the position and orientation of the mobile platform) of the Stewart platform are relatively easy to derive. There are several different approachs to this problem, according to the different selections of the coordinate systems. Nguyen *et al* [6], Cleary *et al* [7], Fichter [8], Do and Yang [9, 10] have all derived the inverse kinematics based on their particular coordinate systems.

Hence, the reachable workspace of the Stewart platform was studied by many researchers. The workspace of the platform was mapped out through methods based on a complete discretization of the Cartesian-space [7, 9, 11]. A geometric approach for this problem was proposed by Gosselin *et al* [12]. This approach was extended by Merlet [13] to take into account all the constraints (i.e., the legs lengths range, mechanical limits on the joints, and legs interference) limiting the workspace. Merlet also showed how to perform a trajectory verification to check if the desired trajectory was inside the workspace [13]. However, if we consider the problem of singularities, which will be discussed later, the actual allowable workspace may be even smaller.

When the Jacobian matrix of a manipulator loses ranks, the manipulator is said to be in a singular configuration. To determine these singular configurations, the classical method is to monitor the condition number of the Jacobian matrix. In contrast, several researchers tried to locate the singular configurations geometrically. Hunt described a singular configuration, in which case all the six lines associated to the robot links intersected one line [14]. Fichter [11] described another singular configuration which is obtained by rotating the mobile platform around the vertical axis by an angle of $\pm\frac{\pi}{2}$. Finally, this problem was successfully solved by Merlet in 1989 [15]. A new method based on Grassmann line geometry was proposed and it was shown that a singular configuration is obtained when the variety spanned by the lines associated to the robot links has a rank less than 6. Then, a set of geometric rules were used to establish the constraints on the position and orientation parameters that must be satisfied to obtain the various singular configurations [15].

The forward kinematic equations (to determine the position and orientation of the mobile platform given the legs lengths) of the Stewart platform have no known closed-form solution.

Several researchers found that solving the forward kinematics problem is equivalent to solving a $16th$-order polynomial in one variable [16, 17, 18, 19, 20]. Merlet also presented a geometric proof to show that the number of assembly modes is at most 16 [20]. The order of the polynomial can be reduced to 8 if the mobile platform is restricted to be either above or below the base. Unfortunately, because the numerical resolution of the equivalent polynomials requires significant computing time and yields many solutions, it seems that this approach is neither useful for finding the analytical solution of forward kinematics problem nor useful for the real-time control. In addition, this method can only be applied to parallel manipulators with triangular mobile platforms, and can not be extended to the more general case of parallel manipulators in which the mobile platform and the stationary base are both hexagons or other shapes.

Cleary was able to avoid the forward kinematics problem of the Stewart platform by mounting a passive serial linkage between the centers of mobile platform and stationary base and then monitoring the position and orientation of their prototype platform through sensors on the passive serial linkage [7]. But the achievable accuracy of this method is quite questionable.

Another method of computing the forward kinematics is the iterative numerical method. Nguyen *et al* applied Powell's Direction Set Method to solve the nonlinear equations [6], while Dieudonne *et al* used Newton's method to converge to a solution [21]. Newton's method makes use of the platform's Jacobian matrix to update the estimations of the platform's Cartesian-space parameters, and the result can converge to the required accuracy in a couple of iterations when a good starting point is given. We will apply Newton's method to solve the forward kinematics problem in this thesis.

Do and Yang used Newton-Euler equations of motion to solve the inverse dynamics of the Stewart platform [10]. They also ran a simulation to compute the required actuating forces for given trajectories. Kai Liu *et al* used Lagrangian approach to derive the dynamic equations of the Stewart platform in Cartesian-space and then used Jacobian transformation to obtain the actuating forces in link-space [22]. Zhang and Song proposed a more efficient method for

manipulator inverse dynamics based on the virtual work principle [23]. We will apply the virtual work principle method to derive the inverse dynamics of the Stewart platform in our coordinate system.

Although the inverse dynamics problem for the Stewart platform has been solved, it has not been used practically in the Cartesian-space controller design yet. A standard computed-torque control law in Cartesian-space was briefly discussed by Kai Liu *et al* in [22], but no further details were given. No other published paper on the Cartesian-space control of the Stewart platform has been found.

In order to produce accurate motion cues, the motion simulator must be able to reproduce accelerations as much as possible within its limited motion range. This is done by applying the motion drive algorithm, commonly known as wash-out filter, to the original trajectory. Initially the motion drive algorithms employed linear filter elements with fixed parameters. In an attempt to overcome the limitations of linear formulations, nonlinear adaptive wash-out algorithms were developed. The performances of different motion drive algorithms were investigated by Reid and Nahon *et al* [24].

1.2 System Description

The geometry design of our motion simulator follows a standard 6-6 Stewart platform (both the mobile platform and the stationary base are hexagons), however, this simulator is inverted and suspended from the ceiling. The electrohydraulic actuators are under tension, so they are less likely to have buckling problems, allowing us to use narrower actuators. This inverted design also provides operators the easy entrance and exit, without the need for an access ramp. However, this inverted design makes the maintenance of ceiling-mounted hydraulic equipment more difficult, and there is a requirement for the ceiling's height and rigidity. The simulator uses six 1.5 inch bore, 54 inch stroke electrohydraulic actuators controlled by three-stage, proportional valves. 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. Each actuator can

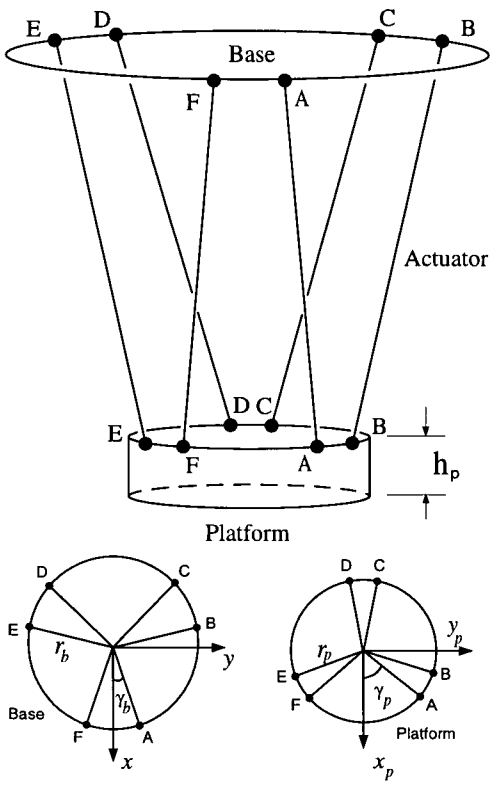

Figure 1.2: Inverted simulator geometry.

provide as much as 4000 N force and 1.5 m/s velocity. The position sensor of each actuator is a Temposonic magnetostrictive wire transducer.

The geometry of the simulator is shown in Figure 1.2. While the dimensions of the simulator are shown in Table 1.1. The notations used here are the same as those used in [3].

Table 1.1: Simulator Dimensions.

Platform Radius, r_p	0.668 m
Platform Actuator Angle, γ_p	53.45°
Base Radius, r_b	1.133 m
Base Actuator Angle, γ_b	7.75°
Platform Nominal Height, h_p	0.203 m

As will be shown later, the simulator has the performance specified in Table 1.2.

Table 1.2: Simulator Performance.

	Displacement	Velocity	Acceleration
x-axis	±0.2 m	±1 m/s	±10 m/s^2
y-axis	±0.2 m	±1 m/s	±10 m/s^2
z-axis	±0.2 m	±1 m/s	±10 m/s^2
Roll	±20°	±30°/s	±400°/s^2
Pitch	±20°	±30°/s	±400°/s^2
Yaw	±20°	±30°/s	±400°/s^2

The remainder of this thesis is organized as follows. Chapter 2 discusses the kinematics of the Stewart platform, including the inverse kinematics and the forward kinematics. The inverse kinematics part is the same as derived in [3]. The forward kinemetics part revises the error in [3]. The singularity issues have been briefly discussed in [3] and are further discussed in this chapter; the issue of workspace is addressed as well. Chapter 3 gives the equations of the rigid body dynamics, where the dynamics of legs are included to give a complete model. A model of the electrohydraulic actuator is given in Chapter 4, where the valve dynamics are also included to give a more precise model. This model is validated using experimental data. The control approach, its implementation, and experimental results are discussed in Chapter 5. Path planning issues are discussed in this chapter as well. The software simulation based on the model derived before is introduced in Chapter 6, the simulation code was initially written by P. Drexel [3] and was further modified by the author. Several case studies are performed to further compare software simulation results and experimental results. In Chapter 7, conclusion and plans for future works are presented.

Chapter 2

Platform Kinematics

In this chapter we derive the inverse kinematic equations of the Stewart platform, and then give the definition and derivation of the Jacobian matrix based on our coordinate frames. Singularities of the platform and joints are also discussed. At last, the platform's forward kinematics are solved numerically, and the computing time is discussed. The notations used in this chapter and the following chapters are mainly taken from P. Drexel's thesis, for the purpose of consistency.

2.1 Inverse Kinematics

In this section we will use vector algebra to attain closed-form equations of the inverse kinematics of the Stewart platform, which map the position and orientation of the mobile platform to the lengths of the six actuators. This part has been done similarly in [3] and some other references. We just follow their derivations here.

Figure 2.3 shows that two coordinate frames $\{\mathbf{P}\}$ and $\{\mathbf{B}\}$ are assigned to the mobile platform and the stationary base, respectively. The origin of the frame $\{\mathbf{P}\}$ is located at the centroid P of the mobile platform, the z_p axis is pointing upward, and the x_p axis is perpendicular to the line connecting the centers of the two platform attached joints P_A and P_F. Similarly, frame $\{\mathbf{B}\}$ has its origin at the centroid B of the base, the z axis is pointing upward, and the x axis is perpendicular to the line connecting the centers of the two base attached joints B_A and B_F. The configuration of the platform is specified by the position of the origin of frame $\{\mathbf{P}\}$ with respect to frame $\{\mathbf{B}\}$ and the orientation of frame $\{\mathbf{P}\}$ with respect to frame $\{\mathbf{B}\}$. The position of frame $\{\mathbf{P}\}$ is represented by vector ${}^b\mathbf{d}_p = [x\ y\ z]^T$, which contains the Cartesian

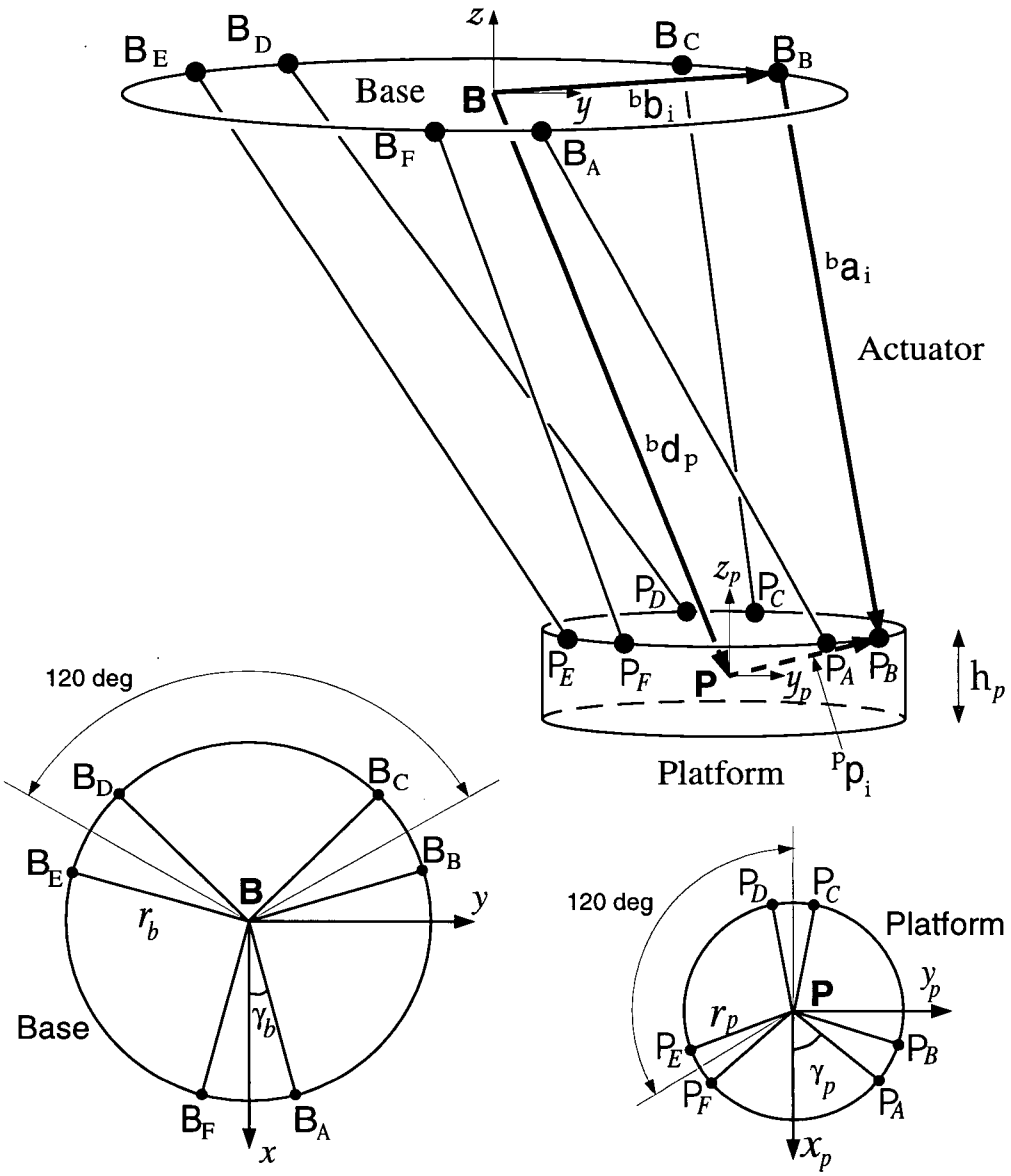

Figure 2.3: Vector definitions for inverse kinematics.

coordinates x, y, z of the origin of frame $\{\mathbf{P}\}$ with respect to frame $\{\mathbf{B}\}$. The orientation of frame $\{\mathbf{P}\}$ with respect to frame $\{\mathbf{B}\}$ can be described by the orientation matrix ${}^{b}\mathbf{R}_{p}$, whose columns are the coordinates of basis axes of frame $\{\mathbf{P}\}$ in frame $\{\mathbf{B}\}$. We define the position

of the center of i^{th} platform attached joint P_i with respect to frame $\{\mathbf{P}\}$ to be ${}^{p}\mathbf{p}_i$, which is a fixed vector. Similarly, we define the position of the center of i^{th} base attached joint B_i with respect to frame $\{\mathbf{B}\}$ to be ${}^{b}\mathbf{b}_i$, which is also a fixed vector. Then, the i^{th} actuator vector ${}^{b}\mathbf{a}_i$, which is the vector from the center of i^{th} base attached joint B_i to the center of i^{th} platform attached joint P_i, can be expressed as,

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

Since the centers of joints are arranged in pairs at 120 degree intervals around a circle, as shown in Figure 2.3, we then have,

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

and

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

where r_p is the radius of the mobile platform circle, r_b is the radius of the base circle, γ_{pi} and γ_{bi} are defined for each actuator as shown in Table 2.3, and h_p is the nominal height of the platform. Note that (2.2) assumes that centroid P of the platform is vertically centered and that centers of the joints are at the top of the platform.

Table 2.3: Platform and Base Actuator End Point Angles.

i	γ_{pi}	γ_{bi}
A	$\gamma_p = 53.45°$	$\gamma_b = 7.75°$
B	$120° - \gamma_p$	$120° - \gamma_b$
C	$120° + \gamma_p$	$120° + \gamma_b$
D	$-120° - \gamma_p$	$-120° - \gamma_b$
E	$-120° + \gamma_p$	$-120° + \gamma_b$
F	$-\gamma_p$	$-\gamma_b$

The platform's orientation matrix, ${}^{b}\mathbf{R}_p$, is defined using roll-pitch-yaw angles ϕ_p, θ_p, and ψ_p. We specify the order of rotation as $x - y - z$: first yaw about the x axis through an angle ψ_p, then pitch about the y axis through an angle θ_p, and finally roll about the z axis

through an angle ϕ_p. Since the successive rotations are relative to the fixed frame, the resulted transformation matrix is given by,

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

where $\mathrm{c}\psi_p = \cos\psi_p$, $\mathrm{s}\psi_p = \sin\psi_p$, etc. We choose roll-pitch-yaw angles because they can easily reflect the platform's actual physical orientation and make things easier when we specify trajectories in the Cartesian-space.

The length of actuator i can be easily obtained from (2.1),

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

For each actuator, $i = A \ldots F$, (2.5) expresses its length, given the platform's position and orientation.

2.2 Platform Jacobian

The velocity kinematics of parallel manipulators are also given by Jacobian transformations, as in the case of serial manipulators. However, unlike those of serial manipulators, the Jacobian matrices of parallel manipulators are generally defined as transformations from the velocities of platforms in the Cartesian-space to the velocities of actuators in the link-space. We define the Jacobian matrix of Stewart platform 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, ${}^b\mathbf{v}_p = {}^b\dot{\mathbf{d}}_p$ is the platform translational velocity and ${}^b\omega_p$ is the platform 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 roll-pitch-yaw angle rotation matrix ${}^b\mathbf{R}_p$ defined above

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

Each row of the Jacobian matrix corresponds to one of the platform's six actuators. We can obtain the i^{th} row of the Jacobian matrix by differentiating both sides of (2.5), and substituting (2.1) into the result,

$$ \dot{l}_i = \frac{1}{2} \frac{2\, {}^b\mathbf{a}_i{}^T {}^b\dot{\mathbf{a}}_i}{\sqrt{{}^b\mathbf{a}_i{}^T {}^b\mathbf{a}_i}} = \frac{{}^b\mathbf{a}_i{}^T {}^b\dot{\mathbf{a}}_i}{l_i} \tag{2.9} $$

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

$$ = \frac{\left({}^b\mathbf{R}_p{}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i\right)^T ({}^b\mathbf{v}_p) + \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i\right)^T \left({}^b\omega_p \times ({}^b\mathbf{R}_p{}^p\mathbf{p}_i)\right)}{l_i} \tag{2.11} $$

$$ = \frac{\left({}^b\mathbf{R}_p{}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i\right)^T ({}^b\mathbf{v}_p) + \left(({}^b\mathbf{R}_p{}^p\mathbf{p}_i) \times ({}^b\mathbf{R}_p{}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i)\right)^T ({}^b\omega_p)}{l_i} \tag{2.12} $$

$$ = \frac{1}{l_i} \begin{bmatrix} \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i\right)^T & \left(({}^b\mathbf{R}_p{}^p\mathbf{p}_i) \times ({}^b\mathbf{d}_p - {}^b\mathbf{b}_i)\right)^T \end{bmatrix} \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \tag{2.13} $$

where ${}^b\dot{\mathbf{R}}_p = {}^b\omega_p \times {}^b\mathbf{R}_p$ and $\mathbf{a} \bullet (\mathbf{b} \times \mathbf{c}) = (\mathbf{c} \times \mathbf{a}) \bullet \mathbf{b}$ have been applied.

Then, we can get the Jacobian matrix by putting together the velocities of six actuators,

$$\dot{\mathbf{l}} = \begin{bmatrix} \dot{l}_A \\ \dot{l}_B \\ \dot{l}_C \\ \dot{l}_D \\ \dot{l}_E \\ \dot{l}_F \end{bmatrix} = \begin{bmatrix} \left({}^b\mathbf{R}_p{}^p\mathbf{p}_A + {}^b\mathbf{d}_p - {}^b\mathbf{b}_A\right)^T & \left(({}^b\mathbf{R}_p{}^p\mathbf{p}_A) \times ({}^b\mathbf{d}_p - {}^b\mathbf{b}_A)\right)^T \\ \left({}^b\mathbf{R}_p{}^p\mathbf{p}_B + {}^b\mathbf{d}_p - {}^b\mathbf{b}_B\right)^T & \left(({}^b\mathbf{R}_p{}^p\mathbf{p}_B) \times ({}^b\mathbf{d}_p - {}^b\mathbf{b}_B)\right)^T \\ \vdots & \vdots \\ \left({}^b\mathbf{R}_p{}^p\mathbf{p}_F + {}^b\mathbf{d}_p - {}^b\mathbf{b}_F\right)^T & \left(({}^b\mathbf{R}_p{}^p\mathbf{p}_F) \times ({}^b\mathbf{d}_p - {}^b\mathbf{b}_F)\right)^T \end{bmatrix} \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \tag{2.14}$$

or,

$$\dot{\mathbf{l}} = \mathbf{J}\left({}^b\mathbf{d}_p, {}^b\mathbf{R}_p\right) \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \tag{2.15}$$

where the Jacobian matrix depends on the platform's position ${}^b\mathbf{d}_p$ and orientation ${}^b\mathbf{R}_p$. This result is the same as in [3].

2.3 Workspace and Singularities

In this section we discuss the reachable workspace, mechanical constraints, and singularities in our system. We obtain a reachable workspace where the system is free of mechanical constraints and singularities that can be used for control system development.

2.3.1 Reachable workspace

Because of the length limits of the actuators, the Stewart platform has a limited reachable workspace. The boundary of this workspace is hard to describe, because it has 6 dimensions (x, y, z, ψ_p, θ_p, ϕ_p). The positioning workspace (i.e., the region of the three-dimensional Cartesian-space that can be attained by a manipulator with a given orientation) has been described through methods based on complete discretization of the Cartesian-space [7, 9, 11]. For a given orientation, we can calculate the inverse kinematics of the Stewart platform at a certain position, and check if the resulted actuator lengths are within the limits so as to determine if that position is inside the reachable positioning workspace. Because the workspace

is continuous, we can map it out by discretizing the Cartesian-space by small enough steps. Figure 2.4 shows the positioning workspace of our Stewart platform when $\psi_p = \theta_p = \phi_p = 0°$. But the actual reachable workspace is much smaller because we have to allow rotational

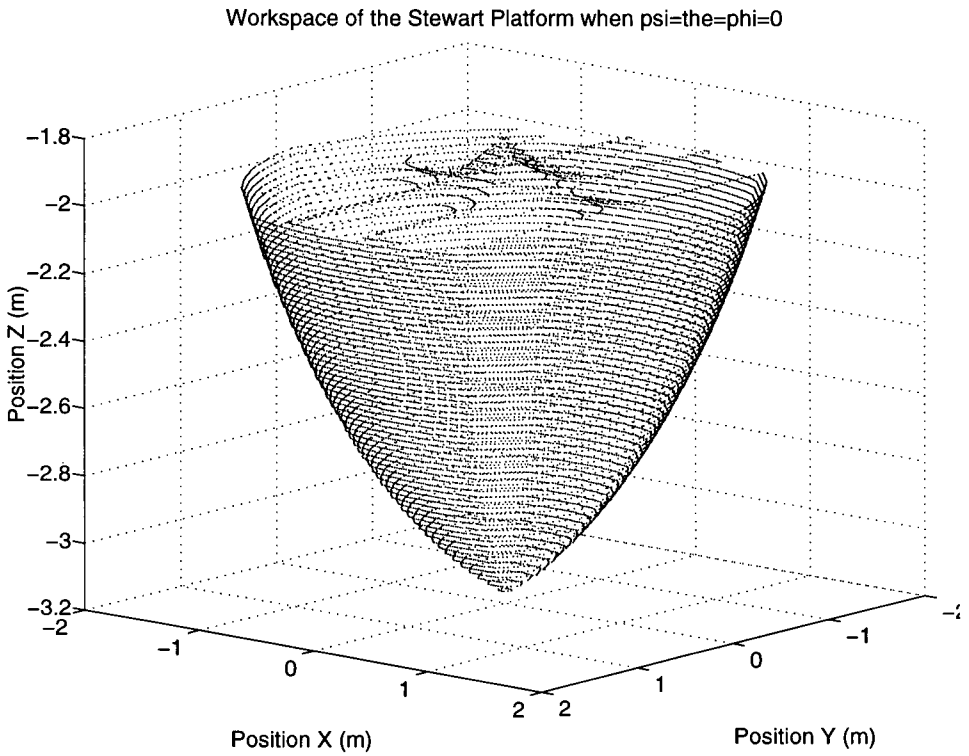

Figure 2.4: Translational workspace of the Stewart platform when $\psi_p = \theta_p = \phi_p = 0°$.

movements. To search the boundary of the workspace, we first discretize the Cartesian-space by small enough steps and then we rotate the Stewart platform at each position to check if the actuator lengths derived from the inverse kinematics are still within the limits. The workspace becomes smaller when we allow larger rotational movements. Figure 2.5 shows the actual workspace of our Stewart platform when we set a range limit of $-20°$ to $20°$ for ψ_p, θ_p, and ϕ_p. For the purpose of a general usage, we only considered the limits of actuator lengths in the above definition of the reachable workspace. But the motion range of our system is also limited by other mechanical constraints.

