Design and Development of Advanced Load Sensors for the International Space Station

by

Amir R. Amir

B.S., Aeronautics and Astronautics
Massachusetts Institute of Technology, 1993

S.M., Aeronautics and Astronautics
Massachusetts Institute of Technology, 1995

Submitted to the Department of Aeronautics and Astronautics
in Partial Fulfillment of the Requirements for the Degree of

Engineer in Aeronautics and Astronautics

at the

Massachusetts Institute of Technology

September 1998

Signature of Author .....
Department of Aeronautics and Astronautics
September 1, 1998

Certified by .....
Dava J. Newman, Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by .....
Jaime Peraire
Chairman, Department Graduate Committee

Design and Development of Advanced Load Sensors for the International Space Station

by

Amir R. Amir

Submitted to the Department of Aeronautics and Astronautics
on September 1, 1998 in partial fulfillment of the requirements for the
Degree of Engineer in Aeronautics and Astronautics

ABSTRACT

In preparation for the construction of the International Space Station (ISS) a risk mitigation experiment was conducted to quantify the crew-induced disturbances to the microgravity environment on board a spacecraft during a long duration space flight. Achieving a microgravity environment for scientific experiments is one of the primary objectives of the ISS. While numerous measurements have been made to characterize the overall acceleratory environment on the Space Shuttle and on Mir, the contribution of astronaut motion to the disturbances was little understood.

During the first phase of the ISS Program, the stay of U.S. astronauts on the Russian Orbital Complex Mir, the Enhanced Dynamic Load Sensors (EDLS) Spaceflight Experiment measured from May 1996 to May 1997 the forces and moments that astronaut exerted on the space station. Using four instrumented crew restraining and mobility devices, a handhold, two foot loops, and a touchpad, 133 hours of data was recorded during nominal crew activities and scientific experiments.

The thesis gives a historical overview of the research that has been conducted to quantify the crew-spacecraft interaction. A description of the EDLS experiment set-up and timeline as well as the custom-designed experiment hardware and software is provided. Due to an on-orbit failure of the original data acquisition system, a replacement computer was used to continue the experiment. The post-flight efforts to calibrate the replacement hardware, catalog the data files, and the tests to determine the condition of the sensors are presented. A cross-platform EDLS-specific software package was developed to aid in the analysis of the spaceflight data. The requirements, underlying signal processing equations, and the implementation in MATLAB are discussed. A preliminary design of advanced sensors for the ISS is developed in the thesis. While, retaining the proven strain-gage based method of sensing forces and moments, the restraining portion of the sensors was redesigned to aid astronauts better and can be easily exchanged for a different functionality. While having a volume of only 5800 cubic centimeters, the sensor electronics unit (SEU) incorporates most of the features of the original computer eight times its size. The SEU features an advanced embedded computer system and a Java-based operating system. Feedback on the loads applied can be provided in near real-time to the crew to aid the astronauts in maintaining a quiescent environment on the station during critical microgravity experiments.

Thesis Supervisor: Dava J. Newman, Ph.D.
Title: Associate Professor of Aeronautics and Astronautics

ACKNOWLEDGMENTS

My foremost thanks belong to my advisor Professor Dava Newman. I am deeply grateful for all your support, encouragement, enthusiasm, and trust. Thank you for allowing me to work on an exciting spaceflight experiment. I am greatly appreciative of the guidance provided by my committee members Professors Jack L. Kerrebrock, Manuel Martinez-Sanchez, and Stanley I. Weiss. Your support has been a great honor for me.

Thanks is owed to NASA's Office of Life Sciences Research which funded the EDLS risk mitigation experiment for the International Space Station under contract NAS1-18690. Sherwin Beck has been the co-investigator on EDLS from NASA the general liaison to the agency. Thank you so much for all your help during the First Phase I Research Symposium in Houston. It was a real pleasure to attend the symposium with you and I hope to see you again soon.

The people of Payload Systems, Inc. and Midé Inc. in Cambridge built the EDLS spaceflight hardware and much the success of the experiment is due to their knowledge and ingenuity. To Tienie: I am very grateful to you for explaining the sensor technology and so much else. The more I know you, the more I am amazed about your knowledge—you have been practically a co-advisor to me and I thank you for that. To Kim: Thank you so much for answering my questions about EDLS in general, finding all the bits of information I asked for, and being a great liaison on this project. To Joe. I do not know how many times I called you or visited you to ask you a question but I know that you always had time for me and answered my questions patiently. Thanks! It was a great pleasure working with you. To Ed. While Joe is the software wizard, Ed and Stephan are the hardware wizard. Thanks so much for all the information you provided and building the ground sensors. I won't forget your help! Last not least, thanks to you Javier for keeping everything organized and for allowing me to roam around your facilities and take time from your employees. I am also indebted to Paul Bauer for his work on the strain gages and for being the source of great ideas.

Several students worked with me on EDLS: fellow graduate student and office mate Natasha Neogi, UROPers Jennifer Bonnell, Larry Pilkington, and Shane Suehisa. You did a terrific job and I could not have done it if was not for your enormous help. Your enthusiasm for the project, your eagerness to learn, and the willingness to do boring but necessary work made it so nice to work with you.

Simone, you have my deep gratitude not only for your valuable technical advice but also for becoming such a good friend over the course of the last year. Thanks also belong to another Italian, Guido Baroni. It was fun to have you here at MIT this past summer. I am looking forward seeing the advanced EDLS sensors work with your ELITE system on the International Space Station a few years from now.

Many other people have helped me throughout this project and I would like to express my sincere appreciation to them. They are MIT Aero/Astro Librarian Eileen Dorschner, Kim Kilman from the ISS Program Library, Peter Bruckner and Joel Gwynn of Designer's CADD, Inc., Craig A. Haller of Macraigor Systems Inc., and Chris Lines from the Motorola Semiconductor Group who generously provided me with a Motorola PowerPAQ. To Liz Zotos. Thanks for your patience.

Thanks are also due to Dr. Charles Oman, director of the MVL, and the other students in the lab, specifically Rex, Luca, Susan, Joe, and Jen, for creating such a pleasant environment to work. My family

has always been an important part of my life and what I am I owe to them. I would like to thank or all their sacrifices, support, encouragement, help, and love.

ACRONYMS

A/D	Analog-to-Digital
AFR	Anchor Foot Restraint
ARIS	Active Rack Isolation System
ASIC	Application Specific Integrated Circuit
CDR	Commander
COTS	Commercial Off-the-Shelf
CSA	Canadian Space Agency
d.o.f.	degree(s) of freedom
D/A	Digital-to-Analog
DLS	Dynamic Load Sensors
EDLS	Enhanced Dynamic Load Sensors
EMI	Electromagnetic Interference
ESA	European Space Agency
ESM	Experiment Support Module
ESRD	Experiment Support Requirements Document
EVA	Extra-vehicular Activity
FCS	Flight Crew Systems
FD	Flight Day
FE	Flight Engineer
FEM	Finite Element Model
FFT	Foot restraint 1, Foot restraint 2, Touchpad
FHT	Foot restraint 1, Handhold, Touchpad
FR	Foot restraint or Functional Requirement
HH	Handhold
ISS	International Space Station
ISSCB	International Space Station Control Board
IVA	Intra-Vehicular Activity
JSC	NASA Johnson Space Center
LaRC	NASA Langley Research Center
LDCR	Long Duration Crew Restraint
LDFR	Long-Duration Foot Restraint
LEO	Low Earth Orbit
LeRC	NASA Lewis Research Center

MASU	Mir Auxiliary Sensor Unit
MGMAIT	Microgravity Multidisciplinary Analysis Integration Team
MiSDE	Mir Structural Dynamics Experiment
MIT	Massachusetts Institute of Technology
MODE	Middeck 0-Gravity Dynamics Experiment
MODE-R	Middeck 0-Gravity Dynamics Experiment Reflight
MSIS	Man-Systems Integration Standards
MVL	Man-Vehicle Laboratory
NASA	National Aeronautics and Space Administration
NASA	National Aeronautics and Space Administration
NASDA	National Space Development Agency of Japan
NRC	National Research Council
OARE	Orbital Aerodynamic Research Experiment
PCB	Printed Circuit Board
PCMCIA	Personal Computer Memory Card International Association
PCMCIA	Personal Computer Memory Card International Association
POCC	Payload Operations Control Center
POSA	Payload Operations Support Area
PSI	Payload Systems Inc.
R&MA	Restraint and Mobility Aid
RAM	Random Access Memory
RCS	Reaction Control System
RHA	Rack Handle Assembly
RMS	Root Mean Square
ROM	Read-Only Memory
RSA	Russian Space Agency
SAMS	Space Acceleration Measurement System
SCSI	Small Computer Serial Interface
SDFR	Short-Duration Foot Restraint
SEU	Sensor Electronics Unit
SRU	Sensor Restraint Unit
SSF	Space Station Freedom
STS	Space Transportation System
TP	Touchpad
TSH	Triaxial Sensor Head
URL	Uniform Resource Locator
USB	Universal Serial Bus

VAIT	Vehicle Analysis and Integration Team
VAT	Vehicle Analysis Team
VIPT	Vehicle Integrated Product Team

TABLE OF CONTENTS

LIST OF TABLES

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Appendix A

Appendix B

Appendix C

List of Figures

Chapter 5

Chapter 6

Chapter 7

CHAPTER

1

INTRODUCTION

The engineering work described in this thesis has been carried out within the framework of the Enhanced Dynamic Load Sensors (EDLS) spaceflight experiment conducted on the Russian Space Station *Mir* from May 1996 to May 1997. EDLS was a follow-on experiment to the Dynamic Load Sensors (DLS) Space Shuttle middeck experiment flown on mission STS-62 in March of 1994.

NASA provided funding for EDLS as a risk mitigation experiment for the International Space Station (ISS). The primary objective of both DLS and EDLS was to measure the forces and moments resulting from *regular* astronaut activities. One key motivation for spaceflight is to achieve a "zero-g" or, more precisely, *microgravity* environment for research. As a result, disturbances to this environment such as vibrations and accelerations from equipment and the astronauts should be minimized to assure that material science, life science, and physics experiments as well as astronomical observations have the best possible conditions in orbit.

1.1 Scope of the Thesis

This thesis encompasses two components. While they are distinct, one leads logically to the next. The first component is the process of converting the raw EDLS flight data into a form usable for further analysis. This work required a detailed understanding of the experimental hardware and software of which some was developed more than ten years ago. In addition, mistakes as well as areas for improvement in the design were identified. The second thesis component is the preliminary design of a new set of sensors that benefit from the experiences made on the Shuttle and on Mir. This work is well beyond what is required in the contract for EDLS. It represents the first step in the development of advanced sensors to measure astronaut-induced loads that could be built into some ISS modules permanently to aid in the operation of the station. From a less ambitious perspective, it is the first step in the development of an EDLS follow-up experiment on ISS sometime in the first or second decade of the next century.

1.2 The DLS and EDLS Experiments

While numerous measurements of disturbances caused by mechanical systems as well as the space environment in low earth orbit (LEO) had been made, little was known about astronaut-induced disturbances. The first experimental data came from an experiment on *Skylab* in 1973 in which the astronauts executed a set of prescribed activities and the resulting forces were measured. These activities included "vigorous soaring" as a worst-case scenario, resulting in loads close to 500 Newtons. The *peak* values from the experiment were used as the characteristic force-level to be expected from crew motion. As a result, ISS requirements expected such a load from the astronauts.

While DLS was restricted to a few days in orbit, EDLS could record data over several months. The longer timeframe yielded a larger database and the ability to quantify the change in crew motion during adaptation to the microgravity environment. During the three-day DLS experiment data on some 600 astronaut *events* where recorded. An event is a specific crew motion lasting a few seconds, such as a landing or a push-off. The experimental hardware consisted of three sensors, a hand-hold, a foot restraint, and a touchpad, similar in appearance to the devices found in the Space Shuttle orbiter to assist the crew in moving about the spacecraft. A photograph of the DLS sensors taken prior to the Shuttle flight is shown in Figure 1.1. Data recording and storage was performed by an external unit the size of a Space Shuttle middeck locker (approximately 0.046 m^3 or 46 liters in volume) and weighing close to 27 kilograms.

The EDLS experiment was conducted on the space station *Mir* through the U.S.-Russian cooperation known as Shuttle-Mir Program, or officially, as Phase 1 of the International Space Station Project. The EDLS hardware consisted of the same devices used on DLS as well as one additional foot restraint identical to the original one (in the lower right corner of Figure 1.1) to bring the total number of sensors from three to four. The software for the *Mir* data recording computer was changed to include "event detection," to avoid a waste of storage capacity by recording continuously on the long-duration space missions.

