Doctoral Dissertations and Master's Theses

Spring 5-2017

Development of an Emulated Free-Floating Environment for On-Earth Testing of Space Robots

Narendran Muraleedharan
Embry-Riddle Aeronautical University

Follow this and additional works at: https://commons.erau.edu/edt

Part of the Mechanical Engineering Commons

DEVELOPMENT OF AN EMULATED FREE-FLOATING ENVIRONMENT FOR
ON-EARTH TESTING OF SPACE ROBOTS

by

Narendran Muraleedharan

A Thesis Submitted to the College of Engineering Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Mechanical Engineering

Embry-Riddle Aeronautical University

Daytona Beach, Florida

May, 2017

DEVELOPMENT OF AN EMULATED FREE-FLOATING ENVIRONMENT FOR ON-EARTH TESTING OF SPACE ROBOTS

by

Narendran Muraleedharan

This thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Iacopo Gentilini, Associate Professor, Prescott Campus, and Thesis Committee Members Dr. Douglas Isenberg, Assistant Professor, University of South Alabama, Dr. Charles Reinholtz, Professor and Department Chair, Daytona Beach Campus, Dr. Sergey Drakunov, Professor and Associate Dean, Daytona Beach Campus, and Mr. Leslie Quiocho, Robotics Engineer, NASA Johnson Space Center, and has been approved by the Thesis Committee. It was submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering.

Thesis Review Committee:

Iacopo Gentilini
Digitally signed by Iacopo Gentilini
Date: 2017.05.03 09:11:54 -07'00'

Iacopo Gentilini, Ph.D.
Committee Chair

Douglas R. Isenberg
Digitally signed by Douglas R. Isenberg
DN: cn=Douglas R. Isenberg, o=University of South Alabama, ou=Department of Electrical and Computer Engineering, email=disenberg@southalabama.edu, c=US
Date: 2017.05.03 13:10:51 -05'00'

Douglas R. Isenberg, Ph.D.
Committee Member

Sergey V. Drakunov
Digitally signed by Sergey V. Drakunov
DN: cn=Sergey V. Drakunov, o=ERAU, ou=COAS, email=drakunov@erau.edu, c=US
Date: 2017.05.03 13:51:35 -04'00'

Sergey Drakunov, Ph.D.
Committee Member

LESLIE QUIOCHO
Digitally signed by LESLIE QUIOCHO
DN: c=US, o=U.S. Government, ou=NASA, ou=PM, cn=LESLIE QUIOCHO, 0.9.2342.19200300.100.1.1=lquiocho
Date: 2017.05.03 12:31:44 -05'00'

Leslie J. Quiocho, M.S.
Committee Member

Jean-Michel Dhainaut
Digitally signed by Jean-Michel Dhainaut
Date: 2017.05.03 12:19:43 -04'00'

Jean-Michel Dhainaut, Ph.D.
Graduate Program Chair,
Mechanical Engineering

Digitally signed by Charles Reinholtz
DN: cn=Charles Reinholtz, o=Embry-Riddle, ou=Mechanical Engineering, email=reinholc@erau.edu, c=US
Date: 2017.05.04 16:18:35 -04'00'

Charles F. Reinholtz, Ph.D.
Department Chair,
Mechanical Engineering

Maj Mirmirani
Digitally signed by Maj Mirmirani
DN: cn=Maj Mirmirani, o=Embry-Riddle Aeronautical University, ou=College of Engineering, email=mirmiram@erau.edu, c=US
Date: 2017.05.04 15:45:28 -05'00'

Maj Mirmirani, Ph.D.
Dean, College of Engineering

Christopher Grant
Digitally signed by Christopher Grant
Date: 2017.05.04 16:57:29 -04'00'

Christopher Grant, Ph.D.
Associate Vice President of Academics

_____
Date

Acknowledgements

Abstract

Researcher:	Narendran Muraleedharan
Title:	Development of an Emulated Free-Floating Environment for On-Earth Testing of Space Robots
Institution:	Embry-Riddle Aeronautical University
Degree:	Master of Science in Mechanical Engineering
Year:	2017

The ability to perform experiments on space robotic systems within a laboratory setting is crucial to development and testing of satellites and space robots prior to launch. One of the most widely used techniques which recreates the on-orbit motion of space robots and targets is hardware-in-the-loop simulation. This method requires extensive knowledge of the space robot model dynamic parameters. This research proposes a method which uses force feedback to control a robotic platform on which the space robot is mounted. The robotic platform is driven in such a way that the gravity-compensated forces and torques at the mounting interface is nullified. This method requires minimal knowledge of the system model and dynamic parameters. In this thesis, simulations are performed on both two-dimensional and three-dimensional systems and experimental validation on a two-dimensional system is conducted for proof-of-concept.

Table of Contents

Chapter

List of Tables

List of Figures

Chapter I

Introduction

This thesis proposes a method for emulating planar and spatial free-floating environments on Earth using minimalist knowledge of the target system model and dynamic parameters. The method is tested in simulation on planar and spatial systems and experimentally validated on a planar system.

Significance of the Study

A significant amount of research has been conducted in the field of space robotics over the last decade and continues to grow. The growth and interest in the subject are because the risk factor involved in launching human space-flight missions is still very high [1, 2]. Space agencies are moving towards the use of remotely piloted or unmanned autonomous systems to eliminate the human risk factor and the need for life support systems. This removal can significantly decrease the weight of the system and reduce mission costs [3]. Space Robots (SRs) can be used to perform a variety of missions on orbit includes performing maintenance or on-orbit servicing (OOS) operations on other satellites, debris removal, and assisting astronauts in various tasks. The number of on-orbit failures of such complex space robotic systems and satellites has exceeded the number of rocket launch failures in recent years [4, 5]. The limited ability for direct human intervention and maintenance during such missions can inhibit problem solving capabilities during failures seen in such missions. Therefore, the performance and systems on such SRs need to be tested extensively prior to launch [6].

Statement of the Problem

To be able to perform ground testing on hardware components and systems such as computer vision systems and berthing mechanisms on-board a space robot, a simula-

tion may not be sufficient. The need for testing hardware components calls for the ability to perform testing in a laboratory setting.

Furthermore, due to the complex dynamic coupling between manipulator joints of a space robot and the body, the body cannot be fixed and needs to be allowed to move in an emulated free-floating environment.

The methods used by space agencies and test facilities today are either limited by emulation time, purely two-dimensional motion or require detailed knowledge of the system model and dynamic parameters which are difficult to accurately measure or estimate.

Purpose Statement

The purpose of this thesis is to demonstrate that a hardware planar and spatial free-floating environment can be emulated using force-feedback control.

Delimitations

For the purpose of testing in simulation, ideal conditions are assumed by ignoring measurement noise.

For experimental validation, hardware and sensors are chosen to best meet the needs and closely match simulated measurements. In this case, a light-weight Space Robot (SR) with a four-link manipulator is used and the sensors' ranges, calibrations, and the operational work area of the Robotic Platform (RP) are chosen to meet the specific needs of this space robot.

Limitations

Many limitations were encountered through the research which required temporary or long-term solutions. These limitations were primarily seen in the experimental setup, and have been ignored in simulations. Some of these include measurement noise,

misalignment of force-torque (FT) transducer, latency in data transmission, and processing capabilities.

Certain limitations in hardware (e.g. power availability, discretization) have been modeled in simulations as well.

A major limitation faced in this research is time constraint, which did not allow for experimental validation of a complete three-dimensional testbed.

Definition of Terms

Space Robot	A satellite with an attached robotic manipulator
Robotic Platform	A Robotic manipulator used as the test bed platform on which the FT sensor and space robot are mounted
Free-floating	A theoretically friction-less environment in which linear and angular momentum are conserved and not exchanged with the surroundings
Dynamic Model	Equations of motion of a system that account for static and dynamic responses to input
Mechanical Model	Purely mechanical equations of motion of a system
Electro-mechanical Model	Combined electrical and mechanical equations of motion of a system
Dynamic Integration	Integration of system-generalized velocities and accelerations to generate trajectory
Zero Mass Moments	Masses of rigid bodies in the system
First Mass Moments	Products of masses and center of masses of rigid bodies in the system
Second Mass Moments	Inertia tensors of rigid bodies in the system

Generalized Coordinates	Vector of variables that define the linear and angular positions of the system
Generalized Velocities	Derivative of the generalized coordinates that define the linear and angular velocities of the system
Generalized Accelerations	Derivative of the generalized velocities that define the linear and angular accelerations of the system
Generalized Forces	Vector of forces and torques in the directions of the generalized coordinates
Hardware-in-the-Loop	A hybrid simulation technique where the equations of motion of the system are simulated and a tracking controller is used to drive the system through the simulated trajectory
System Excitation	Inputs provided to the system in terms of voltages to actuators or thruster impulses

List of Acronyms

AR&D	Autonomous Rendezvous and Docking
ASR	Actual Space Robot
AT	Actual Trajectory
CAD	Computer Aided Design
DLR	German Aerospace Center
DOF	Degrees of Freedom
EPOS	European Proximity Operation Simulator
ESA	European Space Agency
FF	Force-feedback
FT	Force-Torque
HIL	Hardware-in-the-Loop

MSR	Modeled Space Robot
MT	Modeled Trajectory
NASA	National Aeronautics and Space Administration
OOS	On-orbit Servicing
PCI	Peripheral Component Interconnect
PCIe	Express Peripheral Component Interconnect
PD	Proportional, Derivative
PID	Proportional, Integral, Derivative
RMS	Root Mean Square
RP	Robotic Platform
SR	Space Robot
USB	Universal Serial Bus
VES	Vehicular Emulation System

<center>**Chapter II**</center>

<center>**Literature Review**</center>

A key concern is that motion of the space robot's robotic manipulator joints results in significant displacement and rotation of the spacecraft body due to dynamic coupling [7]. This displacement means the spacecraft body cannot be fixed and needs to be allowed to move freely. Therefore, a realistic and reliable method is needed to emulate a free-floating environment.

Review of Current Solutions

Various solutions are already available including Hardware-in-the-loop (HIL) simulations, air-bearing tables, and parabolic flight [8] and have their own merits and limitations. Below is a review of some of the common solutions used to emulate a part of complete free-floating environment.

Air-bearing Table. An air-bearing table is a smooth, flat surface, generally made of polished granite on which the space robot or target test satellite is placed [9] as shown in Figure 2.1.

The spacecraft floats on a thin gas film created on air bearings mounted on the body. Surface friction is eliminated and the spacecraft is only subject to air friction, allowing it to translate and rotate in two dimensions. Air-bearing tables are used at many testing facilities including the Stanford Aerospace Robotics Laboratory and National Technical University of Athens to study spacecraft control, docking and collision avoidance systems [10, 11].

The major hurdle with the use of air-bearing tables is that the system is limited to purely planar motion. The design of the space robot and satellite are also limited by placement of air bearings and pressurized gas chambers.

<center>6</center>

Figure 2.1: Multiple Spacecraft on an Air-bearing Table at the Stanford University's Aerospace Robotics Laboratory *(Source: Stanford University Free Flyers Project Page [https://web.stanford.edu/group/arl/projects/free-flyers])*
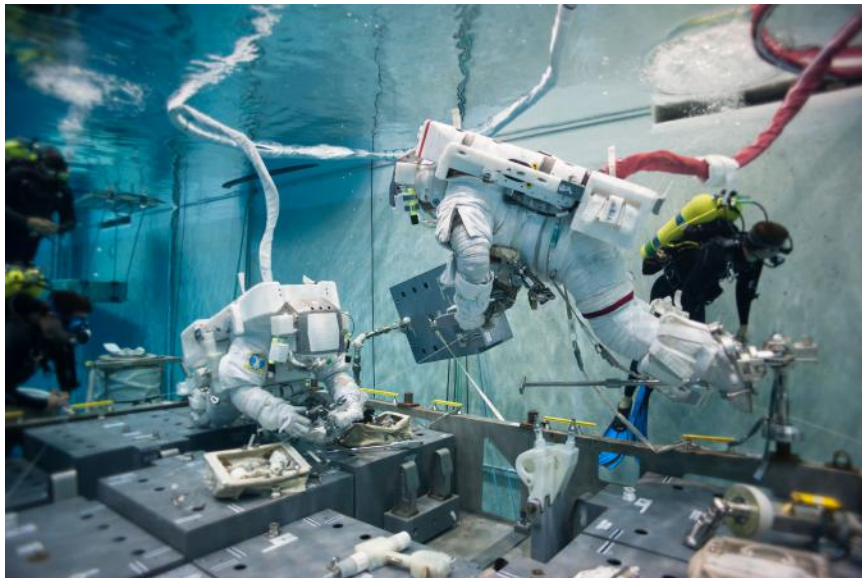
Figure 2.2: Extra-vehicular Activity Test at the NASA Johnson Space Center Neutral Buoyancy Laboratory *(Source: Wikimedia Commons [https://upload.wikimedia.org/wikipedia/commons/5/5d/Terry_Virts_simulates_extravehicular_activity_in_the_Neutral_Buoyancy_Laboratory.jpg])*

Underwater Neutral-Buoyancy System. The NASA Johnson Space Center uses a large pool to emulate a micro-gravity and partly free-floating condition in the Neutral Buoyancy Lab as seen in Figure 2.2. This method is very effective for practicing Extra Vehicular Activity (EVA) for astronauts and human missions that are generally difficult to perform due to constrictions in space suits. However, this method is not ideal to emulate a pure free-floating environment because the viscosity of the fluid introduces unwanted inertial and viscous effects which prevent space robots from moving the way they would in orbit [12].

Parabolic Flight. An aircraft performing parabolic flight has been used to emulate a true spatial micro-gravity environment as well. Figure 2.3 shows scientific passengers aboard the European Space Agency's (ESA) Zero-G aircraft performing micro-gravity experiments during parabolic flight.

Figure 2.3: Micro-gravity Experiments Being Performed Aboard the ESA's Zero-G Aircraft *(Source: ESA Human Spaceflight News [http://m.esa.int/Our_Activities/Human_Spaceflight/ESA_Parabolic_Flight_Campaign_successfully_completed_in_Bordeaux])*

Parabolic flight has also been used for testing of a 4 Degree-of-freedom (DOF) space robotic arm. The testing was conducted jointly by the Tokyo Institute of Technology and the National Space Development Agency of Japan. This solution allows for testing in gravitational accelerations down to 0.02g; however, the aircraft can be in parabolic flight for short periods of time up to 20s [13]. The limited experiment time makes the method not viable for testing long time-consuming missions including docking and on-orbit servicing.

Figure 2.4: On-Orbit Servicing (OOS) Simulator at the German Aerospace Center (DLR) European Proximity Operations Simulator (EPOS) Facility *(Source: DLR Institute for Robotics and Mechatronics [http://rmc.dlr.de/rm/de/staff/roberto.lampariello/])*

Hardware-in-the-Loop Simulation. HIL simulations or Hybrid Experiment Systems as seen in Figure 2.4 is a common tool used by many space agencies and test facilities to emulate a free-floating environment for ground testing of Autonomous Rendezvous and Docking (AR&D), and OOS missions. This method utilizes a dynamic model

of the system and simulates the system to generate a target trajectory. This target trajectory is then tracked in real-time by a RP on which the target SR is mounted. HIL simulation is a very powerful tool which can be used to test hardware systems for the spacecraft or control methods which are difficult to test purely in software. The Canadian Space Agency has used HIL simulations to test the performance and integrate Neptec's laser camera system [14].

An admittance model can also be used to handle physical interactions and external contact forces on the spacecraft using Force-Torque (FT) sensors on the RP end-effector. This approach has been studied and used by test facilities including the German Aerospace Center's (DLR's) European Proximity Operations Simulator (EPOS) facility [15, 16, 17, 18] and the Vehicle Emulation System Model II (VES II) test bed at MIT [19]. Studies have also shown that an FT sensor can be placed at the base of the robot as opposed to the end-effector, which allows the admittance model to act with respect to measured forces and torques also on the structure [20].

Another benefit is that HIL simulation systems allow testing for extended periods of time. However, in order to simulate the SR dynamics, a precise knowledge of the SR's equations of motion and H-matrix parameters including inertia tensors for each body are required [8]. Dynamic parameters such as inertia tensors for each body are difficult to accurately measure or estimate.

Suspension System. Mechanical gantry and boom-based systems using pulleys and counter-weights have been suggested and developed at the Carnegie Mellon University Robotics Institute as a testing method which would not require active control or power for the vertical axis. The system was tested and the frictional disturbances to the robot were shown to be typically less than 0.02g. However, the use of a counterweight increases the translational inertia along the vertical axis by 10% [21]. The difference in vertical inertia is significant for testing of small scale manipulators and SRs and to repli-

cate the free-floating trajectory, the dynamic parameters such as inertia must be as close to the actual value as possible. An error in dynamic parameters of as little as 10% can result in significant inconsistencies in the generated trajectory [6].

Gravity Compensation. Gravity compensation is not a new topic in the field of robotics and has been studied extensively for manipulator functions for human-assistance robots, and compliance control. Luca and Panzieri proposed and proved mathematical convergence of a learning gravity compensation method in which an iterative compensation scheme is used to achieve set-point regulation on a hanging robotic manipulator. In this method, a first control iteration is performed with proportional-derivative (PD) control. The unknown gravitational effect is computed by reading the control effort at steady state. A feed-forward term is generated and applied to the next iteration. The procedure is repeated and the gravitational effect estimate is improved in time [22]. Research has also been conducted on controlling the vertical axis actively. A hybrid approach using counterweights and a motor to control the vertical axis has been developed by Xu and White in which the tension in the suspension cable is measured and regulated about the counterweight static reaction [23].

It can be shown mathematically that any manipulator equilibrium position is globally asymptotically stable with a gravity-compensated PD control scheme if the gravity compensation term is computed in real time [24]. To compute the gravitational effect term, the masses, center of masses of each body the SR is composed of, and measured joint angles are used to estimate the location of the center of gravity of the SR.

Summary

Various methods of emulating free-floating environments to act as testbeds for AR&D and OOS missions are used by facilities around the world, each with their own merits and limitations. The method proposed in this thesis uses force-feedback (FF) with gravity compensation. The goal is to provide an emulation system that is capable of running for extended periods of time with minimal knowledge of the system dynamic model and parameters. In the future, the hardware testbed will be expanded into three dimensions for a true spatial free-floating environment.

Chapter III

Methodology

A free-floating environment is one in which there is no exchange of linear or angular momentum between the free-floating body (or set of bodies) and the surroundings. In order to maintain a free-floating environment, the total linear and angular momentum of the bodies remains constant unless acted upon by an external force. External forces can be in the form of contact, thrusters, solar wind, etc. For a SR moving independently as considered in this research, thrusters induce the only external forces.

Conserving linear and angular momentum implies that no forces or torques are applied by the SR to the surroundings or vice versa. The approach taken by the proposed FF control solution involves measuring forces and torques at the SR mounting location and driving the RP in such a way that the measured forced and torques are regulated about zero for a planar system and a gravity-compensation vector for a spatial system.

The solution involves an RP with a FT transducer on the end-effector on which the SR or target simulation body is to be mounted. The RP is a robotic manipulator capable of moving the end-effector to at least 3 DOF for a planar system and 6 DOF for a spatial system. The FT transducer is a 6-axis sensor capable of measuring forces and torques along three orthogonal axes. Since the FT transducer is mounted on the RP end-effector, the measures forces and torques are in the directions of the RP end-effector coordinate frame.

Control inputs to the RP actuators are provided in order to satisfy the control goal of driving the forces and torques measured at the interface between the RP and SR to zero or a gravity-compensated vector. Under ideal circumstances, when the forces and torques between the SR and RP are eliminated, linear and angular momentum of the SR are conserved, hence resulting in a free-floating environment. It is important to consider that

13

there are other means to exchange momentum with the surroundings (e.g. air friction). For the purposes of this research, the SR will be moved very slowly with respect to ambient air minimizing drag forces. This assumption is safe since the target application for the free-floating environment testbed is to emulate AR&D and OOS procedures that are characterized by slow relative velocities.

In order to test and demonstrate the viability of the proposed method, simulations and experimental validations are performed. Firstly, the viability of FF control is tested in a planar setting where the SR is only expected to experience motion about a plane. The planar system is modeled and tested in simulation, and compared against the current industry standard (i.e. HIL simulation). The planar system is then implemented in a laboratory setting to perform experimental validation. Finally, an FF control is simulated in a spatial setting where gravity compensation is also required in both the RP and SR joints.

Planar System Simulation

The goal of testing a planar FF control system is to study the viability of FF control without the effects of gravity. A simulation is set up and the performance of FF control is compared against HIL simulation where a 10% measurement error in knowledge of H-matrix and G-vector parameters is introduced. Performance metrics used are errors in trajectory and changes in linear and angular momentum.

System Dynamic Modeling. Dynamic models are implemented for both the SR alone in free-floating environment and the SR mounted on an RP with an FT sensor. The model of the SR alone is used to generate a reference trajectory for comparison and also as the desired trajectory for HIL simulation. For the planar system, a 3-axis FT sensor is used in simulation which measures forces about the horizontal plane and torque about the normal to the plane.

The SR portion of the complete system model is identical to the separate SR model.

Figure 3.1 depicts a top-view of the planar system used for simulation.
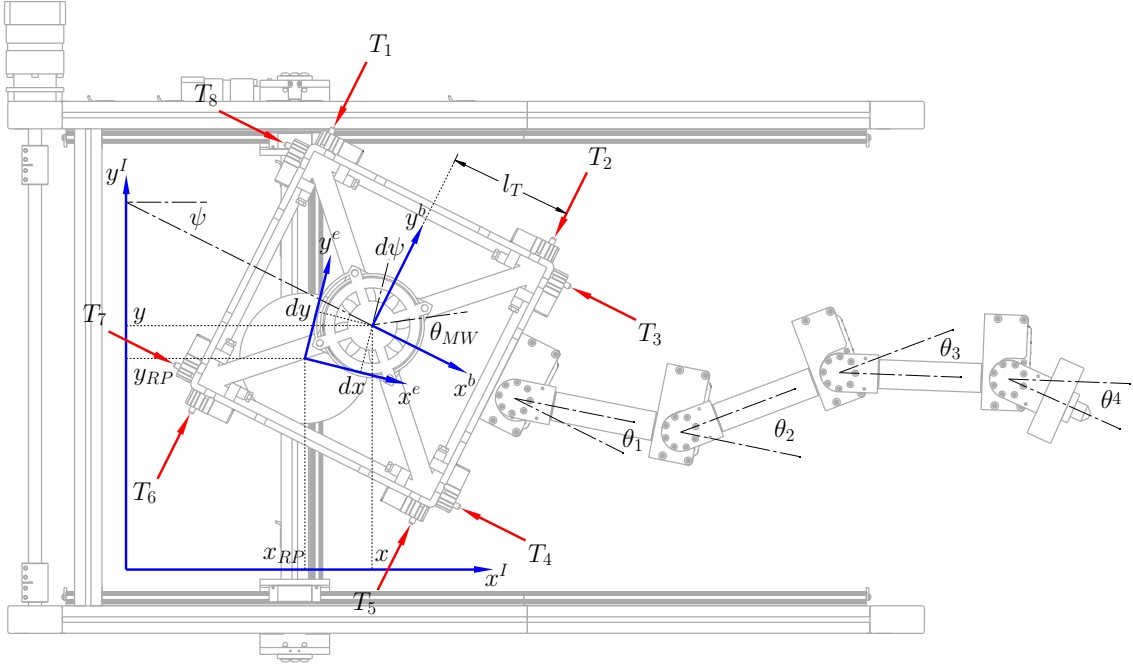
Figure 3.1: Coordinate Frames and Variables Visualization of an RP with an FT Transducer and an SR Used for Planar Simulation Dynamic Modeling

The RP is capable of translation along the inertial (considered as the RP base frame) x and y axes and rotation about the z axis. The deflections in the FT transducer are exaggerated in the figure as the transformation between the RP end-effector e and SR base b frames. The SR is comprised of a momentum wheel at the center of the SR body, a 4-link planar manipulator and 8 reaction control thrusters that generate forces in the directions T_1 to T_8.

