# Experimental Validation of the Efficient Robotic Transportation Algorithm for Large-scale Flexible Space Structures

by

Masahiro Ono

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2007



Author .....
Department of Aeronautics and Astronautics
August 10, 2007

Certified by .....
Steven Dubowsky
Professor
Thesis Supervisor

Accepted by .....
David L. Darmofal
Associate Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students



AERO

# Experimental Validation of the Efficient Robotic Transportation Algorithm for Large-scale Flexible Space Structures

by

Masahiro Ono

Submitted to the Department of Aeronautics and Astronautics
on August 10, 2007, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

## Abstract

A new large space structure transportation method proposed recently is modified and experimentally validated. The proposed method is to use space robots' manipulators to control the vibration, instead of their reaction jets. It requires less fuel than the reaction jet-based vibration control methods, and enables quick damping of the vibration. The key idea of this work is to use the *decoupled controller*, which controls the vibration mode and rigid body mode independently. The performance of the proposed method and the control algorithm is demonstrated and quantitatively evaluated by both simulation and experiments.

Thesis Supervisor: Steven Dubowsky
Title: Professor

# Acknowledgments

I am most appreciative of my advisor Professor Steven Dubowsky for his guidance, for his advice, and for giving me this wonderful opportunity. I would also like to thank the Japan Aerospace Exploration Agency (JAXA) for their financial support of this research.

I am lucky to have bright, talented, and diligent research partners; Peggy Boning has given me a lot of helpful opinions and advices, and corrected my poor English in this thesis; Tatsuro "Ted" Nohara has always stayed in the lab with me until past midnight and helped with the experiments; Prof. Yoji Kuroda, Dr. Jamie Nichol, Dr. Matthew Lichter, and Amy Bilton designed and developed the experimental system together, which was essential to my work; undergraduates, Andrew R. Harlan, Patrick R. Barragan, and Ta Y. Kim, help me a lot with hardware design, manufacturing, and software development. I would like to thank them all again. I have greatly enjoyed working with them. Thanks also to all the other members of the Field and Space Robotics Laboratory, particularly Chris Brooks, Chris Ward, Dimitrios Tzeranis, Dr. Jean-Sebastien Plante, Dr, Kenjiro Tadakuma, Dr. Karl Iagnemma, Lauren DeVita, Martin Udengaard, Samuel Kesner, Steve Peters, Dr. Yan Shaoze, Yoshiyuki Ishijima.

I would also like to thank my friends and family for their support over the past two years. Father and Mother, thank you very much for encouraging and supporting me to study in a foreign country to pursue my dreams. I am very happy to be surrounded by many wonderful friends, who always cheer me up.

Finally, I would like to dedicate this thesis to my grandfather, Kazuyoshi Ono, who was a professor of civil engineering at Kanazawa University, and passed away last February. He was not only a wonderful grandfather, but also a respected engineer, who dedicated his life to improving railway technology. As an engineer, I am also determined to dedicate my life to the development of technology to contribute to human society, just like my grandfather did with his life.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

Robots have been playing important roles in many aspects of human civilization since the end of the twentieth century. Factories, which were once filled with numerous diligent workers, are now running on various industrial robots which work for 24 hours/day, 7days/week without any complaints for ceaseless drudgery. Robots are suitable for work in environments which are dull, dirty, and dangerous, such as factories, mines, underwater, or toxic areas.

Robots are playing major roles not only on earth but also in deep space; the history of planetary exploration is almost the history of the robotic space probes. While human beings have walked only on two celestial bodies, robotic probes have visited all the planets in our solar system, and landed on two planets other than Earth, two moons, and two asteroids, all of which resulted in great scientific discoveries. However, there is still one place in space where humans play the central role; Low-Earth orbit. There is increasing need for on-orbit service of satellites and on-orbit construction of space structures. With current technology, astronauts still have to conduct dangerous space walks just to replace the broken gyroscopes of the Hubble Space Telescope or to connect cables and fasten bolts to add modules to the International Space Station.

On-orbit human operation is undesirable for three main reasons; cost, safety, and accessibility. Manned space flight costs much more than unmanned flight. For

example, one Space Shuttle flight costs over 800 million dollars, and the International Space Station costs 2.5 billion dollars [NASA's FY 2008 Budget] per year to maintain.

Manned space flight is a dangerous adventure. As of April 2007, there have been 249 manned space flights in history (not including suborbital flight), and five of them (Apollo 1, Soyuz 1, Soyuz 11, Challenger, and Columbia) encountered serious accident with loss of crews, which resulted in an approximately 2% death rate.

Manned space flight has problems with accessibility since cost and safety issue put limit on the number of manned space flights. In practice, four or five manned flights per year are the limit for both US and Russia. Moreover, once an accident occurs, a manned space program stops for several years. The International Space Station is an example to show the limited accessibility of on-orbit construction by human beings; it has been nine years since the start of construction, and it is not finished yet. Long construction periods increase the budget, which again limits the number of space flights and worsens the accessibility, causing a vicious cycle.

It is questionable if it is worth spending 800 million dollars and taking 2% risk of the loss of human life to send human to space just to connect some cables and fasten bolts. With the cost of five manned missions, 20 to 30 robotic space missions would be possible, making on-orbit construction and service more accessible without any danger to life.

Therefore Human operations in space should be replaced by robots as much as possible. Using an intelligent fleet of robots, large space structures such as the International Space Station can be constructed more unexpensively and quickly without the risk to life. With on-orbit servicing by robots, satellites do not have to be abandoned when they run out of fuel or need repair; they can be refueled or repaired by robots, and brought back to life. Space robots may contribute to the solution to global energy problem. They would be a necessary technology for the construction of space solar power stations, although the launch cost has to be dramatically reduced to be economically feasible[1][15].

The MIT Field and Space Robotics Laboratory has been working on various problems in the design and control of space robotics for many years, including intelligent

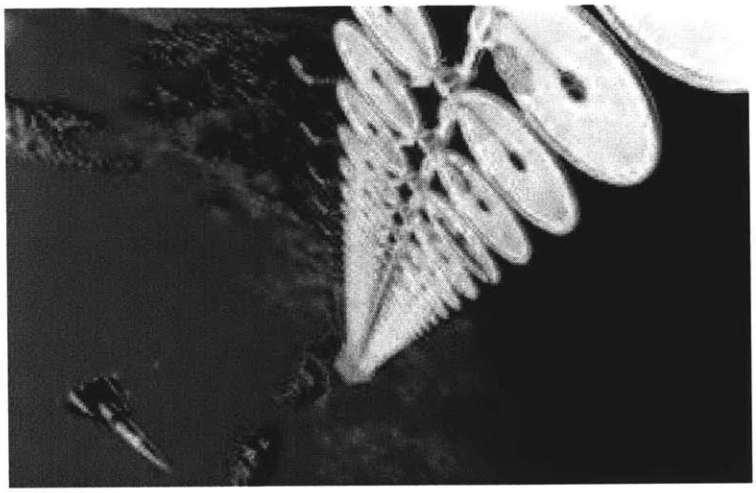

Figure 1-1: NASA's Sun Tower concept [15]

vision-based sensing , safe path planning, free-flying manipulator control, and assembly of large space structures[6][13][14][17][24]. This thesis focuses on a experimental validation of the transportation algorithms for large flexible space structure, which is a necessary step for the construction of large-scale architectures on orbit.

## 1.2 Background and Literature Review

When dealing with large space structures, vibration control is an important engineering problem for the following reasons:

1. Large space structures are very flexible. To minimize the expensive launch cost, the space structures are designed to have minimum weight, which results in the thin and flexible structure. Thus the vibration of space structures cannot be neglected.

2. Vibrations may be excited by robots' actuators, such as thrusters and reaction wheels.

3. The Natural damping of these space structures will be small. In the zero-gravity and vacuum of the space there are few mechanisms that can dissipate vibrational

energy. Therefore the vibrations of large flexible space structures are hard to suppress without active control.

4. Finally, because of the large size of space structures, they have low natural frequency, which results in very slow natural damping.

Thus the robots that transport large space structures must actively control the vibration of these structures. Failing to do so would delay the construction, or result in the damage of the structures and/or the robots.

Since space robots do not have fixed base and the dynamics of robots and flexible large space structures are highly coupled, active vibration damping is more difficult for the on-orbit systems than for the ground-based systems.

There has been a significant amount of work in the area of controlling flexible space structures and spacecraft [3][4][5][7][21][23]. An on-orbit vibration control experiment was conducted using Japanese satellite ETS-VI, which was quite successful [20]. Most of these past work focuses on the vibration control of the flexible structures rigidly attached to the spacecraft's body such as solar arrays and antennas. Although they employ different control schemes, the reaction jet attitude control system (RCS) is commonly used for the actuation. Another approach proposed in [8] is to use piezoelectric actuators placed on the flexible structure. This work is interesting in that it uses only the position and velocity measures of the spacecraft body by exploiting the dynamics model.

However, these on-orbit vibration control methods are not suitable for the future large space structure for two reasons. Firstly, in the past work, the flexible structure is modeled as a part of the spacecraft such as a solar array or an antenna. However, the future large space structure should be a passive object held by other spacecraft (ie. space robots), since it is inefficient (or even impossible) to implement thrusters, reaction wheels, or piezoelectric actuators in all modularized components of the large space structures. In this work, the flexible structure is modeled as a passive object held by space robots.

Secondly, in the most of the past work, the vibration is controlled by reaction jet. However, fuel is a limited and expensive resource on the orbit. A significant amount of fuel has to be consumed to dissipate the huge vibration energy of the large space structure, which is quite inefficient in terms of the cost. In this work, manipulators are used to control the vibration. Manipulators can be powered by electrical energy, which can be obtained unlimitedly from solar array. Therefore this approach allows space robots to reduce fuel consumption, which will result in the lower construction cost. This approach will also reduce the effect of the plume impingement problem.[22][25]

The theoretical basis of this work is the a vibration control algorithm using manipulators proposed by Y. Ishijima, et al.[12]. His approach is to use a *decoupled controller* to control the coupled dynamics of robot-flexible structure system, and he successfully proved the effectiveness of the controller in simulations.

The major contribution of this work is the experimental validation of this new vibration control algorithm using manipulators. Several modifications to the original theory are also made to give the robustness to sensor, actuator, and model errors found in the real systems.

## 1.3 Experiment Methodology

On-orbit space robotics experiments are difficult due to the expensive cost and long development periods of the spacecraft. The average cost of the current satellite is in the order of $100M, and need approximately a 10- year development period. [1]

Therefore ground-based technology validation is necessary before conducting on-orbit experiment. Using a flat table is a suitable experiment method for this purpose. By floating objects on polished flat table using air bearings, 2-D non gravity inertia space with little friction can be realized on ground. Several space robotics experiments have been conducted using this method.[2][19]

[1]Recently emerging small satellite technology may be the solution for the cost and development period. Several 1kg class satellites developed by university laboratories have been launched and worked successfully on orbit[10][16]. However, the capabilities of small satellites are limited, and not suitable for complicated space robotics experiment in the existing circumstances.

The Field and Space Robotics Lab developed the flat table-based *FSRL Free Flying Space Robotics Test Bed* (Figure 1-2) with two free-floating dual-arm robots. Using the test bed, this work experimentally validated the effectiveness of the new large space structure transportation algorithm which actively controls vibrations using robots' manipulators while it controls the rigid body mode using thrusters.

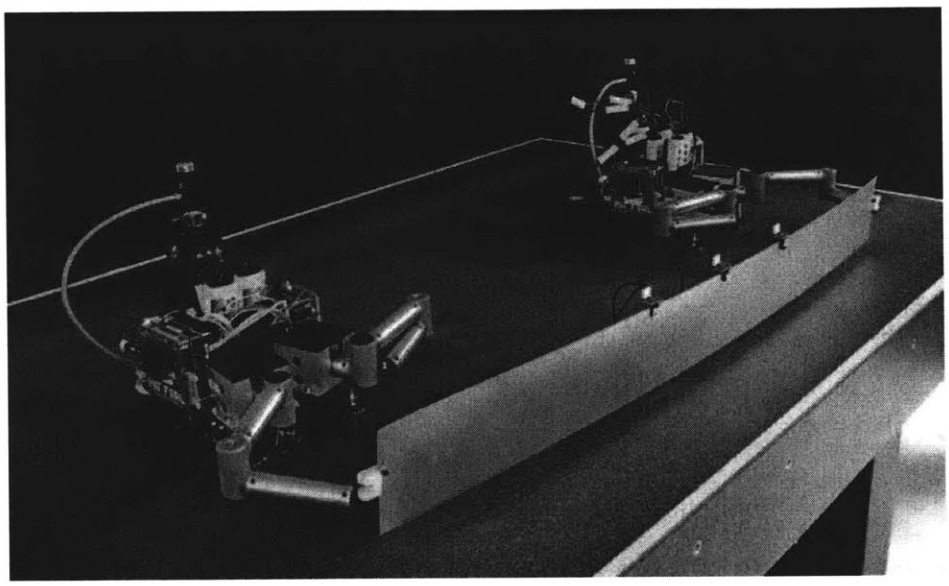

Figure 1-2: FSRL Free Flying Space Robotics Test Bed

## 1.4 Problem Statement and Approach

The objective and the approach of this work is summerized as follows;

**Problem Statement**

1. Develop an efficient and robust algorithm to transport large space structures on orbit with actively controlling the vibration

2. Demonstrate and quantitatively evaluate the algorithm in laboratory experiment.

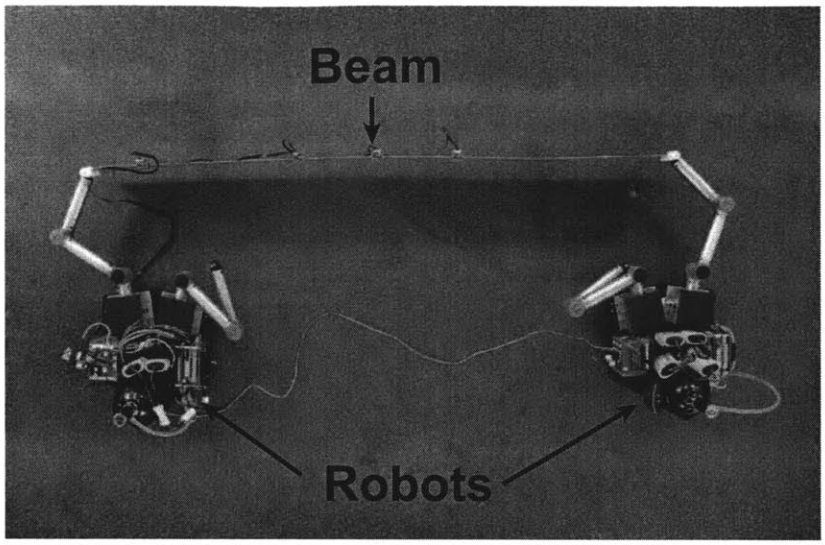


Figure 1-3: Top view of the FSRL Free Flying Space Robotics Test Bed

**Approach is to:**

1. use robots' manipulators for the vibration control of flexible structure while using their thrusters to control the position of the center of mass during the transportation operations.

2. Conduct an experiment using the FSRL Free Flying Space Robotics Test Bed.

## 1.5 Thesis Outline

The theoretical basis of the decoupled controller are described and discussed in Chapter 2. The simulation results are presented in Chapter 3, followed by the experimental result in Chapter 4. The mechanism and functions of the FSRL Free Flying Space Robotics Test Bed developed in this work are described in the appendix.

# Chapter 2

# Theory

## 2.1 System Dynamics Model

To design the controller and develop the simulator, the dynamics of the beam and the robots has to be modeled properly. The dynamics of the beam is modeled as a linear time-invariant (LTI) system. The following equation of motion describes the rigid-body dynamics of the beam.

$$\begin{aligned} m_b \ddot{\boldsymbol{x}}_b &= \sum_{i=1}^{2} \boldsymbol{F}_{mp,i} \\ \boldsymbol{I}_b \dot{\boldsymbol{\omega}}_b &= \sum_{i=1}^{2} \boldsymbol{r}_{mp,i} \times \boldsymbol{F}_{mp,i} \end{aligned} \tag{2.1}$$

where $m_b$ is the mass of the beam, $\boldsymbol{I}_b$ is the inertia of the beam, $\boldsymbol{x}_b$ is the position of the center of mass of the beam, $\boldsymbol{\omega}_b$ is the angular velocity of the beam, $\boldsymbol{r}_{mp,i}$ is the vector from the beam's center of mass to the point where the $i$th robot holds the beam by its manipulator, and $\boldsymbol{F}_{mp,i}$ is the force applied to the beam by the manipulator of the $i$th robot. Figure.2-1 gives the graphical interpretation of these vectors.

The linear equation of motion of the beam vibration is obtained by modal decomposition of the Euler-Bernoulli beam equation as follows[18].

