NASA Contractor Report 195076

NASA-CR-195076
19950019666

Simulating the Dynamic Interaction of a Robotic Arm and the Space Shuttle Remote Manipulator System

Steven L. Garrahan and Robert H. Tolson
The George Washington University
Joint Institute for Advancement of Flight Sciences
Langley Research Center, Hampton, Virginia

Robert L. Williams II
Ohio University, Athens, Ohio

Cooperative Agreement NCC1-104

April 1995

National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia 23681-0001

Abstract

Industrial robots are usually attached to a rigid base. Placing the robot on a compliant base introduces dynamic coupling between the two systems. The Vehicle Emulation System (VES) is a six degree-of-freedom hydraulically driven platform that is capable of modeling the interaction of the manipulator and the base. The VES employs a force-torque sensor as the interface between the dynamics of the robot and the admittance model of the base.

A computer simulation of the VES is presented. Each of the hardware and software components are described, and it is shown how they are modeled in the simulation. The Simulink software package from the Mathworks is used as the programming environment. The performance of the simulation is compared with experimental hardware results to validate its accuracy.

A second computer simulation is outlined. This simulation models the dynamic interaction of a robot and a flexible base, and can act as a comparison to the simulated motion of the VES. It may also simulate the disturbances that any general robot may impose on a compliant base. The simulation applies to any two dynamic systems if at least one is defined in mode space.

Results are presented that compare the simulated VES motion with the motion of the VES hardware using the same admittance model. The two computer simulations are compared to determine how well the VES is expected to emulate the desired motion.

Simulation results are given for robots mounted to the end effector of the Space Shuttle Remote Manipulator System (SRMS). It is shown that for fast motions of the two robots studied, the SRMS experiences disturbances on the order of centimeters. Larger disturbances are possible if different manipulators are used.

Table of Contents

List of Figures

List of Tables

Nomenclature

$\{B\}$	Base coordinate system
c	Single degree-of-freedom damping coefficient
c_d	Discharge coefficient
C_p	Pressure coefficient
C_x	Flow coefficient
$[C]$	Mutli degree-of-freedom damping matrix
$[C_{proc}]$	Force-torque sensor sensitivity matrix
DOF	Degree-of-freedom
DOSS	Dextrous Orbital Servicing System
$E(k)$	Error signal used in the VES control algorithm
$\{\eta(t)\}$	Modal coordinates
f	Sampling frequency of force-torque sensor
$f_i(X)$	i^{th} constraint equation
F	Frequency of the mass-spring-damper system
$F(t_n)$	Discrete force at time t_n
J	Jacobian matrix
J_M	Modified Jacobian matrix
J_{NR}	Newton-Raphson Jacobian matrix
k	Single degree-of-freedom spring coefficient
k_{ad}	Gain of the analog-to-digital converter

K_{DA}	Gain of the digital-to-analog converter
K_D	Position gain
K_P	Derivative gain
K_T	Transducer gain
$[K]$	Multi degree-of-freedom spring matrix
L_i	Length of the i^{th} actuator
$\dot{L}_i$	Velocity of the i^{th} actuator
$\ddot{L}_i$	Acceleration of the i^{th} actuator
λ	Root of the characteristic equation
m	Single degree-of-freedom mass
$[M]$	Multi degree-of-freedom mass matrix
Δp	Differential pressure
P_L	Load pressure
P_S	Supply pressure
$\{P\}$	Platform coordinate system
P_c	Vector from the base of the robot to the center of mass of the robot
${}^l({}^jP_k)$	Vector from point j to point k, expressed in the l coordinate system
ϕ	Discrete state transition matrix
$[\Phi]$	Modeshapes matrix
$[\Phi_l]$	Lower 3xn modeshapes matrix
$[\Phi_u]$	Upper 3xn modeshapes matrix

Q	The flow through an aperture
${}^{B}_{A}R$	Rotation matrix from frame A to frame B
$\Re$	Skew-symmetric position-cross matrix
ρ	Fluid density
$[S_{proc}]$	Force-torque sensor sensitivity matrix
SRMS	Shuttle Remote Manipulator System
τ	Sampling period of the force-torque sensor
$\dot{\Theta}_c$	Angular velocity of the center of mass of the robot
$\ddot{\Theta}_c$	Angular acceleration of the center of mass of the robot
VES	Vehicle Emulation System
V_{ex}	Excitation voltage of the force-torque sensor
$\{V_R\}$	Output voltage from the force-torque sensor
W	Wrench (vector of forces and moments)
$\underline{W_{ext}}$	External wrench due to the motion of the robot
ω	Natural frequency of the admittance model
ω_d	Damped natural frequency
$x(n)$	Discrete one degree-of-freedom translation
$\dot{x}(n)$	Discrete one degree-of-freedom velocity
$\ddot{x}(n)$	Discrete one degree-of-freedom acceleration
$x(t)$	One degree-of-freedom translation
$\dot{x}(t)$	One degree-of-freedom velocity

$\ddot{x}(t)$	One degree-of-freedom acceleration
$\ddot{x}(t_n)$	Discrete acceleration at time t_n
$\{\bar{x}(t)\}$	State of the mobile base
X	Cartesian position vector
$\hat{X}$	Euler position vector
$\{X(t)\}$	Multi degree-of-freedom generalized coordinates
ζ	Damping value
ZOH	Zero-order-hold

1. Introduction

1.1 Background and Motivation

Conventional robotic systems are designed to operate in industrial environments under the influence of gravity, mounted on rigid bases. However, as scientists continue to expand their research into the realm of outer space, the importance of space robotics is increasing. Human extra-vehicular activity is costly and involves potential danger to human life, so considerable research is being performed in the field of space robotics.

The area of control algorithms for space robots is the subject of substantial investigation. Previous research includes the study of free-floating robots [1], trajectory planning through the use of kinematic redundancies of the manipulator [2], and the use of satellite attitude control torques to control satellite mounted robots [3], among others. Validation of robotic facilities and control algorithms is difficult in conventional laboratories, due to the influence of gravity.

Researchers at the Massachusetts Institute of Technology (MIT), under a grant from NASA, have developed an Earth-based tool for the study of the dynamics of space robots [4], [5], [6], [7]. They have designed and built the Vehicle Emulation System (VES) to study the effects of mounting a robot on a flexible base. The VES is shown in figure 1.1. The main hardware component of the VES is a six degree-of-freedom hydraulically driven Stewart platform. The VES employs a force-torque sensor as the interface between the robot and the control software. This sensor measures the static effects due to gravity, and the dynamic effects due to the motion of the robot. Baker [4]

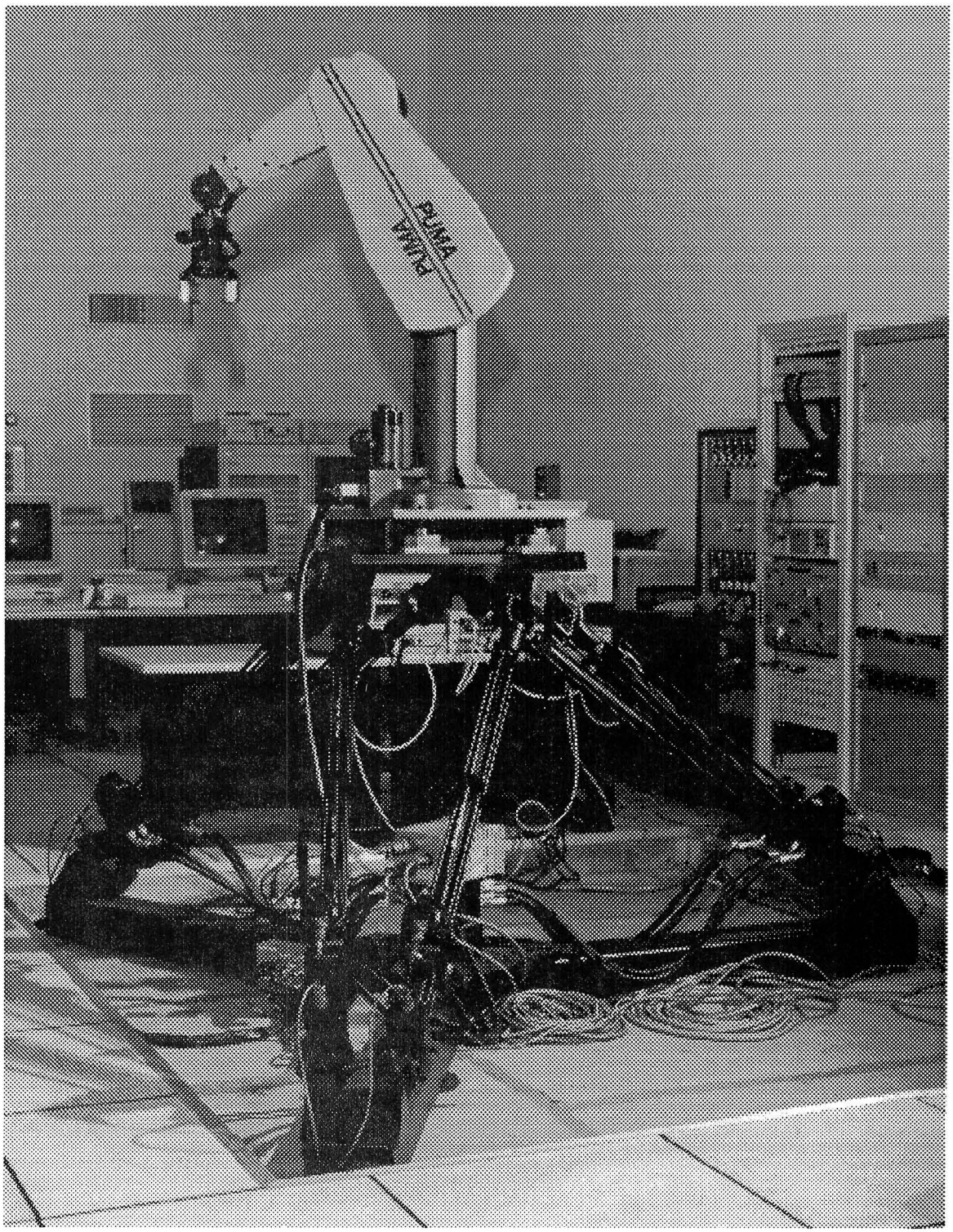


Figure 1.1 Vehicle Emulation System with PUMA 560 Robot

offers a detailed description of the force-torque sensor, and an iterative algorithm for removing the effects of gravity, for space-based experiments. He also presents a real-time, on-line, approach to this problem, which is similar to the method outlined in [8]. Idris has evaluated the concept of controlling the VES using the VxWorks real-time operating system [5]. The kinematic design of the Stewart platform, including a discussion of the hydraulic components and all related hardware, is presented by Mueller [6]. Kuklinski furnishes an overview of the entire VES [7]. In addition, he presents a detailed description of the safety routines that are employed by the VES, and dynamic performance characteristics of the equipment. Williams has studied the kinematics of the Stewart platform [9]. He describes methods for computing the forward and inverse position and velocity kinematics of the platform, and gives examples to compare different methods of computation.

In operation, the Stewart platform is commanded to emulate the motion of a compliant base, through software admittance control (refer to figure 1.2). Admittance control uses forces and moments (termed a wrench) as input variables to determine the current state of a dynamic system. The platform is moving under the robot as it emulates a compliant base. This motion generates a feedback wrench which is also measured by the sensor. The combined analog wrench is digitized and used as the forcing function to the admittance model defined in the control software. The effects of gravity are removed through gravity compensation. The equations of motion are integrated to provide the position and orientation (termed a pose) of the platform, and the VES inverse kinematics are solved to determine the actuator lengths that will move the platform to the new

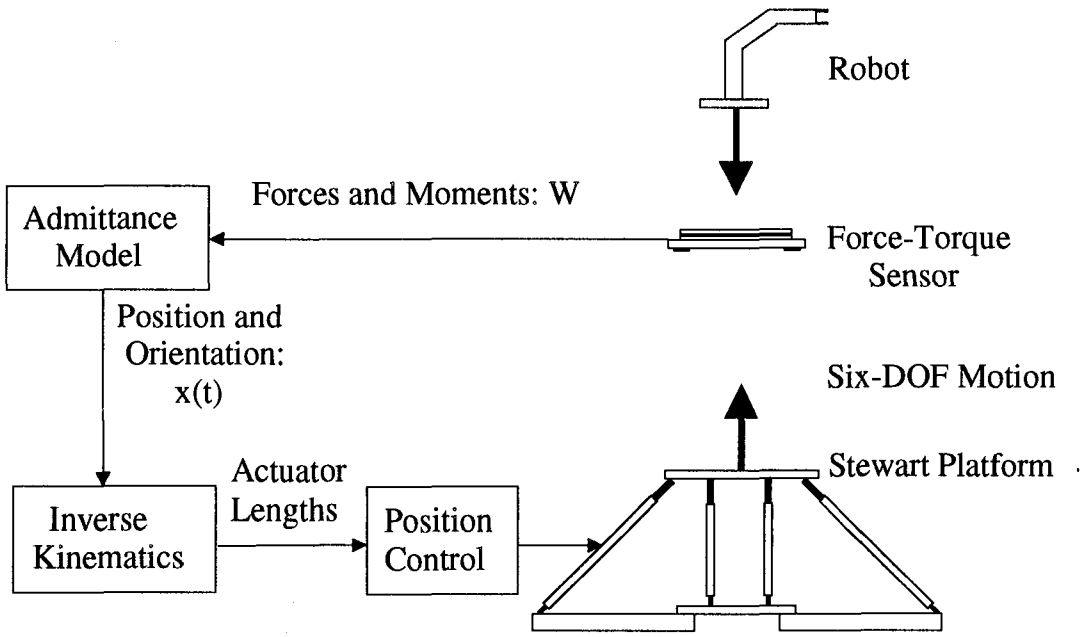


Figure 1.2 VES Admittance Control

location. The platform is controlled through position-derivative control. The MIT VES has been used to study decoupled mass-spring-damper systems [7], and robots mounted on massive mobile bases [10].

NASA Langley Research Center has a Vehicle Emulation System that was built at MIT. The primary purpose of the NASA VES is to study the dynamics of robots mounted on the Space Shuttle Remote Manipulator System (SRMS). The VES may be used to simulate other compliant bases, including satellite-mounted robots. The SRMS is a flexible arm that is expected to experience disturbances under the influence of an active robot mounted to the end effector. The process of controlling and minimizing these disturbances is called disturbance compensation. Papadopoulos has developed an

admittance model for the SRMS in four configurations [11], that will be used in conjunction with the VES to study methods of disturbance compensation.

1.2 Overview of Document

The research conducted is presented in the ensuing four chapters.

Chapter 2 presents a computer simulation of the VES. The simulation is designed to study the effects of hardware and software changes before they are actually implemented. In addition, the simulation is utilized to analyze the predicted motion of the VES, and investigate whether component limits are expected to be exceeded. The operation of the Vehicle Emulation System is outlined in this chapter. Descriptions of the hardware and software components, and their interface, are presented. The software routines are outlined, and it is shown how they are combined to perform either admittance or trajectory control on the Stewart Platform. The forward position and velocity kinematics presented in [9] are expanded upon to derive the forward acceleration kinematics of the Stewart platform, for use in simulating the platform's motion.

This computer model is used to simulate the action of the VES as it emulates the motion of the SRMS-robot dynamic interaction. A separate computer simulation is developed in Chapter 3 that models the SRMS-robot dynamics. This tool is designed to validate the accuracy of the VES hardware, which is subject to numerous errors, including friction in the actuators, and a time delay between commanded motion and realization of this motion in the hardware. This second simulation is used to model the effects of mounting any general robot on the SRMS. This simulation can predict the motions of any

hypothetical robot on a flexible support structure, under the influence of a theoretical control strategy.

The results of the research are included in Chapter 4. High and low frequency noise is shown to affect the force-torque sensor that provides the interface between the externally mounted robot and the VES. To verify the accuracy of the model, the simulated VES motion is compared with similar motions of the MIT VES. The simulation of the robot-SRMS dynamics is compared with the VES model, to determine how closely the VES will emulate the intended motion.

Chapter 5 summarizes the research and presents some concluding remarks.

2. Computer Simulation of the Vehicle Emulation System

This chapter presents a computer simulation of the Vehicle Emulation System. Descriptions of all major components are provided, and it is shown how they are modeled in software.

The Vehicle Emulation System was designed and built by the Massachusetts Institute of Technology, under a contract from NASA. The hardware consists of a Stewart platform and a force-torque sensor. The platform, shown in figure 1.1, is computer controlled. At each simulation time step, the software computes the desired position and orientation of the center of the platform top plate, solves the inverse kinematics to determine the appropriate leg lengths, checks the safety requirements, and communicates the reference leg lengths to the actuators.

The Stewart platform is a hydraulically actuated platform. Each of the six prismatic legs is connected to the fixed base through two-DOF universal joints, and to the mobile top plate with three-DOF spherical joints. Equation 2.1 is the Kutzbach equation, which is used to determine the mobility, or number of degrees of freedom, of a manipulator [12].

$$M = 6(N-1) - 5J_1 - 4J_2 - 3J_3 - 2J_4 - J_5 \qquad (2.1)$$

In this equation, J_i is the number of joints with i-degrees of freedom, and N is the number of links, including the fixed base link. For the Stewart platform, there are fourteen links, two associated with each actuator, plus the top plate, and fixed link, so evaluation of equation (2.1) yields

$$M = 6(14-1) - 5(6) - 4(6) - 3(6) = 6.$$

Therefore, the platform has six degrees of freedom.

Figure 2.1 shows a side view of the VES, including the reference coordinate systems that are involved. The inertial system, {I}, is fixed in space at the base of the platform, with its Z-axis up. The center of the platform top plate is where the platform frame, {P}, is located. These two frames are related through an X-Y-Z rotation about the fixed inertial axes, and a translation. In the figure, the platform is horizontal not rotated, so the {I} and {P} frames are aligned. The manufacturer provides the location of the force sensor's coordinate system, {S}, which is fixed on the sensor. The sensor is rigidly attached to the platform. The VES will be operated with a robot (or robots) mounted on

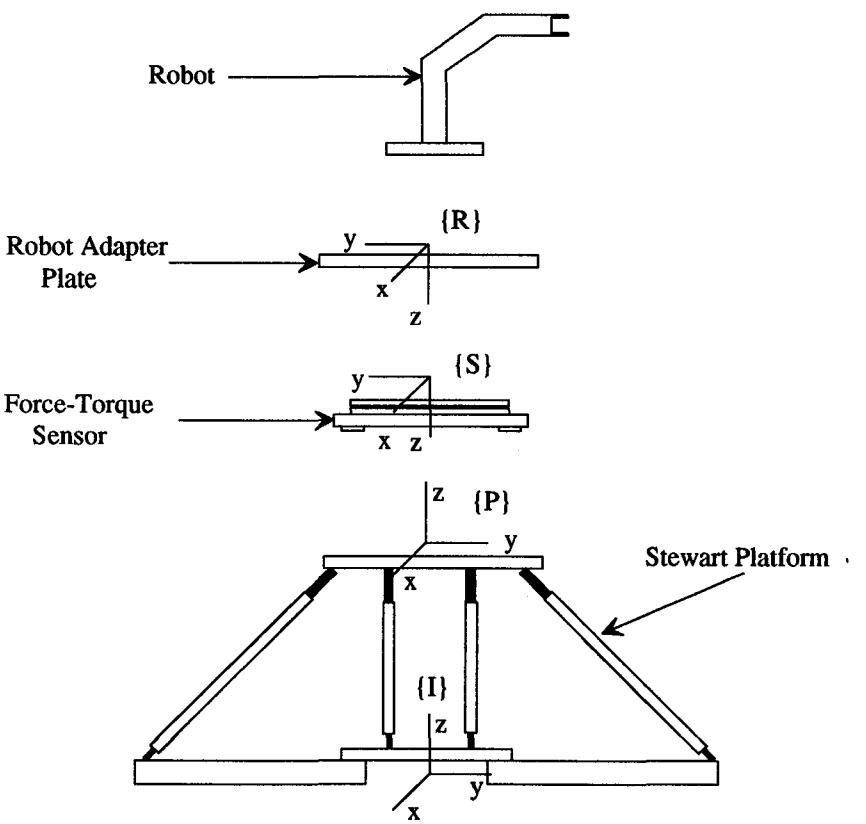


Figure 2.1 Side View of Each Component of the VES, with Coordinate Frames

the force sensor, with an adapter plate mounting the robot to the sensor. The dimensions of the adapter plate are known, and the coordinate frame, {R}, is located at the geometric center of the plate (the plate is assumed to be of uniform mass distribution). In addition to these frames, the attached robot has frames associated with each link.

There are a number of complications that may arise when utilizing a complex mechanical system such as the Vehicle Emulation System. These include the following:

- The Vehicle Emulation System is composed of a variety of complex components. If hardware changes are to be made, the system would be out of service while these changes occur. The final effects of the modifications would not be known until they were completed, and the system was again operational.
- The NASA VES is primarily intended to be used to observe the interaction of a robot with a mobile base. The task of mounting a different robot on the platform involves designing a new adapter plate to make the connection between the robot and the force sensor, and then devising a method for lifting the robot onto the system. The PUMA adapter plate was relatively simple to design, but plates for the DOSS, and other robots, could be more complicated. In addition, it might be desired to place robots that are not readily available in the lab, or theoretically designed robots, on the VES.
- All mechanical systems have physical limits associated with their operation. If it is possible to determine whether these limits will be exceeded before an experiment is performed, it could prevent damage to costly equipment.

These concepts were the motivation for developing a computer-based simulation of the VES. The Simulink software package, manufactured by The Mathworks as part of their MATLAB software, was chosen as the simulation environment [13]. Simulink is menu-driven, and includes a large variety of built-in functions, including filters, s- and z-domain functions, and non-linear capabilities. The user creates a schematic of the system, and can numerically integrate the equations of motion using Runge-Kutta, discrete state space methods, Adams-Gear, or other techniques. The results are available to be analyzed in the MATLAB environment.

The major VES components that are modeled are the actuator dynamics and position control system, the software routines, the robot, and the force sensor. The Stewart platform forward kinematics are also solved, to provide the actual position and orientation, velocity, and acceleration of the platform in Cartesian space. This information is fed back to the robot model, to include the motion of the base in the calculations of the robot's base wrench. This wrench serves as the input to the admittance model that the platform is emulating. Figure 2.2 shows the interaction between the functions. The software calculations, the position control loop, and the robot's control system are performed at specific clock rates, which are controlled by the external clock. The clock increments time in intervals that are equal to the highest sampling rate associated with the system. The position control loop, operating at 2900 Hz, requires the fastest clock. The system time is propagated through each block. The following sections describe the Simulink blocks included in fig 2.2.

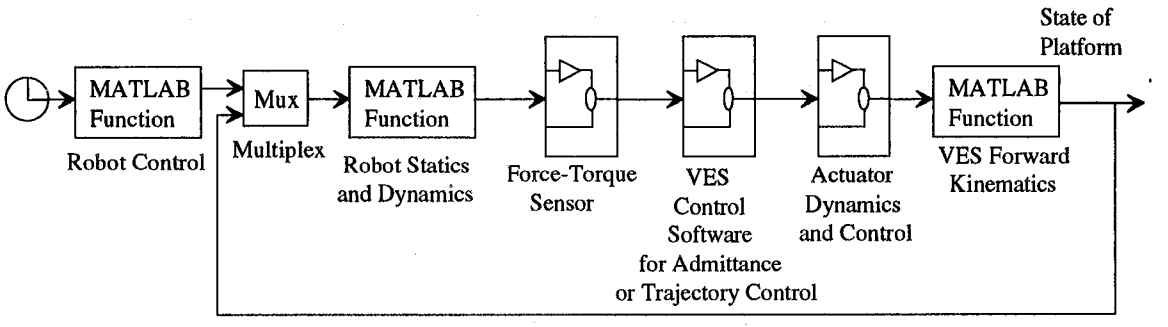


Figure 2.2 VES Computer Simulation: Main Flow Diagram

2.1 Robot Model

This section describes the Robot Control and Robot Statics and Dynamics blocks from fig 2.2, and defines the simulated robot.