During the *Mir* stay of U.S. astronaut Shannon Lucid (Mission NASA 2 to Mir) in 1996, some 2.5 GBytes of EDLS data were recorded. Due to a failure of the EDLS computer system, no data were recorded during NASA 3. In order to continue the experiment a similar computer system, made by the same manufacturer for another experiment, was utilized on the NASA 4 *Mir* mission. An adapter and new software were sufficient to continue the EDLS experiment with the new computer. Approximately

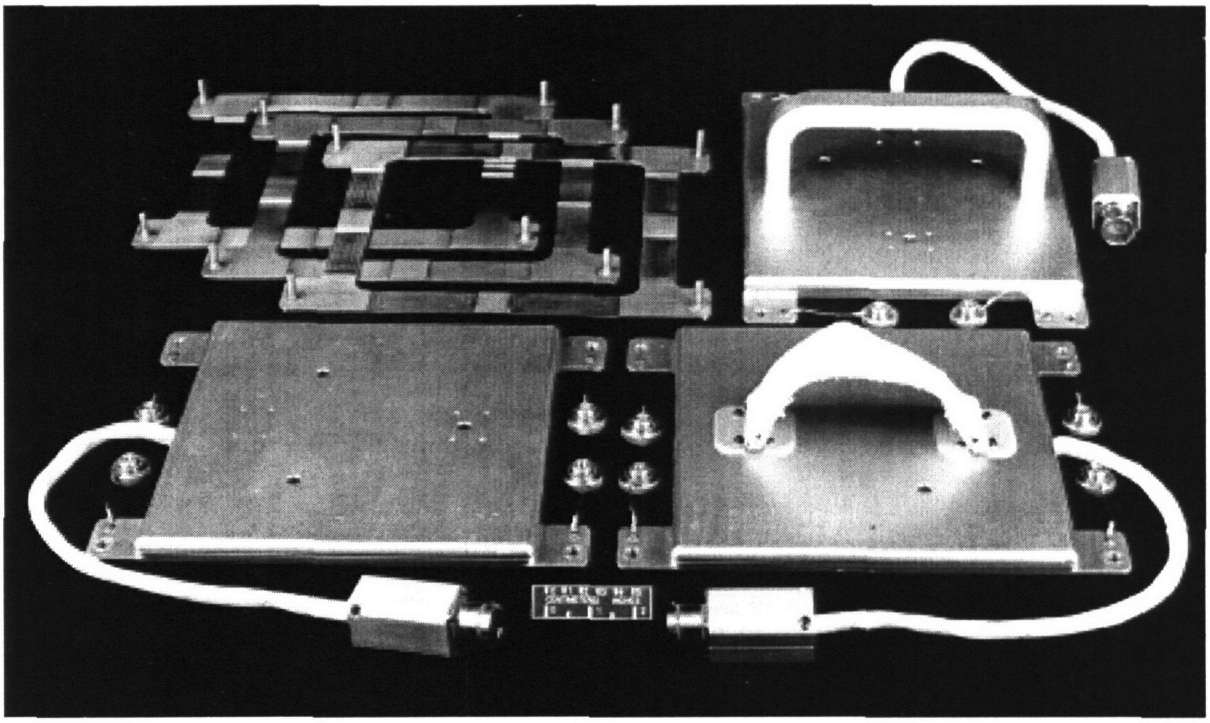


Figure 1.1: The photographs shows the DLS sensors: a touchpad, a foot restraint, and a handhold as well frames to anchor the sensors to the floor or walls of the orbiter. Each sensor measures about 24 by 24 centimeters (NASA Image 94-05316).

2.2 GBytes of data were recorded during Jerry Linenger's (Mission NASA 4) visit to *Mir* in 1997. While the items needed to rescue the EDLS experiment were not difficult to build, the on-orbit switch of computers resulted in a much more difficult and time-consuming post-flight data processing effort as is described in this thesis.

Following the collision of a *Progress* resupply vehicle with the *Spektr* module of *Mir* in June 1997 and the resulting difficulties on the orbital complex, further crew time was not available to record additional EDLS data. The experimental hardware was returned to Earth on Shuttle flight STS-89 in January 1998.

1.3 Sensors for the International Space Station

The main objective for the International Space Station is to be a "world-class laboratory" by providing scientists, engineers, and entrepreneurs (1) a research platform with a prolonged exposure to microgravity and (2) the presence of human inhabitants to execute and supervise experiments. Various research activities require an absolutely quiescent environment of 10^{-6} g or less. To provide such a

condition, Boeing offers the microgravity Active Rack Isolation System (ARIS), which is stated to attenuate structural dynamic vibrations [42]. At a cost of about $20 million and large mass and power requirements, such a system is expensive, not only in monetary terms. If the effects from crew motion are better quantified, then less-sensitive experiments can be conducted without employing isolation systems at a substantially lower cost and complexity. ISS operation will include a so-called "microgravity mode," in which machinery on board ISS will operate to minimize vibrations and accelerations. During this time, the human crew must also adjust their motions accordingly to minimize forces exerted on the craft.

Official ISS documents clearly recognize the importance of investigating disturbances due to crew motion and mitigating the effects but as of mid-1998 have no plans for dealing with the issues. An advanced version of the EDLS sensors could not only store data but also provide immediate feedback to the astronauts on the magnitude of the reaction forces they cause and thus would be a valuable tool for the crew in determining whether they need to adjust their motion or not. The stored data would be examined in detail later by researchers to ensure that their experiment was not subjected to excessive loads. In conjunction with a system that has accelerometers mounted at various locations on the station to monitor the microgravity environment, a good picture of the vibrational environment and the sources for the accelerations would be obtained.

The computer system originally used with the DLS/EDLS sensors was designed in the late 1980s as a generic data acquisition and storage system to support space shuttle middeck experiments. The replacement computer used represented a somewhat more advanced version of the original system. It maintained the same size as the predecessor model but incorporated early 1990s technology to boast more features. Due to the fast pace in the electronics field, the latest commercial off-the-shelve hardware outperforms earlier custom-made systems by a large margin for a fraction of the mass, volume, and power. By making use of the latest advancements in electronics and by including only components of relevance to the load sensors, a next generation of sensors would permit a miniaturization so that the entire data acquisition and storage system could be incorporated into the sensors themselves. The lack of umbilical cables from the sensor to an external computers system, would result in a simpler, safer, and mobile system.

Many lessons were learned from observing the astronauts using the EDLS sensors on videotape and direct feedback from the crew. This information helped improve the mechanical portion of the sensors;

that is, the top restraint portion of the sensors (i.e., the metal rail on the handhold and the canvas loops on the foot restraints).

1.4 Thesis Outline

Chapter 2 "Space Stations and Microgravity" provides background information on the *Mir* Orbital Complex where the EDLS data was gathered and the Shuttle-Mir Program through which the EDLS experiment was conducted. The chapter also includes a fairly detailed description of International Space Station for which the advanced sensors were designed and a general classification of the microgravity environment. As the name implies, Chapter 3 "Previous Research on Astronaut-Spacecraft Interaction" discusses the prior research efforts on microgravity disturbances due to crew motion and the key results obtained.

The motivation, objectives, history as well as technology of the DLS/EDLS spaceflight experiments are the subjects of Chapter 4. Chapter 5 "Processing and Analysis of EDLS Experimental Data" explains the postflight calibration process of the sensors, the processing of EDLS data and the validation of the data through ground tests. Chapter 6 "Design and Development of Advanced Load Sensors" discusses the requirements for measuring crew-disturbances on ISS, the mistakes that were made in the development of the original equipment, and ends with the presentation of the preliminary design of the advanced sensors. The seventh and final thesis chapter describes the next steps in the analysis of the EDLS data, and outlines the path to complete the development of the advanced sensors.

CHAPTER

2

SPACE STATIONS AND MICROGRAVITY

This chapter begins with a description of the Russian Mir Orbital Complex followed by background information on the International Space Station which includes the Shuttle-Mir Program, as part of which the EDLS experiment was conducted. The information provided herein is more extensive than is necessary for later chapters from a purely technical perspective, but is useful to gain an understanding for the overall environment in which the EDLS experiment was carried out and in which the advanced sensors would be used in.

Section 2.3 explains the microgravity environment and the disturbances acting on it aboard spacecraft in LEO. The section is concluded with a brief discussion of the measurements that have been taken over the last decades to characterize the microgravity environment on the Space Shuttle and Soviet space stations. The last part of this chapter examines the microgravity requirements for the International Space Station.

2.1 Mir Orbital Complex

While the United States placed its emphasis on low-cost access to space through the development of the Space Shuttle, the Soviet Union favored long-duration space flight through the construction of space stations. The Russian *Mir* Orbital Complex is the last in a long line of Soviet/Russian space stations. The history of Soviet stations can be described as one of gradual improvement, continuous upgrade of equipment, and quick recovery from failures. The first generation of Soviet space stations allowed only a temporary presence in space since they could not be resupplied or refueled. They were launched unmanned and later occupied by crews. Although there were two types, *Almaz* ("Diamond") military stations and *Salyut* civilian stations, both were called *Salyut* ("Salute") to confuse the West. The world's first space station, *Salyut* 1, was launched into orbit unmanned on April 19, 1971. The crew of *Soyuz* 11 lived aboard the station for three weeks but died upon return to Earth as air leaked from their cabin. The three following first-generation stations did not reach orbit or broke up before

they were manned. The Soviet program recovered from this string of failures and from 1974–77 had three successful stations in orbit [48], [23], [36].

Salyut 6 and 7 were the second-generation of Soviet space stations. The former was operated from 1977 to 1982 and the latter from 1982 until 1986. They allowed a long-duration stay through the existence of two docking ports of which one was used by the automated resupply vehicle *Progress*. A total of 26 crews visited the two stations where the longest stay was 237 days [23].

On February 19, 1986, the Soviet Union launched the *Mir* (meaning "Peace" and "World" in Russian) space station. Development of this third-generation station had begun in 1976 involving 200 scientific, project and construction organizations of 20 ministries and departments [45]. Mir became the world's first permanent space habitat and the first station designed to be expanded over time. At the time of launch the station weighed 20,400 kg and consisted of only one module with a total pressurized volume of 90 cubic meters [46]. The *Mir* core module (sometime called the Base Block) featured six ports with the fore and aft port serving as docking stations and the four radial ports in a node serving as connectors for additional modules. It provided basic services such as living quarters, life support, power, and some scientific research capabilities. Crews were brought to Mir with Soyuz-TM spacecraft and supplies with the Progress-M cargo transport—an improved version of the venerable Progress ferry, which was used until 1989 [46]. *Mir*'s first crew consisted of cosmonauts Leonid Denisovich Kizim and Vladimir Aleksandrovich Solovyev, who occupied the station from March 13 to July 16, 1986 [46].

Six modules have been added over the years to give *Mir* its present configuration, which is summarized in Table 2.1.

Table 2.1: The Mir Modules [40], [47], [48]

Module	Launch Date	Primary Purpose	Pressurized Volume
Core Module or Base Block	February 1986	Habitation, power, life support	90 m^3
Kvant ("Quantum")	March 1987	Astrophysics	40 m^3
Kvant II ("Quantum II")	November 1989	Logistics (airlock, solar arrays, etc.)	61 m^3
Kristall ("Crystal")	May 1990	Materials processing	61 m^3
Spektr ("Spectrum")	May 1995	Geophysical sciences	62 m^3
Docking Module	November 1995	Shuttle-Mir docking capability	N/A
Priroda ("Nature")	November 1996	Remote sensing, U.S. facilities	66 m^3

The complex with docked Progress-M and Soyuz-TM spacecraft is more than 32 meters long and about 27 meters wide. The total mass of *Mir* exceeds 91,000 kg. The station, with the modules labeled, is shown in Figure 2.1.

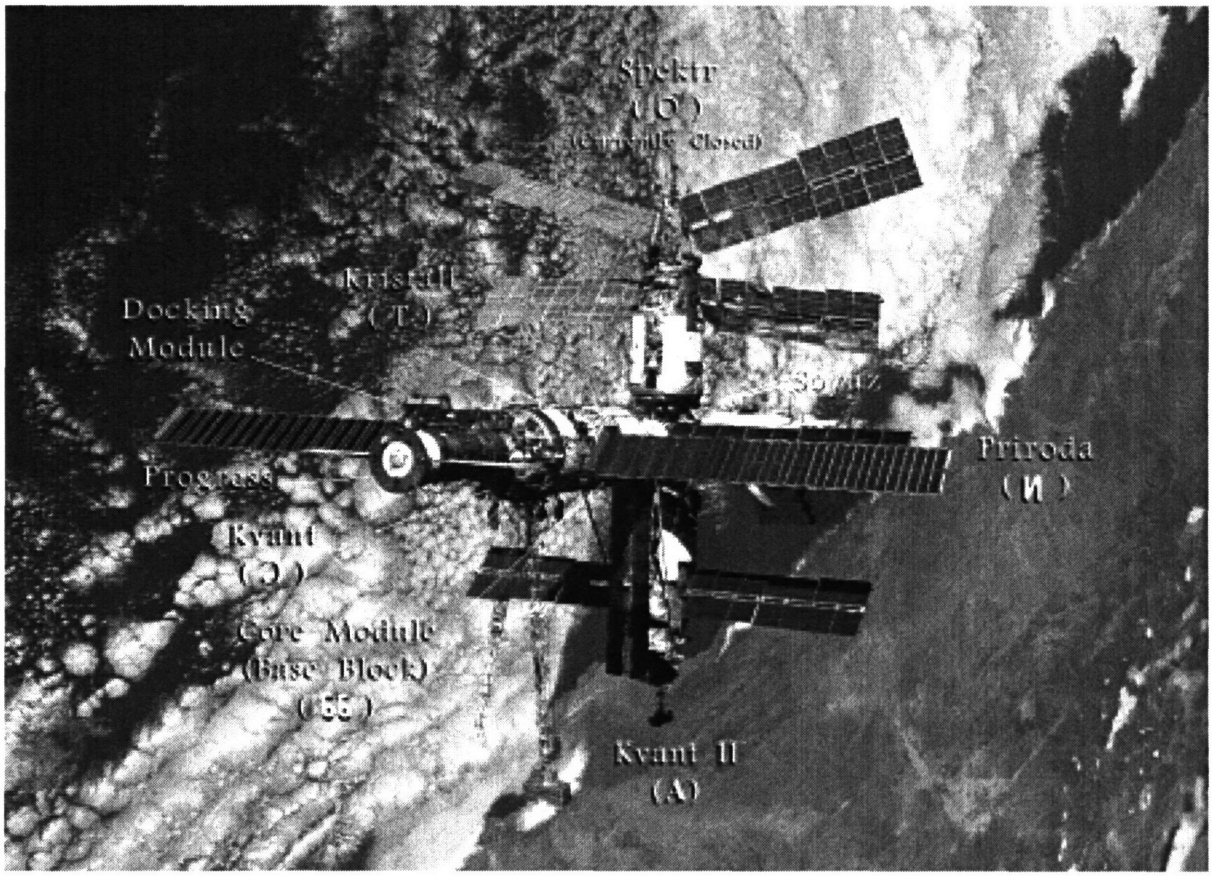


Figure 2.1: This photograph of Mir was taken by the crew of Shuttle Mission STS-86. All modules and the Progress resupply vehicle are labeled. The cyrillic letters in parentheses are the Russian abbreviations for the Mir components (NASA Image STS86-370-25) [40].