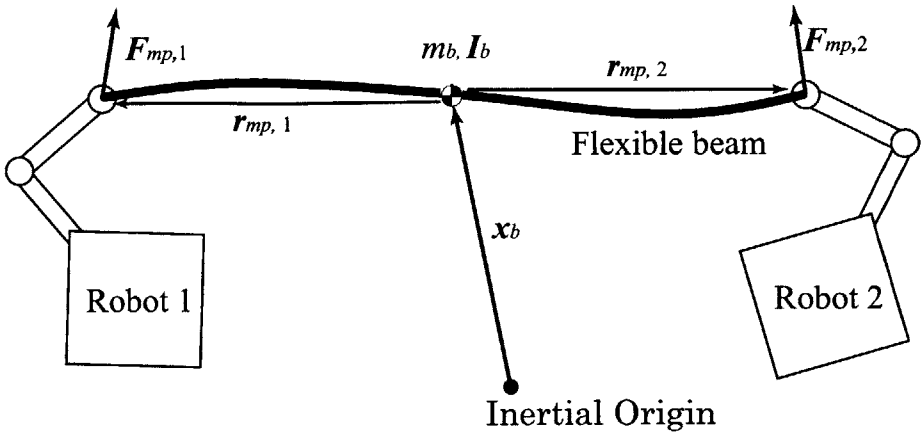


Figure 2-1: Definitions of vectors appeared in the equation of motion Eq. (2.1)

$$\ddot{\boldsymbol{q}} + 2\boldsymbol{Z}\boldsymbol{\Omega}\dot{\boldsymbol{q}} + \boldsymbol{\Omega}^2 = \boldsymbol{M}^{-1}\boldsymbol{F}_{md} \tag{2.2}$$

where $q$ is the modal coordinates of vibration modes. $\boldsymbol{Z}$ and $\boldsymbol{\Omega}$ are diagonal matrices whose diagonal elements are damping ratios and natural frequencies of corresponding vibration modes respectively. $\boldsymbol{M}$ in Eq. (2.2) is the diagonal modal mass matrix, whose $n$th diagonal elements is defined as follows using mode shape $\boldsymbol{\Phi}_n$;

$$M_n = m\int_{x=0}^{x=L} \boldsymbol{\Phi}_n(x)\cdot\boldsymbol{\Phi}_n(x)dx \tag{2.3}$$

where $x$-axis runs along the beam's length. $\boldsymbol{F}_{md}$ is the modal force vector, whose $n$th element is defined as follows;

$$F_{md,n} = \sum_{i=1}^{2} \boldsymbol{\Phi}_n(x)\cdot\boldsymbol{F}_{mp}dx \tag{2.4}$$

## 2.2 Estimator

Accelerometers are used to capture the beam's vibration. Using Kalman Filter, noisy accelerometer data is combined with beam's dynamics described in Eq. (2.2) to obtain the maximum likelihood sequence estimation of beam's vibration mode parameters. Vision sensors such as laser range finder can be combined to enhance the resolution.

Refer to [6][14] for further details about the vibration mode estimator used in this work.

## 2.3 Decoupled Controller

A decoupled controller proposed by Y. Ishijima is used and modified to control the coupled dynamics of the beam and the robots [12]. The decoupled controller used in this work consists of three controllers; the vibration controller, the rigid-body mode controller, and the manipulator compliance controller.

*The vibration controller* is a LQR state feedback controller using robots' manipulators which controls the beam's vibration and rigid-body mode. It takes the estimated states of vibration and rigid-body mode of the beam as the input, and provides the force commands of the manipulator. The details of the controller are described in Section 2.3.2.

*The rigid-body mode controller* is a classical PD controller using robot's thrusters which controls robot's position and velocity. It takes the robot's position and velocity, and determines the thruster force. The details of the controller are described in Section 2.3.3.

*The manipulator compliance controller* is added to bridges the above two controllers. It behaves like springs and dampers which connect two robots to the beam as Figure 2-4. It takes the relative position of end effectors from the robot body and outputs compliance force at the end effectors. The details of the controller are described in Section 2.3.4.

The entire controller design is shown in the block diagram Figure.2-2.

### 2.3.1 Modification to thr original algorithm

The major difference between Ishijima's original algorithm[12] and the modified algorithm used in this work is summarized as follows.

1. In the modified algorithm, the vibration controller does not control the rigid-

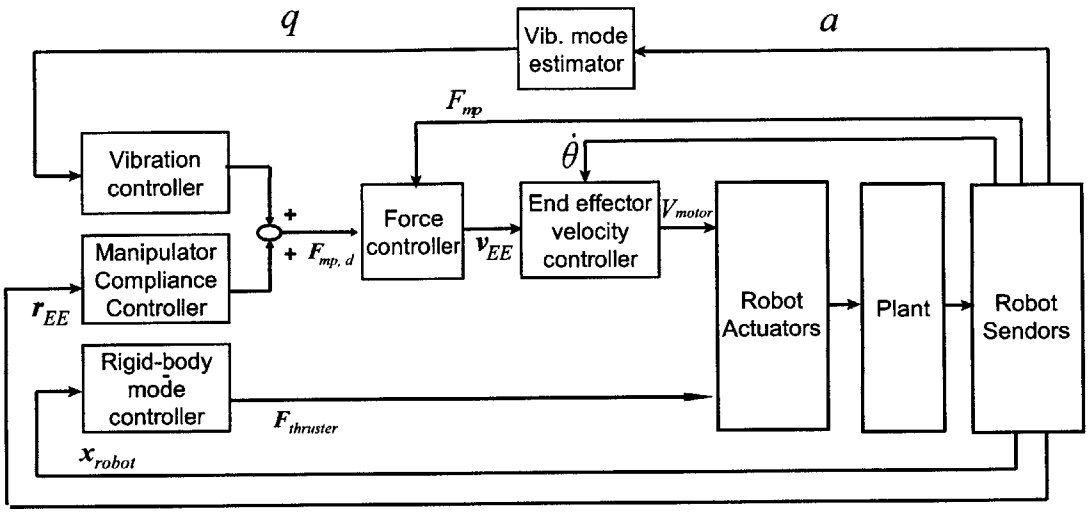


Figure 2-2: Controller block diagram

$a$ - Accelerations of the beam — $q$ - Vibration mode parameters
$F_{mp}$ - Measured manipulator force — $F_{mp,d}$ - Commanded manipulator force
$\dot{\theta}$ - Manipulator joint velocities — $V_{motor}$ Motor PWM duty ratio
$r_{EE}$ - End effector position — $v_{EE}$ - End effector velocity
$x_{robot}$ - Robot position and velocity — $F_{thruster}$ - Thruster force

body mode of the beam

2. The manipulator compliance controller is added to the modified controller

In the original algorithm, the rigid-body mode of the beam and the position of the robots are controlled independently; the vibration controller controls the beam's vibration mode and rigid-body mode simultaneously while the rigid-body mode controller controls the position (ie. rigid-body mode) of the robots. In this setting the two controllers have to be perfectly synchronized so that the robots do not get away from the beam. However, in the real-world, it is impossible due to the sensor and actuation error. If the robots get too away from the beam, the manipulators are completely stretched and enter the singular point, which disables the vibration control.

In the modified algorithm, the rigid-body mode of the beam is not explicitly controlled. However, it is constrained by the manipulator compliance controller. Therefore, by controlling the robots' position using the rigid-body mode controller, the beam follows the robots and thus its rigid body-mode is controlled.

### 2.3.2 Vibration Controller

This controller is a simple state feedback controller described as follows;

$$\begin{bmatrix} F_{mp,y1} \\ F_{mp,y2} \end{bmatrix} = \boldsymbol{K}_{lqr} \begin{bmatrix} \hat{\boldsymbol{q}} \\ \dot{\hat{\boldsymbol{q}}} \\ \hat{F}_{mp,y1} \\ \hat{F}_{mp,y2} \end{bmatrix} \tag{2.5}$$

where the scalar $F_{mp,yi}$ is the perpendicular component (y-components in Figure 2-3) of the force applied to the beam applied by the $i$th robot's manipulator while $\hat{F}_{mp,yi}$ is the measured (estimated) manipulator force. Vectors $\hat{\boldsymbol{q}}$ and $\dot{\hat{\boldsymbol{q}}}$ are estimated vibration mode parameters.

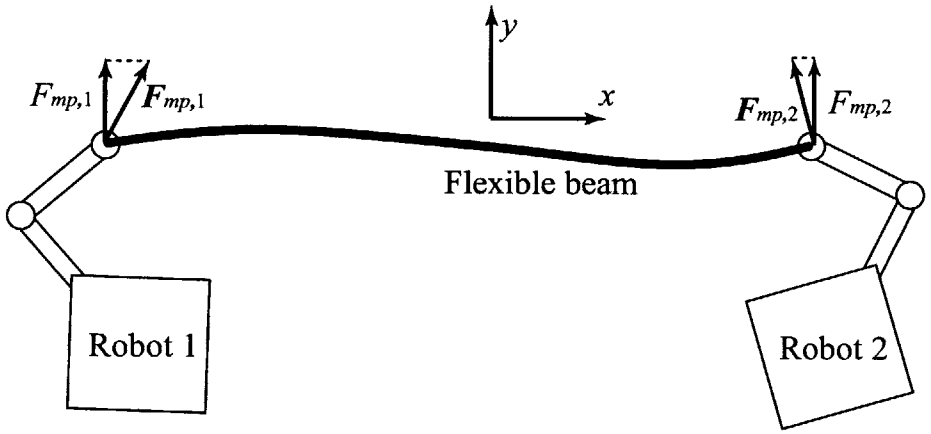


Figure 2-3: Vibration Control

Note that the right-hand side input vector includes measured manipulator force $\hat{F}_{mp,y1}$ and $\hat{F}_{mp,y2}$. This is because the delay in the low-level controllers (Section 2.3.5)

is also taken into account to obtain the LQR feedback gain $K_{lqr}$. Since LQG regulators are very sensitive to errors in the model, the close loop system easily becomes unstable unless the dynamics including low-level controller delay is well modeled[9].

The original equation of motion of the beam vibration (Eq.2.2) does not include the delay. By approximating the delay of the low-level force controller as a first-order delay, the equation of motion of the beam vibration including the delay is described as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{2.6}$$

where

$$\boldsymbol{x} = \begin{bmatrix} \hat{\boldsymbol{q}} \\ \dot{\hat{\boldsymbol{q}}} \\ F_{mp,y1} \\ F_{mp,y2} \end{bmatrix} \tag{2.7}$$

$$\boldsymbol{u} = \begin{bmatrix} F_{mp,y1,des} \\ F_{mp,y2,des} \end{bmatrix} \tag{2.8}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{1} & 0 & 0 \\ -2\boldsymbol{Z}\boldsymbol{\Omega}\dot{\boldsymbol{q}} & -\boldsymbol{\Omega}^2 & \boldsymbol{M}^{-1}\Phi_1 & \boldsymbol{M}^{-1}\Phi_2 \\ 0 & 0 & -1/\tau & 0 \\ 0 & 0 & 0 & -1/\tau \end{bmatrix} \tag{2.9}$$

$$\boldsymbol{B} := \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/\tau & 0 \\ 0 & 1/\tau \end{bmatrix} \tag{2.10}$$

where $\tau$ is the time constant of the low-level controller delay and $F_{mp,yi}$ is the actual manipulator force while $F_{mp,yi,des}$ is the desired (commanded) manipulator force (i.e. the output of the vibration controller).

Using this model, the state feedback gain $K_{lqr}$ is obtained by solving infinite

horizon Riccati equation so that the following performance metric $J$ is minimized.

$$J = \int_0^\infty (\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u}) dt \tag{2.11}$$

### 2.3.3 Rigid-Body Mode Controller

The rigid-body mode controller is a simple PD controller which controls the position and orientation of each robot. The output of the controller is the thruster force.

$$\boldsymbol{F}_{th} = \boldsymbol{K}_{p,th}(\boldsymbol{x}_{robot} - \boldsymbol{x}_d) + \boldsymbol{K}_{d,th}(\dot{\boldsymbol{x}}_{robot} - \dot{\boldsymbol{x}}_d) \tag{2.12}$$

where $\boldsymbol{x}_d$ and $\dot{\boldsymbol{x}}_d$ are the desired position and velocity respectively. $\boldsymbol{K}_{p,th}$ and $\boldsymbol{K}_{d,th}$ are the proportional and differential gain.

### 2.3.4 Manipulator Compliance Controller

The manipulator compliance controller is also a simple PD controller which behaves like springs and dampers between the manipulator's end effector and the reference point, which is fixed to the robot's local coordinate (see Figure 2-4).

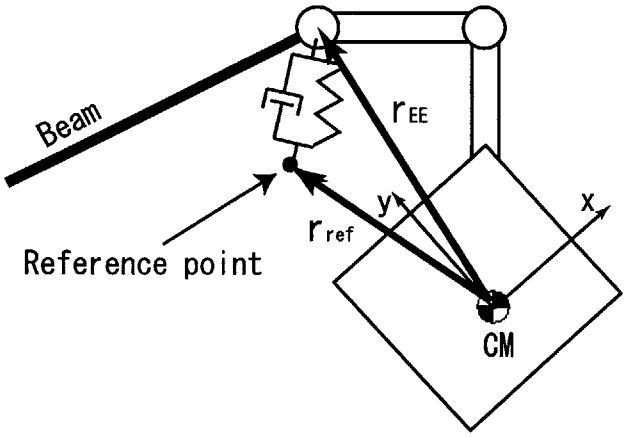


Figure 2-4: The concept of the manipulator compliance controller.

The manipulator force command is obtained by the following equation.

$$\boldsymbol{F}_{mp} = \boldsymbol{K}_{p,mcc}(\boldsymbol{r}_{EE} - \boldsymbol{r}_{ref}) + \boldsymbol{K}_{d,mcc}(\dot{\boldsymbol{r}}_{EE} - \dot{\boldsymbol{r}}_{ref}) \tag{2.13}$$

where $r_{EE}$ is the end effector position, $r_{ref}$ is the position of the reference point, $\boldsymbol{K}_{p,mcc}$ and $\boldsymbol{K}_{d,mcc}$ is the proportional and differential gain. Note that Eq. (2.13) is described in the robot's local coordinates that are fixed to the robot's body.

This controller plays a critical role to bridge the vibration controller (Section 2.3.2) and the rigid-body mode controller (Section 2.3.3). Since the vibration controller does not consider the beam's rigid-body mode, the beam does not follow robots without this controller.

### 2.3.5 Low-level Controllers

Several low-level controllers are used to control the manipulators (See the block diagram, Figure 2-2).

**End effector force controller** is necessary to output the desired end effector force given by the vibration controller (Section 2.3.2) and the manipulator compliance controller (Section 2.3.4).

**End effector velocity controller** is necessary to implement the end effector force controller.

Both low-level controllers are implemented by simple PDI feedback controllers.

## 2.4 Vibration Controller with Thrusters

In later chapters of this thesis, the performance of the proposed decoupled controller is compared to an approach where the vibration is controlled by thrusters with manipulator licked. This subsection explains how this approach is applied to our problem.

The control algorithm is the same as the vibration controller described in Section 2.3.2, with the exception that the actuators are thrusters, not manipulators. Thruster force is obtained from the state feedback controller;

$$F_{th} = K_{lqr}\boldsymbol{q} \tag{2.14}$$

where $\boldsymbol{q}$ is the vibration coordinates.

To obtain the optimal state feedback gain $K_{lqr}$ in Eq. 2.14, the same dynamics model used in Section 2.1 is utilized. For this controller, the manipulators are not actively controlled, but they simply connect the beam and the robots. The control input is the thruster force of both robots.

The equation of motion of the robots and the beam are repeated below.

$$m_1\ddot{\boldsymbol{x}}_{robot1} = -\boldsymbol{F}_1 + \boldsymbol{F}_{th,1} \tag{2.15}$$

$$m_2\ddot{\boldsymbol{x}}_{robot2} = -\boldsymbol{F}_2 + \boldsymbol{F}_{th,2} \tag{2.16}$$

$$\ddot{\boldsymbol{q}} + 2\boldsymbol{Z}\boldsymbol{\Omega}\dot{\boldsymbol{q}} + \boldsymbol{\Omega}^2 = \boldsymbol{M}^{-1}(\boldsymbol{\Phi}_1\boldsymbol{F}_1 + \boldsymbol{\Phi}_2\boldsymbol{F}_2) \tag{2.17}$$

where $F_{th,i}$ is the thruster force of $i$th robot.