There are numerous methods for controlling manipulators, including resolved rate control, joint control, inverse position control, vision systems, and force control. The Robot Control block implements the desired type of control, specified by the user during the initialization routine. Currently, the code implements joint trajectory control. Additional control algorithms can be included, as long as the algorithm involves the calculation of each joint angle, rate, and acceleration for each simulation time step. If any parameter is not directly calculated, time integration or differentiation of the values that are calculated will result in a complete set of parameters. It is important that each of these elements are computed, because they are required by the Robot Statics and Dynamics code.

The Robot Statics and Dynamics block computes the wrench exerted on the force sensor due to the motion of the robot and the mobile base, and the effects due to gravity. The computations are based on the Newton-Euler equations included in Appendix A. At

each simulation time step, the base motion is combined with the motions of the joints and propagated from the robot's first link to the end link. The force and moment at the center of the link are also determined with the Newton-Euler equations. Then, the process is reversed, to calculate the wrench between links as they interact. These calculations are dependent on having an accurate model of the manipulator.

Two robots that are available to be placed on the VES are the PUMA 560 and Dextrous Orbital Servicing System (DOSS). The PUMA 560 shown in fig 2.3 is a six degree-of-freedom serial robot produced by Unimation. The robot consists of a base link, a waist joint, a shoulder, an elbow, and a three-DOF wrist. The three wrist frames are located at the same point, and perform a roll-pitch-roll rotation. Table 2.1 contains the mass and inertia values, and the center of mass location for each link.

The Dextrous Orbital Servicing System is a seven-DOF serial robot. Figure 2.4 shows that the arm has shoulder and wrist roll, yaw, and pitch joints, along with an elbow pitch joint. The extra degree-of-freedom associated with the DOSS allows it to move in its workspace while optimizing some aspect of its motion. Table 2.2 contains the mass and inertia values, and the center of mass location for each link of the DOSS.

During the initialization routine associated with the VES simulation, the user is presented with a menu of robots from which to choose. Additional manipulators are created by substituting the new mass parameters into the files. Hypothetical manipulators can also be simulated, as may robots with payloads that might be difficult to manage in a laboratory environment.

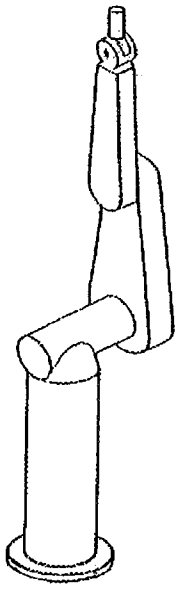


Figure 2.3 PUMA 560 Robot

Link i	Mass (kg)	Center of Mass (meters)			Principle Moments of Inertia (kg-m^2)		
		X	Y	Z	I_{XX}	I_{YY}	I_{ZZ}
0*	20.00	0.000	-0.015	-0.380	0.000	0.000	0.000
1*	9.00	0.000	0.135	-0.005	0.000	0.000	0.350
2	17.40	0.068	0.006	-0.016	0.130	0.524	0.539
3	4.80	0.000	-0.070	0.014	0.066	0.013	0.086
4	0.82	0.000	0.000	-0.019	$1.80e^{-3}$	$1.80e^{-3}$	$1.30e^{-3}$
5	0.34	0.000	0.000	0.000	$0.30e^{-3}$	$0.30e^{-3}$	$0.40e^{-3}$
6	0.09	0.000	0.000	0.032	$0.15e^{-3}$	$0.15e^{-3}$	$0.40e^{-4}$

* Note: These values are not included in [14]. They were estimated from the geometry of the PUMA 560.

Table 2.1 PUMA 560 Mass Parameters[14]

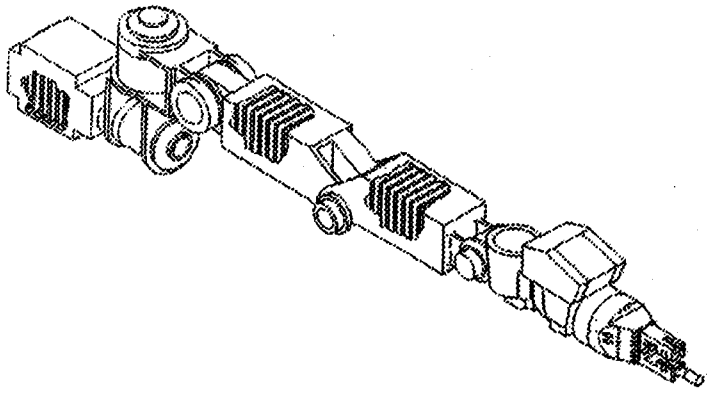


Figure 2.4 Dextrous Orbital Servicing System[11]

Link i	Mass (kg)	Center of Mass (meters)			Principle Moments of Inertia (kg-m^2)		
		X	Y	Z	I_{XX}	I_{YY}	I_{ZZ}
0*	0.00	0.000	0.000	0.000	0.000	0.000	0.000
1	9.61	0.000	-0.117	$1.25e^{-2}$	0.161	0.089	0.143
2	8.95	0.131	$-8.89e^{-3}$	$-1.02e^{-2}$	0.097	0.215	0.201
3	20.98	0.295	$-1.29e^{-2}$	$1.07e^{-2}$	0.175	0.023	0.23
4	16.52	0.267	$6.71e^{-2}$	0.000	0.109	0.436	0.450
5	6.90	$5.08e^{-2}$	$-9.14e^{-3}$	$-9.14e^{-3}$	0.712	0.082	0.090
6	3.68	0.000	$-7.42e^{-2}$	$2.54e^{-3}$	0.046	0.034	0.046
7	22.29	0.000	$-4.90e^{-2}$	$1.73e^{-2}$	0.389	0.315	0.166

* Note: These values are not included in [11]. They are the parameters for the mounting block, and were set to zero for simulation purposes.

Table 2.2 DOSS Mass Parameters[11]

2.2 Force-Torque Sensor

The operation of the force-torque sensor is described in this section. Then it is shown how the sensor is modeled in the VES computer simulation.

During admittance model simulations, the interface between the robot and the VES is the force-torque sensor. The active robot exerts forces and moments on the sensor, which processes the disturbance and sends a signal to the software. The control software directs the Stewart platform to move in accordance with the admittance model being emulated.

The six channel force-torque sensor, model OR6-5-1 Biomechanics Platform, produced by Advanced Mechanical Technology, Inc. (AMTI) is used by the VES. The design of the sensor is described in detail in refs. [4], and [15]. Figure 2.5 shows the structure of the force sensor with the appropriate axes. The sensor is comprised of rigid top and bottom plates, connected by load cells in the four corners.

The sensor measures the applied forces and moments through the use of strain gages connected in Wheatstone bridge formations, as in figures 2.6 and 2.7. Figure 2.7 displays the circumference of a single load cell, as if it had been peeled off and laid flat, exposing the fourteen gages. The asterisk is located at the bottom of the cell and faces the outside corner of the load cell, as indicated in fig 2.5. Figure 2.8 shows a side view of a single cell, with two sets of orthogonal axes. When a horizontal force, F_X, is applied as shown, axis C_S experiences a compression due to shear, while axis T_S is in tension. Each cell contains a similar arrangement, with the outputs combined additively in the legs of Wheatstone bridges. A purely vertical force, F_Z, causes both C_S and T_S to be in

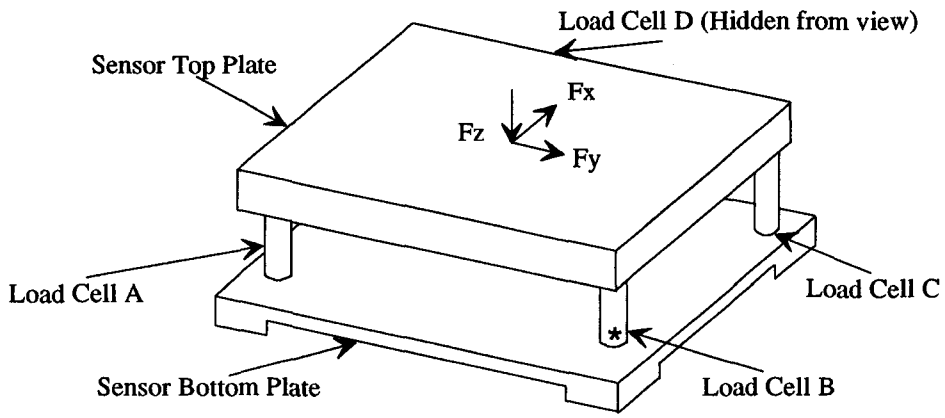


Figure 2.5 Schematic View of the Force-Torque Sensor, Showing the Load Cells[15]

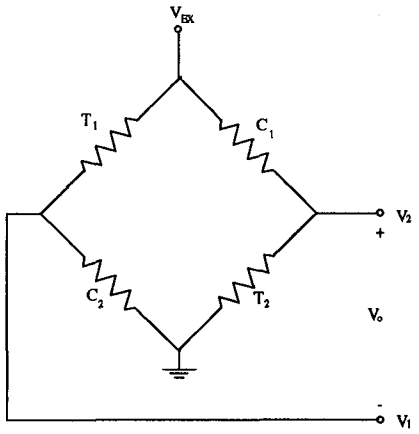


Figure 2.6 Strain Gage

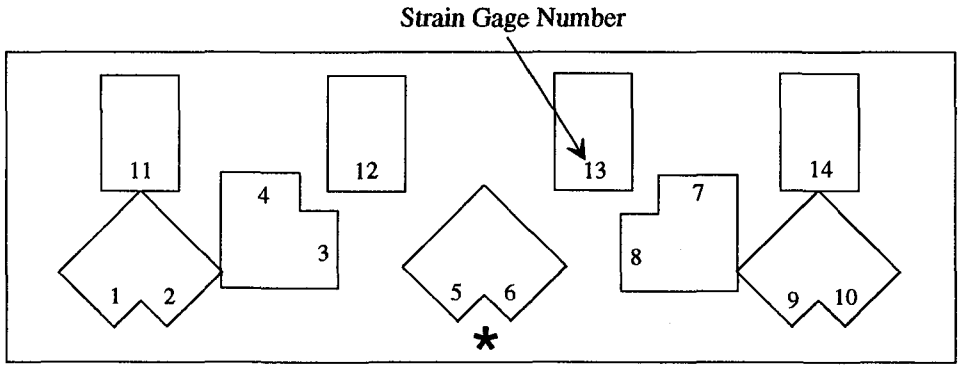


Figure 2.7 Location of Fourteen Strain Gages on a Single Load Cell[15]

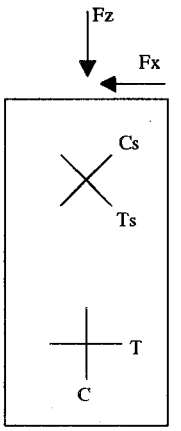


Figure 2.8 Side View of a Single Load Cell, with Orthogonal Sets of Axes[15]

compression, which, when configured in adjacent branches of the bridge, will tend to cancel each other, producing no net response. Referring to fig 2.7, it is observed that the gages are arranged symmetrically around the load cell.

Under an applied wrench, Wheatstone bridge elements T_1 and T_2 in fig 2.6 experience tension, while C_1 and C_2 undergo compression of the same magnitude, ΔR. ΔR is a resistance which is proportional to the applied wrench component, W_i, according to $\Delta R = \beta W_i$, where β is a constant and $W_i = F_x,\ F_y,\ F_z,\ M_x,\ M_y$, or M_z. Assuming that the base resistance values in each branch of the bridge, R, are the same, mesh analysis produces

$$\begin{aligned} V_1 &= Vex - 2I_1(R + \Delta R) \\ V_2 &= Vex - 2I_2(R - \Delta R) \end{aligned} \tag{2.2}$$

$$I_1 = I_2 = \frac{Vex}{4R}$$

so that

$$V_o = V_2 - V_1 = \frac{\Delta R}{R} Vex = \frac{\beta W_i}{R} Vex, \tag{2.3}$$

where V_o is the output voltage from figure 2.6.

The force sensor outputs six voltages that reproduce the applied wrench, according to equation (2.4),

$$\{V_s\} = [S]\{W_s\}, \qquad (2.4)$$

where $\{W_s\}$ is the applied six-component wrench in the sensor coordinate system, $\{V_s\}$ is the corresponding output voltage vector, and $[S]$ is the 6x6 sensitivity matrix. For an ideal sensor, $[S]$ is a diagonal matrix with each element equal to the quantity β/R, defined for the corresponding Wheatstone bridge. However, any such real device contains crosstalk between the channels, caused both by a bending of the sensor top plate and imperfect strain gages. As a result, the sensitivity matrix contains off-diagonal terms that represent crosstalk.

The output voltages from the force sensor are typically in the millivolt range, and contain a noise component from the excitation voltage, V_{ex}. The signals are passed through the Analog Devices wide bandwidth strain gage signal conditioners, model B31. The signal conditioners first amplify the voltages by an amount A_i, and then filter them through a two-pole low-pass Butterworth filter. The filter has a cut-off frequency of 20 Hz, but this is adjustable with the addition of resistors and capacitors. The Analog I/O Board, #DT1401, from Data Translation, Inc., converts the analog signal to digital. The board has a gain of 305 mV/ADU (analog to digital unit). Therefore, the final wrench values that are communicated to the VES software are

$$VRi = \frac{\beta A i Vex}{KadR} Wi, \qquad (2.5)$$

or

$$\{VR\} = [Sproc]\{W\}, \tag{2.6}$$

where K_{ad} is the gain of the A/D converter, and $[S_{proc}]$ is the sensitivity matrix that now includes all terms for processing the signal.

- **Simulation Model of the Force-Torque Sensor**

Even though the force sensor employs Wheatstone bridges to determine the external wrench, equation (2.6) illustrates that the process reduces to a matrix operation. It is not possible to alter the force sensor internally to change the values in eq (2.3), so modeling the sensor at the resistor level serves no purpose. For this reason the sensor is modeled as a matrix, as indicated by the matrix multiplication block in fig 2.9. The other blocks in fig 2.9 are external to the sensor.

Experimentation has shown that the excitation voltage, V_{ex}, contains a high frequency noise component, along with a very low frequency drift component due to environmental factors (refer to section 4.1 for details). The low frequency drift occurs over periods of hours, which is much longer than any proposed simulation, so it is ignored. However, the presence of the low frequency drift implies that the force sensor will need to be re-calibrated periodically. The high frequency component is modeled (fig 2.9) as Gaussian noise with a zero mean that is different for each channel. In addition, a bias is added to each wrench component, to simulate stray voltages that are present on each channel. This offset is taken into account during the calibration routine, as outlined in the next section. After amplification, the force sensor signal is filtered through 20 Hz two-pole Butterworth filters, to remove some of the noise.

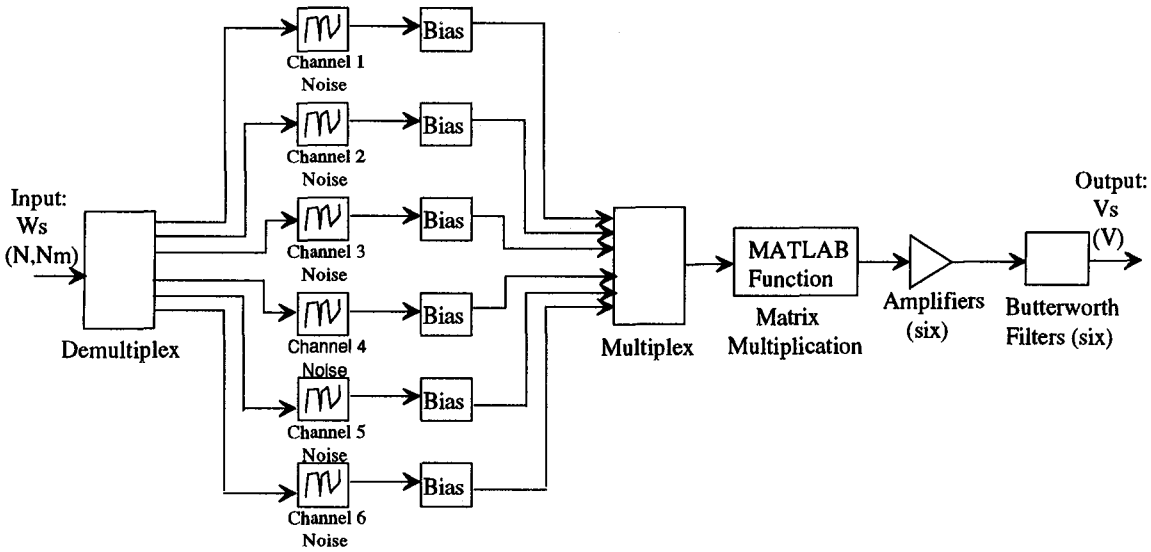


Figure 2.9 VES Computer Simulation: Force-Torque Sensor Model

2.3 Software

This section describes the control software utilized by the VES, and explains how the code was modified for the VES simulation.

The VES control software implements either admittance or trajectory control on the platform, and then the inverse kinematics are solved to determine the lengths of the six legs, which are commanded to the VES. Figure 2.10 shows the VES architecture, including the software/hardware interface. The control software is developed on a Sun Sparc Station using the C programming language, and loaded onto a Motorola 68030 processor (Heurikon #HK68/V30XE), named the redSlave. The software is designed to run in real-time on the redSlave using the VxWorks operating system. A second processor, the blueSlave, based on the Motorola 68020 computer chip (Heurikon #HK68/V20), is also available. The blueSlave can be used to handle some of the control

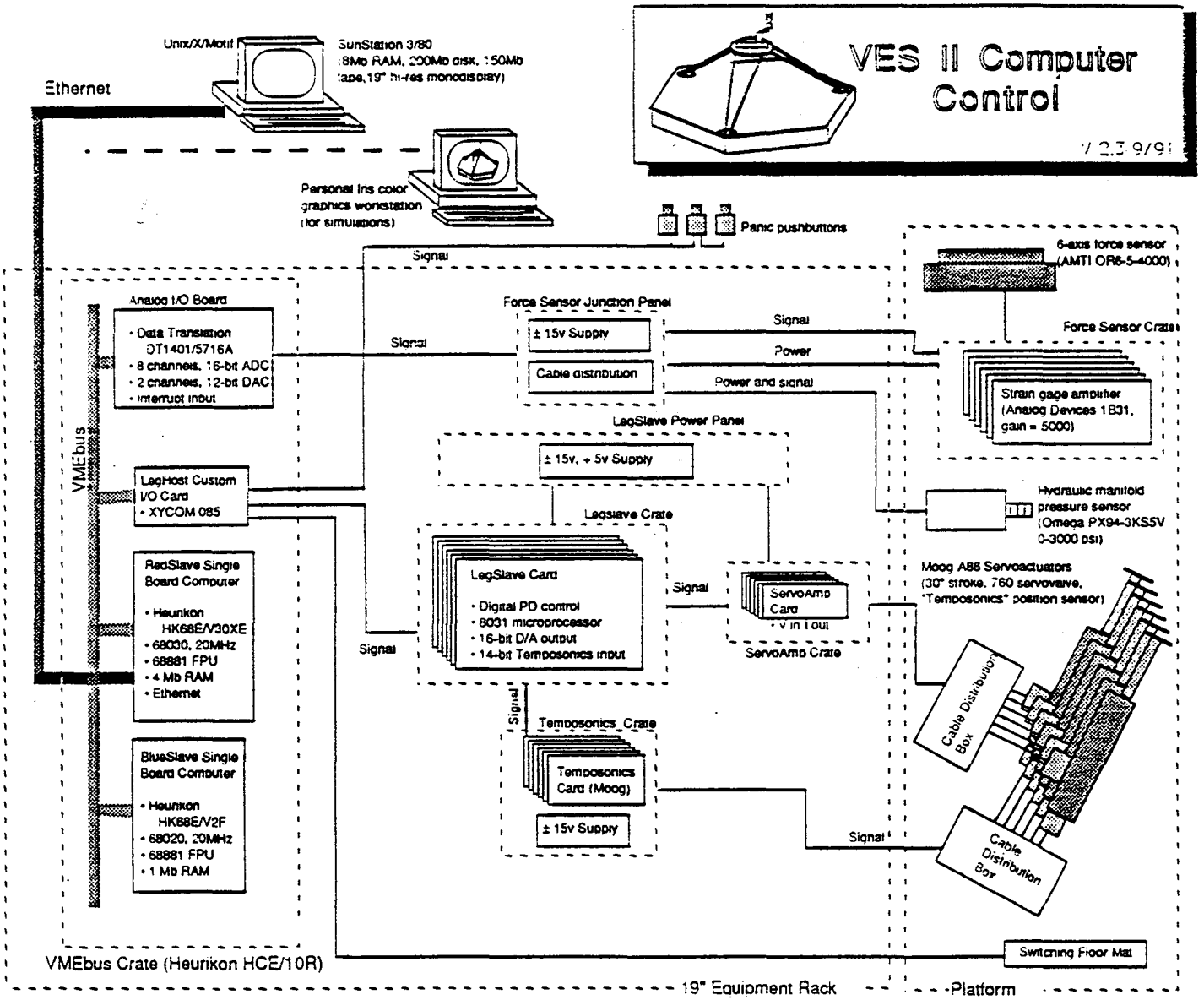


Figure 2.10 VES Architecture[6]

routines, or control the robotic manipulator. The two processors communicate using a VMEBus.

The software tasks involve computing the desired trajectory, and communicating with the environment. The routines include:

- Force Measurement: measuring the force applied to the force-torque sensor, which includes the motions of the robot, gravity effects, and the acceleration of the mobile base,
- Gravity Compensation: computing the static wrench due to the robot and subtracting it from the measured wrench, resulting in the dynamic wrench,
- Admittance Model: determining the next state for a simulated admittance model,
- Trajectory Generation: determining the next state for trajectory control of the platform,
- Inverse Kinematics: solving the inverse kinematics to determine the actuator lengths at the current simulation time step,
- Safety Checks: checking the safety requirements to ascertain whether the platform is stable and under control,
- Leg Communication: communicating the new leg lengths to the actuators.

Appendix B contains a detailed description of the gravity compensation method. Appendix C describes the algorithm for each software routine.

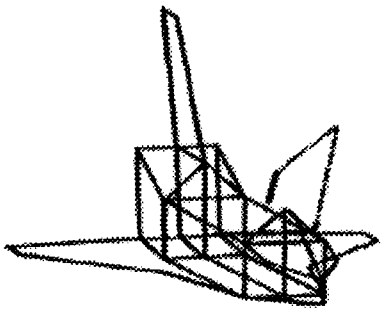
The main calling routine for the software control code is testPlatform.c. This

program is an interface that allows the user to select admittance control, trajectory control, the calibration routine, or the data collection routine. The calibration routine is used to compute the bias on the force sensor. Each of the six channels on the force sensor are sampled one hundred times and the average bias and standard deviation are computed. Each time the force measurement code is called, this average bias is subtracted from the measured input. The data collection routine permits the user to select system variables to be saved during execution so that they may be analyzed later. The admittance and trajectory control algorithms are described in the following paragraphs.

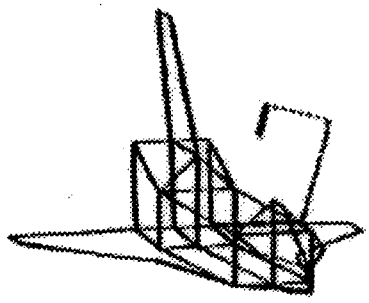
Admittance control allows the VES to emulate the motion of a complex dynamic system, as it interacts with the environment. The NASA Langley VES will primarily investigate the interaction of one or more robotic manipulators and the Shuttle RMS in four configurations, using the admittance models developed by Papadopoulos [11]. These four configurations, shown in fig 2.11, are termed unberth, low hover, deploy, and capture, and are operational arm configurations for the SRMS.