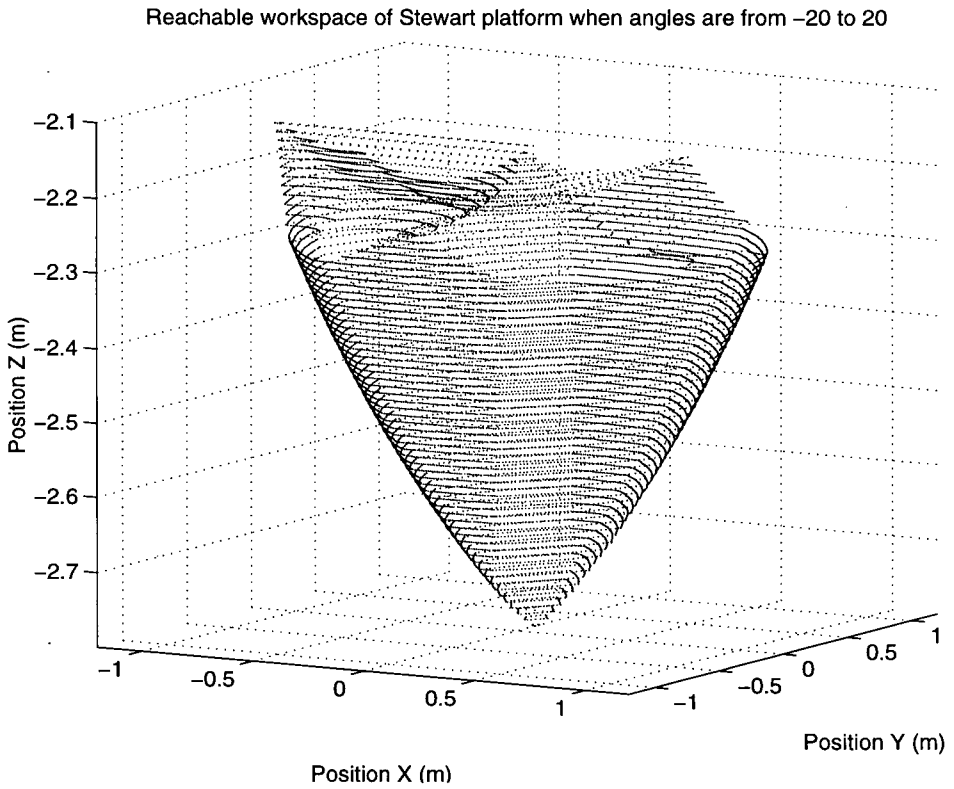

Figure 2.5: Translational workspace of the Stewart platform with arbitrary angles from $-20°$ to $20°$

2.3.2 Mechanical constraints

Our Stewart platform is ceiling mounted, so its motion along the z axis is limited by the height of the room. Another constraint is the interference between actuators and the chair top, which is used to protect the operator. We have to consider these two types of constraints when we search the boundary of the safe workspace. Figure 2.6 shows the projection of the safe workspace on the $y - z$ plane when there is no rotation. And Figure 2.7 shows the projection of the safe workspace on the $y - z$ plane when ψ_p, θ_p, and ϕ_p are from $-20°$ to $20°$. We can see that the safe workspace becomes much smaller when we allow rotations. We can achieve larger displacement and larger rotation in a pre-planned trajectory, if the displacement and the rotation are combined such that the pre-planned trajectory does not violate the mechanical

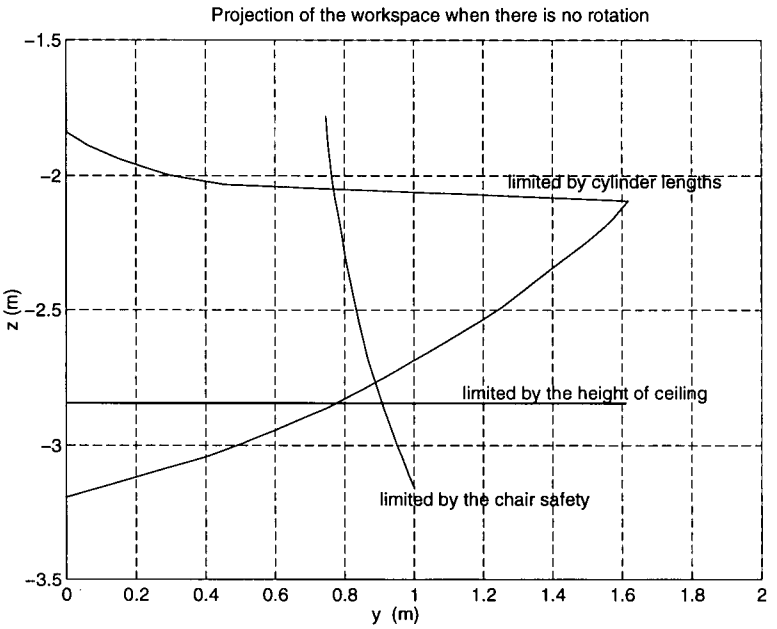

Figure 2.6: Projection of the safe workspace on the $y - z$ plane when there is no rotation.

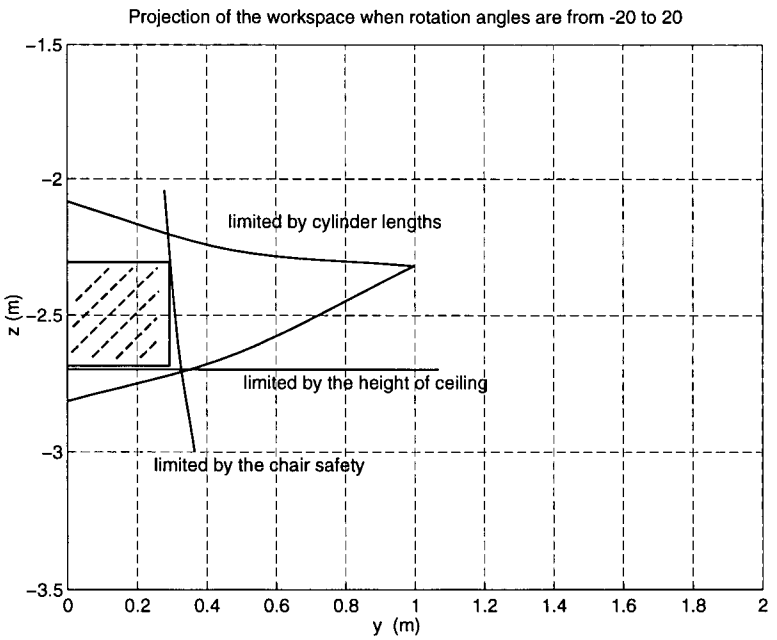

Figure 2.7: Projection of the safe workspace when ψ_p, θ_p, and ϕ_p are from $-20°$ to $20°$.

constraints. But when the platform is manipulated by a joy-stick, we would like to set the

translational and rotational limits to prevent mechanical damage We can define a cylindrical volume where the system is reachable and free of mechanical constraints. However, we have to check if there is any singularity inside that cylindrical volume.

2.3.3 Singularities

The singularities in the design of our Stewart platform include platform singularity and U-joints singularities.

The platform singularity has been analyzed successfully by Merlet through Grassmann geometry [15]. The idea is that *a parallel manipulator will be in a singular configuration if, and only if, there is a subset spanned by n of its lines which has a rank less than n* . For example, subset of rank 1 is a line in the 3D space; subsets of rank 2 are either a pair of skew lines in the 3D space or lines lying in a 2D plane and passing through the same point on that plane. For a 2D parallel manipulator in Figure 2.8, the singular configuration is obtained when

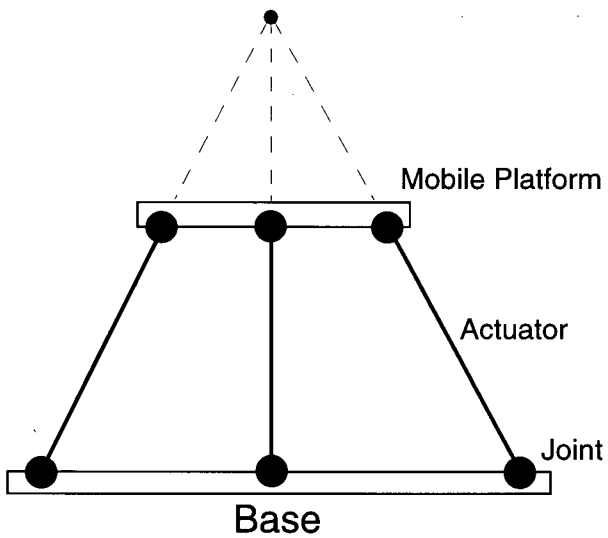

Figure 2.8: Singular configuration for the 2D parallel manipulator

the three lines (actuators) intersect. If the mobile platform and the base are symmetric as shown in Figure 2.8, we get a singular configuration when the mobile platform and the base

are parallel. When a parallel manipulator reaches a singular configuration, it gains one or more uncontrollable degrees of freedom and will rotate and/or translate without a change in the actuator lengths. Merlet uses a set of geometric rules to establish the constraints on the position and orientation parameters that must be satisfied to obtain the various singular configurations. However, the calculation of such geometric rules is more complicated than that of the condition number of platform Jacobian, which is used in the classical method to determine these singular configurations numerically. By plotting the ratios of the largest to smallest condition number of platform Jacobian while varying two of the platform's six position and orientation parameters, we can obtain a graphical description of the platform's singular configurations. For example, Figure 2.9 shows a plot of the condition number ratios versus varying pitch angle θ_p and roll angle ϕ_p for the platform at its nominal position ($x = 0$ m, $y = 0$ m, $z = -2.5$ m and $\psi_p = 0°$). (See Appendix D.3 for MATLAB source code). The "singularity boundary" is actually six-

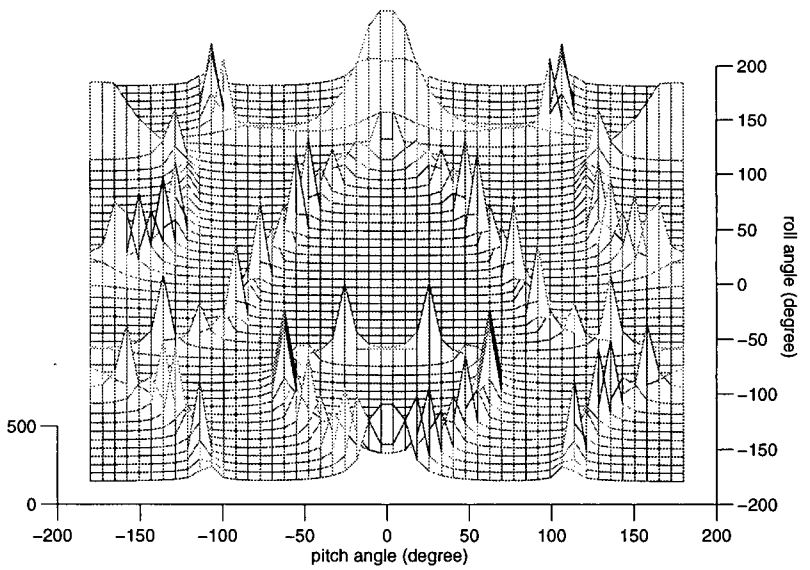

Figure 2.9: Condition number of the platform Jacobian versus pitch & roll angle

dimensional, and thus hard to describe. However, by plotting condition number ratios versus two parameters while varying the other four parameters up to their limits, we can confirm that the cylindrical volume defined in the last section is free of singularity. So, we don't need to

monitor the condition number of the Jacobian as long as the motion of the Stewart platform is within that cylindrical volume.

We use U-joints in the design of our Stewart platform, because U-joints provide a large range of motion and are relatively easy to manufacture [3]. However, there are singularities when the base of the U-joint and the attached actuator share a common axis. To prevent singularities, we should monitor the angle between axis of the U-joint base and axis of the attached cylinder, and prepare to stop the platform before it enters a singular configuration.

We use the method of discretization again to study the singularities of the U-joints; a small enough interval is used to make sure that any possible singular point will be tested.

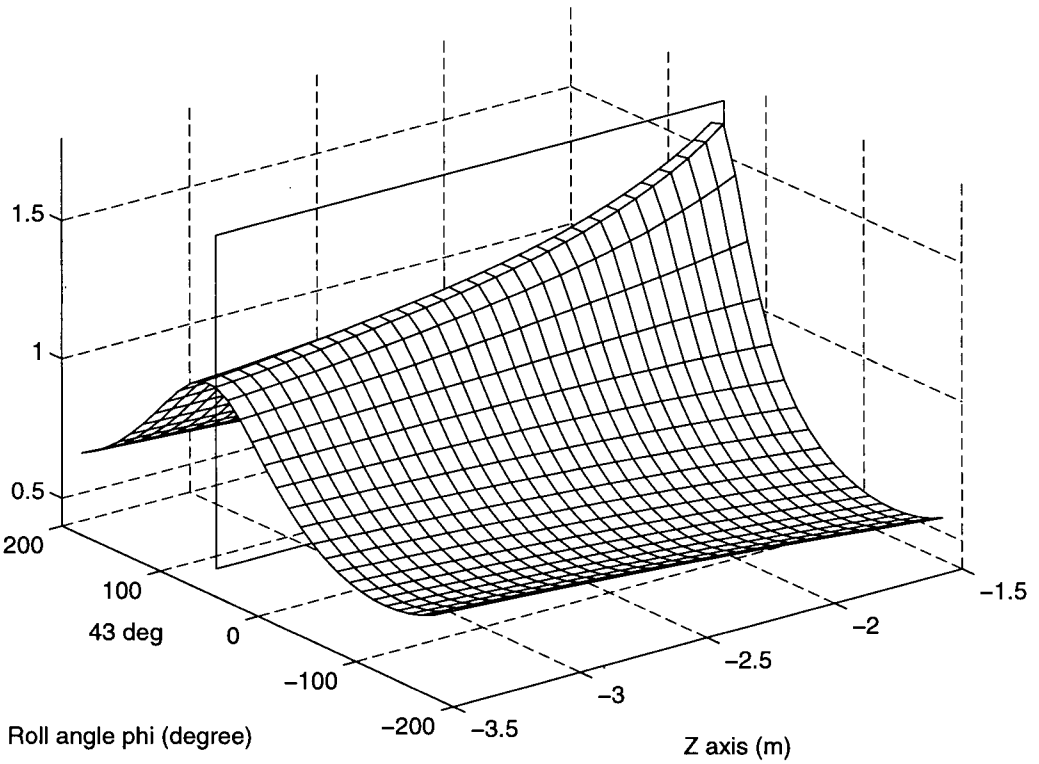

Figure 2.10: Inverse of U-joint angle versus roll angle and z

We come to the conclusion that there is no singularity for the U-joints attached to the base

within the whole possible workspace, and that there is no singularity for the U-joints attached to the mobile platform within the cylindrical volume we defined before. So, we do not need to monitor the angle of the U-joints as long as the motion of the Stewart platform is within that cylindrical volume. A plot of the inverse of U-joint angle of joint P_A versus roll angle and z is shown in Figure 2.10.

Thus, we obtain a cylindrical workspace that is safe and reachable. This workspace is described by, $-2.7m < Z_p < -2.3m$, $-0.2m < \sqrt{X_p^2 + Y_p^2} < 0.2m$, and $-20° < \psi_p, \theta_p, \phi_p < 20°$ when the chair is mounted;

2.4 Forward Kinematics

In this section we will study the forward kinematics of the Stewart platform, which map lengths of the six actuator to the position and orientation of the mobile platform.

As mentioned before, there is no known closed-form solution to the forward kinematics of the Stewart platform. Dieudonne *et al* used Newton's method to solve the forward kinematics problem numerically [21]. The mathematics of this method is relatively straightforward, and the convergence is quadratic. It makes use of the platform's Jacobian matrix to update the estimations of the platform's Cartesian-space parameters, and the result can converge to the required accuracy in a couple of iterations when a good starting point is given. Here we use Newton's method for multiple equations and variables to calculate the platform's forward kinematics iteratively.

For multiple equations and variables, Newton's method is

$$\mathbf{X}_{j+1} = \mathbf{X}_j - \left(\frac{\partial \mathbf{g}(\mathbf{X}_j)}{\partial \mathbf{X}_j}\right)^{-1} \mathbf{g}(\mathbf{X}_j) \tag{2.16}$$

where $\mathbf{X}$ is a vector of the variables we wish to estimate, $\mathbf{g}$ is a vector function which approaches zero as the estimation of $\mathbf{X}$ improves and j is the iteration count. For the Stewart platform described in this thesis, we select

$$\mathbf{X}^T = \begin{bmatrix} {}^b\mathbf{d}_p^T & \psi_p & \theta_p & \phi_p \end{bmatrix} \tag{2.17}$$

and

$$\mathbf{g}(\mathbf{X}) = \begin{bmatrix} \| \left({}^{b}\mathbf{R}_p \left(\psi_p, \theta_p, \phi_p\right) {}^{p}\mathbf{p}_A + {}^{b}\mathbf{d}_p - {}^{b}\mathbf{b}_A \right) \| - l_A \\ \vdots \\ \| \left({}^{b}\mathbf{R}_p \left(\psi_p, \theta_p, \phi_p\right) {}^{p}\mathbf{p}_F + {}^{b}\mathbf{d}_p - {}^{b}\mathbf{b}_F \right) \| - l_F \end{bmatrix} = \begin{bmatrix} \|{}^{b}\mathbf{a}_A\| - l_A \\ \vdots \\ \|{}^{b}\mathbf{a}_F\| - l_F \end{bmatrix} \tag{2.18}$$

where ${}^{b}\mathbf{a}_i$ is the estimated length of actuator i and l_i is the actual length of actuator i. By differentiating (2.18), we can obtain,

$$\dot{\mathbf{g}} = \mathbf{J} \begin{bmatrix} {}^{b}\mathbf{v}_p \\ {}^{b}\omega_p \end{bmatrix} = \mathbf{J} \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{B} \end{bmatrix} \begin{bmatrix} {}^{b}\mathbf{v}_p \\ \dot{\psi}_p \\ \dot{\theta}_p \\ \dot{\phi}_p \end{bmatrix} \tag{2.19}$$

So, the partial derivative of $\mathbf{g}$ with respect to $\mathbf{X}$ is given by,

$$\frac{\partial \mathbf{g}(\mathbf{X})}{\partial \mathbf{X}} = \mathbf{J}\left(x_p, y_p, z_p, \psi_p, \theta_p, \phi_p\right) \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{B} \end{bmatrix} \tag{2.20}$$

Substituting (2.17), (2.18), and (2.20) into (2.16) gives the following iteration

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ \psi_p \\ \theta_p \\ \phi_p \end{bmatrix}_{j+1} = \begin{bmatrix} x_p \\ y_p \\ z_p \\ \psi_p \\ \theta_p \\ \phi_p \end{bmatrix}_{j} - \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{B}^{-1} \end{bmatrix} \mathbf{J}^{-1} \begin{bmatrix} \|{}^{b}\mathbf{a}_A\| - l_A \\ \vdots \\ \|{}^{b}\mathbf{a}_F\| - l_F \end{bmatrix} \tag{2.21}$$

Given $\mathbf{l}$, we can obtain $\mathbf{X} = [x_p, y_p, z_p, \psi_p, \theta_p, \phi_p]^T$, provided that $\mathbf{J}$ and $\mathbf{B}$ are non-singular. Note that we can use the LU decomposition rather than compute the inverses of $\mathbf{B}$ and $\mathbf{J}$ explicitly.

We should mention that there was an error in the forward kinematics derivation in [3], where $\begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{B}^{-1} \end{bmatrix}$ was not included in the equation of iteration. Actually, the converging rate is better than claimed in [3].

Now, we look into the computing time of the forward kinematics. For our system, we use 16 bits A/D converters to convert the signals of actuator lengths, which range about 1.5 m. The resolution is about $\frac{1.5}{2^{16}}$ m, which is about 0.023 mm. The maximum velocity of each actuator is about 1.4 m/s. When the sampling frequency is 200 Hz, which is the one we are using, the maximum difference in actuator length between two adjacent sampling points is about ± 7 mm. So, assuming that Newton's method converges along the trajectory, we can always have an estimated starting point within ± 7 mm in actuator length from the actual point.

For a current platform configuration and corresponding actuator lengths, we add a displacement of ± 7 mm to the length of each actuator to represent the worst case of actuator lengths at the next sampling point, and we use current platform configuration as the starting point to calculate the next platform configuration, which is according to the actuator lengths with additional displacement of ± 7 mm, through the forward kinematics. Notice that we are calculating for 64 possible worst cases here. We record the largest estimation error in actuator length after two iterations, and then plot it versus x position and pitch angle in Figure (2.11).

We can see that the estimation error after two iterations is far smaller than the resolutions of actuator length sensors. The results are similar for the trajectory near the boundary of the cylindrical volume defined before. We should say that the result after two iterations is accurate enough for the real-time control of our system.

We can finish the computation of forward kinematics within 1 ms in our VME-based real-time system.

We derived the kinematics of the Stewart platform in this chapter. The workspace issue was studied in details and the result was used in the control program to ensure that the Stewart platform is always in the safe workspace. The numerical solution of the forward kinematics was evaluated using simulation, and it was concluded that result would reach the desired value after two iterations in the real-time control.

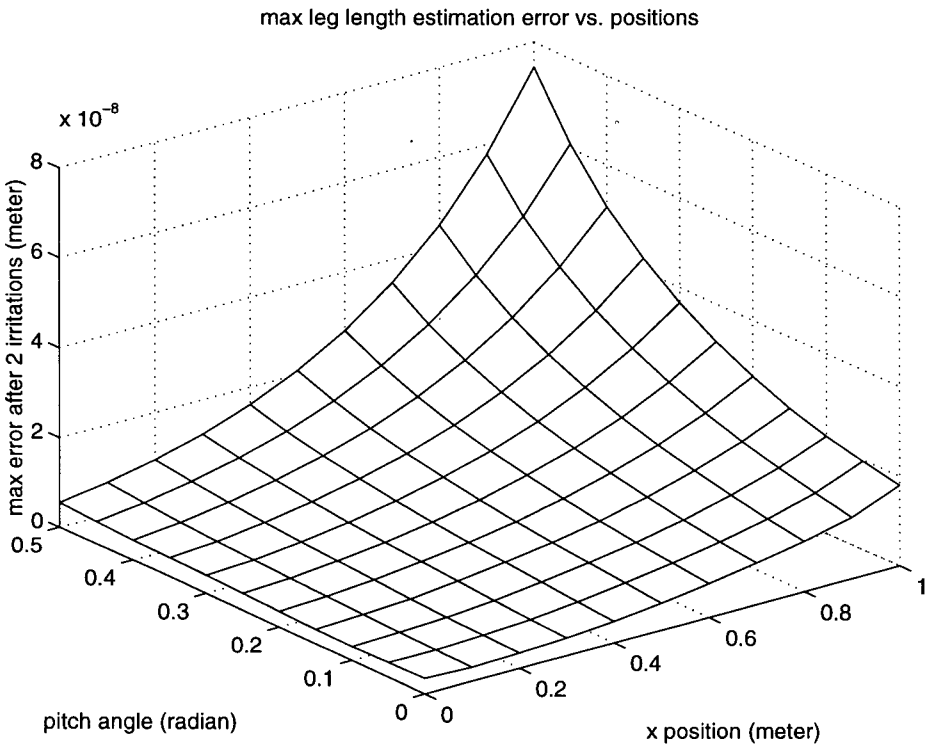

Figure 2.11: Maximum possible estimation error in actuator length after two iterations of Newton's method

Chapter 3

Platform Dynamics

In this chapter, we discuss the dynamic equations of the Stewart platform. A complete model of the inverse dynamics including the leg dynamics and the mobile platform dynamics is derived first, and a simplified model which only includes the mobile platform dynamics is given for the purpose of simulation and controller design.

3.1 Complete Model

We use the virtual work principle to formulate the dynamic equations of rigid body motion of the Stewart platform. A complete model of the inverse dynamics includes both the mobile platform dynamics and the leg dynamics. The diagram of a leg is shown in Figure 3.12.

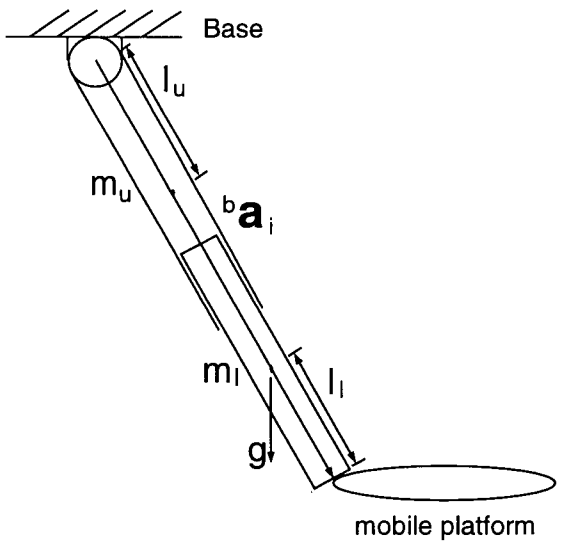

Figure 3.12: Diagram of leg i

Each leg consists of two parts: the upper fixed part with mass m_u, and the lower moving part with mass m_l. The distance from the base connection (the center of base attached joint) to the mass center of upper part is l_u while the distance from the platform connection (the center of platform attached joint) to the mass center of lower part is l_l. In the following dynamics analysis, the frictions in the actuators and U-joints are neglected to simplify the problem. Also we assume that the moment of inertia of leg i about the axis ${}^b\mathbf{a}_i$ is negligible and then each leg can be modeled as a slender rod.

The virtual work δw, done in translating the platform by the virtual distance ${}^b\mathbf{v}_p\delta t$, rotating the platform by the virtual angle ${}^b\omega_p\delta t$, and rotating the legs by the virtual angles ${}^b\omega_i\delta t$, is

$$\begin{aligned}\delta w &= m_p{}^b\mathbf{g}^{Tb}\mathbf{v}_p\delta t + \sum_{i=1}^{6}\left(m_l{}^b\mathbf{g}^{Tb}\mathbf{v}_{il} + m_u{}^b\mathbf{g}^{Tb}\mathbf{v}_{iu}\right)\delta t + \\ &\quad \mathbf{f}^T\mathbf{l}\delta t - \begin{bmatrix} {}^b\mathbf{f}_p \\ {}^b\tau_p \end{bmatrix}^T \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix}\delta t - \sum_{i=1}^{6}{}^b\tau_i{}^{Tb}\omega_i\delta t \end{aligned} \tag{3.22}$$

where $\mathbf{g}$ is the gravity acceleration, $\mathbf{l} = [l_A, l_B, l_C, l_D, l_E, l_F]^T$ is the six vector of the actuator lengths, $\mathbf{f}$ is the six vector of the actuator forces, m_p is the mass of the platform, ${}^b\mathbf{v}_p$ is the translational velocity of the platform, ${}^b\mathbf{v}_{il}$ is the translational velocity of the lower part of leg i, ${}^b\mathbf{v}_{iu}$ is the translational velocity of the upper part of leg i, ${}^b\mathbf{f}_p$ is the 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, ${}^b\omega_p$ is the angular velocity of the platform, ${}^b\tau_i$ is the torque acting about the base connection of each leg, and ${}^b\omega_i$ is the angular velocity of each leg around its base connection. Please notice that the effect of gravity is considered separately here.

Now we derive the expressions of ${}^b\mathbf{v}_{il}$ and ${}^b\mathbf{v}_{iu}$. Referring to Figure 3.12, we have

$${}^b\mathbf{a}_{iu} = \frac{l_u}{\|{}^b\mathbf{a}_i\|}{}^b\mathbf{a}_i \tag{3.23}$$

and

$${}^b\mathbf{a}_{il} = \frac{\|{}^b\mathbf{a}_i\| - l_l}{\|{}^b\mathbf{a}_i\|}{}^b\mathbf{a}_i \tag{3.24}$$

where ${}^b\mathbf{a}_{iu}$ is the vector from the center of i^{th} base attached joint B_i to the mass center of i^{th}

upper leg, and ${}^b\mathbf{a}_{il}$ is the vector from the center of i^{th} base attached joint B_i to the mass center of i^{th} lower leg.