Since the manipulators do not move in this case, the acceleration of the robots $\ddot{\boldsymbol{x}}_{robot1}$ and $\ddot{\boldsymbol{x}}_{robot2}$ are related to the vibration acceleration $\ddot{\boldsymbol{q}}$ by the following equation.

$$\ddot{\boldsymbol{x}}_{robot1} = \boldsymbol{\Phi}_1^T\ddot{\boldsymbol{q}} \tag{2.18}$$

$$\ddot{\boldsymbol{x}}_{robot2} = \boldsymbol{\Phi}_2^T\ddot{\boldsymbol{q}} \tag{2.19}$$

By substituting Eq.2.15 and Eq.2.16 into Eq.2.17 using Eq.2.18 and Eq.2.19, the following equation of motion is obtained:

$$\{1 + M^{-1}(m_1\boldsymbol{\Phi}_1\boldsymbol{\Phi}_1^T + m_2\boldsymbol{\Phi}_2\boldsymbol{\Phi}_2^T)\}\ddot{\boldsymbol{q}} + 2\boldsymbol{Z}\boldsymbol{\Omega}\dot{\boldsymbol{q}} + \boldsymbol{\Omega}^2 = \boldsymbol{M}^{-1}(\boldsymbol{\Phi}_1\boldsymbol{F}_{th,1} + \boldsymbol{\Phi}_{th,2}\boldsymbol{F}_2) \tag{2.20}$$

Using Eq.2.20, the state feedback gain $K_{lqr}$ is obtained by solving infinite horizon Riccati equation so that the following performance metric $J$ is minimized.

$$J = \int_0^\infty (\boldsymbol{q}^T \boldsymbol{Q} \boldsymbol{q} + \boldsymbol{F}_{th}^T \boldsymbol{R} \boldsymbol{F}_{th}) dt \tag{2.21}$$

The rigid body mode is controlled by the same PD controller as Section 2.3.3. The thruster force is obtained from the sum of the LQR vibration controller and the rigid-body mode controller outputs.

# Chapter 3

# Simulation

Simulation studies were conducted to demonstrate the ability of the decoupled controller described in the previous section, and to design the experiment described in Chapter 4. For this, a simulation model with the same scale as the FSRL Free-Flying Space Robotics Test Bed was developed and executed. It was also used for the parameter tuning of the experiment.

## 3.1 Settings

In the simulation two robots transported the beam to the destination, which was 0.5 m away from the initial position as shown in Figure 3-1. The entire maneuver took approximately 10 seconds. The profile of the position of the robots is shown in Figure 3-2. The position was controlled by PD controller (Section 2.3.3). However, due to the thruster saturation, the thrust profile was close to bang-bang control. (See Figure 3-5 for the thrust profile.)

Due to the limit of the thrust ($\sim 0.1$ N), visible vibration could not be exited by robot's maneuver alone. Thus, the initial condition, $[q_1(0), \dot{q}_1(0)] = [0, 2.0]$ (1st mode vibration), were set.

Table 4.1 below shows the system parameters used in this simulation. These parameters were taken from the FSRL Free-Flying Space Robotics Test Bed (Figure 1-2).

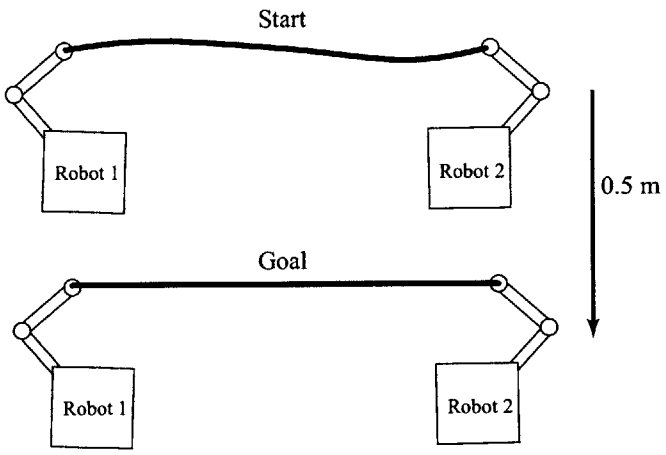


Figure 3-1: Simulation model overview

## 3.2 Performance Comparison

To quantitatively evaluate the performance of the proposed decoupled controller, the simulation results of the following three methods are compared.

(a) No vibration control - Two robots simply transport the beam using thrusters without controlling the vibration. Manipulators are locked.

(b) Vibration control with thrusters - Two robots transports the beam and control the vibration using their thrusters. Manipulators are locked.

(c) Vibration control with manipulators (the proposed algorithm; *the decoupled controller*) - Two robots transport the beam using their thrusters while controlling the vibration using their manipulators.

Figure 3-2 summerizes the timeline of the simulation and the function of thrusters/manipulators of each control method.

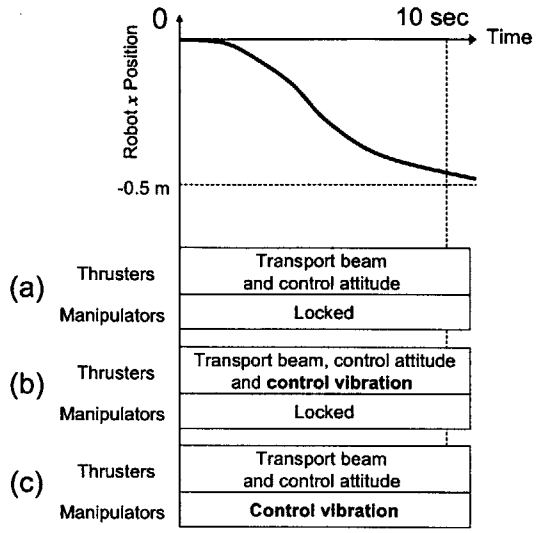


Figure 3-2: Simulation timeline and the function of thrusters and manipulators in each control method

## 3.3 Performance metric

The performance of the three control methods were compared using the following two metrics.

1. **Fuel consumption** - the total amount of the fuel consumed until the robots reached within 3 cm distance from the goal point.

2. **Damping ratio** - the damping ratio of the first mode vibration of the beam.

The damping ratio was defined as $\zeta$ in the following vibration equation and was obtained from the nonlinear least squares curve fitting method. Levenberg-Marquardt algorithm was used for the minimization.

$$x = A\exp(-\zeta\omega t)\sin(\sqrt{1-\zeta^2}\omega t + \phi) \tag{3.1}$$

## 3.4 Result

**Damping Ratio and Fuel Consumption** Figure 3-3 summarizes the damping ratio and the fuel consumption of three control methods. (c) The vibration control

Table 3.1: Parameters of the simulation model

| | | | |
|---|---|---|---|
| Beam | Mass properties and dimensions | Mass | 0.337 kg |
| | | Length | 1.22 m |
| | | Width | 0.128 m |
| | | Thickness | 0.80 mm |
| | Physical properties | Material | Aluminum |
| | | Young Modulus | 70.3 Gpa |
| | | Density | $2.70 \times 10^3$ kg |
| | Natural frequencies | 1st mode | 2.82 Hz |
| | | 2nd mode | 7.77 Hz |
| | Damping ratio | 1st mode | 0.1 |
| | | 2nd mode | 0.1 |
| Robot | Mass properties | Mass | 7.0 kg |
| | | Inertia | 0.040 kg·m$^2$ |
| | Actuators | Maximum thruster force | 0.1 N |
| | | Maximum manipulator force | 1.0 N |
| | Sensors | Accelerometer noise density | $160 \times 10^{-5}$ m/s$^2/\sqrt{Hz}$ |
| | | Force sensor noise density | 1.0 N/$\sqrt{Hz}$ |

with manipulators doubled the damping ratio and consumed almost the same amount of fuel compare to (a), while (b) the vibration control with thrusters made little improvement in damping ratio and consumed significantly larger amount of fuel than (a).

Figure 3-4 shows the accumlated fuel consumption versus time, and Figure 3-5 compares the thruster output force profiles of the three control methods for 20 seconds. (b) Vibration control with thrusters consumed substantially larger amount of fuel than the other two while (c) vibration control with manipulators consumed just slightly more fuel than (a)no vibration control. Fuel consumption rate of (b) vibration control with thrusters did not converge to zero because the thrusters reacted to the residual vibration of the beam.

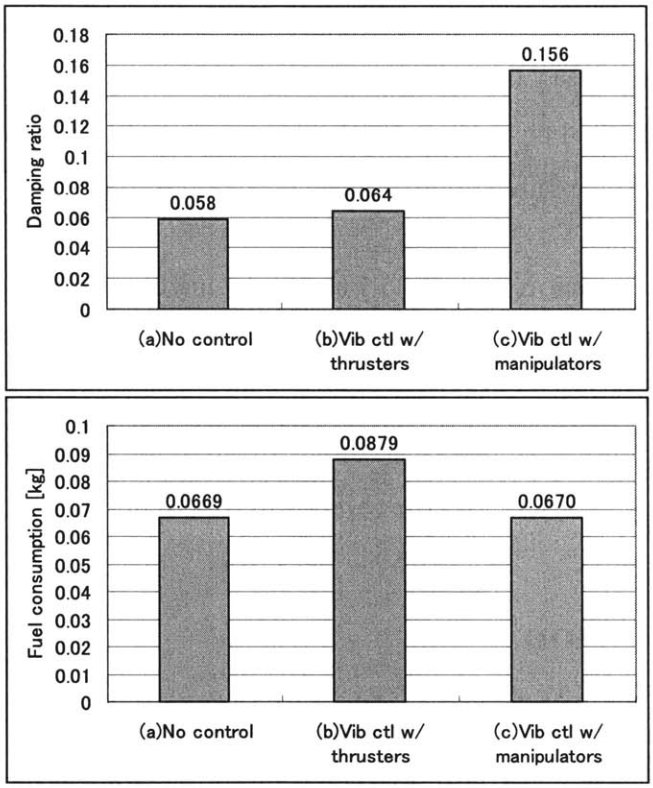


Figure 3-3: Damping ratio and fuel consumption of three control methods

**Vibration** Figure 3-6 compares the magnitude of vibrations with three control methods; (a)no vibration control, (b)vibration control with thrusters, and (c)vibration control with manipulators (the proposed controller). It is obvious that (c)vibration control with manipulators (Figure 3-6-(c)) damps the vibration most quickly. The other two (Figure 3-6-(a) and (b)) does not have visible difference. This is because the thruster has very tiny thrust (0.1N, Table 4.1), and robots are massive compared to the beam mass. A relatively large spike is observed at the begging of the simulation in (c)vibration control with manipulators. This is because the vibration state estimator (Kalman filter) had initial error and Vibration Controller calculated the manipulator force based on the unprecise information. After the estimator converged ($t \approx 0.5$), the controller successfully controlled the vibration.

**Speed of transportation** Figure 3-7 shows the history of the position of the beam center of mass. Transportation speed of (b) vibration control with thrusters is substantially slower than the other two methods. This is because the thrusters have to control the vibration in (b), and thus cannot spend all of its power only to transport the beam (See Figure 3-5 for the thruster output force profile).

## 3.5 Conclusion

The simulation result showed that the proposed decoupled controller has an ability to contol the vibration and transport the beam quickly while requiring less fuel than the vibration control with thrusters. Thus the proposed controller is more efficient than the vibration control with thrusters.

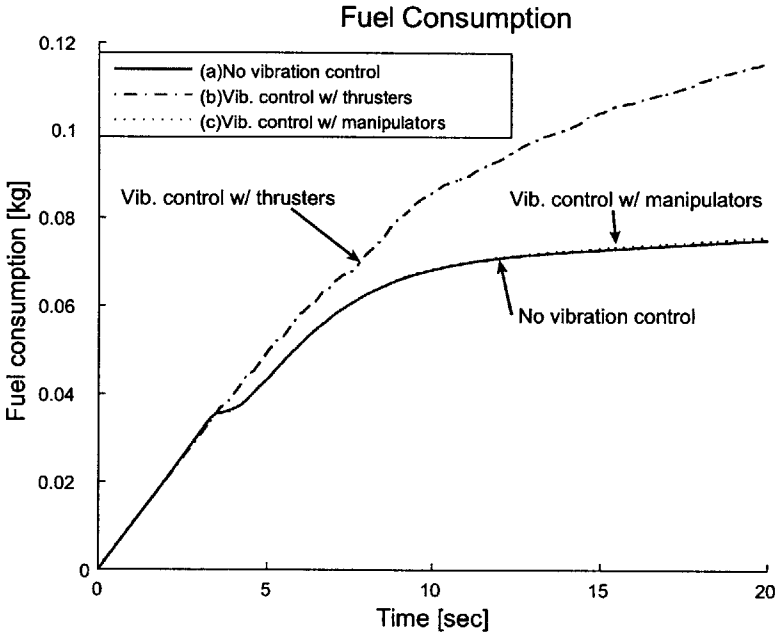


Figure 3-4: Fuel consumption; comparison of three control methods

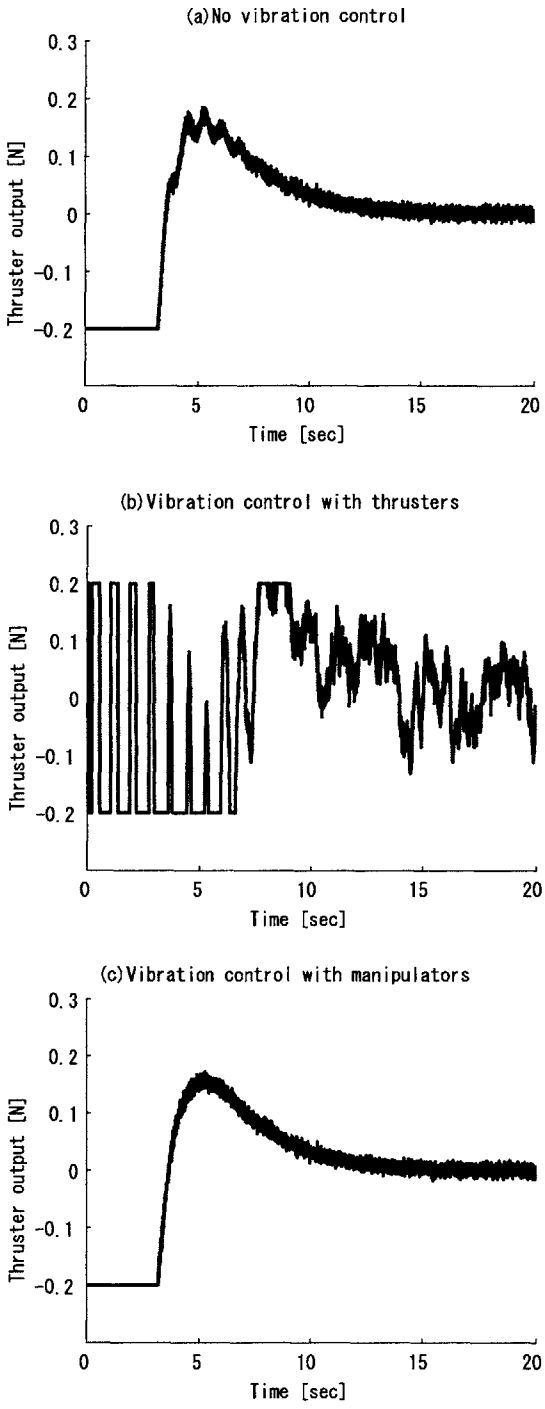


Figure 3-5: Thrsuter output force profile ($x$ direction). (a)Top: No vibration control. (b)Middle: Vibration control with thrusters. (c)Bottom: Vibration control with manipulators.

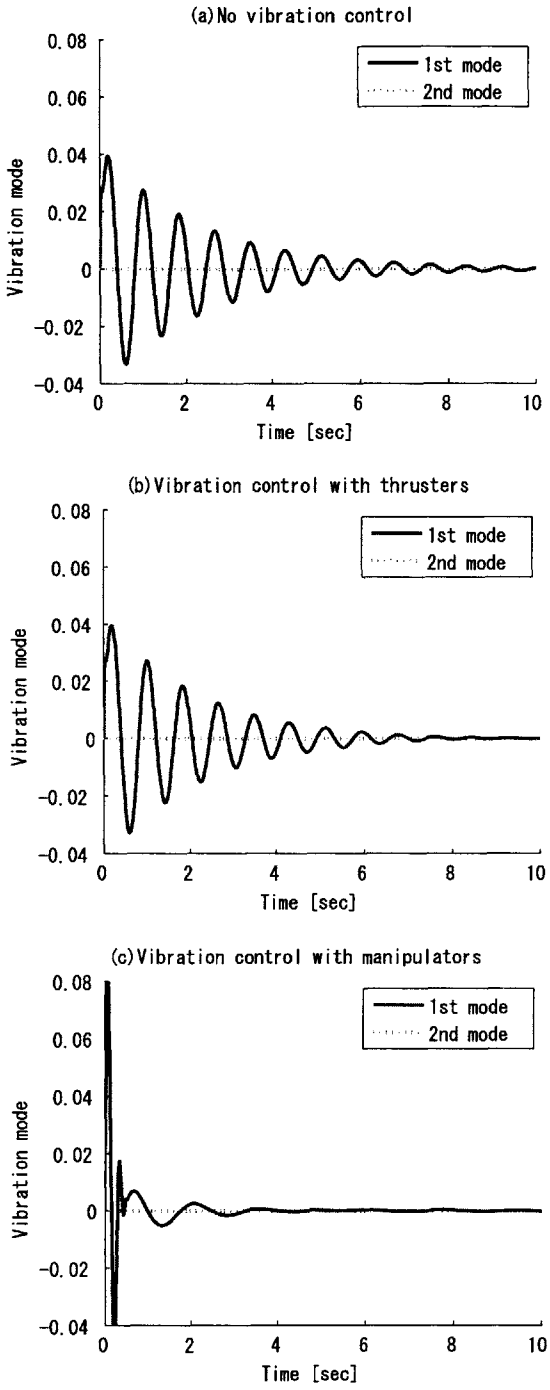


Figure 3-6: Comparison of three control method in terms of vibration. (a)Top: No vibration control. (b)Middle: Vibration control with thrusters. (c)Bottom: Vibration control with manipulators.