Six of the seven algorithms outlined above are required for admittance control; the program flow is shown in fig 2.12. The sampling rate of the control software is limited by the execution time of the code, and is currently between 75 and 100 Hz. The clock interrupt routine handles the timing for the system, and if the control loop is completed before the next time step, it waits before closing the loop. TestPlatform.c invokes the safety checks routine, to ensure proper initialization of this subroutine.

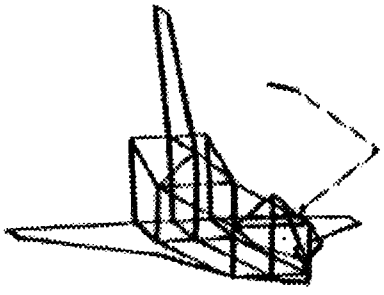
Trajectory control is used to command the platform to follow a pre-planned



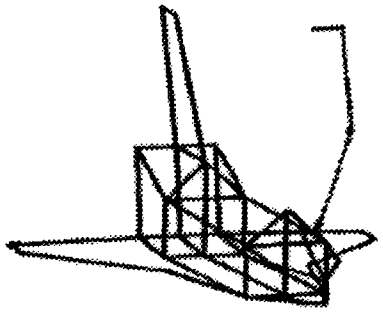
(a) Unberth



(b) Low Hover



(c) Deploy



(d) Capture

Figure 2.11 Four Operational Configurations of the SRMS: (a) Unberth (b) Low Hover (c) Deploy (d) Capture[11]

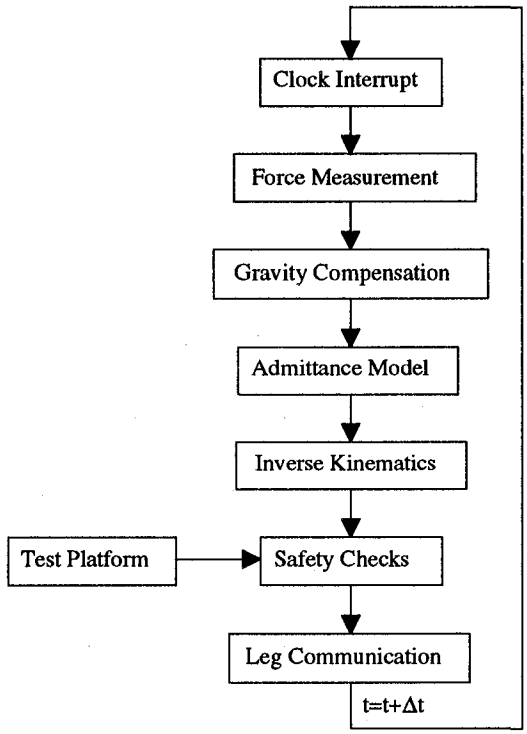


Figure 2.12 VES Software Flow for Admittance Control

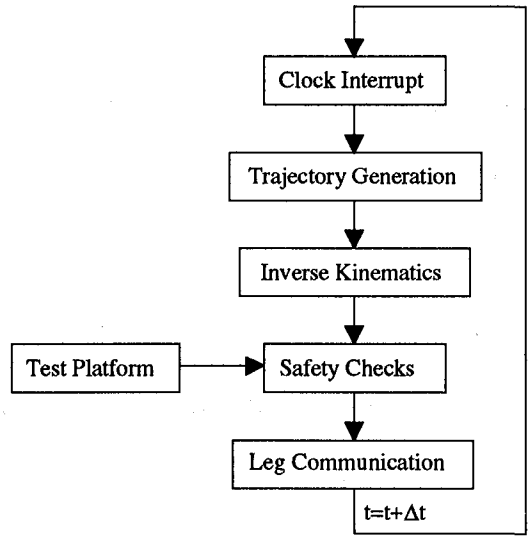


Figure 2.13 VES Software Flow for Trajectory Control

motion. Currently, linear and sinusoidal motion is allowed. Figure 2.13 illustrates the flow between the four routines involved. Trajectory control is executed at 100 Hertz, and the clock interrupt routine controls the timing.

- **VES Control Software Simulation**

The VES Control Software block shown in fig 2.2 implements the software routines described in the previous section. The original C programming language subroutines are designed to run under the VxWorks real-time operating environment, and contain a considerable amount of VxWorks-specific commands. Some of the tasks performed by the VxWorks code involve multi-tasking, task-locking, and interfacing with the hardware. Most of the tasks involve coordinating the code to run in real-time, which does not occur using Simulink, so they were removed from the code.

The Simulink Control Software accomplishes all of the calculations performed by the original code, with the timing requirements taken care of by the external clock described in section 2.1. Simulink outputs the following system parameters to the MATLAB workspace to be analyzed after the simulation: the total wrench applied to the force sensor; the static and dynamic wrench due to the robot; the position, velocity, and acceleration of the platform; and the leg lengths, velocities and accelerations. The leg velocity and acceleration calculations are not performed by the original code. They have been included so that they can be compared with the output of the position/derivative control loop, which is the actual state of each leg.

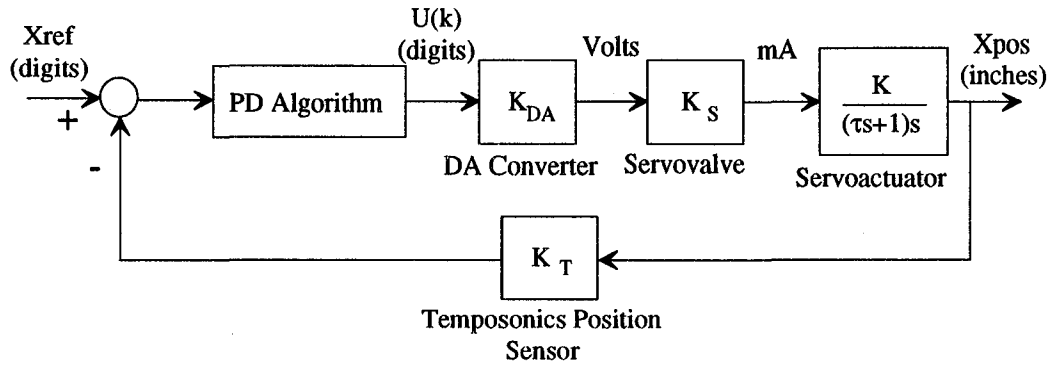


Figure 2.14 Position/Derivative Control Loop

2.4 Actuator Dynamics and Control

The dynamics and control of the Stewart platform actuators are discussed. Then it is shown how they are modeled in the VES simulation.

The six VES leg actuators are operated under closed-loop control, as illustrated in fig. 2.14. The position/derivative (PD) control algorithm is implemented digitally, in assembly language, with user-specified gains K_P and K_D. The six controllers, or LegSlaves, are based on the 8-bit Intel 8031 microprocessor. The PD algorithm is of the form

$$U(k) = K_P E(k) - K_D(X(k) - X(k-4)),$$

where K_P and K_D are proportional and derivative gains, respectively. The Temposonics transducer operates at 2900 Hz, so that new feedback values are available every 341 μs. The PD algorithm averages two successive values from the Temposonics counter to get $X(k)$, and ignores the third measurement. Thus, $X(k)$ is available every 1023 μs. This number is subtracted from the reference leg command, resulting in the error, $E(k)$. The PD code saves the previous four values of $X(k)$, so that $X(k-4)$ can be used in the

equation. This creates an averaging for the velocity signal, and prevents the subtraction of two nearly identical numbers when the platform is moving at a slow speed, resulting in the position feedback changing by about one bit per sample period.

The 16 bit digital to analog (D/A) converter uses one bit for the sign, and fifteen bits for the binary number representation. The D/A converter has a gain of

$$K_{DA} = \frac{10\ Volts}{(256)(2^{15}\ digits)} = 3.05x10^{-4}\ Volts/digit, \tag{2.7a}$$

where the factor of 256 scales the number from sixteen to fifteen bits, and there is a maximum output of ± 10 volts. The servo-amplifier has a gain of 4 mA/volt.

Position feedback information is provided through the Temposonics position transducer. The operation of the transducer is based on the principle of transmitting an ultrasonic pulse through the center of the actuator. Each LegSlave board contains a fourteen bit, 40 MHz, counter to measure the transit time of the pulse. The inverse of the pulse speed is 9.09 μs/inch, resulting in a transducer gain of

$$K_T = (40MHz)(9.09\mu s/inch) = 363.6\ counts/inch. \tag{2.7b}$$

Mueller provides a derivation of the actuator dynamics [6]. The method is briefly described here. Equation (2.8) is the flow through an aperture

$$Q = c_d A\sqrt{\frac{2}{\rho}\Delta p}, \tag{2.8}$$

where c_d is the discharge coefficient, A is the area, ρ is the fluid density, and the differential pressure is given by

$$\Delta p = p_S - p_L. \tag{2.9}$$

In equation (2.9), p_S and p_L are the supply and load pressures, respectively. The area of the aperture is the product of the circumference of the spool, w, and displacement of the spool, x. If the input current, i, is proportional to the displacement through a proportionality constant, k_j, the area becomes

$$A = wx = wk_j i. \tag{2.10}$$

Combining the above three equations results in a non-linear expression for the flow:

$$Q = c_d wk_j i\sqrt{\frac{2}{\rho}(p_S - p_{L)}}, \tag{2.11}$$

This expression is linearized in a Taylor series expansion about the valve displacement and load pressure, assuming a constant supply pressure, as shown in equation (2.12).

$$dQ = \frac{\delta Q}{\delta i}di + \frac{\delta Q}{\delta p_L}dp_L \tag{2.12}$$

Taking partial derivatives of equation (2.11), and evaluating them at the operating point of the actuator, yields the linearized flow and pressure coefficients

$$\left.\frac{\delta Q}{\delta i}\right|_{op} = c_d w\sqrt{\frac{2}{\rho}(p_S - p_L)} = C_x \tag{2.13a}$$

and

$$\left.\frac{\delta Q}{\delta p_L}\right|_{op} = \frac{-c_d wk_j i}{\sqrt{\frac{2}{\rho}(p_S - p_L)}} = -C_p. \tag{2.13b}$$

Integration of equation (2.12) about the operation point produces the flow equation

$$dQ\big|_{op} = dQ_{op} = \left.\frac{\delta Q}{\delta i}\right|_{op} i_{op} + \left.\frac{\delta Q}{\delta p_L}\right|_{op} p_{L_{op}} = C_x i_{op} - C_p p_{L_{op}}. \tag{2.14}$$

If the fluid is assumed to be incompressible, the actuator's continuity equation is

$$dQ_{op} = A_e \dot{y}, \tag{2.15}$$

where $\dot{y}$ is the velocity of the piston, and the effective area is related to applied pressure through equation (2.16).

$$p_L = \frac{m\ddot{y}}{A_e} \tag{2.16}$$

In this equation, m is the mass of the piston and actuator rod. Combining equations (2.14)-(2.16), and taking the Laplace transform results in the linearized open-loop transfer function for the actuator,

$$\frac{y(s)}{i(s)} = \frac{C_x}{\frac{C_p m}{A_e} s^2 + A_e s} = \frac{K}{s(\tau s + 1)}. \tag{2.17}$$

- **Simulation Model of the Actuator Dynamics and Control**

The VES control software concludes by communicating the new leg lengths to the actuators, and reading back the actual actuator lengths. The Simulink model performs these operations, along with converting the leg velocities and accelerations to SI units. These values are processed by the forward kinematics, along with the leg lengths, to determine the actual state of the platform, in Cartesian coordinates.

Figure 2.15 shows the position/derivative control block for one of the leg actuators. Each of the six PD control blocks are identical in structure, with differing parameters in the Servovalve and Actuator transfer function, depending on whether an individual leg is retracting or extending. Equation (2.17) presents the servo-valve and actuator transfer function. The value for K is 1.18 inches/sec/mA for actuator extension,

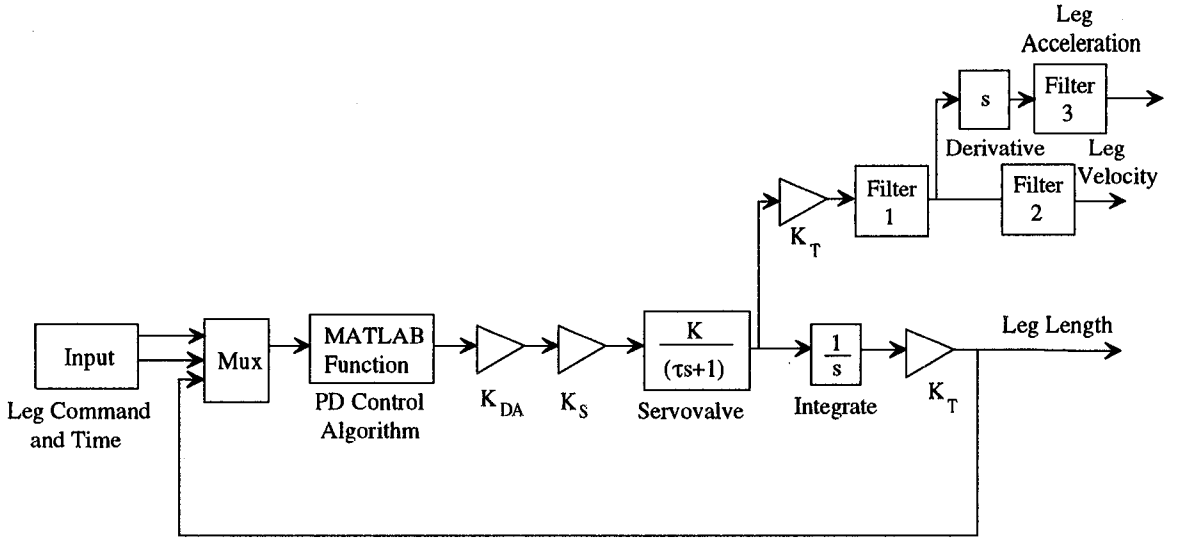


Figure 2.15 VES Computer Simulation: Position/Derivative Control Loop

and 0.77 inches/sec/mA for retraction, and τ =0.0064 seconds. The gains for the digital to analog converter, servo-amplifier, and Temposonics feedback transducer are given in section 2.5.1. K_T is the transducer gain defined in eq 2.7b. It has been determined that the bandwidth of the VES actuators is approximately 6-8 Hz [7]. If higher frequencies are generated by the admittance model, they will not be reproduced during simulation.

Appendix E contains a derivation that shows that there are possible stability problems with the admittance control concept as it is employed by the VES. The proof is for a hypothetical system, so the stability problems in the actual hardware might not be as pronounced at the proof indicates. Referring to fig 2.15, Filters 1 through 3 were added to the simulation to reduce the stability problems. Filter 1 is a second order Butterworth filter with a cut-off frequency of 30 rad/sec (4.8 Hz), and Filters 2 and 3 are third order Butterworth filters with cut-off frequencies of 30 rad/sec. These filters create the bandwidth (6-8 Hz) associated with the hardware. Filters with different cut-off

frequencies (40 and 50 rad/sec) were experimented with in the software to cover the range of 6-8 Hz, but could not adequately reduce the stability problems described in Appendix E. This could be due to linear filters approximating the non-linear dynamics of the actuators.

2.5 Forward Kinematics

The inverse position kinematics solution provides the joint angles or actuator lengths required to reach a specified end-effector pose within the workspace of a robot. The forward kinematics solution performs the opposite function, i.e., given the actuator values or joint angles, compute the Cartesian pose. Forward kinematics can be used in simulations, such as the one presented here. For the Simulink model, the actuator lengths, velocities, and accelerations produced by the PD control loop are processed by the forward kinematics solution to determine the actual motion of the platform. This information serves two purposes. First, it is fed back to the model of the robot to include the base motion in the calculation of the wrench due to the robot. Second, it is compared to the commanded motion from the admittance model defined in the software, to estimate the accuracy with which the simulated VES hardware models the desired motion.

Williams derives the forward position and velocity kinematics for the NASA Stewart platform [9]. The next part of this section follows the development of his equations. The equations are then expanded upon to derive the forward acceleration kinematics.

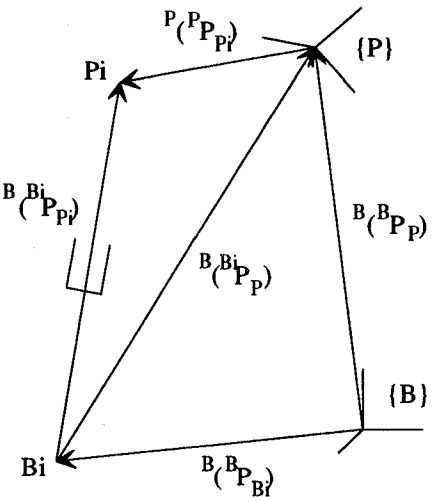


Figure 2.16 Vector Description of a Single Actuator[9]

The vector description of the i[th] actuator leg is shown in fig 2.16, where {P} and {B} are the platform and base frames, respectively, and B_i and P_i represent the locations of the i[th] base and platform ball joints. The super and subscript notation on the position vectors is as follows: ${}^{l}({}^{j}P_{k})$ is the vector from point j to point k, expressed in the l coordinate system. If frame $\{B_i\}$ is aligned with frame {B}, then the following vector loop closure equation is valid

$$ {}^{B}({}^{Bi}P_{Pi}) = {}^{B}({}^{B}P_{P}) - {}^{B}({}^{B}P_{Bi}) + {}^{B}_{P}R\,{}^{P}({}^{P}P_{Pi}), \tag{2.18} $$

where ${}^{B}_{P}R$ is the following X-Y-Z fixed-axis rotation matrix [16] :

$$ \left[{}^{I}_{P}R\right] = \begin{bmatrix} \cos\gamma\cos\beta & \cos\gamma\sin\beta\sin\alpha - \sin\gamma\cos\alpha & \cos\gamma\sin\beta\cos\alpha + \sin\gamma\sin\alpha \\ \sin\gamma\cos\beta & \sin\gamma\sin\beta\sin\alpha - \cos\gamma\cos\alpha & \sin\gamma\sin\beta\cos\alpha - \cos\gamma\sin\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix} \tag{2.19} $$

The vectors describing the ball joint locations are ${}^{P}({}^{P}P_{Pi}) = \{P_{ix}, P_{iy}, 0\}^{T}$ and ${}^{B}({}^{B}P_{Bi}) = \{B_{ix}, B_{iy}, 0\}^{T}$, and are known from geometry. The location of the platform with respect to the base is ${}^{B}({}^{B}P_{P}) = \{x, y, z\}^{T}$. Substituting these parameters into eq (2.18), and simplifying yields

$$ {}^{B}({}^{Bi}P_{Pi}) = \begin{Bmatrix} x + R(1,1)P_{ix} + R(1,2)P_{iy} - B_{ix} \\ y + R(2,1)P_{ix} + R(2,2)P_{iy} - B_{iy} \\ z + R(3,1)P_{ix} + R(3,2)P_{iy} \end{Bmatrix}, \tag{2.20} $$

where the super- and sub-scripts have been omitted from the rotation matrix. In eq (2.20), the unknowns are the platform position, $\{x, y, z\}^{T}$, and orientation $\{\theta_x, \theta_y, \theta_z\}^{T}$, contained in the rotation matrix. Using the Euclidean norm to describe the i^{th} leg,

$$ L_i^2 = \left\| {}^{B}({}^{Bi}P_{Pi}) \right\|_2, \tag{2.21} $$

the i^{th} constraint equation is

$$ \begin{aligned} f_i(X) = x^2 + y^2 + z^2 + 2(P_{ix}R(1,1) + P_{iy}R(1,2))(x - B_{ix}) + 2(P_{ix}R(2,1) + P_{iy}R(2,2))(y - B_{iy}) + \\ 2(P_{ix}R(3,1) + P_{iy}R(3,2))z - 2(xB_{ix} + yB_{iy}) + (B_{ix}^2 + B_{iy}^2)^2 + (P_{ix}^2 + P_{iy}^2)^2 - L_i^2 = 0 \end{aligned} \tag{2.22} $$

for i=1,2,...,6, and $X = \{x, y, z, \theta_x, \theta_y, \theta_z\}^{T}$. Eqs (2.22) represent six coupled, non-linear equations, with six unknowns.

Williams presents a method for determining the pose of the platform using the Newton-Raphson iterative technique to solve eqs (2.22). Starting with an initial guess for the unknowns, solve

$$ X_{new} = X_{old} - \left[\frac{\partial f_i}{\partial x_j} \right]^{-1} f_i(X_{old}) = X_{old} - \left[J_{NR} \right]^{-1} f_i(X_{old}), \tag{2.23} $$

where J_{NR} is the Newton-Raphson Jacobian matrix. Then, using the new value as an initial guess, iterate on eq (2.23) until convergence to the desired degree of accuracy, thus solving the forward position kinematics.

The Jacobian is a multidimensional form of the derivative that maps velocities from one space into another. For the VES, the Jacobian maps actuator velocities into Cartesian velocities according to eq (2.24).

$$\{\dot{X}\} = [J]\{\dot{L}\} \tag{2.24}$$

Taking the time derivative of eq (2.22) and simplifying yields

$$\dot{L}_i = \frac{1}{2L_i}\sum_{j=1}^{6}\frac{\partial f_i}{\partial X_j}\{\dot{\hat{X}}_j\} = [J_M]^{-1}\{\dot{\hat{X}}\}, \tag{2.25}$$

where J_M is the modified Jacobian matrix. This equation is similar to an inverted form of eq (2.24). However, the Cartesian rotational velocities in eq (2.24) are different than the Euler rotational velocities in eq (2.25):

$$\{\dot{X}\} = \{\dot{x},\dot{y},\dot{z},\omega_x,\omega_y,\omega_z\}^T \neq \{\dot{\hat{X}}\} = \{\dot{x},\dot{y},\dot{z},\dot{\theta}_x,\dot{\theta}_y,\dot{\theta}_z\}^T. \tag{2.26}$$

The relationship between the velocities is given by the 3-2-1 Euler rotation

$$\{\dot{\theta}\} = [A]\{\omega\}, \tag{2.27a}$$

or

$$\begin{Bmatrix}\dot{\theta}_x\\ \dot{\theta}_y\\ \dot{\theta}_z\end{Bmatrix} = \begin{bmatrix}1 & \sin\theta_x\tan\theta_y & \cos\theta_x\tan\theta_y\\ 0 & \cos\theta_x & -\sin\theta_x\\ 0 & \sin\theta_x/\cos\theta_y & \cos\theta_x/\cos\theta_y\end{bmatrix}\begin{Bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{Bmatrix}. \tag{2.27b}$$

The relation between the Newton-Raphson Jacobian given in eq (2.23) and the inverted modified Jacobian in eq (2.25) is

$$[J_M]^{-1} = \left[\frac{1}{2L_i} J_{NR}\right]. \tag{2.28}$$

Finally, the Jacobian in eq (2.24) is related to the Newton-Raphson Jacobian through eq (2.29).

$$[J]^{-1} = \begin{bmatrix} [J_M^{-1}]_{UL} & [J_M^{-1}]_{UR}[A] \\ [J_M^{-1}]_{LL} & [J_M^{-1}]_{LR}[A] \end{bmatrix} \tag{2.29}$$

Equations (2.28), (2.29), and (2.24) present an algorithm for computing the forward velocity kinematics. First, calculate the Newton-Raphson Jacobian, and divide the i^{th} row by the i^{th} leg length, to obtain the inverted modified Jacobian. Then, create the $[A]$ matrix and solve eq (2.29) for the inverted Jacobian. Finally, invert this matrix and evaluate eq (2.24). This illustrates that J_{NR} can be used for both the position and velocity kinematics, instead of determining both J_{NR} (for position) and J (for velocity) at each simulation time step.