Most of the EDLS flight data was recorded in the Priroda module of Mir. As mentioned earlier, Priroda was the last module to be added to the orbital complex. After its launch from the Baikonur Cosmodrome in Kazakhstan on April 23, 1996, it docked with the station on schedule on April 26. As the name hints, its primary purpose was to add Earth remote sensing capability to *Mir*. It also carried 1,000 kg of hardware and supplies for several U.S.-Russian science experiments including EDLS. The specifications for Priroda are as follows [58]:

Length	Mass	Max. Diameter	Pressurized Volume
13 m	19,700 kg	4.35 m	66 m^3

Despite its large pressurized volume, the module is cramped with equipment, giving the astronauts little space to move in as is evident in Figure 2.2, which shows the interior of the Priroda module.

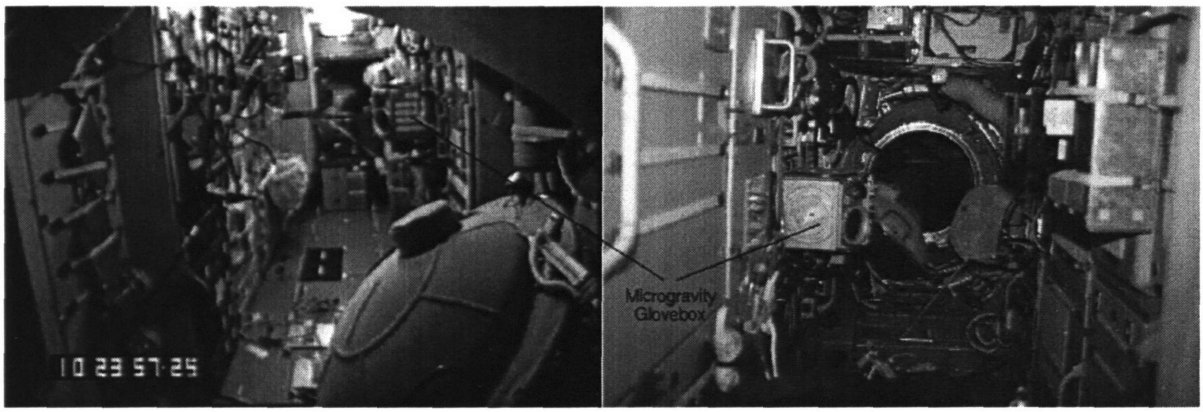


Figure 2.2: The two photographs show the interior of the Mir Priroda module where most of the EDLS data was taken. The left image is a view towards the end cone with the hatch in the foreground. The right image is a view facing towards the transfer node. For reference, the microgravity glovebox is shown in both pictures (NASA Mir-21/22 H-8mm Onboard Video ID#53, NASA Image xxx).

2.2 The International Space Station

The International Space Station (ISS) is the largest international civilian scientific and technological project that has ever been undertaken. Under the leadership of the United States, sixteen nations—USA, Russia, Japan, Canada, Germany, France, Italy, United Kingdom, Belgium, Denmark, Netherlands, Norway, Spain, Sweden, Switzerland, and Brazil—are building a multi-billion dollar orbiting research facility [51].

2.2.1 Historical Perspective

America's first space station was *Skylab*. Originally, NASA planned to send ten missions to the moon, but because of more pressing national concerns, Apollo 18, 19, and 20 were cancelled. One of the three Saturn rockets intended for the flight was kept in flight-ready status, while the others were made museum exhibits.[1] The third stage of the flightworthy booster was modified into Skylab by converting the gigantic fuel tank into cabins. Skylab was launched into orbit on May 14, 1973 on top of a smaller Saturn rocket left over from the Apollo test program. The 75-ton station provided very generous quarters for the crew—the main work area was a cylinder 14.7 m long and 6.6 m in diameter. The cylinder

1. The two unused Saturn boosters are located at the Johnson Space Center in Houston, Texas and at the Marshall Space Flight Center in Huntsville, Alabama, respectively.

was partitioned with a grid made with triangular gaps, so that the astronauts could anchor themselves by pushing toggles on the soles of their boots into the holes. Skylab hosted three crews each with three astronauts for a total 171 days. The first manned mission lasted 28 days, the second 59 days, and the third 84 days. The station had been abandoned for five years when it reentered the atmosphere completely uncontrolled on July 11, 1979 [36].

The Space Transportation System or STS is NASA's name for the collective Space Shuttle Program. On April 12, 1981, the agency launched the first Space Shuttle, *Columbia*, into orbit. Since that time a fleet of space shuttle orbiters has served a dual role. It is both a transportation system bringing satellites and other payloads into orbit and a platform for short-duration microgravity experiments. Since 1992, NASA is transitioning from Shuttle-based research to International Space Station research. It is doing it in three phases: Phase I, the so-called Shuttle-Mir Program, is a cooperation with Russia involving crew exchanges that has been concluded successfully in June 1998. Phase II is the establishment of the initial capability for long-term research on ISS with the on-orbit assembly of U.S. and Russian modules while Phase III will expand the station's infrastructure through the addition of hardware from all international partners. Details on the three phases are provided in Section 2.2.3 and Section 2.2.4.

The origin of ISS dates back to January 25, 1984 when U.S. President Reagan announced in his State of the Union speech support for a permanent human tended space station intended for completion in the early 1990s. Only days before the address, he had sent private memos to European, Japanese, and Canadian leaders inviting them to join the project. In 1988, the project was named *Space Station Freedom* (SSF). From 1984 until 1992, NASA spent $8 billion on a complex series of design reviews, but no elements of the station were completed [36].

Due to large budget constraints, poor management, and the inability to meet schedules, SSF was subject to heavy criticism. On March 9, 1993, President Clinton declared a 90-day ultimatum to redesign the station to reduce costs and asked for three station design options, costing $9 billion, $7 billion, and $5 billion respectively. An advisory panel under the chairmanship of MIT President Dr. Charles M. Vest, presented A, B, and C options to the administration. While options A and B were scaled-down versions of SSF, Option C represented a complete departure from the Freedom concept. President Clinton decided for an amalgam of Option A and B. The recommendation of the Vest Committee to bring Russia into the project, led to the signing of a major agreement between the two countries in

September of 1993. In response, the name of the project was changed to International Space Station Alpha (ISSA) and later to International Space Station (ISS) [36].

Although the ISS design stems largely from the work done for *Space Station Freedom*, there are several major design differences between SSF and ISS. Most of these were due to the fact that the orbital inclination for the station and hence the solar beta angle was changed. The solar beta angle is defined as the angle between the orbital plane and the ecliptic. Its maximum value is the sum of the inclination and the tilt of the Earth's equator with the ecliptic (23.5°). The fundamental effect of the beta angle is the amount of sunlight received by the spacecraft during a given orbit.

The inclination of the SSF orbit, 28.5°, was determined by the latitude of the Kennedy Space Center launch site. Once, Russia became a partner in the station, it was necessary to change the orbit since an inclination of 28.5° made it difficult to reach the space station when launching from the Baikonur Cosmodrome, located at a latitude of 45.9°. Therefore, driven by the Russian launch site, an inclination of 51.6° was chosen.[2] The beta angle varies with time of the year and orbital precession effects. For the SSF design, the solar beta angle would have varied from –52° to +52°, while for the ISS it will vary from –75° to +75° with a frequency of approximately four cycles per year and an average angle of approximately 30°. One example of a resulting effect is that during early station assembly, not enough power will be available during high beta angles. A positive effect for the microgravity environment is the fact that at large solar beta angles, the frontal area of the photovoltaic arrays is dramatically reduced, which decreases the drag acceleration levels [37], [61].

2.2.2 Purpose, Objectives, and Organization of ISS

The main purpose of the station has been stated as:

> Provide an Earth orbiting facility that houses experiment payloads, distributes resource utilities, and supports permanent human habitation for conducting research and science experiments in a microgravity environment [42].

Based on this purpose, specific objectives were declared, which can be formulated as follows [37]:

- Develop a world-class orbiting laboratory for conducting high-value scientific research.
- Provide access to microgravity resources as early as possible in the assembly sequence.

2. The ISS ground track covers about 75% of the world's landmass and 95% of the world's population [42].

- Develop the ability to live and work in space for extended periods.
- Develop an effective international cooperation.
- Provide a testbed for developing 21st century technology.

Early on, it was decided that designing and building the station was to be the responsibility of a single prime contractor. In August 1993, following the change from SSF to ISSA, The Boeing Company was selected as the prime contractor and in January 1995, NASA and Boeing's Defense and Space Group, Missiles and Space Division signed a $5.63 billion contract for the design and development of the station. Under the agreement, Boeing is responsible for the integration and verification of the International Space Station system and the design, analysis, manufacture, verification and delivery of the U.S. on-orbit segments of the station [53]. Within NASA, the Johnson Space Center (JSC) was selected as the lead center for ISS [36].

Besides, NASA and Boeing, the ISS Program consists of [37]:

- Russian Space Agency (RSA) with its contractors Rocket Space Corporation Energia (RSC-E) and Krunichev Space Center (KhSC)
- Canadian Space Agency (CSA) with its contractor Spar Aerospace
- National Space Development Agency of Japan (NASDA) with its contractor Mitsubishi Heavy Industries
- European Space Agency (ESA) with its contractor Daimler-Benz Aerospace

2.2.3 Phase I: The Shuttle-Mir Program

In 1992, the United States signed an agreement with Russia to allow long-duration space flights of U.S. astronauts on Mir, the flight of a Russian cosmonaut on a Space Shuttle mission, Shuttle/Mir dockings, and joint science programs. In September 1993, U.S. Vice President Gore and Russian Prime Minister Chernomyrdin announced a further expansion of the human space flight cooperation of the two countries. The implementation consisted of three phases of which the Shuttle-Mir Program was the first [36].

The four primary objectives of Phase I are [30]:

1. To reduce the technical risk associated with the construction and operation of the International Space Station.

2. To reduce scientific risk and enhance long duration experiment performance and science utilization for the ISS.
3. To combine international space operations and joint space technology demonstrations.
4. To provide early opportunities for extended scientific, technologic and engineering research and testing.

These objectives were achieved primarily by an exchange of crews; joint ground control operations; the docking of the Shuttle with Mir to assist in crew exchange; resupply and payload activities; the exchange of personnel to develop an understanding of each space agency's design, development, test, and training, and operational philosophies. Phase I began with Shuttle flight STS-60 in February of 1994, which carried Russian cosmonaut Sergei Krikalev along with five U.S. astronauts into orbit. Altogether, the Shuttle-Mir Program had seven astronauts on Mir, involved 10 Shuttle flights to Mir, 9 dockings, and the delivery of 9 payloads such as a docking module (to simplify docking between Mir and the orbiter), replacement solar panels, food, clothing, experiment supplies, and spare station equipment. Table 2.2 lists the NASA missions to Mir.

Table 2.2: U.S. Astronauts on the Russian Space Station Mir [40], [47]

Mission	Astronaut	Launch	Landing	Length of Stay
NASA 1 / Mir 18	Norman Thagard	March 14, 1995 on Soyuz TM-21	July 7, 1995 on STS-71	114 days
NASA 2 / Mir 21	Shannon Lucid	March 22, 1996 on STS-76	September 26, 1996 on STS-79	188 days
NASA 3 / Mir 22	John Blaha	September 16, 1996 on STS-79	January 22, 1997 on STS-81	132 days
NASA 4 / Mir 22 / Mir 23	Jerry Linenger	January 12, 1997 on STS-81	May 24th, 1997 on STS-84	132 days
NASA5 / Mir 23 / Mir 24	Michael Foale	May 15, 1997 on STS-84	October 6, 1997 on STS-86	142 days
NASA 6 / Mir 24	David Wolf	September 25, 1997 on STS-86	January 25, 1998 on STS-89	120 days
NASA 7 / Mir 24 / Mir 25	Andrew Thomas	January 22, 1998 on STS-89	June 7, 1998 on STS-91	127 days

With the return of Andrew Thomas from *Mir*, the most visible part of the Shuttle-Mir Program has been concluded. From August until November 1998 one more Soyuz flight and two more flights of Progress-M unmanned ferries to Mir are planned before the official completion of Phase I. [40]

The science program of Phase 1 included scientific research from many disciplines. The microgravity research emphasized the characterization of the microgravity (acceleratory and vibrational) environment, but also included a select amount of research involving protein crystal growth, cell culturing, and some combustion science and fluid physics experiments.

As part of the technology and system validation, several technologies and subsystems for ISS were tested. The main area of interests were systems engineering and crew support.

Systems engineering tests performed to validate designs fell into the following categories: (1) Life Support, (2) EVA Technology, (3) Vibration Isolation / Microgravity Environment, (4) Assembly and Maintenance, (5) Loads and Dynamics, (6) Data Processing System, (7) Contamination and Radiation, (8) Micrometeoroid/Orbital Debris [40].