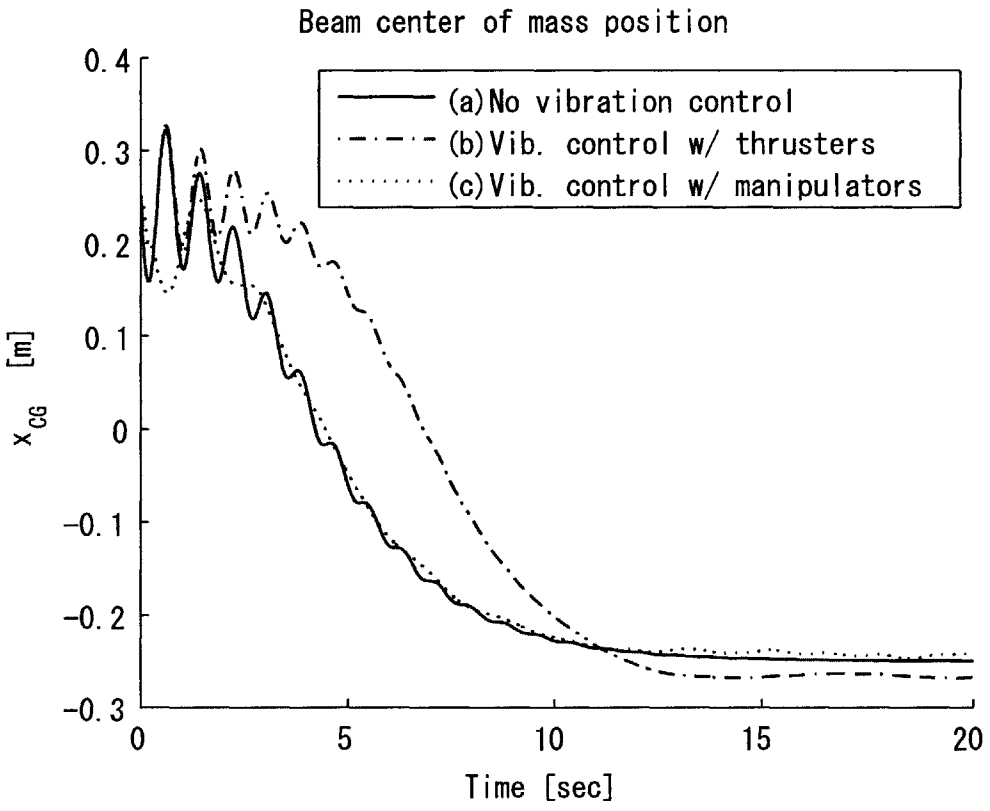


Figure 3-7: Position of the beam center of mass

# Chapter 4

# Experimental Results

To demonstrate the proposed control algorithm, the experimental study was conducted using the Free Flying Space Robotics Test Bed.

## 4.1 Experimental System

This section briefly describes the FSRL Free-Flying Space Robot Test Bed. (See Figure 4-1). It consists of a flat granite table, two free-flying dual-arm robots, and flexible beams. Robots float on the flat table using $CO_2$ air bearing, emulating weightlessness and frictionlessness in 2 dimensional plain.

**Table** A 1.3 m $\times$ 2.2 m granite table with a polished surface is used for the experiments. The MIT Space System Lab has an octagonal flat epoxy floor with a 16 feet diameter, which will be used for larger scale experiments in the future.

**Robots** Figure 4-3 shows an experimental space robot. It is equipped with two manipulators, eight thrusters, two position sensors, four manipulator joint angle encoders, and two force/torque sensors. The robot has 7 DOF in total (2 DOF translation, 1 DOF rotation, and 4 DOF manipulator joints), all of which are controllable and observable. All actuators and sensors are controlled by the on-board computer and powered by on-board batteries, so that the robot can work without any cables

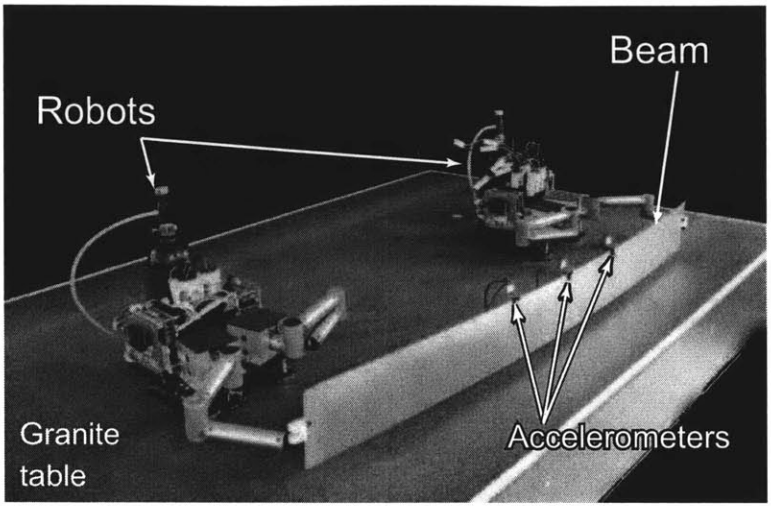


Figure 4-1: Overview of the entire experiment system

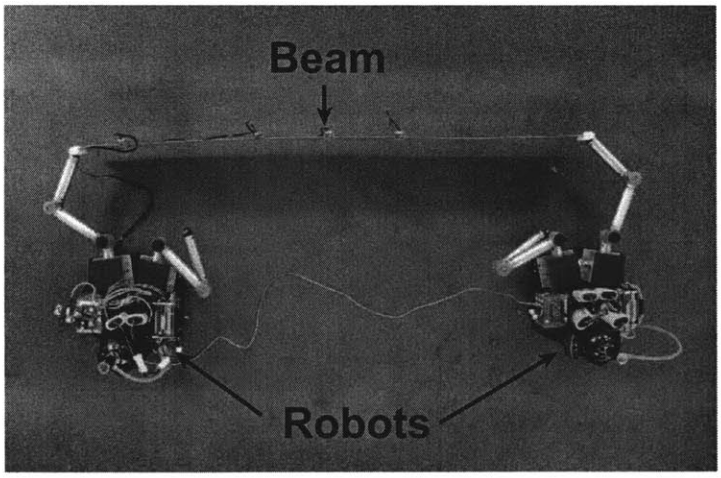


Figure 4-2: Top view of the experiment system

connected to the outside. The robot has a Wireless LAN adapter to give users access to the on-board computer from the outside. The detail of the robot hardware and software is described in Appendix A.

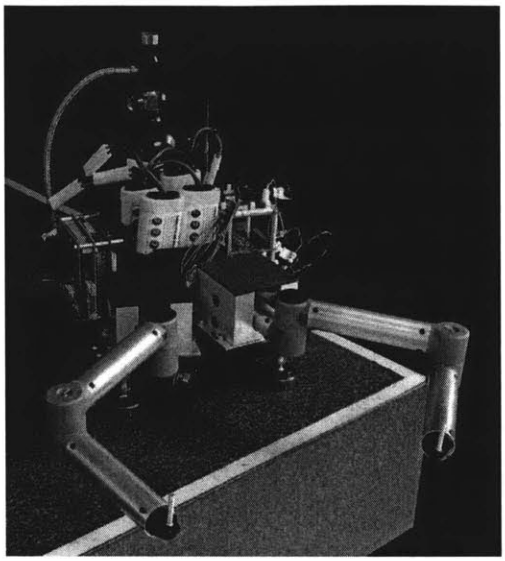

Figure 4-3: An experimental space robot

**Beam** The beam is supported by and pin-jointed to the end effectors of the robots' manipulators, as shown in Figure 4-1. Three accelerometers on the top of the beam are used to measure its vibration. It is made of aluminum, and is 1.22 m long, 0.80 mm thick, and has a 2.8 Hz lowest natural frequency. [11]

## 4.2 Experiment Setting

Robots transported the beam while controlling its vibration. Only the first mode vibration was controlled in this experiment due to the limit of controller bandwidth. Experiments were conducted on two cases; the parallel transportation, and the rotational transportation.

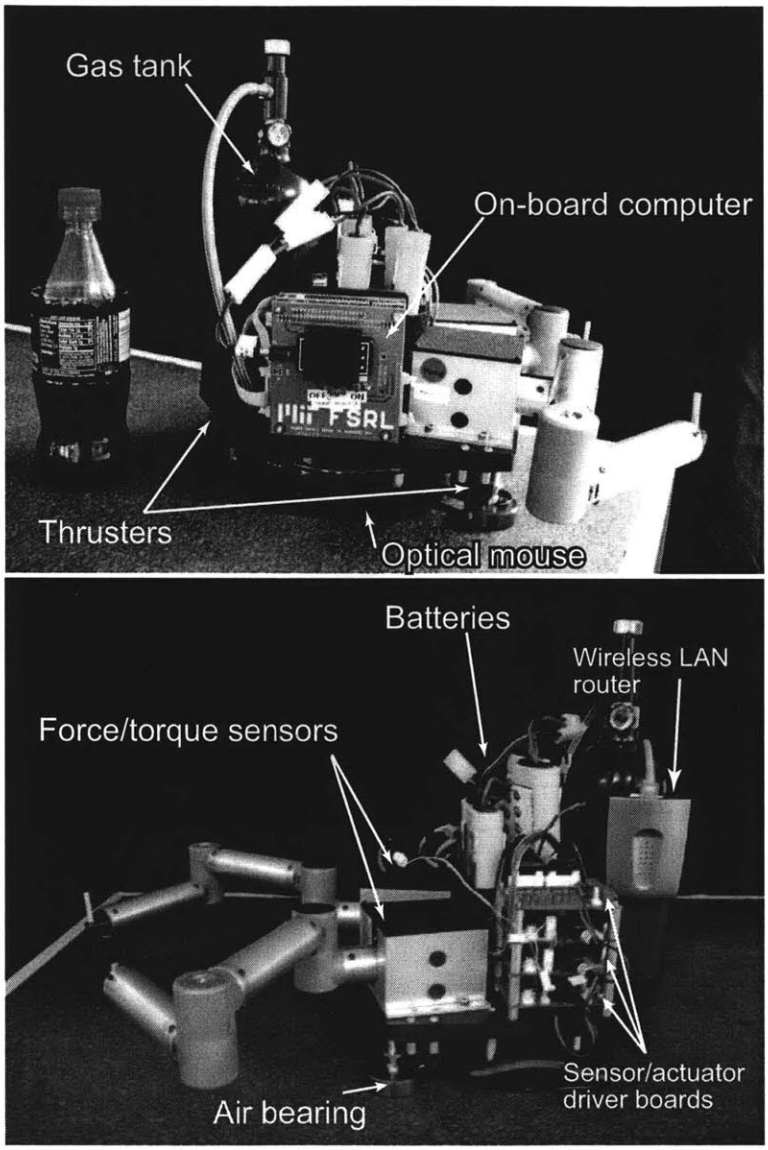


Figure 4-4: Side view of the robot and the placements of components

### 4.2.1 Case 1: Parallel transportation

Two robots held both ends of the beam and moved parallel to each other towards $-X$ direction by 0.5 m, as shown in Figure 4-5. The position of both robots was controlled by PD controller (Section 2.3.3).

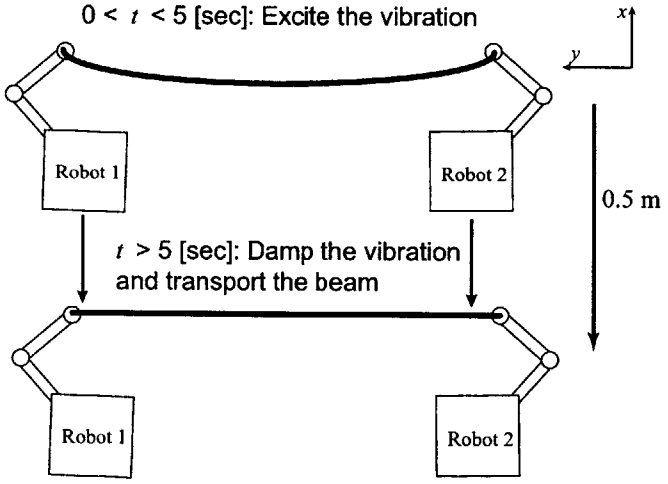


Figure 4-5: Case 1: Parallel transportation

### 4.2.2 Case 2: Rotational transportation

Two robots held both ends of the beam. The right robot rotated -30 degrees around the left robot, as shown in Figure 4-6. The position of the right robot is controlled by PD controller in polar coordinate system around the left robot. The PD controller of the left robot maintained its position and oriented it to the right robot.

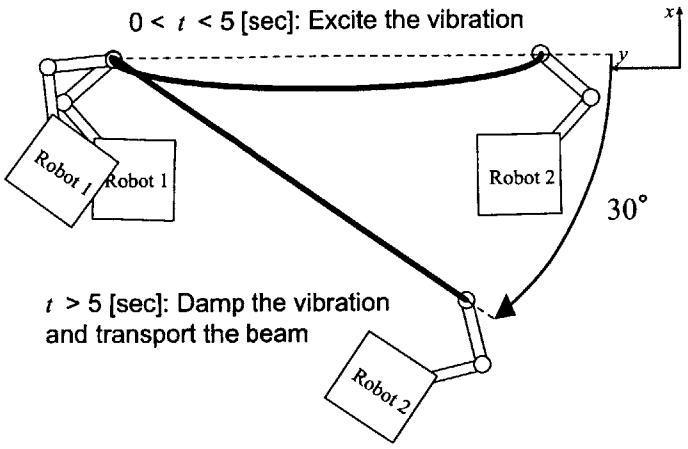


Figure 4-6: Case 2: Rotational transportation

## 4.3 Control Methods

In order to quantitatively evaluate the proposed controller's performance, the following three control methods (same as the ones in the simulation, Section 3.2) were tested with the same experimental settings.

(a) No vibration control - Two robots simply transport the beam using thrusters without controlling the vibration. Manipulators are locked.

(b) Vibration control with thrusters - Two robots transports the beam and control the vibration using their thrusters. Manipulators are locked.

(c) Vibration control with manipulators (the proposed algorithm; *the decoupled controller*) - Two robots transport the beam using their thrusters while controlling the vibration using their manipulators.

These will be referred as "the experiment (a), (b), and (c)" in this chapter.

**Experiment Timeline** Figure 4-7 shows the timeline of the experiment (a), (b), and (c). Due to the limit of the thrust ($\sim 0.1$ N), visible vibration can not be exited by robot's maneuver alone. Thus, vibration was actively excited using the two robot's manipulators for the first 5 seconds of the experiment.

The manipulators were given a preprogrammed sinusoidal motion with a frequency of 1.0 Hz for the first 5 seconds. Meanwhile, the robots' position and orientation were maintained by thrusters. Then at $t = 5$ sec, the vibration controller was turned on in experiment (b) and (c). In experiment (b) the thrusters controlled the vibration and transported the beam while the manipulators were locked. In experiment (c) the manipulators controlled the vibration, and the thrusters transported the beam. In experiment (a), the manipulator joints were locked after $t = 5$ sec while the thrusters transported the beam.