The next part of the derivation involves the forward acceleration kinematics. This is an extension of the velocity kinematics, and begins with the second time derivative of eq (2.22):

$$\begin{aligned} 2\dot{L}_i^2 + 2L_i\ddot{L}_i &= \sum_{j=1}^{6} \frac{\partial}{\partial t}\left(\frac{\partial F_i}{\partial X_j}\frac{\partial \hat{X}}{\partial t}\right) \\ &= \sum_{j=1}^{6} \frac{\partial^2 F_i}{\partial X_j^2}\frac{\partial X_i}{\partial t}\frac{\partial \hat{X}}{\partial t} + \sum_{j=1}^{6} \frac{\partial F_i}{\partial X_j}\frac{\partial}{\partial t}\dot{\hat{X}} \\ &= \sum_{j=1}^{6} \frac{\partial^2 F_i}{\partial X_j^2}\dot{\hat{X}}^2 + \sum_{j=1}^{6} \frac{\partial F_i}{\partial X_j}\ddot{\hat{X}} \end{aligned} \tag{2.30}$$

where the $\hat{X}$ terms are Euler coordinates, and $\frac{\partial^2 F_i}{\partial X_j^2}$ is the derivative of the Newton-Raphson Jacobian. Equation (2.30) is solved for $\ddot{L}_i$ to acquire the inverse kinematics, as in eq (2.31), or for $\ddot{\hat{X}}$ to acquire the forward kinematics solution, as in eq (2.32).

$$\ddot{L}_i = \frac{1}{2L_i}\left(\sum_{j=1}^{6}\frac{\partial^2 F_i}{\partial X_j^2}\dot{\hat{X}}^2 + \sum_{j=1}^{6}\frac{\partial F_i}{\partial X_j}\ddot{\hat{X}} - 2\dot{L}_i^2\right) \tag{2.31}$$

$$\ddot{\hat{X}} = \left(\sum_{j=1}^{6}\frac{\partial F_i}{\partial X_j}\right)^{-1}\left(2\dot{L}_i^2 + 2L_i\ddot{L}_i - \sum_{j=1}^{6}\frac{\partial^2 F_i}{\partial X_j^2}\dot{\hat{X}}^2\right) \tag{2.32}$$

In each of these equations, eq (2.27) relates the Cartesian and Euler velocities. Equation (2.33) is the time derivative of this equation, and shows the relation between the Cartesian and Euler accelerations.

$$\begin{Bmatrix}\ddot{\theta}_x\\ \ddot{\theta}_y\\ \ddot{\theta}_z\end{Bmatrix} = \begin{bmatrix}1 & \sin\theta_x\tan\theta_y & \cos\theta_x\tan\theta_y\\ 0 & \cos\theta_x & -\sin\theta_x\\ 0 & \sin\theta_x/\cos\theta_y & \cos\theta_x/\cos\theta_y\end{bmatrix}\begin{Bmatrix}\dot{\omega}_x\\ \dot{\omega}_y\\ \dot{\omega}_z\end{Bmatrix} + \begin{bmatrix}\tan\theta_y & 1/\cos\theta_y & 0\\ 0 & 0 & -\cos\theta_y\\ 1/\cos\theta_y & \tan\theta_y & 0\end{bmatrix}\begin{Bmatrix}\dot{\theta}_x\dot{\theta}_y\\ \dot{\theta}_y\dot{\theta}_z\\ \dot{\theta}_z\dot{\theta}_x\end{Bmatrix} \tag{2.33}$$

Therefore, the forward kinematics solution for the Cartesian accelerations is

$$\{\ddot{X}\} = \{\ddot{x},\ddot{y},\ddot{z},\ddot{\theta}_x,\ddot{\theta}_y,\ddot{\theta}_z\}^T, \tag{2.34}$$

where $\ddot{x}$, $\ddot{y}$, and $\ddot{z}$ are computed from eq (2.32).

Equations (2.23), (2.24), and (2.32) present the forward kinematics equations that are utilized by the Simulink software. The position is obtained iteratively, while the velocity and acceleration are obtained in one step.

3. Computer Simulation of SRMS-Manipulator Interaction

This chapter presents an independent but approximate computer simulation of the SRMS-manipulator dynamic interaction that will act as a comparison to the simulated motion of the VES. The model also serves to simulate the disturbances that any general robot may impose on a flexible base. The model assumes small angle motion of the base, and applies to any two dynamic systems if at least one is defined in mode space.

An SRMS admittance model has been developed from a NASTRAN model that includes 132 degrees-of-freedom [11]. Numerically integrating 132 coupled differential equations is computationally expensive. Transforming to mode space and truncating the mode shapes matrix reduces the number of degrees of freedom to be integrated, thus reducing the number of computations.

The following paragraphs describe the SRMS-manipulator computer simulation. Figure 3.1 shows the elements that comprise the simulation.

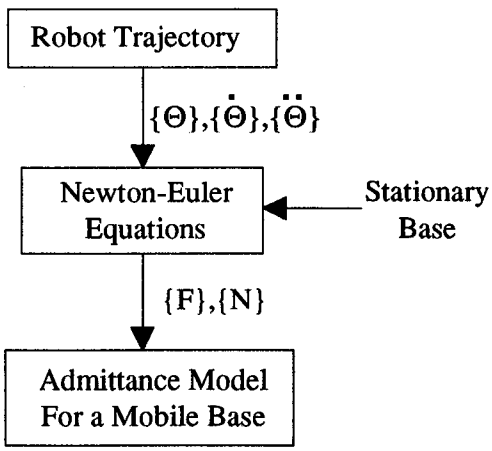


Figure 3.1 Control Loop for a Model of the Interaction of a Mobile Base and an Active Robot

During the simulation, the robot is commanded to move in space. This is accomplished through trajectory generation, resolved-rate control, a vision system, force control, or another scheme. Currently, each joint of the robot is commanded to follow the path of a fifth-order polynomial, from an initial to a final angle, with zero initial and final velocity, and zero initial and final acceleration. At the new simulation time step, each joint has associated with it an angle, an angular rate, and an angular acceleration.

Craig [16] and Luh, et al [17] present an algorithm for computing the wrench at the base of the robot, due to its motion, based on the Newton-Euler equations. Beginning with link 1 and propagating to link n, the link velocities and accelerations are computed. The Newton-Euler equations are also evaluated for each link. Then, the process is reversed from link n to link 1, this time calculating the forces and moments of interaction between successive links. The final reaction wrench at link 1 is transformed to the base of the robot. These calculations are performed assuming the robot is attached to a fixed base. The Newton-Euler iterative equations are included in Appendix A.

The wrench that is generated is due to the motion of the robot alone, and is input to an admittance model. The admittance model is discussed in the following paragraphs. This simulation is an approximation to the motion, since it assumes that the motion due to a moving robot on a moving base is equal to the sum of the motion of the base acting on a stationary robot, and the motion of the robot acting on a stationary base. The equations involved are non-linear, so splitting the wrench in this manner is not exact.

- **Augmenting the Modal Mass Matrix**

The following derivation assumes small angle motion of the flexible base, and that the base mass, spring, and damping matrices do not change during the motion. The results are valid for any flexible base that is defined in mode space.

In general, the robot's motion causes the flexible base to vibrate and move. These combined motions create a wrench at the base of the manipulator. The wrench components can be split into a part due to the base motion acting on a stationary robot, $[M_1]\{\underline{\ddot{x}}\}$, and a part due to the robot's motion acting on a stationary base, $\{\underline{W_{ext}}\}$, so that the admittance model has the form

$$[M]\{\ddot{\bar{x}}\}+[C]\{\dot{\bar{x}}\}+[K]\{\bar{x}\}=-[M_1]\{\underline{\ddot{x}}\}+\{\underline{W_{ext}}\}, \tag{3.1}$$

where

$\{\bar{x}\}=\{X,\Theta\}^T$ is the state of the n-DOF mobile base, comprised of a position vector, and angular rotations,

$\{\underline{\ddot{x}}\}$ is the acceleration vector for the robot, at its center of mass, and includes the centripetal acceleration, $\omega\times(\omega\times r)$, and the tangential acceleration, $\dot{\omega}\times r$,

$\{\underline{W_{ext}}\}=\{F_{ext},N_{ext}\}^T$ is the external wrench due to the motion of the robot, comprised of the force and moment vectors,

$[M]=\begin{bmatrix} m_{base} & 0 \\ 0 & I_{base} \end{bmatrix}$ is the nxn mass matrix of the mobile base, made up of the mass and inertia sub-matrices,

$[C]=\begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix}$ is the nxn damping matrix, comprised of four submatrices,

$[K] = \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix}$ is the nxn stiffness matrix, comprised of four sub-matrices,

and

$[M_1] = \begin{bmatrix} m_1 & 0 \\ 0 & {}^{c}I_1 \end{bmatrix}$ is the mass matrix of the robot, made up of the mass and inertia sub-matrices, defined at its center of mass.

Equation (3.1) is an nx1 vector equation describing the motion of the robot-base interaction. The equation is split into the following two (n/2)x1 position and orientation vectors

$$\begin{aligned} & m_{base}\ddot{X} + c_1\dot{X} + c_2\dot{\Theta} + k_1 X + k_2\Theta \\ &= -m_1\{\ddot{X} + \ddot{\Theta}_c \times P_c + \dot{\Theta}_c \times (\dot{\Theta}_c \times P_c)\} + F_{ext} \\ &= -F_c + F_{ext} \end{aligned} \tag{3.2}$$

and

$$\begin{aligned} & I_{base}\ddot{\Theta} + c_3\dot{X} + c_4\dot{\Theta} + k_3 X + k_4\Theta \\ &= -{}^{c}I_1\ddot{\Theta}_c - P_c \times \{m_1\{\ddot{X} + \ddot{\Theta}_c \times P_c + \dot{\Theta}_c \times (\dot{\Theta}_c \times P_c)\}\} + N_{ext}, \\ &= -{}^{c}I_1\ddot{\Theta}_c - P_c \times F_c + N_{ext} \end{aligned} \tag{3.3}$$

where $\{\ddot{\underline{x}}\}$ has been expanded to show the acceleration terms. In these two equations $\dot{\Theta}_c$ and $\ddot{\Theta}_c$ are the angular velocities and accelerations of the center of mass of the manipulator. If the robot is assumed to be rigid, and its center of mass frame is aligned with the frame of the mobile base, then $\dot{\Theta}_c = \dot{\Theta}$ and $\ddot{\Theta}_c = \ddot{\Theta}$. P_c is the vector location of the center of mass of the robot, defined in the robot's base coordinate frame.

The conversion from physical to mode space is $\bar{x} = \Phi\eta$, where Φ is the nxn modeshape matrix. The matrix is composed of two (n/2)xn upper and lower matrices such that

$$X = \Phi_u \eta$$
$$\Theta = \Phi_l \eta \tag{3.4}$$

Mapping the motion into mode space, equations (3.2) and (3.3) become

$$\begin{aligned} &m_{base}\Phi_u\ddot{\eta} + c_1\Phi_u\dot{\eta} + c_2\Phi_l\dot{\eta} + k_1\Phi_u\eta + k_2\Phi_l\eta = \\ &-m_1\{\Phi_u\ddot{\eta} + \{\Phi_l\ddot{\eta}\}\times P_c + \{\Phi_l\dot{\eta}\}\times(\{\Phi_l\dot{\eta}\}\times P_c)\} + F_{ext} \end{aligned} \tag{3.5}$$

and

$$\begin{aligned} &I_{base}\Phi_l\ddot{\eta} + c_3\Phi_u\dot{\eta} + c_4\Phi_l\dot{\eta} + k_3\Phi_u\eta + k_4\Phi_l\eta = \\ &-{}^cI_1\Phi_l\ddot{\eta} - P_c\times\{m_1\{\{\Phi_u\ddot{\eta}\} + \{\Phi_l\dot{\eta}\}\times P_c + \\ &\{\Phi_l\dot{\eta}\}\times(\{\Phi_l\dot{\eta}\}\times P_c)\}\} + N_{ext} \end{aligned} \tag{3.6}$$

Equations (3.5) and (3.6) are now re-combined into the following nx1 equation representing the motion of the combined system

$$\begin{aligned} &\begin{bmatrix} m_{base} & 0 \\ 0 & I_{base} \end{bmatrix}\begin{bmatrix} \Phi_u \\ \Phi_l \end{bmatrix}\ddot{\eta} + \begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix}\begin{bmatrix} \Phi_u \\ \Phi_l \end{bmatrix}\dot{\eta} + \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix}\begin{bmatrix} \Phi_u \\ \Phi_l \end{bmatrix}\eta = \\ &\begin{Bmatrix} F_{ext} \\ N_{ext} \end{Bmatrix} - \begin{bmatrix} m_1 & 0 \\ 0 & {}^cI_1 \end{bmatrix}\begin{bmatrix} \Phi_u \\ \Phi_l \end{bmatrix}\ddot{\eta} + \\ &\begin{Bmatrix} -\{\{\Phi_l\ddot{\eta}\}\times P_c + \{\Phi_l\dot{\eta}\}\times(\{\Phi_l\dot{\eta}\}\times P_c)\} \\ \{-P_c\times\{m_1\{\{\Phi_u\ddot{\eta}\} + \{\Phi_l\dot{\eta}\}\times P_c + \{\Phi_l\dot{\eta}\}\times(\{\Phi_l\dot{\eta}\}\times P_c)\}\}\} \end{Bmatrix} \end{aligned}, \tag{3.7}$$

or

$$M\Phi\ddot{\eta} + C\Phi\dot{\eta} + K\Phi\eta = -M_1\Phi\ddot{\eta} - \begin{Bmatrix} W_u \\ W_l \end{Bmatrix} + \begin{Bmatrix} F_{ext} \\ N_{ext} \end{Bmatrix} \tag{3.8}$$

This is the mode space version of eq (3.1). Pre-multiplying by Φ^T and using mass-normalized mode shapes yields

$$I\ddot{\eta} + \begin{bmatrix} \ddots & & \\ & 2\zeta\omega_i & \\ & & \ddots \end{bmatrix}\dot{\eta} + \begin{bmatrix} \ddots & & \\ & \omega_i^2 & \\ & & \ddots \end{bmatrix}\eta = -\Phi^T M_1\Phi\ddot{\eta} - \Phi_u{}^T W_u - \Phi_l{}^T W_l + \Phi^T W_{ext}, \tag{3.9}$$

for i=1,2,...,n and

$\Phi^T M\Phi = I$,

$\Phi^T C\Phi = diag[2\zeta\omega]$, is the diagonal matrix of damping values, and

$\Phi^T K\Phi = diag[\omega^2]$, is the diagonal matrix of natural frequencies.

Equation (3.9) has cross product terms embedded within it. Some of these are altered using the skew-symmetric position-cross matrix:

$$P\times = \begin{bmatrix} 0 & -P_z & P_y \\ P_z & 0 & -P_x \\ -P_y & P_x & 0 \end{bmatrix} = \Re \tag{3.10}$$

Simplifying eq (3.9) and bringing the $\ddot{\eta}$ terms to the left-hand side of the equation produces

$$A\ddot{\eta} = -[2\zeta\omega]\dot{\eta} - [\omega^2]\eta + [\Phi_u{}^T + \Phi_l{}^T\Re]m_1\{\{\Phi_l\dot{\eta}\}\times\{\Re\Phi_l\dot{\eta}\}\} + \Phi^T\underline{W_{ext}} \tag{3.11}$$

where

$$A = I + \Phi^T M_1\Phi - \Phi_u{}^T m_1\Re\Phi_l + \Phi_l{}^T\Re m_1\Phi_u - \Phi_l{}^T\Re m_1\Re\Phi_l$$

Pre-multiplying by A^{-1} decouples the $\ddot{\eta}$ terms, facilitating the conversion to state space.

$$\ddot{\eta} = [A]^{-1}\{-[2\zeta\omega]\dot{\eta} - [\omega^2]\eta + [\Phi_u{}^T + \Phi_l{}^T\Re]m_1\{\{\Phi_l\dot{\eta}\}\times\{\Re\Phi_l\dot{\eta}\}\} + \Phi^T\underline{W_{ext}}\}. \tag{3.12}$$

Equation (3.12) represents the n coupled, non-linear, mode space equations of motion for the mobile base-robot interaction. This is the modal representation of the admittance model shown in eq (3.1). In general, there is no closed-form solution to these equations. Numerical integration methods, such as the Runge-Kutta equations, are used to determine their solution. Truncating the modeshapes matrix, Φ, reduces the number of differential equations to be integrated. Equations (3.13) show the conversion back to physical space:

$$\begin{aligned} \bar{x} &= \Phi\eta \\ \dot{\bar{x}} &= \Phi\dot{\eta} \end{aligned} \tag{3.13a}$$

and

$$\begin{aligned} \ddot{\bar{x}} &= \Phi\ddot{\eta} \\ &= \Phi A^{-1}\{-[2\zeta\omega]\dot{\eta} - [\omega^2]\eta \\ &\quad +[\Phi_u{}^T + \Phi_l{}^T\Re]m_1\{\{\Phi_l\dot{\eta}\}\times\{\Re\Phi_l\dot{\eta}\}\} + \Phi^T\underline{W_{ext}}\} \end{aligned} \tag{3.13b}$$

This derivation is valid for the interaction of any two dynamic systems, if at least one is defined in mode space. There are two drawbacks to this method. The first is that the mass parameters of the manipulator must be known, but this is an issue that arises in any computer simulation. The second is that the nxn matrix of mass values, A, must be inverted at each time step. This can become computationally intensive and time-consuming if a large number of modes are included in the simulation. For slow motions of the manipulator its mass matrix $[M_1]$ does not appreciably change during a single simulation time step. In this situation, the matrix A can be computed and inverted only every N time steps, where N is the number of time steps over which the matrix is assumed constant.

4. System Performance

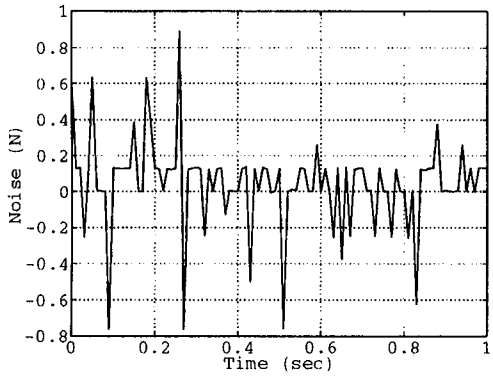
This chapter presents experimental results. High frequency noise and low frequency drift are shown to be present on the measurements from the force sensor. The VES simulation, performing trajectory and admittance control motions, is compared to experimental hardware results from MIT. The computer simulations developed in chapters 2 and 3 are compared with the PUMA 560 mounted on the SRMS end effector. Results are also given for the DOSS, a proposed flight arm, mounted on the SRMS.

4.1 Force-Torque Sensor Performance

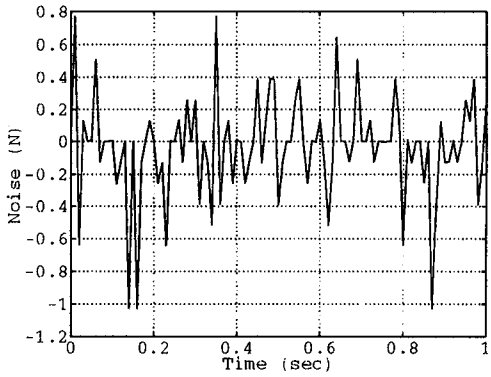
The force-torque sensor manufactured by Advanced Mechanical Technology, Inc., and described in section 2.2, is analyzed. Long and short duration experiments, that verify the presence of noise components on the measurements, are discussed.

4.1.1 Short Term Experiments

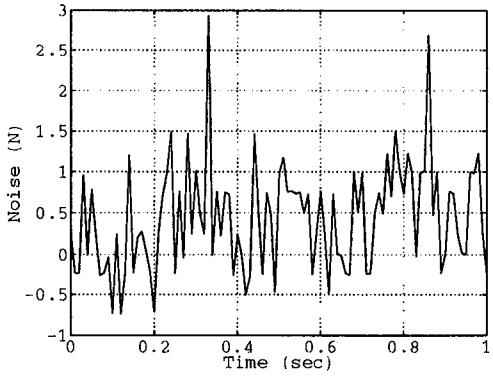
A number of short term experiments were performed on the force sensor to analyze the presence of high frequency noise on the output of the sensor. The sensor was unloaded, so it should have measured no wrench components. Figure 4.1 shows representative results for one second of data collected at 100 Hz. The forces in the X- and Y- directions have maximum amplitudes under 1.0 N, while the force in the Z-direction is slightly larger, with spikes up to 3.0 N. The moments are all under 1.0 Nm.



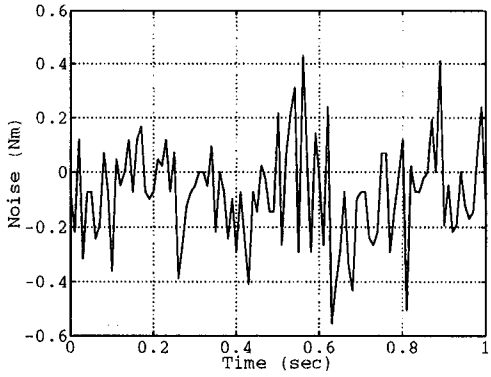
(a) X Force



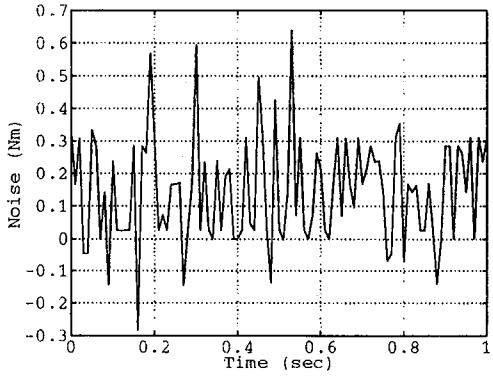
(b) Y Force



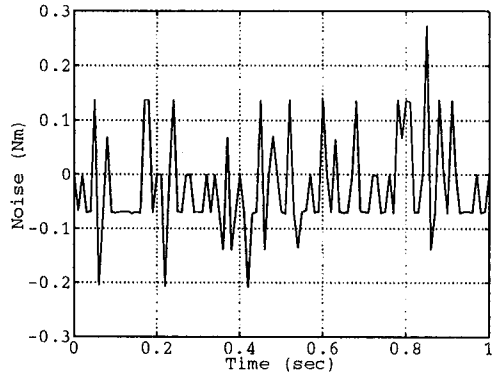
(c) Z Force



(d) X Moment



(e) Y Moment



(f) Z Moment

Figure 4.1 Noise Associated with the Forces and Moments Measured by the Force-Torque Sensor for a 1 Second Experiment Sampling at 100 Hz, (a) X Force, (b) Y Force, (c) Z Force, (d) X Moment, (e) Y Moment, (f) Z Moment

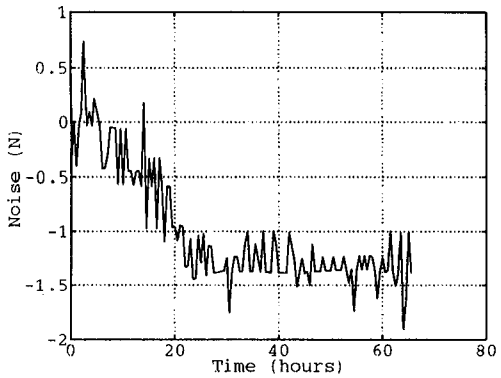
The force-torque sensor is composed of strain gages in Wheatstone bridge formations. These are all linear devices, and should not introduce any extraneous noise components. The source of the noise was traced back to the ground, which was not a clean signal.