Space Robot Model. The generalized coordinates vector of the planar SR, $\boldsymbol{q}_{SR2}$ is defined as:

$$\boldsymbol{q}_{SR2} = \begin{bmatrix} x_{SR2} \ y_{SR2} \ \psi_{SR2} \ \theta_{MW} \ \theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \end{bmatrix}^T, \tag{III.1}$$

where x_{SR2}, y_{SR2} and ψ_{SR2} define the planar position and orientation (pose) of the SR body frame with respect to an inertial frame, θ_{MW} is the rotation angle of the momentum wheel and θ_1 through θ_4 are the joint angles of the planar robotic manipulator on the SR.

The Lagrangian of the system $L(\boldsymbol{q}_{SR2}, \dot{\boldsymbol{q}}_{SR2})$ is computed as the difference between the total kinetic and potential energies in the system. The Euler-Lagrange formulation is used to derive the closed form equations of motion in the form:

$$\frac{d}{dt} \frac{\partial L(\boldsymbol{q}_{SR2}, \dot{\boldsymbol{q}}_{SR2})}{\partial \dot{\boldsymbol{q}}_{SR2}} - \frac{\partial L(\boldsymbol{q}_{SR2}, \dot{\boldsymbol{q}}_{SR2})}{\partial \boldsymbol{q}_{SR2}} + \boldsymbol{f}_f = \boldsymbol{f}_{SR2} = \begin{bmatrix} F_x \ F_y \ \tau_\psi \ \tau_{MW} \ \tau_1 \ \tau_2 \ \tau_3 \ \tau_4 \end{bmatrix}^T , \quad \text{(III.2)}$$

where $\boldsymbol{f}_f$ is the vector of frictional forces and $\boldsymbol{f}_{SR2}$ is the vector of forces and torques along the generalized coordinates of the planar SR.

In a free-floating environment, the frictional forces along the x_{SR2}, y_{SR2} and ψ_{SR2} directions are zero, resulting in the vector of frictional forces:

$$\boldsymbol{f}_f = [0\,0\,0\,f_{f,MW}\ f_{f,1}\ f_{f,2}\ f_{f,3}\ f_{f,4}]^T . \quad \text{(III.3)}$$

Friction in the momentum wheel and manipulator joints are modeled using viscous and Coulombic models:

$$f_{f,i} = \alpha_i \dot{\theta}_i + \beta_i \text{sgn}(\dot{\theta}_i), \quad \text{(III.4)}$$

where i is the momentum wheel or any joint of the manipulator, α_i is the coefficient of viscous friction, β_i is the coefficient of Coulombic friction, and sgn() is the signum function that returns a 1 for a positive input argument and a -1 for a negative input argument.

In Equation (III.2), $\boldsymbol{f}$ is the vector of the generalized forces on the SR. F_x and F_y are the forces applied along the generalized coordinates x and y. τ_ψ is the generalized

force applied about the generalized coordinate ψ. These forces and the torque are the sum of the forces and the torque applied by the force-torque sensor $F_{x,S}$, $F_{y,S}$, and $\tau_{\psi,S}$ and the resultant forces and torque applied by the thrusters $F_{x,T}$, $F_{y,T}$ and $\tau_{\psi,T}$:

$$\begin{bmatrix} F_x \\ F_y \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} F_{x,S} \\ F_{y,S} \\ \tau_{\psi,S} \end{bmatrix} + \begin{bmatrix} F_{x,T} \\ F_{y,T} \\ \tau_{\psi,T} \end{bmatrix} \tag{III.5}$$

The thrusters are modeled by applying forces along the x^b and y^b axes and a torque about the z^b axis using the equation:

$$\begin{bmatrix} F^b_{x,T} \\ F^b_{y,T} \\ \tau^b_{\psi,T} \end{bmatrix} = M \begin{bmatrix} T_1 \ T_2 \ T_3 \ T_4 \ T_5 \ T_6 \ T_7 \ T_8, \end{bmatrix}^T \tag{III.6}$$

where the forces from the thrusters are represented by T_1 through T_8 and M is a matrix that maps the individual thruster forces to forces and a torque in the body frame axes. M is obtained from the force and moment equilibrium equations of the thrusters and reactions:

$$F^b_{x,T} = T_7 + T_8 - T_2 - T_3 \tag{III.7}$$

$$F^b_{y,T} = T_5 + T_6 - T_1 - T_2 \tag{III.8}$$

$$\tau^b_{\psi,T} = l_T \left(T_1 + T_3 + T_5 + T_7 - T_2 - T_4 - T_6 - T_8 \right), \tag{III.9}$$

$$
\boldsymbol{M} = \begin{bmatrix} 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 \\ -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ l_T & -l_T & l_T & -l_T & l_T & -l_T & l_T & -l_T \end{bmatrix}, \tag{III.10}
$$

where l_T is the arm of the moment induced by the thrusters about the z-axis of the body frame.

The body frame forces and torque are transformed into the inertial frame via a rotation about the z^I (parallel to z^b) axis by the angle ψ:

$$
\begin{bmatrix} F_{x,T} \\ F_{y,T} \\ \tau_{\psi,T} \end{bmatrix} = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_{x,T}^b \\ F_{y,T}^b \\ \tau_{\psi,T}^b \end{bmatrix} \tag{III.11}
$$

The torque at the momentum wheel is given by τ_{MW}, and τ_1 through τ_4 are the torques at each of the manipulator joint angles. Standard DC geared motors are used to actuate the momentum wheel and each of the manipulator links. The motors are modeled using the following equation:

$$
\frac{C_i V_i}{R_i} = J_i N_i \ddot{\theta}_i + b_{v,i} N_i \dot{\theta}_i + b_{C,i} \text{sgn}(\dot{\theta}_i) + \frac{\tau_i}{N_i} + \frac{C_i N_i \dot{\theta}_i}{R_i}, \tag{III.12}
$$

as in [25] where armature inductance is assumed to be negligible. Here, V_i is the armature voltage for motor i, R_i is the armature resistance for motor i, C_i is both the motor torque constant and back electromotive force constant, N_i is the gear-ratio, J_i is the armature moment of inertia, $b_{v,i}$ is the viscous friction of the motor, and $b_{C,i}$ is the Coulombic friction.

Using the geared DC motor model, the generalized forces vector $\boldsymbol{f}_{SR2}$ is mapped

to the input vector $\boldsymbol{u}$ as:

$$\boldsymbol{u} = \tilde{\boldsymbol{J}}_m \ddot{\boldsymbol{q}}_{SR2} + \tilde{\boldsymbol{n}}(\dot{\boldsymbol{q}}_{SR2}) + \boldsymbol{R}_m \boldsymbol{f}_{SR2}. \qquad \text{(III.13)}$$

The input vector is given by:

$$\boldsymbol{u} = \begin{bmatrix} F_{x,T} & F_{y,T} & \tau_{\psi,T} & V_{MW} & V_1 & V_2 & V_3 & V_4 \end{bmatrix}^T, \qquad \text{(III.14)}$$

where V_{MW} is the armature voltage applied to the momentum wheel and V_1 through V_4 are the armature voltages applied to the DC motors actuating each of the manipulator links. $\tilde{\boldsymbol{J}}_m$ is a diagonal matrix defined as:

$$\tilde{\boldsymbol{J}}_m = \text{diag}([0\ 0\ 0\ \tilde{J}_{m,MW}\ \tilde{J}_{m,1}\ \tilde{J}_{m,2}\ \tilde{J}_{m,3}\ \tilde{J}_{m,4}]), \qquad \text{(III.15)}$$

where:

$$\tilde{J}_{m,i} = \frac{R_i J_i N_i}{C_i}. \qquad \text{(III.16)}$$

Additionally, $\tilde{\boldsymbol{n}}$ is a vector defined by the equation:

$$\tilde{\boldsymbol{n}} = [0\ 0\ 0\ \tilde{n}_{MW}\ \tilde{n}_1\ \tilde{n}_2\ \tilde{n}_3\ \tilde{n}_4]^T, \qquad \text{(III.17)}$$

where:

$$\tilde{n}_i = \frac{R_i b_{v,i} N_i \dot{\theta}_i + R_i b_{C,i} \text{sgn}(\dot{\theta}_i) + C_i^2 N_i \dot{\theta}_i}{C_i}. \qquad \text{(III.18)}$$

Furthermore, $\boldsymbol{R}_m$ is also a diagonal matrix that is defined as:

$$\boldsymbol{R}_m = \text{diag}([1\ 1\ 1\ R_{m,MW}\ R_{m,1}\ R_{m,2}\ R_{m,3}\ R_{m,4}]) \qquad \text{(III.19)}$$

where:

$$R_{m,i} = \frac{R_i}{N_i C_i}. \qquad \text{(III.20)}$$

It is well known that the equations of motion can be expressed in the following generalized form:

$$\boldsymbol{H}_{SR2}(\boldsymbol{q}_{SR2})\ddot{\boldsymbol{q}}_{SR2} + \boldsymbol{n}_{SR2}(\boldsymbol{q}_{SR2}, \dot{\boldsymbol{q}}_{SR2}) = \boldsymbol{f}_{SR2}, \qquad \text{(III.21)}$$

where $\boldsymbol{H}_{SR2}(\boldsymbol{q}_{SR2})$ is the system mass matrix and $\boldsymbol{n}_{SR2}(\boldsymbol{q}_{SR2}, \dot{\boldsymbol{q}}_{SR2})$ is a vector of generalized Coriolis, centripetal, and frictional forces and torques.

By substituting Equation (III.21) into Equation (III.13), the following electro-mechanical equations of motion for the SR are obtained:

$$\hat{\boldsymbol{H}}_{SR2}(\boldsymbol{q}_{SR2})\ddot{\boldsymbol{q}}_{SR2} + \hat{\boldsymbol{n}}_{SR2}(\boldsymbol{q}_{SR2}, \dot{\boldsymbol{q}}_{SR2}) = \boldsymbol{u}_{SR2}, \qquad \text{(III.22)}$$

where:

$$\hat{\boldsymbol{H}}_{SR2}(\boldsymbol{q}_{SR2}) = \tilde{\boldsymbol{J}}_m + \boldsymbol{R}_m \boldsymbol{H}_{SR2}(\boldsymbol{q}_{SR2}), \qquad \text{(III.23)}$$

and:

$$\hat{\boldsymbol{n}}_{SR2}(\boldsymbol{q}_{SR2}, \dot{\boldsymbol{q}}_{SR2}) = \boldsymbol{R}_m \boldsymbol{n}_{SR2}(\boldsymbol{q}_{SR2}, \dot{\boldsymbol{q}}_{SR2}) + \tilde{\boldsymbol{n}}_{SR2}. \qquad \text{(III.24)}$$

Equation (III.22) is the composite electro-mechanical equations of motion used and integrated to simulate the free-floating SR.

Force-Torque Transducer Model. In order to simulate the FT sensor mounted on the RP's end-effector, a spring damper system model is used.

The forces and the torque measured by the sensor are described by the following

equations:

$$F_{x,S}^e = K_x dx + b_x \dot{dx}, \tag{III.25}$$

$$F_{y,S}^e = K_y dy + b_y \dot{dy}, \tag{III.26}$$

and

$$\tau_{\psi,S}^e = K_\psi d\psi + b_\psi \dot{d\psi}. \tag{III.27}$$

where K_x, K_y and K_ψ are the linear and torsional spring constants of the FT sensor and b_x, b_y and b_ψ are the linear and torsional damping constants. dx, dy, and $d\psi$ are the relative translational and rotational displacements of the force and torque sensor, respectively.

This model is also used in the complete system model (below) to apply forces into the generalized directions of the FT sensor spring-dampers.

Robotic Platform Model. The RP model consists of a planar-motion platform with the force and torque sensor mounted onto the end-effector. The generalized coordinate vector for the RP, $\boldsymbol{q}_{RP2}$, is defined as:

$$\boldsymbol{q}_{RP2} = [x_{RP2}\ y_{RP2}\ \psi_{RP2}\ dx\ dy\ d\psi\ \theta_{MW}\ \theta_1\ \theta_2\ \theta_3\ \theta_4]^T, \tag{III.28}$$

where x_{RP2} and y_{RP2} are the coordinates of the RP end-effector frame e along the inertial axes x^I and y^I, ψ_{RP2} is the rotation of the RP end-effector about z^I, and θ_i are the joint angles of the momentum wheel and the manipulator joints.

The translational joints are driven by lead screws rotated by DC geared motors. Linear power transmission is modeled using the lead screw mechanical advantage equation:

$$F_{x,RP} = \frac{2\pi\eta}{l}\tau_{x,RP} \quad \text{and} \quad F_{y,RP} = \frac{2\pi\eta}{l}\tau_{y,RP}, \tag{III.29}$$

where $F_{x,RP}$ and $F_{y,RP}$ are the linear forces along the generalized coordinates x_{RP2} and y_{RP2}, respectively. Additionally, η is the mechanical advantage efficiency, l is the pitch of the threaded rod and $\tau_{x,RP}$ and $\tau_{y,RP}$ are the torques applied by the actuators. The rotational joint is driven by a standard DC geared motor.

The Euler-Lagrange formulation is utilized to derive the system mass matrix, $\mathbf{H}_{RP2}(\mathbf{q}_{RP2})$, along with the vector of generalized Coriolis, centripetal, and frictional forces and torques, $\mathbf{n}_{RP2}(\mathbf{q}_{RP2}, \dot{\mathbf{q}}_{RP2})$, for the RP. These terms are combined with motors modeled in the form of Equation (III.13) to obtain the following electro-mechanical equations of motion for the RP:

$$\hat{\mathbf{H}}_{RP2}(\mathbf{q}_{RP2})\ddot{\mathbf{q}}_{RP2} + \hat{\mathbf{n}}_{RP2}(\mathbf{q}_{RP2}, \dot{\mathbf{q}}_{RP2}) = \mathbf{u}_{RP2}. \quad\quad (III.30)$$

The derivation of the equations of motion for the complete system given in Equation (III.30) follows the same procedure used to derive the equations of motion for the SR previously.

Control Strategies. In the following section, the implemented control strategies and performance evaluations are introduced.

Hardware-in-the-Loop Simulation Verification. A computed-torque trajectory tracking controller is utilized by the RP for the HIL simulations that are presented in the next section. For the HIL simulations, a trajectory is generated either in real-time or off-line by numerically integrating the SR's equations of motion. This trajectory is then tracked by the RP's computed-torque tracking controller.

The vector of desired generalized coordinates for the RP, obtained by numerically

integrating the SR's equations of motion is defined as:

$$\boldsymbol{q}_D(t) = \left[x\ y\ \psi\ 0\ 0\ 0\ \theta_{MW}\ \theta_1\ \theta_2\ \theta_3\ \theta_4 \right]^T.$$ (III.31)

Here, the displacements of the force-torque sensor, dx. dy and $d\psi$ are assumed to be negligible as a result of high spring constants.

The trajectory-tracking controller utilizes an outer feedback loop defined as:

$$\boldsymbol{w} = \ddot{\boldsymbol{q}}_D + \boldsymbol{K}_p(\boldsymbol{q}_D - \boldsymbol{q}_{RP2}) + \boldsymbol{K}_d(\dot{\boldsymbol{q}}_D - \dot{\boldsymbol{q}}_{RP2}),$$ (III.32)

where $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are constant positive-definite diagonal gain matrices. This control term is utilized in conjunction with the following inner feedback linearizing control law:

$$\boldsymbol{u}_{RP2} = \hat{\boldsymbol{H}}_{RP2}(\boldsymbol{q}_{RP2})\boldsymbol{w} + \hat{\boldsymbol{n}}_{RP2}(\boldsymbol{q}_{RP2}, \dot{\boldsymbol{q}}_{RP2}).$$ (III.33)

Application of the above controller to Equation (III.30) results in a decoupled closed-loop system that is stable about the zero trajectory error. Figure 3.2 shows the block diagram for the tracking controller.
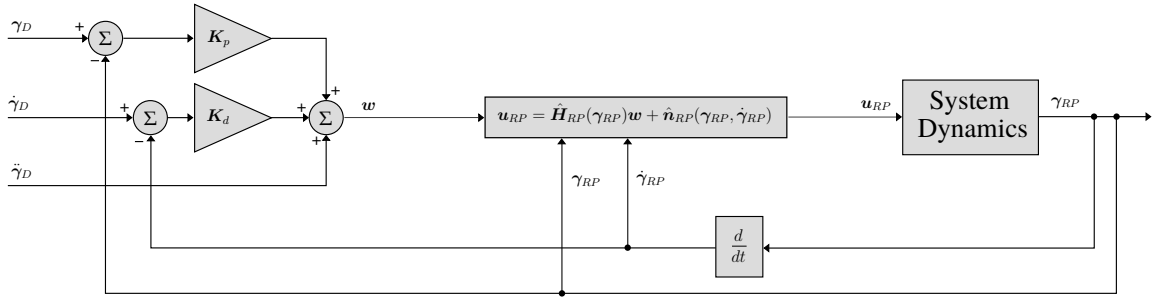
Figure 3.2: HIL Simulated-trajectory Tracking Controller Block Diagram.

Force-Feedback Control. A force-feedback control strategy is proposed as an alternative method to recreating free-floating SR motion. In this approach, non-zero forces and torque measured by the force-torque sensor are nullified by strategically actuating the translational and rotational links of the RP. Figure 3.3 is a block diagram of the proposed FF controller.
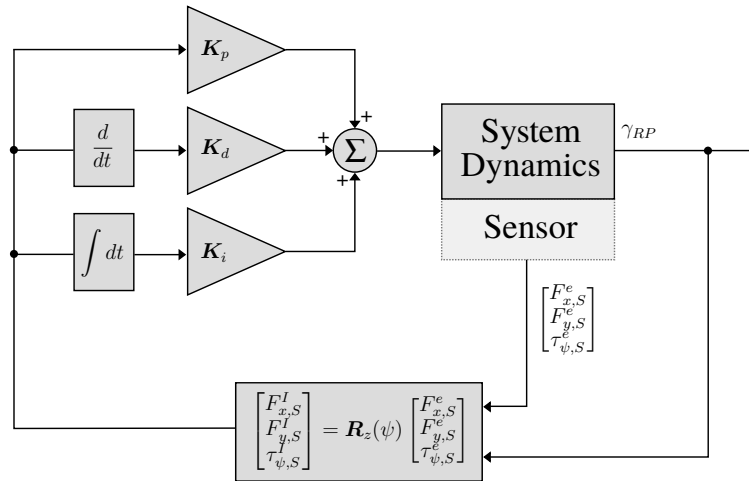
Figure 3.3: FF Controller Block Diagram.

The measured forces and torque, F_x^e, F_y^e, and τ_ψ^e are first transformed into forces and torque along the inertial frame axes, (i.e. F_x^I, F_y^I and τ_ψ^I) via a rotation through the angle $\psi_{RP2} \approx \psi_{RP2} - d\psi_{RP2}$ as:

$$
\begin{bmatrix} F_{x,S}^I \\ F_{y,S}^I \\ \tau_{\psi_{RP2},S}^I \end{bmatrix} = \begin{bmatrix} \cos(\psi_{RP2}) & -\sin(\psi_{RP2}) & 0 \\ \sin(\psi_{RP2}) & \cos(\psi_{RP2}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_{x,S}^e \\ F_{y,S}^e \\ \tau_{\psi_{RP2},S}^e \end{bmatrix}. \tag{III.34}
$$

A Proportional, Integral, and Derivative (PID) controller is used to drive the RP actuators in order to nullify the forces and torque at the interface:

$$
\begin{bmatrix} u_{RP,x} \\ u_{RP,y} \\ u_{RP,\psi_{RP2}} \end{bmatrix} = \boldsymbol{K}_p \begin{bmatrix} F_{x,S}^I \\ F_{y,S}^I \\ \tau_{\psi_{RP2},S}^I \end{bmatrix} + \boldsymbol{K}_i \int \begin{bmatrix} F_{x,S}^I \\ F_{y,S}^I \\ \tau_{\psi_{RP2},S}^I \end{bmatrix} dt + \boldsymbol{K}_d \frac{d}{dt} \begin{bmatrix} F_{x,S}^I \\ F_{y,S}^I \\ \tau_{\psi_{RP2},S}^I \end{bmatrix} \tag{III.35}
$$

Here, $\boldsymbol{K}_p$, $\boldsymbol{K}_i$ and $\boldsymbol{K}_d$ are the constant positive-definite gain matrices for the PID control terms.

The use of an integral term in the controller adds potential instability and drift in the robot trajectory. This instability is observed in the simulation results.

Figure 3.3 depicts a block diagram of the force-feedback controller where $\boldsymbol{R}_z(\psi_{RP2})$ is the rotation matrix about the z-axis through the angle ψ_{RP2}. It should be noted that this method does not require any knowledge of the SR's equations of motion or the dynamical parameters which, incidently, are required to generate the target trajectory for the HIL Simulated-Trajectory Tracking Control strategies.

Simulation Environment Development. To perform a large number of simulations with various inputs, a high performance simulation tool is necessary. As a result of

this research, a flexible and powerful dynamic model simulation tool was developed.

The simulation tool is capable of integrating the system equations of motion de-fined earlier. Additionally, the simulation tool was developed to have real-time graphical visualization of the system, live plotting, flexible model definition and configuration sys-tem and a scripting language interpreter to allow for fast debugging and testing without re-compilation.

The program is written in C++ and is operated from a command line.

Dynamics Integration Engine. The initial simulation tool used for planar FF control development and testing integrates the closed-form electro-mechanical equations of motion. The generalized velocities and accelerations are computed using the equations of motion and a 4th order Runga-Kutta integrator is used to generate the trajectory as shown in the equations below:

$$k_1(t) = \text{dynamics}(X(t), u(t)), \tag{III.36}$$

$$k_2(t) = \text{dynamics}(X(t) + k_1(t)\frac{dt}{2}, u(t)), \tag{III.37}$$

$$k_3(t) = \text{dynamics}(X(t) + k_2(t)\frac{dt}{2}, u(t)), \tag{III.38}$$

$$k_4(t) = \text{dynamics}(X(t) + k_3(t)dt, u(t)), \tag{III.39}$$

and:
$$X(t+dt) = X(t) + \left(\frac{k_1(t)}{6} + \frac{k_2(t)}{3} + \frac{k_3(t)}{3} + \frac{k_4(t)}{6}\right)dt. \tag{III.40}$$

In Equations (III.36) to (III.40), $X(t)$ is a time-varying state vector comprised of

the generalized coordinates and accelerations:

$$\boldsymbol{X}(t) = [\boldsymbol{q}_i \quad \dot{\boldsymbol{q}}_i]^T,$$
(III.41)

where $\boldsymbol{q}_i$ is the generalized coordinates vector of system i —either the planar SR alone ($\boldsymbol{q}_{SR2}$) or the complete system including the RP ($\boldsymbol{q}_{RP2}$). The function dynamics() is an algorithm that implements the equations of motion of the system to compute $\dot{\boldsymbol{X}}(t)$ given the current state $\boldsymbol{X}(t)$ and inputs $\boldsymbol{u}(t)$. The vectors $\boldsymbol{k}_1(t)$ through $\boldsymbol{k}_4(t)$ are intermediate variables used in the integration algorithm. The time step in the simulation is dt.