Taking derivatives of the above equations and considering the derivatives of (2.1),

$$ {}^b\dot{\mathbf{a}}_i = \left[I, -\mathbf{S}\left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right] \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \tag{3.25} $$

we then have

$$ \begin{aligned} {}^b\mathbf{v}_{il} &= \frac{l_l {}^b\mathbf{a}_i{}^T {}^b\dot{\mathbf{a}}_i}{\|{}^b\mathbf{a}_i\| {}^b\mathbf{a}_i{}^T {}^b\mathbf{a}_i} {}^b\mathbf{a}_i + \frac{\|{}^b\mathbf{a}_i\| - l_l}{\|{}^b\mathbf{a}_i\|} {}^b\dot{\mathbf{a}}_i \\ &= \left(\frac{l_l}{\|{}^b\mathbf{a}_i\| {}^b\mathbf{a}_i{}^T {}^b\mathbf{a}_i} {}^b\mathbf{a}_i {}^b\mathbf{a}_i{}^T \left[I, -\mathbf{S}\left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right] + \frac{\|{}^b\mathbf{a}_i\| - l_l}{\|{}^b\mathbf{a}_i\|} \left[I, -\mathbf{S}\left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right] \right) \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \\ &= \mathbf{h}_{il} \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \end{aligned} \tag{3.26} $$

and

$$ \begin{aligned} {}^b\mathbf{v}_{iu} &= \frac{-l_u {}^b\mathbf{a}_i{}^T {}^b\dot{\mathbf{a}}_i}{\|{}^b\mathbf{a}_i\| {}^b\mathbf{a}_i{}^T {}^b\mathbf{a}_i} {}^b\mathbf{a}_i + \frac{l_u}{\|{}^b\mathbf{a}_i\|} {}^b\dot{\mathbf{a}}_i \\ &= \left(\frac{-l_u}{\|{}^b\mathbf{a}_i\| {}^b\mathbf{a}_i{}^T {}^b\mathbf{a}_i} {}^b\mathbf{a}_i {}^b\mathbf{a}_i{}^T \left[I, -\mathbf{S}\left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right] + \frac{l_u}{\|{}^b\mathbf{a}_i\|} \left[I, -\mathbf{S}\left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right] \right) \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \\ &= \mathbf{h}_{iu} \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \end{aligned} \tag{3.27} $$

where $S\left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)$ is the skew symmetric matrix of $\left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)$.

The angular velocity of each leg around its base connection is given by,

$$ \begin{aligned} {}^b\omega_i &= \frac{{}^b\mathbf{a}_i \times {}^b\dot{\mathbf{a}}_i}{{}^b\mathbf{a}_i{}^T {}^b\mathbf{a}_i} = \frac{{}^b\mathbf{a}_i \times \left[I, -\mathbf{S}\left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right]}{{}^b\mathbf{a}_i{}^T {}^b\mathbf{a}_i} \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \\ &= \mathbf{t}_i \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \end{aligned} \tag{3.28} $$

And the torque about the base connection of each leg can be expressed as,

$$ {}^b\tau_i = (I_{li} + I_{ui})\,{}^b\alpha_i + \dot{I}_{li}\,{}^b\omega_i \tag{3.29} $$

where I_{li} is the inertia variable of the lower part of leg i, I_{ui} is the inertia constant of the upper part of leg i, and ${}^b\alpha_i$ is given by taking derivatives of (3.28),

$$ {}^b\alpha_i = \mathbf{t}_i \begin{bmatrix} {}^b\mathbf{a}_p \\ {}^b\alpha_p \end{bmatrix} + \dot{\mathbf{t}}_i \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \tag{3.30} $$

When the effect of gravity is considered separately, the force acting on the platform can be expressed as,

$$ {}^b\mathbf{f}_p = m_p\,{}^b\mathbf{a}_p \tag{3.31} $$

where m_p is the mass of the platform and ${}^b\mathbf{a}_p$ (three vector) is the translational acceleration of the platform's center of mass.

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

$$ {}^b\mathbf{L}_p = {}^b\mathbf{I}_p\,{}^b\omega_p = {}^b\mathbf{R}_p\,{}^p\mathbf{I}_p\mathbf{R}_p^{T\,b}\omega_p \tag{3.32} $$

where ${}^p\mathbf{I}_p$ is the platform's inertia matrix expressed with respect to the platform frame and has the form

$$ {}^p\mathbf{I}_p = \begin{bmatrix} {}^p\mathbf{I}_{p_{xx}} & 0 & 0 \\ 0 & {}^p\mathbf{I}_{p_{yy}} & 0 \\ 0 & 0 & {}^p\mathbf{I}_{p_{zz}} \end{bmatrix}. \tag{3.33} $$

Taking the derivative of (3.32) gives the torque on the platform as

$$ {}^b\tau_p = {}^b\mathbf{I}_p\,{}^b\alpha_p + {}^b\omega_p \times \left({}^b\mathbf{I}_p\,{}^b\omega_p\right) \tag{3.34} $$

where ${}^b\alpha_p$ (three vector) is the angular acceleration of the platform about its center of mass and we assume that the inertia of the platform does not change with time in the platform frame.

The principle of virtual work states 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 $(\delta w = 0)$ [29]. Substituting

(2.6), (3.26), (3.27), and (3.28) into (3.22) and applying the principle of virtual work, we then have

$$\left(\begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix} m_p {}^b\mathbf{g} + \sum_{i=1}^{6} \left(\mathbf{h}_{il}{}^T m_l {}^b\mathbf{g} + \mathbf{h}_{iu}{}^T m_u {}^b\mathbf{g} \right) + \mathbf{J}^T \mathbf{f} - \begin{bmatrix} {}^b\mathbf{f}_p \\ {}^b\tau_p \end{bmatrix} - \sum_{i=1}^{6} (\mathbf{t}_i)^T {}^b\tau_i \right) \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \delta t = 0 \tag{3.35}$$

We can eliminate the arbitrary time interval δt and the arbitrary vector $\begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix}$. If $\mathbf{J}$ is non-singular, we can express the actuator forces as,

$$\mathbf{f} = \left(\mathbf{J}^T\right)^{-1} \left(\begin{bmatrix} {}^b\mathbf{f}_p \\ {}^b\tau_p \end{bmatrix} + \sum_{i=1}^{6} (\mathbf{t}_i)^T {}^b\tau_i - \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix} m_p {}^b\mathbf{g} - \sum_{i=1}^{6} \left(\mathbf{h}_{il}{}^T m_l {}^b\mathbf{g} + \mathbf{h}_{iu}{}^T m_u {}^b\mathbf{g} \right) \right) \tag{3.36}$$

where the expressions of $\mathbf{h}_{il}$, $\mathbf{h}_{iu}$, $\mathbf{t}_i$, ${}^b\tau_i$, ${}^b\mathbf{f}_p$, and ${}^b\tau_p$, can be found in (3.26), (3.27), (3.28), (3.29), (3.31), and (3.34), respectively. We now have obtained an expression of the actuator forces in the link-space in terms of the position, velocity, and acceleration of the platform in the Cartesian-space. This is the complete model of the inverse dynamics of the Stewart platform. The presented derivation is more straightforward, compared with the derivations in [10], [22], and [23]. The expression of final result is also simpler. However, the complete dynamic model is still a complicated one. We simplify the model in the next section by neglecting the leg dynamics.

3.2 Simplified Model

For our Stewart platform, the mass of the six legs is considerablly smaller than that of the mobile platform. In the rigid body dynamic equations, the leg dynamics part is more complicated and less important than the mobile platform dynamics part. If we neglect the leg dynamics, equation (3.36) becomes

$$\mathbf{f} = \left(\mathbf{J}^T\right)^{-1} \left(\begin{bmatrix} {}^b\mathbf{f}_p \\ {}^b\tau_p \end{bmatrix} - \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix} m_p {}^b\mathbf{g} \right) = \left(\mathbf{J}^T\right)^{-1} \begin{bmatrix} m_p {}^b\mathbf{a}_p - m_p {}^b\mathbf{g} \\ {}^b\mathbf{I}_p {}^b\alpha_p + {}^b\omega_p \times \left({}^b\mathbf{I}_p {}^b\omega_p \right) \end{bmatrix} \tag{3.37}$$

We now check how much the leg dynamics affect the calculation of forces. Each leg weights about 15 kg. We assume that the payload weights 250 kg and is evenly distributed in a cylindrical volume with height of 1.0 *meter* and radius of 0.6 *meter*. We run the simulation with and without considering the leg dynamics, and plot the required forces for several trajectories. Figure 3.13 shows the actuator forces when the motion trajectory is from the home position with an acceleration of $\ddot{x} = g$. Figure 3.14 shows the actuator forces when the motion trajectory is a sinusoidal wave along the x axis with amplitude of 0.01 *meter*. Figure 3.15 shows the actuator forces when the motion trajectory is from the home position with an acceleration of $\ddot{y} = g$ and also a sinusoidal pitch angle. We can see that the leg dynamics count less than 10%. So, we

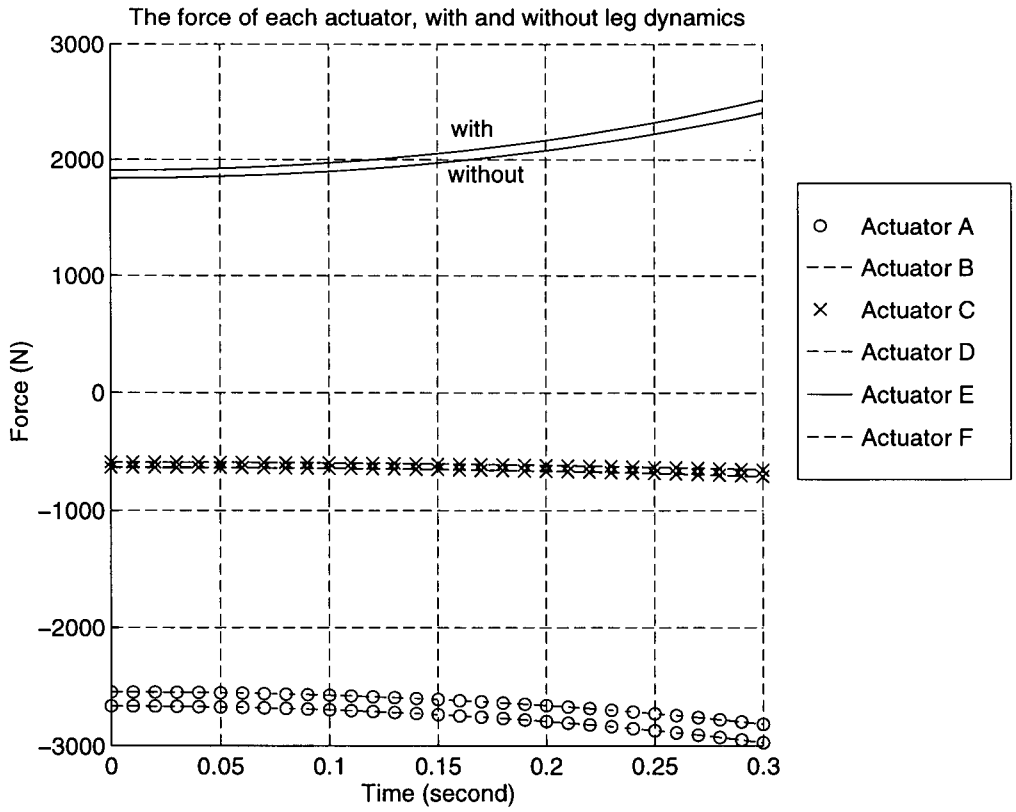

Figure 3.13: Motion from the home position with an acceleration of $\ddot{x} = g$

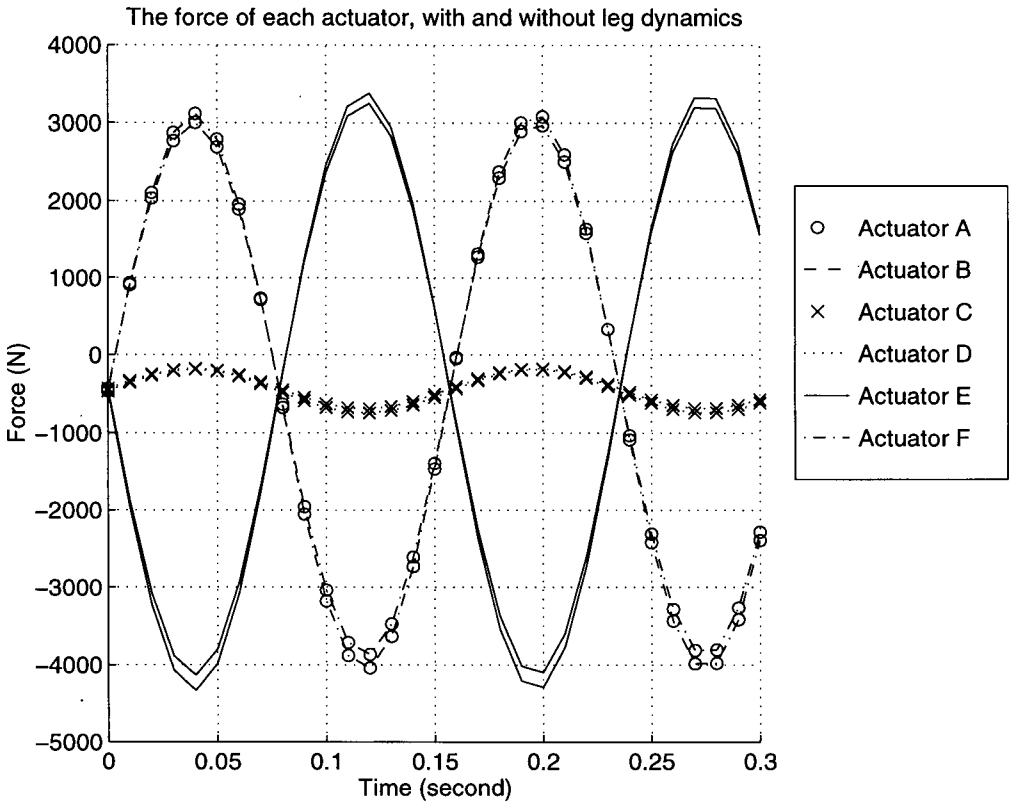

Figure 3.14: Sinusoidal motion along x axis

neglect the leg dynamics to avoid the huge computation in the simulation and real-time control. See Appendix D.4 for more details.

If we define the matrices $\mathbf{D}$ and $\mathbf{E}$ as in [3], the simplified dynamic model can be expressed as the same as in [3],

$$\mathbf{f} = \left(\mathbf{J}^T\right)^{-1} \left(\mathbf{D} \begin{bmatrix} {}^b\mathbf{a}_p \\ {}^b\alpha_p \end{bmatrix} + \mathbf{E} \right) \tag{3.38}$$

where

$$\mathbf{D} = \begin{bmatrix} m_p & 0 & 0 & \\ 0 & m_p & 0 & 0 \\ 0 & 0 & m_p & \\ & 0 & & {}^b\mathbf{I}_p \end{bmatrix} \tag{3.39}$$

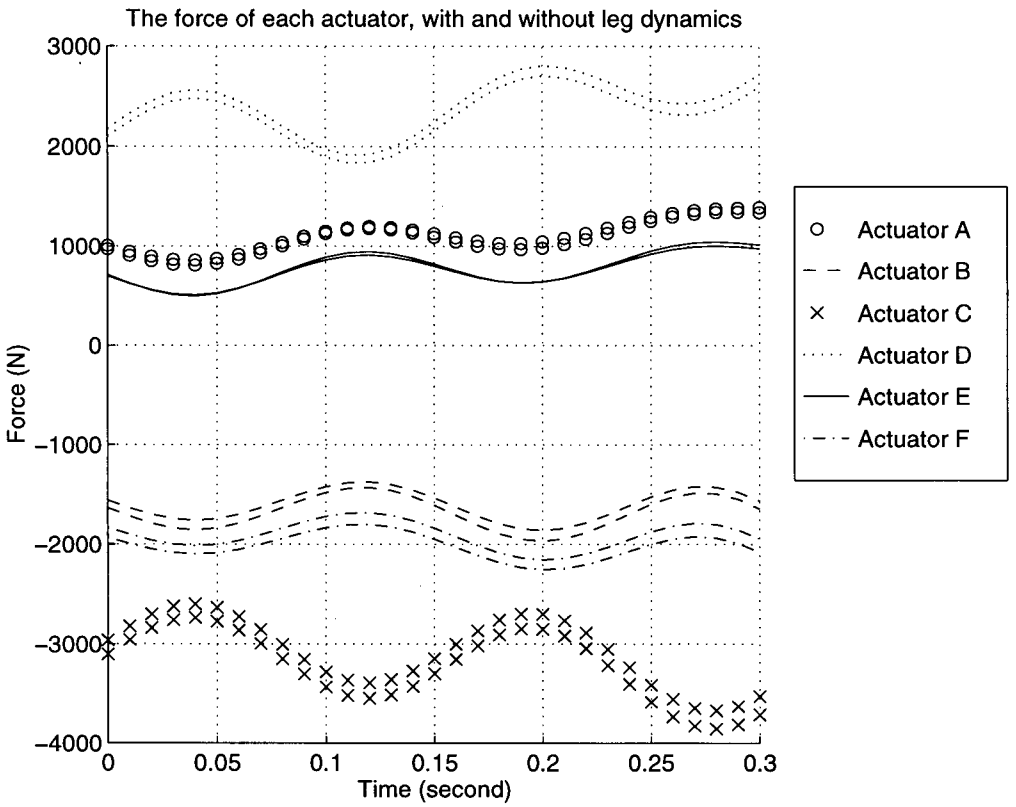

Figure 3.15: Motion with constant acceleration along y axis and a sinusoidal pitch angle

and

$$\mathbf{E} = \begin{bmatrix} 0 \\ 0 \\ m_p g \\ {}^b\omega_p \times \left({}^b\mathbf{I}_p {}^b\omega_p \right) \end{bmatrix} \tag{3.40}$$

We discussed the dynamics of Stewart platform in this chapter. The platform dynamic equations were derived in a simpler way than previously reported ([10], [23] and [22]). The dynamic simulations were then employed to show empirically that the platform leg dynamics could be neglected in predicting system trajectories. This simplified the simulation work and real-time control.

Chapter 4

Actuator Dynamics

The dynamics of electrohydraulic actuators are important in the control of the Stewart platform. In this chapter, we will derive the dynamic model of the electrohydraulic actuator, then linearize the model for the purpose of control and simulation, and in the last section we will validate the model using experimental data.

4.1 Deriving the Model

Each electrohydraulic actuator in our Stewart platform consists of an asymmetrical cylinder controlled by a proportional valve in a three-way configuration. Due to the financial limits, standard industrial cylinders with low-friction seals rather than the hydrostatic cylinders were used. The size of each cylinder is 1.5 inch bore, 60 inch stroke. The valves used are small size Rexroth 4WRDE three-stage proportional valves. Please refer to [3] for the details of the selection of these components.

Sensors in the system include the magnetostrictive wire length transducer for the length of each actuator, the linear variable differential transducer (LVDT) for the main stage spool position of each valve, and the single port pressure transducer for the pressure of each cylinder's blind end. The detailed description of the sensors can be found in [3].

Electrohydraulic servo systems are difficult to model because of the presence of nonlinearities. These nonlinearities include: nonlinear servo valve flow/pressure characteristics, variations in the trapped fluid volume in the system, the effects of nonlinear viscous and coulomb friction between the cylinder and the piston, and the effect of flow forces on the valve spool positions.

We follow the standard mathematical approach presented in [30] and [31] to model the electrohydraulic actuator. To analyze electrohydraulic servo systems, we should consider the effects of the following factors: the fluid flows through variable valve openings, the compressibility of fluid, the load forces and viscous frictions, and the dynamic responses from valve opening command values to actual valve spool positions. Here we study the hydraulic part first. Figure 4.16 shows a typical three-way valve/cylinder connection.

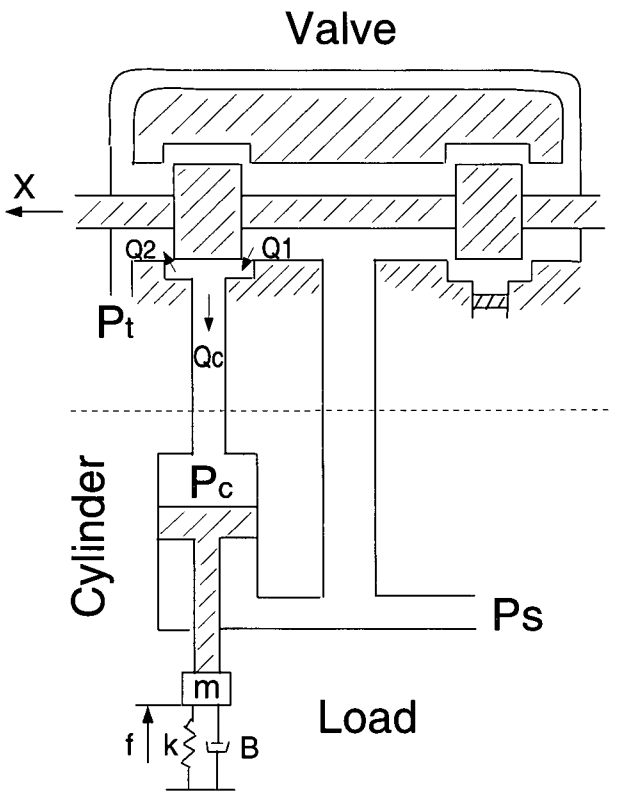

Figure 4.16: Definition of three-way connection parameters.

For the valve, assuming that there is no leakage, we can write the flow rate Q_c from the valve to the blind end of the cylinder as

$$Q_c = Q_1 - Q_2 \tag{4.41}$$

where Q_1 is the flow rate from the supply pressure port to the cylinder and Q_2 is the flow rate

from the cylinder to the tank. Using the orifice equation for turbulent flow [31], which is only valid for sharp edged orifices with adequately large flows, we can write expressions for Q_1 and Q_2 as

$$Q_1 = \begin{cases} C_d w (U + X_v) \sqrt{\frac{2}{\rho}(P_s - P_c)}, & X_v > -U \\ 0, & X_v < -U \end{cases} \tag{4.42}$$

and

$$Q_2 = \begin{cases} C_d w (U - X_v) \sqrt{\frac{2}{\rho}(P_c - P_t)}, & X_v < U \\ 0, & X_v > U \end{cases} \tag{4.43}$$

where C_d is the discharge coefficient, w is the port width of the valve, U is the valve's underlap, X_v is the spool position of the valve with respect to the center, ρ is the density of 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 pressure.

The compressibility of the fluid can be expressed as

$$V = V_u - \frac{V_u}{\beta} P \tag{4.44}$$

where β is the bulk modulus of the fluid, P is the pressure in the container, V_u is the volume of the fluid before compression (equivalent uncompressed volume), and V is the compressed (measured) fluid volume. In our system, $P = P_c$ is the pressure at the blind end of the cylinder, V is the total volume contained between the piston and the blind end of the cylinder, and V_u is the equivalent uncompressed volume. The transient flow rates associated with fluid compressibility are proportional to rates of change of pressure and may be expressed as

$$Q_p = \frac{V_u}{\beta} \frac{dP}{dt} \tag{4.45}$$

where Q_p is the transient flow rates associated with fluid compressibility, and β, P and V_u are defined as before.

The flow through the valve is the sum of the flow associated with the movement of piston (change in volume) and the flow associated with fluid compressibility. So we can express the

flow rate Q_c from the valve to the blind end of the cylinder as,

$$Q_c = \dot{V} + Q_p = \dot{V} + \frac{V_u}{\beta}\dot{P}_c \tag{4.46}$$

Considering the load forces and viscous frictions, we have,

$$P_c A - P_s a = M\ddot{l} + B\dot{l} + Kl + f \tag{4.47}$$

where 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. We assume $K = 0$ in the modeling.

We can solve the above equations for $\dddot{l}$ by differentiating equation (4.47) and substituting (4.41), (4.42), (4.43), and (4.46) into the result. The derived equation is,

$$\dddot{l} = \frac{1}{M}\left[\frac{\beta A}{V_u}\left(C_d w\sqrt{\frac{2}{\rho}}h\left(X_v, P_c\right) - \dot{V}\right) - B\ddot{l} - \dot{f}\right] \tag{4.48}$$

where

$$V_u \approx A\left(l - L\right)/\left(1 - \frac{P_c}{\beta}\right), \tag{4.49}$$

$$\dot{V} = A\dot{l}, \tag{4.50}$$

$$P_c = \frac{M\ddot{l} + B\dot{l} + f + P_s a}{A}, \tag{4.51}$$

$$h\left(X_v, P_c\right) = \begin{cases} \left(U + X_v\right)\sqrt{P_s - P_c}, & X_v > U \\ \left(U + X_v\right)\sqrt{P_s - P_c} - \left(U - X_v\right)\sqrt{P_c - P_t}, & -U < X_v < U \\ -\left(U - X_v\right)\sqrt{P_c - P_t}, & X_v < -U \end{cases} \tag{4.52}$$

and L is the stroke length of the cylinder.

The above equations can also be written as

$$\dddot{l} = f_0\left(l, \dot{l}, \ddot{l}, f, \dot{f}, S\right) \tag{4.53}$$

where S represents the system parameters.

In addition to the dynamic response in the hydraulic part, we have a dynamic response in the electrical part. The valve spool position X_v is controlled through a D/A board, and is supposed to be proportional to the command value of V_{DA}, which is the output of D/A board. However, the actual valve response is a dynamic one. The valve response can be approximately modeled as a second order closed-loop system with the transfer function

$$X_v(s) = \frac{1}{\frac{1}{\omega_e^2}s^2 + \frac{2\xi_e}{\omega_e}s + 1} V_{DA}(s) \tag{4.54}$$

where ω_e is the electrical undamped natural frequency and ξ_e is the electrical damping ratio. Parameters in this model can be estimated from the specification of the valve and then verified through the experiment.

Combining the electrical part model with the hydraulic part model, we can finally obtain a fifth-order nonlinear model of the electrohydraulic system.

4.2 Linearizing the Model

The above nonlinear model of the electrohydraulic system gives us a relatively complete description of the actuator dynamics. The hydraulic part of the model can be seen roughly as the combination of an integrator and a spring-damper system, while the electrical part is a second order closed-loop system. The natural frequency and damping ratio of the equivalent linear system determine most characteristics of the actuator dynamics. We now linearize the model to get a basic idea about the dynamic response of the actuator.

We first linearize the function of Q_c. The underlap U of the valve, which is $55.4 \times 10^{-6}\ m$, can be neglected in the calculation of flow rates. The controlled pressure P_c at the blind end of the cylinder is fast varying but within a small range for general maneuvers, so we can use the average value of P_c, which is about $\frac{1}{2}P_s$, to represent P_c, where we assume $\Delta P_c \ll \frac{1}{2}P_s$. Then, the equation (4.41) becomes

$$Q_c = C_d w \sqrt{\frac{P_s}{\rho}} X_v = K_q X_v \tag{4.55}$$

where K_q is the defined flow gain.