The presence of the noise on the force and moment measurements is capable of corrupting the results of experiments. The data will not be reliable if the reaction wrench that the force sensor is reading is on the order of the noise component.

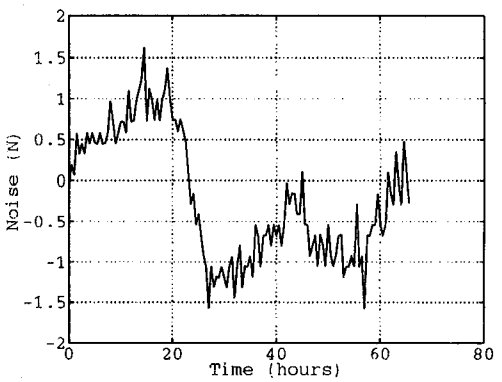
4.1.2 Long Term Experiments

Longer duration experiments were performed to determine the effects of the environment on sensor readings. Figure 4.2 illustrates tests that were performed over a period of 65 hours, with data collected at 30 minute intervals. The data is observed to contain a low frequency component with a period of approximately 24 hours. Figure 4.3 shows a time history of the temperature in the laboratory, for the duration of the experiment. There appears to be a correlation between temperature and sensor measurements.

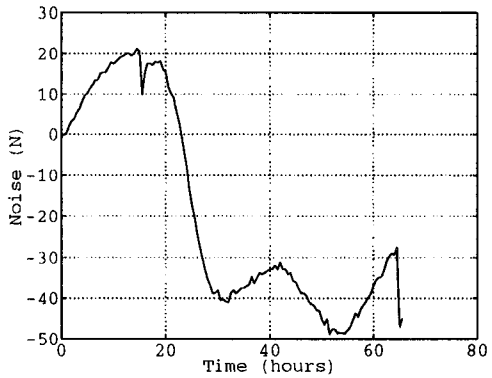
Advanced Mechanical Technology, Inc. states that the sensor has a $0.01\%/^{0}C$ sensitivity change with temperature over the region from -17 to + $52^{0}C$ ["Model OR6-5-1 Biometrics Platform Instruction Manual", Advanced Mechanical Technology, Inc., Newton, Massachusetts, January 1987]. Therefore, the $2.5^{0}C$ temperature fluctuation over the duration of the experiment does not account for the large oscillations in sensor measurements. It is possible the sensor is sensitive to other environmental factors, such as



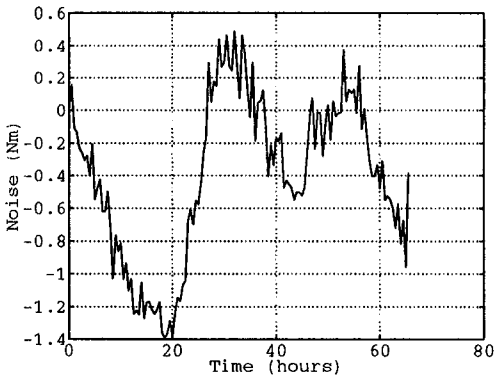
(a) X Force



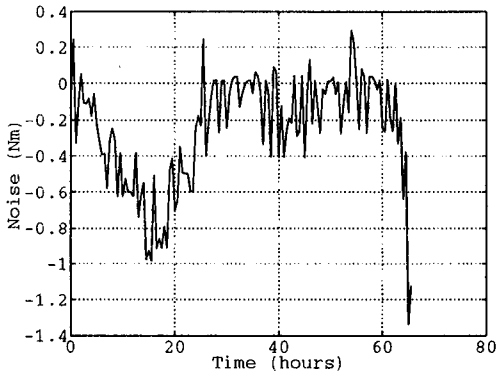
(b) Y Force



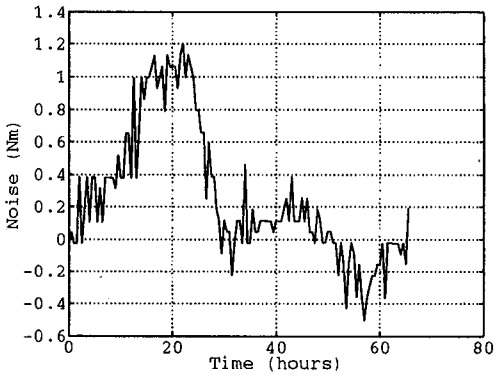
(c) Z Force



(d) X Moment



(e) Y Moment



(f) Z Moment

Figure 4.2 Long-Term Drift Associated with the Forces and Moments Measured by the Force-Torque Sensor for a 65 Hour Experiment Sampling Every 30 Minutes, (a) X Force, (b) Y Force, (c) Z Force, (d) X Moment, (e) Y Moment, (f) Z Moment

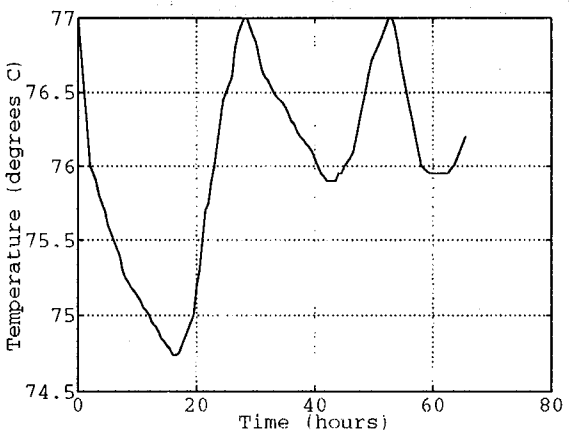


Figure 4.3 Temperature Time History for Duration of 65 Hour Experiment

electro-magnetic fields and computer activity in the laboratory. During the evening hours, when the temperature drops, the computer activity in the laboratory also declines. This could affect the power signal that powers the sensor.

Most experiments will be performed on the order of minutes, as opposed to hours. These results show, however, that the sensor will need to be re-calibrated every 3 to 4 hours to ensure reliable readings that are independent of long-term drift. The force in the Z-direction is most susceptible to environmental effects, so it will need to be monitored more closely than the other wrench readings.

4.2 VES Computer Simulation Validation

Section 2.3 illustrated that the VES is capable of admittance and trajectory control. Kuklinski has studied the MIT Stewart Platform dynamic motion capabilities [7]. His experiments involved both trajectory and admittance control on the platform. The NASA Langley VES is not currently operational, so the VES computer simulation described in

chapter 2 is compared to the results from MIT. In this section, all motions are about the platform frame, {P}, from fig 2.1. The rotations about the x-, y-, and z-axes are alpha, beta, and gamma, respectively.

4.2.1 Trajectory Control

The VES operates under PD control, with user-specified values for the position and derivative gains. The VES simulation was used to track a sinusoidal motion in the z-direction, with an amplitude of ± 0.2m and frequency of 0.5 Hz. The position and derivative gains were 8500 and 0, respectively. Figure 4.4 shows the commanded z-translation, and the simulated z-translation, as computed by the forward kinematics code (refer to fig 2.2). The error between the commanded and simulated positions is presented in fig 4.5a. Figure 4.5b shows the position error associated with the MIT VES hardware performing the same maneuver with identical gains. All experiments were performed for a duration of three seconds, with a 100 Hz sampling rate.

The results show that the platform motion lags behind the commanded translation, which is common in mechanical systems. Comparing figs 4.5a and 4.5b, it is observed that the simulation data are similar to the MIT experimental results. The larger position error associated with fig 4.5b could be due to friction or hysteresis in the hardware, which is not currently modeled in the computer simulation. After the NASA VES is operational, experimental test results can be compared to the simulation, which can be altered accordingly. The simulation was not changed to model the MIT results exactly because all physical systems are different, and the simulator is supposed to model the NASA VES. In

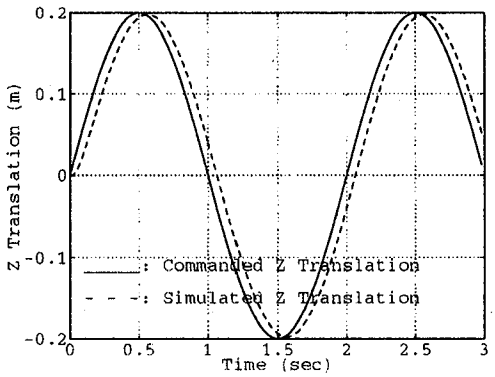
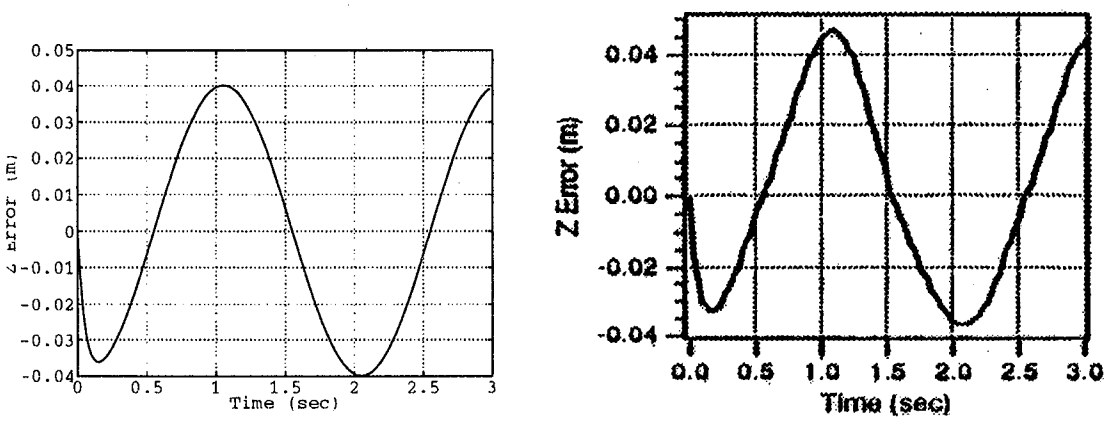


Figure 4.4 Commanded and Simulated Z-Translation for Trajectory Control: ±0.2m Amplitude, 0.5 Hz Frequency



(a) VES Computer Simulation (b) MIT VES Hardware[7]

Figure 4.5 Position Errors when Tracking Z-Translation of VES: (a) VES Computer Simulation (b) MIT VES Hardware[7]

addition, the data from each leg actuator is required to modify the simulation and such data was not available from MIT.

As a second test, the simulation was used to track sinusoidal motion about the x-axis, with an amplitude of $\pm 20^0$, 0.5 Hz frequency, and the same PD gains. The commanded and simulated rotation, from the computer simulation of the Stewart platform, are illustrated in figure 4.6. Figures 4.7a and 4.7b show the rotational errors between the commanded and simulated rotation from the computer simulation, and errors from the same experiments performed at MIT. The simulation errors are slightly larger, but are close enough to verify that the simulator performs rotational control correctly.

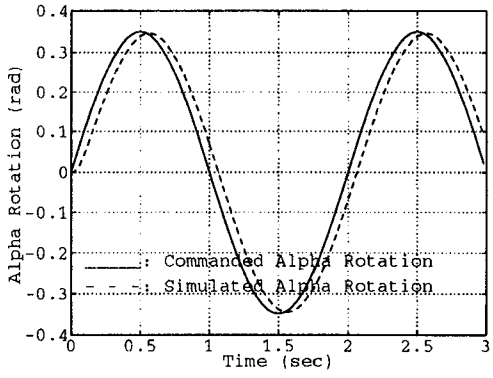
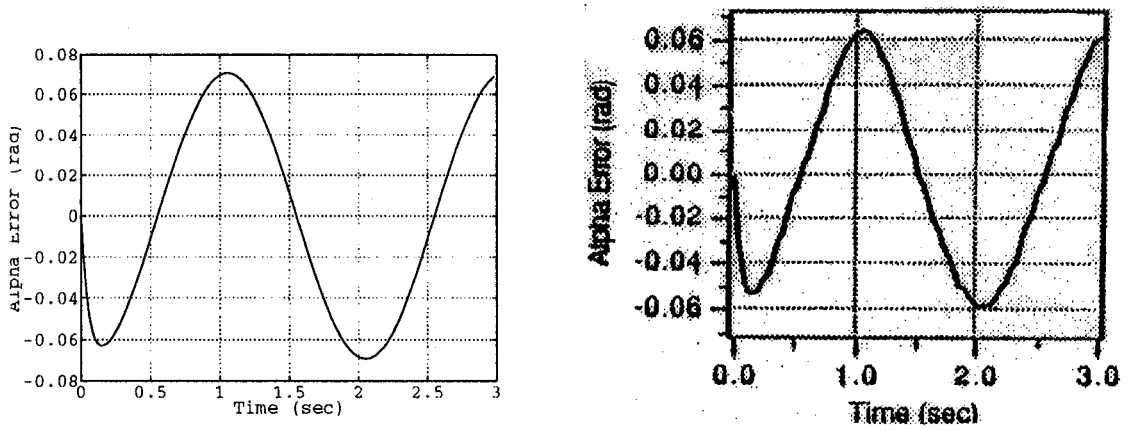


Figure 4.6 Commanded and Simulated Alpha-Rotation for Trajectory Control: $\pm 20^0$ Amplitude, 0.5 Hz Frequency



(a) VES Computer Simulation (b) MIT VES Hardware[7]

Figure 4.7 Position Errors when Tracking Alpha-Rotation of VES: (a) VES Computer Simulation (b) MIT VES Hardware[7]

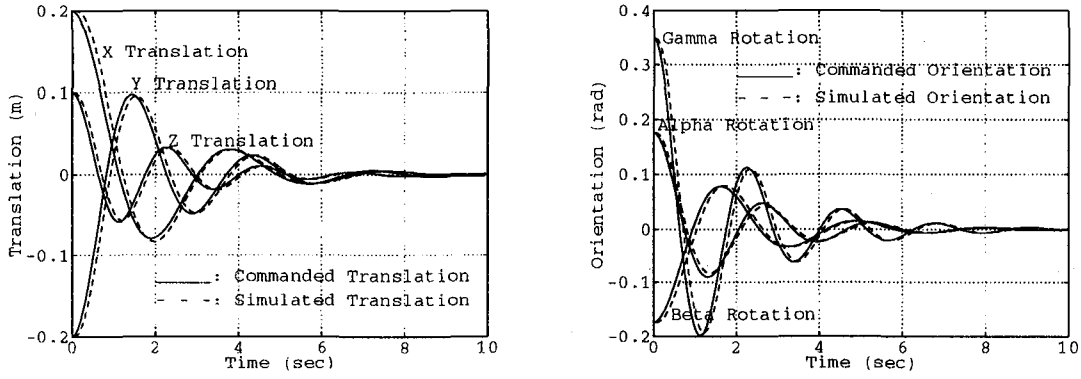
4.2.2 Admittance Trajectories

The VES Simulink simulation was used to emulate an admittance model, without the presence of an added mass on the force sensor. The simulated motion was that of an unforced damped harmonic oscillator. The mass, spring, and damping parameters are shown in Table 4.1. The generalized coordinates are: $\{x, y, z, \alpha, \beta, \gamma\}^T$.

Figure 4.8 shows the commanded positions and orientations, and the results from the VES simulation. The differences between the commanded and simulated platform positions are shown in fig 4.9a. The corresponding results from the MIT hardware are shown in fig 4.9b. The orientation errors are shown in figures 4.10a and 4.10b. With the exception of the rotation about the y-axis (beta rotation), the simulation errors are up to two times larger than the hardware errors. The phase lags associated with the MIT results

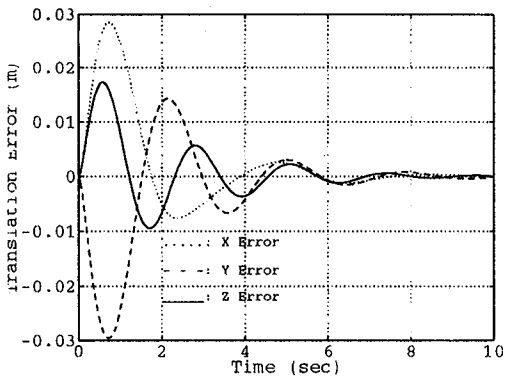
Mass Parameters	Spring Constants	Damping Constants
m_1 : 500 kg	K_{11} : 1500 N/m	C_{11} : 500 N/m/s
m_2 : 500 kg	K_{22} : 2500 N/m	C_{22} : 500 N/m/s
m_3 : 500 kg	K_{33} : 4000 N/m	C_{33} : 500 N/m/s
I_{44} : 250 kg m^2	K_{44} : 1500 N m/rad	C_{44} : 250 N m/rad/s
I_{55} : 250 kg m^2	K_{55} : 1000 N m/rad	C_{55} : 250 N m/rad/s
I_{66} : 250 kg m^2	K_{66} : 2000 N m/rad	C_{66} : 250 N m/rad/s

Table 4.1 Admittance Model Parameters[7]

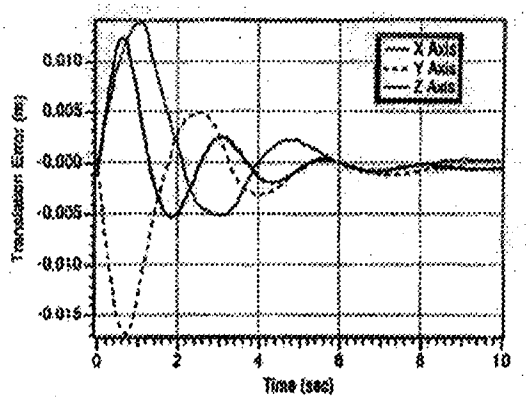


(a) Platform Position (b) Platform Orientation

Figure 4.8 Commanded and Simulated Platform Position and Orientation for Admittance Trajectories, Using a Decoupled Mass-Spring-Damper, (a) Position, (b) Orientation

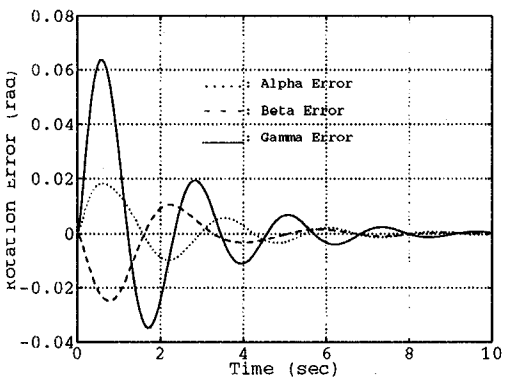


(a) VES Computer Simulation

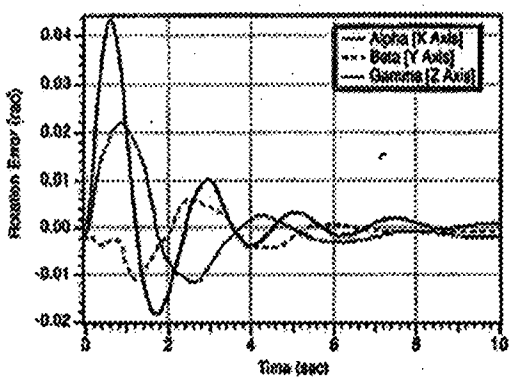


(b) MIT VES Hardware[7]

Figure 4.9 Platform Position Errors from Admittance Control of VES: (a) VES Computer Simulation (b) MIT VES Hardware[7]



(a) VES Computer Simulation



(B) MIT VES Hardware[7]

Figure 4.10 Platform Orientation Errors from Admittance Control of VES: (a) VES Computer Simulation (b) MIT VES Hardware[7]

are also larger. The sources of the inconsistencies between the software simulations and the hardware experiments cannot be determined without data from each actuator for the duration of the experiment, which was not available. The simulation results can be

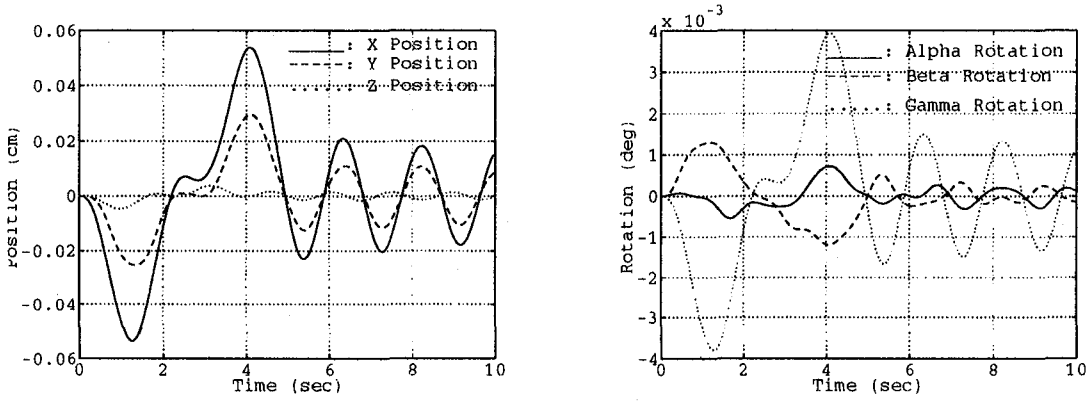
improved once data is available from the NASA VES. These results show that the software simulation is capable of modeling the hardware for admittance control.

4.3 Comparison of the Two Computer Models With the PUMA 560 and DOSS Mounted To the SRMS End Effector

The previous section determined that the VES software simulation is able to model the VES hardware. This section presents a comparison of the VES simulation with the SRMS-robot computer model developed in chapter 3.

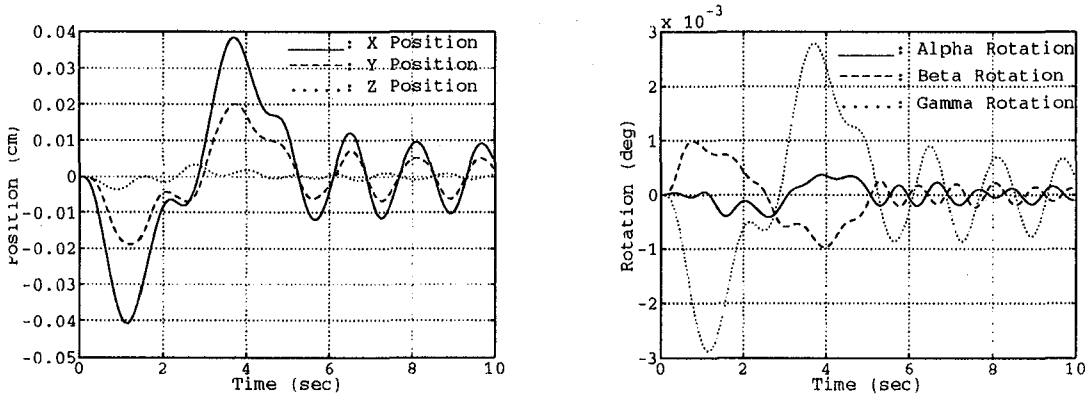
Both computer simulations were executed with the PUMA 560 mounted on the SRMS end effector. The SRMS was modeled with a damping value, $\zeta = 0.02$. The first joint of the PUMA 560 was moved 30^0 in five seconds, while the remaining five joints were stationary. This motion involved moving the robot's most massive joint in a short period of time, and was intended to create a considerable wrench on the SRMS end effector.