Equations (III.22) and (III.30) are re-arranged to solve for the generalized acceleration as follows:

$$\ddot{\boldsymbol{q}}_i = \left[\hat{\boldsymbol{H}}_i(\boldsymbol{q}_i) \right]^{-1} (\boldsymbol{u}_i - \hat{\boldsymbol{n}}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)).$$
(III.42)

The simulation tool utilizes the C++ Armadillo library [26] due to its high performance and ease of integration.

Real-time Visualization Engine. OpenSceneGraph [27] is used as a 3D graphics engine to visualize the SR and RP during simulation to aid controller development and testing. Computer Aided Design (CAD) models were made representing the SR and RP system and were loaded using the OpenSceneGraph engine. Each of the objects representing separate bodies in the CAD model are animated independently.

Figure 3.4 is a screenshot of the simulation tool with the RP and SR mounted on it:
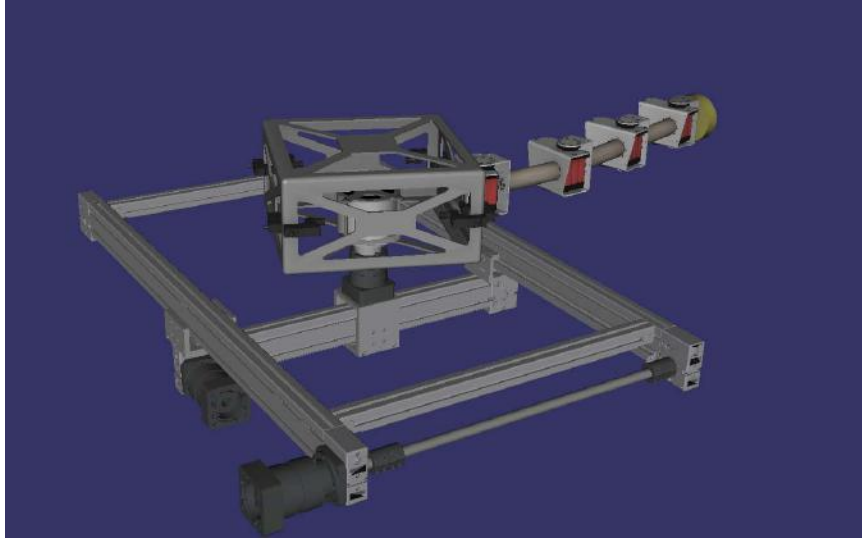
Figure 3.4: Screenshot of Robot Dynamics Simulator Developed for FF Controller Development and Testing

The simulation tool allows for transformations of objects in the CAD model using matrices to define 3D rotation, translation and scaling. These animations can be defined for each body in the model definitions. Translations are used to move the SR body frame or RP translational joints. Rotations are used for the SR orientation, RP rotational joint, momentum wheel and SR manipulator links.

Program Model Input System.   To allow for flexibility and simulation of various robotic systems (in this case, the RP and SR), the simulation tool was developed to accept a set of user defined text files defining the model parameters, variables and so forth.

The model definition file (*.model) holds information such as CAD model path, paths to other definition files, paths to scripts and configuration parameters that are read and used by the compiled program.

The variable definition file (*.vars) defines the generalized coordinates and input variables to be used. These variables are accessed from other parts of the program. Wrap-

ping and saturation of inputs can also be defined here.

The constant parameters definition file (*.params) holds constant parameters used in the dynamics() function. These parameters are not hard-coded to allow for fast testing and debugging. These parameters are also the zero, first and second mass moments to which an error is introduced to test HIL simulation.

The forward kinematics definition file (*.fk) is used to define serial robot links. The definition is used by the simulation tool to compute the center of mass, linear and angular momentums of the system.

The animations definition file (*.anim) is used define 3D animations defined above to visualize the robot in real-time during simulation. Object names from the CAD model, animation type (i.e. translate, rotate, scale), center, axis, factor and variable or property are defined for each object that needs to be animated.

Property Tree.   A property tree system is developed and used to handle variables that are critical to the simulation and for external variables used to test sub-systems. The model input interpreter allows properties from the tree to be used as variables for animations. Properties can also be logged and post-processed. The property tree is most useful in combination with the scripting system supported by the simulation tool.

Scripting Integration.   Furthermore, to allow for flexibility and fast debugging and development, the simulation tool was integrated with Lua scripting. Lua is a powerful and efficient scripting language that supports procedural programming, object-oriented programming, functional programming and data-driven programming and has been used in many industrial applications [28]. Lua scripting can be used in the simulation tool to run the core controller algorithm or additional scripts. The scripting is deeply integrated with many of the tool's core functions including access to graphics, robot dynamics, properties, and plotting. Lua scripts can be used to visualize and transform external models

(e.g. center of mass identifier, ground), generate custom live plots or write to file during simulation. Lua scripts in the tool also have access to the property tree and allows managing variables that affect animation, dynamics and so forth.

The combination of the above modules make the simulation tool a very powerful and high performance tool not only for verification but also future development. Aside from this research, the tool was also used to develop controllers for an omni-directional spherical robot [29], an under-actuated aircraft, and used to demonstrate and test many other robotic systems including 6 DOF manipulators, a four-wheel drive robot, an anthropomorphic arm, and the spatial free-floating environment system emulation testbed robot used for 3D free-floating environment emulation.

Planar System Experimental Validation

A planar free-floating environment testbed was constructed similar to the one simulated to prove viability and performance.

Apparatus and Hardware Setup. The following sections provides an overview of the hardware setup and equipment used to construct the testbed. Figure 3.5 is a diagram shows the connections and communications between components in the RP testbed.
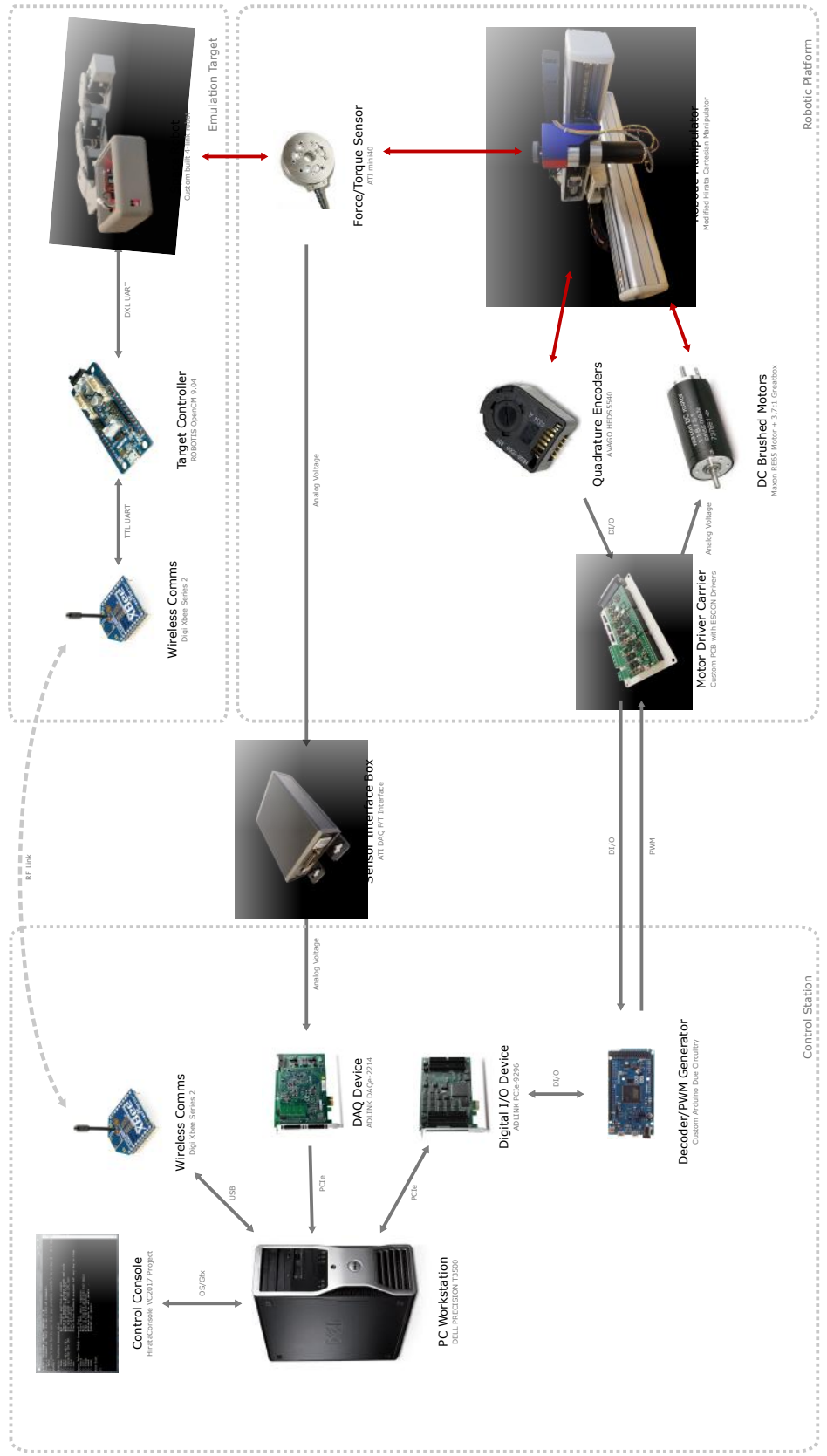
Figure 3.5: Experimental setup layout and communications diagram

Emulation Target

Custom built 4-link robotic

Target Controller
ROBOTIS OpenCM 9.04

Wireless Comms
Digi Xbee Series 2

Force/Torque Sensor
ATI mini40

Robotic Manipulator
Modified Hirata Cartesian Manipulator

Quadrature Encoders
AVAGO HEDS5540

DC Brushed Motors
Maxon RE65 Motor + 3.7:1 Greatbox

Motor Driver Carrier
Custom PCB with ESCON Drivers

Robotic Platform

Sensor Interface Box
ATI DAQ F/T Interface

DXL UART

TTL UART

Analog Voltage

RF Link

DI/O

Analog Voltage

DI/O

PWM

Analog Voltage

Wireless Comms
Digi Xbee Series 2

DAQ Device
ADLINK DAQe-2214

Digital I/O Device
ADLINK PCIe-P296

Decoder/PWM Generator
Custom Arduino Due Circuitry

Control Console
HirataConsole VC2017 Project

PC Workstation
DELL PRECISION T3500

USB

PCIe

PCIe

DI/O

OS/Gfx

Control Station

31

Figure 3.6: Complete Planar System Including RP, FT Sensor and SR

Figure 3.6 is a picture of the complete planar testbed system including the RP, FT sensor, SR and control station.

Robotic Platform. The RP is a modified Hirata Cartesian industrial robot [30]. The actuators were replaced with Maxon RE65 DC geared motors [31] to match the simulation model and a rotational axis was added to the end of the translational axes as seen in Figure 3.6. The RP has a functional work area of approximately 400 mm about the *x* axis and 320 mm about the *y* axis. The FT sensor cable limits the *z* axis rotation to approximately 720 degrees (two revolutions).

Force-Torque Transducer. An ATI mini40 [32] FT transducer was used to measure the forces and torques between the end-effector of the RP and the SR base frame. The specific sensor was selected based on performance characteristics to match the measured forces and torques in simulation. Figure 3.7 is a figure of the ATI mini40 sensor

mounted on the system with the SR body mounted on it:
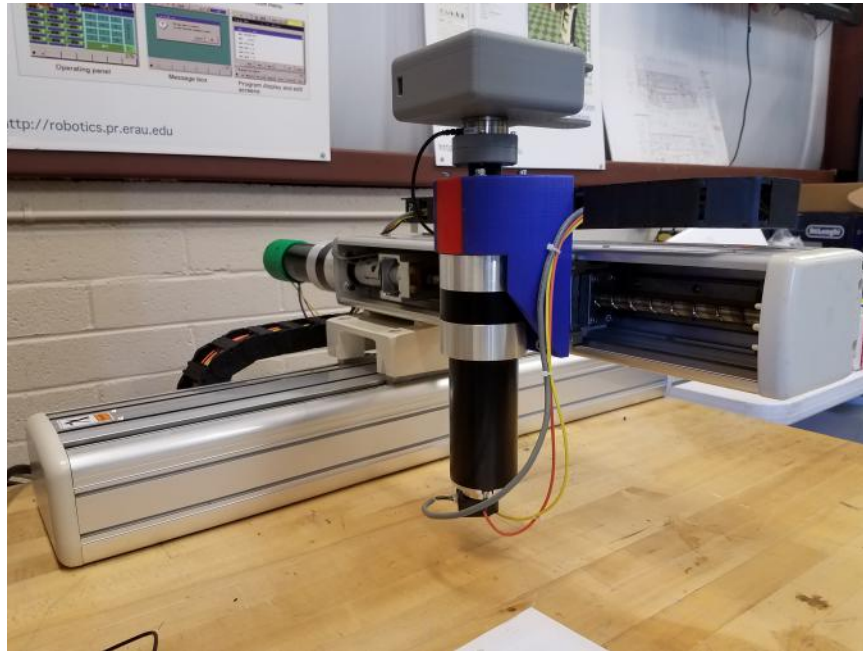
Figure 3.7: SR Body Mounted on an ATI mini40 FT Sensor on the RP End Effector

As seen in Figure 3.7, the FT sensor is the only form of physical contact between the SR and RP. Therefore, theoretically by regulating the forces and torques measured by the sensor about zero, the linear and angular momentum of the SR can be conserved.

An ATI data acquisition and interface unit is used to connect the FT sensor to the control station.

Space Robot. The SR is one similar to that used in simulation; however, the SR has no momentum wheel nor reaction control thrusters. For experimental validation, the linear and angular momentum are the performance metrics used and the 4-link planar robotic manipulator on the SR is sufficient in creating required system excitation inputs to test the performance. Additionally, thrusters are external forces applied to the system

which would change the momentum.

Four Robotis Dynamixel MX-28AT servos [33] are used for the planar manipulator on the SR. Custom SR body and links were 3D printed and assembled. A Robotis OpenCM Cortex-M3-based embedded microcontroller [34] is used to control the MX-28AT servos and a Digi Xbee RF Module [35] is used to communicate with the control station. The RF module is used to physically isolate the SR from the RP except through the FT sensor. The MX-28AT servos are operated in a velocity control mode where a goal speed is applied to the servos instead of a target position. The goal speed inputs allow the SR to more closely re-create the inputs used in simulation.

Control Station Interfacing. A desktop computer is used as the control station and performs critical tasks including running the core controller, managing soft limits, and logging data. As shown in Figure 3.5, the control station uses a DAQe-2213 analog data acquisition [36] and a PCIe-7296 digital input and output cards [37] that communicate with the computer through Express Peripheral Component Interconnect (PCIe). The Xbee communicates using a Universal Serial Bus (USB) FTDI breakout.

In order to interface the RP with the PCIe cards, a custom printed circuit board (PCB) was designed and fabricated. An additional circuit consisting of two Arduino Dues was constructed to interface the digital input and output pins on the PCIe-7296 card with the custom PCB. The function of the Arduino Due circuit is to decode quadrature signals from the three encoders on the RP and generate an absolute position digital output signal the PCIe-7296 card can read. The circuit reads digital motor commands and generates PWM signals to run the RP actuators.
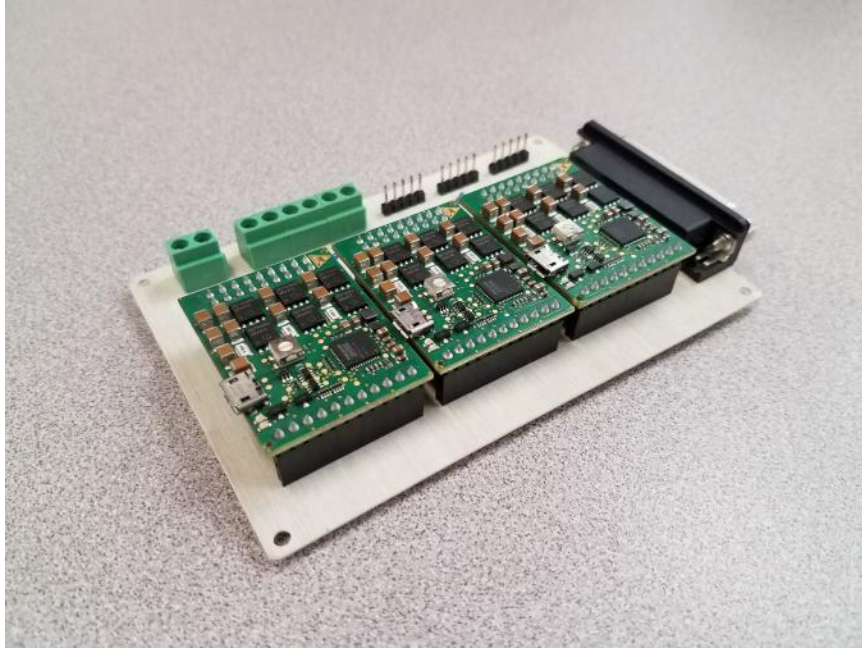
Figure 3.8: Fabricated Motor Driver Carrier Board Top Side

The custom PCB is a motor driver carrier board that interfaces the motors and encoders on the RP side to the PCIe adapter shield on the control station side.

Figure 3.8 is a picture of the top side of the motor driver carrier board with the ESCON motor drivers. Figure 3.9 shows the motor driver carrier board connected to the RP actuators (yellow, green and orange cables), the encoders (gray shielded cables) and the Arduino Due circuit through the DB-25 connector:
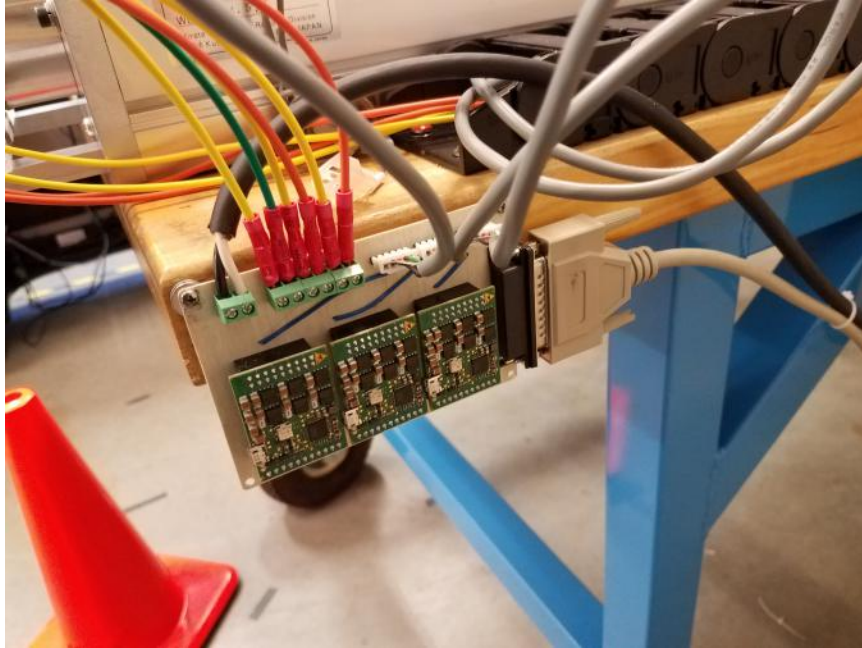
Figure 3.9: Motor Driver Carrier Board Mounted on Robotic Platform Table with Motor, Encoders, Power and Data Lines Connected

Also seen in Figure **??** is the DAQe-2213 card above the PCIe-7296 adapter shield used for analog data acquisition. The ATI FT sensor interface unit outputs six analog voltage signals that correlate to the forces and torques measured. A calibration matrix is required to obtain the individual forces and torques from these voltages. The analog voltages are measured by the DAQe-2213 and used in the control software.

Control Software Development. The control station software handles reading the forces and torques through the DAQe-2213 board, reading decoded encoder positions through the PCIe-7296 board, applying motor control commands through the PCIe-7296 board, communicating with the SR and logging all states.

The program is written in C++ and Visual Studio 2017 is used to compile and debug.

Console. A console is used to control the functions of the RP controller, control

the SR, and monitor the controller functions. The console is a live interactive window

that allows the operator to enter commands to perform various functions on both the SR

and RP including homing, moving to position, setting motor speeds, stopping, starting

the controller, starting the SR input sequence, and moving the SR manipulator to various

configurations.

Figure 3.10 is a screenshot of the control console displaying the help output.
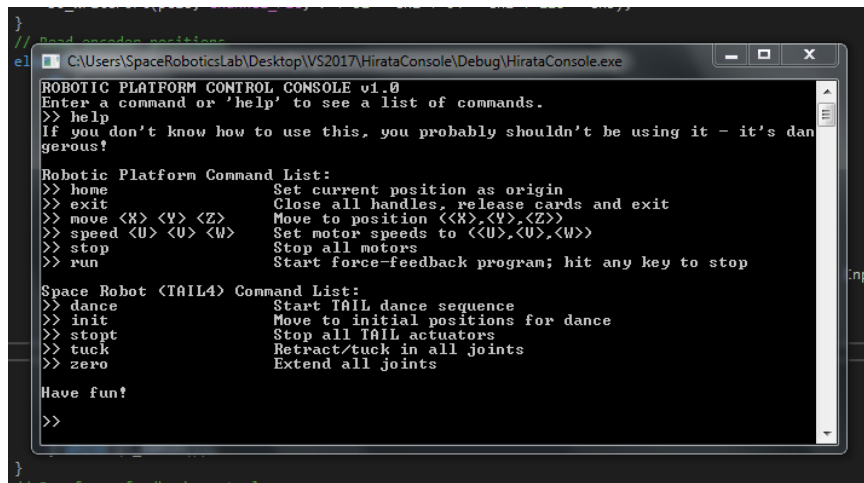
Figure 3.10: Control Station Console Used to Access and Perform Various Functions on
the RP and SR

Controller Implementation. The controllers discussed in the Control Strategies

section below are implemented in the program. The only matrix operation is the multipli-

cation of a 6x6 matrix with a 6x1 vector, which is performed using nested loops, eliminat-

ing the need for a linear algebra library.

The PCIe cards are accessed using ADLINK's PCIS-DASK and D2K-DASK li-

braries. The libraries allow sampling at very high frequencies, allowing for very fast con-

trol response.

Data Collection Scheme. Data are collected from the RP and SR for post processing and analyzing the performance of the controller.

Robotic Platform State Logging. The states of the RP are defined by the encoder positions, which are ready through the PCIe-7296 card using the PCIS-DASK library. These states are written to file along with the corresponding SR states.

Space Robot State Logging. The SR communicates to the control station with a Digi Xbee. The control station side of the Xbee is connected through a USB. A file handle is used to communicate with the USB serial port to log the SR states over time.