We then define the initial value of V_u to be V_0 and linearize the model about this particular operating point. From equation (4.55), (4.46), and (4.47), it is clear that,

$$K_q X_v = \frac{V_0 M}{A\beta}\dddot{l} + \frac{V_0 B}{A\beta}\ddot{l} + A\dot{l} \tag{4.56}$$

where K_q, V_0, M, A, β, and B are all constant parameters. By taking Laplace transform of the above equation, we can get the transfer function

$$L(s) = \frac{\frac{K_q}{A}X_v(s) - \frac{V_0}{A^2\beta}sF(s)}{\frac{V_0 M}{A^2\beta}s^3 + \frac{V_0 B}{A^2\beta}s^2 + s} = \frac{\frac{K_q}{A}V_{DA}(s)}{s(\frac{1}{\omega_h^2}s^2 + \frac{2\xi_h}{\omega_h}s + 1)(\frac{1}{\omega_e^2}s^2 + \frac{2\xi_e}{\omega_e}s + 1)} - \frac{\frac{V_0}{A^2\beta}F(s)}{\frac{1}{\omega_h^2}s^2 + \frac{2\xi_h}{\omega_h}s + 1} \tag{4.57}$$

where $\omega_h = A\sqrt{\frac{\beta}{V_0 M}}$ is the hydraulic undamped natural frequency and $\xi_h = \frac{B}{2A}\sqrt{\frac{V_0}{\beta M}}$ is the hydraulic damping ratio. We validate this linearized model in the next section using experimental data and also identify the undetermined parameters in the model.

4.3 Validating the Model

We use spectral analysis on the single actuator system to validate its dynamic model. By using white noise as input signal and analyzing the cross spectral density between the output signal and input signal of the system, we can get an experimentally estimated transfer function of the system. This method has been mentioned in [32] and some other books. For the detailed derivation, please see Appendix B.

In the experiment, the supply pressure is 400 PSI and the sampling frequency is 500 Hz. The cylinder rod weights about 5.5 kg, and a mass of 15.2 kg is attached to it as the additional load, so the total mass M as used in equation (4.57) is 20.7 kg. We first generate a band-limited digital white noise signal using the Matlab software package, and then low-pass filter the white noise signal using a discrete second-order Butterworth filter with cut-off frequency of 200 Hz. The processed white noise signal is then applied to the input of the system, i.e., the signal is used as the command value of the valve opening. The input signal and output signals are sampled at the frequency of 500 Hz for a long enough duration. At last, we perform the spectral analysis of the recorded input and output sequences in Matlab to obtain the estimated transfer function of the system.

We have two transfer functions in serial: one is from the command value of valve opening to the actual valve spool position, the other is from the valve spool position to the actuator length. The comparisons of simulation results (linearized models in the previous section) and experimental results are shown in Figure 4.17 and Figure 4.18. We can see that experimentally estimated transfer functions match theoretically derived transfer functions. The experiment is done around the operating point where the actuator length is $l_0 = 2.47\ m$. In Figure 4.17, the length unit is *meter*, while the valve opening unit is *voltage*, which is from the valve spool position sensor signal and equals to $0.00035\ meter$.

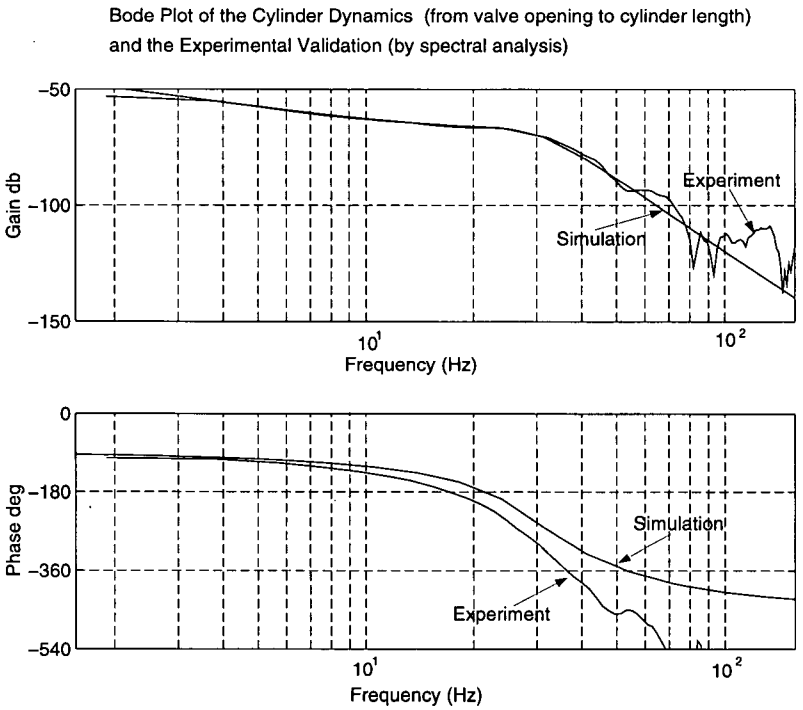

Figure 4.17: Simulation result and experimental result of the cylinder dynamics

The system parameters that give the above simulation results are shown in Table 4.4. Please note that values of β and B are not easy to measure, and we identify their values by matching the simaltion results with experimental results. And the corresponding ω_h and ξ_h are given by,

$$\omega_h = A\sqrt{\frac{\beta}{V_0 M}} = A\sqrt{\frac{\beta}{(l_0 - L) \times A\ M}} \approx 30Hz \tag{4.58}$$

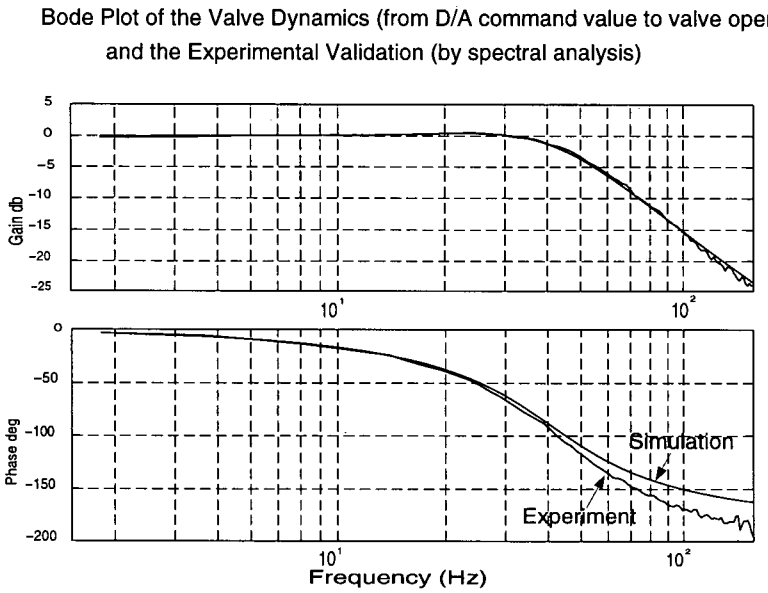

Figure 4.18: Simulation result and experimental result of the valve dynamics

Table 4.4: Actuator System Parameters.

Parameter	Value	Definition
β	700 MPa	Effective Fluid Bulk Modulus
A	$1.14 \times 10^{-3}\ m^2$	Area of the Piston
a	$6.33 \times 10^{-4}\ m^2$	Annulus Area
C_d	0.432	Effective Discharge Coefficient
w	5.7 mm	Port Width of the Valve
ρ	858.2 kg/m^3	Density of the Fluid
L	1.37 m	Stroke Length of the Cylinder
B	2500 N s/m	Viscous Damping Coefficient
P_s	400 PSI	Current Supply Pressure
P_t	0 PSI	Tank Pressure
U	55.4×10^{-6} m	Underlap of the Valve

and

$$\xi_h = \frac{B}{2A}\sqrt{\frac{V_0}{\beta M}} \approx 0.323 \tag{4.59}$$

Parameters of the valve dynamics are given by $\omega_e \approx 40Hz$ and $\xi_e \approx 0.6$. They are also obtained by matching the simaltion results with experimental results.

The effective discharge coefficient C_d is identified to be 0.432. And the flow gain K_q is then

given by,

$$K_q = C_d w \sqrt{\frac{P_s}{\rho}} \approx 0.14 \ m^2/s \tag{4.60}$$

while $K_q/A \approx 122.8 \ /s$ is the gain from valve spool position (in meter) to the actuator velocity (in m/s). The equivalent gain from the command value of valve opening (in voltage) to the actuator velocity (in m/s) is given by $122.8 \times 0.00035 = 0.043 \ m/s/volt$.

We derived the dynamic model of the electrohydraulic actuator in this chapter. We then linearized the model and validated it experimentally. Results showed that the linearized actuator model and the experimental data fit each other well. This model is the base of the design of link-space controllers.

Chapter 5

Control Implementation

Chapter 5 gives an overview of the control implementation on the Stewart platform. We introduce the layout of our control system first, and then study the design and performance of different types of basic link-space controllers. The pressure-feedback link-space controller is then proposed to ensure the stability of the system, and its performance is studied via both simulation and experiment. The Cartesian-space control of the Stewart platform is also discussed. In the final section, we address the motion drive algorithm.

5.1 Control Layout

Our Stewart platform based motion simulator is hydraulically actuated. The hydraulic actuation system includes a hydraulic power supply unit, a fluid distribution system, and six electrohydraulic actuators. The electrical system includes an electrical signal distribution box, a VME-based real-time system running VxWorks, and a workstation serving as user interface. Figure 5.19 shows the whole motion simulator system.

To ensure the safety of the motion simulator, a safety system is established, which includes "home" valves, isolation valves, pressure relief valves, motion limit switches, fluid temperature and level switches, a supply pressure transducer, and panic buttons. The safety system does not affect the performance of the simulator. Details of its design can be found in [3].

To simulate a motion using the Stewart platform, we should generate the motion trajectory in the Cartesian-space first. This trajectory can be a pre-planned one from the computer or a real-time one following the signal from the hand-control. The actual lengths of actuators are obtained from sensors through analog to digital converters. Then the control algorithm is

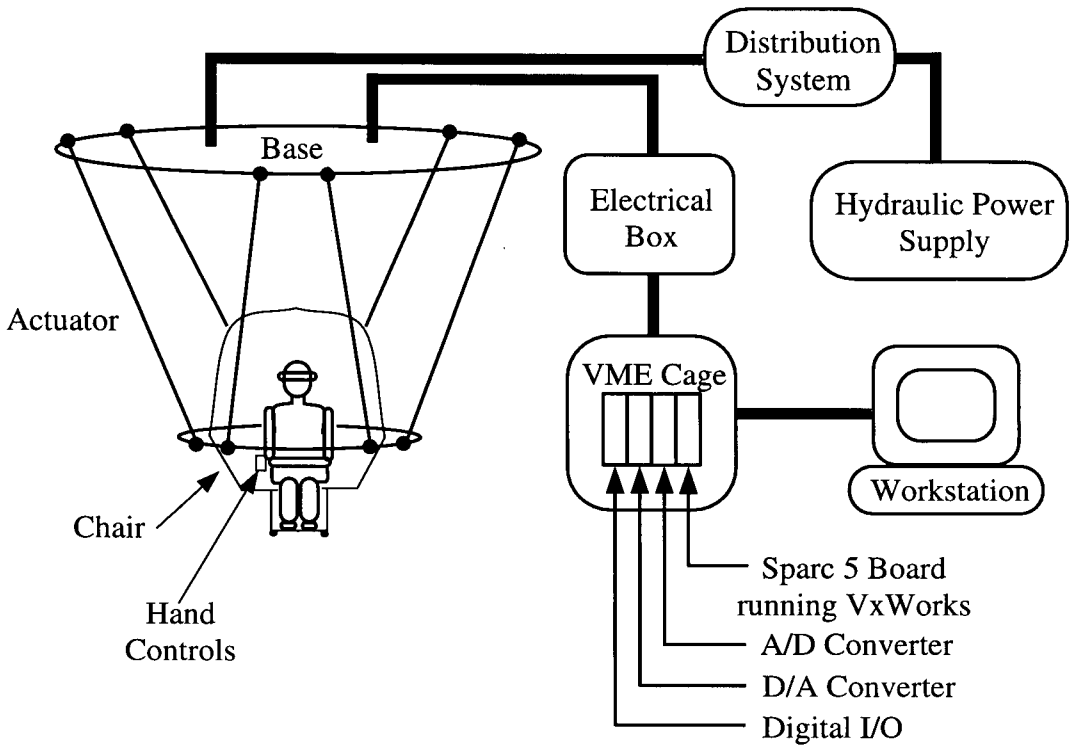

Figure 5.19: Diagram of the Stewart platform based motion simulator

performed either in Cartesian-space or link-space, and the command value of valve opening is derived for each actuator. By applying the derived command value to each valve through digital to analog converter, we can control the main-stage spool position of each valve, and hence the flow rate and the actuator length.

In the control of Stewart platforms, link-space controllers are generally used because they are relatively easy to design.

5.2 Basic Link-space Control

In the link-space control, we should perform the inverse kinematics to convert the Cartesian-space trajectory to the desired lengths of actuators, and then control each actuator individually to follow its desired length. In practice, most commonly used link-space controllers in the

control of hydraulically actuated Stewart platforms are Proportional-Integral-Derivative (PID) controllers and pre-filter controllers. Descriptions of various controllers can be found in [31], [30], and other books on the control of electrohydraulic actuators.

5.2.1 PID controller

Proportional gain (P) controller is the simplest approach to the control of electrohydraulic actuator. As we can see in Figure 5.20, the open-loop transfer function of the electrohydraulic actuator is nearly an integrator at low frequencies, i.e. the velocity of the actuator is nearly proportional to the control command value. So, a proportional gain controller with proper loop gain can make the closed-loop system stable.

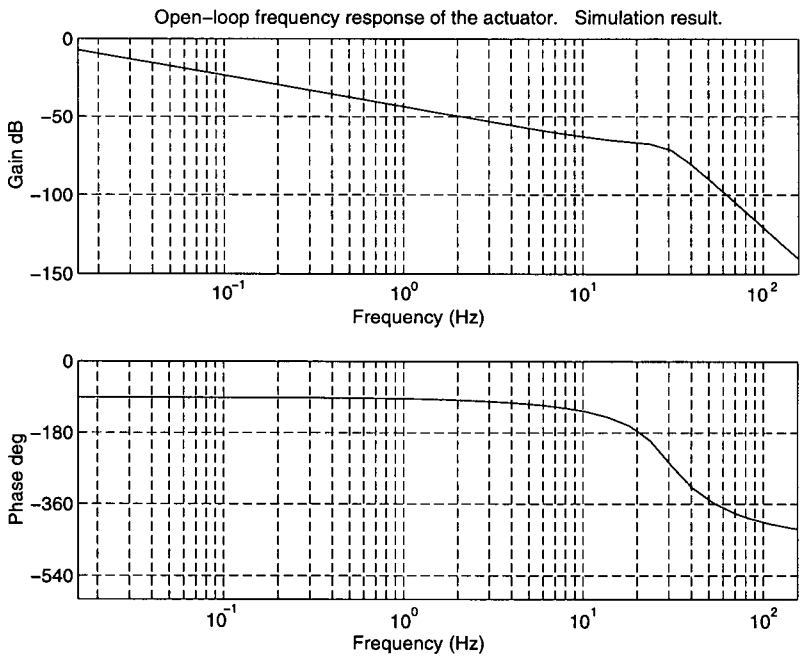

Figure 5.20: Linearized open-loop transfer function of the electrohydraulic actuator

But variations in the volume of trapped fluid V_0, in the viscous damping coefficient B, and in the effective load inertia M introduce uncertainties in the natural frequency and damping ratio of the hydraulic resonant mode, which is caused by the fluid compressibility. Figure 5.21,

Figure 5.22, and Figure 5.23 show the effects of these uncertainties on open-loop transfer functions of the electrohydraulic actuator when V_0, B, and M are varying from 0.5 to 2 times of their original values, respectively.

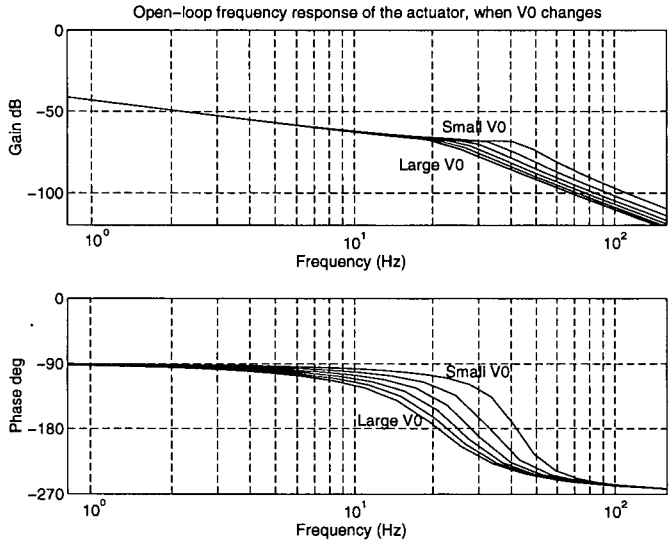

Figure 5.21: Effects of variations in V_0 from 0.5 to 2 times of its original value

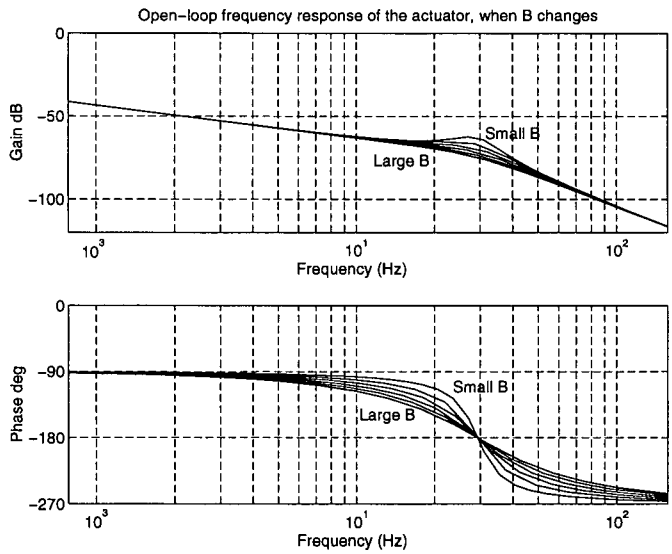

Figure 5.22: Effects of variations in B from 0.5 to 2 times of its original value

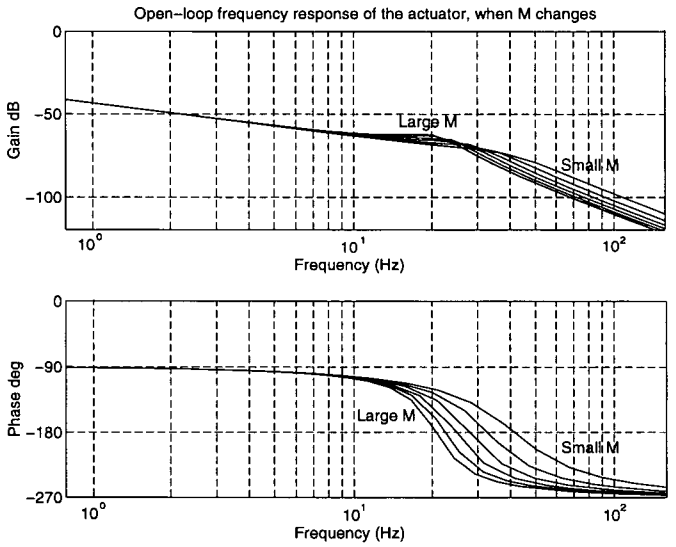

Figure 5.23: Effects of variations in M from 0.5 to 2 times of its original value

In addition, we have to consider the nonlinearity and uncertainty of the flow gain which is assumed to be a constant K_q in the simulation. Figure 5.24 shows effects of variations in V_0, B, and M, when they are combined together.

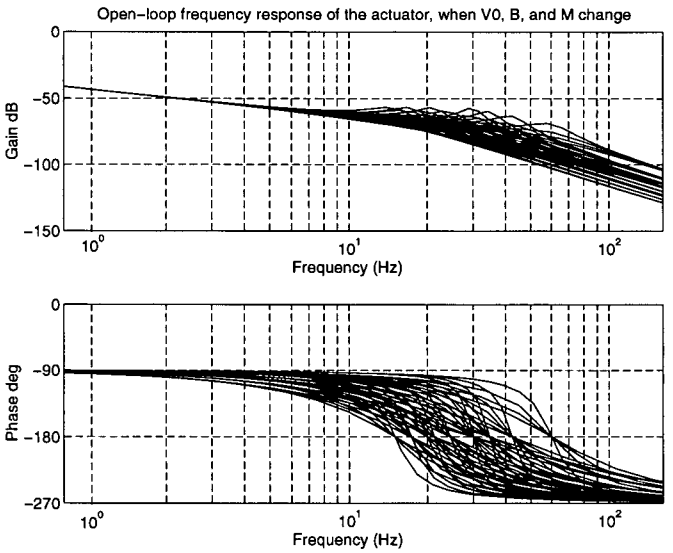

Figure 5.24: Effects of variations in V_0, B, and M

All these uncertainties influence the system response and then limit the achievable performance. To guarantee robustness, we have to design the controller for the worst case condition, that is for the system with the highest flow gain and the highest resonant peak value. This design strategy sacrifices the performance at the low frequencies to provide robustness at high frequencies. The closed-loop system has a slow response due to the conservative loop gain.

The loop gain in our design is 200 *volt/meter*, which means the computer system outputs a control command value of 1 *volt* for an length error of 1/200 *meter*. The closed-loop system is stable. Simulation curve of the closed-loop response and some experimental points are compared and shown in Figure 5.25.

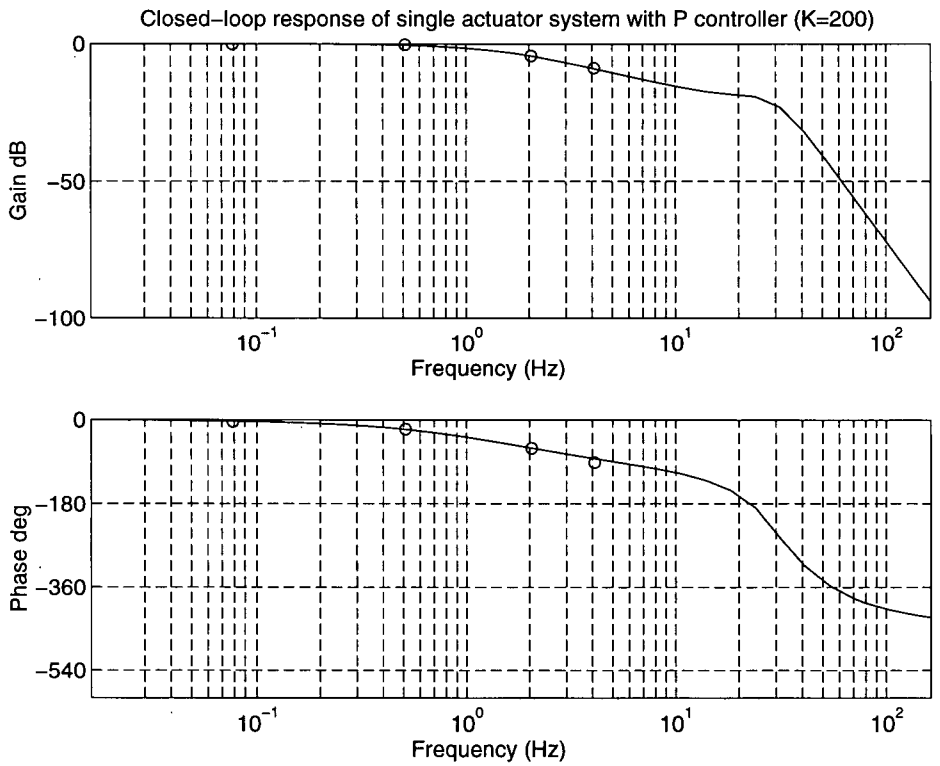

Figure 5.25: Comparison of the simulation curve and some experimental points

In the ideal case, we assume that all the six actuators have the same dynamic characteristics and move simultaneously. But in the practice, each actuator has slightly different parameters. Also the volume of trapped fluid V_0 and the effective load inertia M are different for each

actuator, when actuator lengths are not equal to each other. So the six actuators do not have the same frequency response. Figure 5.26 shows the system response of a 0.5 Hz sinusoidal wave input along the z axis. We plot the the desired actuator lengths and the actual actuator lengths in the link-space. As we can see, there are slight differences in the actual actuator lengths, even though the desired actuator lengths are the same.

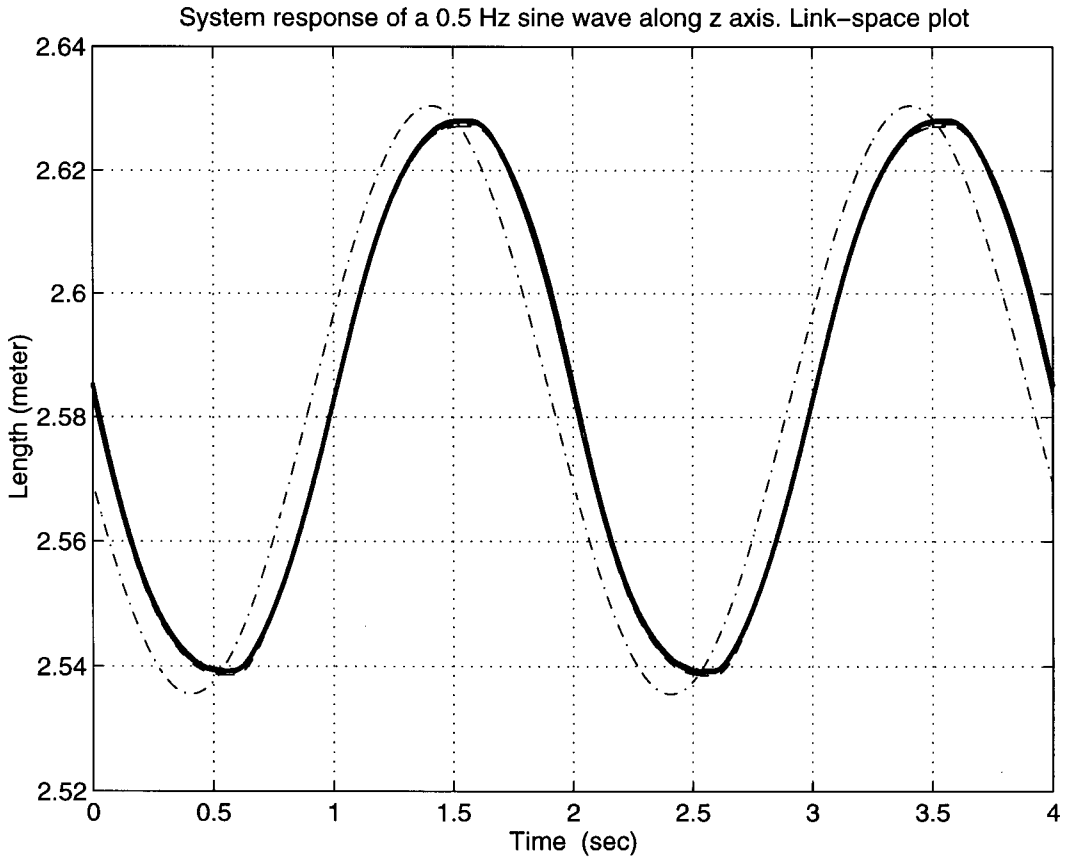

Figure 5.26: Link-space response of a sinusoidal wave input along the z axis.