Standards for crew support were established in the following areas: (1) Medical Support, (2) Habitation, (3) Environmental and Advanced Life Support [40].

Some tests and experiments led to specific design enhancements and modifications to ISS. For example, evaluation of docking and rendezvous maneuvers of the Shuttle with Mir, led to a placement of additional lighting of the Space Station to enable use of the orbiter's star trackers during proximity operations [52].

The Enhanced Dynamic Load Sensors Experiment was conducted within Phase 1 as a so-called *risk mitigation experiment*. Its results, to be discussed in Chapter 5, are leading to changes in the specifications for space station windows and possibly other components.

2.2.4 ISS Phases II and III

As was mentioned earlier, the second and third phases of the International Space Station program constitute construction and operation of the station. According to Revision D of the Assembly Sequence from May 31, 1998, construction of ISS will require approximately 50 flights (excluding resupply and logistics flights) over a period of more than five years [19], [49]. U.S. flights dedicated to assembly are referred to as "A" flights (e.g., 12A) and Russian assembly flights as "R" flights (e.g., 2R). Flights involving the transportation of science experiments to the station are called utilization flights (e.g., Flight UF3). Labels "J" and "E" indicate flights involving Japanese and European hardware respectively [37].

The first element of ISS to be lifted into orbit is the Russian-built Control Module or Functional Cargo Block with the Russian acronym FGB. It is currently scheduled to be launched aboard a Russian Proton Rocket in November 1998 as Flight 1A/R. The FGB, named in May 1998 Zarya ("Sunrise"), is a self-supporting module with propulsive control, fuel storage, rendezvous, and docking capability [50], [56]. The second component (Flight 2A) in orbit will be the Unity node built by Boeing. The node will be mated with the FGB and serve as a connector for future modules [55]. The third component of ISS will be the Service Module, which is the primary Russian station contribution. The module will provide life support for all early elements and initial living quarters. After the attachment of a Soyuz capsule as an assured crew return vehicle for emergencies in July 1999, the first crew will begin to live on ISS [50]. The U.S. Laboratory Module will be brought to the ISS in October of 1999 (Flight 5A). This element will provide equipment for research and technology development and house all the necessary systems to support a laboratory environment and control the U.S. segment of the ISS. Phase II of the ISS Program is expected to be concluded in January 2000. Phase III involves completing the assembly and operation of the station therafter. The station is expected to be completed January 2004 [19].

The ISS is comprised of the following seven segments:

1. United States On-orbit Segment (USOS)

   The U.S. modules making up the USOS are the U.S. Laboratory and the U.S. Habitation Module.

2. International Ground System

3. Attached Pressurized Module (APM)

   The APM is also called the Columbus Orbital Facility (COF) and is analogous to the U.S. Laboratory. It is being supplied by the European Space Agency.

4. Japanese Experiment Module (JEM)

5. Russian Segment (RS)

   Russia is furnishing the Service Module (SM) and two Research Modules.

6. Mini-Pressurized Logistics Module (MPLM)

   The MPLM is built by the Italian Space Agency (Agenzia Spaziale Italiana, ASI) and its contractor Alenia for NASA under a subcontract.

7. Mobile Servicing System (MSS)

   The Canadian-built MSS provides external robotic operations support for the on-orbit space station.

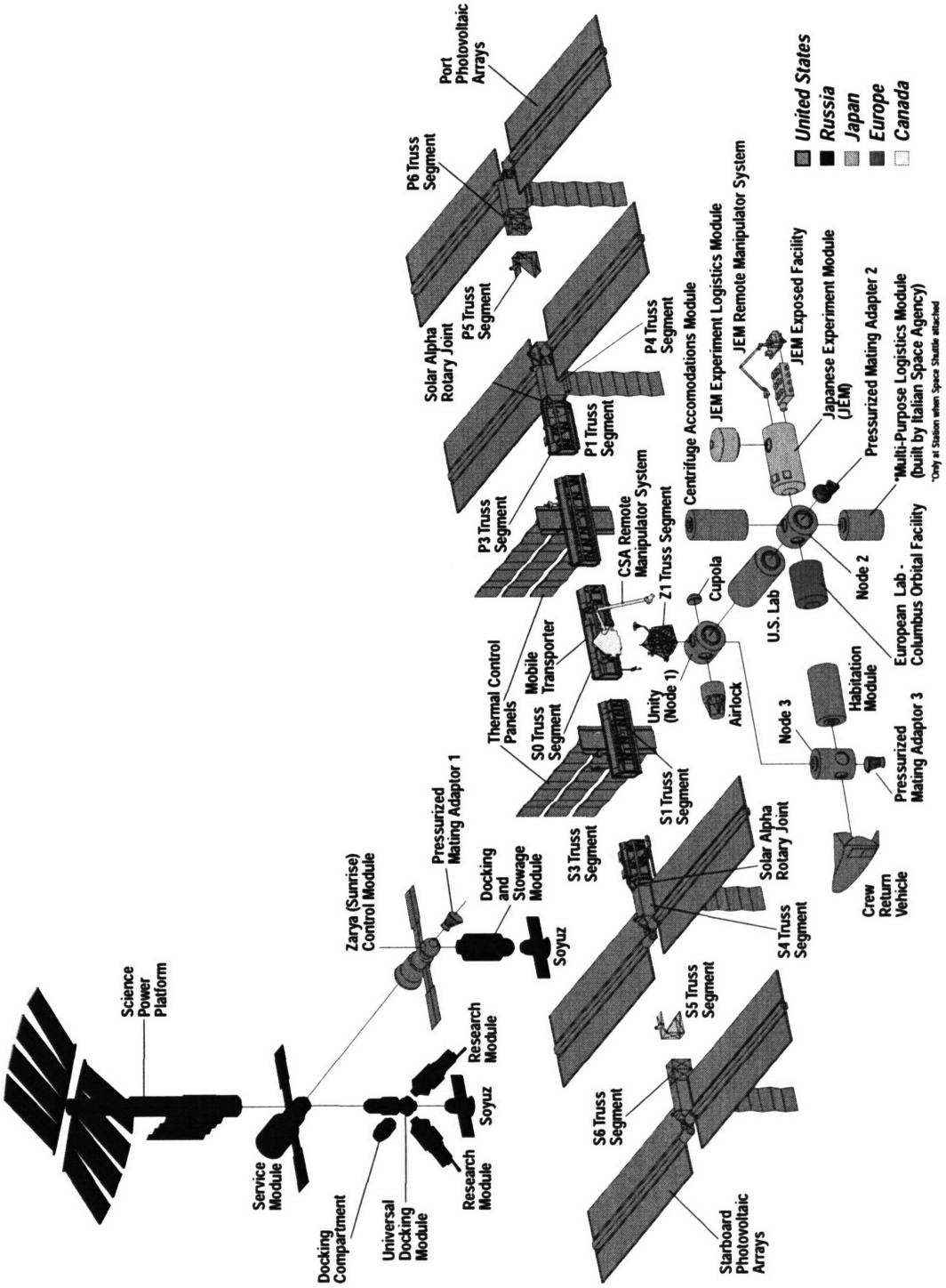


Figure 2.3: The figure shows the components making up the International Space Station and the country or agency responsible for it [59].

Figure 2.3 shows how the components of the space station will be assembled. As can be deduced from the above description, ISS consists not only of on-orbit hardware but includes also ground facilities. The on-orbit Space Station and portions of the Space Station Ground Segment will be linked via the Tracking and Data Relay Satellite System (TDRSS) for audio, video, data, and command communications. For the purpose of this thesis, the term "space station" or "ISS" will imply the on-orbit component only. The space station will operate as an integrated vehicle with an integrated crew, a single commander, and English as the language of operations [37].

It is planned to have the following characteristics once completed [59]:

- Volume: 1,200 cubic meters at a pressure of 101 kPa (14.7 psi)
- Total mass: 456,000 kg
- Dimensions (including solar panels): 108.6 m by 79.9 m
- Maximum power output: 110 kW
- Orbital inclination: 51.6°
- *Average* orbital altitude: 407 km (220 nmi)
- Crew: up to seven (three until assembly is complete)

Figure 2.4 shows the International Space Station once completed.

The pressurized living and working space aboard the completed ISS will be roughly equivalent to the passenger cabin volume of two Boeing 747 jetliners or about three times more than on Mir. Figure 2.5 shows the interior of the U.S. Laboratory mock-up at the Johnson Space Center. As is evident from the photographs, the astronauts will have much more volume to move about than on Mir.

2.3 The Microgravity Environment

With the exception of the Apollo moon missions, manned spacecraft operate in low-Earth orbit, which implies altitudes from about 100 to about 600 km. In the case of the International Space Station, the altitude will vary from 300 to 400 km (see Figure 2.6 on page 47). The discussion of the microgravity environment in this section deals exclusively with the situation on manned spacecraft in LEO.

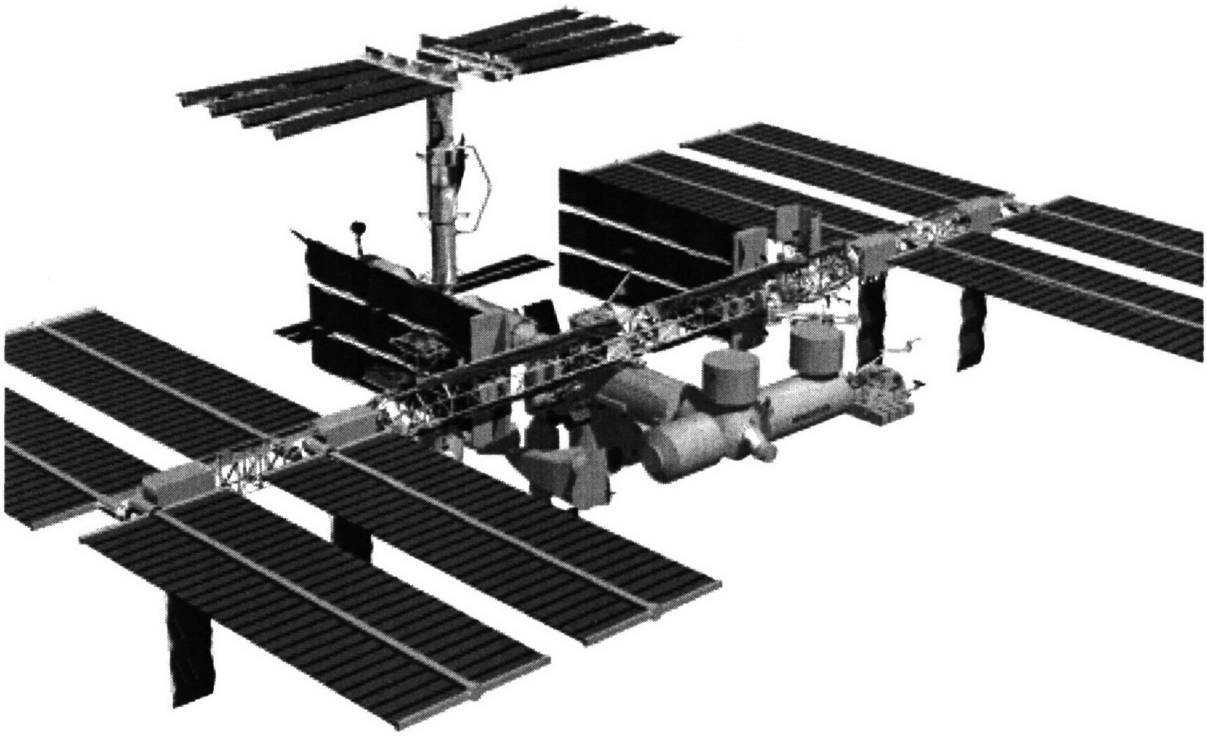


Figure 2.4: This computer-generated image shows the International Space Station when assembly is complete [57].

Figure 2.5: The photographs show the mock-up of the U.S. Laboratory at the Johnson Space Center [69].

The absence of a gravity effect is called *weightlessness, free fall,* or *zero-g*. In Earth orbit, the effect arises from Earth's gravitational pull being compensated by a centrifugal force due to a forward (or tangential) velocity large enough such that the distance from the center of the Earth remains constant.

In practice, zero-g cannot be achieved, since an orbiting spacecraft is subject to various small forces produced by the space environment. As a result, for most practical applications in LEO, the gravitational effect can be reduced to 10^{-6} g (=1 μg); a level of 10^{-7} g can be achieved over a very small region near the center of mass of the spacecraft. For this reason the term *microgravity* (μg) rather than zero-g is best to describe the condition in orbit properly.

2.3.1 Classification of On-Orbit Disturbances

The spacecraft environment on typical manned spacecraft during orbital flight is disturbed by a number of different accelerations and vibrations. The accelerations experienced on orbit are classified fundamentally as: (1) Quasi-steady or residual accelerations and (2) Non-steady accelerations.

2.3.2 Quasi-steady Accelerations

Quasi-steady or residual accelerations are due to external forces acting on the spacecraft and determined by the external configuration of the vehicle and the parameters of the orbit. By their nature these accelerations act for long periods of time. They are usually defined as accelerations with frequencies below 0.01 Hz, which uncouples the quasi-steady state environment from the vibratory environment since that limit value for the frequency is generally an order of magnitude less than the first structural mode. Calculations of the quasi-steady state microgravity environment are based on rigid-body dynamics.