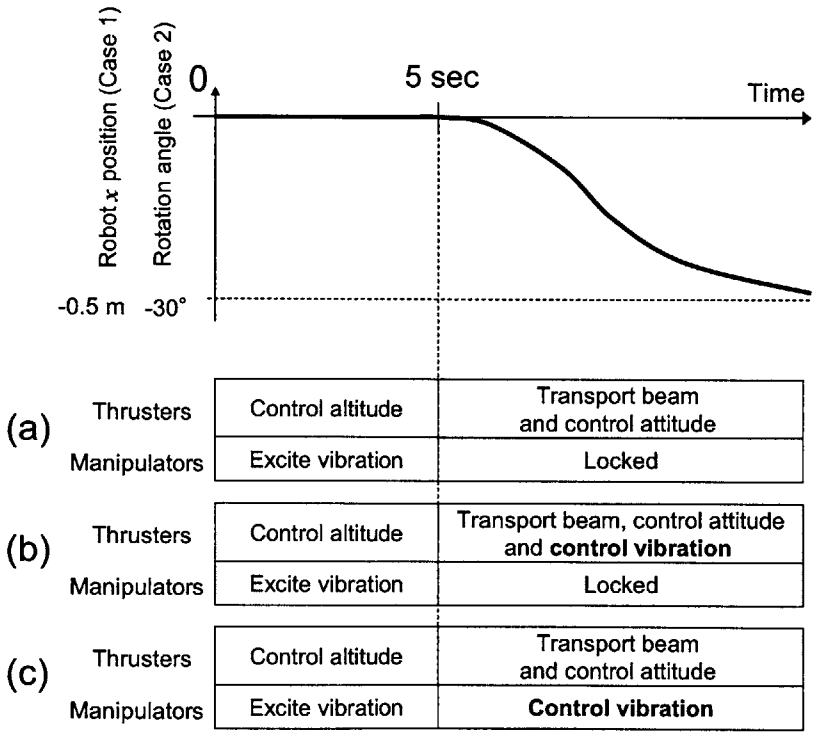


Figure 4-7: Experiment timeline

**Objective Function** In the experiment (b) and (c), the parameters in the objective function (Eq. 2.11 and Eq. 2.21) were set as follows. See Section 2.3.2 and 2.4 for the definition of $\boldsymbol{x}$ and $\boldsymbol{u}$.

$$J = \int_0^\infty (\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u}) dt \tag{4.1}$$

$$Q = \begin{pmatrix} 10^4 & 0 & 0 & 0 \\ 0 & 10^4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.2}$$

$$R = \begin{pmatrix} 10^3 & 0 \\ 0 & 10^3 \end{pmatrix} \tag{4.3}$$

**Parameters** The system parameters used in the experiment is shown in Table 4.1.

Table 4.1: Parameters for the experimental system

| | | | |
|---|---|---|---|
| Beam | Mass properties and dimensions | Mass | 0.337 kg |
| | | Length | 1.22 m |
| | | Width | 0.128 m |
| | | Thickness | 0.80 mm |
| | Physical properties | Material | Aluminum |
| | | Young Modulus | 70.3 Gpa |
| | | Density | $2.70 \times 10^3$ kg |
| | Natural frequencies | 1st mode | 2.82 Hz |
| | | 2nd mode | 7.77 Hz |
| Robot | Mass properties | Mass | 7.0 kg |
| | | Inertia | 0.040 kg·m$^2$ |
| | Actuators | Maximum thruster force | 0.1 N |
| | | Maximum manipulator force | 0.6 N |
| | Sensors | Accelerometer noise density | $160 \times 10^{-5}$ m/s$^2/\sqrt{Hz}$ |
| | | Force sensor noise density | 1.0 N/$\sqrt{Hz}$ |

## 4.4 Performance metric

The performance of the three control methods were compared using the following two metrics.

1. **Fuel consumption** - the total amount of the fuel ($CO_2$ gas) consumed between $t = 5$ sec (the start of the vibration control) and the time when the robots reached within 3 cm distance (Case 1) or 2° (Case 2) from the goal point.

2. **Damping ratio** - the damping ratio of the first mode vibration of the beam.

The fuel consumption was estimated from the thruster command in the controller log. The thruster's specific impulse ($I_{SP}$) was assumed to be 10 sec. It did not inculde $CO_2$ gas used to float the robots.

The time-series vibration amplitude data was obtained from the Kalman filter. The damping ratio was defined as $\zeta$ in the following vibration equation and was obtained from the nonlinear least squares curve fitting method. Levenberg-Marquardt algorithm was used for the minimization.

$$x = A\exp(-\zeta\omega t)\sin(\sqrt{1-\zeta^2}\omega t + \phi) \tag{4.4}$$

## 4.5 Results

### 4.5.1 Damping Ratio and Fuel Consumption

Ten experiments were conducted for each control method in both cases. Table 4.2 and Table 4.3 show the average damping ratio and fuel consumption. The first column of the table shows the total fuel consumption. The total fuel consumption of (a) no control can be regarded as the fuel used for rigid-body mode control. Thus, the difference of the total fuel consumption between (a) and (b), and (a) and (c) is the fuel used for the vibration control, which is shown in the second column of the table.

Figure 4-8 compares the damping ratio and fuel consumption of three control methods for both cases. The boxes indicate the average while the bars indicate the

standard deviation.

Table 4.2: Performance comparison of the three control methods (Case 1) (Average of 10 runs)

| **Case 1** | Total fuel consumption [g] | Fuel used for vibration control [g] | Damping ratio |
|---|---|---|---|
| (a) No control | 84.4 | 0 | 0.117 |
| (b) Vib control by thrusters | 103.3 | 18.9 | 0.081 |
| (c) Vib control by manipulators | 93.0 | 8.6 | 0.219 |

Table 4.3: Performance comparison of the three control methods (Case 2) (Average of 10 runs)

| **Case 2** | Total fuel consumption [g] | Fuel used for vibration control [g] | Damping ratio |
|---|---|---|---|
| (a) No control | 64.1 | 0 | 0.116 |
| (b) Vib control by thrusters | 79.6 | 15.5 | 0.085 |
| (c) Vib control by manipulators | 73.4 | 9.3 | 0.189 |

Approximately the same results were obtained from both cases. The damping ratio of (c) was about the double of (a) and (b). The damping ratio of (b) was even smaller than (a), which means that the vibration controller with thrusters degraded the performance of the vibration control.

Although the total fuel consumptions (the first column of Table 4.2 and 4.3) did not differ significantly, the fuel used for vibration control in (c) was significantly smaller than (b). This was because the robots were very heavy ($\sim$ 7 kg each) while the beam was thin and light ($\sim$ 0.3 kg each), so most of the energy of the thrusters went to the rigid-body mode control.

This result generally agreed with the simulation result shown in Figure 3-3, except two points. The first one was that the damping ratio of (b) was smaller than (a) in the experiment while they were almost the same in the simulation. This implies that some part of the experimental hardware was not modeled properly in the simulation. Finding a better simulation model is the future work. The other difference between

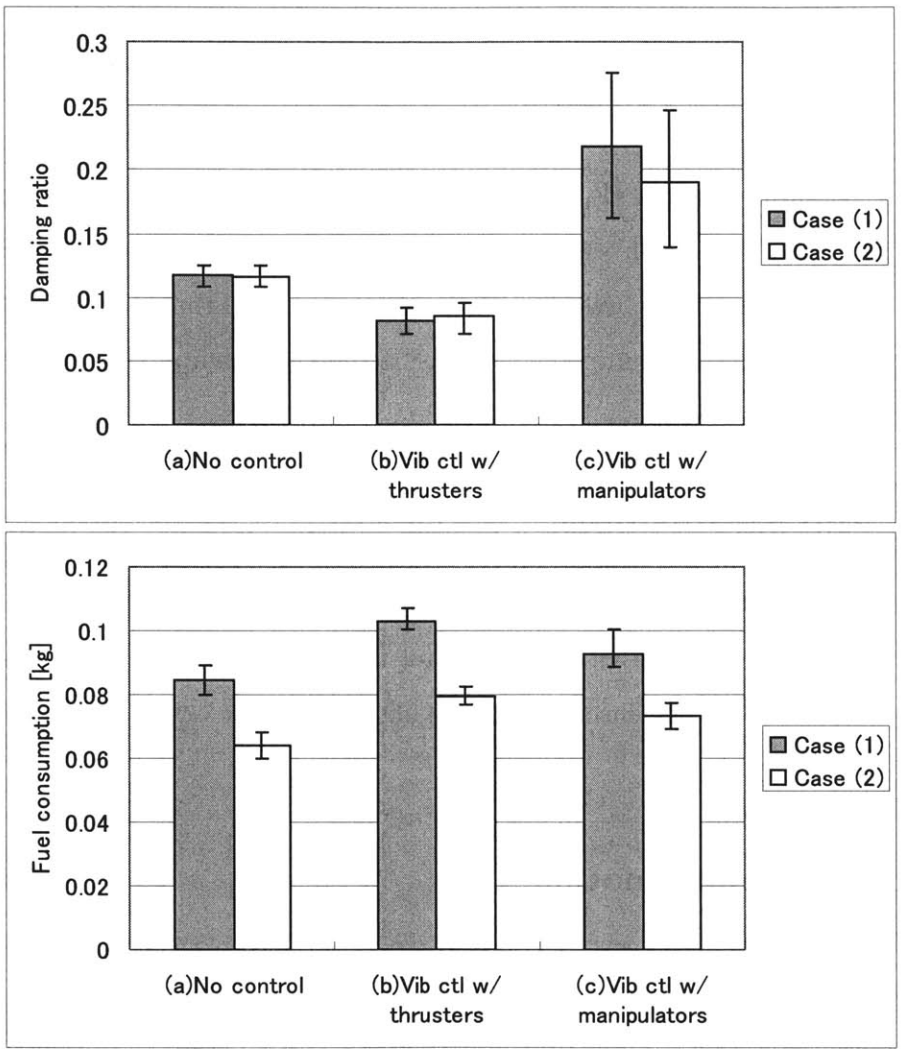


Figure 4-8: Damping ratio and fuel consumption of three control methods for both cases

the experiment and simulation was that (c) consumed more fuel than (a) in the experiment while the fuel consumption of (a) and (c) was almost the same in the simulation. In the experiment, the attitude of the robots in (c) was not as stable as in the simulation due to the disturbance and actuation error. A substantial amount of fuel was consumed to maintain the attitude of the robots, which made this difference.

### 4.5.2 Vibration

Figure 4-9 shows the first mode vibration of the three control methods in the Case 1. In Experiment (c), vibration was damped in the shortest amount of time. However, relatively larger disturbance was observed in experiment (c) than experiment (a) and (b) after the vibration was damped. This was because the manipulator force controller reacted to the sensor noises.

### 4.5.3 End effector force

Figure 4-10 shows the end effector force applied to the beam in the experiment (c) of the Case 1. Most of the vibration control maneuver was done in the first three seconds. Reactions to the sensor noise were observed as the occasional spikes in this graph.

### 4.5.4 Rigid-body motion

Figure 4-11 shows the position of the Robot 1 in the experiment (c) of the Case 1. Robot's rigid body mode was successfully controlled by thrusters while vibration was controlled by manipulators.

## 4.6 Conclusion

The experiment results on both cases showed that the proposed decoupled controller has an ability to contol the vibration quickly while requiring less fuel than the vibration control with thrusters. Thus the proposed controller is more efficient than the

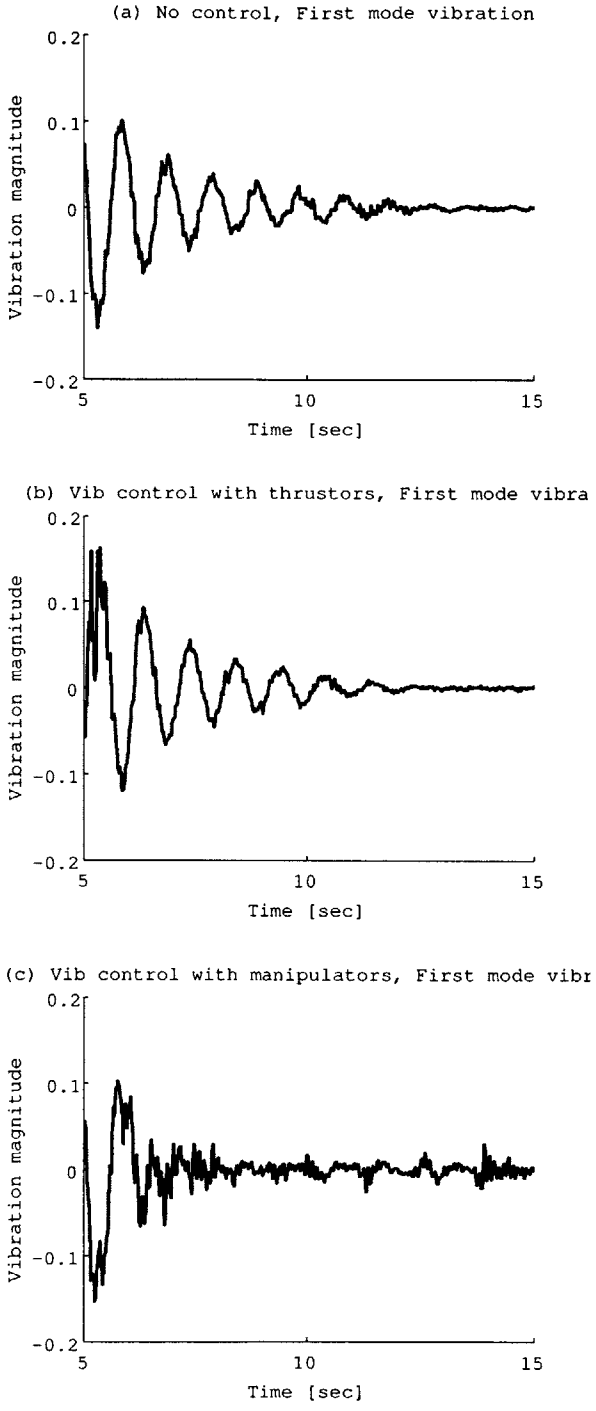


Figure 4-9: The first mode vibration of the experiment. Top: (a)No vibration control. Middle: (b)Vibration control with thrusters. Bottom: (c)Vibration control by manipulators

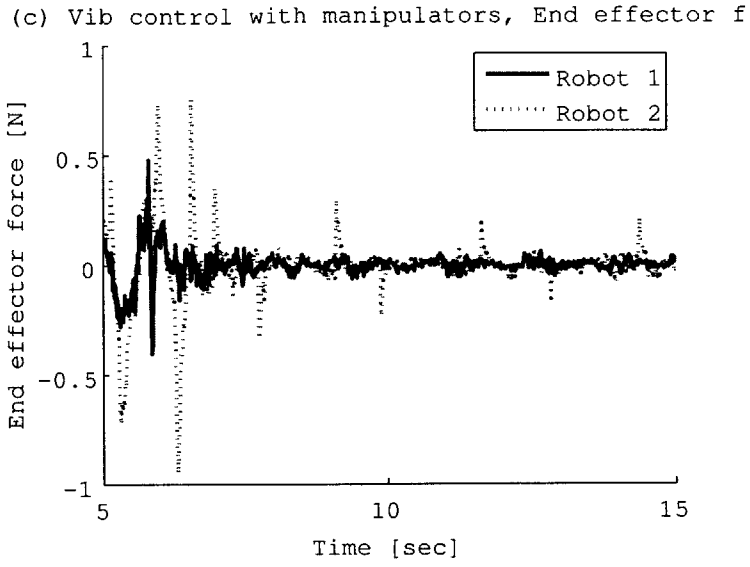


Figure 4-10: End effector force applied to the beam in the experiment (c): Vibration control by manipulators

vibration control with thrusters.

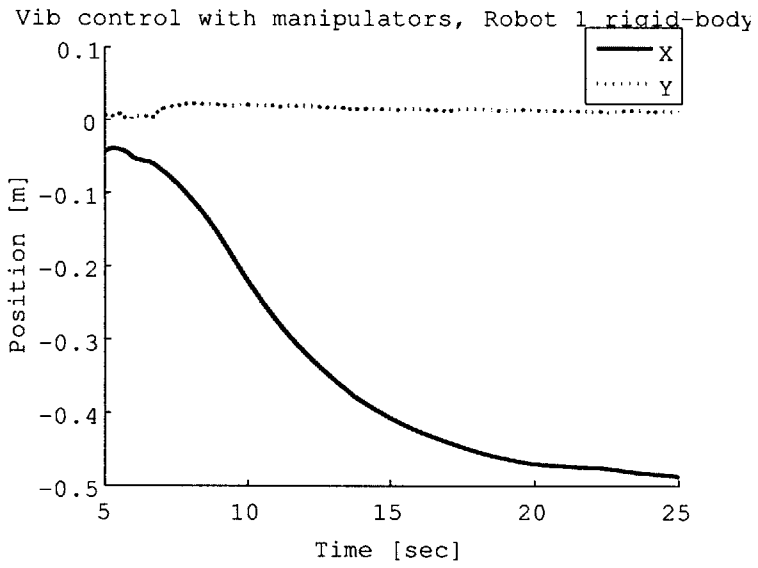


Figure 4-11: Rigid-body motion of the Robot 1 in the experiment (c): Vibration control by manipulators

# Chapter 5

# Summary & Conclusions

This work experimentally validated a large space structure transportation algorithm, which actively controlled the vibration of the structure using space robots' manipulators. The algorithm was shown to double the damping ratio than the classical thruster-based vibration control, while it consumes less amount of fuel.