To check how this differences affect the motion trajectory, we perform the forward kinematics to convert the actuator lengths to the trajectory in the Cartesian-space. Figure 5.27 shows the same system response in the Cartesian-space, where only the translational displacements are plotted. We can see that levels of disturbances in x and y axis are acceptable. But when input frequency increases, the performance becomes worse.

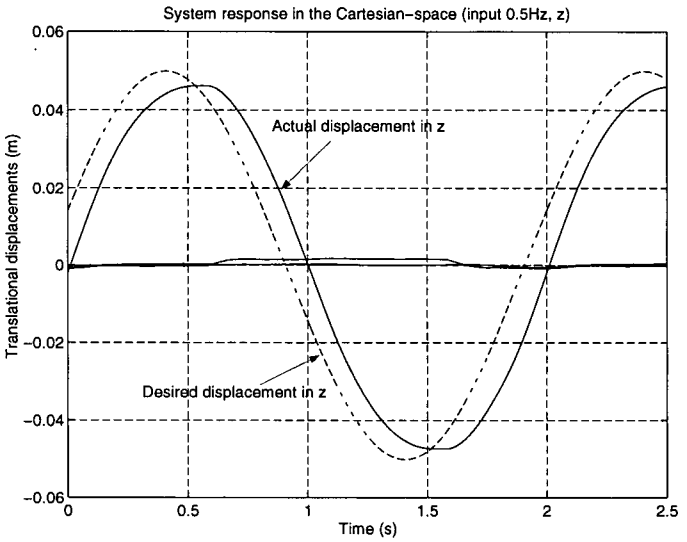

Figure 5.27: Cartesian-space response of a sinusoidal wave input along the z axis.

The system response in Cartesian-space of a 0.2 Hz sinusoidal wave input along the y axis is shown in Figure 5.28

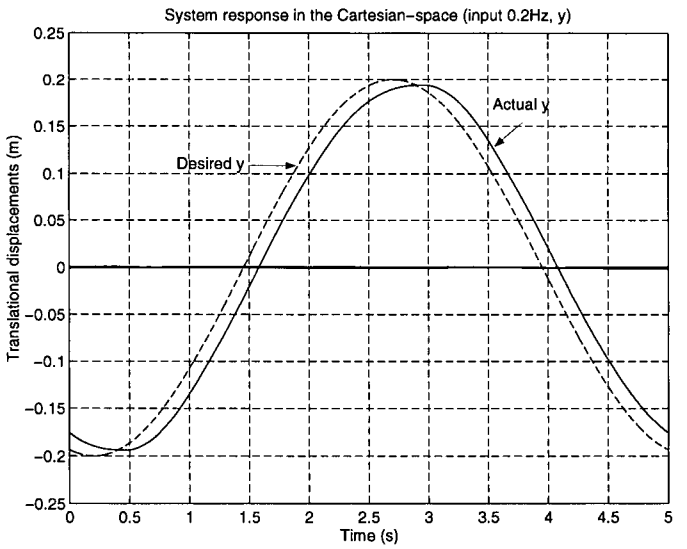

Figure 5.28: Cartesian-space response of a sinusoidal wave input along the y axis.

A proportional derivative (PD) controller $K_p\tau_d s + K_p$ can be used to improve the system

response at high frequencies. In the case where the valve response is very fast, i.e. ω_e is much larger than ω_h, we can select the crossing frequency to be near the hydraulic undamped natural frequency ω_h and apply the PD controller to introduce a nearly 90 degrees phase lead at that frequency so as to have enough phase margin and to stablize the closed-loop system. Unfortunately, our valve response is not fast enough, compared with hydraulic undamped natural frequency ($\omega_e \approx 1.4\omega_h$). The valve dynamics would introduce a substantial phase lag at the crossing frequency and make the closed-loop system unstable. So the PD controller is not applied.

For some applications of electrohydraulic actuators, a proportional integrative (PI) controller $\frac{K_p\tau_i s+K_p}{\tau_i s}$ can be used to increase the open-loop gain at low frequencies and to improve the system response. However, the PI controller should be tuned carefully, because the open-loop transfer function then contains a double integrator and can cause serious stability problems. Because the controller stability is critical for the motion simulator system, we prefer only to use a P controller to avoid risks with regard to stability.

5.2.2 Pre-filter controller

We now look at a feed-forward compensation controller which is generally used in tracking systems. Figure 5.29 shows a tracking control system with this type of feed-forward compensation controller.

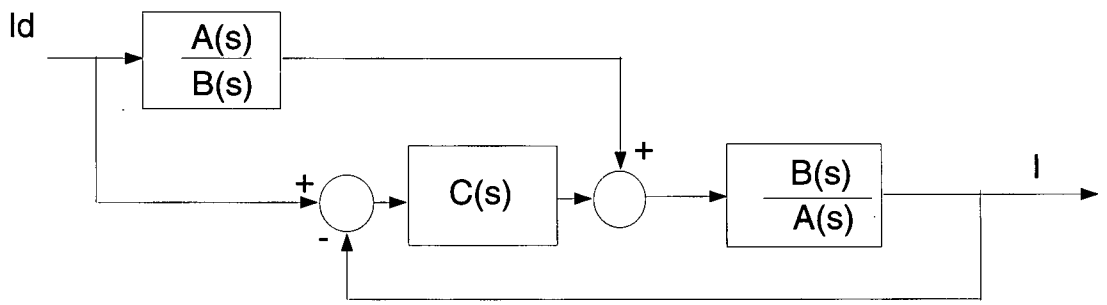

Figure 5.29: Block diagram of the tracking control system

The control law is

$$u(s) = \frac{A(s)}{B(s)} l_d(s) + C(s)(l_d(s) - l(s)) \tag{5.61}$$

The steady state closed-loop response can be easily found to be $l(s) = l_d(s)$, when $C(s)$ is not equal to $-\frac{A(s)}{B(s)}$. The feed-forward part of the controller helps to reduce and eliminate the tracking error, while the feedback part keeps the system stable and reduces disturbances.

For the high order plant like our electrohydraulic system, which has a 5th order linearized model, it is not reasonable to fully realize the feed-forward controller, i.e. calculate up to 5th order derivatives of the desired trajectories. We can just omit the high order derivatives from the feed-forward, and this will only cause bounded error in tracking [33]. And we select the $C(s)$ to be a constant C equal to the proportional loop gain.

Without considering external force, we can rewrite the plant model Equation (4.57) as,

$$L(s) = \frac{\frac{K_q}{A} V_{DA}(s)}{s(\frac{1}{\omega_h^2}s^2 + \frac{2\xi_h}{\omega_h}s + 1)(\frac{1}{\omega_e^2}s^2 + \frac{2\xi_e}{\omega_e}s + 1)} \tag{5.62}$$

So $B(s)$ as in Figure 5.29 becomes $\frac{K_q}{A}$, while

$$A(s) = s(\frac{1}{\omega_h^2}s^2 + \frac{2\xi_h}{\omega_h}s + 1)(\frac{1}{\omega_e^2}s^2 + \frac{2\xi_e}{\omega_e}s + 1) \tag{5.63}$$

Omitting the 3rd, 4th, and 5th order derivatives, we can obtain the control law of the simplified feed-forward compensation controller as

$$u(s) = \frac{A}{K_q}\left[(\frac{2\xi_h}{\omega_h} + \frac{2\xi_e}{\omega_e})s^2 + s\right] l_d(s) + C(l_d(s) - l(s)) \tag{5.64}$$

That is the same as the 2nd order pre-filter controller introduced in [2], which is

$$V_{DA}(s) = K_a s^2 + K_v s + K_p(l_d(s) - l(s)) \tag{5.65}$$

Substituting the values of ξ_h, ω_h, ξ_e, and ω_e into the above equation and considering the open loop gain of 200, we can obtain the parameters of the pre-filter in Equation(5.65) as $K_a = 0.19$, $K_v = 23.3$, and $K_p = 200$. The stability margins remain the same as of the P controller with loop gain of 200. A simulation curve and some experiment points of the closed-loop frequency response are plotted in Figure 5.30. Comparing the plot to Figure 5.25, we can

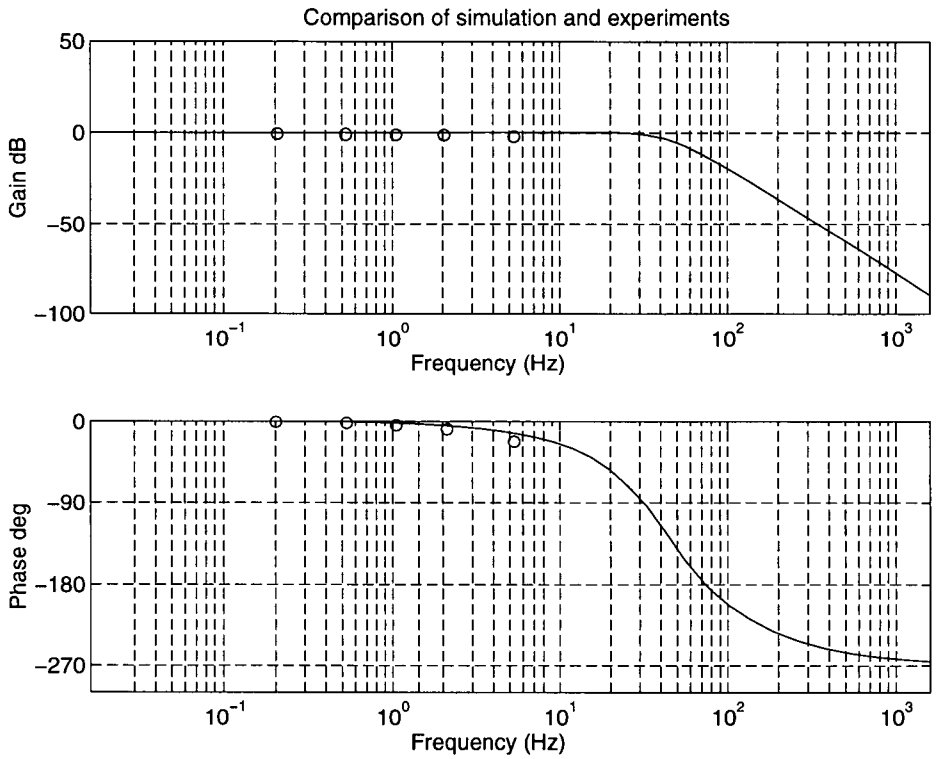

Figure 5.30: Closed-loop response of single actuator system with pre-filter controller

see that the closed-loop response is much improved. When the desired trajectory is pre-planned, the desired velocity and the desired acceleration are generally available, and we can apply the pre-filter off-line to save the in-loop computation time.

The closed-loop response in the time domain is plotted in Figure 5.31, for the input frequency of 1.0 Hz. The response in time domain is not a perfect sinusoidal wave, because there are uncertainty and nonlinearity in the system.

The system response changes dramatically when the system parameters change. The application of such a controller requires a good identification of the system parameters. K_a and K_v should be finely tuned according to the system parameters. Unfortunately, this is not always the case when we apply the control algorithm.

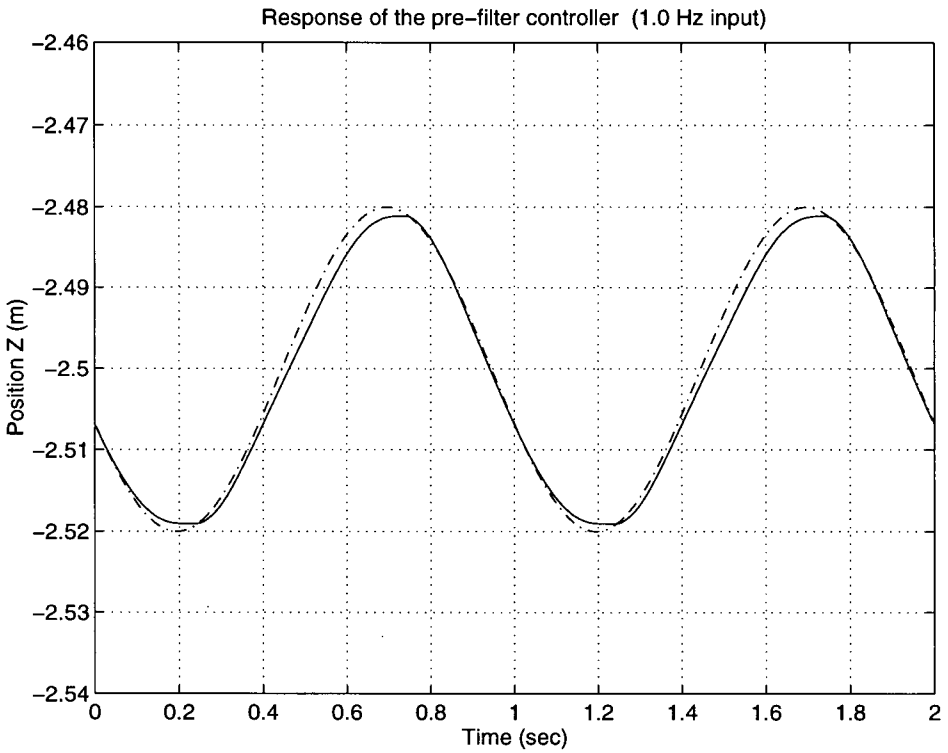

Figure 5.31: Closed-loop response of system with pre-filter controller

In the following section, we introduce the pressure-feedback control of electrohydraulic actuators which is an approach similar to the motor-shaft position control of elastic joints.

5.3 Pressure-feedback Link-space Control

To achieve better performance, one should be able to increase the loop gain and thus decrease the phase lag while still ensuring stability. As analyzed in the last section, the simple proportional gain controller becomes unstable if the loop gain is too high, because of the hydraulic resonant mode caused by the fluid compressibility. In this section, we introduce a new type of pressure-feedback controller, which can ensure the stability of the system. The performance of such type of controller is studied via simulations and experiments.

5.3.1 Description of the controller

It is known that end-point position control of a mass actuated by a motor through a flexible transmission has poor stability robustness, but motor-shaft position control has much better stability robustness. When the motor-shaft position is not directly available, it can be computed by measuring the end-point position and the stress on the motor shaft. A similar approach can be applied to the control of electrohydraulic actuators, where the compressibility of fluid is analogous to the flexibility of transmission, the actuator length is analogous to the end-point position, and cylinder pressure measurement is analogous to the stress measurement on the motor shaft.

The measurement of uncompressed fluid trapped in the cylinder rather than the end-point position of the cylinder is used as the feedback signal. It turns out that such a control method is equivalent to a P/PD controller with negative pressure feedback. Its performance is much better as it allows for a much larger position loop gain while keeping the system stable.

We revise the actuator model presented in Section 4.1 to provide the access to this new type of controller design. In this approach, V_u, the equivalent uncompressed volume of the fluid trapped between the valve opening and the blind end of the cylinder, is used as an intermediate variable.

We rewrite the compressibility equation as in (4.44),

$$V = V_u - \frac{V_u}{\beta} P_c \tag{5.66}$$

and the flow equation as in (4.46),

$$Q_c = \dot{V} + \frac{V_u}{\beta} \dot{P}_c \tag{5.67}$$

From the above equations, Q_c and V_u are related by

$$Q_c = \dot{V}_u (1 - \frac{P_c}{\beta}) \tag{5.68}$$

and when pressure P is a constant value, we have $Q_c = \dot{V}$.

Considering the load forces and viscous frictions, we have,

$$P_c A - P_s a = M\ddot{l} + B\dot{l} + f \tag{5.69}$$

Substituting (5.69) into (5.66) and considering

$$V = A(l - L) \tag{5.70}$$

we can get

$$V = V_u(1 - P_c/\beta) = V_u\left(1 - \frac{1}{\beta A}(P_s a + M\ddot{l} + B\dot{l} + f)\right) = A(l - L) \tag{5.71}$$

That is,

$$\frac{A}{1 - P_s a/A\beta}\left(\frac{V_u M}{A^2\beta}\ddot{l} + \frac{V_u B}{A^2\beta}\dot{l} - (l - L)\right) + \frac{V_u}{A\beta - P_s a}f = V_u \tag{5.72}$$

When V_u is constant, i.e. $Q_c = 0$, the actuator behaves like a spring-damper system. When V_u is relatively large and slowly varying, we can consider the actuator model as a spring-damper model with time varying parameters. The uncompressed fluid volume V_u and the external force f are the inputs, while the length $(l - L)$ is the output. The linearized time varying transfer function is

$$L(s) = \frac{K_v V_u(s) - K_f F(s)}{\frac{1}{\omega_n{}^2}s^2 + \frac{2\xi}{\omega_n}s + 1} \tag{5.73}$$

where $\omega_n = A\sqrt{\frac{\beta}{V_u M}}$ is the undamped natural frequency, $\xi = \frac{B}{2A}\sqrt{\frac{V_u}{\beta M}}$ is the damping ratio, $K_v = \frac{1 - P_s a/A\beta}{A}$ is the volume gain, and $K_f = \frac{V_u}{A^2\beta}$ is the force gain.

This is equivalent to the following spring-damper system: where $k = \frac{A^2\beta}{V_u}$ and the other parameters are the same as above. $K_v V_u$ is the position input and $kK_f f$ is the force input to the system.

To achieve the performance similar to that of the motor-shaft position control, the measurement of uncompressed fluid trapped in the cylinder, V_u, rather than the end-point position of the cylinder, l, is used as the feedback signal. So the hydraulic resonance is avoided.

The desired value of V_u is given by,

$$V_{ud} = A(l_d - L) \tag{5.74}$$

where l_d is the desired actuator length.

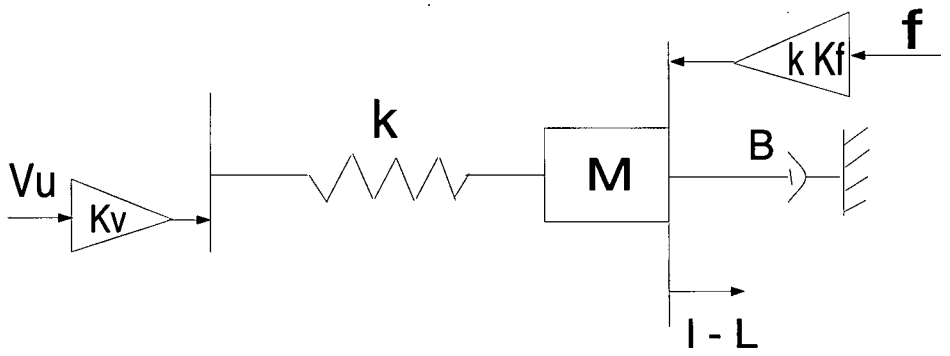

Figure 5.32: Equivalent spring-damper system

A simple pressure-feedback controller is given by,

$$X_v = K_{new}(V_{ud} - V_u) \tag{5.75}$$

Substituting (5.66), (5.70), and (5.3.1) into the above equation and assuming $\beta >> P_c$, we can then further simplify the control law to be

$$X_v = K_{new}A(l_d - l) - \frac{K_{new}A(l_d - L)}{\beta}P_c \tag{5.76}$$

a proportional gain controller with negative pressure feedback. For certain fluid, the range of β is generally given. To get the accurate value of β, one can use parameter identification scheme.

In the following sections, we study the performance of this new type of pressure-feedback controller via simulations and experiments.

5.3.2 Simulation results and discussions

We use Simulink to do the simulation. The diagram of single actuator system with proportional gain controller is shown in Figure 5.33.

When the controller loop gain K_p is too high, the system starts to oscillate. Figure 5.34 shows the oscillation when $K_p = 800$.

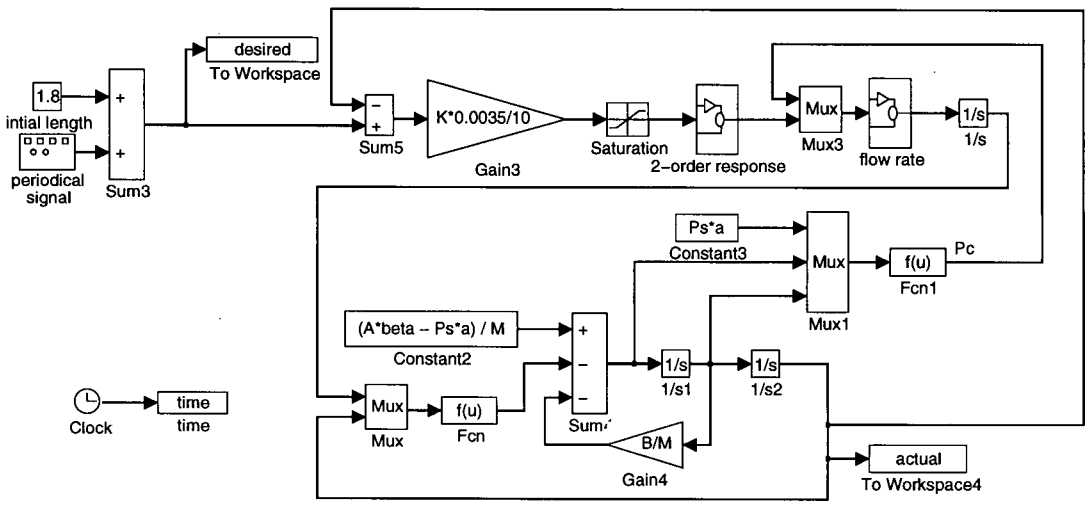

Figure 5.33: Single actuator system with proportional controller

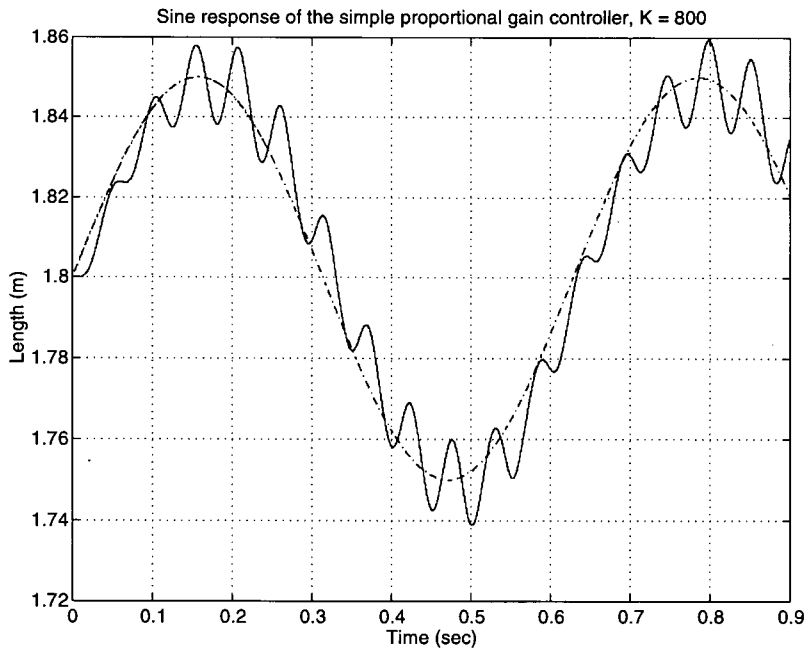

Figure 5.34: The sine wave response of P controller, high loop gain

Figure 5.35 shows the diagram of the single actuator system with the pressure-feedback controller.

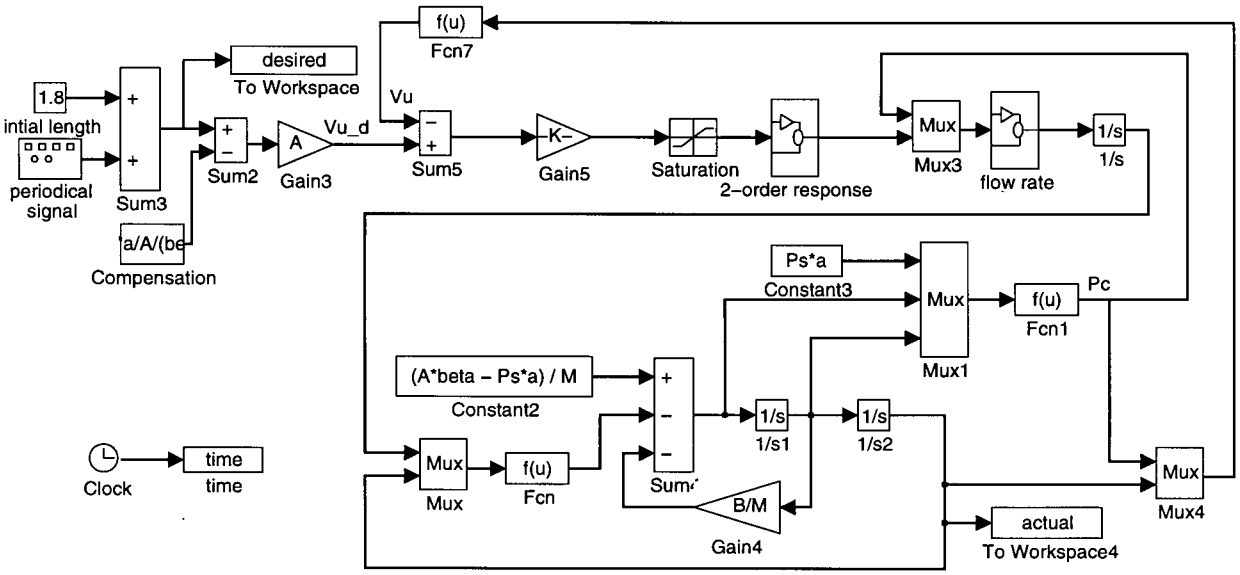

Figure 5.35: Single actuator system with pressure-feedback controller

The system response of the pressure feedback controller is similar to that of the proportional gain controller, if the loop gains are the same and are small enough to keep system stable. One of the simulation results is shown in Figure 5.36, when the input is 2.0 Hz, and the loop gain is 200. We can see the significant phase lag.

To improve the system performance, we should increase the loop gain. For the proportional gain controller, if we increase the loop gain to 600, the system response will become unstable. But for the pressure-feedback controller, when the loop gain is 1200, the system response is still stable, and the system performance is much improved. See Figure 5.37 for the simulation result. The performance of the proposed pressure-feedback controller is much better than that of a simple proportional gain controller as it allows for a much larger loop gain while keeping the system stable.