The motions resulting from the simulations are shown in figures 4.11-4.18. Comparison of the figures shows that the VES simulation and the SRMS-robot simulation produced nearly identical results for all SRMS configurations. The increased damping associated with the Simulink results is due to the added filters (refer to fig 2.15). The close comparison between the models illustrates two points. First, it appears to be valid to assume in the SRMS-robot model that the wrench due to the robot could be split into two components, one due to the articulating robot on a stationary base, and another due to the compliant base under a stationary robot. This assumption appears to be valid. Second,



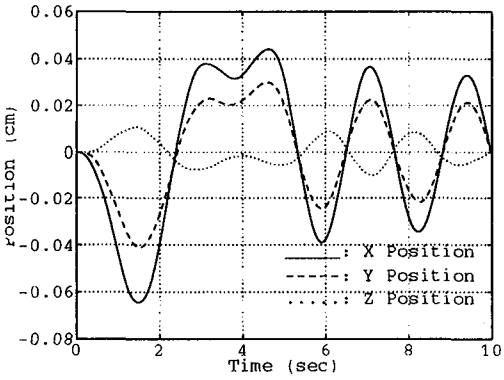
(a) Position (b) Orientation

Figure 4.11 SRMS-Robot Simulation: SRMS End Effector Motion for SRMS Unberth Configuration, with PUMA 560, (a) Position, (b) Orientation

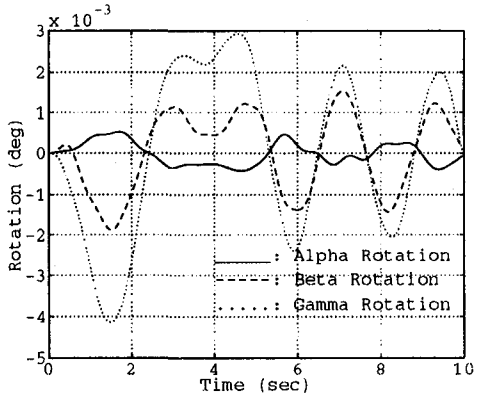


(a) Position (b) Orientation

Figure 4.12 VES Simulation: Commanded Platform Motion for SRMS Unberth Configuration, with PUMA 560, (a) Position, (b) Orientation

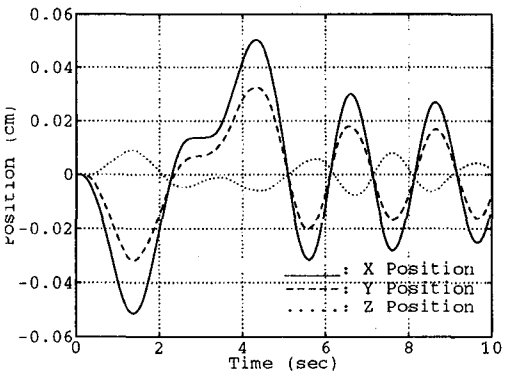


(a) Position

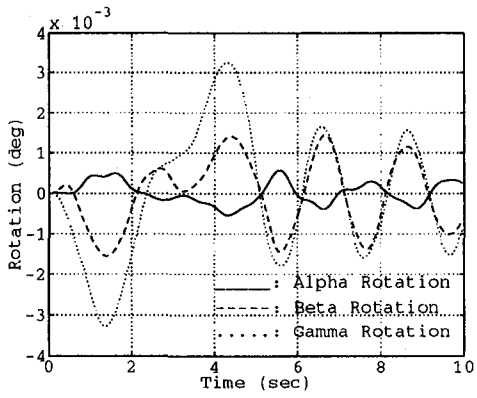


(b) Orientation

Figure 4.13 SRMS-Robot Simulation: SRMS End Effector Motion for SRMS Low Hover Configuration, with PUMA 560, (a) Position, (b) Orientation

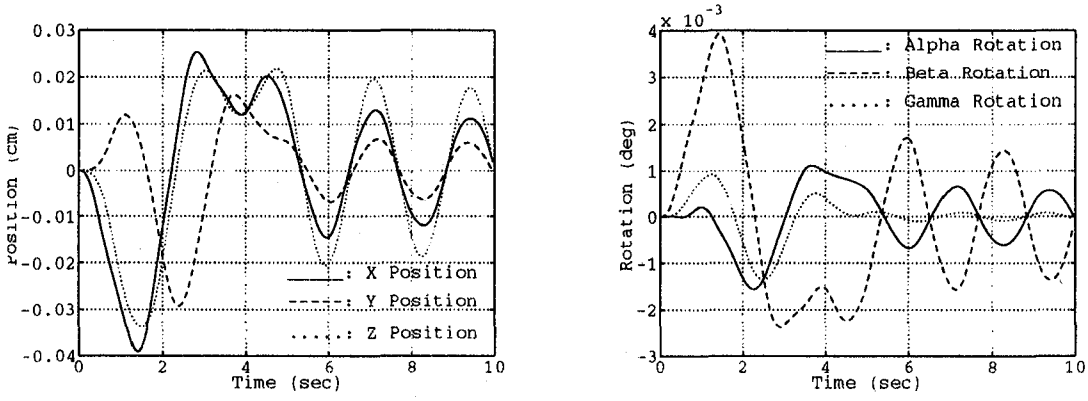


(a) Position



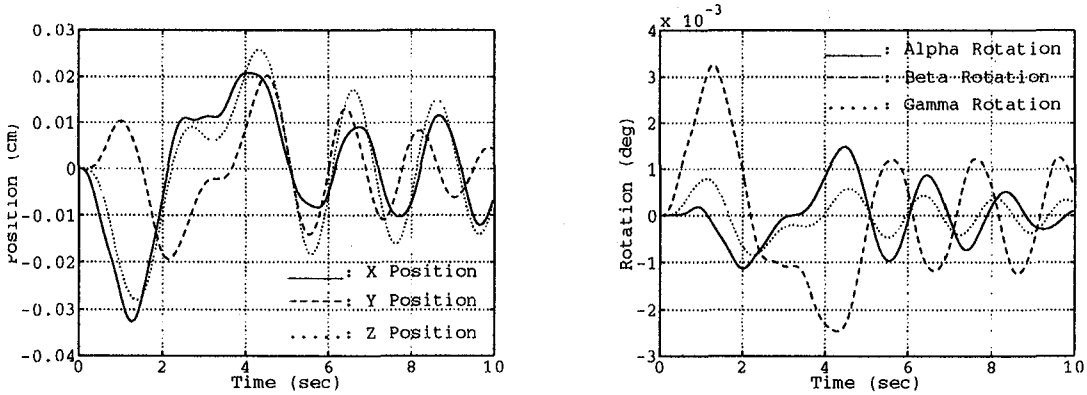
(b) Orientation

Figure 4.14 VES Simulation: Commanded Platform Motion for SRMS Low Hover Configuration, with PUMA 560, (a) Position, (b) Orientation



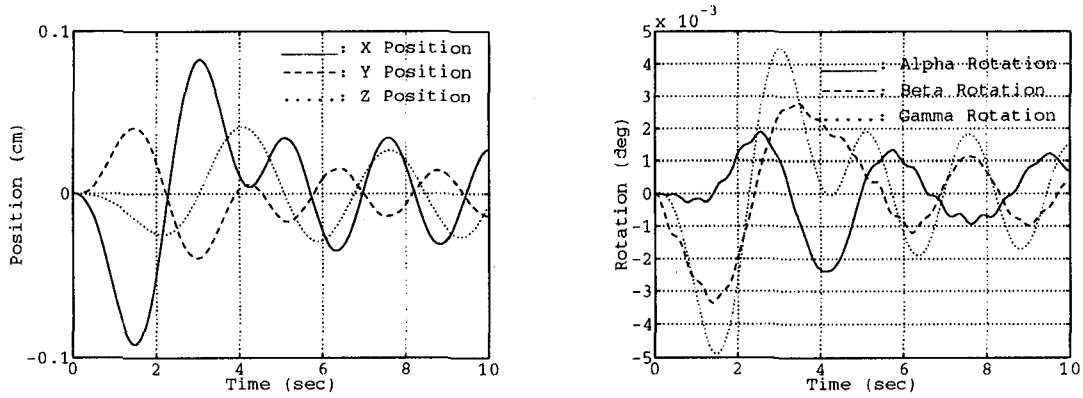
(a) Position (b) Orientation

Figure 4.15 SRMS-Robot Simulation: SRMS End Effector Motion for SRMS Deploy Configuration, with PUMA 560, (a) Position, (b) Orientation



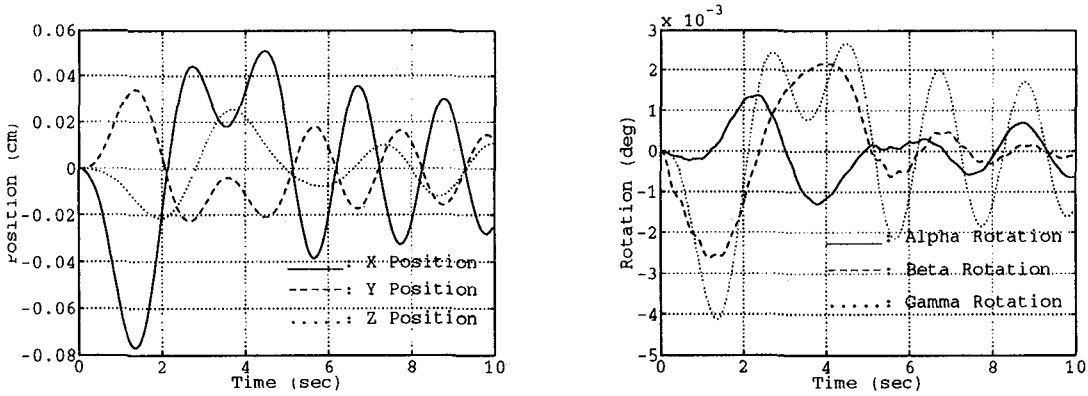
(a) Position (b) Orientation

Figure 4.16 VES Simulation: Commanded Platform Motion for SRMS Deploy Configuration, with PUMA 560, (a) Position, (b) Orientation



(a) Position (b) Orientation

Figure 4.17 SRMS-Robot Simulation: SRMS End Effector Motion for SRMS Capture Configuration, with PUMA 560, (a) Position, (b) Orientation

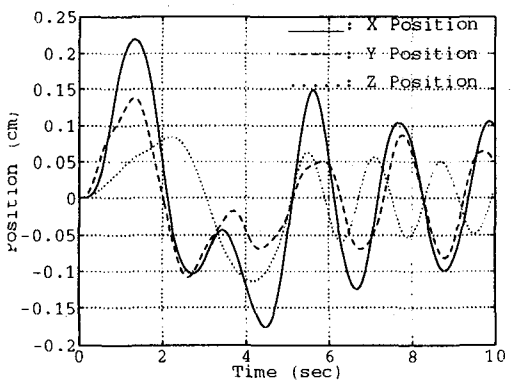


(a) Position (b) Orientation

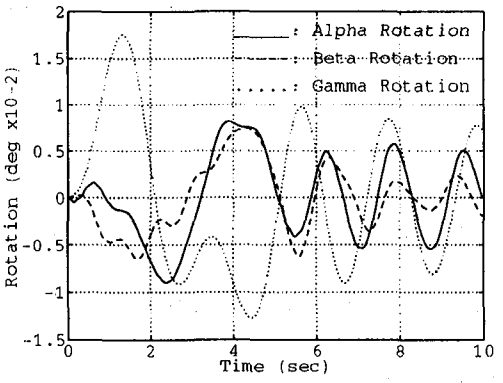
Figure 4.18 VES Simulation: Commanded Platform Motion for SRMS Capture Configuration, with PUMA 560, (a) Position, (b) Orientation

the SRMS-robot model can be used as a comparison to the actual VES motion, and can predict anticipated motions before they are implemented in the hardware.

Both computer simulations were executed with the shoulder pitch of the DOSS moving 30^0 in five seconds. Figures 4.19-4.26 compare the SRMS end effector motions for the two simulations. The DOSS is a proposed flight arm, and these results show that the anticipated motion of the SRMS end effector is on the order of centimeters for this maneuver. Different DOSS motions could produce larger SRMS disturbances, as could a payload attached to the DOSS.

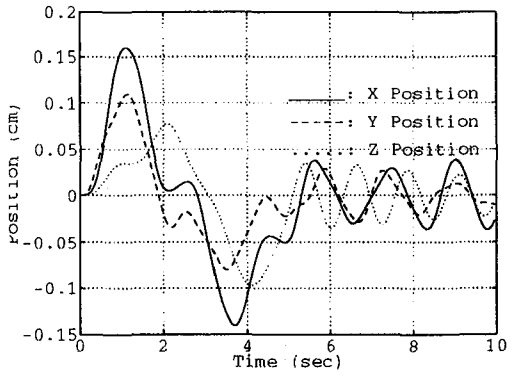


(a) Position

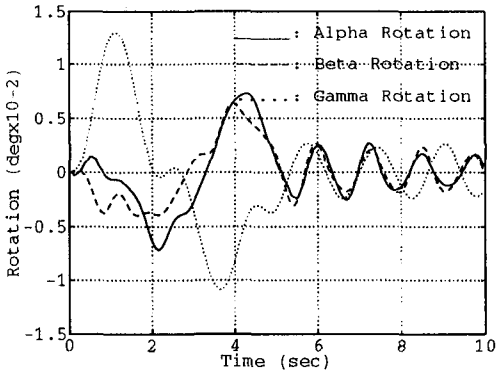


(b) Orientation

Figure 4.19 SRMS-Robot Simulation: SRMS End Effector Motion for SRMS Unberth Configuration, with DOSS, (a) Position, (b) Orientation

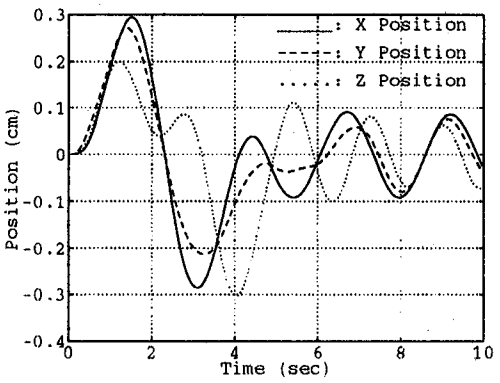


(a) Position

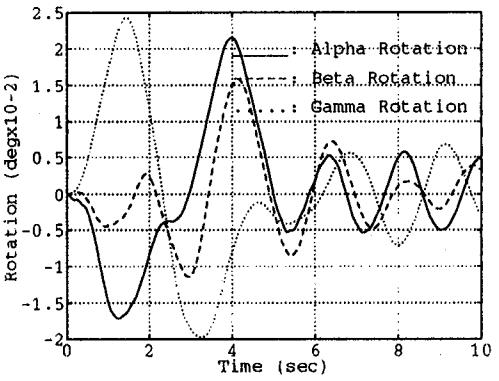


(b) Orientation

Figure 4.20 VES Simulation: Commanded Platform Motion for SRMS Unberth Configuration, with DOSS, (a) Position, (b) Orientation

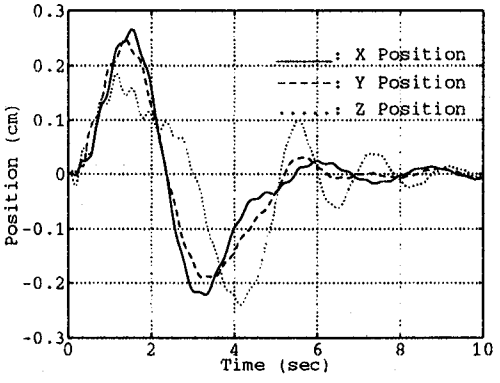


(a) Position

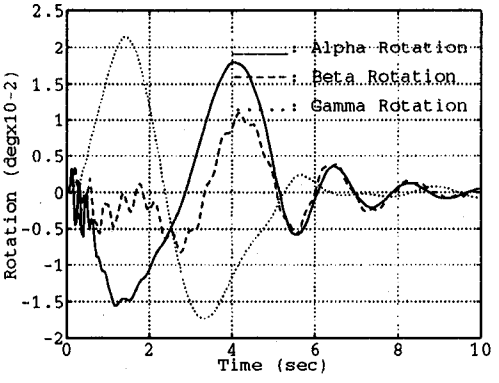


(b) Orientation

Figure 4.21 SRMS-Robot Simulation: SRMS End Effector Motion for SRMS Low Hover Configuration, with DOSS, (a) Position, (b) Orientation

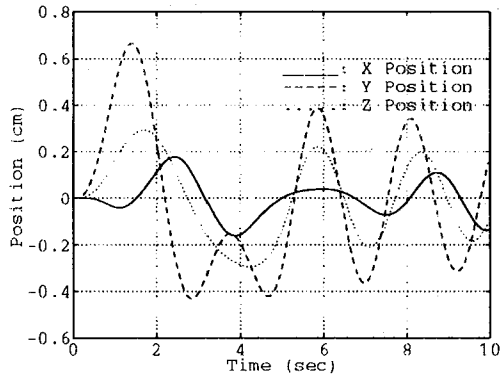


(a) Position

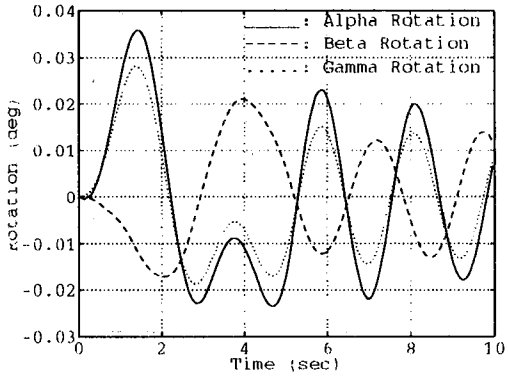


(b) Orientation

Figure 4.22 VES Simulation: Commanded Platform Motion for SRMS Low Hover Configuration, with DOSS, (a) Position, (b) Orientation

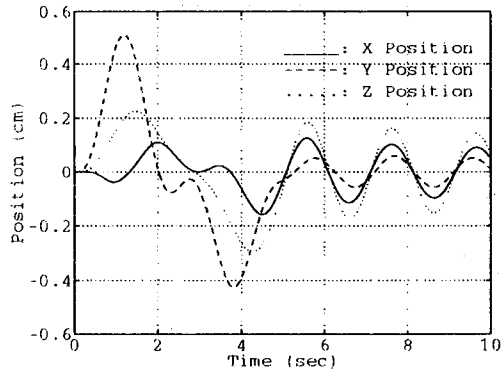


(a) Position

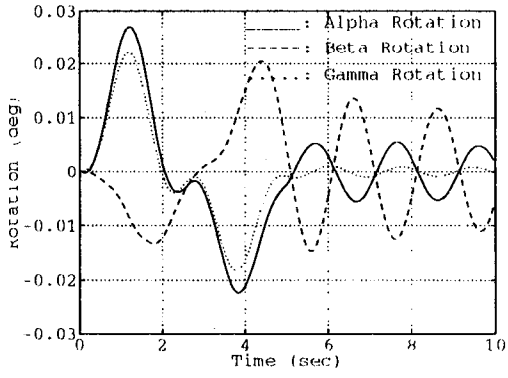


(b) Orientation

Figure 4.23 SRMS-Robot Simulation: SRMS End Effector Motion for SRMS Deploy Configuration, with DOSS, (a) Position, (b) Orientation

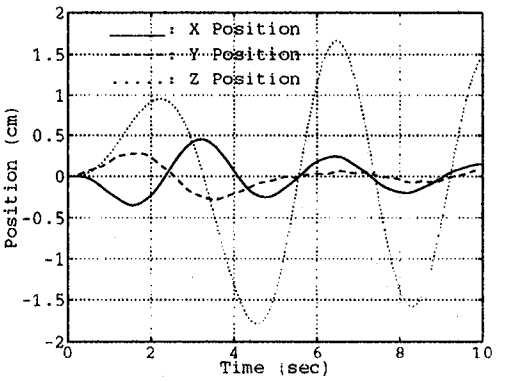


(a) Position

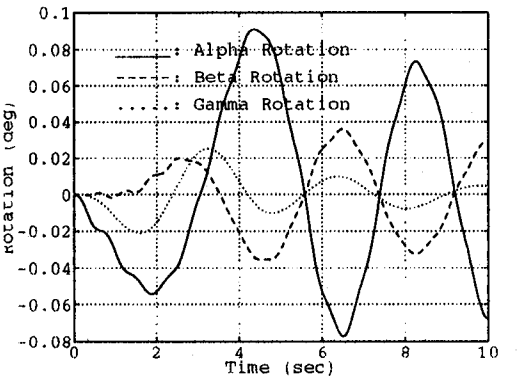


(b) Orientation

Figure 4.24 VES Simulation: Commanded Platform Motion for SRMS Deploy Configuration, with DOSS, (a) Position, (b) Orientation

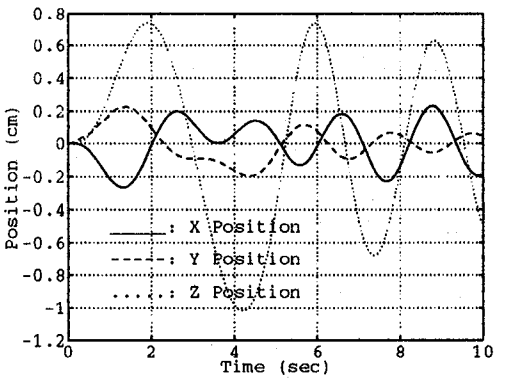


(a) Position

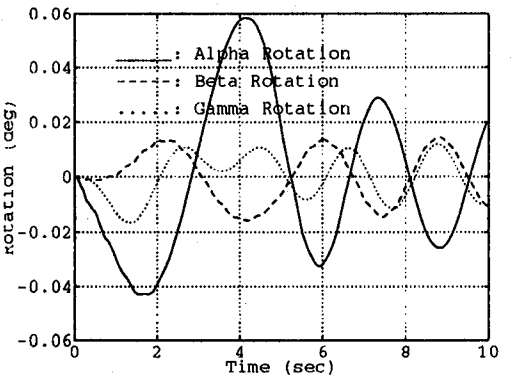


(b) Orientation

Figure 4.25 SRMS-Robot Simulation: SRMS End Effector Motion for SRMS Capture Configuration, with DOSS, (a) Position, (b) Orientation



(a) Position



(b) Orientation

Figure 4.26 VES Simulation: Commanded Platform Motion for SRMS Capture Configuration, with DOSS, (a) Position, (b) Orientation

5. Conclusions

The Vehicle Emulation System is used to simulate the interaction of an active robotic arm and a mobile base. The force-torque sensor is the interface between the two dynamical systems. The operation of the VES hardware and software was described. A computer model of each component, and their interaction, was presented. The computer simulation was developed using the Simulink software package from the Mathworks. The simulation included models of the robot, force-torque sensor, VES control software, actuator dynamics and control, and VES forward kinematics.

A computer simulation of the dynamic interaction of an active robotic arm on a mobile base was presented. The mass matrix of the robot was coupled to the mass matrix of the mobile base, defined in mode space. The input to the modified admittance model is the wrench due only to the robot's motion.

This VES simulation output was compared with experimental results from the MIT VES, and shown to be an accurate representation of the hardware for both trajectory and admittance control. The model was also compared with the computer simulation of the robot-mobile base interaction. This comparison verified the accuracy with which the VES simulation modeled the dynamics. It also showed that splitting the wrench into two components, one due to the articulating robot on a stationary base, and another due to the compliant base under a stationary robot, in the robot-mobile base simulation, was a valid assumption.

Simulations were executed such that the most massive joint of each robot was commanded to move a large distance in a short amount of time. Results from the two

simulations indicate that the SRMS end effector is expected to be displaced by no more than two centimeters for the worst case motion. However, different robots or the same robots with large payloads could cause the SRMS to experience increased disturbances. Future studies can involve investigating the effects different robots have on the SRMS-robot computer simulation developed in chapter 3, to determine the reaction of the SRMS end effector.

References

1. Alexander, H. L., Cannon, R. H. Jr, "An Extended Operational-Space Control Algorithm for Satellite Manipulators", Journal of the Astronautical Sciences, Vol. 38, October-December, 1990, pp. 473-485.

2. deSilva, C. W., "Trajectory Design for Robotic Manipulators in Space Applications", Journal of Guidance Control and Dynamics, Vol. 14, May-June 1991, pp. 670-674.

3. Longman, R. W., "Attitude Tumbling Due to Flexibility in Satellite-Mounted Robots", The Journal of the Astronautical Sciences, Vol. 38, No. 4, October-December 1990, pp. 487-509.