In Chapter 1, the motivation for this work and relevant background literature were presented. Space robotics is a promising substitution for expensive and dangerous human operation on orbit. Among the various robotic operations expected in the future, this work especially focused on the transportation of the large, flexible space structures. One of the key engineering issues is the vibration control during the transportation. In order to address the issue, the robots' manipulators were used to control the vibration during the transportation of the structure.

Chapter 2 presented the theory of decoupled controller. It controls the vibration mode using manipulators while controlling the rigid body mode by thrusters. LQR algorithm is used for the vibration controller while classical PD controller is used for the rigid-body mode controller. The manipulator compliance controller is introduced to bridge the vibration controller and the rigid-body mode controller.

Chapter 3 presented the simulation result. The performance of the presented controller is compared with other two approaches: (a) no vibration control and (b) the vibration control with thrusters. The simulation result showed that the decoupled controller can damp the vibration quicker than (a) and (b) while consumes less amount

of fuel than (b).

Chapter 4 presented the experimental result, which mostly agreed with the simulation result. The experiment was conducted using FSRL Free Flying Space Robotics Test Bed. The test bed consists of a flat granite table, two free-flying dual-arm robots, and flexible beams. The Robots float on the flat table using $CO_2$ air bearing, emulating weightlessness and frictionlessness in 2 dimensional plain. The performance of the decoupled controller is compared with other two control methods; (a) no vibration control and (b) the vibration control with thrusters. The experiments were conducted on two cases; the parallel transportation and the rotational transportation. The results of both cases showed that the decoupled controller doubled the damping ratio compared to (a) and (b), and consumes less amount of fuel than (b).

In conclusion, the proposed large space structure transportation algorithm is more efficient than existing approaches in terms of the fuel consumption and the reaction time to damp vibration damping. Using this algorithm, the construction of the large space structure can be done with less amount of fuel, which is a limited and expensive resource on the orbit. By combining this work with other space robotics work such as [6][13][14][17][24], space robots can be more capable, efficient, and intelligent than now, which will contribute improving the human civilization by enabling easier and more inexpensive development of the space.

# Appendix A

# Experiment System Details

This chapter describes the detail of the experiment system design and how to use it.

## A.1 Experimental Space Robots

The free-flying robots are the most complicated parts of the experimental system. Each robot is equipped with actuators and sensors to control and observe its 7 DOF motion. It has a computer and the batteries on board, so that it can operate without any support from outside, emulating an actual on-orbit space robot.

All components of the robot and their placement are shown in Figure A-1.

### A.1.1 Robot Subsystems

**Air Bearings** Three round air bearings support the robot's mass. Pressurized $CO_2$ gas flows out of the bottom of the air bearings, which creates a thin $CO_2$ layer between the table and the bearing. The robots float on the table with little friction. The $CO_2$ gas is stored in a tank on each robot.

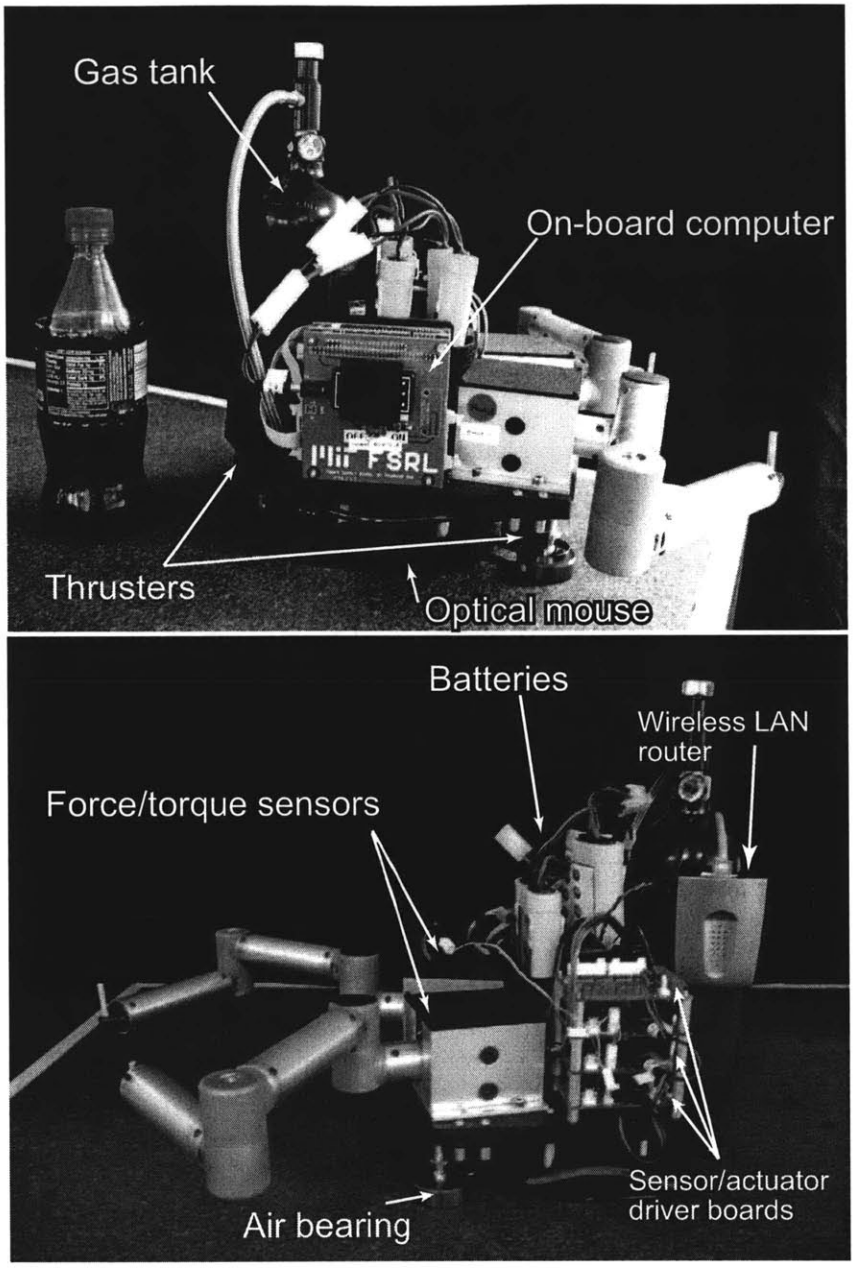


Figure A-1: Side view of the robot and the placements of components

**Thrusters** Each robot has eight cold gas thrusters, which enables 3 degree of freedom maneuvers. Each thruster has $\sim 0.1$ N thrust, though it has not been precisely measured yet. Thrusters share the $CO_2$ gas tank with the air bearings. They are controlled by eight solenoid valves, which can only turn on or off (no intermediate

state). Intermediate thrust can be generated by pulse width modulation (PWM).

The eight thrusters are clustered in three locations at the bottom of the robot as shown in Figure A-2. Arrows in the figure indicate the placement and the orientation of the thrusters.

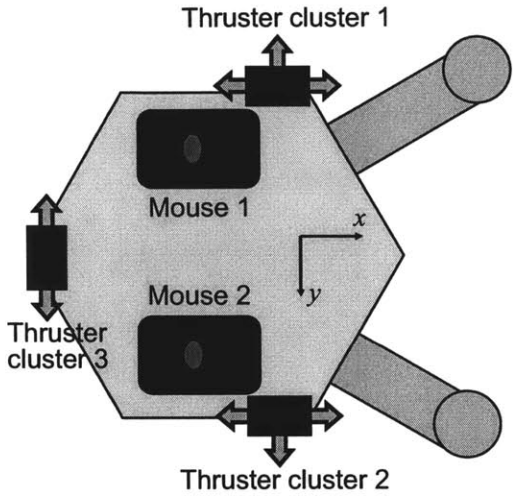


Figure A-2: Bottom view of the robot; placement of mice and thrusters

**Manipulators and Encoders** Each robot is equipped with two two-joint manipulators, adding 4 DOF to the robot. Each manipulator is powered by two DC motors, and its joint angles are read by two optical angle encoders.

**Force/Torque Sensors** Force/Torque Sensors are placed at the base of each manipulator. Figure A-3 provides the overview of the architecture of the sensor. It measures 3 DOF forces and torque ($F_x$, $F_y$, and $T_xy$) by using four strain gauges. Most part of the forces and torques applied to the sensor is carried by four flexures, which deforms linearly with the magnitude of forces and torque. The deformation is measured by four strain gauges installed along with the flexures, from which the forces and torque applied to the manipulator are estimated.

See Section ?? for the detail about the measurement and the calibration of the sensor.

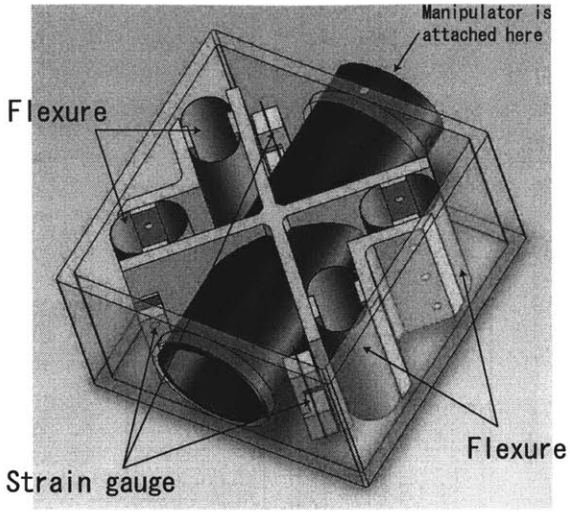


Figure A-3: Force/torque sensor architecture

**Position and Velocity Sensors** Two optical mice are placed at the bottom of the robot as Figure A-2 to measure the position, orientation, and velocity of the robot.

### A.1.2 Electronics

The robot's electronics consists of an on-board computer, several sensor/actuator driver boards, and CAN (Controller Area Network) communication bus. Figure A-4 is the electronics diagram which shows all the electronics components and the connections.

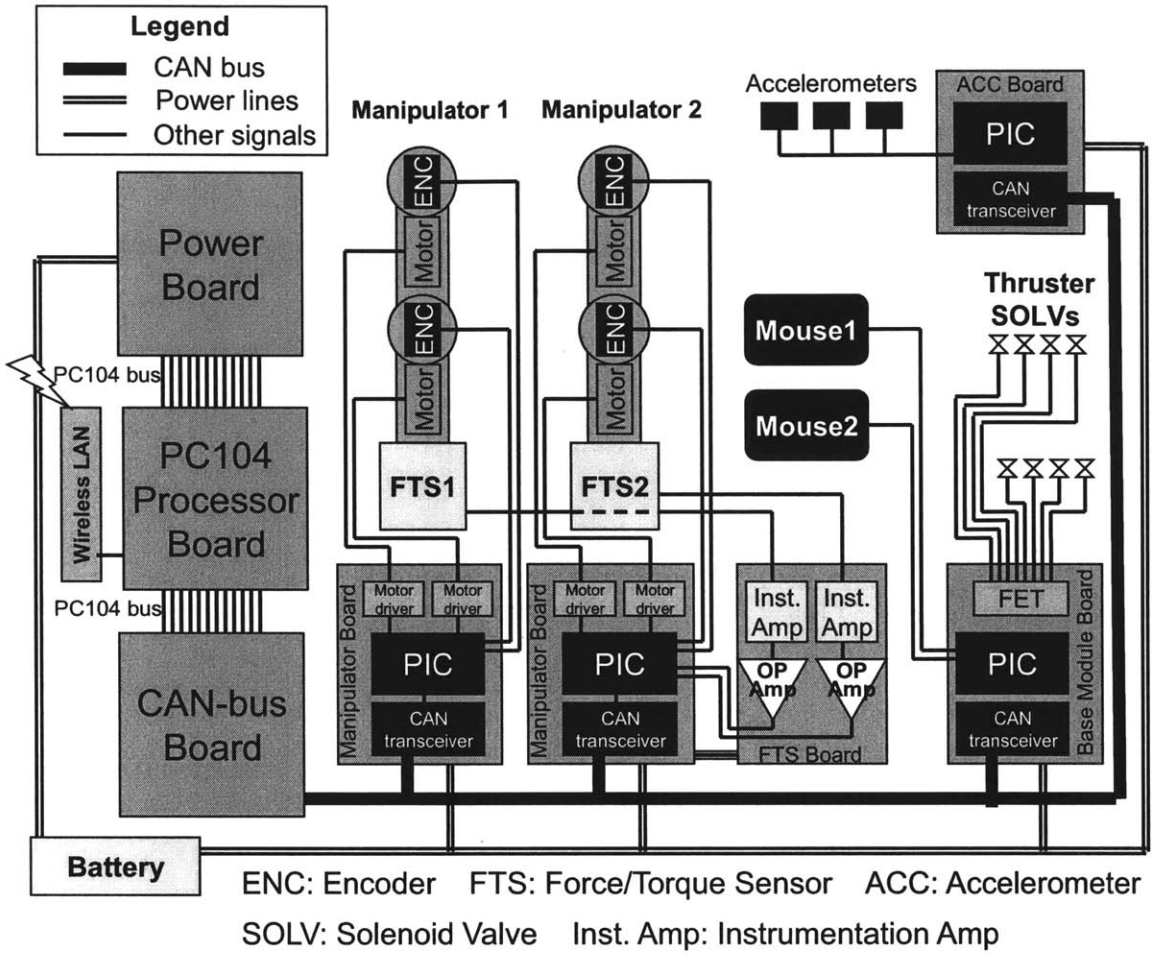


Figure A-4: Robot's Electronics Diagram.

**On-board Computer** The robot's on-board computer is the Diamond Systems Morpheus PC/104 System with 400MHz Intel®ULV Celeron®Processor and 128 Mbyte RAM. It is an PC/AT compatible machine with ISA expansion bus (PC/104 bus), on which a CAN-bus board (Softing CAN-AC2-104) is placed. It is also equipped with Wireless LAN router to allow wireless access to the on-board computor from outside.

**Sensor/Actuator Driver Boards** Four kinds of custom electronics boards were used to drive on-board sensors and actuators; manipulator boards, force/torque sensor boards, base module boards, and accelerometer boards. All of them except the

force/torque sensor board have Microchip PIC18F6680 MCU (Micro Controller Unit) for the operation of the sensors and actuators.

The manipulator board controls DC motors and reads the digital angle encoders. The motor's speed is controlled by PWM (pulse width modulation) and the rotation direction is controlled with the H-bridge circuit. The H-bridge circuit also serves as electromagnetic brake of the motors. The signal from the optical angle encoders are interpreted by counter IC and then read by PIC MCU.

The base module board controls thrusters and reads mice. The thrusters' solenoid valves are opened and closed by FETs (field effect transistors), whose gate are controlled by PIC's IO port. The data from the mice are directly fed to PIC and interpreted.

The force/torque sensor board has instrument amps and operational amps to amplify the signal of the force/torque sensors. Amplified signals are A/D converted by the PIC on manipulator boards.

The accelerometer board interprets the digital PWM signal from Accelerometers on the beam.

### A.1.3 Software

The real-time software of the on-board computer runs on Matlab xPC Target. It is developed by Simulink and compiled in remote PC, and then loaded on the on-board computer.

The controller cycle is 200 Hz. The sampling time of the encoders, the force/torque sensors, and the motors is also 200 Hz, while it is 50 Hz for accelerometers and 20 Hz for mice.

## A.2 CAN-bus

A CAN (Controller Area Network) bus is utilized for the communication between the on-board computors and MCUs.

In the experiment described in Chapter 4, one on-board computer controls two robots. Therefore two robots are also connected with CAN-bus to allow inter-robot communication.

# Appendix B

# Force/Torque Sensor Manual

This chapter explains how to calibrate and use the force/torque sensor for the experimental space robots.

First you have to define the zero-point angles of the manipulator joints (See Section B.1).

Then there are three things you may have to do before using the sensor.

(a) **Gain and offset correction** (Section B.2); set the proper gain and offset of the force/torque sensor board by adjusting the potentiometers.

(b) **Full calibration** (Section B.4.1); calibrate all sensor parameters using the FSRL Force/Torque Sensor Calibration Equipment.

(c) $\boldsymbol{H'_{out}}$ **calibration** (Section B.4.3); calibrate only the zero-point and $H'_{out}$ (See Section B.3, Eq B.6 for the definition) *without* using the FSRL Force/Torque Sensor Calibration Equipment.

When you install a new force/torque sensor, install a new force/torque sensor board, or changed the wirings, you must perform a (a) gain and offset correction followed by a (b) full calibration.