Control Strategies. The control goal for the planar free-floating environment testbed is to emulate motion of the SR in a true free-floating environment using force-feedback control. Comparing against a simulation of the actual system is very difficult since small errors in dynamic parameters can lead to significant disparities in trajectory. Therefore, the performance metric used will be conservation of linear momentum. The linear momentum of the SR can be expressed using the equation:

$$\boldsymbol{p}_{SR2} = m_{SR2}{}^I_I\dot{\boldsymbol{r}}_{cm,SR2}, \qquad\qquad (\text{III.43})$$

where $\boldsymbol{p}_{SR2}$ is the linear momentum of the of the planar SR, m_{SR2} is the total (measured) mass of the SR, and ${}^I_I\dot{\boldsymbol{r}}_{cm,SR2}$ is the velocity of the center of mass of the SR with respect to an inertial measurement frame, in this case considered to be equivalent to the RP base frame.

Unless an external force is provided (e.g. using thrusters or contact), the linear momentum of the system must remain constant. Given a static initial condition, the linear momentum must remain zero throughout the emulation under ideal circumstances. The

zero linear momentum requires the center of mass velocity to be zero, leaving the center of mass of the SR stationary throughout the emulation.

For simplification, the center of mass of the SR is used as the performance metric for analyzing the control methods. The center of mass of the SR is estimated using two different methods. The first method uses torque measurements about the FT sensor's x and y axes, which are not used by the FF controller. The center of mass along the plane of motion can be given by the equations:

$$^e x_{cm,SR2} = \frac{^e \tau_y}{m_{SR2}\, g_z},$$ (III.44)

and:

$$^e y_{cm,SR2} = -\frac{^e \tau_x}{m_{SR2}\, g_z},$$ (III.45)

where $^e x_{cm,SR2}$ and $^e y_{cm,SR2}$ are the x and y coordinates of the center of mass of the SR with respect to the RP end-effector frame, $^e \tau_y$ is the torque measurement about the FT sensor's y axis, $^e \tau_x$ is the torque measurement about the FT sensor's x axis and g_z is the z component of the gravitational acceleration vector.

The center of mass position in the inertial frame can be computed using the equation:

$$
\begin{aligned}
{}^I_I r_{cm,SR2} &= {}^I_I r_e + {}^I T_e\, {}^e r_{cm,SR2} \\
&= \begin{bmatrix} x_{RP2} \\ y_{RP2} \\ 0 \end{bmatrix} + \frac{1}{m_{SR2}\, g_z} R_z(\psi_{RP2}) \begin{bmatrix} \tau_y^e \\ -\tau_x^e \\ 0 \end{bmatrix}.
\end{aligned}
$$ (III.46)

This approach is capable of computing an estimate purely based on torque readings and does not require any information about the SR model. The approach is expected

to be reliable only for low frequency system excitations since forces in the x and y axes can translate into moments due to the displacement of the center of mass along the z axis that are not accounted for in the planar system. Measuring the fixed z component of the SR center of mass allows for compensation due to these induced moments as well.

Another approach used to estimate the center of mass of the SR takes advantage of a static model of the SR and the joint variables. The center of mass of a multi-body system can be computed using the equation:

$$ {}^{I}_{I}\boldsymbol{r}_{cm,SR2} = \frac{\Sigma m_i {}^{I}_{I}\boldsymbol{r}_{cm,i}}{\Sigma m_i}, \tag{III.47} $$

where m_i is the mass of body i and ${}^{I}_{I}\boldsymbol{r}_{cm,i}$ is the local center of mass of the body i for each body in the multi-body system.

This approach is expected to provide estimates of similar accuracies even for high frequency system excitations; however, errors in static parameter identification of the SR can lead to inaccurate CM estimates.

Figure 3.11 illustrates the center of mass estimates made using the torque measurements (magenta) and the static model (blue). It is observed that the estimates are within $5mm$ of each other:
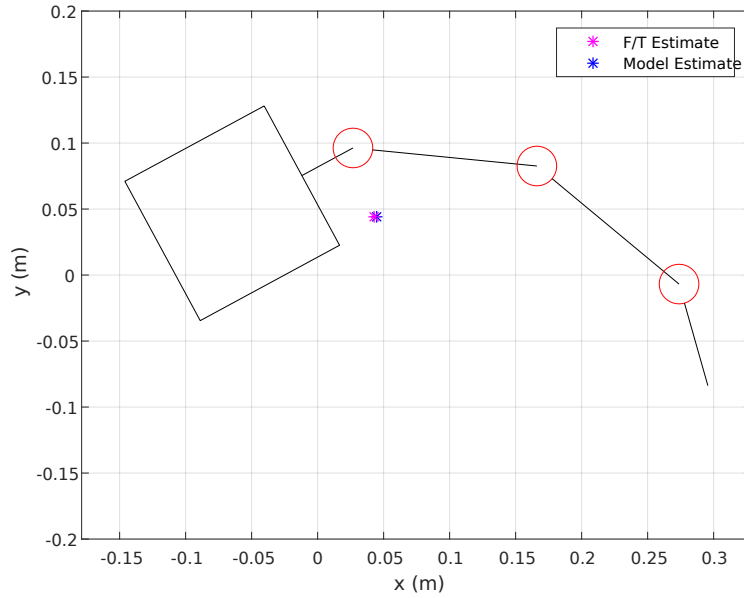
Figure 3.11: Center of Mass Estimates of the SR Made Using Measured Torques and Static Model

Pure Force-Feedback Control. Pure FF control is tested on the planar free-floating environment testbed using the implementation depicted by the controller block diagram in Figure 3.12.

As used in the 2D simulation, a PID controller is used to compute the RP input vector. The feedback loop is closed through the strain gauge voltage readings from the FT sensor and also the computation of the transformation matrix to transform the measured forces from the RP end-effector frame e to the inertial frame I.

Firstly, the measured voltages $\boldsymbol{v}_{sg}$ are unbiased using the calibration bias voltages $\boldsymbol{v}_{sg,0}$:

$$\boldsymbol{v}_{sg,calib} = \boldsymbol{v}_{sg} - \boldsymbol{v}_{sg,0}, \tag{III.48}$$

41

where $\boldsymbol{v}_{sg,calib}$ is the vector of calibrated voltages. The forces and torques in the RP end-effector frame $^{e}\boldsymbol{\xi}$ axes are computed by multiplying a calibration matrix C_{calib} with the calibrated strain gauge voltages $\boldsymbol{v}_{sg,calib}$:

$$^{e}\boldsymbol{\xi} = \begin{bmatrix} ^{e}F_x \, ^{e}F_y \, ^{e}F_z \, ^{e}\tau_{\phi} \, ^{e}\tau_{\theta} \, ^{e}\tau_{\psi} \end{bmatrix}^{T} = \boldsymbol{C}_{calib}\boldsymbol{v}_{sg,calib}. \tag{III.49}$$
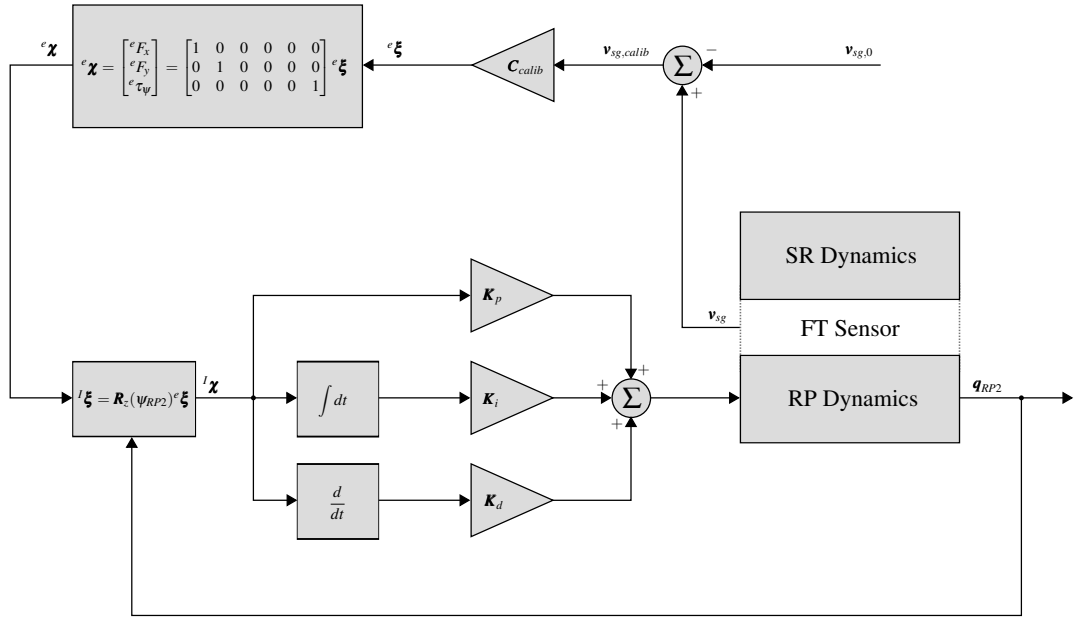
Figure 3.12: Block diagram of pure force-feedback control for the planar free-floating environment testbed

The calibration matrix is provided by the FT sensor manufacturer. The forces and torques required for the pure FF controller, $^{e}F_x$, $^{e}F_y$ and $^{e}\tau_{\psi}$, are selected to the form of a

vector $^e\boldsymbol{\chi}$ from the vector of all measured forces and torques:

$$^e\boldsymbol{\chi} = \begin{bmatrix} ^eF_x \\ ^eF_y \\ ^e\tau_\psi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} {}^e\boldsymbol{\xi}. \qquad \text{(III.50)}$$

The forces and torques in the inertial frame $^I\boldsymbol{\chi} = \boldsymbol{R}_z(\psi_{RP2})^e\boldsymbol{\chi}$ are used as inputs to a PID controller to yield inputs for the three planar RP actuators.

Center of Mass Regulator. Due to errors in measurement noise and the integral term in the pure FF controller, the SR drifts over time. In order to attempt to minimize the drift, a simple center of mass regulation controller is tested for the x and y axes while leaving control of the rotational axis to pure FF control. The center of mass regulation controller drives the RP in a manner that maintains the center of mass of the SR.

Equations (III.48) and (III.49) are used in the same fashion as the pure FF controller to attain the forces and torques measured by the FT sensor in the RP end-effector frame $^e\boldsymbol{\xi}$. The center of mass of the SR in the inertial frame is computed using Equation (III.46).

A PID controller is used to drive the RP actuators in order to maintain the center of mass of the SR.

The center of mass regulation controller output is augmented with the pure force-feedback controller to generate the control signals to be sent to the RP.

Figure 3.13 is a block diagram of the center of mass regulator used for testing.

43

Figure 3.13: Block Diagram of Center of Mass Regulation Controller Used for Testing

Spatial System Simulation

The use of FF control to emulate a spatial free-floating environment testbed is tested in simulation as part of this research.

The spatial testbed consists of a 7 DOF robotic manipulator as the RP with a 6-axis FT sensor on the end-effector. The SR is a 6 DOF robotic manipulator mounted on the FT sensor.

System Dynamic Modeling. The complete spatial testbed system consists of 19 generalized coordinates. As solving closed-form dynamic model requires an excessive amount of computational power, the Newton-Euler method is used to model the spatial system. The equations of motion using this approach are written separately for each body in the multi-body system and the inverse dynamics are solved for in real-time for each body. It can be analytically proven that by substitution of reaction forces, the closed form equations of motion using the Euler-Lagrange formulation can be attained [24].

A general framework of the Newton-Euler balance of forces and torques is followed for each body with different defining parameters. The defining equations are the Newton Equation for translational motion:

$$ {}^l\boldsymbol{f}_l - {}^l\boldsymbol{f}_{l+1} + m_l{}^l\boldsymbol{g} = m_l{}^l_I\ddot{\boldsymbol{r}}_{cm,l}, \tag{III.51} $$

and the Euler Equation for rotational motion:

$$ {}^l\boldsymbol{\tau}_l - {}^l\boldsymbol{\tau}_{l+1} + {}^l\boldsymbol{f}_l \times {}^l_{l-1}\boldsymbol{r}_{cm,l} - {}^l\boldsymbol{f}_{l+1} \times {}^l_l\boldsymbol{r}_{cm,l} = {}^l_l\boldsymbol{J}^l_{l-1}\dot{\boldsymbol{\omega}}_l + {}^l_{l-1}\boldsymbol{\omega}_l \times {}^l_l\boldsymbol{J}^l_{l-1}\boldsymbol{\omega}_l, \tag{III.52} $$

where:

- $\boldsymbol{f}_l$ is the vector of forces applied by link $l-1$ on link l

- $\boldsymbol{f}_{l+1}$ is the vector of forces applied by link l on link $l+1$

- m_l is the mass of link l

- ${}^l\boldsymbol{g}$ is the gravitational acceleration

- ${}^l_I\boldsymbol{r}_{cm,l}$ is the position of the center of mass of link l with respect to the inertial frame

- ${}^l\boldsymbol{\tau}_l$ is the torque applied by link $l-1$ on link l

- $^l\boldsymbol{\tau}_{l+1}$ is the torque applied by link l on link $l+1$

- $^l_{l-1}\boldsymbol{r}_{cm,l}$ is the position of the center mass of link l with respect to the joint of the previous link $l-1$

- $^l_l\boldsymbol{r}_{cm,l}$ is the position of the center of mass of link l

- l with respect to the joint of link l

- $^l_l\boldsymbol{J}$ is the inertia tensor of link l

- $^l_{l-1}\boldsymbol{\omega}_l$ is the angular velocity of link l from link $l-1$

all measured with respect to the link l frame.

The following equations are used to compute the linear and angular velocities and accelerations for each link in the serial robotic system:

$$
^l_{l-1}\boldsymbol{\omega}_l = \begin{cases} ^lT_{l-1}{}^{l-1}_{l-2}\boldsymbol{\omega}_{l-1} & \text{prismatic joint} \\ ^lT_{l-1}\left({}^{l-1}_{l-2}\boldsymbol{\omega}_{l-1} + \dot{q}_l\,[0\ 0\ 1]^T\right) & \text{revolute joint} \end{cases}, \qquad \text{(III.53)}
$$

$$
^l_{l-1}\dot{\boldsymbol{\omega}}_l = \begin{cases} ^lT_{l-1}{}^{l-1}_{l-2}\dot{\boldsymbol{\omega}}_{l-1} & \text{prismatic joint} \\ ^lT_{l-1}\left({}^{l-1}_{l-2}\dot{\boldsymbol{\omega}}_{l-1} + \ddot{q}_l\,[0\ 0\ 1]^T + \left(\dot{q}_l{}^{l-1}_{l-2}\boldsymbol{\omega}_{l-1} \times [0\ 0\ 1]^T\right)\right) & \text{revolute joint} \end{cases},
$$
$$
\text{(III.54)}
$$

and:

$$
^l_l\ddot{\boldsymbol{r}}^l = \begin{cases} ^lT_{l-1}\left({}^{l-1}_{l}\ddot{\boldsymbol{r}}_{l-1} + \ddot{q}_l\,[0\ 0\ 1]^T\right) + \left(2\dot{q}_l{}^l_{l-1}\boldsymbol{\omega}_l\right) \times \left(^lT_{l-1}\,[0\ 0\ 1]^T\right) \\ ^l_{l-1}\dot{\boldsymbol{\omega}}_l \times {}^l_{l-1}\boldsymbol{r}_l + {}^l_{l-1}\boldsymbol{\omega}_l \times \left({}^l_{l-1}\boldsymbol{\omega}_l \times {}^l_{l-1}\boldsymbol{r}_l\right) & \text{prismatic joint} \\ ^lT_{l-1}{}^{l-1}_{l}\ddot{\boldsymbol{r}}_{l-1} + {}^l_{l-1}\dot{\boldsymbol{\omega}}_l \times {}^l_{l-1}\boldsymbol{r}_l + {}^l_{l-1}\boldsymbol{\omega}_l \times \left({}^l_{l-1}\boldsymbol{\omega}_l \times {}^l_{l-1}\boldsymbol{r}_l\right) & \text{revolute joint} \end{cases},
$$
$$
\text{(III.55)}
$$

where q_l is the joint variable of link l - the displacement for a prismatic joint or the rotation angle for a revolute joint.

The linear acceleration of the center of mass is computed for each link using the equation:

$$_I^l \ddot{\pmb{r}}_{cm,l} = {}_I^l \ddot{\pmb{r}}_l + {}_{l-1}^l \dot{\pmb{\omega}}_l \times {}_l^l \ddot{\pmb{r}}_{cm,l} + {}_{l-1}^l \pmb{\omega}_l \times \left({}_{l-1}^l \pmb{\omega}_l \times {}_l^l \pmb{r}_{cm,l} \right). \tag{III.56}$$

The angular acceleration of the actuator rotor on link l is computed using the equation:

$$_{l-1}^{l-1} \dot{\pmb{\omega}}_{m,l} = {}_{l-2}^{l-1} \dot{\pmb{\omega}}_{l-1} + N_m \ddot{q}_l \, [0 \ 0 \ 1]^T + \left(N_m \dot{q}_l \, {}_{l-1}^l \pmb{\omega}_l \right) \times [0 \ 0 \ 1]^T, \tag{III.57}$$

where N_m is the gear ratio of the motor.

Using the joint linear and angular velocities and accelerations, the forces and torques used for the balance equations are computed as follows:

$$^l \pmb{f}_l = {}^l T_{l+1} \, {}^{l+1} \pmb{f}_{l+1} + m_l \, {}_I^l \ddot{\pmb{r}}_{cm,l}, \tag{III.58}$$

and:

$$^l \pmb{\tau}_l = {}^l T_{l+1} \, {}^{l+1} \pmb{\tau}_{l+1} + \left({}^l T_{l+1} \, {}^{l+1} \pmb{f}_{l+1} \right) \times {}_l^l \pmb{r}_{cm,l} + {}_l^l J_{l-1}^l \dot{\pmb{\omega}}_l + {}_{l-1}^l \pmb{\omega}_l \times \left({}_l^l J_{l-1}^l \pmb{\omega}_l \right) \tag{III.59}$$

$$+ N_m \ddot{q}_l J_m [0 \ 0 \ 1]^T + \left(N_m \dot{q}_l J_{m l-1}^l \pmb{\omega}_l \right) \times [0 \ 0 \ 1]^T - {}^l \pmb{f}_l \times \left({}_{l-1}^l \pmb{r}_l + {}_l^l \pmb{r}_{cm,l} \right),$$

where J_m is the rotor inertia of the actuator. It is also assumed that for all the links, the actuator spins along the joint axis.

It is important to note that Equations (III.58) and (III.59) are re-arrangements of Equations (III.51) and (III.52) respectively transformed into the link l frames. The grav-

itational acceleration is removed as it is assumed that the base frame is fixed and a negative gravitational acceleration is used as the initial condition.

Finally, the generalized joint forces can be computed for a prismatic joint by:

$$f_l = {}^l\boldsymbol{f}_l^T \, {}^l T_{l-1} [0\ 0\ 1]^T + N_m \, J_m \, {}^{l-1}_{l-1} \dot{\boldsymbol{\omega}}_{m,l}^T [0\ 0\ 1]^T + \alpha_l \, \dot{q}_l + \beta_i \, \mathrm{sgn}(\dot{q}_l), \qquad \text{(III.60)}$$

and joint torques for a revolute joint by:

$$\tau_l = {}^l\boldsymbol{\tau}_l^T \, {}^l T_{l-1} [0\ 0\ 1]^T + N_m \, J_m \, {}^{l-1}_{l-1} \dot{\boldsymbol{\omega}}_{m,l}^T [0\ 0\ 1]^T + \alpha_l \, \dot{q}_l + \beta_i \, \mathrm{sgn}(\dot{q}_l). \qquad \text{(III.61)}$$

Finally, the dynamic model of a link can be obtained using Equations (III.53) to (III.61). The same equations can also be used to model the complete system using the Newton-Euler recursive algorithm, assuming the system is comprised of a set of links in serial arrangement.

Space Robot Model. The SR alone in free-floating environment (used to generate comparison reference trajectories) is modeled as closed-form equations of motion using the Euler-Lagrange formulation as was done for the planar system. The closed form equations were used because the system only has 12 generalized coordinates and the simulation tool is capable of simulating the system with the available computational resources. The generalized coordinates of the spatial SR are given by the equation:

$$\boldsymbol{q}_{SR3} = \left[x_{SR3} \ y_{SR3} \ z_{SR3} \ \phi_{SR3} \ \theta_{SR3} \ \psi_{SR3} \ q_{SR3,1} \ q_{SR3,2} \ q_{SR3,3} \ q_{SR3,4} \ q_{SR3,5} \ q_{SR3,6} \right]^T, \quad \text{(III.62)}$$

where x_{SR3}, y_{SR3}, and z_{SR3} represent the position of the SR base frame with respect to the inertial frame as follows:

$${}^I_I\boldsymbol{r}_b = [x_{SR3} \ y_{SR3} \ z_{SR3}]^T, \qquad \text{(III.63)}$$

48

ϕ_{SR3}, θ_{SR3} and ψ_{SR3} represent the orientation of the SR base frame with respect to the inertial frames in a XYZ Euler rotation sequence as follows:

$$^I T_b = R_z(\psi_{SR3})R_y(\theta_{SR3})R_x(\phi_{SR3}), \tag{III.64}$$

where $R_z(\psi)$ is a z-axis rotation by angle ψ, $R_y(\theta)$ is a y-axis rotation by angle θ, $R_x(\phi)$ is an x-axis rotation by angle ϕ, and $q_{SR3,1}$ through $q_{SR3,2}$ are the joint variables of the 6 DOF spatial robotic manipulator on the SR.

Similar to the planar SR model, the electro-mechanical equations of motion are derived to the form:

$$\hat{H}_{SR3}(q_{SR3})\ddot{q}_{SR3} + \hat{n}_{SR3}(q_{SR3}, \dot{q}_{SR3}) = u_{SR3}. \tag{III.65}$$

Force-Torque Transducer Model. The FT sensor is modeled in a similar way as that of the planar system, using separate spring-damper models for each axis. In addition to Equations (III.25), (III.26), and (III.27), the following equations are used:

$$F_{z,S}^e = K_z dz + b_z \dot{dz}, \tag{III.66}$$

$$\tau_{\psi,S}^e = K_\psi d\psi + b_\psi \dot{d\psi}, \tag{III.67}$$

and:

$$\tau_{\theta,S}^e = K_\theta d\theta + b_\theta \dot{d\theta}. \tag{III.68}$$

An XYZ Euler rotation sequence is used in the placement of the torsional spring-damper combinations used to measure torque. Due to very large spring constants K_ϕ, K_θ and K_ψ, the deflections in the rotation angles of the FT transducer $d\phi$, $d\theta$ and $d\psi$

are negligible and the rotations are ignored in the controller. However, the rotations are modeled in the dynamic model used for simulation. The transformation from the RP end-effector frame e and the SR body frame b as a result of the FT sensor are given by:

$$_e^e\mathbf{r}_b = [dx \; dy \; dz]^T,$$

$$\text{(III.69)}$$

and:

$$^eT_b = R_z(\psi)R_y(\theta)R_x(\phi).$$

$$\text{(III.70)}$$

The FT sensor is modeled as a set of serial passive joints using the Newton-Euler dynamic modeling approach using Equations (III.25) through (III.27) and Equations (III.66) through (III.68). The equations are implemented by applying the forces and torques computed using the spring-damper models as the inputs corresponding to the passive generalized coordinates defining the sensor.