Assuming a circular orbit and a local rotating frame attached to the spacecraft, the equation defining the quasi-steady state accelerations can be written as:

$$\mathbf{a} = \mathbf{a}_{\text{environment}} - \mu\left(\frac{\mathbf{r}_p}{r_p^3} - \frac{\mathbf{r}_{cm}}{r_{cm}^3}\right) - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_{p-cm}) - \dot{\boldsymbol{\omega}} \times \mathbf{r}_{p-cm} \tag{2.1}$$

where $\mathbf{a}_{\text{environment}}$ is the acceleration due to the environment, $\mu = GM = 3.986 \times 10^{14}\ \text{m}^3/\text{s}^2$ is Earth's gravitational constant, $\mathbf{r}_p$ is the vector from the center of the Earth to a point p on the station, $\mathbf{r}_{cm}$ is the vector from the center of the Earth's to the station center of mass, and $\mathbf{r}_{p-cm}$ is the vector from the station center of mass to the point of interest. The quantity $\boldsymbol{\omega}$ is the station's rotational rate

vector whose three components are ω_x, $(\omega_y - n)$, and ω_z. The variables ω_x, ω_y, and ω_z are the station's Euler rates and n is the angular orbit rate, which assumed to be positive.

The first term of Eqn. (2.1), is the acceleration vector caused by the external environment, the second term is the gravity gradient acceleration acting on a point that is a distance r_{p-cm} from the center of mass, the third term is the centripetal acceleration, and the fourth term is the tangential acceleration. Each term is discussed below.

Environmental Forces

The first term of the equation encompasses all the accelerations induced by the external environment, such as aerodynamic drag, solar pressure, quasi-steady impact of ambient micrometeoroids, tidal forces due to the Earth-moon orientation. The largest contributor to the environmental acceleration is drag. The acceleration due to drag is represented by:

$$a_{\text{drag}} = \frac{1}{2m} C_D A \rho \left(\frac{R_A + R_P}{2} \right)^2 \omega^2 \tag{2.2}$$

where m is the mass of the station, C_D the drag coefficient, A the cross-sectional area, ρ the air density, $((R_A + R_P)/2)^2$ is the square of the semi-major axis of the orbit expresssed in terms of the apogee and perigee radii, and ω the angular velocity in orbit [63].

For the *Mir* space station, the acceleration due to aerodynamic drag has been estimated to be less than 2×10^{-5} g (prior to the docking of the Spektr and Priroda module), while the more aerodynamic Shuttle orbiter, experiences an acceleration of approximately 1 to 5×10^{-6} g [12].

Gravity Gradient Acceleration

At the center of mass of the spacecraft in orbit, the centrifugal acceleration cancels out Earth's gravitational acceleration. All other points experience a slightly smaller or larger gravitational acceleration but are constrained in their path since the spacecraft is a rigid body. As a result, a spatial separation from the center of mass, induces a gravity gradient acceleration. This acceleration is the main contributor to the acceleration magnitude sensed by a payload aboard the ISS.

Reference [61] gives the following equations for the gravity gradient accelerations in the x, y, and z direction:

$$a_{GGx} = -\mu\left(\frac{\Delta x}{r_{cm}^3}\right), \qquad a_{GGy} = -\mu\left(\frac{\Delta y}{r_{cm}^3}\right), \qquad a_{GGz} = \mu\left(\frac{2\Delta z}{r_{cm}^3}\right) \tag{2.3}$$

As the equations show, the gravity gradient induced acceleration along the z or nadir direction is twice as large as the x or flight path direction and the y direction. For example, for a station orbital altitude of 407 km (220 nmi), a 1 μg acceleration will be experienced at a distance of 4.11 m in the z direction and 8.22 m in the x and y directions from the center of mass [61].

Centrifugal Acceleration

There are two rotation-induced accelerations disturbing the microgravity environment. They are non-physical forces; they arise from kinematics and are not due to physical interactions. The first of them is the centrifugal acceleration. For a perfectly nadir-pointing vehicle, the nominal centrifugal acceleration is caused by the once-per-orbit rotation required for an Earth pointing attitude. In this case, the nominal centrifugal acceleration vector lies in the orbit plane normal to the angular momentum vector. Most spacecraft are pointing towards Earth for communication, observation, and other reasons. To eliminate thie centrifugal contribution to the acceleration environment, the vehicle would have to be inertially-oriented [61], [63].

Tangential Acceleration

The tangential acceleration is caused by the angular acceleration experienced by a rigid body. The angular acceleration arises from a change in direction of angular rates from nominal angular rotation.

2.3.3 Non-Steady Accelerations

Non-steady accelerations are generated within the spacecraft. These disturbances can either be *oscillatory* or *transient* (i.e., spikes) due to singular events. Oscillatory accelerations (i.e., vibrations) are those that are periodic in nature with a characteristic frequency ranging from a tenth of a Hertz to several hundred Hertz. Sources for these disturbances are reciprocating pumps, fans, valves, motors, gyros, antenna dither motion, and acoustic noise from sources such as fans, duct inlets and outlets, pumps, and blowers. For example, the Space Shuttle orbiter has a refrigerator/freezer with a pump that causes significant vibrations at 22 Hz [3]. While, a single source of disturbances may not be significant, if multiple sources are superimposed, the effect can be substantial.

The transient accelerations have typically durations of less than a second and are nonperiodic. The energy in the disturbance is typically spread across the frequency range from the sub-Hertz to the hundreds of Hertz range. The Space Shuttle orbiter for example has several structural modes in the 1 to 10-Hz frequency regime which are often excited by transient accelerations. Origins for these disturbances are thruster firings, satellite launches, docking impacts, robotic arm motion, and of course various crew activities.

2.3.4 Measurements of the Microgravity Environment

As interest in microgravity for research grew, so did interest to quantify the acceleratory environment. On the Soviet *Salyut* station, instruments originally intended for other measurements, were used to determine the spacecraft's *steady-state* acceleration environment. Examples of these instruments were triaxial magnetometer, solar and stellar photometers, and angular motion transducers [12].

First a geophone was used to measure *non-steady* accelerations, then, beginning in 1980, a triaxial accelerometer package (IMU). In a "Resonance" experiment, all non-critical equipment was shut down to create the baseline quiescent environment, which was then compared with the acceleratory environment during standard operation. The IMU results are summarized in Table 2.3.

Table 2.3: Typical Maximum Acceleration Levels on *Salyut-7* Space Station [12]

	Maximum Acceleration [g]		
Activity	**x Direction**	**y Direction**	**z Direction**
Unmanned Spacecraft	10^{-5}	10^{-5}	10^{-5}
Standard Crew Activities	2×10^{-4} to 2×10^{-5}	10^{-3} to 10^{-5}	10^{-3} to 10^{-5}
Crew Exercise (tread mill)	10^{-4}	8×10^{-3}	8×10^{-3}

In 1992, as part of the Russian-French Mission Antares on *Mir*, two sets of microaccelorometers developed by the French were installed on the Russian orbital complex. The devices measured accelerations over a range of +/– 100 mg with a bandwidth of 0.1–400 Hz and a resolution of 5×10^{-5} g per axis. The Resonance experiment from Salyut was repeated with the French microaccelerometers on Mir and the data from the quiescent condition compared to the time when a crew member was exercising on the treadmill. A general increase in acceleration levels across the measured frequency range was observed with an additional peak at 4 Hz, which was attributed to the cycling frequency on the treadmill [12].

During the first years of Space Shuttle flights, science experiments which were most sensitive to acceleration levels, incorporated their own accelerometers to determine the microgravity conditions. In

1986, the Space Acceleration Measurements System (SAMS) was conceived by the NASA Lewis Research Center (LeRC) as a general purpose system to measure low-level accelerations at experiment locations in the Shuttle orbiter with the ability to be flown many times [6].

The SAMS debut was on flight STS-40, the Spacelab Life Sciences (SLS-1) Mission, which lasted from June 5th to June 14th, 1991. Altogether seven accelerometer systems flew on that mission to characterize the microgravity environment.[3] It was found that during the crew sleep period, the acceleration magnitude in the Spacelab module ranged from 10^{-6} to 10^{-4} g. Magnitudes increased to 10^{-4} g during nominal crew activity. Firing of the RCS[4] vernier thrusters resulted in acceleration shifts of several hundred μg and firing of the primary thrusters increased the acceleration environment to as much as 10^{-2} g [32].

The SAMS units were developed for both a pressurized habitable environment and the space vacuum environment. Three single-axis sensors with a sensitivity of 1 μg are combined into an integral unit, a triaxial sensor head (TSH) that also includes the electronics for prefiltering and amplifying the sensor signal. Each TSH is connected to the data acquisition and storage unit by a sensor head cable that can be up to 6 meters (20 feet) long. In the orbiter, two SAMS units are mounted in the middeck, Spacelab, or SPACEHAB module and a third in the Orbiter cargo bay. The data is stored on 2 GByte hard drives but raw data from the SAMS unit mounted in the Space Shuttle Cargo Bay may also be downlinked to the Payload Operations Control Center (POCC) for near-real-time data display and analysis for investigators on the ground [54].

While the SAMS instruments can acquire acceleration data over a frequency range from 0 to 100 Hz, SAMS accuracy is poor at the very low frequencies. The Orbital Aerodynamic Research Experiment (OARE) has been used successfully to complement SAMS in the 0–0.01 Hz frequency range with higher accuracy measurements. After each flight the data is processed and made available to interested scientists and engineers. SAMS has flown 20 times on the Space Shuttle and in August 1994, a modified SAMS unit with two triaxial sensor heads was installed on *Mir* to support science experiment

3. One of the systems was specific to a crystal growth experiment and SAMS, even though general in purpose, was also used in conjunction with a specific experiment on its maiden flight.
4. The Space Shuttle orbiter Reaction Control system (RCS) consists of 44 individual thrusters. There are 38 primary thrusters each with a rated thrust of 3.9 kN (870 lb) and 6 vernier thrusters each with a rated thrust of 107 N (24 lb) [3].

from the U.S. and Russia on the orbital complex. SAMS recorded acceleration data on Mir a from September 1994 to May 1998 [5], [73].

The SAMS program is sponsored by NASA's Microgravity Science and Application Division (MSAD) and operated at NASA's Lewis Research Center (LeRC). SAMS data is analyzed and disseminated by the LeRC Principal Investigator Microgravity Services (PIMS) projects whose purpose is to aid principal investigatos of microgravity science experiments to evaluate the effects of varying acceleration levels on their experiment [5].

While, SAMS has been the most often flown accelerometer system, there are many other systems. Some are for a specific purpose such as atmospheric drag determination, while others, like SAMS, are general purpose systems. An overview of the available systems can be found in Reference [3].

The importance of acceleration measurement systems has been expressed through a challenge by the Space Studies Board of the National Research Council (NRC) in its document "Toward a Microgravity Research Strategy":

> "The g-level must be measured accurately, locally, frequently, and synchronously with every experiment." [44]

For the International Space Station, a follow-on project to SAMS, called SAMS-II has been initiated. According to the developers, "[t]he SAMS-II project sets out to improve upon those operational areas that SAMS and OARE were found through experience to be lacking while, at the same time, leverage the positive feedback from the performances of SAMS, OARE, and other successful shuttle-based instruments." [44]

The EDLS experimental hardware and the SAMS hardware shared some technology, such as the WORM drive, and the on-going effort is to correlate the data between the two experiments. The design efforts for SAMS-II are parallel to the preliminary design work for advanced EDLS sensors and thus, it is helpful to follow the flowdown of requirements and design approach for SAMS-II.

While each microgravity science discipline (biotechnology, combustion, fluid mechanics, low temperature physics, and material science, etc.) has its own peculiar acceleration measurement and data analysis requirements, a parent-set of systems requirements for SAMS-II, known as the Experiment Support Requirements Document (ESRD) were established and are summarized in Table 2.4.

Table 2.4: Summary of SAMS-II Experiment Support Requirements [44]

1.	Acquire Acceleration Data	2.	Allocate Control to Users	3.	Provide Data to Users
1.1	Measure accelerations with an accuracy and resolution better than the acceleration environment envelope of the ISS program	2.1	Give the principal investigator control of the data acquisition parameters	3.1	Supply information in a selectable format
1.2	Acquire the acceleration data with correlated time information	2.2	Give the on-orbit crew control of the data acquisition parameters	3.2	Supply information within a selectable amount of time
1.3	Measure accelerations with a selectable frequency range				
1.4	Measure accelerations in, on and /or near the experiment sample/chamber apparatus				

SAMS-II consists of three basic hardware elements: (1) a series of on-orbit measurements systems called Remote Triaxial Sensors or RTS (2) a centralized on-orbit data acquisition and control system called Control Unit or CU, and (3) a ground operating system called Ground Operations Equipment or GOE. The SAMS-II RTS will be linked with the CU via the ISS Ethernet Network and the CU linked with the GOE via the ISS Downlink stream. Distribution of SAMS-II data to investigators around the world would be accomplished via the Internet [44].

Due to technology improvements since SAMS, primary hardware elements for the SAMS-II on-orbit data acquisition and control system are commercially available. Some custom modification and packing of elements is necessary to ensure launch vibration load and on-orbit environment survivability. Many functions that were conducted on the Shuttle and Mir post-flight will be expected by the research community to be delivered by SAMS-II in near-real-time aboard ISS and are thus incorporated into the CU hardware and software [44].

Since the sensing unit needs to be as close as possible to the location of the experiment location and as small as possible, the RTS is divided into two hardware elements. The RTS Sensor Enclosure (RTS-SE) contains the accelerometers only, while the RTS Electronics Enclosure (RTS-EE) provides low-level data processing and an interface capability to the Control Unit. The specifications for the SAMS-II system are shown in Table 2.5.