To maintain the precision of the sensor, a (c) $H'_{out}$ calibration should be done at least once in a week.

When you attach a beam or some objects to the end effector, you have to conduct a (c) $H'_{out}$ calibration with the objects attached to the end effector, or load the past calibration data. See page 68 for the reason.

When you find that the sensor measurement is not accurate, you should first try a (c) $H'_{out}$ calibration. If it does not solve the problem, then try a (b) full calibration.

It is recommended to read Section B.3 to understand the theory of measurement and calibration before you read the Section B.4.1 - B.4.3.

## B.1 Zero-point Angle Definition

The zero-point angles of the manipulator joints are defined as the reference points of the measurement. In this work, the zero-point angles are defined as $[\theta_1, \theta_2] = [-25°, 125°]$ for the right manipulator and $[25°, -125°]$for left manipulator. See Figure B-3 for the definition of $\theta_1$ and $\theta_2$.

Zero-point angles may be defined differently as needed. Generally, the offset error is large when the joint angles are far from the zero-point angles. Thus the zero-point angles should be chosen as a point near the center of the expected range of motion.

## B.2 Gain and Offset Correction

As is explained in Section A.1.1, the force/torque sensor is made of four strain gauges, each of which outputs voltage potential difference proportional to the strain. The voltage potential difference is amplified by the force/torque sensor board (Figure B-1) and then A/D converted by the manipulator board. The force/torque sensor board has eight channels. Each channel has two potentiometers (blue boxes) to adjust the gain and offset of the amplifier. The following equation is the model of the signal amplification by the force/torque sensor.

$$V_{out} = V_{offset} + KV_{in} \tag{B.1}$$

where $V_{in}$ is the output of the strain gauge (input to the board) and $V_{out}$ is the output of the board. Two potentiometers adjust $K$ and $V_{offset}$. Note that there is a coupling between $K$ and $V_{offset}$ in the actual hardware (i.e. Adjusting the gain ($K$) potentiometer also affects $V_{offset}$).

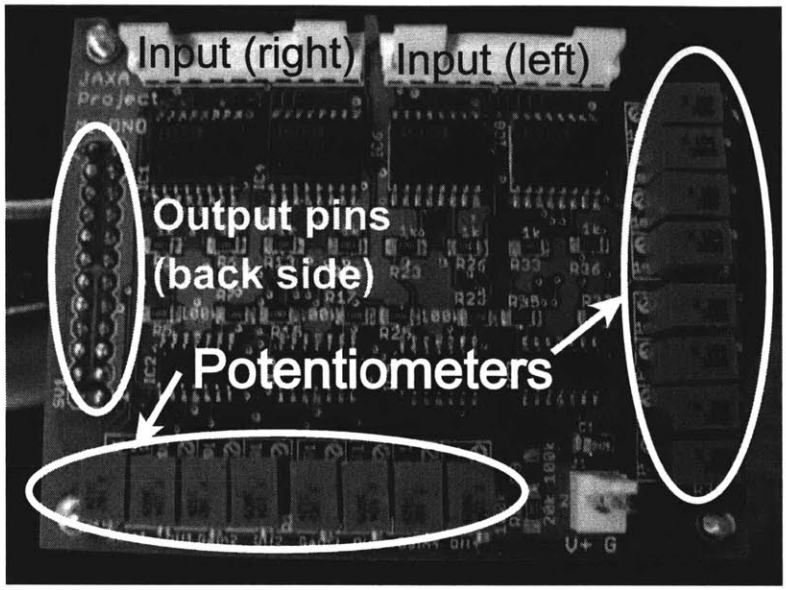


Figure B-1: Force/torque sensor board

Both potentiometers have to be adjusted so that the output is in the range $0[V] < V_{out} < 5[V]$. $V_{out}$ should be about 2.5 V when the manipulate joints are at the zero-point angles (Section B.1). Bring the manipulators joints to the zero-point angles, watch the output voltage using oscilloscope, and adjust the potentiometers as shown in Figure B-2. Table B.1 shows the pin assignment of the force/torque sensor board.

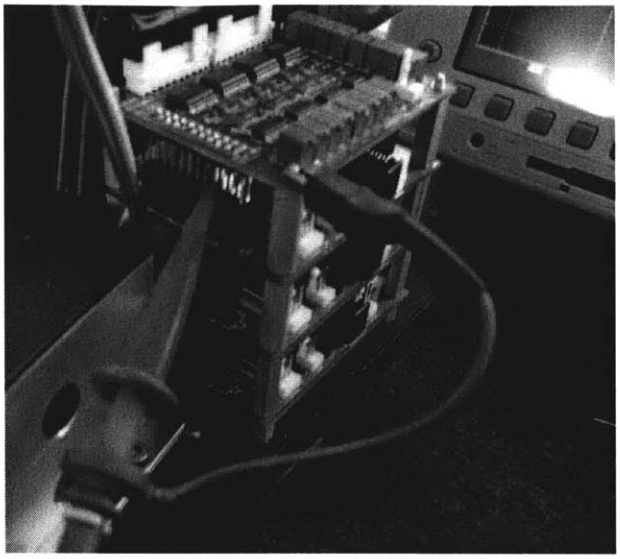

Figure B-2: Gain and Offset Correction

## B.3 Force/Torque Sensor Theory of Measurement and Calibration

### B.3.1 Sensor Model

#### Basic Sensor Model

A linear sensor model of a force/torque sensor is assumed.

$$\boldsymbol{z} = \boldsymbol{z}_0 + \boldsymbol{H}\boldsymbol{f} \tag{B.2}$$

where $\boldsymbol{z} = [z_1 \quad z_2 \quad z_3 \quad z_4]^T$ are the sensor outputs (the measurements of four strain gauges), $\boldsymbol{z}_0$ is the zero-point output vector, and $\boldsymbol{f} = [F_x \quad F_y \quad T_{xy} \quad F_z \quad T_{yz} \quad T_{zx}]^T$ is the input force and torque. Since $z_i$ is the output of 10-bit AD converter, its range is from 0 to 1023.

#### Out-of-plane Force Coupling

In Eq.B.2, $\boldsymbol{f}$ contains not only the in-plane forces and torque ($F_x$, $F_y$, and $T_{xy}$) but also the out-of-plane force and torques($F_z$, $T_{yz}$, and $T_{zx}$). See Figure B-3 for the

Table B.1: Pin Assignment of Connector JX of the force/torque sensor

| Pin # | Fuction |
|---|---|
| 1-2 | GND |
| 3-4 | +5V |
| 5 | Channel 1 out (sensor 1-1) |
| 6 | Channel 2 out (sensor 1-2) |
| 7 | Channel 3 out (sensor 1-3) |
| 8 | Channel 4 out (sensor 1-4) |
| 9 | Channel 5 out (sensor 2-1) |
| 10 | Channel 6 out (sensor 2-2) |
| 11 | Channel 7 out (sensor 2-3) |
| 12 | Channel 8 out (sensor 2-4) |
| 13-16 | Not Connected |

definition of the coordinates. Only the in-plane force is to be measured, but the effect of the out-of-plane force to the output is not negligible. Due to this coupling of the out-of-plane force, it is impossible to calculate the input force $\boldsymbol{f}$ from the sensor output $\boldsymbol{z}$ by simply multiplying with the pseudo-inverse of $\boldsymbol{H}$, because the dimension of $\boldsymbol{f}$ (six) is larger than the dimension of $\boldsymbol{z}$ (four).

This problem can be solved by using the joint angles of the manipulator. The effect of the out-of-plane force can be predicted from joint angles, if the following two assumptions are true.

**Assumption 1** The load to the end effector is constant.

**Assumption 2** No external out-of-plane torque are applied to the manipulator.

Assumption 1 is true for the experiments in this work. Assumption 2 is also true, since the beam is pin-jointed to the end effector. With these assumptions, the only source of out-of-plane force and torques is the mass of the manipulator and the mass of the object attached to the end effector. Thus $F_z$ is constant, and $T_{yz}$ and $T_{zx}$ are functions of joint angles given in Eq. B.3.

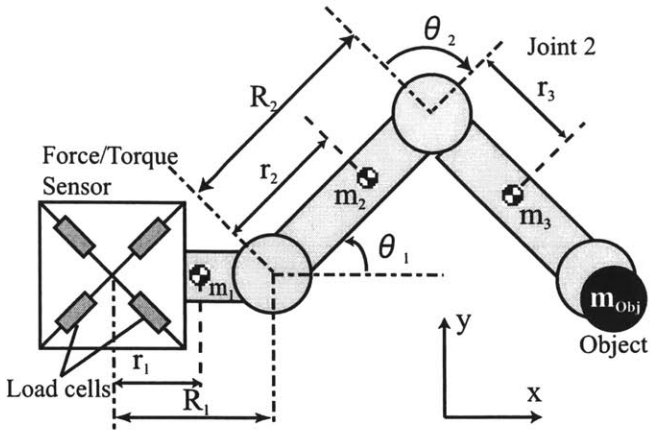


Figure B-3: Manipulator and force/torque sensor model

$$\boldsymbol{f}_{out} = \begin{pmatrix} F_z \\ T_{yz} \\ T_{zx} \end{pmatrix} = \begin{pmatrix} g(m_1 + m_2 + m_3 + m_{Obj}) \\ gm_2 r_2 \sin\theta_1 + gm_3\{R_2 \sin\theta_1 + r_3 \sin(\theta_1 + \theta_2)\} \\ \quad +gm_{Obj}\{R_2 \sin\theta_1 + R_3 \sin(\theta_1 + \theta_2)\} \\ gm_1 r_1 + gm_2(R_1 + r_2 \cos\theta_1) \\ \quad +gm_3\{R_1 + R_2 \cos\theta_1 + r_3 \cos(\theta_1 + \theta_2)\} \\ \quad +gm_{Obj}\{R_1 + R_2 \cos\theta_1 + R_3 \cos(\theta_1 + \theta_2)\} \end{pmatrix} \tag{B.3}$$

where

$g$: Gravitational acceleration on the ground

$m_i$: Mass of $i$th link

$m_{Obj}$: Mass of the object held by the end effector

$r_i$: Distance from $(i-1)$th joint to CG of $i$ th link

$R_i$: Distance from $(i-1)$th joint to $i$ th joint

$\theta_i$: Joint angle of $i$th joint.

See also Figure B-3 for the definition of parameters.

Eq. B.3 can also written in matrix form as follows.

$$\boldsymbol{f}_{out} = \boldsymbol{f}_{out,0} + \boldsymbol{G}(\boldsymbol{\Theta} - \boldsymbol{\Theta}_0) \tag{B.4}$$

where

$$\begin{aligned}
\boldsymbol{f}_{out,0} &= g \begin{pmatrix} m_1 + m_2 + m_3 \\ 0 \\ m_1 r_1 + m_3 R_1 + m_3 R_1 \end{pmatrix} + \boldsymbol{G}\boldsymbol{\Theta}_0 \\
\boldsymbol{G} &= g \begin{pmatrix} 0 & 0 & 0 & 0 \\ m_2 r_2 + m_3 R_2 & 0 & m_3 r_3 & 0 \\ 0 & m_2 r_2 + m_3 R_2 & 0 & m_3 r_3 \end{pmatrix} \\
\Theta &= \begin{pmatrix} \sin(\theta_1) \\ \cos(\theta_1) \\ \sin(\theta_1 + \theta_2) \\ \cos(\theta_1 + \theta_2) \end{pmatrix} \\
\Theta_0 &= \begin{pmatrix} \sin(\theta_{1,0}) \\ \cos(\theta_{1,0}) \\ \sin(\theta_{1,0} + \theta_{2,0}) \\ \cos(\theta_{1,0} + \theta_{2,0}) \end{pmatrix}
\end{aligned}$$

### Adjusted Sensor Model

To separate the effect of the out-of-plane force from the sensor output, the sensor model (Eq.B.2) is decomposed as follows

$$\begin{aligned}
\boldsymbol{z} &= \boldsymbol{z}_0 + \boldsymbol{H}\boldsymbol{f} \\
&= \boldsymbol{z}_0 + \boldsymbol{H}_{in}\boldsymbol{f}_{in} + \boldsymbol{H}_{out}\boldsymbol{f}_{out}
\end{aligned} \tag{B.5}$$

where

$$\boldsymbol{f}_{in} = \begin{pmatrix} F_x \\ F_y \\ T_{xy} \end{pmatrix}, \quad \boldsymbol{f}_{out} = \begin{pmatrix} F_z \\ T_{yz} \\ T_{zx} \end{pmatrix}$$

By substituting Eq.B.4 to Eq. B.5, the following *adjusted sensor model* is obtained.

$$\boldsymbol{z} = \boldsymbol{z}'_0 + \boldsymbol{H}_{in}\boldsymbol{f}_{in} + \boldsymbol{H}'_{out}(\boldsymbol{\Theta} - \boldsymbol{\Theta}_0) \tag{B.6}$$

where

$$\begin{aligned} \boldsymbol{z}'_0 &= \boldsymbol{z}_0 + \boldsymbol{H}_{out}\boldsymbol{f}_{out,0} \\ \boldsymbol{H}'_{out} &= \boldsymbol{H}_{out}\boldsymbol{G} \end{aligned}$$

This adjusted sensor model (Eq.B.6) is the basis of all force/torque sensor measurement and calibration.

Note that $\boldsymbol{H}'_{out}$ includes the mass of the object held by the end effector $m_{Obj}$ (See Eq.B.3). Therefore $\boldsymbol{H}'_{out}$ varies for the different mass at the end effector. This is the reason why the $H'_{out}$ has to be recalculated when a different object is attached to the end-effector (See page 62).

### B.3.2 Theory of Calibration

Calibration is necessary to obtain $\boldsymbol{z}'_0$, $\boldsymbol{H}_{in}$, and $\boldsymbol{H}'_{out}$ in Eq.B.6.

#### Zero-point Calibration

The zero-point $\boldsymbol{z}'_0$ can be obtained by averaging several measurements of the sensor outputs $\boldsymbol{z}$ when $\boldsymbol{f}_{in} = 0$ and $\boldsymbol{\Theta} = \boldsymbol{\Theta}_0$. (Substitute $\boldsymbol{f}_{in} = 0$ and $\boldsymbol{\Theta} = \boldsymbol{\Theta}_0$ to Eq.B.6 and you will obtain $\boldsymbol{z} = \boldsymbol{z}'_0$.)

### Observation Matrices

To obtain $\boldsymbol{H}_{in}$ and $\boldsymbol{H}'_{out}$, the sensor outputs $\boldsymbol{z}$ are measured for several times with different $\boldsymbol{f}_{in}$ and $\boldsymbol{\Theta}$. Let $\boldsymbol{z}^i$, $\boldsymbol{f}^i_{in}$, and $\boldsymbol{\Theta}^i$ be the value of $\boldsymbol{z}$, $\boldsymbol{f}_{in}$, and $\boldsymbol{\Theta}$ in the $i$th measurement. After $N$ measurements, the following matrices are build from the measured data.

$$\boldsymbol{Z} = \left[ \boldsymbol{z}^1 - \boldsymbol{z}_0 \,\middle|\, \boldsymbol{z}^2 - \boldsymbol{z}_0 \,\middle|\, \cdots \,\middle|\, \boldsymbol{z}^N - \boldsymbol{z}_0 \right] \tag{B.7}$$

$$\boldsymbol{F} = \left[ \begin{array}{c|c|c|c} \boldsymbol{f}^1_{in} & \boldsymbol{f}^2_{in} & \cdots & \boldsymbol{f}^N_{in} \\ \boldsymbol{\Theta}^1 - \boldsymbol{\Theta}_0 & \boldsymbol{\Theta}^2 - \boldsymbol{\Theta}_0 & \cdots & \boldsymbol{\Theta}^N - \boldsymbol{\Theta}_0 \end{array} \right] \tag{B.8}$$

$\boldsymbol{Z}$ is a 4 by $N$ matrix and $\boldsymbol{F}$ is a 7 by $N$ matrix.

The least square estimates of $\boldsymbol{H}_{in}$ and $\boldsymbol{H}'_{out}$ are obtained by multiplying the pseudo-inverse of $\boldsymbol{F}$ with $\boldsymbol{Z}$ as follows;

$$[\hat{\boldsymbol{H}}_{in} \quad \hat{\boldsymbol{H}}_{out}] = \boldsymbol{Z}\boldsymbol{F}^T \left\{ (\boldsymbol{F}\boldsymbol{F}^T)^{-1} \right\}^T. \tag{B.9}$$

### B.3.3 Theory of Measurement

Given the sensor output $\boldsymbol{z}$ and the joint angles, the in-plane forces can be estimated as follows using the pseudo-inverse of $\boldsymbol{H}_{in}$.

$$\boldsymbol{f}_{in} = \boldsymbol{H}^+_{in} \{ \boldsymbol{z} - \boldsymbol{z}'_0 - \boldsymbol{H}'_{out}(\boldsymbol{\Theta} - \boldsymbol{\Theta}_0) \} \tag{B.10}$$

where

$$\boldsymbol{H}^+_{in} = (\boldsymbol{H}^T_{in}\boldsymbol{H}_{in})^{-1}\boldsymbol{H}^T_{in}. \tag{B.11}$$

## B.4 Calibration of Force/Torque Sensor

### B.4.1 Full Calibration

Full calibration gives all three parameters in Eq.B.6; $\boldsymbol{z}'_0$, $\boldsymbol{H}_{in}$, and $\boldsymbol{H}'_{out}$.