Robotic Platform Model. For the complete spatial testbed including the RP, FT sensor, and SR, the Newton-Euler dynamic modeling approach outlined by Equations (III.53) through (III.61) are used. The generalized coordinates vector of the spatial testbed $\mathbf{q}_{RP3}$ is given by:

$$\mathbf{q}_{RP3} = [q_{RP3,1} \; q_{RP3,2} \; q_{RP3,3} \; q_{RP3,4} \; q_{RP3,5} \; q_{RP3,6} \; q_{RP3,7} \; dx \; dy \; dz \; d\psi \; d\theta \; d\phi$$

$$\quad\quad q_{SR3,1} \; q_{SR3,2} \; q_{SR3,3} \; q_{SR3,4} \; q_{SR3,5} \; q_{SR3,6}]^T.$$

$$\text{(III.71)}$$

The base frame of the RP is assumed to be stationary and in an inertial frame.

Control Strategies. The control method utilized is a gravity-compensated-zero force-feedback controller. The use of a zero FF controller has been shown in the planar system. To incorporate the gravity compensation term, the resulting gravitational effect of

the SR must be computed.

Real-time Gravity Compensation. The gravitational effect of the SR on the force-torque sensor results in a force vector in the downward direction with magnitude equal to the weight of the SR. This vector only has one component in the inertial coordinate frame z-axis z^I:

$$\mathbf{f}_g^I = \begin{bmatrix} f_{x,g}^I \\ f_{y,g}^I \\ f_{z,g}^I \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \Sigma m_\beta g \end{bmatrix}, \tag{III.72}$$

where $F_{k,g}^I$ is the resultant gravitational force in the k axis, m_β is the mass of the rigid body connected to coordinate frame β and g is the acceleration due to gravity.

The gravitational effect also results in moments about the inertial x^I and y^I axes due to the location of the center of mass of the SR not being directly over the force-torque sensor:

$$\boldsymbol{\tau}_g^I = \begin{bmatrix} \tau_{x,g}^I \\ \tau_{y,g}^I \\ \tau_{z,g}^I \end{bmatrix} = \begin{bmatrix} [0\ 1\ 0]^{T}\ {}_b^I\mathbf{r}_{cm,SR} \\ [1\ 0\ 0]^{T}\ {}_b^I\mathbf{r}_{cm,SR} \\ 0 \end{bmatrix} \Sigma m_\beta g, \tag{III.73}$$

The forward kinematics of the SR and the G-vector parameters, including masses and vector of first mass moments of each link, are required to be known to compute the center of mass of the SR. These parameters are, however, easily measurable unlike, the second mass moments which are required for HIL simulation.

The location of the center of mass of the SR with respect to the body frame mea-

sured in the inertial frame is computed using the equation:

$$
{}_b^I\boldsymbol{r}_{cm,SR} = \frac{1}{\Sigma m_\beta}\Sigma({}_b^I\boldsymbol{r}_\beta m_\beta + {}^I\boldsymbol{T}_\beta{}_\beta^\beta\boldsymbol{\Gamma}), \tag{III.74}
$$

with both summations being across all the bodies connected to coordinate frames represented by β. ${}^I\boldsymbol{T}_\beta$ is the rotation matrix from the inertial frame to frame β and ${}_\beta^\beta\boldsymbol{\Gamma}$ is the vector of first mass moments of frame β measured with respect to frame β.

A true spatial free-floating environment emulation requires both the base of the SR and the manipulator on the SR to respond in a similar manner to control inputs as they would in a free-floating micro-gravity environment. This emulation requires gravity compensation to also be performed on the joints of the SR manipulator.

The strategy applied to perform gravity compensation on the SR manipulator is to compute the forces and torques due to gravity and augment the raw control inputs vector $\boldsymbol{u}$ with the gravitational effect vector $\boldsymbol{\tau}_{g,SR3}(\boldsymbol{q}_{SR3})$. The gravitational effect vector can be computed using the total gravitational potential energy of the system as follows:

$$
\boldsymbol{\tau}_{g,SR3}(\boldsymbol{q}_{SR3}) = \frac{\partial U(\boldsymbol{q}_{SR3})}{\partial \boldsymbol{q}_{SR3}}, \tag{III.75}
$$

where $U_g(\boldsymbol{q}_{SR3})$ is the total gravitational potential energy of the system as a function of the generalized coordinates.

The pure mechanical equations of motion of the spatial SR can be written in the form:

$$
\boldsymbol{H}_{SR3}(\boldsymbol{q}_{SR3})\ddot{\boldsymbol{q}}_{SR3} + \boldsymbol{n}_{SR3}(\boldsymbol{q}_{SR3},\dot{\boldsymbol{q}}_{SR3}) = \boldsymbol{f}_{SR3}, \tag{III.76}
$$

similar to that of the planar SR from Equation (III.21). In Equation (III.76), the vector of Coriolis, centripetal, frictional and gravitational forces $\boldsymbol{n}_{SR3}(\boldsymbol{q}_{SR3},\dot{\boldsymbol{q}}_{SR3})$ can be separated into the vector of Coriolis, centripetal and frictional forces and torques $\boldsymbol{\tau}_{c,SR3}(\boldsymbol{q}_{SR3},\dot{\boldsymbol{q}}_{SR3})$

that are functions of both the generalized coordinates and the generalized velocities, and the vector of gravitational forces and torques $\boldsymbol{\tau}_{g,SR3}(\boldsymbol{q}_{SR3})$ that are only functions of the generalized coordinates, resulting in:

$$\boldsymbol{H}_{SR3}(\boldsymbol{q}_{SR3})\ddot{\boldsymbol{q}}_{SR3} + \boldsymbol{\tau}_{c,SR3}(\boldsymbol{q}_{SR3},\dot{\boldsymbol{q}}_{SR3}) + \boldsymbol{\tau}_{g,SR3}(\boldsymbol{q}_{SR3}) = \boldsymbol{f}_{SR3}. \qquad \text{(III.77)}$$

Comparing equivalent physical systems on Earth (left) and in free-floating environment (right), results in:

$$\boldsymbol{H}_{SR3}(\boldsymbol{q}_{SR3})\ddot{\boldsymbol{q}}_{SR3} + \boldsymbol{\tau}_{c,SR3}(\boldsymbol{q}_{SR3},\dot{\boldsymbol{q}}_{SR3}) + \boldsymbol{\tau}_{g,SR3}(\boldsymbol{q}_{SR3}) - \boldsymbol{f}_{SR3} =$$
$$\boldsymbol{H}_{SR3}(\boldsymbol{q}_{SR3})\ddot{\boldsymbol{q}}_{SR3} + \boldsymbol{\tau}_{c,SR3}(\boldsymbol{q}_{SR3},\dot{\boldsymbol{q}}_{SR3}) - \boldsymbol{f}_{0,SR3}$$
$$\implies \boldsymbol{f}_{SR3} = \boldsymbol{f}_{0,SR3} + \boldsymbol{\tau}_{g,SR3}(\boldsymbol{q}_{SR3}) \qquad \text{(III.78)}$$

Equation (III.78) yields an augmented generalized forces and torques vector $\boldsymbol{f}_{SR3}$ applied to the joints with $\boldsymbol{f}_{0,SR3}$ being the generalized forces and torques that would be applied to the joints in zero-gravity.

The electro-mechanical equations of motion for the spatial SR are also attained using the same procedure used for the planar SR, which allows Equation (III.13) to be re-written as:

$$\boldsymbol{u} = \tilde{\boldsymbol{J}}_m \ddot{\boldsymbol{q}}_{SR3} + \tilde{\boldsymbol{n}}(\dot{\boldsymbol{q}}_{SR3}) + \boldsymbol{R}_m \boldsymbol{f}_{SR3}$$
$$\implies \boldsymbol{f}_{SR3} = \boldsymbol{R}_m^{-1} \left(\boldsymbol{u} - \tilde{\boldsymbol{J}}_m \ddot{\boldsymbol{q}}_{SR3} - \tilde{\boldsymbol{n}}(\dot{\boldsymbol{q}}_{SR3}) \right). \qquad \text{(III.79)}$$

The above Equation can also be similarly applied to the free-floating scenario with the zero-gravity input vector $\boldsymbol{u}_0$ and the zero-gravity generalized forces and torques

$\boldsymbol{f}_{0,SR3}$:

$$\boldsymbol{u}_0 = \tilde{\boldsymbol{J}}_m \ddot{\boldsymbol{q}}_{SR3} + \tilde{\boldsymbol{n}}(\dot{\boldsymbol{q}}_{SR3}) + \boldsymbol{R}_m \boldsymbol{f}_{0,SR3}$$

$$\implies \boldsymbol{f}_{0,SR3} = \boldsymbol{R}_m^{-1}\left(\boldsymbol{u}_0 - \tilde{\boldsymbol{J}}_m \ddot{\boldsymbol{q}}_{SR3} - \tilde{\boldsymbol{n}}(\dot{\boldsymbol{q}}_{SR3})\right). \tag{III.80}$$

Substitution Equations (III.79) and (III.80) into Equation (III.78) produces a simple relationship between the augmented control inputs necessary for the system on Earth and the raw control inputs to be applied to the system in zero-gravity:

$$\boldsymbol{u} = \boldsymbol{u}_0 + \boldsymbol{R}_m \boldsymbol{\tau}_{g,SR3}(\boldsymbol{q}_{SR3}). \tag{III.81}$$

The augmented control input $\boldsymbol{u}$ is applied to the joints of the SR during simulation.

Compensated Force-Feedback Control. To perform gravity-compensated-zero force-feedback control, the end-effector of the RP is moved in the respective directions to drive the vectors of forces and torques measured by the force-torque sensor at the interface to the vectors of expected gravitational forces and torques computed in real-time, with both the control parameters and control goal measured in the same reference frame.

The vectors of forces and torques measured by the sensor in the RP end-effector frame e is transformed into the inertial frame as follows:

$$\begin{bmatrix} \boldsymbol{f}_S^I \\ \boldsymbol{\tau}_S^I \end{bmatrix} = \begin{bmatrix} {}^I\boldsymbol{T}_e(\boldsymbol{q}_{RP}) & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & {}^I\boldsymbol{T}_e(\boldsymbol{q}_{RP}) \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_S^e \\ \boldsymbol{\tau}_S^e \end{bmatrix}, \tag{III.82}$$

where ${}^I\boldsymbol{T}_e(\boldsymbol{\gamma}_{RP})$ is a transformation matrix from the end-effector frame to the inertial frame, which is a function of the vector of the joint variables of the RP and represents the forward kinematics. $\boldsymbol{0}_{3x3}$ is a (3x3) zero matrix.

54

The control goal is given as:

$$
\begin{bmatrix} \boldsymbol{f}_S^I \\ \boldsymbol{\tau}_S^I \end{bmatrix} \rightarrow \begin{bmatrix} \boldsymbol{f}_g^I \\ \boldsymbol{\tau}_g^I \end{bmatrix}
$$

$$
\implies \begin{bmatrix} \boldsymbol{f}_S^I \\ \boldsymbol{\tau}_S^I \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 0 \\ \Sigma m_\beta g \\ [0\ 1\ 0]^T \Sigma ({}_b^I \boldsymbol{r}_\beta m_\beta + {}^I\boldsymbol{T}_\beta {}_\beta^\beta \boldsymbol{\Gamma}) g \\ [1\ 0\ 0]^T \Sigma ({}_b^I \boldsymbol{r}_\beta m_\beta + {}^I\boldsymbol{T}_\beta {}_\beta^\beta \boldsymbol{\Gamma}) g \\ 0 \end{bmatrix}. \tag{III.83}
$$

The outer control loop used for the simulations utilizes a PID controller to generate desired translational and rotational accelerations in the inertial frame.

The transpose Jacobian of the 7-DOF RP is used to compute the desired torques in each of the 7-DOF joint in order to accelerate the RP end effector in the desired accelerations generated by the PID controller. It can be shown using the principle of virtual work that the transpose Jacobian of a robot can be used to map task space forces and torques into the joint forces and torques.

The Jacobian is constructed sequentially for each link using Equations (III.84) and (III.85).

$$
\boldsymbol{J} = \begin{bmatrix} J_{P1} & & J_{P7} \\ & \cdots & \\ J_{O1} & & J_{O7} \end{bmatrix} \tag{III.84}
$$

$$
\begin{bmatrix} J_{Pi} \\ J_{Oi} \end{bmatrix} = \begin{bmatrix} {}^{I}\boldsymbol{T}_{i-1}(:,3) \times \left({}^{I}_{I}\boldsymbol{r}_{7} - {}^{I}_{I}\boldsymbol{r}_{i-1} \right) \\ {}^{I}\boldsymbol{T}_{i-1}(:,3) \end{bmatrix} \tag{III.85}
$$

Figure 3.14 is a block diagram of the complete spatial free-floating emulation testbed controller:

Figure 3.14: Block Diagram of Complete Spatial Free-floating Emulation Testbed Controller

Simulation Environment Improvements. Various improvements were made to the simulation tool previously developed for planar conditions to be able to simulate the spatial free-floating environment testbed. The major modification was the addition

of the capability to numerically simulate a dynamic model given by the Newton-Euler equations. Other minor modifications included improvements to the graphics engine integration and controller helper functions for controllers written using Lua scripts.

Figure 3.15 is a screenshot of the simulation tool simulating the spatial testbed.
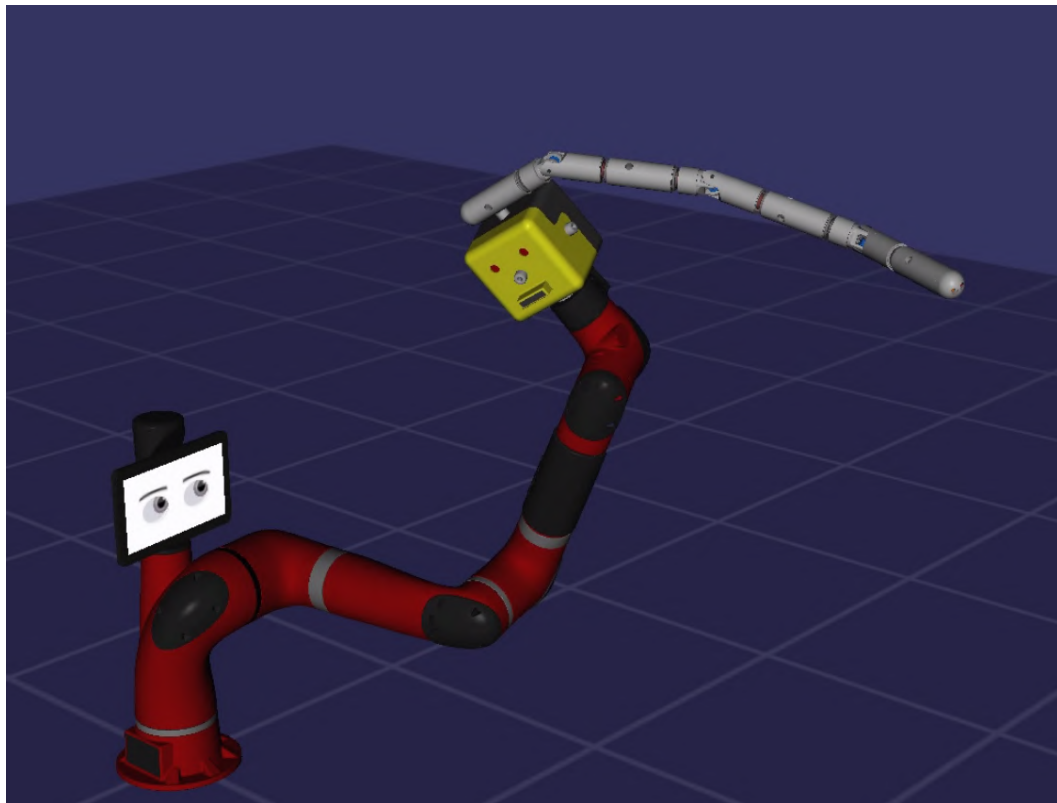
Figure 3.15: Screenshot of robot dynamics simulator tool with spatial free-floating environment testbed.

Newton-Euler Recursive Algorithm. The same dynamics integration module for the planar system simulation is still used to run the dynamics simulation of the robot; however, the Newton-Euler recursive algorithm is implemented to compute the generalized accelerations as opposed to solving a linear system of equations given by the closed-

form dynamic model.

The implementation of the Newton-Euler recursive algorithm requires the linear and angular velocities and accelerations of each link, linear acceleration of the center of mass of each link and angular acceleration of each actuator rotor given by Equations (III.53) through (III.57) to be computed from the base frame up to the end-effector in a so-called forward recursion, also known as an outbound pass. On completion of the forward recursion, the balance forces and torques given by Equations (III.58) and (III.59) are computed starting from the end-effector down to the base in a so-called backward recursion, also known as an inbound pass. The joint forces and torques are also computed simultaneously with the backward recursion.

A function NE0() is created with the input arguments as the generalized coordinates, generalized velocities, generalized accelerations, gravitational acceleration vector, base angular velocity, base angular acceleration and base linear acceleration. The function performs the steps of the Newton-Euler recursive algorithm and returns the joint forces and torques in the form of a vector that corresponds to the vector of generalized forces and torques.

Various calls of the NE0() function are used to solve the direct dynamics of the system. The system mass matrix $\hat{H}$ is constructed by combining the column vector outputs of the NE0() function called with the current generalized coordinate values, zero generalized velocities and vector e_i for the generalized accelerations. The gravitational acceleration vector input argument is passed as a null column vector with three rows. e_i is a column vector of the same size as the generalized coordinates vector with 1 in the ith row and zeros on all other rows. The index i also corresponds to the column of the system

mass matrix being constructed:

$$\hat{H} = \left[\text{NE0}\left(\boldsymbol{q}, \boldsymbol{0}, \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}, \boldsymbol{0}_3 \right) \quad \text{NE0}\left(\boldsymbol{q}, \boldsymbol{0}, \begin{bmatrix} 0 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}, \boldsymbol{0}_3 \right) \quad \dots \text{NE0}\left(\boldsymbol{q}, \boldsymbol{0}, \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}, \boldsymbol{0}_3 \right) \right], \quad \text{(III.86)}$$

where $\boldsymbol{0}$ is a null vector of the size of the generalized coordinates vector, and $\boldsymbol{0}_3$ is a (3x1) zero vector.

The vector of Coriolis, centripetal, gravitational and frictional forces $\hat{\boldsymbol{n}}$ is constructed using the NE0() function with input arguments as the generalized coordinates, generalized velocities, a null vector as the generalized accelerations input and the gravitational acceleration vector:

$$\hat{\boldsymbol{n}} = \text{NE0}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{0}, \boldsymbol{g} \right) \quad \text{(III.87)}$$

Using the system mass matrix and vector of Coriolis, centripetal and frictional forces, the generalized accelerations are solved for in a similar fashion as that of solving a closed-form system as shown in Equation (III.42).

<center>**Chapter IV**</center>

<center>**Results**</center>

Planar System Simulation Results

Two models of the SR are used for the simulations; the first model is considered to be the Actual Space Robot (ASR), and the second is considered to be the Modeled Space Robot (MSR). The MSR model incorporates a 10% increase in the mass of the spacecraft, of the momentum wheel, and of each link.

The SR is tested with fifty-five different sets of excitation inputs comprising of sinusoidal inputs with frequencies ranging from 0.5 to 4 (rad/s) for the robotic arm, various sinusoidal inputs for the thrusters and position and joint tracking controller tests in response to zeroed initial conditions. Both the ASR and MSR are simulated in free-floating conditions under each of these test sets in order to generate the ASR trajectories (ATs) and the MSR trajectories (MTs). The ATs are used as the reference trajectories hereafter. The dynamical, electrical and controller parameters used for the simulations are available online at [38].

Hardware-in-the-Loop Simulation Verification. In order to verify the proper implementation of the HIL trajectory-tracking controller, the ASR is mounted on the RP with the controller tracking the AT (HIL-ASR-AT). The HIL trajectory-tracking controller is then utilized to track the MT with the ASR mounted on the RP (HIL-ASR-MT). This scenario emulates a realistic HIL simulation set-up where the MT would have uncertainties in the measured dynamic parameters. Finally, the force-feedback controller is utilized with the ASR mounted on the RP (FF-ASR).
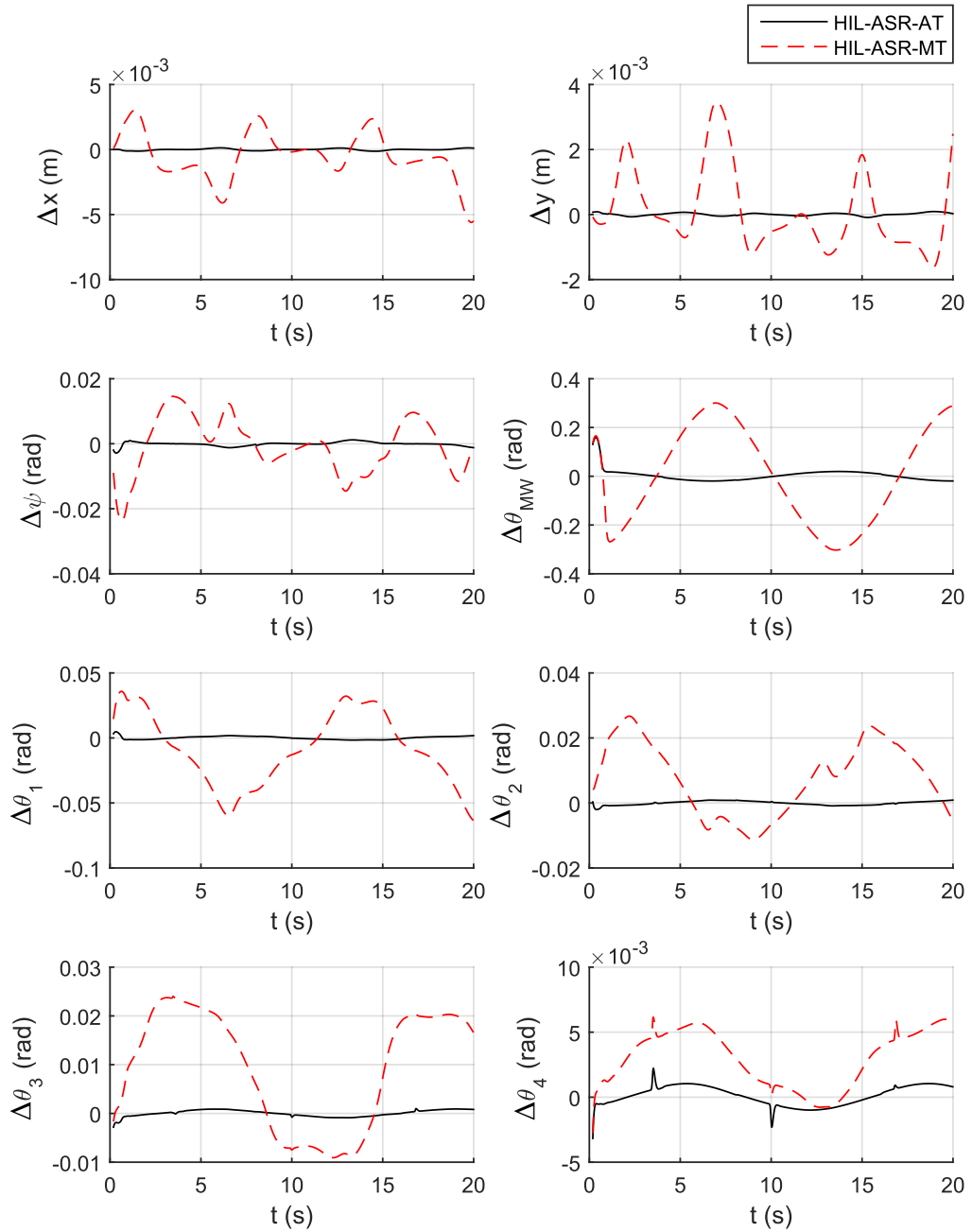
<center>60</center>

Figure 4.1: Error Terms Associated with HIL Simulated-trajectory Tracking Controller Tracking the MT and the AT for a Low Frequency Test Set

Figure 4.1 depicts plots of the error terms versus time for the HIL-ASR-MT and HIL-ASR-AT scenarios for one of the test sets. As seen from the plots, the errors generated with the HIL-ASR-AT scenario in reference to AT are significantly smaller than the errors generated with the HIL-ASR-MT scenario in reference to AT. The RMS position error of the HIL-ASR-AT scenario, $\Delta r_{RMS} = \sqrt{\Delta x_{RMS}^2 + \Delta y_{RMS}^2}$, is found to be 99.1% less than the RMS position error of the HIL-ASR-MT scenario for the first test set. The RMS orientation error ($\Delta \psi_{RMS}$) of the HIL-ASR-AT scenario is found to be 93.3% less than that of the HIL-ASR-MT scenario. The RMS of momentum wheel joint angle errors of the HIL-ASR-AT scenario, $\Delta \theta_{RMS} = \sqrt{\sum_i \Delta \theta_{i,RMS}^2}$, are found to be 93.2% less than that of the HIL-ASR-MT scenario.