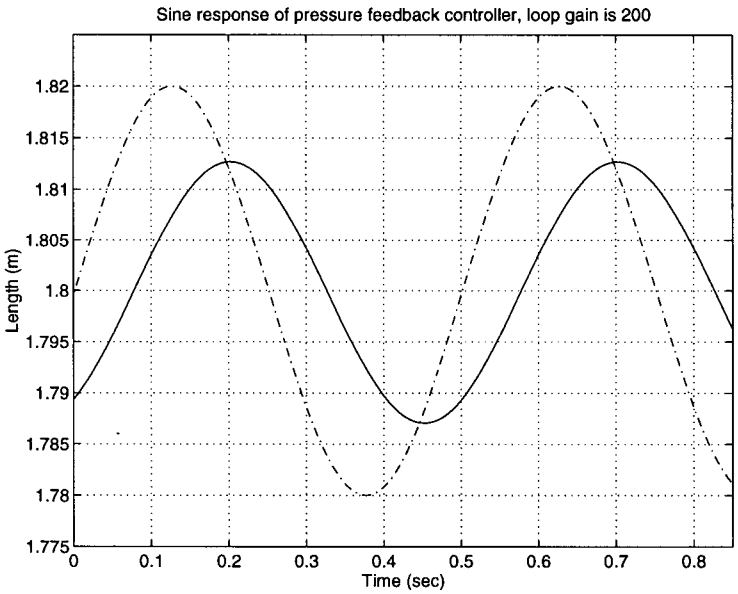

Figure 5.36: Response of single actuator system with pressure-feedback controller

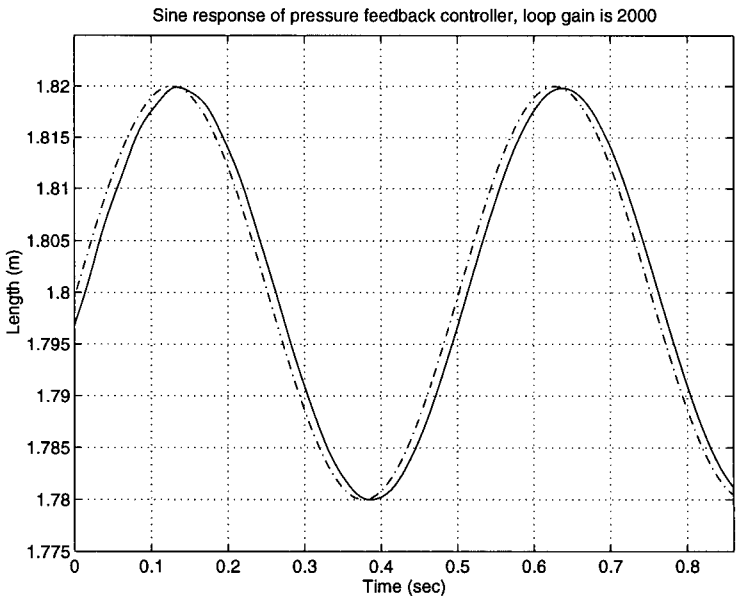

Figure 5.37: Response of single actuator system with pressure-feedback controller

The step responses for different step sizes are plotted in Figure 5.38. For the larger step

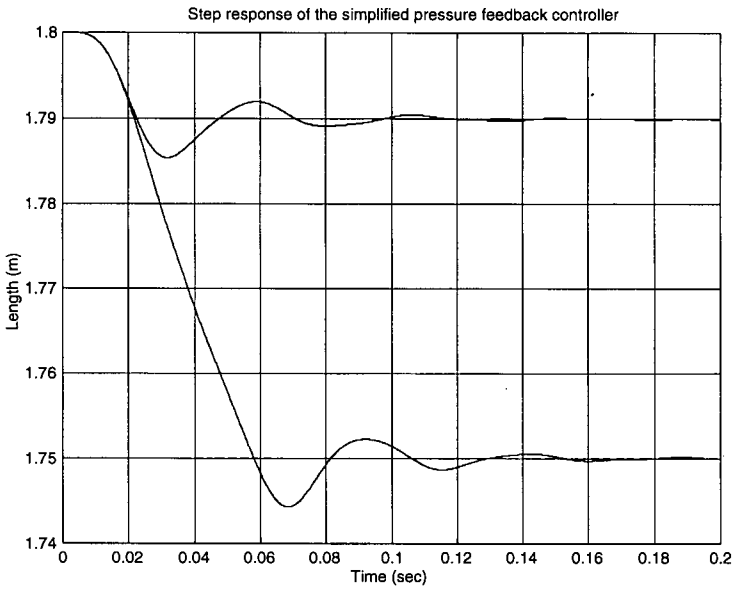

Figure 5.38: Step response of single actuator system with pressure-feedback controller

size, the system reachs the steady state with a slightly larger overshot and a longer time. This is similar to the result when the proportional gain controller is applied.

In Figure 5.39, we can observe a small oscillation in the valve opening, which is due to the second order valve dynamics. The frequency of the valve oscillation is according to the natural frequency of the valve dynamics. In our system, the natural frequency of valve dynamics is about 1.4 times of the hydraulic resonant frequency. We can achieve a better performance if a valve with higher natural frequency is used.

The pressure-feedback would generate a steady-state tracking error due to the variance in the load/pressure. This effect of load could be compensated using the rigid body dynamic model of the Stewart platform.

5.3.3 Experimental verification

The pressure-feedback controller can stabilize the actuator system.

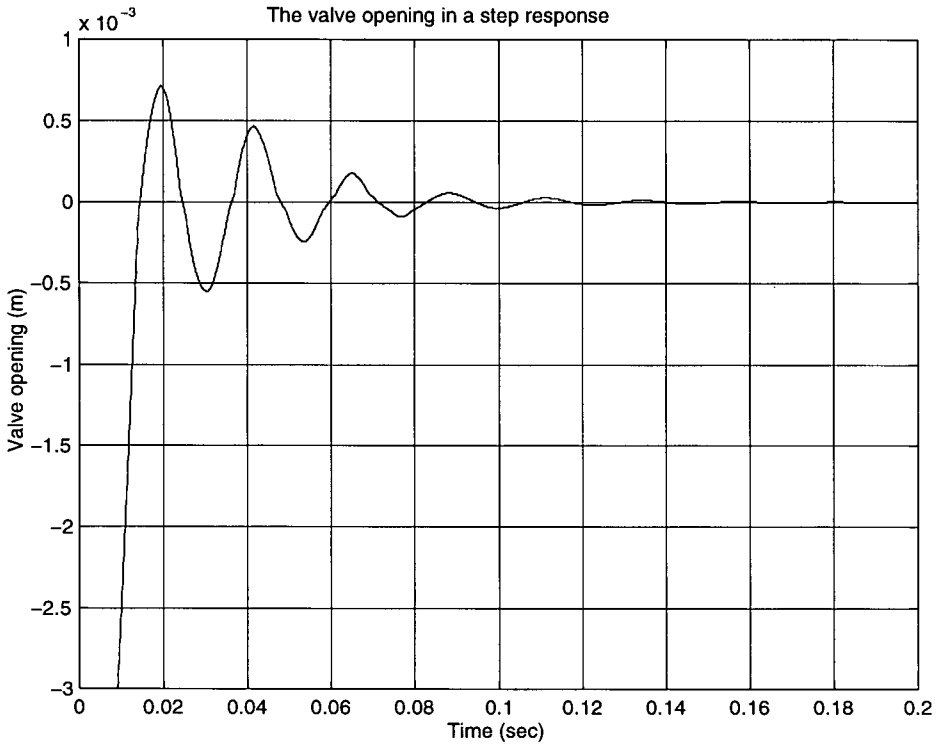

Figure 5.39: Valve step response when the pressure-feedback controller is used

We have observed dramatic oscillation when we increased the loop gain of the proportional gain controller by 3 times to be 600.

But for the pressure-feedback controller, we can increase the loop gain to 1200 while keeping the system stable. The experimental results of a single actuator are shown in Figure 5.40 and Figure 5.41 for the sinusoid input and step input, respectively. The loop-gain selected is 2000, and the closed-loop system has a cut-off frequency of about 15 Hz.

Through experiments, we successfully proved that this new type of pressure-feedback controller is practically useful. This approach of the pressure-feedback controller design can be applied to the general electrohydraulic actuator control to improve the performance and guarantee the stability.

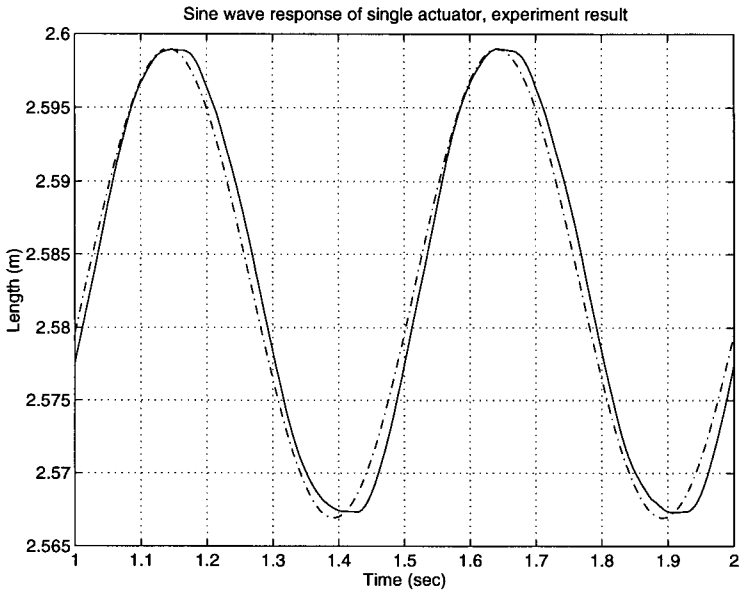

Figure 5.40: Sine response when the pressure-feedback controller is used

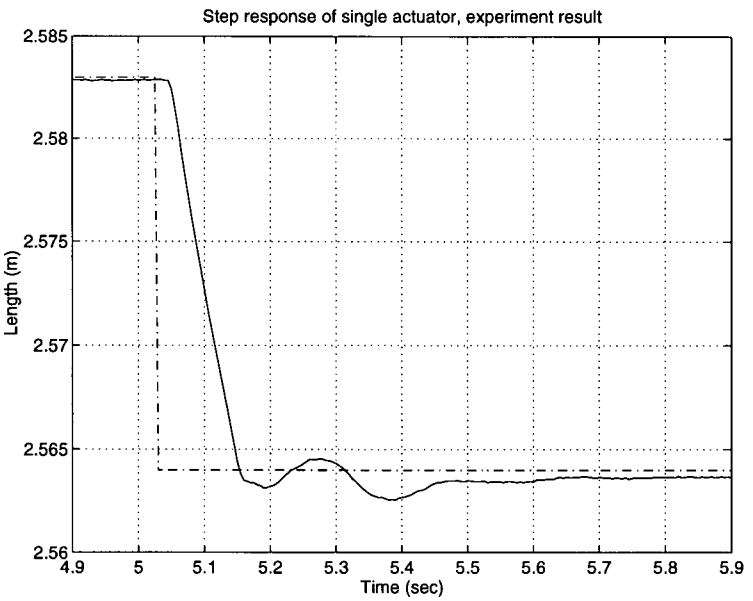

Figure 5.41: Step response when the pressure-feedback controller is used

5.4 Cartesian-space Controller

In this section, we discuss the design of Cartesian-space controller.

The desired trajectories of the Stewart platform are generally given in Cartesian-space. The feedback signals of the Stewart platform are generally obtained in link-space as actuator lengths/velocities. We can apply the inverse kinematics to derive the desired trajectories in link-space, which can be done off-line in some cases, and use the link-space controller. Or, we can apply the forward kinematics to derive the actual position/orientation of the Stewart platform in Cartesian-space, and use the Cartesian-space controller. The Cartesian-based control schemes perform more computations in the loop and may run at a lower sampling frequency, this is a drawback of using the Cartesian-based methods.

Block diagram of a simple Cartesian-space controller is shown in Figure 5.42. The error

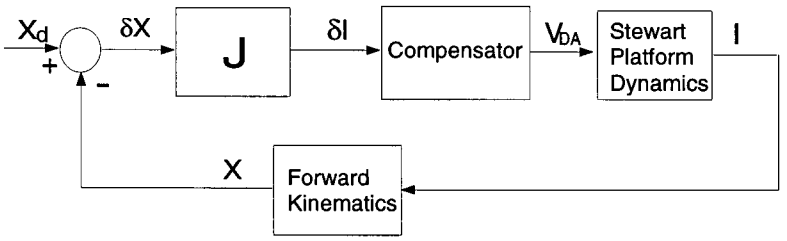

Figure 5.42: Simple Cartesian-space control block diagram

signal in Cartesian-space is transformed to link-space through the Jacobian matrix. When the compensator is the same as that in a link-space controller, the performance of this type of Cartesian-space controller is exactly the same as that of the link-space controller. For the control of the Stewart platform, where forward kinematics require more computation, this type of Cartesian-space controller is not recommended.

Another type of Cartesian-space controller makes use of the error signal in Cartesian-space to compute the desired force F in Cartesian-space and then transforms it to τ in link-space through the Jacobian matrix, see Figure 5.43. We can also apply the standard computed-torque control laws in Cartesian-space. But this type of controller requires actuators with force output,

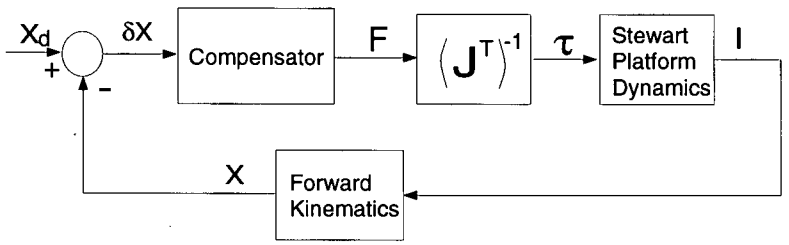

Figure 5.43: Cartesian-space controller

such as torque motors. For the electrohydraulic actuators we are using, the actuator dynamics can not be neglected, and the output force can not be controlled directly. So we were not able to apply the Cartesian-space controllers to our Stewart platform.

5.5 Motion Drive Algorithm

The ability of a motion simulator to produce accurate motion cues is determined by two factors, the physical constraints of the motion system and the characteristics of the motion drive algorithm [24]. As discussed in Section 2.3, our platform has a small motion envelope compared to the real excavator maneuvers being simulated, so it is necessary to modify the data from real excavator maneuvers by using a motion drive algorithm, which is commonly known as wash-out filter.

Wash-out filters attempt to generate the most accurate motion cue possible within the physical constraints of the motion system. The inputs to the wash-out filters should be translational accelerations and angular velocities. It is generally agreed that these signals are the elements of motion that are sensed by humans [24].

Classical wash-out filters are presently used by most aircraft simulator manufacturers and are relatively straightforward. Classical wash-out filters employ linear high-pass and low-pass filters along with input scaling and limiting to constrain the simulator motions to be within the capabilities of the motion system hardware. The scaling and filter parameters are fixed and are selected so that the simulator response to a worst case input is adequately constrained.

Diagram of a simple classical wash-out filter is presented in Figure 5.44.

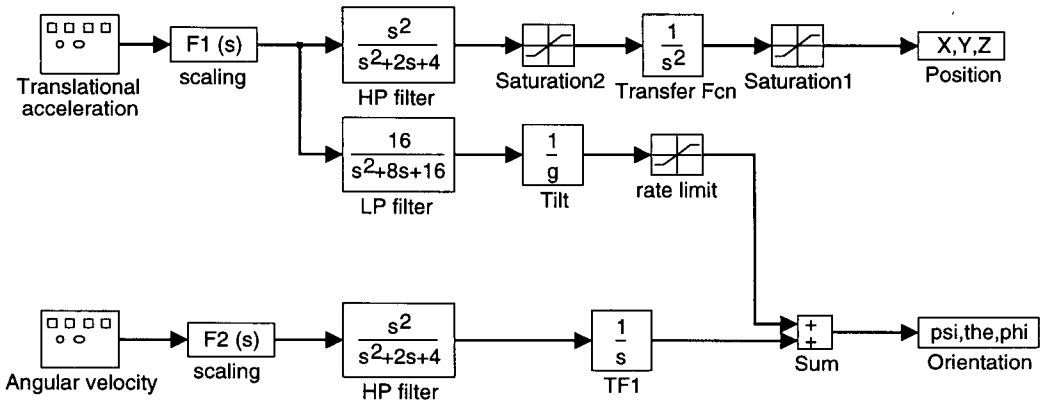

Figure 5.44: A classical wash-out filter configuration

The translational acceleration signal and the angular velocity signal are both scaled and high-pass filtered to make the motion cue remain in the safe workspace of the motion simulator.

There is a tilt-coordination channel passing the low-pass filtered translational acceleration to the rotation angles. The low frequency acceleration signals in X and Y direction are simulated by tilting the mobile platform and making use of the gravity to generate similar senses.

The above classical wash-out filter only gives a basic idea about the motion cue generation. There are other types of motion drive algorithms, which are more complex and have better performances. There is still considerable opportunity for the improvement of motion cue generation.

In this chapter, a high-performance controller using pressure feedback was developed. This controller was studied via both simulation and experiment. It was shown that the pressure-feedback controller can ensure the stability of the system. Experimental results indicated that the frequency response of the Stewart platform using pressure-feedback controller was very good. This intuitive and simple approach could be useful in a wide range of hydraulic applications.

Chapter 6

Simulation of Combined Dynamics

The purpose of the software simulation is to simulate the response of the Stewart platform on the computer for given controller parameters and a pre-planned trajectory. This can help us study the performances of different types of controllers. To simulate the motion of Stewart platform on the computer, we should combine the actuator dynamics with the rigid body dynamics.

6.1 Combined Dynamics

To simplify the derivation, we will not consider the leg dynamics, because the leg dynamics part is more complicated and less important than the mobile platform dynamics part, as discussed in Section 3.2. Then the derivation is basically the same as that in [3].

We rewrite the rigid body dynamics (3.38) as,

$$\mathbf{f} = \left(\mathbf{J}^T\right)^{-1} \left(\mathbf{D} \begin{bmatrix} {}^b\mathbf{a}_p \\ {}^b\alpha_p \end{bmatrix} + \mathbf{E} \right) \tag{6.77}$$

where $\mathbf{D}$ and $\mathbf{E}$ are defined as in (3.39) and (3.40).

The actuator dynamics are described by (4.48). We rewrite it for all the six actuators in a single vector,

$$M\ddot{\mathbf{l}} = \begin{bmatrix} \frac{\beta A}{V_{u1}} \left(C_d w \sqrt{\frac{2}{\rho}} h\left(X_{v1}, P_{c1}\right) - \dot{V}_1 \right) \\ \vdots \\ \frac{\beta A}{V_{u6}} \left(C_d w \sqrt{\frac{2}{\rho}} h\left(X_{v6}, P_{c6}\right) - \dot{V}_6 \right) \end{bmatrix} - B\ddot{\mathbf{l}} - \dot{\mathbf{f}} \tag{6.78}$$

where $\mathbf{l}$ is a six vector of the actuator lengths and $\mathbf{f}$ is a six vector of the actuator forces.

The derivative of (6.77) is given by,

$$\dot{\mathbf{f}} = -\left(\mathbf{J}^T\right)^{-1}\dot{\mathbf{J}}^T\left(\mathbf{J}^T\right)^{-1}\left(\mathbf{D}\begin{bmatrix}{}^b\mathbf{a}_p\\ {}^b\alpha_p\end{bmatrix}+\mathbf{E}\right)+\left(\mathbf{J}^T\right)^{-1}\left(\dot{\mathbf{D}}\begin{bmatrix}{}^b\mathbf{a}_p\\ {}^b\alpha_p\end{bmatrix}+\mathbf{D}\begin{bmatrix}{}^b\dot{\mathbf{a}}_p\\ {}^b\dot{\alpha}_p\end{bmatrix}+\dot{\mathbf{E}}\right) \tag{6.79}$$

where

$$\dot{\mathbf{D}} = \begin{bmatrix}0 & 0\\ 0 & \mathbf{S}\left({}^b\omega_p\right)\end{bmatrix}\mathbf{D}+\mathbf{D}\begin{bmatrix}0 & 0\\ 0 & \mathbf{S}^T\left({}^b\omega_p\right)\end{bmatrix} \tag{6.80}$$

and

$$\dot{\mathbf{E}} = \begin{bmatrix}0\\ 0\\ 0\\ {}^b\alpha_p\times\left({}^b\mathbf{I}_p{}^b\omega_p\right)+{}^b\omega_p\times\left({}^b\omega_p\times\left({}^b\mathbf{I}_p{}^b\omega_p\right)+{}^b\mathbf{I}_p{}^b\alpha_p\right)\end{bmatrix} \tag{6.81}$$

The Cartesian-space variables can be expressed in terms of the link-space variables, and vice versa. Detailed derivations can be found in Appendix C. If we express $\dot{\mathbf{f}}$ in terms of link-space variables and substitute it into (6.78), the combined dynamics in link-space can be expressed as

$$\begin{aligned}\dddot{\mathbf{l}} \;=\; & \left(M\mathbf{I}+\left(\mathbf{J}^T\right)^{-1}\mathbf{D}\mathbf{J}^{-1}\right)^{-1}\left\{\begin{bmatrix}\frac{\beta A}{V_{u1}}\left(C_d w\sqrt{\frac{2}{\rho}}h\left(X_{v1},P_{c1}\right)-\dot{V}_1\right)\\ \vdots\\ \frac{\beta A}{V_{u6}}\left(C_d w\sqrt{\frac{2}{\rho}}h\left(X_{v6},P_{c6}\right)-\dot{V}_6\right)\end{bmatrix}+\right.\\ & \left(-B+\left(\mathbf{J}^T\right)^{-1}\dot{\mathbf{J}}^T\left(\mathbf{J}^T\right)^{-1}\mathbf{D}-\left(\mathbf{J}^T\right)^{-1}\dot{\mathbf{D}}+2\left(\mathbf{J}^T\right)^{-1}\mathbf{D}\mathbf{J}^{-1}\dot{\mathbf{J}}\right)\mathbf{J}^{-1}\ddot{\mathbf{l}}+\\ & \left(\left(\mathbf{J}^T\right)^{-1}\dot{\mathbf{D}}-\left(\mathbf{J}^T\right)^{-1}\dot{\mathbf{J}}^T\left(\mathbf{J}^T\right)^{-1}\mathbf{D}-2\left(\mathbf{J}^T\right)^{-1}\mathbf{D}\mathbf{J}^{-1}\dot{\mathbf{J}}\right)\mathbf{J}^{-1}\dot{\mathbf{J}}\mathbf{J}^{-1}\dot{\mathbf{l}}+\\ & \left.\left(\mathbf{J}^T\right)^{-1}\mathbf{D}\mathbf{J}^{-1}\ddot{\mathbf{J}}\mathbf{J}^{-1}\dot{\mathbf{l}}+\left(\mathbf{J}^T\right)^{-1}\dot{\mathbf{J}}^T\left(\mathbf{J}^T\right)^{-1}\mathbf{E}-\left(\mathbf{J}^T\right)^{-1}\dot{\mathbf{E}}\right\}\end{aligned} \tag{6.82}$$

where $\mathbf{I}$ is a six by six identity matrix. The combined dynamics are expressed as a set of six coupled, third order differential equations where the inputs are the six valve spool positions $X_{v1}\ldots X_{v6}$. The valve position of each actuator is controlled by the command value V_{DA}, and the valve response can be approximately modeled as a second order closed-loop system. So the model of the Stewart platform is a fifth order nonlinear system.

It is also possible to model the Stewart platform in Cartesian-space, if we express $\dot{\mathbf{f}}$, $\ddot{\mathbf{l}}$, and $\ddot{\mathbf{l}}$ in terms of Cartesian-space variables in (6.78). That model would be useful for the simulation when applying the Cartesian-space controller.

We ran some simulations of the closed-loop system using link-space pressure-feedback controller. The simulation code was original written by Drexel in the C language [3], and was further modified by the author for the new control structure. The simulation results and the experiment data are compared in the next section.

6.2 Case Studies

We compare the simulation results and the experimental results through case studies in this section. The pressure-feedback controller with the loop gain of 1200 is used. Lengths of the six actuators (link-space response) are plotted for each case, with both simulation results and experimental results. The responses in the Cartesian-space are also plotted for the first two cases.

In the first case, we oscillate the platform along the x axis with an amplitude of 0.05 meter at 1.0 Hz. The simulation result is shown in Figure 6.45, while the experiment result is shown in Figure 6.46.

We also plot the Cartesian-space in Figure 6.47.

In the second case, we oscillate the platform about the z axis with an amplitude of 0.02 meter at 1.0 Hz. The simulation result and experiment result are plotted in Figure 6.48 and Figure 6.49, respectively. And the Cartesian-space tracking is shown in Figure 6.50. Please notice that Figure 6.50 shows the displacement along the z axis rather than the actual position variable z_p.

In the third case, we apply a 0.005 meter step input to the platform's x_p coordinate. The results are plotted in Figure 6.51 and Figure 6.52.

In the fourth case, we apply a 0.005 *radian* step input to the platform's ϕ coordinate. The results are plotted in Figure 6.53 and Figure 6.54.