4. Baker, J. E., "Gravity Compensation for Experiments in Space Robotics", Master of Science Thesis, Department of Mechanical Engineering, M.I.T., Cambridge, MA, September, 1992.

5. Idris, H. R., "Real-Time Control of a Six-Degree-of-Freedom Vehicle Emulation System", Master of Science Thesis, Department of Mechanical Engineering, M.I.T., Cambridge, MA, April, 1992.

6. Muller, U. B., "Design of a Second Generation Experimental System for Studying Mobile Manipulator Control", Master of Science Thesis, Department of Mechanical Engineering, M.I.T., Cambridge, MA, June, 1992.

7. Kuklinski, A. E., "The Practical Implementation of an Experimental Tool for the Dynamic Interactions in Mobile Manipulator Systems", Master of Science Thesis, Department of Mechanical Engineering, M.I.T., Cambridge, MA, June, 1993.

8. West, H., Papadopoulos, E., Dubowsky, S., Cheah, H., "A Method for Estimating the Mass Properties of a Manipulator By Measuring the Reaction Moments at Its Base", Proceedings of the 1989 IEEE International Conference on Robotics and Automation, Vol. 3, Scottsdale, AZ, May 14-19, 1989, pp. 1510-1516.

9. Williams, R. L., "Kinematics of an In-Parallel Actuated Manipulator Based on the Stewart Platform Mechanism", NASA Technical Memorandum 107585, NASA Langley Research Center, March 1992.

10. Hootsmans, N. A. M., Dubowsky, S., "Large Motion Control of Mobile Manipulators Including Vehicle Suspension Characteristics", Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, April 1991, pp. 2336-2341.

11. Papadopoulos, L., Tolson, R. H., "Admittance Model for the Shuttle Remote Manipulator System in Four Configurations", NASA Contractor Report 4555, NASA Langley Research Center, November 1993.

12. Mabie, H. H., Reinholtz, C. F., Mechanisms and Dynamics of Machinery, John Wiley & Sons, New York, 1987.

13. "SIMULINK: A Program for Simulating Dynamic Systems", User's Guide, The Mathworks, Inc., Natick, MA, March 1992.

14. Armstrong, B., Oussama, K., Burdick, J., "The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm", Proceedings of the IEEE 1986 International Conference on Robotics and Automation, Vol. 1, San Francisco, CA, April 7-10, 1986, pp. 510-518.

15. Carignan, F. J., Cook, N., H., "Force Measuring Platform and Load Cell Therefor Using Strain Gages to Measure Shear Forces", United States Patent Number 4493220, January 15, 1985.

16. Craig, J. J., Introduction to Robotics, Mechanisms and Control, Second Edition, Addison-Wesley Publishing Company, Reading, MA, 1989.

17. Luh, J. Y. S., Walker, M. W., Paul, R. P. C., "On-Line Computational Scheme for Mechanical Manipulators", Journal of Dynamic Systems, Measurement, and Control, Vol. 102, June 1980, pp. 69-76.

18. D'Azzo, John, Linear Control System Analysis and Design, 3rd edition, McGraw-Hill Book Company, New York, 1988.

Appendices

A. Newton-Euler Equations

This Appendix presents the iterative Newton-Euler dynamics algorithm, as outlined by Craig [16]. In this thesis, the equations are used to determine the forces and moments generated at the base of a serial robot, given the joint angles, rates, and accelerations of the robot. The approach is based on the work done previously by Luh, Walker, and Paul on the topic of computational schemes for manipulators [17]. The following equations are from [16] and illustrate the outward iteration of velocities and accelerations, and the inward iteration of joint forces and moments.

Outward iterations: $i : 0 \rightarrow [n-1]$

$$
\begin{aligned}
{}^{i+1}\omega_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1},\\
{}^{i+1}\dot{\omega}_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\dot{\omega}_i + {}^{i+1}_{i}R\,{}^{i}\omega_i \times \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1},\\
{}^{i+1}\dot{v}_{i+1} &= {}^{i+1}_{i}R\left({}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times \left({}^{i}\omega_i \times {}^{i}P_{i+1}\right) + {}^{i}\dot{v}_i\right),\\
{}^{i+1}\dot{v}_{Ci+1} &= {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{Ci+1} + {}^{i+1}\omega_{i+1} \times \left({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{Ci+1}\right) + {}^{i+1}\dot{v}_{i+1},\\
{}^{i+1}F_{i+1} &= m_{i+1}\,{}^{i+1}\dot{v}_{Ci+1},\\
{}^{i+1}N_{i+1} &= {}^{Ci+1}I_{i+1}\,{}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{Ci+1}I_{i+1}\,{}^{i+1}\omega_{i+1}.
\end{aligned}
$$

Inward iterations: $i : [n] \rightarrow 1$

$$
\begin{aligned}
{}^{i}f_i &= {}^{i}_{i+1}R\,{}^{i+1}f_{i+1} + {}^{i}F_i,\\
{}^{i}n_i &= {}^{i}N_i + {}^{i}_{i+1}R\,{}^{i+1}n_{i+1} + {}^{i}P_{Ci} \times {}^{i}F_i + {}^{i}P_{i+1} \times {}^{i}_{i+1}R\,{}^{i+1}f_{i+1}.
\end{aligned}
\qquad \text{[C1]}
$$

where

${}^{i}\omega_i$ is the angular velocity of link i in its own frame,

${}^{i}v_i$ is the linear velocity of link i in its own frame,

${}^{i}v_{C_i}$ is the linear velocity at the center of mass of link i,

m_i is the mass of link i,

${}^{C_i}I_i$ is the inertia of the i^{th} link at its center of mass, in its own coordinate frame,

$\dot{\theta}_i$ and $\ddot{\theta}_i$ are the joint rate and acceleration of the i^{th} joint,

${}^{i}\hat{Z}_i$ is the unit vector in the z-direction for link i,

${}^{i}P_{i+1}$ is the position vector from link i to link (i+1),

${}^{i}P_{C_i}$ is the position vector from link i to its center of mass,

${}^{i}_{i+1}R$ is the rotation matrix from link (i+1) to link i,

${}^{i}F_i$ and ${}^{i}N_i$ are the forces and moments at the center of mass of link i, and

${}^{i}f_i$ and ${}^{i}n_i$ are the forces and moments at link i.

B. Gravity Compensation

Earth-based experimentalists who study space-based systems must take into account the effects of gravity. Submersing the hardware in a neutral buoyancy tank is one possible solution, but this method complicates the dynamics. Air bearing tables are another solution, but they are capable of only two-dimensional motions. The VES is able to perform three-dimensional motions, without complicating the dynamics.

Kuklinski presents an iterative technique for determining the effects of gravity, using the VES [7]. First, the manipulator is moved sufficiently slowly that dynamic effects are negligible, along a pre-planned trajectory, with the platform held stationary. At each time step in the control software, the wrench is read from the force sensor. Then the platform is run under admittance control, as the robot is commanded to follow the same path. The platform position data at each time step is combined with the position output of the admittance model, in a weighted average. These position values are used to command the platform during the next iteration. This process is repeated until the platform position converges to the desired degree of accuracy. This method is limiting because the robot is not allowed to follow a general trajectory to perform a task, as commanded by a hand controller or vision control system. In addition, every experiment must be executed numerous times in order to cancel the gravity effects.

The NASA-Langley VES will employ a different method that can be performed on-line, without iteration. This method depends on the mass parameters of the robot, along with its joint angles at each time step.

Each link of a manipulator has associated with it a mass and a vector defining the location of the center of mass, with regard to a chosen coordinate system. Therefore, an n-DOF robot has 4(n+1) mass parameters, including the fixed base link (link 0). West, et al [8] have developed a method for estimating combinations of the mass parameters of a manipulator that is mounted on a movable force-torque sensor. A modified version is presented here.

A homogeneous transform matrix is used to map one coordinate frame into another [16]. Referring to fig B.1, the location of point Q can be described with reference to frame {A} as

$$^{A}(^{A}P_{Q})=^{A}(^{A}P_{B})+^{A}_{B}R^{B}(^{B}P_{Q}), \tag{B.1}$$

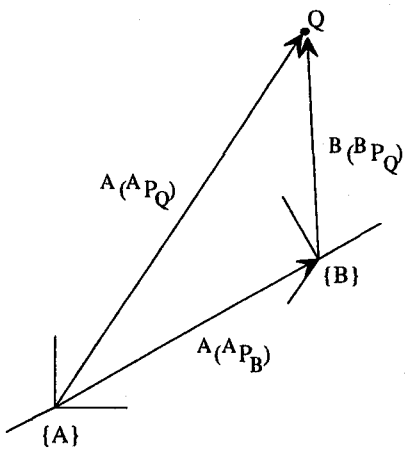


Figure B.1 Vector Diagram to a Point in Space

where ${}^{l}({}^{j}P_k)$ is the vector from j to k, expressed in l coordinates. ${}^{j}_{k}R$ is the 3x3 rotation matrix from frame k to j. A 4x4 transform matrix is used to express the right hand side of eq (B.1) as a single term with the following structure

$$\begin{Bmatrix} {}^{A}({}^{A}P_Q) \\ 1 \end{Bmatrix} = \begin{bmatrix} {}^{A}_{B}R & {}^{A}({}^{A}P_B) \\ \{000\} & 1 \end{bmatrix} \begin{Bmatrix} {}^{B}({}^{B}P_Q) \\ 1 \end{Bmatrix}, \tag{B.2}$$

or

$${}^{A}({}^{A}P_Q) = {}^{A}_{B}T\,{}^{B}({}^{B}P_Q). \tag{B.3}$$

When a stationary object of mass m is placed on an arbitrarily oriented six-DOF force-torque sensor, the following measurements are taken

$$\begin{aligned} F_x &= mg_x \\ F_y &= mg_y \\ F_z &= mg_z \\ M_x &= r_y F_z - r_z F_y \\ M_y &= r_z F_x - r_x F_z \\ M_z &= r_x F_y - r_y F_x \end{aligned} \tag{B.4}$$

where the gravity vector has been rotated according to the following X-Y-Z fixed-axis rotation

$$\left[{}^{I}_{P}R\right] = \begin{bmatrix} \cos\gamma\cos\beta & \cos\gamma\sin\beta\sin\alpha - \sin\gamma\cos\alpha & \cos\gamma\sin\beta\cos\alpha + \sin\gamma\sin\alpha \\ \sin\gamma\cos\beta & \sin\gamma\sin\beta\sin\alpha - \cos\gamma\cos\alpha & \sin\gamma\sin\beta\cos\alpha - \cos\gamma\sin\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix}. \tag{B.5}$$

The angles α, β, and γ are rotations about the inertial X, Y, and Z axes, respectively [18]. The center of mass is located at $\{r_x, r_y, r_z\}^T$ with reference to the force sensor. Note that the moment equations are linearly dependent, since

$$r_i M_i + r_j M_j = -r_k M_k, \tag{B.6}$$

for i , j, and k not equal. For an n-DOF robot, the moment equations at the base due to link i are simplified to the following form using homogeneous transform matrices

$$\begin{Bmatrix} M_x \\ M_y \\ M_z \end{Bmatrix}_i = (m_i x_i g)\begin{Bmatrix} R(3,3){}^B_iT(2,1) - R(2,3){}^B_iT(3,1) \\ R(1,3){}^B_iT(3,1) - R(3,3){}^B_iT(1,1) \\ R(2,3){}^B_iT(1,1) - R(1,3){}^B_iT(2,1) \end{Bmatrix} + (m_i y_i g)\begin{Bmatrix} R(3,3){}^B_iT(2,2) - R(2,3){}^B_iT(3,2) \\ R(1,3){}^B_iT(3,2) - R(3,3){}^B_iT(1,2) \\ R(2,3){}^B_iT(1,2) - R(1,3){}^B_iT(2,2) \end{Bmatrix}$$
$$+(m_i z_i g)\begin{Bmatrix} R(3,3){}^B_iT(2,3) - R(2,3){}^B_iT(3,3) \\ R(1,3){}^B_iT(3,3) - R(3,3){}^B_iT(1,3) \\ R(2,3){}^B_iT(1,3) - R(1,3){}^B_iT(2,3) \end{Bmatrix} + (m_i g)\begin{Bmatrix} R(3,3){}^B_iT(2,4) - R(2,3){}^B_iT(3,4) \\ R(1,3){}^B_iT(3,4) - R(3,3){}^B_iT(1,4) \\ R(2,3){}^B_iT(1,4) - R(1,3){}^B_iT(2,4) \end{Bmatrix},$$

$$i = 1,2,\ldots,n \qquad \text{(B.7)}$$

The total wrench at the base of the robot is the sum of the wrench components due to each link. Equation (B.7) illustrates that for each orientation of the robot and base, the unknown mass parameters can be separated from the known transform matrix values. Since only two of the moments are linearly independent, by placing the robot and base in $4(n+1)/2$ different orientations, all of the parameters can be determined by inverting the following equation

$$\begin{Bmatrix} \begin{Bmatrix} M_x \\ M_y \end{Bmatrix}_1 \\ \begin{Bmatrix} \vdots \\ \vdots \end{Bmatrix} \\ \begin{Bmatrix} M_x \\ M_y \end{Bmatrix}_{2(n+1)} \end{Bmatrix} = g\begin{bmatrix} \cdots & \cdots & \cdots & \cdots \\ \vdots & \ddots & \iddots & \vdots \\ \vdots & \iddots & \ddots & \vdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}\begin{Bmatrix} \begin{Bmatrix} m_0 x_0 \\ m_0 y_0 \\ m_0 z_0 \\ m_0 \end{Bmatrix} \\ \begin{Bmatrix} \vdots \\ \vdots \\ \vdots \\ \vdots \end{Bmatrix} \\ \begin{Bmatrix} m_n x_n \\ m_n y_n \\ m_n z_n \\ m_n \end{Bmatrix} \end{Bmatrix}. \qquad \text{(B.8)}$$

In equation (B.8), M_x and M_y have arbitrarily been chosen as the moments to be measured. The elements of the [4(n+1)]x[4(n+1)] matrix are evaluated from eqs (B.7).

The drawback to this method is that, for certain links of the robot, some of the elements in eqs (B.7) are linearly dependent. For example, the center of mass of link (i+1) cannot be translated parallel to its axis of rotation, with regard to the link i coordinate frame. It is not possible to distinguish between this product of mass and distance, and the link i quantity that also lies in this direction. Only their sum can be determined, resulting in k quantities that can be estimated (k<4(n+1)). This results in a reduced, kxk, form of equation (B.8), which is different for every robot.

The static model of a robot determines the forces and moments it creates due to the effects of gravity. This is also called the gravity compensation model. The dynamic model of a robot takes into account the motion of the robot. The previous section describes a method for estimating combinations of the mass parameters of a manipulator, through inversion of eq (B.8). If the parameters are assumed known, then equation (B.7) is evaluated to determine the moments at the base of the robot, due to each link. The forces at the base are evaluated from the top three equations from eqs (B.4). These are the equations the NASA VES will use to the cancel the effects of gravity.

C. Description of VES Control Software Subroutines

The Vehicle Emulation System employs control software to command the motion of the Stewart platform. The control software is comprised of seven main subroutines. They are:

- Force Measurement
- Gravity Compensation
- Admittance Model
- Trajectory Generation
- Inverse Kinematics
- Safety Checks
- Leg Communication

The following paragraphs briefly describe the algorithm associated with each subroutine.

- **Force Measurement**

The force-torque sensor measures the applied wrench due to an external forcing function. The analog signal is amplified, converted to digital units, and then made available to be read by the software.

The force measurement routine begins by sampling the first six channels of the A/D converter. (The seventh A/D channel is for pressure measurement, which is used in the safety checks software). Each channel contains a bias which is determined by running the calibration routine beforehand. These values are subtracted from the readings, resulting in a digital representation of the actual applied wrench.

Equation (4.6) illustrates the transformation the wrench undergoes as it is measured by the force sensor, amplified, and digitized by the A/D converter. In the software, the digital representation of the wrench is multiplied by the inverse of $[S_{proc}]$, $[C_{proc}]$, to retrieve the analog wrench:

$$\{W\} = [S_{proc}]^{-1}\{V_R\} = [C_{proc}]\{V_R\}, \qquad (C.1)$$

The crosstalk and sensitivity matrices provided by AMTI are in the SI units of measure, so the resulting wrench components are in Newtons and Newton-meters. The wrench is now expressed in the force sensor coordinate system, which is different from the platform frame by a 180° rotation about the x-axis of the platform, and a translation in all coordinates to the force sensor top plate. The code performs this transform and then rotates the wrench to the inertial frame.

- **Gravity Compensation**

The force-torque sensor reads the static, along with the dynamic, wrench created by the manipulator. The gravity compensation routine determines the static effects and subtracts them from the applied wrench.

West, et al [8] derive a procedure to estimate the mass parameters of a manipulator. This is accomplished this through a measurement of the static wrench the robot creates at its base. If these parameters are known, then this method can be inverted to compute the static wrench, assuming the manipulator's joint angles are known. The forces and moments are calculated using the relations $\vec{F} = m\vec{g}$ and $\vec{N} = \vec{r}X\vec{F}$, respectively, for each link. The NASA VES will employ this method of gravity compensation. Refer to Appendix B for a further discussion regarding the equations.

The robot is assumed to be a rigid body. First, the gravity vector is rotated from the inertial frame to the base frame, and then the force exerted by each link of the manipulator is computed. The forces are summed to represent the complete force. The moments are computed using the method presented in Appendix B. Next, the wrench is translated through the robot adapter plate to the force sensor frame, and then rotated and translated through the force sensor to the platform coordinate system. Finally, the wrench is rotated to the inertial frame, through the roll-pitch-yaw transformation, eq (C.1). The force measurement routine reads the entire wrench exerted on the force sensor, which includes the robot static and dynamic wrench, the mass of the force sensor top plate, and the robot adapter plate. The computed static effects due to the robot are added to the masses of the plates, and this sum is subtracted from the measured wrench. The dynamic wrench due to the robot remains, to be used in the admittance model.

- **Admittance Model**

An admittance model determines future states of a dynamic system, given the present input wrench and state. For a general second order system, an admittance model has the form

$$[M]\{\ddot{x}\}+[C]\{\dot{x}\}+[K]\{x\}=[B]\{W\}, \tag{C.2}$$

where $[M]$, $[C]$ and $[K]$ are the systems' mass, spring, and damping matrices, $\{W\}$ is the input wrench, and $[B]$ is the input matrix. Chapter 3 presents a method for transforming an admittance model from physical to mode space. The relationship between physical and mode space is

$$\{x\}=[\Phi]\{\eta\}, \tag{C.3}$$

where $\{\eta\}$ are the modal coordinates. Substituting eq (C.3) into (C.2) and using mass normalized modeshape matrices yields

$$[I]\{\ddot{\eta}\}+\begin{bmatrix}\ddots & & \\ & 2\zeta\omega_i & \\ & & \ddots\end{bmatrix}\{\dot{\eta}\}+\begin{bmatrix}\ddots & & \\ & \omega_i^2 & \\ & & \ddots\end{bmatrix}\{\eta\}=[\Phi^T]\{W\},\quad i=1,2,\ldots,n, \tag{C.4}$$

where ω_i are the natural frequencies of the modeled system, ζ is the damping coefficient, and $[B]$ is the identity matrix. Equation (C.4) is represented in state space as a series of 2n first order systems of the form

$$\begin{aligned}\dot{\eta}_1 &= \eta_2 \\ \dot{\eta}_2 &= -\omega_i^2\eta_1 - 2\zeta\omega_i\eta_2 + ([\Phi]^T\{W\})\Big|_{row(i)}\end{aligned}, \tag{C.5}$$

where η_1 and η_2 are the modal state and its time derivative, respectively. The admittance model is evaluated using a discrete form of the state space equations, so equations (C.5) are discretized to

$$\{\eta_{n+1}\}=[\phi]\{\eta_n\}+[\Gamma]\{u\}. \tag{C.6}$$

Appendix D contains a derivation of the $[\phi]$ and $[\Gamma]$ matrices for the decoupled equations given above. The conversion back to physical space is

$$\begin{aligned}\{x\} &= [\Phi]\{\eta\}, \\ \{\dot{x}\} &= [\Phi]\{\dot{\eta}\}, \\ \{\ddot{x}\} = [\Phi]\{\ddot{\eta}\} &= [\Phi][-2\zeta\omega_n]\{\dot{\eta}\} + [\Phi][-\omega_n{}^2]\{\eta\} + [\Phi][\Phi^T]\{W\}.\end{aligned} \tag{C.7}$$

- **Trajectory Generation**

While the admittance model software computes the motion of a dynamic system, the trajectory generation code commands the VES to follow a pre-planned path. The

system is capable of ramping (linear) and sinusoidal motion. The platform can be instructed to move in one Cartesian direction, or a combination of all directions.

For ramping motion, the user interface allows the user to enter the desired final Cartesian coordinates of the center of the platform top plate, and the trajectory time, t_{final}. The initial coordinates are known. The number of time steps, N, required for the motion is determined from

$$N = 100 t_{final},$$

where the code operates in real time at 100 Hz. Equations (C.8) are used to determine the amount by which to increment each coordinate at each time step. Equation (C.9) computes the new coordinates.

$$\Delta_i = \frac{(i_{final} - i_{initial})}{N}, \qquad i = x, y, z, \theta_x, \theta_y, \theta_z \tag{C.8}$$

$$i_{new} = i_{old} + \Delta_i \tag{C.9}$$

If sinusoidal motion is desired, the user is prompted to enter the amplitude, A_i, and frequency, f_i, of the motion, for each coordinate. Equation (C.10) is used to compute the motion at each time step, t.

$$i_{new} = i_{initial} + A_i \sin(2\pi f_i t), \qquad i = x, y, z, \theta_x, \theta_y, \theta_z \tag{C.10}$$

When the trajectory generation software routine is used, either equation (C.9) or (C.10) is used at each time step.

- **Inverse Kinematics**

For a general robotic manipulator, the inverse kinematics solution is used to determine the joint angles or actuator lengths, given the position and orientation of the end

effector, with respect to some coordinate system. The six ball joint locations at the base of the platform are known from the geometry of the VES, and the Euclidean norm is used to calculate the leg lengths:

$$\vec{L}_i = [{}^B_P T]\{{}^P({}^B P_{P_i})\} - \{{}^B({}^B P_{B_i})\} \qquad i = 1,2,...6, \tag{C.11}$$

$$|L_i| = \|\vec{L}_i\|_2. \tag{C.12}$$

In equation (C.11), ${}^P\{{}^B P_{Pi}\}$ is the Cartesian position of the i^{th} platform ball joint with respect to the base, in platform coordinates, and ${}^B\{{}^B P_{Bi}\}$ is the location of the i^{th} base ball joint with respect to the base, in base coordinates. $[{}^B_P T]$ is the transformation from the platform to the base frame.

The inverse kinematics software routine is as follows. First, the rotation matrix from the platform frame to the inertial frame is calculated. Then the platform joint locations are transformed to the base frame, and the actuator vector lengths are computed using equation (C.11). Finally, the leg lengths are calculated using eq (C.12).

- **Safety Checks**

Various safety requirements are checked at each time step, to ensure that the new leg lengths are within the workspace of the system, and that control over the platform has not been lost. Ref. [7] contains a detailed description of each safety software routine. They are briefly outlined below.

Velocity Check: Checks the difference between successive commanded leg lengths, to determine if they are within required limits.

Position Error Check: Computes the difference between the commanded length of each leg and the measured value, to determine if they are within required limits.

Collide: Determines if the new platform geometry will result in a collision between adjoining legs.

Motion Limits: Checks each commanded Cartesian coordinate to determine if it is within the workspace of the platform (±0.305 meters in translation, ±30° in rotation).

Actuator Limits: Determines if each leg length is within required limits.