These results show that most of the errors in the HIL simulation are not due to controller errors. The majority of errors can be attributed to the difference in dynamic parameters between the ASR and MSR used to generate AT and MT, respectively.

Space Robot Trajectory Error Analysis. Error metrics are defined as the difference between the AT and the trajectories generated by the three scenarios: HIL-ASR-MT, HIL-ASR-AT and FF-ASR.

Figure 4.2 is a plot of the error terms generated by the HIL-ASR-MT scenario and the FF-ASR scenario in reference to AT.

As seen in the plots, the force-feedback control shows significantly less trajectory and joint angle errors as compared to the HIL-ASR-MT scenario for the first test set. The RMS position error of the FF-ASR scenario is 94.2% less than that of the HIL-ASR-MT scenario. Additionally, the RMS orientation error is 96.65% less and the RMS momentum wheel and joint angle errors are 99.93% less than that of the HIL-ASR-MT scenario. It can also be observed that the force-feedback controller out-performs the HIL-ASR-AT scenario in terms of orientation and joint angle errors for the first control input set; how-

ever, this is not the case at higher frequency control inputs as will be discussed in the next section.

Figure 4.2: Error Terms Associated with Both the HIL Simulated-trajectory Tracking Controller Tracking the MT and the FF Controller for a Low Frequency Test Set

The plot in Figure 4.3 compares the position, orientation, momentum wheel and joint angle errors of the FF-ASR scenario with the HIL-ASR-MT scenario for a test set with higher frequencies. It can be seen that the errors are still much larger with the HIL-ASR-MT scenario, but there is a noticeable amount of position and orientation drift in the FF-ASR scenario. This drift is a disadvantage of the force-feedback control method. However, in practice, the frequencies of the system excitation inputs for an SR berthing scenario are expected to be relatively low. Momentum wheel and manipulator joint angle drifts are not as prominent with the highest frequency excitation input sets simulated.
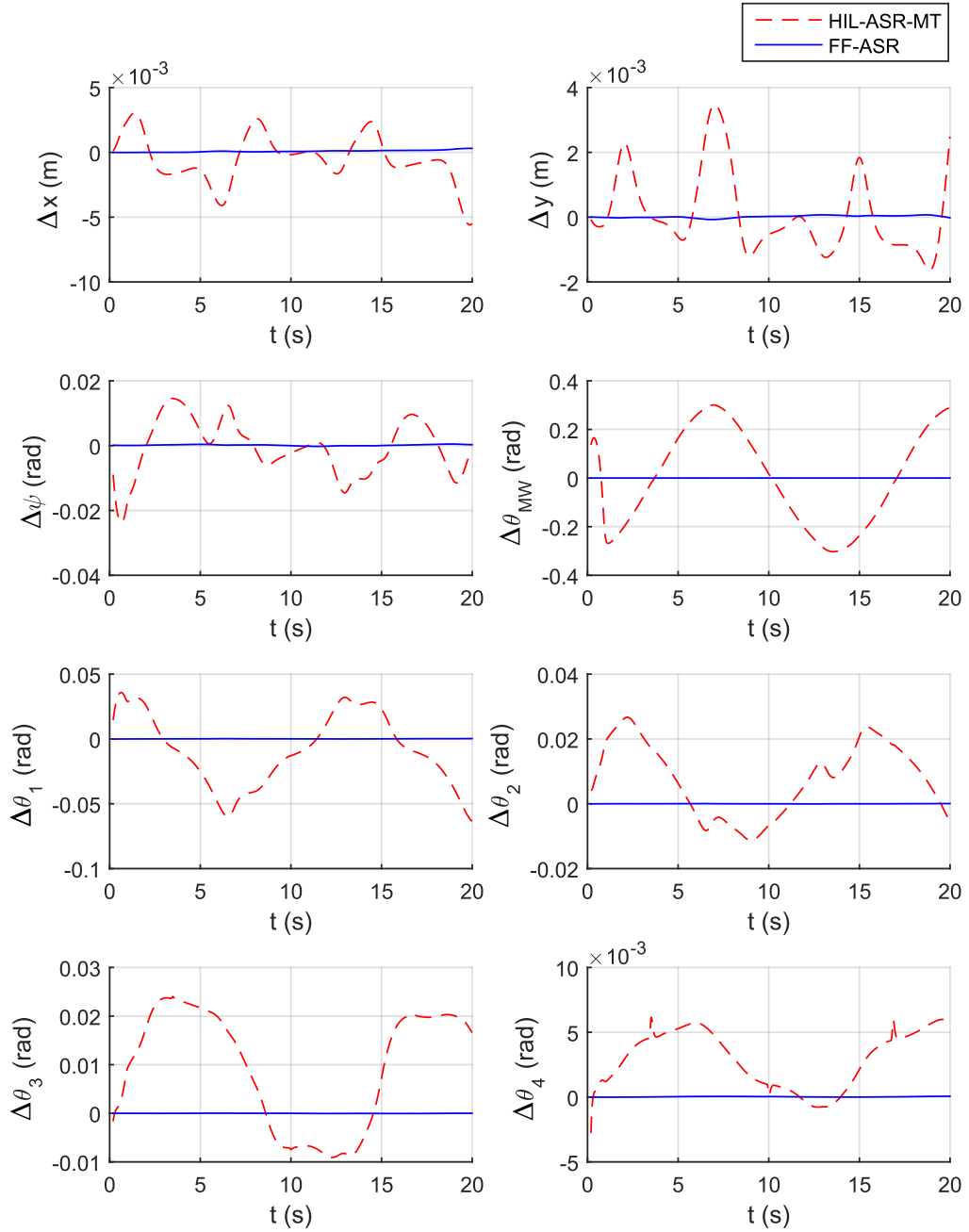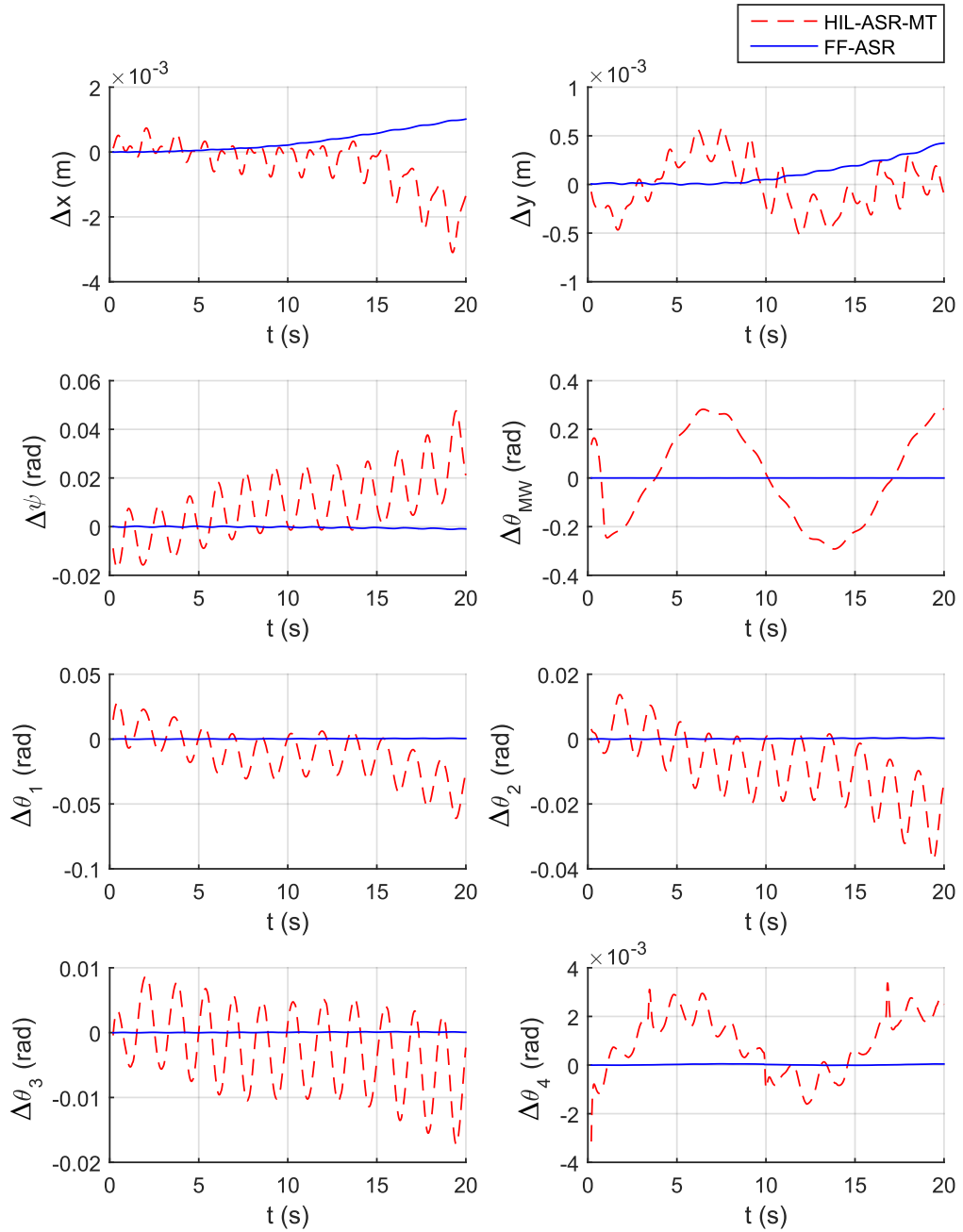
Figure 4.3: Error Terms Associated with Both the HIL Simulated-trajectory Tracking Controller Tracking the MT and the FF Controller for a High Frequency Test Set

Momentum Conservation Error Analysis. The linear momentums in the x^I and y^I axes, and angular momentum about the z^I axis are computed as additional error metrics since a perfectly isolated system without any external forces will conserve linear and angular momentum. The linear momentum vector is computed using the equation:

$$\boldsymbol{p}(t) = \sum_{k \in K} m_k {}_I^I \boldsymbol{r}_k(t) + {}^I \dot{\boldsymbol{T}}_k(t) {}_k^k \boldsymbol{q}, \tag{IV.1}$$

where K is a set of the six bodies comprising the SR, the spacecraft, the momentum wheel, and the four manipulator links. A frame is defined for each body. m_k is the mass of body k, ${}_I^I \boldsymbol{r}_k$ is the vector from the inertial frame to the origin of the frame k measured in the inertial frame, ${}^I \boldsymbol{T}_k$ is the transformation matrix from the inertial frame to frame k, and ${}_k^k \boldsymbol{q}$ is the vector of first mass moments of body k measured in frame k. Only the x and y components of the linear momentum are used for comparison.

The angular momentum vector is calculated using the equation:

$$\boldsymbol{h}(t) = \sum_{k \in K} \left({}_I^I \boldsymbol{r}_k(t) \times \boldsymbol{p}_k(t) - {}_I^I \dot{\boldsymbol{r}}_k(t) \times {}^I \boldsymbol{T}_k(t) {}_k^k \boldsymbol{q} + {}^I \boldsymbol{T}_k(t) {}_k^k \boldsymbol{J} {}_k^k \boldsymbol{\omega}^I(t) \right), \tag{IV.2}$$

where ${}_k^k \boldsymbol{J}$ is the inertia tensor of body k measured with respect to frame k, and ${}_k^k \boldsymbol{\omega}^I(t)$ is the angular velocity vector of body k with respect to the inertial frame measured relative to the k frame. Only the z component of the angular momentum vector is used for comparison.

In order to study the ability of the system to conserve momentum over time, Equations (IV.1) and (IV.2) are used to compute linear and angular momentum.

The plots in Figure 4.4 depict the computed linear and angular momentum components with a low-frequency test set. The FF-ASR system errors are primarily due to high frequency transients in the control response and can be minimized by further improving

the force-feedback controller. A 43.0% reduction of the linear momentum RMS error can be observed when FF-ASR results are compared to HIL-ASR-MT results. However, a 5.9% increase in the angular momentum RMS error is observed.
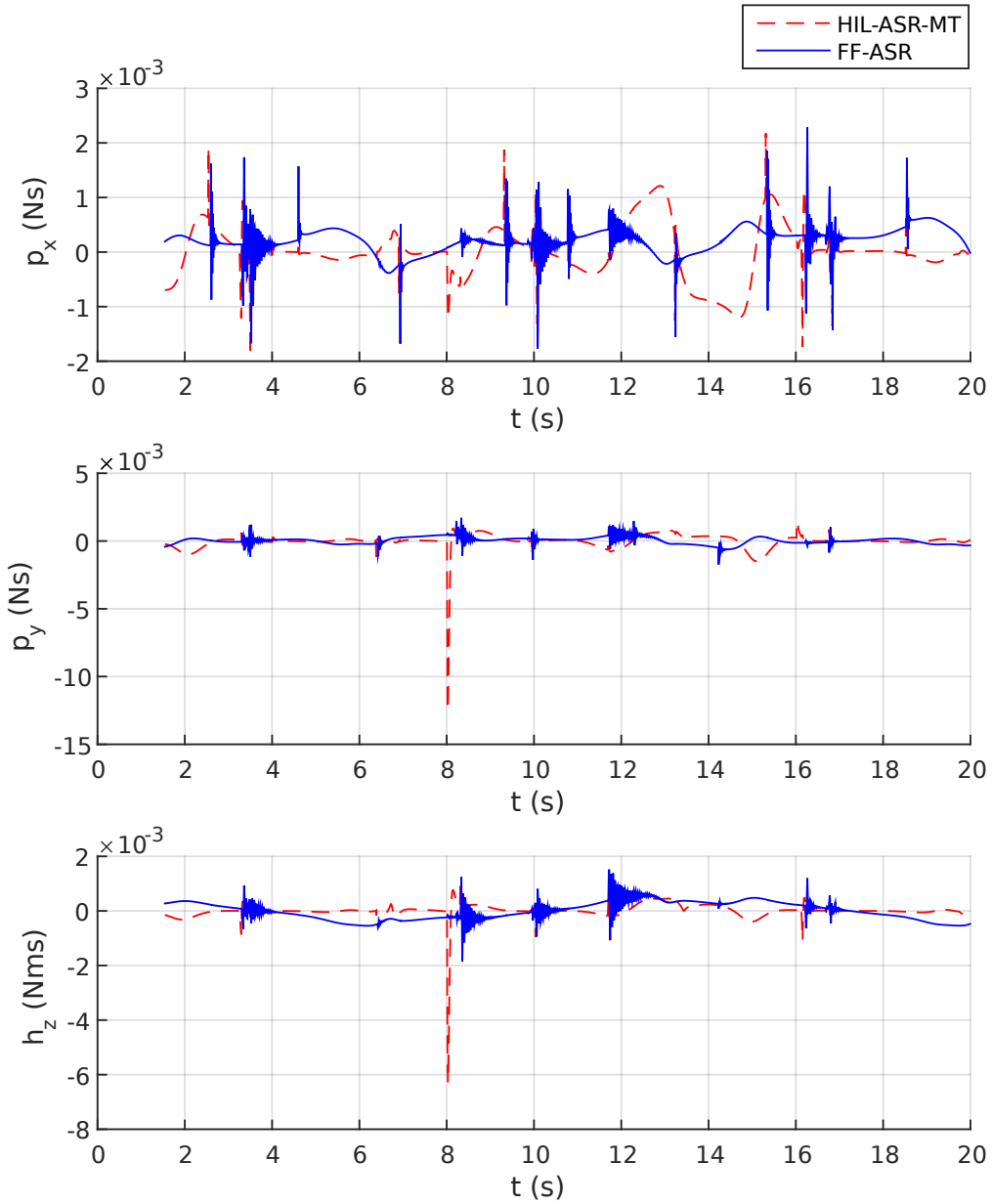
Figure 4.4: Computed Linear and Angular Space Robot Momentum for Both the HIL Simulated-trajectory Tracking Controller Tracking the MT and the FF Controller for a Low Frequency Test Set

Performance Analysis over Multiple System Excitations. Figures 4.5 and 4.6 show the error distribution in momentum components and trajectory over the tested input sets.

In general, the force-feedback control method is able to more accurately recreate the free-floating trajectory. On average using this method, Δr_{RMS} is 90.79% less, $\Delta \psi_{RMS}$ is 96.33% less, and $\Delta \theta_{RMS}$ is 99.77% less than when using the HIL-ASR-MT scenario.
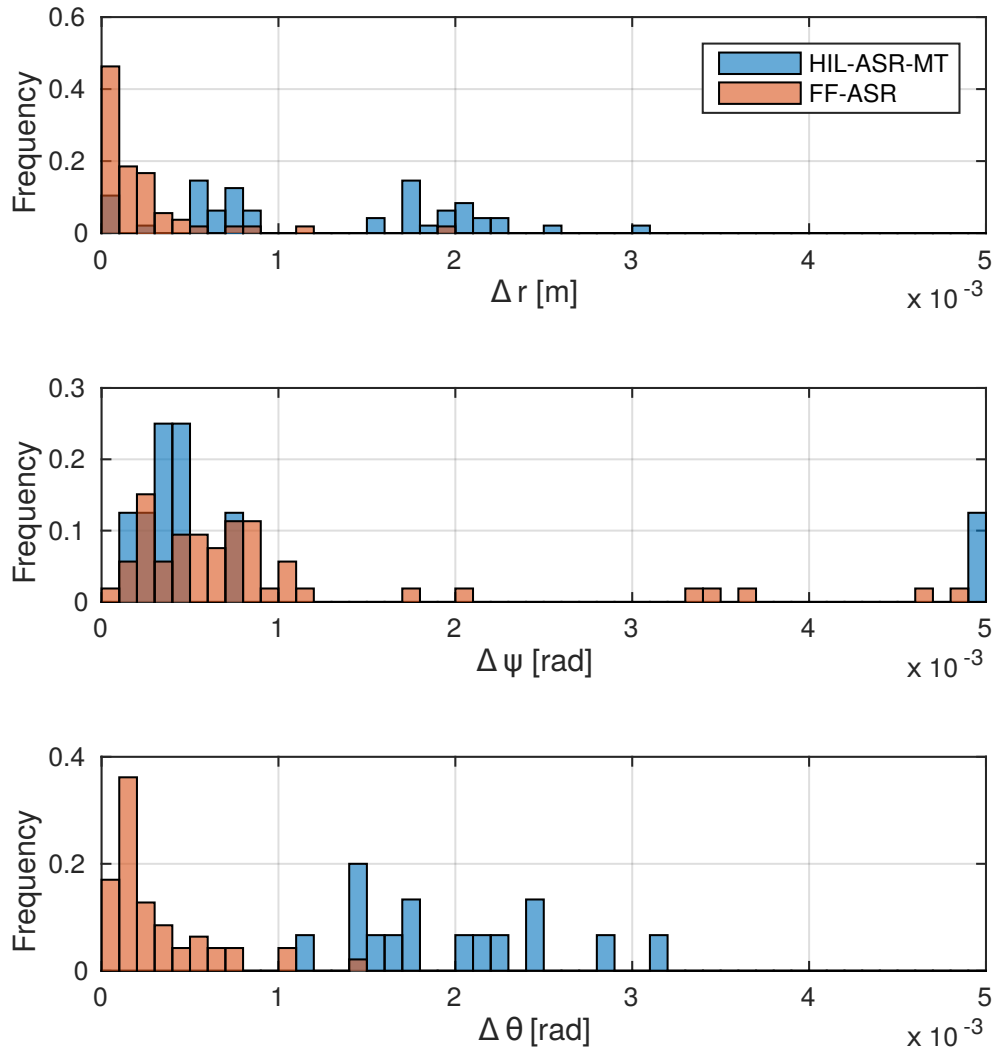
Figure 4.5: Histograms Depicting the Relative Frequencies of Trajectory and Joint Angle Error among Multiple System Excitation Input Sets
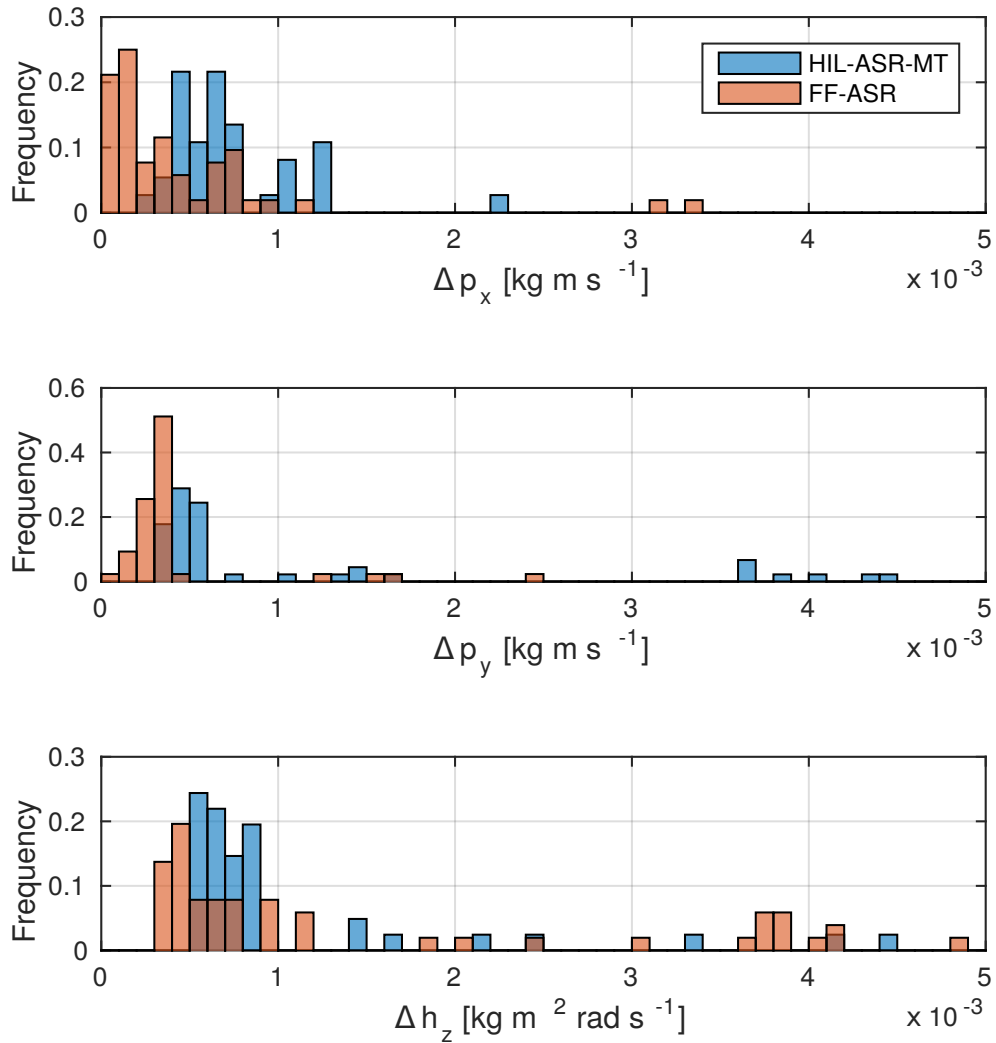
Figure 4.6: Histograms Depicting the Relative Frequencies of Linear and Angular Momentum Error Among Multiple System Excitation Input Sets

Similarly, the FF-ASR scenario out-performs the HIL-ASR-MT scenario when conserving linear and angular momentum over the set of tests. On average, the force-feedback controller's $\Delta p_{x,RMS}$ is 95.26% less, $\Delta p_{y,RMS}$ is 77.91% less and $\Delta h_{z,RMS}$ is 75.16% less than that of the HIL simulation tracking controller. Since the zeroed initial conditions configure the SR so that the links are aligned with the x^I axis, the sinusoidal system ex-

citation inputs cause higher frequency motion along the y^I axis than the x^I axis. This is a possible cause for larger errors in $\Delta p_{y,RMS}$ than $\Delta p_{x,RMS}$ for the force-feedback controller.