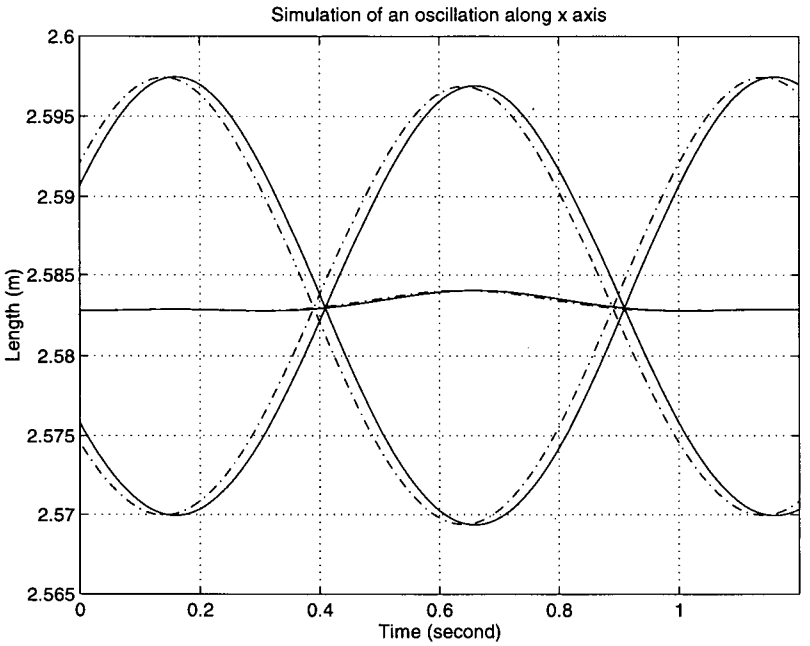

Figure 6.45: Case 1: Sine wave response along x axis, simulation

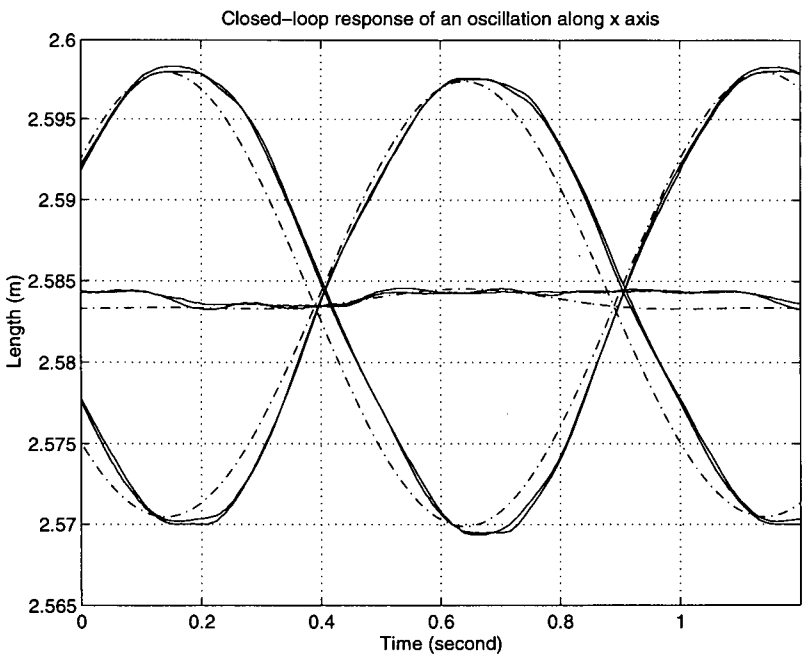

Figure 6.46: Case 1: Sine wave response along x axis, experiment

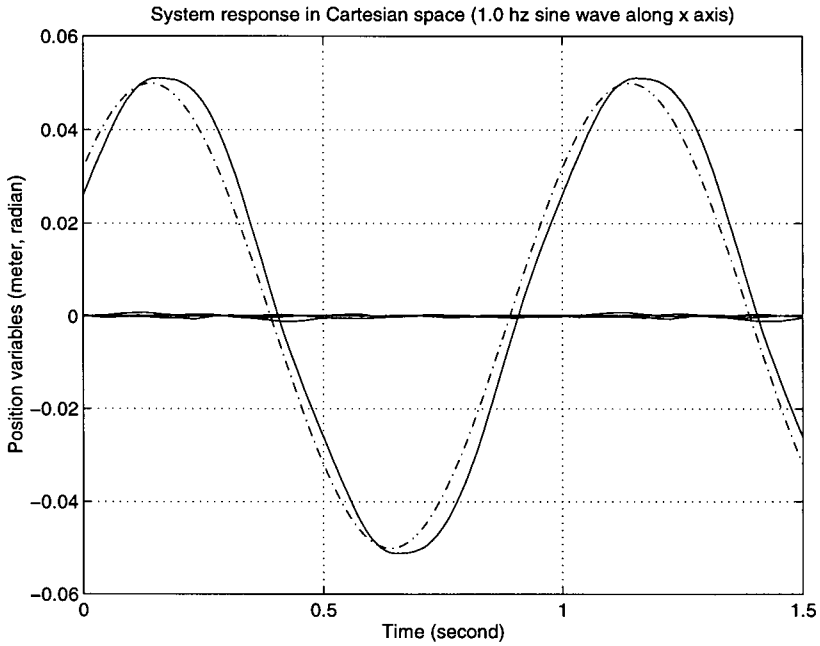

Figure 6.47: Case 1: Cartesian-space response

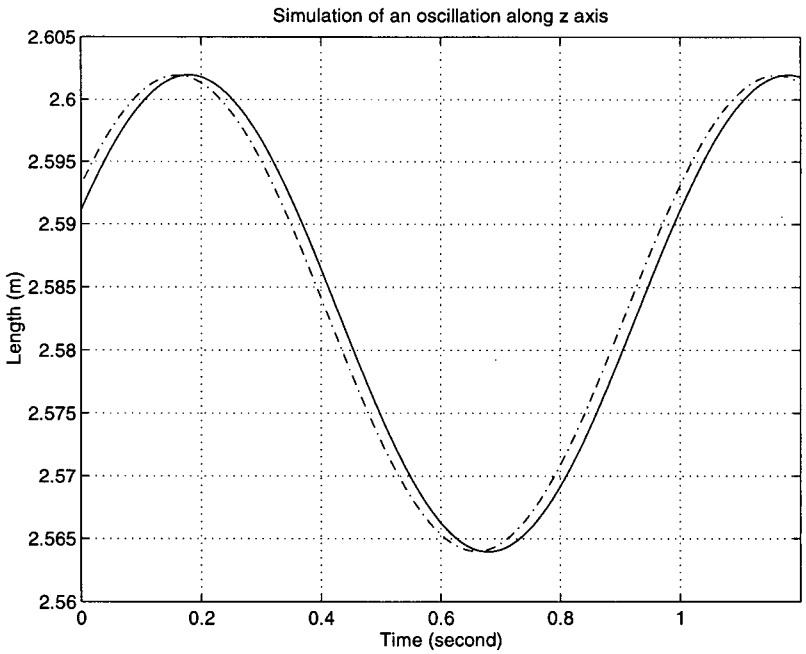

Figure 6.48: Case 2: Sine wave response along z axis, simulation

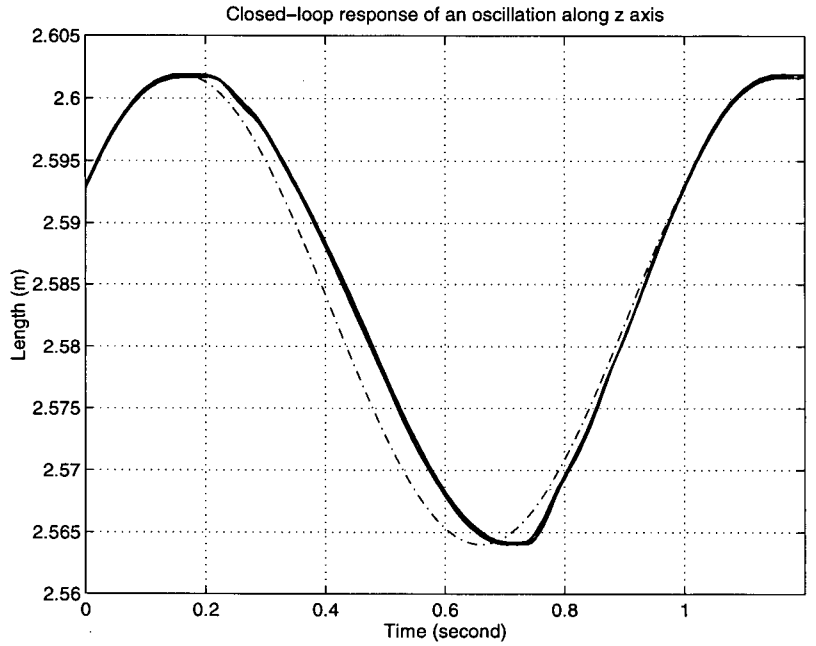

Figure 6.49: Case 2: Sine wave response along z axis, experiment

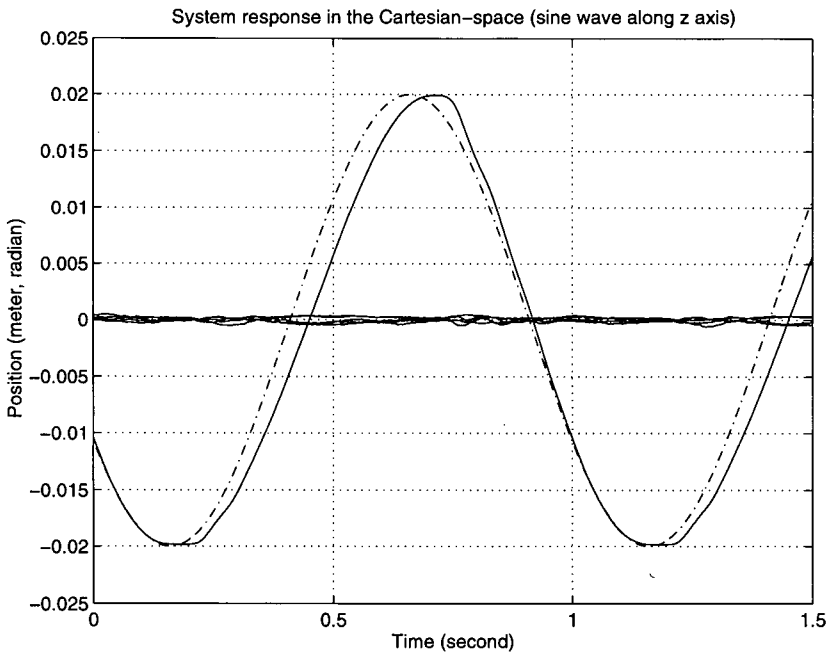

Figure 6.50: Case 2: Cartesian-space response

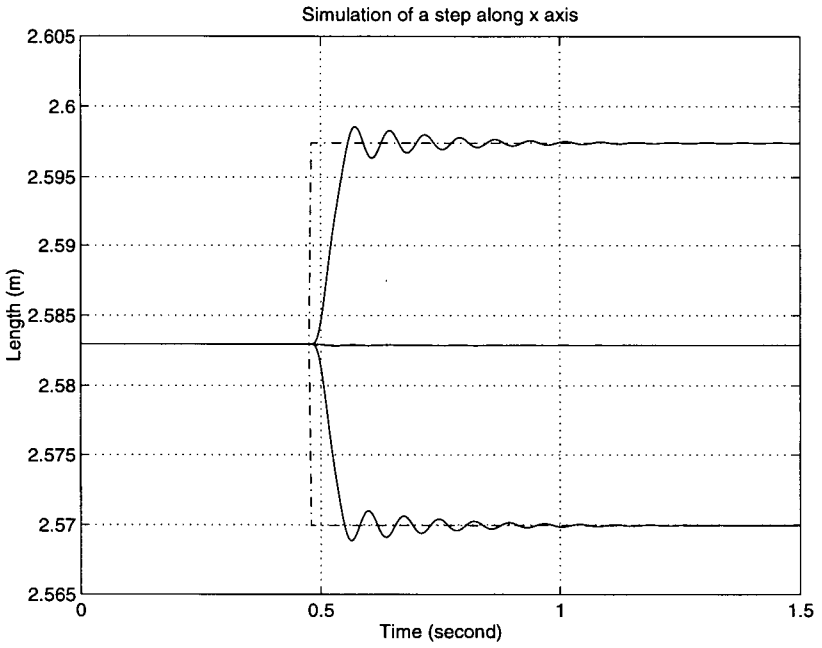

Figure 6.51: Case 3: Step response of x, simulation

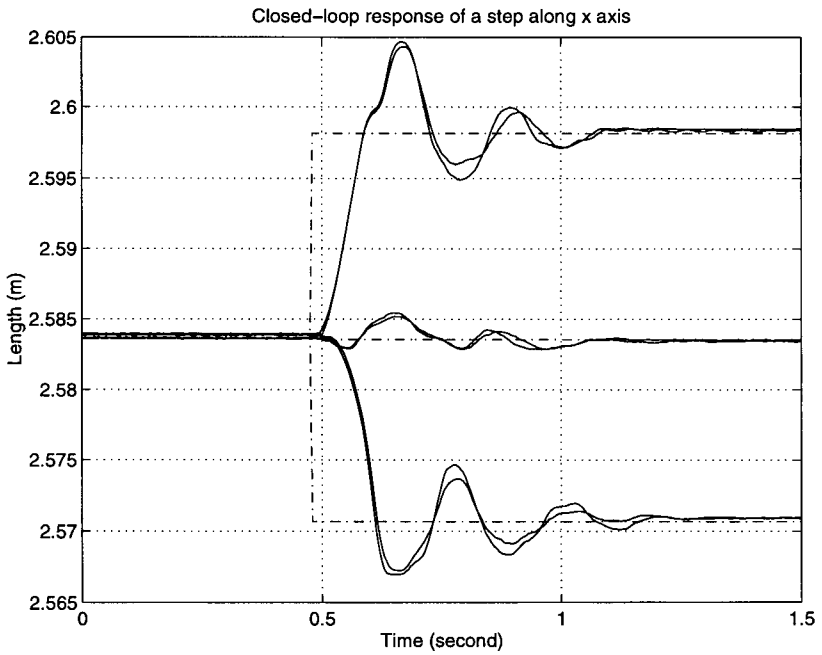

Figure 6.52: Case 3: Step response of x, experiment

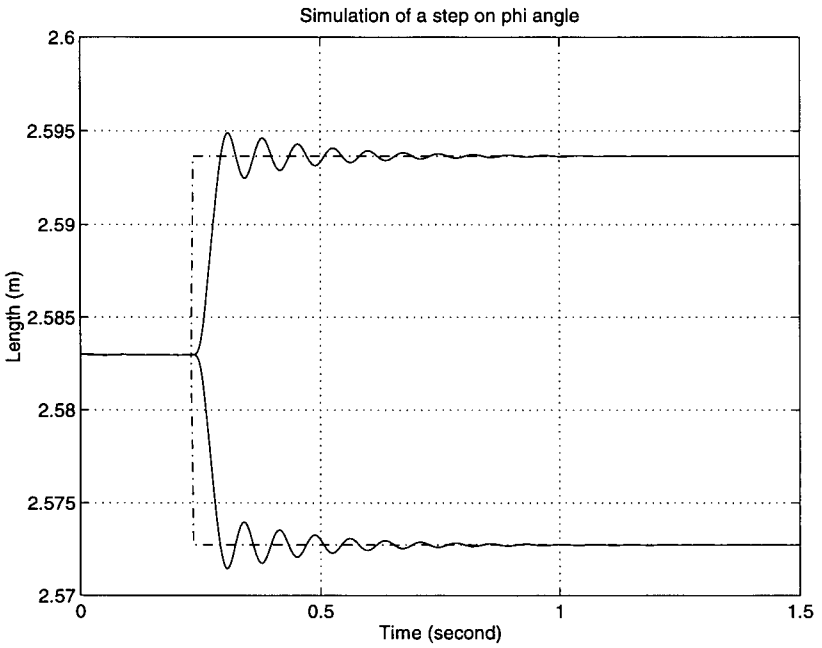

Figure 6.53: Case 4: Step response of roll angle, simulation

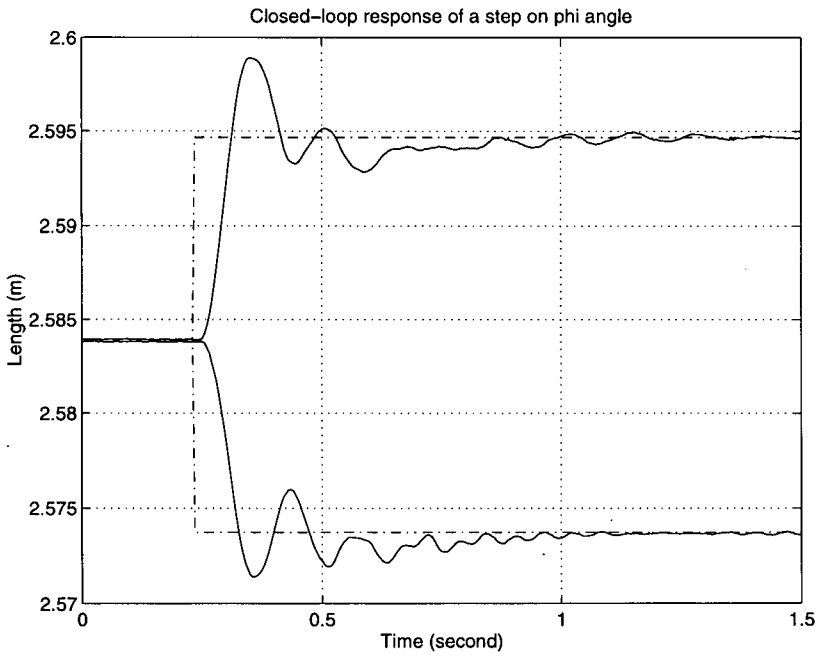

Figure 6.54: Case 4: Step response of roll angle, experiment

Chapter 7

Conclusion and Future works

In this thesis we have presented the modeling, simulation, and control of a hydraulically actuated, ceiling-mounted Stewart platform, which is designed to be a motion simulator.

The mechanical constraints, singularities, and the length limits of the actuators were studied. A cylindrical workspace where the system is free of mechanical constraints and singularities was obtained for control system development. We applied Newton's method to solve the platform's forward kinematics and concluded that the result would reach the desired value after two iterations in the real-time control.

We also presented a derivation of the Stewart platform's complete rigid body dynamics. The derivation was straightforward and the result was relatively simple. We then showed through simulations that the leg dynamics can be neglected in this particular platform design. This simplified the simulation work and real-time control.

A model of the electrohydraulic actuator was derived and then linearized for the purpose of simulation and control. The model was validated using experimental data. Results showed that the linearized actuator model and the experimental data fit each other well. This model is the base of the design of link-space controllers.

Performance of the proportional gain controller was studied. The system response could be improved by pre-filtering the desired trajectory. To ensure the stability of the system, a pressure-feedback controller was proposed. The study showed that this pressure-feedback controller could stabilize the system, and it improved the system performance by allowing a higher loop gain. This intuitive and simple approach could be useful in a wide range of hydraulics applications. The preliminary experimental results indicated that the frequency

response of our Stewart platform using pressure-feedback control is very good.

Some suggestions for further work include the following:

- Improve the valve response by applying the closed-loop control based on the valve spool position
- Investigate the use of on-line identification to determine the bulk modulus value and therefore the pressure gain
- Study the robustness of the proposed pressure-feedback control to measurement errors and experimentally determine the system stability margins
- Examine analytically and experimentally the effects of leg flexibility and platform rigid-body dynamics on the link-space controllers
- Further investigate the possibility of applying Cartesian-space controllers
- Develop code for motion drive algorithms and simulate the excavator motion

Bibliography

[1] D. Stewart,"A Platform with Six Degrees of Freedom," *Proceedings of the Institution of Mechanical Engineers*, 180(15), 371-386, 1965-1966.

[2] S.E. Salcudean et al, "A Six Degree-of-Freedom, Hydraulic, One Person Motion Simulator," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 2437-2443, 1994.

[3] P.A. Drexel, "A Six Degree-of-Freedom, Hydraulic, One Person Motion Simulator," Master's thesis, The University of British Columbia, 1992.

[4] V.E. Gough, "Contribution to discussion to papers on research in automobile stability and control in tyre performance, by Cornell staff," *Proc. Auto. Div. Inst. mech. Engineers*, 171, 392-394, 1956-1957.

[5] V.E. Gough and S. G. Whitehall, "Universal Tyre Test Machine," *Proceedings, Ninth International Technical Congress FISITA*, 117-137, 1962.

[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," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation* , 1873-1878, 1991.

[7] Kevin Cleary and Tatsuo Arai, "A Prototype Parallel Manipulator: Kinematics, Construction, Software, Workspace Results, and Singularity Analysis," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation* , 566-571, 1991.

[8] E.F. Fichter, "Kinematics of a Parallel Connection Manipulator," *ASME Design Engineering Technical Conference* , Paper Number 84-DET-45, 1984.

[9] W.Q.D. Do and D.C.H. Yang, "Inverse Dynamic Analysis and Simulation of a Platform Type of Robot", *ASME Design Engineering Technical Conference*, Paper Number 86-DET-94, 1986.

[10] 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), 209-227, 1988.

[11] E.F. Fichter, "A Stewart Platform Based Manipulator: General Theory and Practical Construction," *International Journal of Robotics Research* , 5(2), 157-181, 1986.

[12] C. Gosselin, "Determination of the workspace of 6-dof parallel manipulators," *Journal of Mechanical Design* , 112(3), 331-336, 1990.

[13] J-P. Merlet, "Trajectory Verification in the Workspace for Parallel Manipulators," *The International Journal of Robotics Research*, 13(4), 326-333, 1994.

[14] K.H. Hunt, *Kinematics geometry of mechanisms*, Clarendon Press, Oxford, 1978.

[15] J-P. Merlet, "Singular Configurations of Parallel Manipulators and Grassmann Geometry" *The International Journal of Robotics Research*, 8(5), 45-56, 1989.

[16] Prabjot Nanua and Kenneth J. Waldron, "Direct Kinematic Solution of a Stewart Platform," *IEEE Transactions on Robotics and Automation* , 431-436, 1989.

[17] M. Griffis and J. Duffy, "A Forward Displacement Analysis of a Class of Stewart Platforms," *Journal of Robotic Systems* , 6(6), 703-720, 1989.

[18] 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.

[19] 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.

[20] J.-P. Merlet, "Direct Kinematics and Assembly Modes of Parallel Manipulators," *International Journal of Robotics Research* , 11(2), 150-162, 1992.

[21] 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.

[22] K. Liu, Mick Fitzgerald, Darren M. Dawson, Frank L. Lewis, "Modeling and Control of a Stewart Platform Manipulator," *American Society of Mechanical Engineers, Dynamic Systems and Control Division (Publication) DSC* v 33, 83-89, 1991.

[23] C.D. Zhang and S.M. Song, "An Efficient Method for Inverse Dynamics of Manipulators Based on the Virtual Work Principle," *Journal of Robotic Systems*, 10(5), 605-627, 1993.

[24] L.D. Reid and M.A. Nahon, "Response of Airline Pilots to Variations in Flight Simulator Motion Algorithms," *Journal of Aircraft*, 25(7), 639-646, 1988.

[25] K. Liu, F. Lewis, G. Lebret, and D. Taylor, "The Singularities and Dynamics of a Stewart Platform Manipulator," *Journal of Intelligent and Robotic Systems*, 8, 287-308, 1993.

[26] L.C.T. Wang and C.C. Chen, "On the Numerical Kinematic Analysis of General Parallel Robotic Manipulators," *IEEE Transactions on Robotics and Automation*, 9(3), 272-285, 1993

[27] Z. Ji, "Workspace Analysis of Stewrat Platforms via Vertex Space," *Journal of Robotic Systems*, 11(7), 631-639, 1994.

[28] H. Pang and M. Shahinpoor, "Inverse Dynamics of a Parallel Manipulator," *Journal of Robotic Systems*, 11(8), 693-702, 1994.

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

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

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

[32] Athanasios Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill Book Company, New York, 1965.

[33] J.-J. E. Slotine and Weiping Li, *Applied Nonlinear Control*, Prentice-Hall Inc., New Jersey, 1991.

Appendix A

Derivatives of Jacobian

We define the Jacobian matrix of Stewart platform as,

$$\dot{\mathbf{l}} = \mathbf{J} \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \tag{A.83}$$

where $\dot{\mathbf{l}}$ is the six vector of actuator velocities, $\mathbf{J}$ is the platform's six by six Jacobian matrix, ${}^b\mathbf{v}_p = {}^b\dot{\mathbf{d}}_p$ is the three vector of the platform's translational velocity and ${}^b\omega_p$ is the three vector of the platform's angular velocity.

Then, we can get the Jacobian matrix as in Section 2.2

$$\dot{\mathbf{l}} = \begin{bmatrix} \dot{l_A} \\ \dot{l_B} \\ \dot{l_C} \\ \dot{l_D} \\ \dot{l_E} \\ \dot{l_F} \end{bmatrix} = \begin{bmatrix} \left({}^b\mathbf{R}_p{}^p\mathbf{p}_A + {}^b\mathbf{d}_p - {}^b\mathbf{b}_A\right)^T & \left(({}^b\mathbf{R}_p{}^p\mathbf{p}_A) \times ({}^b\mathbf{d}_p - {}^b\mathbf{b}_A)\right)^T \\ \left({}^b\mathbf{R}_p{}^p\mathbf{p}_B + {}^b\mathbf{d}_p - {}^b\mathbf{b}_B\right)^T & \left(({}^b\mathbf{R}_p{}^p\mathbf{p}_B) \times ({}^b\mathbf{d}_p - {}^b\mathbf{b}_B)\right)^T \\ \vdots & \vdots \\ \left({}^b\mathbf{R}_p{}^p\mathbf{p}_F + {}^b\mathbf{d}_p - {}^b\mathbf{b}_F\right)^T & \left(({}^b\mathbf{R}_p{}^p\mathbf{p}_F) \times ({}^b\mathbf{d}_p - {}^b\mathbf{b}_F)\right)^T \end{bmatrix} \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \tag{A.84}$$

or,

$$\dot{\mathbf{l}} = \mathbf{J}\left({}^b\mathbf{d}_p, {}^b\mathbf{R}_p\right) \begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \tag{A.85}$$

where the Jacobian matrix depends on the platform's position ${}^b\mathbf{d}_p$ and orientation ${}^b\mathbf{R}_p$.

The first and second derivatives of the Jacobian matrix are also useful for analyzing the platform's overall dynamics. The following are derivations of these derivatives. From (A.84), we can get

$$\mathbf{J}_i^T = \frac{1}{l_i} \begin{bmatrix} {}^b\mathbf{R}_p{}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i \\ \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) \end{bmatrix} \tag{A.86}$$

Using the quotient rule, we take the derivative of (A.86) with respect to time, and get

$$\dot{\mathbf{J}}_i^T = \frac{1}{l_i^2}\left\{ l_i \begin{bmatrix} {}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\mathbf{v}_p \\ -{}^b\mathbf{v}_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) \end{bmatrix} - \begin{bmatrix} {}^b\mathbf{R}_p{}^p\mathbf{p}_i + {}^b\mathbf{d}_p - {}^b\mathbf{b}_i \\ \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) \end{bmatrix} \dot{l}_i \right\} \tag{A.87}$$

Equation (A.87) can be simplified using (A.86) to give

$$\dot{\mathbf{J}}_i^T = \frac{\begin{bmatrix} {}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\mathbf{v}_p \\ -{}^b\mathbf{v}_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) \end{bmatrix} - \mathbf{J}^T\dot{l}_i}{l_i} \tag{A.88}$$

Finally, taking the transpose of (A.88), we have an expression for each row of the first derivative of the platform's Jacobian matrix,

$$\dot{\mathbf{J}}_i\left({}^b\mathbf{d}_p, {}^b\mathbf{R}_p, {}^b\mathbf{v}_p, {}^b\omega_p, \dot{l}_i\right) = \frac{\begin{bmatrix} {}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\mathbf{v}_p \\ -{}^b\mathbf{v}_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) \end{bmatrix}^T - \mathbf{J}_i\dot{l}_i}{l_i} \tag{A.89}$$

To derive the second derivative of the Jacobian matrix we start with (A.88) and use the quotient rule as before, giving

$$\begin{aligned} \ddot{\mathbf{J}}_i^T = \frac{1}{l_i^2}\Bigg\{ l_i \Bigg\{ \begin{bmatrix} {}^b\alpha_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\omega_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) + {}^b\mathbf{a}_p \\ -{}^b\mathbf{a}_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) - 2{}^b\mathbf{v}_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) + \\ \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\alpha_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + \right. \\ \left. {}^b\omega_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right)\right) \end{bmatrix} - \dot{\mathbf{J}}_i^T\dot{l}_i - \mathbf{J}^T\ddot{l}_i \Bigg\} \\ - \left\{ \begin{bmatrix} {}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\mathbf{v}_p \\ -{}^b\mathbf{v}_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) \end{bmatrix} - \mathbf{J}_i^T\dot{l}_i \right\} \dot{l}_i \Bigg\} \end{aligned} \tag{A.90}$$

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.