To conduct full calibration, perform the following procedure.

i. Place the robot in the FSRL Force/Torque Sensor Calibration Equipment. (Section B.4.1)

ii. Run the script `fts_calibrate` to input the data and calculate parameters. (Section B.4.1)

iii. Update the parameters in xPC target, if necessary. (Section B.4.4)

### FSRL Force/Torque Sensor Calibration Equipment

Figure B-4 and B-5 shows FSRL Force/Torque Sensor Calibration Equipment, which was manifactured by Tatsuro "Ted" Nohara. The robot's base is fixed by the L-shape clasp of the equipment. As shown in Figure B-5, six light-weight fishing lines are attached to the manipulator joints. The lines go through the pulleys, and are attached to paper clips at the other end. Weights are hang on the clips to apply in-plane force $\boldsymbol{f}_{in}$ (See Eq.B.6).

Since the forces and torques measured are small, these weights need to be small. Small wrenches with paper clips were found to be an expedient way to produce repeatable weights for the calibration equipment. Seven weights are defined and numbered. For example, the weight #1 is a wrench which weighs 78.8 g. The weight #7 is a cluster of five clips, which weights 6.4 g. The mass of these weights are defined in `fts_weights2loads.m`.

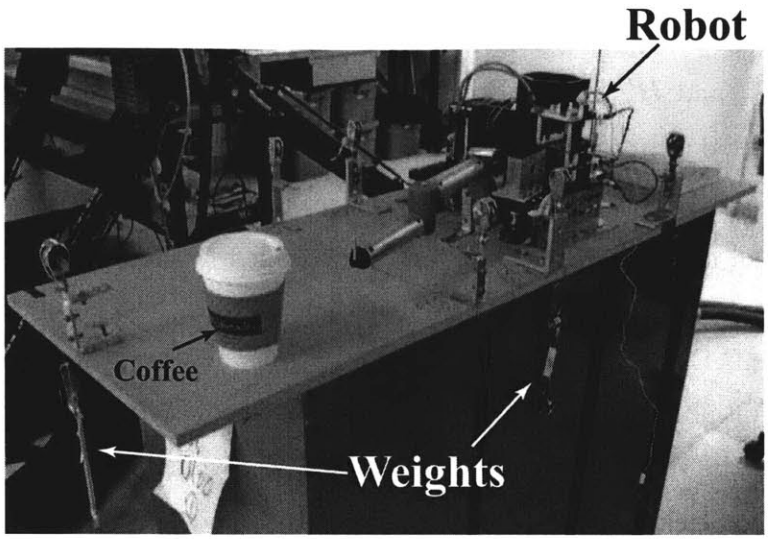


Figure B-4: FSRL Force/Torque Sensor Calibration Equipment

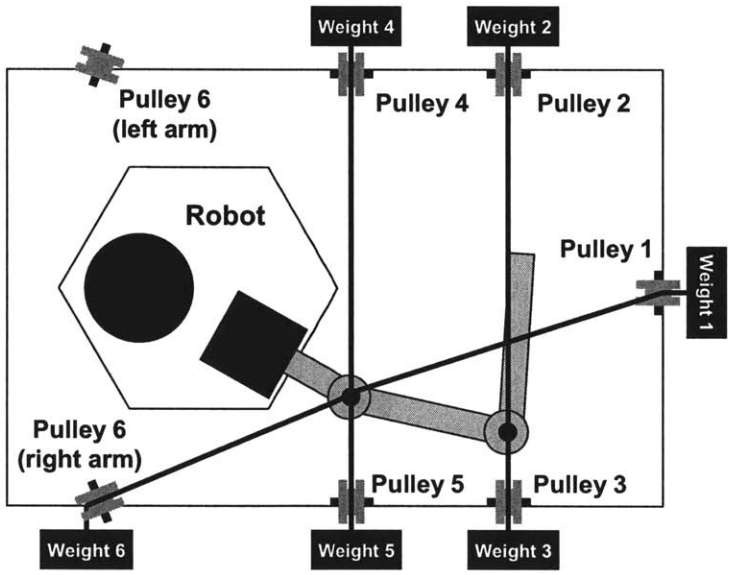


Figure B-5: Schematic of the FSRL Force/Torque Sensor Calibration Equipment showing calibration of the right manipulator

`fts_calibrate`

To obtain $\boldsymbol{H}_{in}$ and $\boldsymbol{H}'_{out}$ using Eq.B.9, $\boldsymbol{F}$ (Eq.B.8) has to be non-singular. To this end, the sensor outputs have to be measured with various weights, which result in various $\boldsymbol{f}_{in}$, and various joint angles, which result in various $\boldsymbol{\Theta}$. A Matlab script `fts_calibrate` helps this procedure, and is described in the following sections.

**To Start `fts_calibrate`** Boot the xPC target (PC104), establish communication, and launch Matlab. Change the directory to `<JAXA_root>\xPC\lib\fts_calibrate`. Then execute the script `fts_calibrate`. It will ask which robot to use, and which manipulator to be calibrated. It loads the necessary model automatically on the xPC target (PC104). It automatically move the joints to the zero-point angles (Section B.1), and measures the zero point ($\boldsymbol{z}'_0$ in Eq.B.6). Sometimes the robots have trouble moving the joints. In such a case, help the robot to move the joint with your hand. Then it waits for the user's command.

**Basic Commands of `fts_calibrate`** The two most important commands are the `i` command and `h` command. Use the `i` command repeatedly to input data, and then use `h` command to calculate $\boldsymbol{H}_{in}$ and $\boldsymbol{H}'_{out}$ and save the result.

The `i` **command** measures the sensor output and joint angles. Use this command to input new data. Hang weights on the clips of FSRL Force/Torque Sensor Calibration Equipment and tell the program which weight you hang.

Table B.2 shows the suggested profile. Angles do not have to be precise ($\sim 10°$ error is allowed). This profile is optimized for forces in the range 1[N] to -1[N]. If you use the sensor with smaller range, lighter weights (similar weights as your maximum range) would work better. This profile is for the right manipulator; for the left manipulator, use the joint angles with the opposite sign.

Table B.2: Full calibration: suggested weight and angle profile

| # | Joint 1 [deg] | Joint 2 [deg] | Weight # | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Pulley 1 | Pulley 2 | Pulley 3 | Pulley 4 | Pulley 5 | Pulley 6 |
| 1 | -25 | 125 | 1 | - | - | - | - | - |
| 2 | -25 | 125 | - | 4 | - | - | - | - |
| 3 | -25 | 125 | - | - | 4 | - | - | - |
| 4 | -25 | 125 | - | - | - | 1 | - | - |
| 5 | -25 | 125 | - | - | - | - | 1 | - |
| 6 | -25 | 125 | - | - | - | - | - | 1 |
| 7 | -45 | 90 | - | - | - | - | - | - |
| 8 | -30 | 15 | - | - | - | - | - | - |
| 9 | 0 | 0 | - | - | - | - | - | - |
| 10 | 30 | 0 | - | - | - | - | - | - |
| 11 | 45 | -90 | - | - | - | - | - | - |

Use the `c` **command** to check the current condition number. Large condition number means the data matrix F is close to singular matrix, and hence, is undesirable. (Condition number should be $\leq 35$ ) If the condition number is too large, add data with different joint angles and weights to lessen the condition number.

Then use the `h` **command** to calculate $\boldsymbol{H}_{in}$ and $\boldsymbol{H}'_{out}$ and save the result. Use the `q` command to terminate the program.

`fts_calibrate` **Command Reference**

`c` Show the current condition number

`d` Delete past data

`h` Calculate $\boldsymbol{H}_{in}$ and $\boldsymbol{H}'_{out}$ and save the result

`i` Input data

`j` Move the arm to the desired joint angles

`l` Show the list of data input so far

`m` Show the secret message from the developer

`mat` Input any Matlab command

`r` Read the log file and load data to the workspace.

`s` Stop joint motors

`w` Write the log to a file

`z` Conduct zero point calibration again .(The data is preserved.)

`help` Show help document

The script also accepts any Matlab commands except those which are listed above. Using `mat` command allows you to enter any Matlab command.

The script generates a log file automatically. If the program ends before you execute the `h` command, you can reload the input data by using the `r` command. The name of the auto-saved log file is '`fts_calibrate_log<date>.mat`'.

### B.4.2 $H'_{out}$ Calibration

The $H'_{out}$ calibration gives two of the three parameters in Eq.B.6; $z'_0$ and $H'_{out}$. The other, $H_{in}$ is calculated by full calibration (Section B.4.1).

To conduct the $H'_{out}$ calibration, follow the following procedure.

i. Attach the same object to the end effector that will be used in the experiment

ii. Turn on the gas and float the robot.

iii. Use the script `fts_auto_zero_calibrate2` to input the data and calculate parameters.

iv. Update the parameters in xPC target, if necessary. (See Section B.4.4)

**Important Note** As is explained in Section B.3.1, $\boldsymbol{H}'_{out}$ depends on the mass of the object attached to the end effector. Thus, for example, if you attach a beam to the end effector in the experiment, you have to attach the same beam in the $\boldsymbol{H}'_{out}$ calibration.

To avoid the trouble of conducting $\boldsymbol{H}'_{out}$ calibration every time you change the object on the end effector, copy all .mat files with the filenames start with "calibrateH_" and "zeroCalibrate_" to other directory. When you change the attached object, copy the corresponding .mat files to `<JAXA_root>\xPC\lib\fts_calibrate`, and then update the parameters in xPC target (See Section B.4.4).

`fts_auto_zero_calibrate2` Change directory to `<JAXA_root>\xPC\lib\fts_calibrate`. Then execute the script `fts_auto_zero_calibrate2`. It will ask which robot to use, and which manipulator to be calibrated. It loads the necessary model automatically on the xPC target (PC104). It automatically moves the joints to the zero-point angles (Section B.1), and measures the zero point ($\boldsymbol{z}'_0$ in Eq.B.6). Then it automatically moves the joints to several positions, and collects data to obtain $\boldsymbol{H}'_{out}$.

When a heavy object is attached, the robots may have trouble moving the joints. In such a case, help the robot to move the joint with your hand.

### B.4.3 Zero Point Calibration

The zero point calibration gives only $\boldsymbol{z}'_0$. It is recommended to conduct the $\boldsymbol{H}'_{out}$ calibration instead of the zero point calibration the precision of the measurement.

To conduct the zero point calibration, use the script `fts_zero_calibration2`. It will ask which robot to use, and which manipulator to be calibrated. It loads the necessary model automatically on the xPC target (PC104). It automatically moves the joints to the zero-point angle (Section B.1), and measure the zero point ($\boldsymbol{z}'_0$ in Eq.B.6).

### B.4.4 Update the Calibration Data

The calibration result is saved in .mat files with filenames starting with "calibrateH_" and "zeroCalibrate_". These data are automatically loaded and used when the model is compiled. However, if the calibration is conducted after the model is compiled, the latest calibration data are NOT used in the model. In such case, you have to execute the script `fts_update` after the model is loaded and before the model is started.

This script's effect is one-time only. It cannot re-write the compiled program. Thus you have to execute `fts_update` EVERY TIME after you load the model, until you recompile the model .

# Bibliography


[1] Mit space solar power workshop, May 2007. `http://web.mit.edu/space_solar_power/index.html`.

[2] T. Arai, N. Matsuo, N. Yamamoto, T. Yairi, and K. Machida. Simulations and experiments of laser guidance for flying effector. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Aug 2004.

[3] Arun K. Banerjee, Nelson Pedreiro, and William E. Singhose. Vibration reduction for flexible spacecraft following momentum dumping with/without slewing. *AIAA Journal of Guidance, Control, and Dynamics*, 24(3):417 – 427, 2001.

[4] J. Ben-Asher, J. Burns, and E. Cliff. Time-optimal slewing of flexible spacecraft. *AIAA Journal of Guidance, Control and Dynamics*, 15(2):360–367, March - April 1992.

[5] H. W. Bennett, C. LaVigna, H. G. Kwatny, and G. Blankenship. Nonlinear and adaptive control of flexible space structures. In *ASME, Transactions, Journal of Dynamic Systems, Measurement, and Control*, volume 115, pages 86 – 94, March 1993.

[6] Amy M. Bilton. Fusion of remote vision and on-board acceleration data for the vibration estimation of large space structures. Master's thesis, Massachusetts Institute of Technology, June 2006.

[7] H. Bitter, H. Fisher, and M. Surauer. Design of reaction jet attitude control systems for flexible spacecraft. In *Proceedings of the Ninth Symposium on Automatic Control in Space*, 1982.

[8] S. di Gennaro. Active vibration suppression in flexible spacecraft attitude tracking. *AIAA Journal of Guidance, Control, and Dynamics*, 21(3):400 – 408, 1998.

[9] John C. Doyle. Guaranteed margins for lqg regulators. *EEE Transactions on Automatic Control*, 23(4):756–757, 1978.

[10] Ryu Funase and et al. niversity of tokyo's cubesat project - its educational and technological significance -. In *Proceedings of the 56th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, Oct 2005.

[11] Andrew Harlan. Design and fabrication of a test apparatus for lightweight flexible space modules for assembly. Bachelor's Thesis, Massachusetts Institute of Technology, Decebmer 2006.

[12] Yoshiyuki Ishijima, Dimitrios Tzeranis, and Steven Dubowsky. The on-orbit maneuvering of large space flexible structures by free-flying robots. In *Proceedings of ISAIRAS 2005 Conference*, 2005.

[13] S. Jacobsen, C. Lee, C. Zhu, and S. Dubowsky. Planning of safe kinematic trajectories for free flying robots approaching an uncontrolled spinning satellite. In *Proceedings of the ASME 27th Annual Biennial Mechanisms and Robotics Conference*, September 2002.

[14] Matthew D. Lichter. *Shape, Motion, and Inertial Parameter Estimation of Space Objects using Teams of Cooperative Vision Sensors.* PhD thesis, Massachusetts Institute of Technology, September 2004.

[15] J. C. Mankins. A technical overview of the "sun tower" solar power satellite concept. *Acta Astronautica*, 50(6):369–377, March 2002.

[16] Koji Nakaya and et al. Tokyo tech cubesat: Cute-i - design and development of flight model and future plan -. In *Proceedings of 21st International Communications Satellite Systems Conference and Exhibit*, Apr 2003.

[17] E. Papadopoulos and S. Dubowsky. Coordinated manipulator/spacecraft motion control for space robotic systems. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, volume 2, pages 1696–1701, 1991.

[18] Mario Paz and William Leigh. *Structural Dynamics Theory and Computation.* Kluwer Academic Publishers, fifth edition, 2004. Ch. 17.5.

[19] A. Saenz-Otero, D. W. Miller, and M. Hilstad. Spheres: A laboratory for formation flight and docking research. In *DCSS Conference*, July 2002.

[20] Takashi Kida; Isao Yamaguchi; Yuichi Chida; Takeshi Sekiguchi. On-orbit robust control experiment of flexible spacecraft ets-vi. *AIAA Journal of Guidance, Control, and Dynamics*, 20(5):865–872, 1997.

[21] W. Singhose and H. Okada. Control of flexible satellites using analytic on-off thruster commands. In *Proceedings of the 2003 AIAA Guidance, Navigation, and Control Conference*, 2003.

[22] S. D. Smith, M. M. Penny, T. F. Greenwood, and B. B. Roberts. Exhaust plume impingement of chemically reacting gas-particle flows. In *AIAA 10th Thermophysics Conference*, May 1975.

[23] J. Turner and J. Junkins. Optimal large-angle single-axis rotational maneuvers of flexible spacecraft. *AIAA Journal of Guidance and Control*, 3(6):578 – 585, Nov - Dec 1980.

[24] D. Tzeranis and S. Dubowsky. Vibration control in the assembly of large flexible structures by teams of space robots. In *Proceedings of SYROCO 2006: 8th IFAC Int'l Symposium on Robot Control*, 2006.

[25] Trevor Williams. Space station flexible dynamics under plume impingement. Technical Report 2, Johnson Space Center, National Aeronautics and Space Administration (NASA)/American Society for Engineering Education (ASEE) Summer Faculty Fellowship Program, Dec 1993.