The complete simulation results with RMS error for multiple system excitation input sets are also listed in Tables 4.1 and 4.2.

Table 4.1 consists of the RMS momentum and trajectory errors for the HIL simulation runs and Table 4.2 consists of the RMS momentum and trajectory errors for the FF simulation runs.

Table 4.1: Momentum and Generalized Coordinate RMS Errors for 55 Test Sets Using the HIL Simulated-trajectory Tracking Controller.

Set	Momentum RMS Error			Generalized Coordinate RMS Error		
	$\Delta p_{x,RMS}$	$\Delta p_{y,RMS}$	$\Delta h_{z,RMS}$	Δr_{RMS}	$\Delta \psi_{RMS}$	$\Delta \theta_{RMS}$
	$(kg\ m/s)$	$(kg\ m/s)$	$(kg\ m^2\ rad/s)$	(m)	(rad)	(rad)
1	6.96E-04	5.61E-04	8.67E-04	2.25E-03	8.59E-03	8.96E-02
2	6.84E-04	5.37E-04	8.73E-04	2.20E-03	8.71E-03	8.97E-02
3	6.37E-04	3.66E-04	8.83E-04	2.05E-03	1.71E-02	8.98E-02
4	6.30E-04	3.60E-04	8.86E-04	2.07E-03	1.88E-02	8.99E-02
5	5.80E-04	3.64E-04	8.54E-04	1.94E-03	1.35E-02	8.91E-02
6	5.80E-04	3.57E-04	8.61E-04	1.94E-03	1.55E-02	8.92E-02
7	5.78E-04	4.03E-04	8.71E-04	2.07E-03	2.36E-02	8.91E-02
8	5.80E-04	4.05E-04	8.68E-04	2.12E-03	2.51E-02	8.91E-02
9	7.34E-04	5.05E-04	7.52E-04	7.37E-04	5.43E-03	8.56E-02
10	7.48E-04	5.21E-04	7.64E-04	7.09E-04	5.72E-03	8.56E-02
11	7.74E-04	4.90E-04	7.49E-04	7.09E-04	8.48E-03	8.57E-02
12	7.87E-04	5.04E-04	7.54E-04	7.23E-04	8.81E-03	8.57E-02
13	1.02E-03	4.92E-04	7.26E-04	7.30E-04	1.27E-02	8.54E-02
14	1.02E-03	5.19E-04	7.38E-04	7.36E-04	1.32E-02	8.54E-02
15	1.24E-03	4.64E-04	6.94E-04	8.81E-04	1.64E-02	8.53E-02
16	1.28E-03	4.73E-04	6.94E-04	8.45E-04	1.62E-02	8.53E-02
17	4.99E-04	4.96E-04	5.62E-04	1.78E-03	8.52E-03	6.46E-02
18	4.97E-04	5.07E-04	5.61E-04	1.75E-03	8.98E-03	6.46E-02
19	4.71E-04	5.11E-04	5.65E-04	1.70E-03	1.73E-02	6.44E-02
20	4.74E-04	5.04E-04	5.64E-04	1.73E-03	1.88E-02	6.44E-02
21	4.29E-04	4.66E-04	5.61E-04	1.57E-03	1.36E-02	6.43E-02
22	4.43E-04	4.68E-04	5.59E-04	1.58E-03	1.52E-02	6.43E-02
23	4.46E-04	3.62E-04	5.69E-04	1.71E-03	2.17E-02	6.41E-02
24	4.49E-04	3.44E-04	5.66E-04	1.73E-03	2.27E-02	6.41E-02
25	6.58E-04	5.43E-04	5.85E-04	5.22E-04	4.91E-03	6.50E-02
26	6.69E-04	5.57E-04	5.98E-04	5.33E-04	5.07E-03	6.50E-02
27	6.88E-04	4.79E-04	6.25E-04	6.49E-04	6.40E-03	6.48E-02
28	7.06E-04	4.79E-04	6.28E-04	6.79E-04	6.68E-03	6.48E-02
29	9.97E-04	4.03E-04	6.12E-04	5.10E-04	8.47E-03	6.56E-02
30	1.00E-03	4.20E-04	6.28E-04	5.22E-04	8.77E-03	6.56E-02
31	1.21E-03	3.51E-04	6.31E-04	5.59E-04	1.13E-02	6.53E-02
32	1.25E-03	3.62E-04	6.29E-04	5.58E-04	1.16E-02	6.53E-02
33	6.17E-04	1.03E-03	1.13E-01	4.29E-02	2.84E-01	9.22E-03
34	2.33E-04	1.37E-03	2.55E-02	1.90E-03	9.72E-03	3.16E-03
35	3.47E-04	1.50E-03	1.05E-02	3.01E-03	2.02E-02	2.84E-03
36	3.09E-04	1.40E-03	8.92E-03	2.56E-03	1.71E-02	2.47E-03
37	2.85E-02	2.15E-02	9.31E-03	1.64E-02	1.08E-02	1.73E-03
38	2.41E-02	2.25E-02	6.42E-03	1.42E-02	7.02E-03	1.20E-03
39	2.17E-02	7.16E-04	6.17E-04	5.02E-03	2.46E-04	1.62E-03
40	5.23E-02	6.31E-02	2.02E-02	2.91E-02	1.23E-02	1.50E-03
41	5.24E-02	1.11E-02	1.55E-02	5.34E-03	9.68E-03	2.13E-03
42	2.54E-02	4.18E-02	1.52E-02	1.86E-03	1.06E-02	2.09E-03
43	7.12E-03	4.09E-03	2.13E-03	3.21E-05	3.04E-04	1.46E-03
44	7.01E-03	4.36E-03	2.42E-03	3.02E-05	3.11E-04	1.50E-03
45	1.06E-02	5.46E-03	3.35E-03	2.34E-04	7.39E-04	1.79E-03
46	3.46E-02	3.03E-02	1.03E-02	2.00E-03	9.57E-03	2.50E-03
47	7.35E-03	3.63E-03	1.49E-03	3.42E-05	4.48E-04	2.26E-03
48	7.33E-03	3.69E-03	1.61E-03	1.80E-05	1.53E-04	1.48E-03
49	5.38E-01	3.37E-01	1.81E-01	9.90E-02	2.52E+00	4.98E+01
50	7.44E-03	3.86E-03	4.44E-03	5.30E-04	2.28E-02	4.06E-01
51	9.35E-03	4.46E-03	7.14E-03	8.92E-04	4.11E-02	7.53E-01
52	3.81E-02	9.74E-03	1.49E-02	2.29E-03	1.36E-01	2.47E+00
53	7.35E-03	3.61E-03	1.47E-03	4.17E-05	4.69E-04	2.63E-02
54	2.26E-03	1.69E-03	4.19E-03	6.70E-04	2.31E-02	4.35E-01
55	1.76E-02	1.49E-02	2.14E-02	1.76E-03	8.42E-02	1.52E+00
AVG	1.68E-02	1.10E-02	9.16E-03	4.92E-03	6.60E-02	1.05E+00
STD	7.21E-02	4.57E-02	2.82E-02	1.46E-02	3.37E-01	6.65E+00

Table 4.2: Momentum and Generalized Coordinate RMS Errors for 55 Test Sets Using the FF Controller.

Set	Momentum RMS Error			Generalized Coordinate RMS Error		
	$\Delta p_{x,RMS}$	$\Delta p_{y,RMS}$	$\Delta h_{z,RMS}$	Δr_{RMS}	$\Delta \psi_{RMS}$	$\Delta \theta_{RMS}$
	$(kg\ m/s)$	$(kg\ m/s)$	$(kg\ m^2\ rad/s)$	(m)	(rad)	(rad)
1	1.25E-04	2.79E-04	3.82E-04	7.93E-05	8.12E-04	1.23E-04
2	1.22E-04	2.79E-04	3.78E-04	7.52E-05	7.49E-04	1.05E-04
3	1.03E-04	2.78E-04	3.67E-04	6.92E-05	6.39E-04	9.15E-05
4	9.30E-05	2.80E-04	3.71E-04	6.86E-05	6.45E-04	8.19E-05
5	8.14E-05	2.80E-04	3.68E-04	5.62E-05	5.10E-04	7.36E-05
6	7.98E-05	2.81E-04	3.75E-04	5.55E-05	5.15E-04	7.08E-05
7	8.61E-05	2.89E-04	4.26E-04	5.81E-05	4.95E-04	8.41E-05
8	8.59E-05	2.88E-04	4.38E-04	5.79E-05	5.03E-04	1.09E-04
9	2.74E-04	3.00E-04	6.32E-04	1.59E-04	2.42E-04	1.40E-04
10	2.69E-04	3.03E-04	6.39E-04	1.57E-04	2.58E-04	1.56E-04
11	2.74E-04	3.06E-04	7.65E-04	1.68E-04	4.31E-04	2.02E-04
12	2.67E-04	3.08E-04	7.80E-04	1.66E-04	4.62E-04	2.15E-04
13	4.01E-04	3.33E-04	9.55E-04	2.63E-04	8.37E-04	3.98E-04
14	3.80E-04	3.33E-04	9.69E-04	2.55E-04	8.66E-04	4.00E-04
15	3.91E-04	3.59E-04	1.10E-03	2.77E-04	1.07E-03	5.10E-04
16	3.73E-04	3.58E-04	1.12E-03	2.70E-04	1.18E-03	5.68E-04
17	1.04E-04	2.90E-04	6.36E-04	8.07E-05	1.05E-03	1.67E-04
18	1.04E-04	2.92E-04	6.25E-04	8.21E-05	1.05E-03	1.64E-04
19	9.61E-05	3.02E-04	5.62E-04	9.00E-05	8.93E-04	1.37E-04
20	9.26E-05	3.02E-04	5.53E-04	8.99E-05	8.89E-04	1.32E-04
21	1.02E-04	3.01E-04	4.95E-04	8.70E-05	7.93E-04	1.29E-04
22	1.09E-04	3.01E-04	4.87E-04	9.02E-05	7.86E-04	1.26E-04
23	1.15E-04	3.00E-04	4.52E-04	9.71E-05	7.04E-04	1.13E-04
24	1.18E-04	2.99E-04	4.49E-04	9.90E-05	6.98E-04	1.18E-04
25	3.22E-04	3.02E-04	4.55E-04	1.89E-04	2.45E-04	1.09E-04
26	3.29E-04	3.00E-04	4.61E-04	1.93E-04	2.49E-04	1.10E-04
27	4.87E-04	3.03E-04	5.65E-04	2.71E-04	2.33E-04	1.28E-04
28	4.94E-04	3.02E-04	5.80E-04	2.75E-04	2.44E-04	1.18E-04
29	7.01E-04	3.32E-04	7.52E-04	3.97E-04	4.47E-04	2.38E-04
30	6.96E-04	3.30E-04	7.67E-04	3.96E-04	4.71E-04	2.51E-04
31	7.37E-04	3.57E-04	9.09E-04	4.28E-04	7.18E-04	3.61E-04
32	7.23E-04	3.56E-04	9.25E-04	4.22E-04	7.53E-04	3.74E-04
33	9.50E-05	1.00E-04	3.11E-02	1.21E-02	7.80E-02	6.90E-03
34	8.31E-05	1.05E-04	7.47E-03	5.42E-04	4.60E-03	1.46E-03
35	1.13E-04	1.35E-04	3.62E-03	8.44E-04	5.68E-03	6.65E-04
36	1.10E-04	1.36E-04	3.06E-03	7.15E-04	4.86E-03	5.52E-04
37	3.13E-03	2.49E-03	1.87E-03	1.95E-03	1.71E-03	2.76E-04
38	6.98E-05	4.10E-04	3.46E-04	3.26E-05	1.47E-04	3.46E-05
39	1.28E-04	1.29E-03	1.12E-03	2.97E-04	2.04E-03	7.44E-05
40	1.86E-04	3.86E-04	4.43E-04	1.14E-04	1.41E-04	7.08E-05
41	8.80E-03	9.88E-03	5.67E-03	1.16E-03	3.37E-03	7.61E-04
42	5.69E-03	1.25E-02	5.77E-03	3.08E-04	3.67E-03	7.56E-04
43	1.13E-03	9.75E-03	4.03E-03	5.99E-05	9.69E-05	4.18E-04
44	9.65E-04	9.81E-03	4.12E-03	5.71E-05	1.02E-04	4.49E-04
45	3.33E-03	9.51E-03	4.16E-03	2.25E-04	6.60E-04	1.01E-03
46	6.05E-03	1.02E-02	4.82E-03	2.83E-04	3.43E-03	6.12E-04
47	6.37E-04	8.80E-03	3.79E-03	8.73E-05	9.88E-04	1.03E-03
48	6.86E-04	9.58E-03	3.88E-03	1.02E-04	8.82E-04	2.28E-04
49	5.67E-05	1.99E-04	4.22E-04	5.18E-05	3.05E-04	3.05E-02
50	7.25E-04	9.44E-03	3.77E-03	1.13E-04	2.00E-04	6.78E-03
51	8.62E-04	8.70E-03	3.78E-03	1.62E-04	6.00E-04	6.22E-03
52	7.55E-04	8.91E-03	3.88E-03	3.92E-05	3.44E-04	1.57E-02
53	6.40E-04	8.82E-03	3.84E-03	9.88E-05	5.10E-04	2.29E-02
54	5.01E-04	1.57E-03	2.03E-03	2.38E-05	3.53E-04	2.22E-02
55	3.60E-04	1.61E-03	2.41E-03	8.66E-05	2.97E-04	6.15E-03
AVG	7.97E-04	2.44E-03	2.28E-03	4.53E-04	2.43E-03	2.40E-03
STD	1.61E-03	3.86E-03	4.30E-03	1.61E-03	1.04E-02	6.12E-03

Planar System Experimental Validation Results

The free-floating environment emulation testbed was tested using two control methods. Firstly, with a pure FF controller where the RP actuators were driven solely to drive the forces and torques measured by the sensor to zero. Additionally, a center of mass regulation controller was augmented to the FF controller output for the second run.

The performance metrics used to compare the performances of the controllers are forces and torques measured at the interface and the velocity of the center of mass.
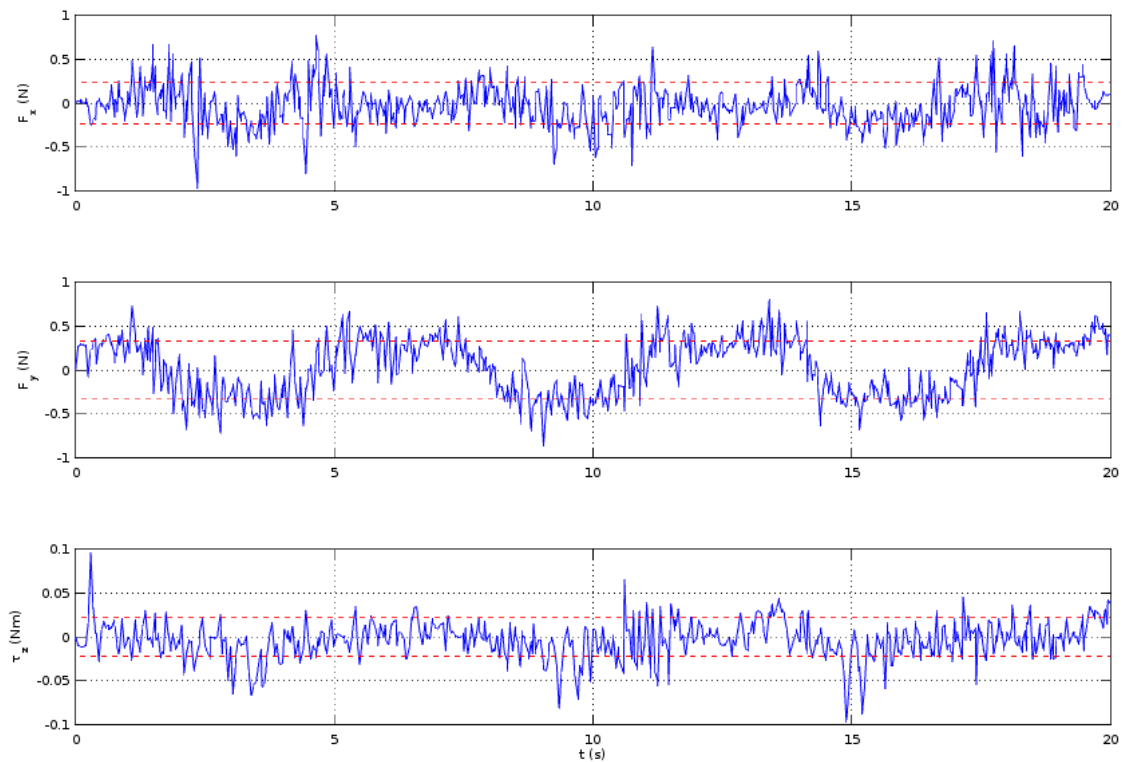

Figure 4.7: Plots of Forces and Torques Measured by the FT Sensor During Experimental Run with a Pure FF Controller

Force-Torque Measurement Analysis. Figure 4.7 consists of plots of forces and torques measured by the FT sensor during the experimental run when using a pure FF controller.

The solid blue curves indicate the measured forces and torques and the dashed red lines indicate the positive and negative bounds of the RMS values. Figure 4.8 shows plots of forces and torques measured by the FT sensor when running the FF controller augmented by the implemented center of mass regulator.
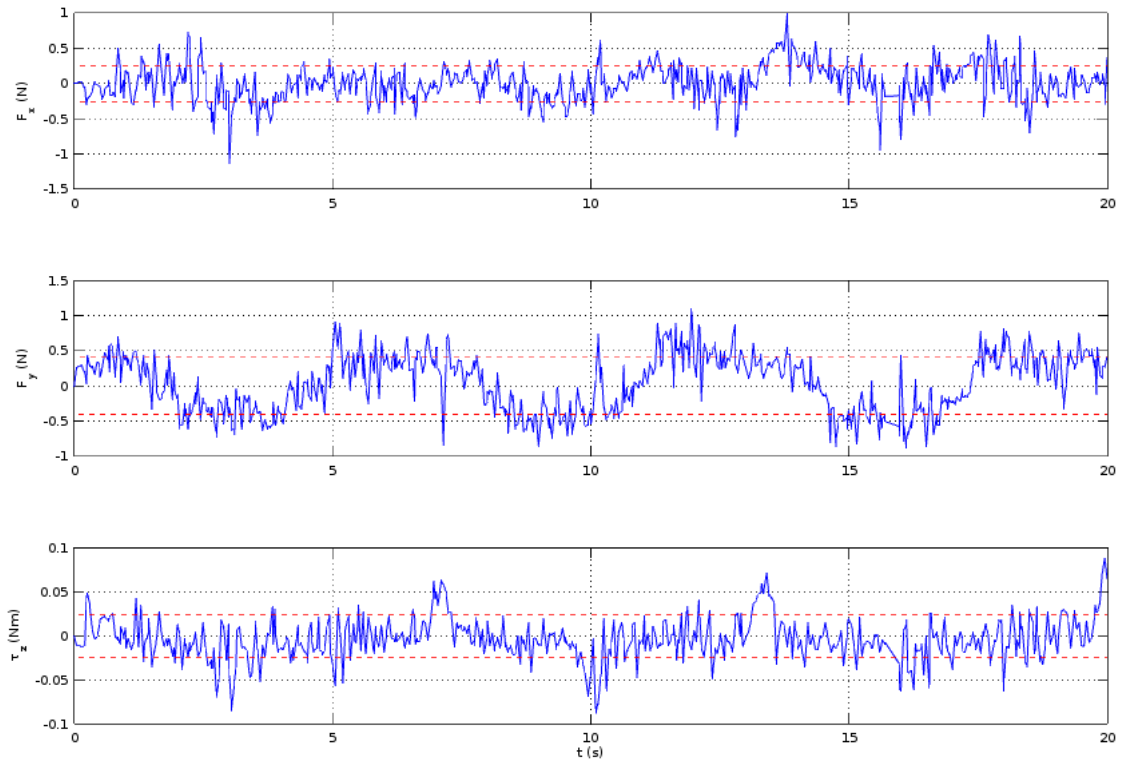

Figure 4.8: Plots of Forces and Torques Measured by the FT Sensor During Experimental Run with a FF Controller Augmented By a Center of Mass Regulator

75

As seen in Figures 4.7 and 4.8, the general trends of measured forces and torques are consistent and the controllers attempt to nullify the measured forces and torques. The RMS forces in the x and y directions and torque around the z direction for the pure FF controller are $0.233N$, $0.332N$ and $0.022Nm$ and are $0.255N$, $0.404N$ and $0.024Nm$ for the augmented controller. The RMS values of the measured forces and torques for the augmented controller are higher since the control goal is no longer solely to minimize those values.

Center of Mass Velocity Analysis. Figure 4.9 consists of plots of the planar center of mass locations and velocities estimated using the x and y torque measurements during the experimental run with the pure FF controller.

Figure 4.10 consists of plots of the estimated planar center of mass locations and velocities during the experimental run with the FF and augmented center of mass regulation controller.

As seen in Figures 4.9 and 4.10, the velocity of the center is bounded for both controller implementations. However, offsets in the center of mass velocities can lead to drift of the center of mass positions. This is clearly observed in the y coordinate of the center of mass in Figure 4.9.

The augmented FF and center of mass regulation controller is capable of decreasing the positional drift, as observed in Figure 4.10 in spite of greater RMS forces and torques.