Equation (A.90) is simplified using (A.88) to give

$$\ddot{\mathbf{J}}_i^T = \frac{\begin{bmatrix} {}^b\alpha_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\omega_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) + {}^b\mathbf{a}_p \\ -{}^b\mathbf{a}_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) - 2{}^b\mathbf{v}_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) + \\ \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\alpha_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\omega_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right)\right) \end{bmatrix} - 2\dot{\mathbf{J}}_i^T\dot{l}_i - \mathbf{J}^T\ddot{l}_i}{l_i} \tag{A.91}$$

Finally, taking the transpose of (A.91), we have an expression for each row of the second derivative of the platform's Jacobian matrix

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

$$\frac{\begin{bmatrix} {}^b\alpha_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\omega_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) + {}^b\mathbf{a}_p \\ -{}^b\mathbf{a}_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) - 2{}^b\mathbf{v}_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right) + \\ \left({}^b\mathbf{b}_i - {}^b\mathbf{d}_p\right) \times \left({}^b\alpha_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right) + {}^b\omega_p \times \left({}^b\omega_p \times \left({}^b\mathbf{R}_p{}^p\mathbf{p}_i\right)\right)\right) \end{bmatrix}^T - 2\dot{\mathbf{J}}_i\dot{l}_i - \mathbf{J}_i\ddot{l}_i}{l_i} \tag{A.92}$$

Appendix B

Spectral Analysis

For a stationary stochastic process $\mathbf{x}(t)$, we can write the mean $E\{\mathbf{x}(t)\}$ and autocorrelation

$$R(\tau) = R_{xx}(\tau) = E\{\mathbf{x}(t+\tau)\mathbf{x}^*(t)\} \tag{B.93}$$

The cross-correlation of two jointly stationary processes is

$$R_{xy}(\tau) = E\{\mathbf{x}(t+\tau)\mathbf{y}^*(t)\} = R^*_{yx}(-\tau) \tag{B.94}$$

The power spectral density (also called power spectrum) $S(\omega)$ or $S_{xx}(\omega)$ of a process $\mathbf{x}(t)$ is the Fourier transform of its autocorrelation:

$$S(\omega) = \int_{-\infty}^{\infty} R(\tau)e^{-j\omega\tau}d\tau \tag{B.95}$$

Since $R(-\tau) = R^*(\tau)$, we easily conclude from the above that $S(\omega)$ is a real function.

The cross spectral density $S_{xy}(\omega)$ of two process $\mathbf{x}(t)$ and $\mathbf{y}(t)$ is the Fourier transform of their cross-correlation:

$$S_{xy}(\omega) = \int_{-\infty}^{\infty} R_{xy}(\tau)e^{-j\omega\tau}d\tau = S^*_{yx}(\omega) \tag{B.96}$$

We now study a given linear system with impulse response function $h(t)$. When a process $\mathbf{x}(t)$ is applied to the input of this system, the resulting output $\mathbf{y}(t)$ is given by,

$$\mathbf{y}(t) = \int_{-\infty}^{\infty} \mathbf{x}(t-\alpha)h(\alpha)d\alpha = \int_{-\infty}^{\infty} \mathbf{x}(\alpha)h(t-\alpha)d\alpha \tag{B.97}$$

The cross-correlation between $\mathbf{y}(t)$ and $\mathbf{x}(t)$ can then be expressed as

$$\begin{aligned} R_{yx}(\tau) &= E\{\mathbf{y}(t)\mathbf{x}^*(t-\tau)\}, \ \textit{assuming stationariness} \\ &= E\{\int_{-\infty}^{\infty} \mathbf{x}(t-\alpha)\mathbf{x}^*(t-\tau)h(\alpha)d\alpha\} \end{aligned}$$

$$\begin{aligned} &= \int_{-\infty}^{\infty} E\{\mathbf{x}(t-\alpha)\mathbf{x}^*(t-\tau)\}h(\alpha)d\alpha \\ &= \int_{-\infty}^{\infty} R_{xx}[(t-\alpha)-(t-\tau)]h(\alpha)d\alpha \\ &= \int_{-\infty}^{\infty} R_{xx}(\tau-\alpha)h(\alpha)d\alpha \\ &= R_{xx}(\tau) * h(\tau) \end{aligned} \tag{B.98}$$

Similarly, we can obtain,

$$R_{yy}(\tau) = R_{yx}(\tau) * h^*(-\tau) \tag{B.99}$$

If $\mathbf{x}(t)$ is white noise, i.e. if $R_{xx}(\tau) = \delta(\tau)$, and $h(t) = 0$ for $t < 0$ (real causal system), then,

$$R_{yx}(\tau) = h(\tau) \tag{B.100}$$

and transfer function of the system when $s = j\omega$ can be expressed as

$$H(j\omega) = \mathcal{L}\{h(t)\}|_{s=j\omega} = \int_0^{\infty} h(t)e^{-j\omega t}dt = \int_{-\infty}^{\infty} R_{yx}(t)e^{-j\omega t}dt = S_{yx}(\omega) \tag{B.101}$$

Similarly, we can also obtain,

$$|H(j\omega)|^2 = S_{yy}(\omega) \tag{B.102}$$

By using white noise, which has a evenly distributed power spectral density, as input signal and analyzing the cross spectral density between output signal and input signal of the system, we can get an experimentally estimated transfer function of the system.

Appendix C

Transformations between Cartesian-space variables and link-space variables

In the simulation and control of the Stewart platform, transformations between Cartesian-space variables and link-space variables are generally needed. We just summarize these transformations here.

The transformation between the platform configuration $\mathbf{X}$ and the actuator lengths $\mathbf{l}$ can be done through inverse/forward kinematics.

$$\mathbf{X} = forward\_kinematics(\mathbf{l}) \tag{C.103}$$

and

$$\mathbf{l} = inverse\_kinematics(\mathbf{X}) \tag{C.104}$$

And Jacobian matrix $\mathbf{J}$ can be derived from the platform configuration.

The velocity level transformations are

$$\begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} = \mathbf{J}^{-1}\dot{\mathbf{l}} \tag{C.105}$$

and

$$\dot{\mathbf{l}} = \mathbf{J}\begin{bmatrix} {}^b\mathbf{v}_p \\ {}^b\omega_p \end{bmatrix} \tag{C.106}$$

And $\dot{\mathbf{J}}$ can be derived as in the previous section.

The acceleration level transformations are

$$\begin{bmatrix} {}^b\mathbf{a}_p \\ {}^b\alpha_p \end{bmatrix} = \mathbf{J}^{-1}\ddot{\mathbf{l}} - \mathbf{J}^{-1}\dot{\mathbf{J}}\mathbf{J}\dot{\mathbf{l}} \tag{C.107}$$

and

$$\ddot{\mathbf{l}} = \dot{\mathbf{J}} \begin{bmatrix} {}^{b}\mathbf{v}_p \\ {}^{b}\omega_p \end{bmatrix} + \mathbf{J} \begin{bmatrix} {}^{b}\mathbf{a}_p \\ {}^{b}\alpha_p \end{bmatrix} \tag{C.108}$$

And $\ddot{\mathbf{J}}$ can also be derived as in the previous section.

Appendix D

Matlab Source Code

D.1 InvKinematics.m

```
function [length] = invkinematics(x,y,z,psi,the,phi)

% Define the system parameters

gammap1 = 53.45;              % unit in degree
gammap = [gammap1,120-gammap1, 120+gammap1, -120-gammap1,-120+gammap1,-gammap1];
gammab1 = 7.75;               % unit in degree
gammab = [gammab1,120-gammab1, 120+gammab1, -120-gammab1,-120+gammab1,-gammab1];

rp = 0.668;        % radius of platform, in meter
rb = 1.133;        % radius of base, in meter
hp = 0.203;        % platform nominal height, in meter

spsi = sin(psi*pi/180);
cpsi = cos(psi*pi/180);
sthe = sin(the*pi/180);
cthe = cos(the*pi/180);
sphi = sin(phi*pi/180);
cphi = cos(phi*pi/180);

% rotation matrix R
bRp(1,1) =  cphi * cthe;
bRp(1,2) = -sphi * cpsi + cphi * sthe * spsi;
bRp(1,3) =  sphi * spsi + cphi * sthe * cpsi;
bRp(2,1) =  sphi * cthe;
bRp(2,2) =  cphi * cpsi + sphi * sthe * spsi;
bRp(2,3) = -cphi * spsi + sphi * sthe * cpsi;
bRp(3,1) = -sthe;
bRp(3,2) =  cthe * spsi;
bRp(3,3) =  cthe * cpsi;

% fixed vector of pp
pp(1,:) = rp*cos(gammap*pi/180);
pp(2,:) = rp*sin(gammap*pi/180);
pp(3,:) = [hp/2,hp/2,hp/2,hp/2,hp/2,hp/2];
```

```
% fixed vector of bb
bb(1,:) = rb*cos(gammab*pi/180);
bb(2,:) = rb*sin(gammab*pi/180);
bb(3,:) = [0, 0, 0, 0, 0, 0];

% bdp
bdp = [x,y,z]';
bDp = [bdp,bdp,bdp,bdp,bdp,bdp];

% actuator vector
bA = bRp*pp +bDp - bb;

for i =1:6,
length(i)  = norm(bA(:,i));
end
```

D.2 FwdKinematics.m

```
% System parameters

gammap1 = 53.45;
gammap = [gammap1,120-gammap1, 120+gammap1, -120-gammap1,-120+gammap1,-gammap1];
gammab1 = 7.75;
gammab = [gammab1,120-gammab1, 120+gammab1, -120-gammab1,-120+gammab1,-gammab1];

rp = 0.668;
rb = 1.133;
hp = 0.203;

pp(1,:) = rp*cos(gammap*pi/180);
pp(2,:) = rp*sin(gammap*pi/180);
pp(3,:) = [hp/2,hp/2,hp/2,hp/2,hp/2,hp/2];

bb(1,:) = rb*cos(gammab*pi/180);
bb(2,:) = rb*sin(gammab*pi/180);
bb(3,:) = [0, 0, 0, 0, 0, 0];

% Estimated value of Cartesian space variables
Assumed = [ 0 0 -2.5 0 0 0];
Assumedp = Assumed';

% Number of measured points
endii=80;

Finalp = zeros(endii,6);

for ii = 1:endii,

% Measured actuator lengths
Givenl = [ currLengthA(ii,1),  currLengthB(ii,1),  currLengthC(ii,1), ...
```

```
          currLengthD(ii,1),  currLengthE(ii,1),  currLengthF(ii,1) ];
%Givenl = [ desLengthA(ii,1),  desLengthB(ii,1),  desLengthC(ii,1),  ...
%           desLengthD(ii,1),  desLengthE(ii,1),  desLengthF(ii,1) ];

for irr = 1:2,

x = Assumedp(1); y = Assumedp(2); z = Assumedp(3);
psi = Assumedp(4); the = Assumedp(5); phi = Assumedp(6);

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

% Rotation matrix
bRp(1,1) =  cphi * cthe;
bRp(1,2) = -sphi * cpsi + cphi * sthe * spsi;
bRp(1,3) =  sphi * spsi + cphi * sthe * cpsi;
bRp(2,1) =  sphi * cthe;
bRp(2,2) =  cphi * cpsi + sphi * sthe * spsi;
bRp(2,3) = -cphi * spsi + sphi * sthe * cpsi;
bRp(3,1) = -sthe;
bRp(3,2) =  cthe * spsi;
bRp(3,3) =  cthe * cpsi;

bdp = [x,y,z]';
bDp = [bdp,bdp,bdp,bdp,bdp,bdp];

% Inverse kinematics of estimated Cartesian space variables
bA = bRp*pp + bDp - bb;
for i = 1:6,
length(i)  = norm(bA(:,i));
end

B(1,1) = cthe * cphi;
B(1,2) = -sphi;
B(1,3) = 0;
B(2,1) = cthe * sphi;
B(2,2) = cphi;
B(2,3) = 0;
B(3,1) = -sthe;
B(3,2) = 0;
B(3,3) = 1;

% Jacobian
for i = 1:6,
J(i,:) = [ ( bRp*pp(:,i) + bdp - bb(:,i) )' ...
          (cross( bRp*pp(:,i), bdp-bb(:,i) ))' ] / length(i) ;
end;
```

```
% Forward kinematics
Assumedp = Assumedp - inv( J * [eye(3) zeros(3,3) ; zeros(3,3) B ] ) * ...
          (length' - Givenl');

end

% Result of one measured point
Finalp(ii,:) = Assumedp';

end

figure;
t=0:0.005:endii*0.005-0.005;
plot(t', Finalp);
```

D.3 Singularities.m

```
function [SVRb,SVRp] = singu(x,y,z,psi,the,phi)

% System parameters

gammap1 = 53.45;
gammap = [gammap1,120-gammap1, 120+gammap1, -120-gammap1,-120+gammap1,-gammap1];
gammab1 = 7.75;
gammab = [gammab1,120-gammab1, 120+gammab1, -120-gammab1,-120+gammab1,-gammab1];

alphap = [0, 120, 120, -120, -120, 0];
betab = [60,60,180,180,-60,-60];

% U-joint angles
jointb = -45;
Jointb = [jointb,jointb,jointb,jointb,jointb,jointb];
jointp = 45;  %45 for chair, 30 for triangle.
Jointp = [jointp,jointp,jointp,jointp,jointp,jointp];

rp = 0.668;
rb = 1.133;
hp = 0.203;

spsi = sin(psi*pi/180);
cpsi = cos(psi*pi/180);
sthe = sin(the*pi/180);
cthe = cos(the*pi/180);
sphi = sin(phi*pi/180);
cphi = cos(phi*pi/180);

bRp(1,1) =  cphi * cthe;
bRp(1,2) = -sphi * cpsi + cphi * sthe * spsi;
bRp(1,3) =  sphi * spsi + cphi * sthe * cpsi;
bRp(2,1) =  sphi * cthe;
```

```
bRp(2,2) =  cphi * cpsi + sphi * sthe * spsi;
bRp(2,3) = -cphi * spsi + sphi * sthe * cpsi;
bRp(3,1) = -sthe;
bRp(3,2) =  cthe * spsi;
bRp(3,3) =  cthe * cpsi;

pp(1,:) = rp*cos(gammap*pi/180);
pp(2,:) = rp*sin(gammap*pi/180);
pp(3,:) = [hp/2,hp/2,hp/2,hp/2,hp/2,hp/2];

bb(1,:) = rb*cos(gammab*pi/180);
bb(2,:) = rb*sin(gammab*pi/180);
bb(3,:) = [0, 0, 0, 0, 0, 0];

bdp = [x,y,z]';
bDp = [bdp,bdp,bdp,bdp,bdp,bdp];

bA = bRp*pp +bDp - bb;

% axis vectors of the upper U-joints
bU(1,:) = rb*cos(betab*pi/180);
bU(2,:) = rb*sin(betab*pi/180);
bU(3,:) = rb*tan(Jointb*pi/180);

% axis vectors of the lower U-joints
pL(1,:) = rp*cos(alphap*pi/180);
pL(2,:) = rp*sin(alphap*pi/180);
pL(3,:) = rp*tan(Jointp*pi/180);

bL = bRp*pL;

% Singularity Value Ratios. unit in 1/rad.
% SVR > 5 is according to angle < 11.5 degree.

SVRb(1) = 1 / acos(  ( bU(:,1)/norm(bU(:,1)) )' * ( bA(:,1)/norm(bA(:,1)) )  );
SVRb(2) = 1 / acos(  ( bU(:,2)/norm(bU(:,2)) )' * ( bA(:,2)/norm(bA(:,2)) )  );
SVRb(3) = 1 / acos(  ( bU(:,3)/norm(bU(:,3)) )' * ( bA(:,3)/norm(bA(:,3)) )  );
SVRb(4) = 1 / acos(  ( bU(:,4)/norm(bU(:,4)) )' * ( bA(:,4)/norm(bA(:,4)) )  );
SVRb(5) = 1 / acos(  ( bU(:,5)/norm(bU(:,5)) )' * ( bA(:,5)/norm(bA(:,5)) )  );
SVRb(6) = 1 / acos(  ( bU(:,6)/norm(bU(:,6)) )' * ( bA(:,6)/norm(bA(:,6)) )  );

SVRp(1) = 1 / acos(  ( bL(:,1)/norm(bL(:,1)) )' * ( -bA(:,1)/norm(bA(:,1)) )  );
SVRp(2) = 1 / acos(  ( bL(:,2)/norm(bL(:,2)) )' * ( -bA(:,2)/norm(bA(:,2)) )  );
SVRp(3) = 1 / acos(  ( bL(:,3)/norm(bL(:,3)) )' * ( -bA(:,3)/norm(bA(:,3)) )  );
SVRp(4) = 1 / acos(  ( bL(:,4)/norm(bL(:,4)) )' * ( -bA(:,4)/norm(bA(:,4)) )  );
SVRp(5) = 1 / acos(  ( bL(:,5)/norm(bL(:,5)) )' * ( -bA(:,5)/norm(bA(:,5)) )  );
SVRp(6) = 1 / acos(  ( bL(:,6)/norm(bL(:,6)) )' * ( -bA(:,6)/norm(bA(:,6)) )  );
```

D.4 Dynamics.m

```
% Cartesian space variables
x=0; y=0; z=-2.5;
dotx = 0; doty = 0; dotz = 0;
ddotx = 0; ddoty = 0; ddotz = 0;
psi=0; the=0; phi=0;
dotpsi=0; dotthe=0; dotphi=0;
ddotpsi=0; ddotthe=0; ddotphi=0;
for jj = 0:0.01:0.3;
% acceleration in y direction
%ddoty = 9.8;
%doty = jj*9.8;
%y = 0.5*jj^2*9.8;
% acceleration in x direction
%ddotx = 9.8;
%dotx = jj*9.8;
%x = 0.5*jj^2*9.8;
www=40;
aaa=0.01;
x=aaa*sin(jj*www);
dotx= aaa*www*cos(jj*www);
ddotx=-aaa*www*www*sin(jj*www);
% sinusoidal the angle
%the=aaa*sin(jj*www);
%dotthe= aaa*www*cos(jj*www);
%ddotthe=-aaa*www*www*sin(jj*www);
switch=0;
% Call the Dynamics function without considering leg dynamics
F = Dynamics(ddotx,ddoty,ddotz,ddotpsi,ddotthe,ddotphi, ...
             dotx,doty,dotz,dotpsi,dotthe,dotphi,x,y,z,psi,the,phi,switch);
force(1,jj*100+1) = F(1);
force(2,jj*100+1) = F(2);
force(3,jj*100+1) = F(3);
force(4,jj*100+1) = F(4);
```

```
force(5,jj*100+1) = F(5);
force(6,jj*100+1) = F(6);

switch=1;

% Call the Dynamics function with considering leg dynamics
F = Dynamics(ddotx,ddoty,ddotz,ddotpsi,ddotthe,ddotphi, ...
             dotx,doty,dotz,dotpsi,dotthe,dotphi,x,y,z,psi,the,phi,switch);

forceleg(1,jj*100+1) = F(1);
forceleg(2,jj*100+1) = F(2);
forceleg(3,jj*100+1) = F(3);
forceleg(4,jj*100+1) = F(4);
forceleg(5,jj*100+1) = F(5);
forceleg(6,jj*100+1) = F(6);

end;

% Plot the required actuator forces for both cases
figure; hold on;
plot(0:0.01:0.3, force(1,:),'yo' );
plot(0:0.01:0.3, force(2,:),'w--' );
plot(0:0.01:0.3, force(3,:),'gx' );
plot(0:0.01:0.3, force(4,:),'r:' );
plot(0:0.01:0.3, force(5,:),'c-');
plot(0:0.01:0.3, force(6,:),'m-.' );

plot(0:0.01:0.3, forceleg(1,:),'yo' );
plot(0:0.01:0.3, forceleg(2,:),'w--' );
plot(0:0.01:0.3, forceleg(3,:),'gx' );
plot(0:0.01:0.3, forceleg(4,:),'r:' );
plot(0:0.01:0.3, forceleg(5,:), 'c-');
plot(0:0.01:0.3, forceleg(6,:),'m-.' );

legend('Actuator A', 'Actuator B','Actuator C','Actuator D',...
      'Actuator E','Actuator F' ,-1)
function [F] = Dynamics(ddotx,ddoty,ddotz,ddotpsi,ddotthe,ddotphi, ...
                  dotx,doty,dotz,dotpsi,dotthe,dotphi,x,y,z,psi,the,phi,switch)

% System parameters

gammap1 = 53.45;
gammap = [gammap1,120-gammap1, 120+gammap1, -120-gammap1,-120+gammap1,-gammap1];
gammab1 = 7.75;
gammab = [gammab1,120-gammab1, 120+gammab1, -120-gammab1,-120+gammab1,-gammab1];

alphap = [0, 120, 120, -120, -120, 0];
betab = [60,60,180,180,-60,-60];
```

```
rp = 0.668;
rb = 1.133;
hp = 0.203;

g=9.8;    % g acceleration
% switch =1 or =0  for with or without considering legs;
mleg=15*switch;

ml = mleg*0.2;    % mass of lower part of leg
mu = mleg*0.8;    % mass of upper part of leg

lu = 0.4;         % distance between upper center and connection
ll = 0.8;         % distance between lower center and connection

% assume the load to be 250 kg
mplatform = 250.0;
pIp = mplatform * [0.6^2/4+1.0^2/12 0 0; 0 0.6^2/4+1.0^2/12 0; 0 0 0.6^2/2];

% velocity
bVp = [dotx; doty; dotz];

% acceleration
bACCp = [ddotx; ddoty; ddotz];

spsi = sin(psi*pi/180);
cpsi = cos(psi*pi/180);
sthe = sin(the*pi/180);
cthe = cos(the*pi/180);
sphi = sin(phi*pi/180);
cphi = cos(phi*pi/180);

bRp(1,1) =  cphi * cthe;
bRp(1,2) = -sphi * cpsi + cphi * sthe * spsi;
bRp(1,3) =  sphi * spsi + cphi * sthe * cpsi;
bRp(2,1) =  sphi * cthe;
bRp(2,2) =  cphi * cpsi + sphi * sthe * spsi;
bRp(2,3) = -cphi * spsi + sphi * sthe * cpsi;
bRp(3,1) = -sthe;
bRp(3,2) =  cthe * spsi;
bRp(3,3) =  cthe * cpsi;

pp(1,:) = rp*cos(gammap*pi/180);
pp(2,:) = rp*sin(gammap*pi/180);
pp(3,:) = [hp/2,hp/2,hp/2,hp/2,hp/2,hp/2];

bb(1,:) = rb*cos(gammab*pi/180);
bb(2,:) = rb*sin(gammab*pi/180);
```

```
bb(3,:) = [0, 0, 0, 0, 0, 0];

bdp = [x,y,z]';
bDp = [bdp,bdp,bdp,bdp,bdp,bdp];

bA = bRp*pp + bDp - bb;

% inertia
Iui = mu*(2*lu)^2/3;

B(1,1) = cthe * cphi;
B(1,2) = -sphi;
B(1,3) = 0;
B(2,1) = cthe * sphi;
B(2,2) = cphi;
B(2,3) = 0;
B(3,1) = -sthe;
B(3,2) = 0;
B(3,3) = 1;

% angular velocity
bWp = B * [dotpsi; dotthe; dotphi];

dotB(1,1) = -sthe*cphi*dotthe - cthe*sphi*dotphi;
dotB(1,2) = -cphi*dotphi;
dotB(1,3) = 0;
dotB(2,1) = -sthe*sphi*dotthe + cthe*cphi*dotphi;
dotB(2,2) = -sphi*dotphi;
dotB(2,3) = 0;
dotB(3,1) = -cthe*dotthe;
dotB(3,2) = 0;
dotB(3,3) = 0;

% angular acceleration
balphap = dotB * [dotpsi; dotthe; dotphi] + B * [ddotpsi; ddotthe; ddotphi];

for i=1:6,

% length of legs
lleg = norm(bA(:,i));

% Calculations of variables as shown in dynamic equations

dotbA(:,i) =  cross( bWp, bRp*pp(:,i) ) + bVp;

ddotbA(:,i) = cross(balphap, bRp*pp(:,i)) + ...
              cross(bWp, cross(bWp,bRp*pp(:,i))) + bACCp;

dotlleg = (bA(:,i)'*dotbA(:,i))/lleg;
```

```
%Ili(i) = ml*( lleg^2 + (lleg-2*ll)*lleg + (lleg-2*ll)^2  )/3;
Ili(i) = ml*( 3*lleg^2  + (2*ll)^2 - 3*(2*ll)*lleg )/3;

dotIli(i) =  ml * (2*lleg-(2*ll)) * dotlleg;

bWi(:,i) = cross( bA(:,i), dotbA(:,i) ) / (norm(bA(:,i))*norm(bA(:,i)));

ti =  cross( bA(:,i), [eye(3), cross(eye(3), bRp*pp(:,i)) ] ) ...
  / (norm(bA(:,i))*norm(bA(:,i)));

balphai =( ( bA(:,i)'*bA(:,i) )*( cross(bA(:,i), ddotbA(:,i)) ) ...
         - 2*( bA(:,i)'*dotbA(:,i) )*( cross(bA(:,i), dotbA(:,i)) ) )...
         /  norm(bA(:,i))^4;


bTAUi = (Ili(i) + Iui) * balphai + dotIli(i) * bWi(:,i);

titimesbTAUi(:,i) = ti'*bTAUi;

hil = (ll/norm(bA(:,i))^3)*bA(:,i)*bA(:,i)'*[eye(3), cross(eye(3), ...
       bRp*pp(:,i))] + ( (norm(bA(:,i)) - ll)/norm(bA(:,i)) )* ...
       [eye(3), cross(eye(3), bRp*pp(:,i)) ];

hiu = (-lu/norm(bA(:,i))^3)*bA(:,i)*bA(:,i)'*[eye(3), cross(eye(3), ...
      bRp*pp(:,i)) ] + ( (lu)/norm(bA(:,i)) )* ...
      [eye(3), cross(eye(3), bRp*pp(:,i)) ];

hmg(:,i) = hil'*ml*[0, 0, -g]' + hiu'*mu*[0, 0, -g]';

end;

bIp = bRp * pIp * bRp';

bFp = mplatform * bACCp;
bFp = bFp - mplatform * [0, 0, -g]';

bTAUp = bIp * balphap + cross(bWp, (bIp*bWp));

% Platform dynamics
for i=1:6,
Jt(:,i) = [ ( bRp*pp(:,i) + bdp - bb(:,i) );  cross( bRp*pp(:,i), ...
          bdp-bb(:,i) ) ] / norm(bA(:,i)) ;
end;

% The required actuator forces for certain trajectory
F = inv(Jt)*([bFp; bTAUp]+titimesbTAUi*[1;1;1;1;1;1] - hmg*[1;1;1;1;1;1]);
```