Table 2.5: Specifications of SAMS-II Units [44]

Specification	Control Units	RTS-EE (each)	RTS-SE (each)
Volume	0.065 m^3	0.00761 m^3	0.00288 m^3
Mass	46 kg	5.0 kg	1.5 kg
Power	350 W	32 W	powered by RTS-EE
Bandwidth	800 kbits/s	160 kbits/s	80 kbits/s
Sample Rates	N/A	N/A	62.5, 125, 250, 500, 1000 Hz
Amplitude	N/A	N/A	0.1 μg–0.1 g 24-bit A/D conversion

Final design and design of the operational aspect of SAMS-II awaits completion of key elements of ISS and the research payloads and facilities, which the system will support. It is expected that the elements will become available early in the next decade [44].

2.4 ISS Microgravity Requirements

Achieving a microgravity environment for scientific experiments is one of the primary objectives of the ISS. Early planning for ISS payloads indicates that over 35% of the science experiments intended for the station will require a microgravity environment. The reduction of the gravitational force has tremendous effects on physical and chemical processes. For example, a microgravity environment eliminates or drastically reduces buoyancy driven convection, sedimentation, and hydrostatic pressure in experiments. The elimination of buoyancy driven convection results in diffusion controlled conditions which are the fundamental transport mechanisms in a number of crystal and material growth processes. Without sedimentation, heterogeneous mixtures or suspension can be maintained and objects can be free-floated. By significantly reducing hydrostatic pressure, liquids can be constrained by surface tension alone. Experimental sensitivities to gravity induced phenomena vary greatly by type of experiment and size of the sample, and are still poorly understood [61].

Microgravity research primarily began in the late 1960's when Dr. Wernher von Braun, Director of NASA's Marshall Space Flight Center, approached MIT's Metallurgy Department[5] with the idea for processing ball bearings in space in order to achieve a perfectly spherical shape. The reaction from Professors August Witt and Harry Gatos was sour "because according to basic solidification principles, the resulting solid from a drop of steel cannot be spherical." However, von Braun did not give up easily

5. Currently the MIT Department of Materials Science and Engineering.

and asked what kind of research would benefit from a reduced gravity environment. At the time the MIT scientists were working on optimizing the properties of semiconductors and expressed that solidifying semiconductors in microgravity might be a worthwhile undertaking. NASA agreed and began to support materials processing research on Skylab [62].

Soviet space stations and the Shuttle have become a platform for extensive microgravity research but the International Space Station is the world's first manned space vehicle developed with microgravity system requirements. These requirements and all others are listed in document SSP 41000 "System Specifications for the International Space Station." [60]

Space station operations will be conducted in modes. There are seven on-orbit modes: (1) Standard, (2) Reboost, (3) Microgravity, (4) Survival, (5) Proximity Operations, (6) Assure Safe Crew Return, and (7) External Operations; and five ground modes: (1) Ground Processing, (2) Space Transport, (3) Personnel Preparation, (4) Operations Planning, and (5) Reconfiguration Preparation.

Figure 2.6 shows a timeline of ISS operation with the periods in microgravity mode identified. The capabilities required for this mode can be summarized simply as "the support of microgravity research by user payloads in a habitable environment." Interestingly, the ISS microgravity mode does *not* include the effects of crew activity, but does include the effects of crew equipment, such as the operation of exercise devices and latched or hinged enclosures. According to the specifications in SSP 41000, "crew effects will be mitigated to the extent possible."

The specifications for the microgravity environment contained in SSP 41000 are binding for all contractors and agencies involved in the space station. The requirements are repeated and elaborated below:

The Space Station shall provide the following microgravity acceleration performance for at least 50 percent of the internal payload locations (excluding nadir window payload location) for 180 days per year in continuous time intervals of at least 30 days [60]:

1. At the centers of the internal payload locations, a quasi-steady (<0.01 Hz) acceleration:
   a. magnitude less than or equal to 1 μg
   b. component perpendicular to the orbital average acceleration vector less than or equal to 0.2 μg
2. At the structural mounting interfaces to the internal payload locations:

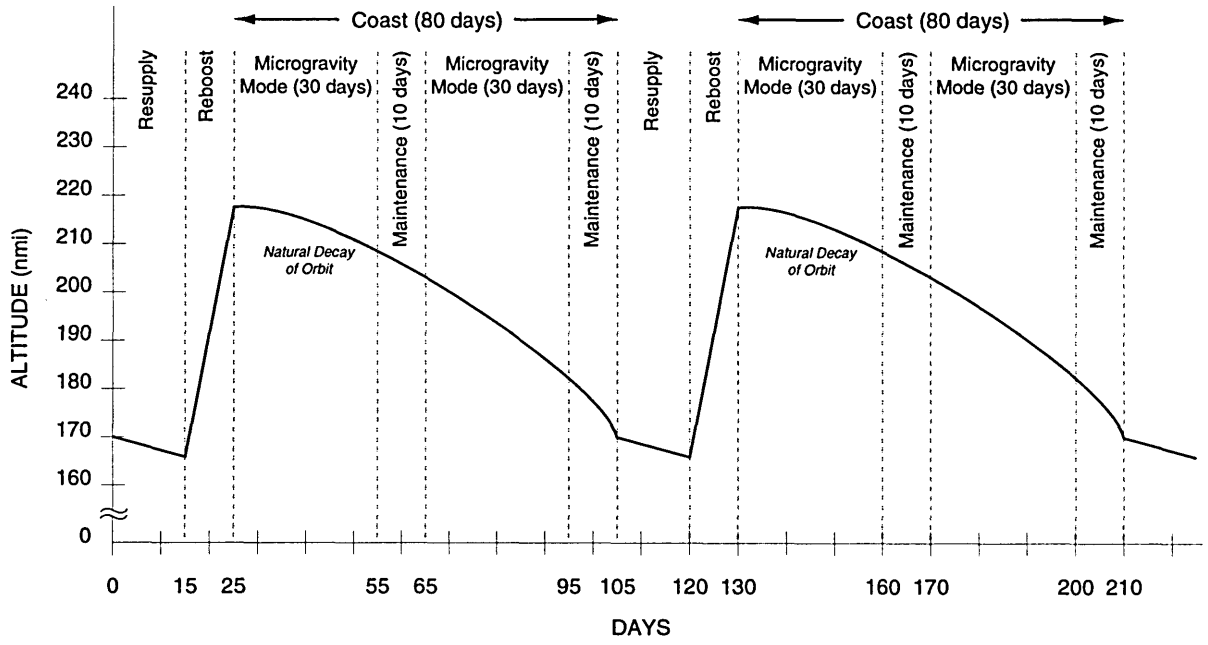


Figure 2.6: The chart shows a timeline of nominal ISS operations. Events such as station resupply, reboost, Orbiter/progress mating, and certain maintenance are scheduled outside the quiescent period [60].

a. a vibratory acceleration limit as defined in Figure 2.7.

b. a transient acceleration limit for individual transient disturbance sources less than or equal to 1000 μg per axis.

c. an integrated transient acceleration limit for individual transient disturbance sources less than or equal to 10 μg seconds per axis over any 10-second interval.

The Space Station shall monitor and record the microgravity environment at selected locations as well as provide microgravity measurement data to payloads at selected locations.

The management and overall implementation of the microgravity requirements defined in SSP 41000 are stated in the Microgravity Control Plan (MGCP), which also considers approaches to mitigate crew induced accelerations.

The prime contractor Boeing has established a Microgravity Multidisciplinary Analysis Integration Team (MGMAIT) to manage the implementation of the microgravity requirements, and the control of the on-orbit acceleration environment. The MGMAIT will have ultimate responsibility for determining the technical acceptability of the ISS design and operation with regards to microgravity performance

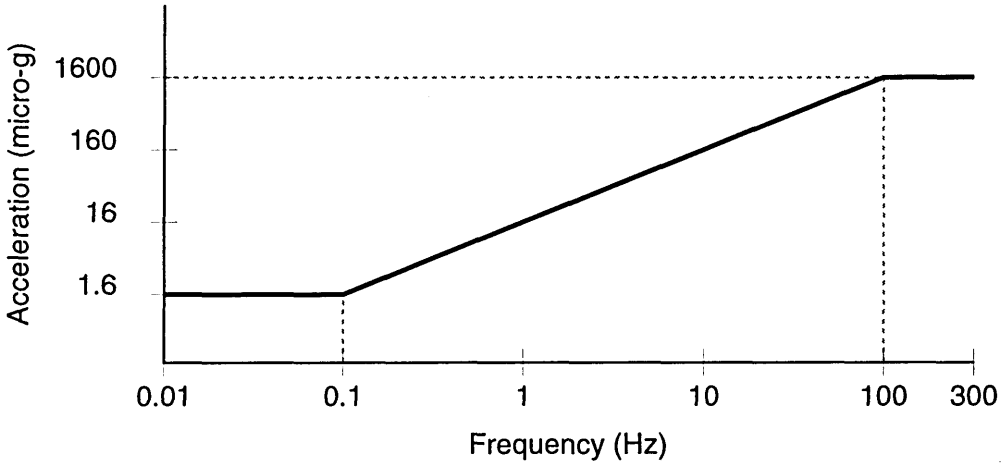


Figure 2.7: ISS microgravity acceleration limits. RMS acceleration magnitude in 1/3 octave bands average over 100 seconds [60].

and obtaining approval for microgravity related investigations. Because of the multidisciplinary nature of the microgravity control, MGMAIT is composed of NASA/contractor personnel from three teams: (1) Guidance, Navigation & Control, Structures, and Crew System Subsystem Architecture and Analysis Team; (2) Environments Team; and (3) Safety Reliability, Maintainability and Quality Assurance Team.

The MGMAIT fits in the ISS organizational structure as follows: the MGMAIT is part of the Vehicle Analysis and Integration Team (VAIT). Within VAIT, the MGMAIT support the Vehicle Analysis Team (VAT). The VAIT reports directly to the Vehicle Integrated Product Team (VIPT), which in turn reports to the International Space Station Control Board (ISSCB).

The MGMAIT has fourteen responsibilities of which two are of particular interest for this thesis. They are as follows:

- The development of crew generated forcing functions which occur in microgravity mode.
- The mitigation of the affects of crew activity by directing the development of candidate control approaches including mechanical designs, operational changes, specialized crew training, and rack isolation and by technically accessing their adequacy.

For the advanced sensors to be built into the space station, close interaction and various proposals with the above described teams would be necessary. How this could be accomplished in the future is dis-

cussed in Chapter 7. The next chapter deals with the past; namely the research on astronaut-spacecraft interaction prior to DLS/EDLS.

CHAPTER

3

PREVIOUS RESEARCH ON ASTRONAUT-SPACECRAFT INTERACTION

This chapter reviews previous research on the effect of crew motion on the spacecraft. The initial concern was that the astronauts would be a significant disturbance source for the vehicle's attitude control system but the fear turned out to be largely exaggerated. With the advent of extensive microgravity research on the Space Shuttle and on Soviet space stations the focus shifted towards maintaining a quiescent environment for experiments. Section 3.1 discusses the early investigations into the attitude disturbances due to the motion of astronauts modeled as "point-masses." These crew disturbance models were deterministic because they were completely specified as functions of time. The opportunity existed to extend the simple point-mass models to more sophisticated dynamic human body models which were developed at the time but rather the research efforts shifted toward a stochastic representations of crew motion as is discussed in Section 3.2. The third and final section of the chapter presents the only space-flight experiment on crew motion prior to DLS—the Skylab T-013 experiment in 1973.

3.1 Early Modeling Efforts

The investigation of astronaut-spacecraft interaction was initially focused on how to include the presence of humans into the design of the vehicle attitude control system. The first person to explicitly raise the issue of astronauts as a source of disturbances to the spacecraft was Robert E. Roberson in 1962. In his technical note "Comments on the Incorporation of Man into the Attitude Dynamics of Spacecraft," [29] he wrote that the astronauts' motion inside the vehicle will result in "a disturbing torque, perhaps the major one." Roberson considered the case of a single astronaut in an otherwise quiescent vehicle and derived an expression for the attitude dynamics.

Let the mass of the vehicle be m_v, $\mathbf{r}(t)$ the position vector of the vehicle frame in an inertial reference frame, and $\mathbf{g}_v$ the position vector of the vehicle's center of mass in the vehicle frame. In addition, let $\mathbf{F}_v$ be the external force and $\mathbf{T}_v$ be the external torque on the vehicle with respect to the local frame.

The vector $\mathbf{H}_v$ is the vehicle's total angular momentum with respect to the inertial frame. The disturbing forces and torques exerted by the astronaut are denoted by $\mathbf{F}_d$ and $\mathbf{T}_d$. Then, the equation of motion for the vehicle is given by

$$m_v \frac{d^2}{dt^2}(\mathbf{r} + \mathbf{g}_v) = \mathbf{F}_v + \mathbf{F}_d \tag{3.1}$$

$$\frac{d\mathbf{H}_v}{dt} = \mathbf{T}_v + \mathbf{T}_d - m_v \mathbf{g}_v \times \ddot{\mathbf{r}} \tag{3.2}$$

Analogously, the equations for the astronauts are

$$m_a \frac{d^2}{dt^2}(\mathbf{r} + \mathbf{g}_a) = \mathbf{F}_a - \mathbf{F}_d \tag{3.3}$$

$$\frac{d\mathbf{H}_a}{dt} = \mathbf{T}_a - \mathbf{T}_d - m_a \mathbf{g}_a \times \ddot{\mathbf{r}}, \tag{3.4}$$

where the subscript "a" for "astronaut" replaced "v" where appropriate. Adding Eqn. (3.1) to (3.3) and Eqn. (3.2) to (3.4) and then eliminating $\ddot{\mathbf{r}}$, which is the acceleration of the vehicle with respect to an inertial frame, to generalize the equation for any point in the orbit, yields the following expression:

$$\frac{d}{dt}(\mathbf{H}_v + \mathbf{H}_a) = \mathbf{T}_v + \mathbf{T}_a - \mathbf{g}_{cm} \times (\mathbf{F}_v + \mathbf{F}_a) + (m_v + m_a)\mathbf{g}_{cm} \times \ddot{\mathbf{g}}_{cm} \tag{3.5}$$

in which $\mathbf{g}_{cm} \equiv (m_v \mathbf{g}_v + m_a \mathbf{g}_a) / (m_v + m_a)$ is used for simplification.