Joint Limits: Checks to make sure that each joint angle has not been exceeded.

Flow Limits: Checks the flow of each actuator by calculating the difference between successive lengths of each leg, to determine if it is within required limits.

Panic Button and Floor Mat: A person standing on the floor mat surrounding the Stewart platform, or pressing the panic button, generates an interrupt to the control software, and the motion is stopped.

- **Leg Communication**

After the inverse kinematics are invoked to determine the new leg lengths, and the safety checks are performed, the leg communication routine computes the following leg command to send to each actuator:

$$LegCommand_i = \frac{|L_i| - LegOffset_i}{LegScale_i}, \qquad i = 1,2,\ldots,6, \tag{C.13}$$

where $LegOffset_i$ and $LegScale_i$ are constants determined by the geometry of the platform. The software then exchanges information with the legHost chip that commands the legs. The code outputs the commanded leg lengths, and reads back the measured lengths.

D. Discrete State Space - Closed Form Solution for a 2-DOF System

The Vehicle Emulation System uses discrete state space methods to integrate the admittance model of the base motion. This model is in modal space, and for a system utilizing n modes, the equations are decoupled into n second-order equations, or 2n first-order equations of the form

$$\begin{aligned} \dot{\eta}_1 &= \eta_2 \\ \dot{\eta}_2 &= -\omega_i^2\eta_1 - 2\zeta\omega_i\eta_2 + ([\Phi]^T\{W\})\big|_{row(i)} \end{aligned}, \tag{D.1}$$

where η are the modal coordinates, ω_i is the natural frequency of the i^{th} mode, ζ is the modal damping, Φ is the modeshapes matrix, and $\{W\}$ is the externally applied wrench. In state-space, the equations have the form

$$\{\dot{\eta}\} = [A]\{\eta\} + [B]\{u\}, \tag{D.2}$$

where

$$[A] = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix}. \tag{D.3}$$

The admittance model is evaluated using a ZOH, so equations (D.2) are discretized to

$$\begin{aligned} \eta(T) &= e^{AT}\eta(T-1) + \int_{T-1}^{T} e^{A(T-\tau)}Bu(\tau)d\tau \\ &= e^{AT}\eta(T-1) + \left[\int_{T-1}^{T} e^{A(T-\tau)}d\tau\right]Bu \\ &= e^{AT}\eta(T-1) - A^{-1}(I - e^{A\tau})Bu \end{aligned} \tag{D.4}$$

or

$$\{\eta_{n+1}\} = [\phi]\{\eta_n\} + [\Gamma]\{u\}. \tag{D.5}$$

Equations (D.4) present a discrete approximation to the admittance model. It is based on the previous state, and current input. The e^{AT} expression is evaluated using Laplace transforms, as follows:

$$\begin{aligned} e^{AT} = \phi &= \ell^{-1}(sI - A)\big|_{t=T} = \ell^{-1}\left\{\begin{bmatrix} s & -1 \\ \omega^2 & s+2\zeta\omega \end{bmatrix}^{-1}\right\} \\ &= \ell^{-1}\left\{\frac{1}{s^2+2\zeta\omega s+\omega^2}\begin{bmatrix} s+2\zeta\omega & 1 \\ -\omega^2 & s \end{bmatrix}\right\} \\ &= \begin{bmatrix} \phi_1+\phi_2 & \phi_3 \\ \phi_4 & \phi_1 \end{bmatrix} \end{aligned} \tag{D.6}$$

where

$$\begin{aligned} \phi_1 &= \ell^{-1}\left\{\frac{s}{s^2+2\zeta\omega s+\omega^2}\right\} = \frac{-1}{\sqrt{1-\zeta^2}}e^{-\zeta\omega T}\sin(\omega T\sqrt{1-\zeta^2} - \cos^{-1}(\zeta)) \\ \phi_2 &= \ell^{-1}\left\{\frac{2\zeta\omega}{s^2+2\zeta\omega s+\omega^2}\right\} = \frac{2\zeta}{\sqrt{1-\zeta^2}}e^{-\zeta\omega T}\sin(\omega T\sqrt{1-\zeta^2}) \\ \phi_3 &= \ell^{-1}\left\{\frac{1}{s^2+2\zeta\omega s+\omega^2}\right\} = \frac{1}{\omega\sqrt{1-\zeta^2}}e^{-\zeta\omega T}\sin(\omega T\sqrt{1-\zeta^2}) \\ \phi_4 &= \ell^{-1}\left\{\frac{-\omega^2}{s^2+2\zeta\omega s+\omega^2}\right\} = \frac{-\omega}{\sqrt{1-\zeta^2}}e^{-\zeta\omega T}\sin(\omega T\sqrt{1-\zeta^2}) \end{aligned} \tag{D.7}$$

and

$$[A]^{-1} = \begin{bmatrix} -2\zeta/\omega & -1/\omega \\ 1 & 0 \end{bmatrix}. \tag{D.8}$$

E. Admittance Control Stability Issues

This Appendix describes the concept of admittance control as it is implemented by the VES. It is shown that using a ZOH on the force-torque sensor's measurements could impose restrictions on the VES software admittance model.

Admittance control involves determining the current state of a dynamic system given its previous state and input forces and moments. If the admittance model is defined in Cartesian space, then the state is composed of the position and orientation. For systems with many degrees-of-freedom, the admittance model may be a truncated form of the equations of motion, defined at a particular point within the system. The VES operates with an admittance model. The VES allows the interaction of two dynamical systems: an active robotic arm, and the Stewart platform, which is emulating the motion of a general flexible mobile base. The force-torque sensor is the interface between the two systems. As this Appendix will illustrate, discretizing the measurements from the sensor may introduce limitations on the admittance model concept. For a robot directly attached to the actual mobile base being emulated, the motions of one mechanism directly affect the other mechanism, and the system is generally stable. Therefore, the limitations are a result of the experimental hardware.

Figure E.1 shows a one degree-of-freedom (DOF) mass-spring-damper system. The equation of motion is

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = 0, \tag{E.1}$$

which yields a stable system for mass, damper and spring values greater than zero. Addition of a second mass, m_1, rigidly attached to mass m, changes the equation to

$$(m+m_1)\ddot{x}(t)+c\dot{x}(t)+kx(t)=0. \tag{E.2}$$

This situation is depicted in fig E.2, and is also stable under the same conditions. Bringing the added mass to the right hand side of equation (E.2) yields

$$m\ddot{x}(t)+c\dot{x}(t)+kx(t)=-m_1\ddot{x}(t)=W, \tag{E.3}$$

where W is the wrench, and is a single force for the 1-DOF case. Finally, introduce a fictitious one degree-of-freedom force sensor/transmitter between the two masses (refer to figure E.3). This sensor/transmitter is massless and dimensionless, so that its presence does not physically alter the model. The sensor/transmitter is designed to measure the analog force, $-m_1\ddot{x}(t)$, that the added mass exerts on the original system, and then transmit this force through to the mass-spring-damper, which then reacts accordingly.

The situation described is the manner in which the VES operates. The reaction wrench, W, due to an articulating external mass on a flexible base is measured by the force-torque sensor. This wrench is digitized, though an analog-to-digital converter, and transmitted to the control software. Calculations are performed in the software to move the Stewart platform, using the desired admittance model of the flexible base.

Returning to the one dimensional illustration, it is observed that the presence of the force sensor/transmitter has changed the situation. If the sensor/transmitter is sampled at intervals, τ, then equation (E.3) is no longer valid. Discretizing the output of the sensor/transmitter introduces a zero-order-hold on the wrench before it is transmitted to the mass-spring-damper. The equation of motion, over the interval $t_n < t < t_{n+1}$, is thus

$$m\ddot{x}(t)+c\dot{x}(t)+kx(t)=-m_1\ddot{x}(t_n)=F(t_n)=constant, \tag{E.4}$$

where t_n is the last time at which the sensor/transmitter updated an output force value,

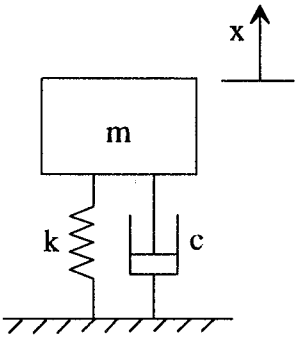


Figure E.1 Mass-Spring-Damper System

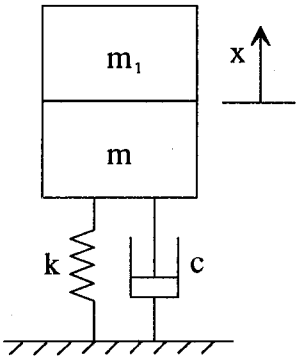


Figure E.2 Mass-Spring-Damper System with an Added Mass

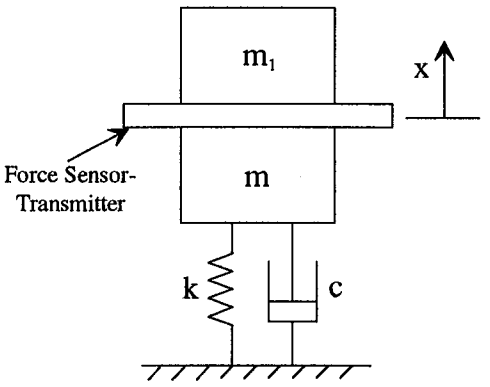


Figure E.3 Mass-Spring-Damper System with an Added Mass and a Force Sensor-Transmitter

the sensor's sampling period is $\tau = t_{n+1} - t_n$, and the feedback force is held constant over the time interval. The initial conditions are: $x(t_n^+) = x(t_n^-)$, and $\dot{x}(t_n^+) = \dot{x}(t_n^-)$. Dividing by m in eq E.3 yields

$$\ddot{x}(t) + 2\zeta\omega\dot{x}(t) + \omega^2 x(t) = r\ddot{x}(t_n) = F(t_n) = constant, \tag{E.5a}$$

or in state space

$$\begin{Bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix} + \begin{Bmatrix} 0 \\ r\ddot{x}(t_n) \end{Bmatrix} \tag{E.5b}$$

where $2\zeta\omega = c/m$, $\omega^2 = k/m$, $r = -m_1/m$, and $\{y_1, y_2\}^T = \{x, \dot{x}\}^T$.

At the beginning of the interval, $t = t_n$, the system has an initial position, $x(t_n)$, velocity, $\dot{x}(t_n)$, and a forcing function that is proportional to the acceleration, $\ddot{x}(t_n)$ so the closed-form solution for $x(t)$ over this interval is

$$\begin{aligned} x(t) = {} & x(t_n)e^{-\zeta\omega(t-t_n)}\left(\cos(\omega_d(t-t_n)) + \frac{\zeta\omega}{\omega_d}\sin(\omega_d(t-t_n))\right) \\ & + \frac{\dot{x}(t_n)}{\omega_d}e^{-\zeta\omega(t-t_n)}\sin(\omega_d(t-t_n)) \\ & + \frac{r\ddot{x}(t_n)}{\omega^2}\left(1 - e^{-\zeta\omega(t-t_n)}\left(\cos(\omega_d(t-t_n)) + \frac{\zeta\omega}{\omega_d}\sin(\omega_d(t-t_n))\right)\right) \end{aligned} \tag{E.6}$$

where $\omega_d = \omega\sqrt{1-\zeta^2}$ is the damped natural frequency. Time differentiation of eq (E.6) produces the velocity and acceleration over the time interval.

$$\begin{aligned} \dot{x}(t) = {} & -\frac{\omega^2}{\omega_d}x(t_n)e^{-\zeta\omega(t-t_n)}\sin(\omega_d(t-t_n)) \\ & + \frac{\dot{x}(t_n)}{\omega_d}e^{-\zeta\omega(t-t_n)}\left(\omega_d\cos(\omega_d(t-t_n)) - \zeta\omega\sin(\omega_d(t-t_n))\right) \\ & + \frac{r\ddot{x}(t_n)}{\omega_d}e^{-\zeta\omega(t-t_n)}\sin(\omega_d(t-t_n)) \end{aligned} \tag{E.7}$$

and

$$\ddot{x}(t) = \omega^2 x(t_n) e^{-\zeta\omega(t-t_n)}\left(\frac{\zeta\omega}{\omega_d}\sin(\omega_d(t-t_n)) - \cos(\omega_d(t-t_n))\right)$$
$$+\frac{\dot{x}(t_n)}{\omega_d} e^{-\zeta\omega(t-t_n)}\left(\omega^2(2\zeta^2-1)\sin(\omega_d(t-t_n)) - 2\zeta\omega\omega_d\cos(\omega_d(t-t_n))\right). \qquad \text{(E.8)}$$
$$+r\ddot{x}(t_n) e^{-\zeta\omega(t-t_n)}\left(\cos(\omega_d(t-t_n)) - \frac{\zeta\omega}{\omega_d}\sin(\omega_d(t-t_n))\right)$$

Equations (E.6)-(E.8) apply over the entire sampling period. Note that they are all functions of the position, velocity, and acceleration at the beginning of the time step. Evaluation of the equations at $t = t_{n+1}$, and writing them in matrix form yields

$$\begin{Bmatrix} x(n+1) \\ \dot{x}(n+1) \\ \ddot{x}(n+1) \end{Bmatrix} = \begin{bmatrix} \alpha e^{-\zeta\omega\tau} & \frac{1}{\omega_d} e^{-\zeta\omega\tau}\sin\omega_d\tau & \frac{r}{\omega^2}(1-\alpha e^{-\zeta\omega\tau}) \\ -\frac{\omega^2}{\omega_d} e^{-\zeta\omega\tau}\sin\omega_d\tau & \frac{\beta}{\omega_d} e^{-\zeta\omega\tau} & \frac{r}{\omega_d} e^{-\zeta\omega\tau}\sin\omega_d\tau \\ -\omega^2\alpha e^{-\zeta\omega\tau} & \frac{\gamma}{\omega_d} e^{-\zeta\omega\tau} & -\frac{r}{\omega_d}\beta e^{-\zeta\omega\tau} \end{bmatrix} \begin{Bmatrix} x(n) \\ \dot{x}(n) \\ \ddot{x}(n) \end{Bmatrix}, \qquad \text{(E.9)}$$

or

$$\{y(n+1)\} = [\phi]\{y(n)\},$$

where

$$\alpha = \cos\omega_d\tau + \frac{\zeta\omega}{\omega_d}\sin\omega_d\tau,$$

$$\beta = \omega_d\cos\omega_d\tau - \zeta\omega\sin\omega_d\tau,$$

and

$$\gamma = \omega^2(2\zeta^2-1)\sin\omega_d\tau - 2\zeta\omega\omega_d\cos\omega_d\tau.$$

Equation (E.9) is the discrete form of eq (E.5b), where the forcing function has been eliminated as a separate term, by including it in the state of the system. Equations (E.9)

are an exact representation of the continuous equations (eqs (E.6)-(E.8)) when they are evaluated at multiples of the sampling period.

When operating in discrete state space, the conditions on stability are that the roots of the characteristic equation of the system are all less than or equal to one. The roots are determined from

$$\det[\lambda I - \phi] = 0. \tag{E.10}$$

The characteristic equation for eq (E.9) is

$$\begin{aligned} &\lambda^3 + e^{-\zeta\omega\tau}\left(\frac{r\omega}{\omega_d}\zeta \sin\omega_d\tau - (2+r)\cos\omega_d\tau\right)\lambda^2 \\ &+\left\{e^{-2\zeta\omega\tau}(r+1) + e^{-\zeta\omega\tau}\left(-\frac{r\omega\zeta}{\omega_d}\sin\omega_d\tau + r\cos\omega_d\tau\right)\right\}\lambda - re^{-2\zeta\omega\tau} = 0 \end{aligned} \tag{E.11}$$

The exponential is also written as

$$e^{-\zeta\omega\tau} = e^{-2\pi\zeta F/f},$$

where $F = \omega/2\pi$, and $f = 1/\tau$ is the sampling frequency of the force sensor. If the added mass, m_1, is removed from the system, then $r = 0$, and the characteristic equation with no damping reduces to

$$\lambda(\lambda^2 - 2\lambda\cos\omega\tau + 1) = 0. \tag{E.12}$$

The roots of this equation are all less than or equal to one for all values of $\omega\tau$:

$$\begin{aligned} \lambda_1 &= 0 \\ \lambda_{2,3} &= \cos\omega\tau \pm i\sin\omega\tau \end{aligned}$$

where $i = \sqrt{-1}$.

On the other hand, neglect damping in eq (E.11), and let the sampling period of the sensor reduce to zero, with $r \neq 0$ (including m_1 again). Under these conditions, the characteristic equation is

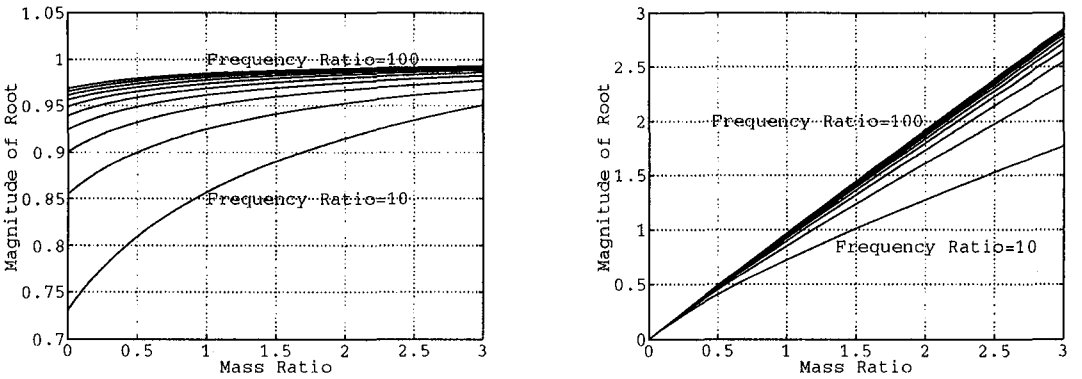
$$(\lambda - 1)^2(\lambda - r) = 0. \tag{E.13}$$

The two repeated roots are oscillatory in nature. The third root is stable if

$$|\lambda| = |r| = \left|-\frac{m_1}{m}\right| \leq 1. \tag{E.14}$$

This equation states that, in the limit as the sampling period of the sensor goes to zero, the largest allowable mass ratio is one. This is due to feeding back the acceleration into the admittance model. It is important to note that when the mass ratio is greater than one, the divergence of the system is dependent only on the number of time steps, and is independent of τ.

In general, the three roots of the characteristic equation are functions of the mass ratio, m_1/m, the frequency ratio, f/F, and the damping. Two of the roots are repeated, and are stable. Figure E.4a shows a plot of the repeated root as the mass and frequency ratios are varied, with a damping coefficient, ζ =0.5. This figure illustrates that as the sampling frequency of the sensor increases, the magnitude of the roots approach one. Figure E.4b shows a plot of the third root as the mass ratio and the ratio of frequencies are varied, with the same damping. In general, control systems should be sampled at least 15 times the largest frequency associated with the system, i.e., $f/F > 15$. The frequency ratio in the figures ranges from 10 to 100 in steps of 10, and as it increases, the magnitude of the third root exceeds one for decreasing mass ratios. The instability is a result of the



(a) Repeated Root (b) Third Root

Figure E.4 Magnitude of the Roots of the Characteristic Equation for Increasing Mass Ratio and Frequency Ratio, with $\zeta = 0.5$, (a) Repeated Root, (b) Third Root

ZOH on the feedback wrench. It is known that stability problems can arise in second order sampled systems [18].

This analysis can be generalized to a linearized N degree-of-freedom system, by writing eq (E.4) in matrix form:

$$[M]\{\ddot{X}(t)\}+[C]\{\dot{X}(t)\}+[K]\{X(t)\}=-[M_1]\{\ddot{X}(t_n)\}, \tag{E.15}$$

where

$[C]$ is the NxN damping matrix,

$[K]$ is the NxN stiffness matrix, and

$[M]$ is the NxN mass matrix.

Equation (E.15) is transformed to mode space by making the substitution

$$\{X(t)\}=[\Phi]\{\eta(t)\}, \tag{E.16}$$

where $\{\eta\}$ is the kx1 vector of the modal coordinates, and $[\Phi]$ is the 6xk modeshapes matrix. The attraction to working in mode space is that the mass, spring, and damping

matrices become decoupled. Substituting eq (E.16) into eq (E.15), and pre-multiplying by $[\Phi]^T$ produces

$$\Phi^T M\Phi\ddot{\eta}(t)+\Phi^T C\Phi\dot{\eta}(t)+\Phi^T K\Phi\eta(t)=-\Phi^T M_1\Phi\ddot{\eta}(t_n). \tag{E.17}$$

The matrix notation has been removed to avoid cluttering the equation. If mass-normalized modeshape matrices are used, then eq (E.17) becomes

$$I\ddot{\eta}(t)+\begin{bmatrix}\ddots & & \\ & 2\zeta\omega_i & \\ & & \ddots\end{bmatrix}\dot{\eta}(t)+\begin{bmatrix}\ddots & & \\ & \omega_i^2 & \\ & & \ddots\end{bmatrix}\eta(t)=-\Phi^T M_1\Phi\ddot{\eta}(t_n)=R\ddot{\eta}(t_n) \tag{E.18}$$

for i=1,2,...,k. This is the matrix form of eq (E.5a). Transforming (E.18) to discrete space yields a matrix similar to eq (E.9). Taking the limit of this matrix as the sampling period approaches zero, yields

$$\phi=\begin{bmatrix}[I] & [0] & [0]\\ [0] & [I] & [0]\\ [-\omega^2] & [-2\zeta\omega] & [R]\end{bmatrix}. \tag{E.19}$$

With no damping, 2N roots of the characteristic equation are one, and N roots are the eigenvalues of $[R]$. If any of the eigenvalues of $[R]$ are greater than one, unstable VES motion is produced. Pre-multiplying eq (E.15) by $[M]^{-1}$ shows that $[R]$ can also be expressed in physical space as

$$[R]=[M]^{-1}[M_1], \tag{E.20}$$

where the stability condition is

$$\left|eig\left(M^{-1}M_1\right)\right|\le 1.$$

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1995	Contractor Report

4. TITLE AND SUBTITLE
Simulating the Dynamic Interaction of a Robotic Arm and the Space Shuttle Remote Manipulator System

5. FUNDING NUMBERS
NCC1-104
233-03-03-02

6. AUTHOR(S)
Steven L. Garrahan, Robert H. Tolson, and
Robert L. Williams II

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The George Washington University
Joint Institute for Advancement of Flight Sciences
NASA Langley Research Center
Hampton, Virginia 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23681-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-195076

11. SUPPLEMENTARY NOTES Langley Technical Monitor: P.W. Goode
The information presented in this report was offered as a thesis by the first author in partial fulfillment of the requirements for the Degree of Master of Science, The George Washington University, December 1994.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 63

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Industrial robots are usually attached to a rigid base. Placing the robot on a compliant base introduces dynamic coupling between the two systems. The Vehicle Emulation System (VES) is a six DOF platform that is capable of modeling this interaction. The VES employs a force-torque sensor as the interface between robot and base. A computer simulation of the VES is presented. Each of the hardware and software components are described and Simulink is used as the programming environment. The simulation performance is compared with experimental results to validate accuracy. A second simulation which models the dynamic interaction of a robot and a flexible base, acts as a comparison to the simulated motion of the VES. Results are presented that compare the simulated VES motion with the motion of the VES hardware using the same admittance model. The two computer simulations are compared to determine how well the VES is expected to emulate the desired motion. Simulation results are given for robots mounted to the end effector of the Space Shuttle Remote Manipulator System (SRMS). It is shown that for fast motions of the two robots studied, the SRMS experiences disturbances on the order of centimeters. Larger disturbances are possible if different manipulators are used.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Robotics, control, remote manipulator, flexible robotic bases, disturbance compensation	107
	16. PRICE CODE A06

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	