The RMS of center of mass velocities for the pure FF controller run for 20 seconds of operation is $0.290m/s$ and that for the augmented FF and center of mass regulation controller is $0.158m/s$. This difference shows that firstly, the FF controller is capable of emulating a free-floating environment to an extent and especially when augmented with a center of mass regulator.
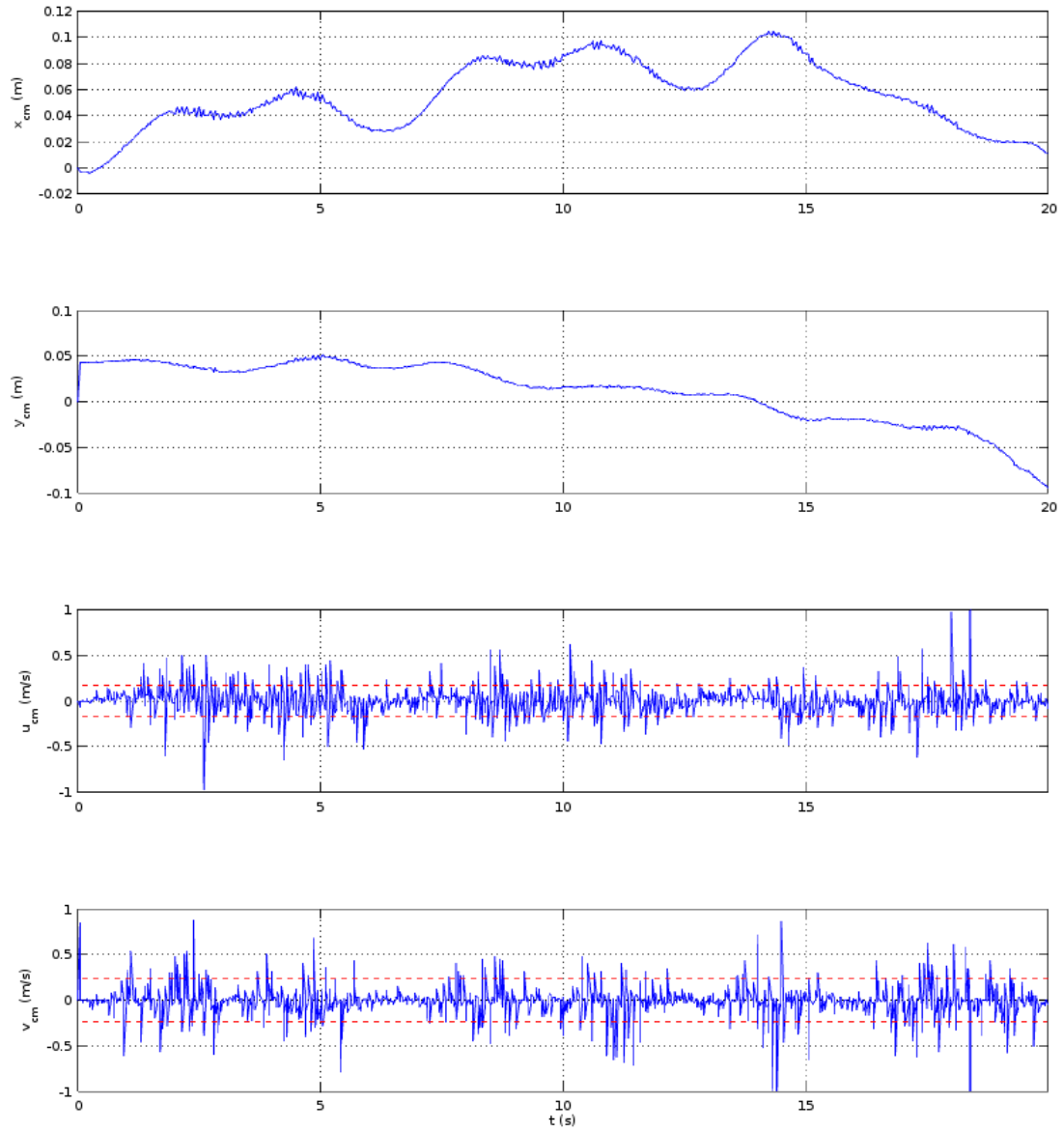
Figure 4.9: Plots of Center of Mass Locations and Velocities During Experimental Run with a Pure FF Controller
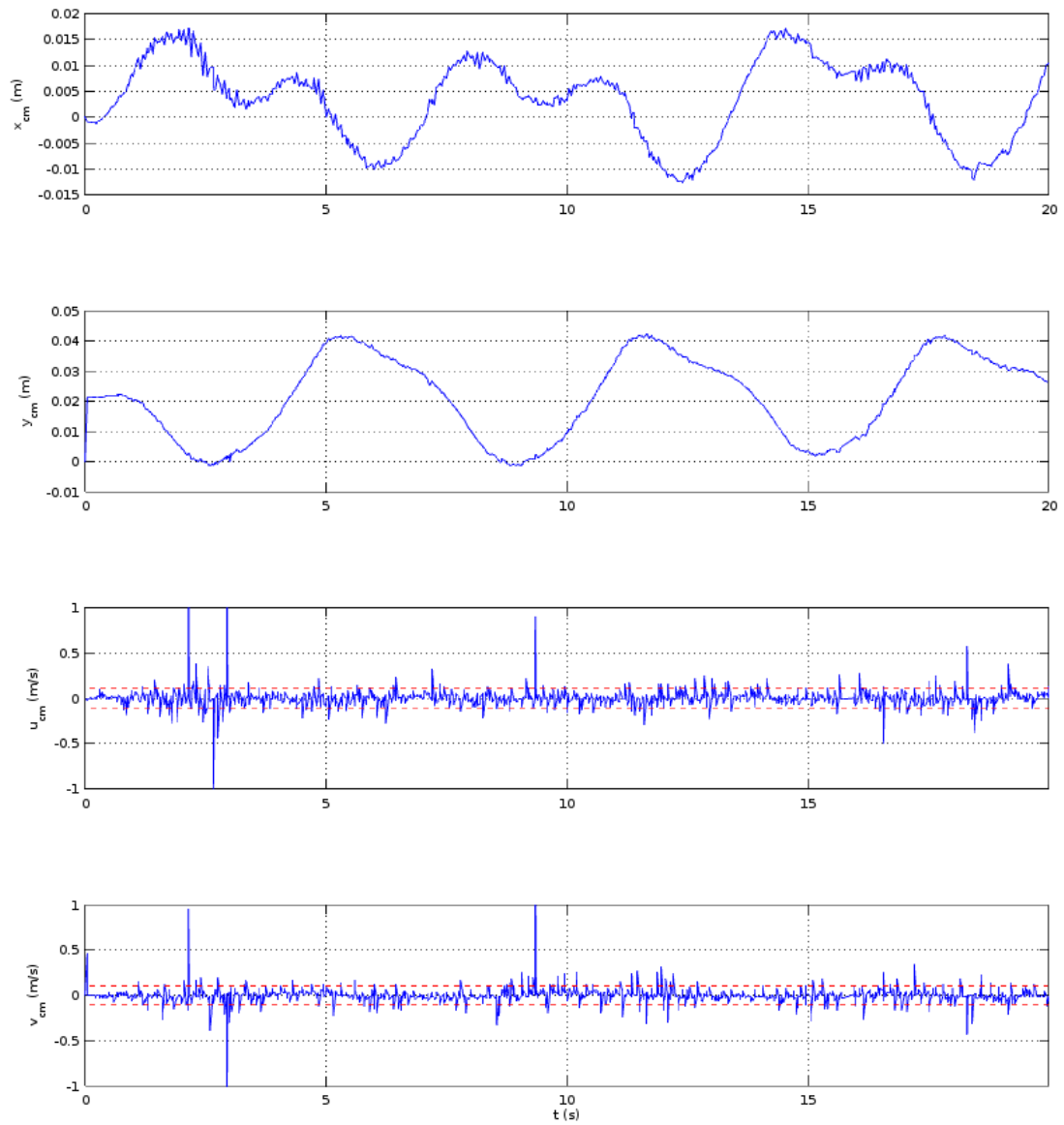
Figure 4.10: Plots of Center of Mass Locations and Velocities During Experimental Run with a FF Controller Augmented By a Center of Mass Regulator

Spatial System Simulation Results

The spatial free-floating environment emulation testbed was tested in simulation. The proposed gravity-compensated FF control was applied to the 7 DOF RP joints.
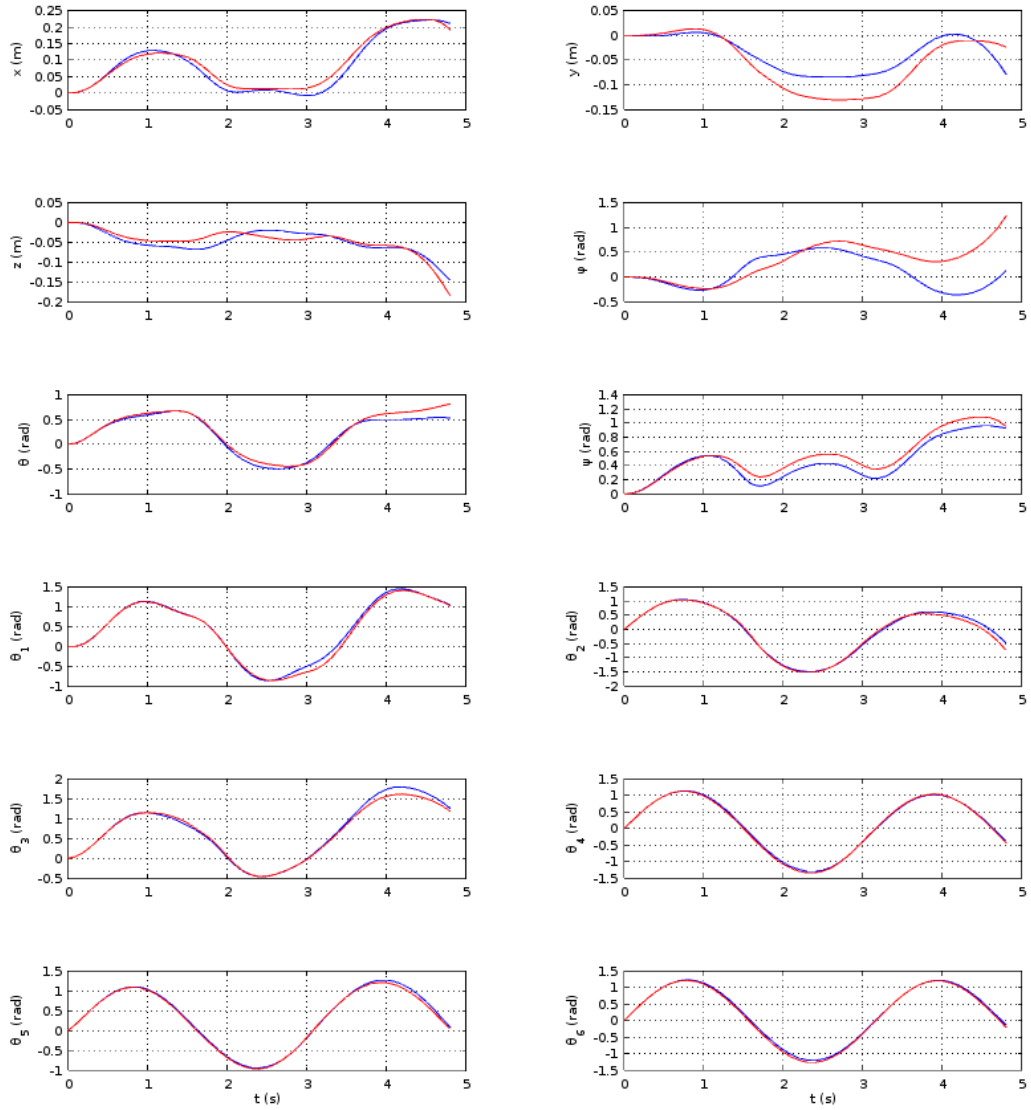

Figure 4.11: Trajectory Plots of SR Simulated in Free-floating Environment (blue) and SR on RP with FF Control (red)

Figure 4.11 consists of plots of the position of the SR body frame with respect to the initial position measured along the RP base frame, XYZ Euler angles representing the rotation between the RP base frame and SR body frame and the joint angles of the SR robotic manipulator.

It can be observed from Figure 4.11 that the trends of motion of the SR in free-floating environment are reproduced by the FF controller. Drifting in position and orientation, as also observed in the planar simulations and experimental validation, are also observed in the spatial simulation results.

Over the simulation period of 5 seconds, the Euclidean norm of RMS position error is $0.027m$, the Euclidean norm of RMS Euler angle error is $0.136rad$ and the Euclidean norm of RMS errors of all joint angles is $0.051rad$. The RMS errors and plots from Figure 4.11 show that the joint variable positions are emulated with much lower errors than the Euler angles. The tendency for the joint angles to drift is also minimal compared to the body position and orientation. This is achieved with a properly implemented gravity compensation algorithm. In order to improve the performance of the FF controller in a spatial emulation testbed, control augmentation is necessary. Control methods such as center of mass regulation can potentially be implemented to minimize errors in space robot body position and orientation.

Chapter V

Discussion, Conclusions and Recommendations

Discussion

Force-Feedback Control Method Viability.

The simulations and experimental validation performed as part of this research show that the use of Force-Feedback (FF) control for emulating a planar and spatial free-floating environment is a viable methodology. The FF control method does not require knowledge of second mass moments, which are difficult to measure or precisely estimate. Finally, the proposed method is augmented with other error-correction-based control methods (e.g. center of mass regulation), is viable for extended run times and was able to partially overcome the observed drift pattern.

Experimental System Disparities and Considerations.

The discrete nature and relatively low resolution of sensors plays a significant role in amplifying the measurement errors in a laboratory setting. Furthermore, the theoretical assumption that gravity does not influence the dynamic behavior of the system in a planar setup is not correct. Misalignments in the Robotic Platform (RP) assembly and angular deflections in the Force-Torque (FT) sensor induces rotations about the RP x and y axes, creating a component of gravity in those directions. These unwanted and unaccounted-for gravitational components resulted in time-varying offsets in the force and torque measurements ultimately causing a positional and orientation drift.

Conclusions

A planar free-floating environment testbed was simulated and the performance was compared to the industry standard solution (i.e. HIL simulation). The HIL simula-

tions were made with an induced 10% error in the masses, first mass moments and inertia tensors of the dynamic model. It was shown on average from the 55 system excitation inputs that the FF controller was more accurately able to recreate a free-floating trajectory that the HIL simulations with the 10% error. On average, the trajectory position errors were 90.79% less, the orientation errors were 96.33% less and the joint angle errors were 99.77% less with the FF controller. The ability for the control strategy to conserve linear and angular momentum was also studied and it was shown that the FF controller was able to reduce the linear momentum errors by 86.5% on average and the angular momentum errors by 75.16%. A drift was observed in position and orientation over extended simulation times.

Experimental validation was also performed for the planar free-floating environment. A complete testbed was constructed and a pure FF controller was implemented. The FF controller resulted in RMS forces of $0.233N$ and $0.332N$ in the x and y directions and a moment of $0.022Nm$ about the z direction, measured at the interface. In an ideal free-floating environment scenario, there would be zero forces and moments to ensure conservation of linear and angular momentum. The RMS center of mass velocity for the FF controller run was $0.290m/s$. Positional and orientation drift was also observed with the FF controller. When augmented with a center of mass regulator to address the drift, the controller resulted in RMS forces of $0.255N$, $0.404N$ and a moment of $0.024Nm$. The resulting RMS center of mass velocity was only $0.158m/s$. These results show that in spite of measuring larger forces and torques at the interface, the amount of drift was reduced.

Finally, a spatial system was tested in simulation and a gravity-compensated FF control was implemented. Simulation with the RP, FT sensor and the SR were run and the resulting trajectories were compared against the Space Robot (SR) simulated in a free-floating environment. The RMS position vector magnitude seen over the simulation time

was $0.027m$, the RMS Euler angle error vector magnitude was $0.136rad$ and the RMS joint angle errors vector magnitude was $0.051rad$.

Recommendations

Control Strategy Improvements.

It was shown with the planar experimental validation that improvement and augmentation of the pure FF controller is effective in improving performance and minimizing the observed drift. Furthermore, various other potential control strategies can be tested and implemented alongside FF control to improve performance.

Building a knowledge of the system dynamic model and dynamic parameters including zero, first and second mass moments of each link during runtime can allow for a transition from a pure FF controller to a different controller. A regressor matrix can be used to identify the parameters during run time.

Testbed Implementation Recommendations.

A complete calibration of the misalignment can potentially be performed to minimize the effect of gravity on the planar system. The calibration procedure would include recording FT sensor readings for various static configurations of the SR. The readings could then be used to construct an angle-dependent calibration matrix that corrects for misalignment.

The calibration procedure could also be a form of supervised learning procedure that could potentially be implemented either off or on-line.

References

[1] Jerry Haber. *Safety Design for Space Operations*. Butterworth-Heinemann, 2013.

[2] U.S. Congress Office of Technology Assessment. Round trip to orbit: Human space-flight alternatives special report, August 1989.

[3] George Bekey, Robert Ambrose, Vijay Kumar, David Lavery, Arthur Sanderson, Brian Wilcox, Junku Yuh, and Yuan Zheng. *Robotics: State of the Art and Future Challenges*. Imperial College Press, 2008.

[4] Angel Flores-Abad, Ou Ma, Khanh Pham, and Steve Ulrich. A review of space robotics technologies for on-orbit servicing. *Progress in Aerospace Sciences*, 68:1–26, 2014.

[5] Alex Ellery, Joerg Kreisel, and Bernd Sommer. The case for robotic on-orbit servicing of spacecraft: Spacecraft reliability is a myth. *Acta Astronautica*, 63(56):632–648, 2008.

[6] N. Muraleedharan, D. R. Isenberg, and I. Gentilini. Recreating planar free-floating environment via model-free force-feedback control. In *2016 IEEE Aerospace Conference*, pages 1–12, March 2016.

[7] Yangsheng Xu. The measure of dynamic coupling of space robot systems. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 615–620, May 1993.

[8] Wenfu Xu, Bin Liang, and Yangsheng Xu. Survey of modeling, planning and ground verification of space robotic systems. *Acta Astronautica*, 68:1629–1649, 2011.

[9] Jana L. Schwartz, Mason A. Peck, and Christopher D. Hall. Historical review of air-bearing spacecraft simulators. *Journal of Guidance, Control and Dynamics*, 26(4):513–522, 2003.

[10] Evangelos Papadopoulos, Iosif S. Paraskevas, Thaleia Flessa, Kostas Nanos, Georgios Rekleitis, and Ioannis Kontolatis. The ntua space robot simulator: Design & results. In *ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA 2008)*, 2008.

[11] Evangelos Papadopoulos, Iosif S. Paraskevas, Thaleia Flessa, and Georgios Rekleitis. The ntua space robot emulator: Design & experiments. In *International Conference on Intelligent Robots and Systems (IROS 2011)*, 2011.

[12] Michael Bloomfield George Hogan and Mark Smith. Case studies of testing at nasa's neutral buoyancy laboratory (nbl) for oil and gas industry risk mitigation. In *Offshore Technology Conference*, May 2015.

[13] Hirokata Sawada, Kyoichi Ui, Makoto Mori, Hiroshi Yamamoto, Ryoichi Hayashi, Saburo Matunaga, and Yoshiaki Ohkami. Micro-gravity experiment of a space robotic arm using parabolic flight. *Advanced Robotics*, 18(3):247–267, 2004.

[14] Farhad Aghili. Optimal control for robotic caputring and passivation of a tumbling satellite with unknown dynamics. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.

[15] Toralf Boge and Ou Ma. Using advanced industrial robotics for spacecraft rendezvouz and docking simulation. In *IEEE International Conference on Robotics and Automation (ICRA 2011)*, pages 1–4, 2011.

[16] Ou Ma, Angel Flores-Abad, and Toralf Boge. Use of industrial robots for hardware-in-the-loop simulation of satellite rendezvous and docking. *Acta Astronautica*, 81(1):335–347, 2012.

[17] S. Dubowsky, I. Paul, and H. West. An analytical and experimental program to develop control algorithms for mobile manipulators. In *Proceedings of the VIIth Symposium on Theory and Practice of Robots and Manipulators*, September 1988.

[18] Marc Garry Carmichael and Dikai Liu. Admittance control scheme for implementing model-based assistance-as-needed on a robot. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 870–873, July 2013.

[19] Steven Dubowsky, William Durfee, Thomas Corrigan, Andrew Kuklinski, and Uwe Müller. A laboratory testbed for space robotics: the ves ii. In *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems (ICRA 1994)*, pages 1562–1569, 1994.

[20] Christian Ott and Yoshihiko Nakamura. Base force/torque sensing for position based cartesan impedance control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2009.

[21] H.B. Brown Jr. and J.M. Dolan. A novel gravity compensation system for space robots. In *ASCE Specialty Conference on Robotics for Challenging Environments (Space94), Albuquerque, NM*, pages 250–258, 1994.

[22] Alessandro De Luca and Stefano Panzieri. Learning gravity compensation in robots: Rigid arms, elastic joints, flexible links. *International Journal of Adaptive Control and Signal Processig*, 7(5):417–433, 1993.

[23] Greg White and Yangsheng Xu. An active z gravity compensation system. Technical Report CMU-RI-TR-92-09, Robotics Institute, Pittsburgh, PA, July 1992.

[24] L Sciavicco and B Siciliano. *Moledlling and Control of Robot Manipulators*. Springer Publishing Company, 2000.

[25] F.L. Lewis, C.T. Abdallah, and D.M. Dawson. *Control of robot manipulators*. Macmillan Pub. Co., 1993.

[26] Conrad Sanderson and Ryan Curtin. Armadillo c++ linear algebra library. `http://arma.sourceforge.net/`, 2017.

[27] OpenSceneGraph Team. Openscenegraph open source high performance 3d graphics toolkit. `http://www.openscenegraph.org/index.php/33-openscenegraph/4-front-page`, 2017.

[28] Tecgraf (formerly the Computer Graphics Technology Group of PUC-Rio) at the Pontifical Catholic University of Rio de Janeiro. Lua the programming language. `https://www.lua.org/`, 2017.

[29] N. Muraleedharan, D. S. Cohen, and D. R. Isenberg. Omnidirectional locomotion control of a pendulum driven spherical robot. In *SoutheastCon 2016*, pages 1–6, March 2016.

[30] Hirata industrial cartesian robots. http://www.hirata.co.jp/en/products/view/171.

[31] USA Maxon Motor. Maxon re65 dc motor. `http://www.maxonmotor.com/maxon/view/product/353295`, 2017.

[32] ATI Industrial Automation. Ati industrial automation: F/t sensor mini40. `http://www.ati-ia.com/Products/ft/ft_models.aspx?id=Mini40`.

[33] ROBOTIS. Robotis dynamixel mx-28at servo. `http://www.robotis.us/dynamixel-mx-28at/`.

[34] ROBOTIS. Robotis opencm embedded controller. `http://en.robotis.com/index/product.php?cate_code=131010`.

[35] Digi International. Digi xbee rf module. `https://www.digi.com/lp/xbee/hardware`.

[36] ADLINK Technology Inc. Adlink daqe-2213 data acquisition card. `http://www.adlinktech.com/PD/web/PD_detail.php?pid=665`.

[37] ADLINK Technology Inc. Adlink pcie-7296 data acquisition and digital i/o card. `http://www.adlinktech.com/PD/web/PD_detail.php?pid=890`.

[38] N. Muraleedharan. SR and RP Dynamical and Electrical Parameters, and Lua Script to Generate Test Sets. `http://robotics.pr.erau.edu/arsr/testset_params.zip`, 2015.