The term $\ddot{\mathbf{g}}_{cm}$ can be expressed in terms of the astronaut's velocity, $\mathbf{v}$, and acceleration, $\mathbf{a}$, by the following equation:

$$\ddot{\mathbf{g}}_{cm} = \mathbf{a} + 2\omega \times \mathbf{v} + \dot{\omega} \times \mathbf{g}_{cm} + \omega \times (\omega \times \mathbf{g}_{cm}) \tag{3.6}$$

where ω is the angular velocity of the vehicle frame with respect to inertial space.

Eqn. (3.5) and (3.6) show how the astronaut's motion enters into the spacecraft attitude dynamics, namely through the astronaut's instantaneous position (through $\mathbf{g}_{cm}$), $\mathbf{v}$, $\mathbf{a}$, and the angular accelera-

tion $\mathbf{H}_a$. While a determination of position, velocity, and acceleration, is a difficult task, the inclusion of the angular momentum requires modeling the astronaut as a non-rigid body.

Roberson, also considered another route, namely measuring the astronaut-induced disturbance forces, $\mathbf{F}_d$, and torques, $\mathbf{T}_d$, directly for typical motion and inserting them into the following equation:

$$\frac{d\mathbf{H}_v}{dt} = \mathbf{T}_v + \mathbf{T}_d - \mathbf{g}_v \times (\mathbf{F}_v + \mathbf{F}_d) + m_v \mathbf{g}_v \times \ddot{\mathbf{g}}_v, \tag{3.7}$$

which was derived by inserting Eqn. (3.1) into (3.2). However, the disturbing forces and torques depend not only on the astronaut him/herself but also on the simultaneous motion of the vehicle. To illustrate this, consider an astronaut pulling on a handrail to achieve a certain relative velocity. The force required to achieve the velocity depend whether the vehicle is toward or away from the astronaut. Hence, the quantities $\mathbf{F}_d$, $\mathbf{T}_d$, are strictly speaking also dependent on the spacecraft motion, which is not completely known until Eqn. (3.7) is solved.

If the dependency on the vehicle motion can be assumed to be weak, Roberson suggested building a library of $\mathbf{F}_d$ and $\mathbf{T}_d$ functions for various tasks based purely on laboratory measurements. While warning that such a procedure would be inherently dangerous, he felt that it seems to be an attractive route to incorporate astronaut motion into the attitude dynamics of spacecraft.

In the years to follow, crew disturbance models continued to be deterministic and treating an astronauts as point mass. The objective of their research efforts was to find the consequence of the astronaut's translation from one point to another within the spacecraft.

In 1965, W. T. Thomson and Y. C. Fung published a paper on their investigation on the effect of periodic motion of the crew on a large *spinning* space station [33]. If crew members "walk" back and forth along the radius of a circular planar satellite or walk circumferentially, the authors found that the astronauts can "rock" the station and eventually make it unstable if they have a periodic motion that is in the neighborhood of an *integral multiple of the half-period* of the station's angular velocity. The exact periods of motion causing instability depends on a number of parameters, such as the mass of the astronaut(s), the type and amplitude of the astronaut motion as well as the stations's geometry and inertia.

Along the same line, Corrado R. Poli published a paper in 1967 in which he developed a mathematical model for a point mass astronaut moving in or on a right circular cylindrical satellite [28]. He assumed that the body axes of the spacecraft align with the geometric center, and the motion of the astronaut is restricted to a plane made up the longitudinal axis of the cylinder and an axis parallel to the front or end of the cylinder. If the astronaut "walks" along the front face or the top/bottom of the of the cylinder, there is no effect on the angular velocity, ω, of the spacecraft. More precisely, an increase or decrease of ω cause by the astronaut walking toward or away from the center of mass of the vehicle is exactly compensated when the crew member return to his/her original position. The same result was obtained when the satellite was spinning and the astronaut moved inside the vehicle along the longitudinal axis. Using computer simulations, Poli investigated a number of general paths in all three dimension astronauts would take along a Project Gemini-size spacecraft.[1] Unlike in the planar cases, the angular velocity did not return to its original value when the astronaut returned to his/her original position since the three governing differential equations were coupled and nonlinear. In the planar case, there was only a single linear differential equation. The final values for ω were often around ± 1°/sec and in some cases as much as ± 1.9°/sec from the reference value. Poli also showed that the paths "walked" by the astronauts were possible from the standpoint of the forces required from the crew member.

3.2 Stochastic Models

The early models were *deterministic* in nature, i.e., that the effect of the crew motion was uniquely determined by a mathematical expression which was logical considering that the spacecraft-astronaut interaction was looked at as the interaction of point-masses. However, the crew disturbance forces and moments are highly complex and thus can cannot be precisely described. For this reason, modeling the disturbances as *stochastic processes* is analytically useful. While, no signal processing terms are explained at this time, some key terms and equations are discussed in Section 5.3.3.

In 1971, Hendricks and Johnson published the first statistical description of the disturbing forces and moments resulting from crew motion [17]. A mock-up console with a seat, display, and switches was built and placed on a load-cell array to measure forces and moments a test subject exerts while performing console operation tasks. Since the experiment was performed in a 1 g environment, the

1. A radius of 5 feet (1.5 m), a length of 10 feet (3.0 m), and a total mass of 7100 lb (3220 kg) was assumed.

"static" load component was removed, leaving "dynamic" forces and moments, when the subject had a velocity relative to the load cell array. It was assumed that these loads would be present in a weightless environment.

With the approximate 0 g wave forms determined, the time functions were transformed into the frequency domain via the Fast Fourier Transform (FFT) to obtain their power spectral densities. Depending on the shape of the PSD curves (unimodal or bimodal), a particular filter was chosen. The spectral density curve from the filter was adjusted through the filter parameters until a least-squares fit was obtained with the PSDs from the experimental data. According to the authors, "good approximations to the PSDs for all crew motions considered were obtained using a linear filter driven by white noise."

3.3 Skylab Crew/Vehicle Disturbance Experiment

The first spaceflight experiment carried out to measure crew disturbances was the Skylab Crew/Vehicle Disturbance Experiment (Skylab Experiment T-013) in 1973. Originally proposed in 1965, the objective of the experiment was to "assess the characteristics and effects of astronaut crew-motion disturbances aboard a manned spacecraft, and to investigate the response of the Apollo Telescope Mount (ATM) Pointing Control System (PCS) to known disturbance inputs." The primary motivation for Experiment T-013 was to aid the designers of stabilization and control systems of future manned spacecraft by verifying the mathematical models that had been developed [34], [35].

The experimental hardware consisted of three main components:

1. Limb Motion Sensing System (LIMS)

   The LIMS was an "exoskeleton" equipped with pivots at 16 principal body rotation points and worn by the astronaut over their regular clothes. Each pivot was equipped with a linear potentiometer so that the LIMS could provide a continuous measurement of limb position relative to the torso of the crew subject.[2]

2. Force Measuring System (FMS)

   The FMS consisted of two Force Measuring Units (FMU's). Each FMU consisted of a sense plate, a base plate, six load cells, load cell caging devices, a calibration check mechanism, and signal conditioning electronics. Both FMU's provided the ability to attach a portable handhold and one unit

2. More precisely, it measured the Euler angles (roll, pitch, and yaw) at the shoulders and hips as well as a single pitch angle at the elbows and knees.

contained foot restraints. The base plate was machined out of aluminum to maximize the stiffness to weight ratio. The system included isolation flexures at both end of each load cell to isolate it from nonaxial loading.

I

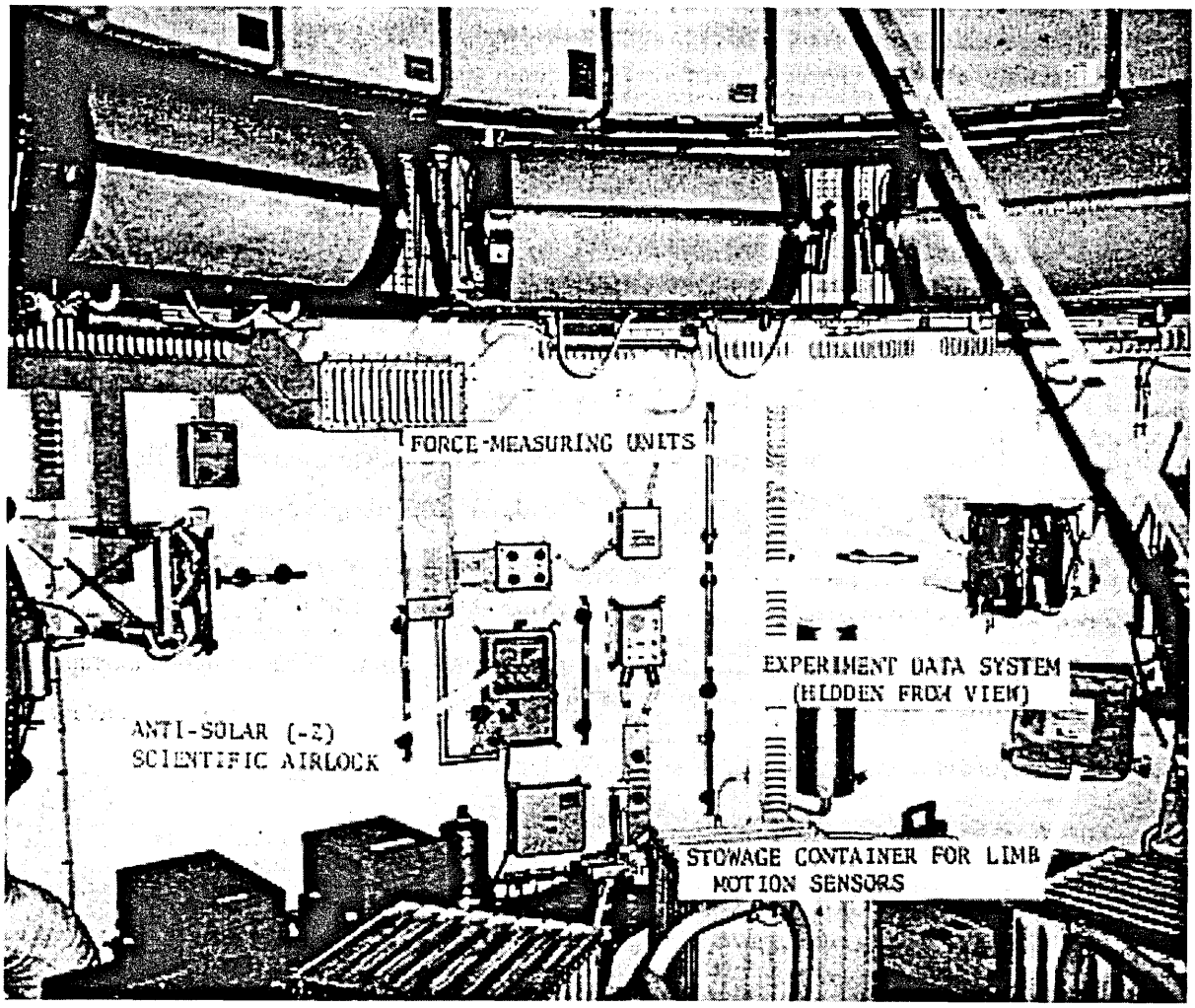


Figure 3.1: The photograph shows the set up of the Skylab Crew/Vehicle Disturbance Experiment set-up in the Skylab Orbital Workshop [35].

3. Experiment Data System (EDS)

The EDS was a computer system that recorded data from 31 analog channels (LIMS and FMS).

In addition, to experiment-specific hardware, a 16-mm camera was used to record the subject's center of mass and torso attitude.

The experiment was performed in the dome area of the Skylab Orbital Workshop (OWS) by Commander Alan L. Bean and Pilot Jack L. Lousma during Skylab-3, which was the third mission overall but the *second manned* mission. Almost all data was recorded on August 16, 1973 (Day-of-Year 228) during a period of less than 80 minutes. The crew had arrived on Skylab on July 28, 1973, that is almost three weeks prior to the experiment, and can be assumed to have already adapted their motion to the microgravity environment.

The test subjects were asked to perform several tasks, which fell into three basic categories:

1. Gross body motions

   These motions included arm and leg movements, breathing / coughing exercises, and soaring across the Skylab OWS from one force measuring unit to the other.

2. Simulated console operations

   The subjects were asked to perform motions typical of working on a console such as flipping switches, pushing buttons, typing, etc.

3. Worst case control system inputs

   In order to determine to maximum force and moments due to the crew that the ATM pointing control system would have to compensate for, one astronaut subject performed vigorous exercise-type motions while restrained to one FMU. For all activities other than soaring the astronauts used FMU 1 exclusively.

Alan Bean served as the only subject for most of the T-013 experiment. He was joined by Jack Lousma for the vigorous soaring activities. However, since only two FMU's between which to soar, the data returned is solely from Commander Bean.

The primary output of the Skylab T-013 crew motion experiment is shown in Figure 3.2. The bar graph shows the average and maximum recorded force for a range of activities. Note that the average force across activities is below 100 N.

Analysis of the recorded force data and an examination of the film, showed that the astronauts were able to achieve soaring velocities of up to 1.9 m/s (6.8 km/h or 4.3 m.p.h.). The separation between the two force measuring units was approximately 3.2 meters. The soaring produced up to 400 N in force and resulted in applied disturbance torques on the order of 1000 Nm which induced a vehicle rate on the order of 0.02 degrees per second as recorded by the Skylab attitude control system. In light of such

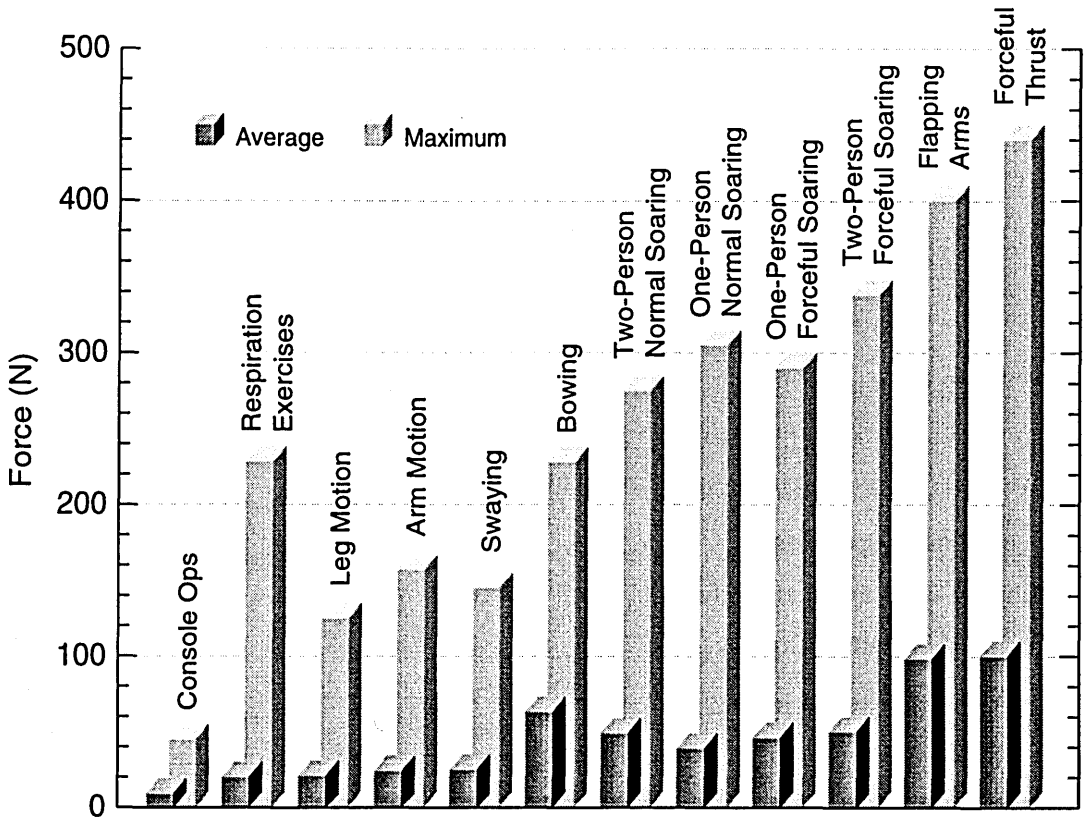


Figure 3.2: The figure shows the average and maximum forces measured for a set of activities during the Skylab Crew/Vehicle Disturbance Experiment T-013 in 1973 [25].

a high force-level, it is useful to examine it more closely. An excerpt from the experiment checklist describes the activity in detail [25]:

1. Release left foot from restraint and crouch for free soaring (use handhold to keep feet on FMU 1).
2. Push off from FMU 1 (with feet), soar to FMU 2, and stabilize with hands only.
3. Position feet on FMU 2, push off to FMU 1, and stabilize with hands only.
4. Push off FMU 1 with hands, turn, and stabilize at FMU 2 with hands only.
5. Push off FMU 2 with hands and return to FMU 1; stabilize with hands only.

What is important is that the subjects were asked to crouch keep themselves in place with a handhold and so maximized their soaring velocity. The resulting impact force is then expected to be very high. Examination of astronaut motions on the Shuttle and on Mir revealed that such a soaring is not typically executed by the crew.

Conway reported that the forces produced during the T-013 experiment on Skylab were generally higher than those measured in 1 g prior to the flight. In the respiratory exercises (breathing, coughing, sneezing), where the subject had one foot in the force sensor, only coughing produced the same force response as on the ground, while sneezing produced up to twice as much and deep breathing over 25 times as much force. Analysis of the LIMS data showed that the body limb motion in flight were executed approximately 35% faster than comparable motions simulated on the ground. The interpretation of this unintuitive observation was that "[a] lack of 1 g restraint on the subject's visceral mass, allowing more acceleration and motion of this mass, appears to provide reasonable explanation for the larger in-flight forces." [35] Video footage recorded during Skylab is sparse compared to the massive amount of Hi-8 video tapes available from every Shuttle or U.S. mission to Mir. An informal measurement of the speed that motions are performed, unveils that astronauts move slower in microgravity that they do on the ground. This observation is in stark contrast to the results obtained by Conway.

Analysis of the data from the Skylab T-013 experiment, continued after the release of the 1976 NASA technical report. The crew motion forces and moments were analyzed for statistical characteristics and frequency content and a handbook for incorporating crew motion effects into the design of a manned spacecraft control system was published in 1979 [25].

Two working models of crew motion disturbances were developed—for the preliminary design, a simple "first-order" model and for the detailed design a stochastic model. The first-order model is a simple time function that uses the peaks of an event, such as soaring, to characterize the disturbance and has smaller loads and noise set to zero as is shown in Figure 3.3. Then the forces and moments applied to the force measuring unit are transformed from the local FMU coordinate frame to the vehicle's coordinate system with its origin at the center of mass.

The stochastic model for T-013 data used the same approach as Hendricks and Johnson [17] described earlier. The power spectral density curves were computed for nine types of activities (console operations, respiratory exercises, deep breathing, arm motions, leg motion, bowing, arm flapping, crouch and push-off, crouch and straighten, and soaring). Unimodal spectral densities were approximated by a single quadratic in the transfer function

$$H(s) = \frac{\tau s}{s^2 + 2\zeta\omega s + \omega^2} \tag{3.1}$$

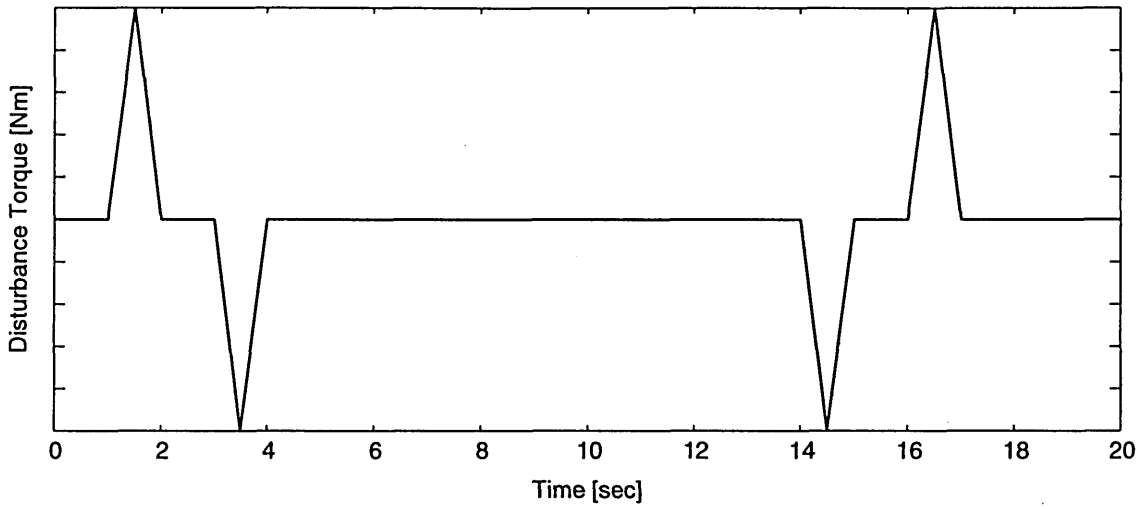


Figure 3.3: The plot shows the deterministic first-order model for soaring. It is a time function consisting of triangular pulses to represent the subject's push-off from one force measuring unit and the subsequent landing on the opposite one. [25]

where τ is the gain, ω the frequency, and ζ the damping.

For the approximation of a bimodal PSD curve, a fourth-order polynomial in the denominator of the transfer function was used:

$$H(s) = \frac{\tau s}{(s^2 + 2\zeta_1\omega_1 s + \omega_1^2)(s^2 + 2\zeta_2\omega_2 s + \omega_2^2)} \tag{3.2}$$

While many of the PSD curves of the spaceflight data contained more than one peak, the quadratic transfer function gave satisfactory results. The parameters τ, ω, and ζ are tabulated in the report and plots of the original and simulated PSD curves are shown.

In June 1991, a small pilot study was undertaken to quantify the forces induced by astronauts during push-offs and landings [24] The investigation used NASA's KC-135A Reduced Gravity Simulation Aircraft[3]. This, specially modified, aircraft flies a series of parabolic flight maneuvers to create approximately 20–25 seconds of weightlessness during each parabola. The load measurements were taken with a Kistler[4] 6-degree-of-freedom force plate mounted to an aluminum plate, which was fastened to

3. The aircraft is commonly referred to as the "Vomit Comet," but officially designated NASA 930.
4. Kistler Instrumente AG in Winterthur, Switzerland.

the aircraft floor. The force plate had a size of 40 cm by 60 cm and used piezoelectric load cells. The load data for each event was sampled over a period of 15 seconds with a frequency of 250 Hz.

The test protocol involved four crew motions—two using the feet and two using both hands. In the foot push-off, the subject pushed off with both feet from the force plate and translated vertically towards the cabin ceiling. In the analogous landing, the subject used his/her hands to push off from the aircraft ceiling translate vertically down and land with both feet on the force plate. In the hand push-off, the subject laid down on the floor and placed both hands on the force plate near the hips, and pushed off to translate toward the ceiling. A vertical hand landing involved pushing off the aircraft cabin ceiling and catching the force plate with both hands. The protocol is illustrated in Figure 3.4; in each case the primary load path was in the z-direction (vertical).

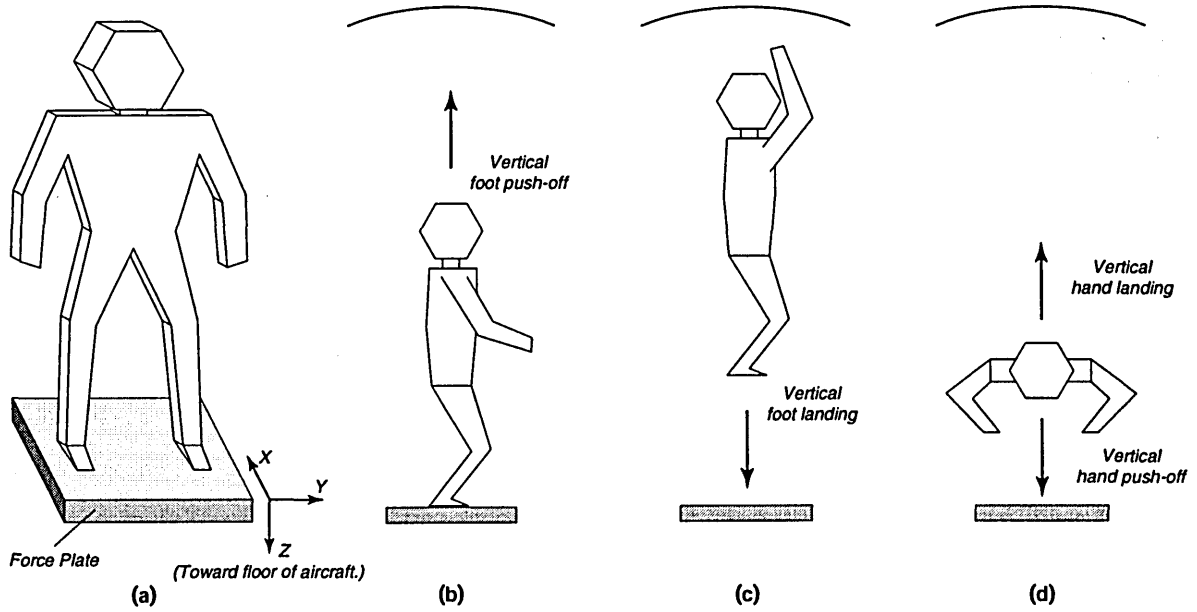


Figure 3.4: (a) This picture shows the axes of the force plate used in the KC-135 pilot study, **(b)** show the vertical push-off from the force plate toward the ceiling, **(c)** shows the vertical landing on the force plate, **(d)** shows how the vertical push-off / landing with the hands was performed. Adapted from [24].

Five subjects were chosen to represent a crew population ranging from a 5th percentile Japanese female and a 95th percentile American male, as defined in NASA's Man-Systems Integration Standards[5]. Four of the subjects had flown over 400 parabolas, while one subject had no experience in

5. The Man-Systems Integration Standards (MSIS) can be found on the World Wide Web at the following URL: http://www-sa.jsc.nasa.gov/FCSD/CrewStationBranch/Msis/online.htm

weightlessness. The results of the experiment are summarized in Table 3.1. Since the KC-135A aircraft has significant vibrations, baseline conditions with no activity were recorded and are also shown in the table. The accuracy level was estimated to be ±13 N.

Table 3.1: Crew Induced Forces in a KC-135 Zero-g Aircraft Pilot Study [24]

		Force in X-Axis [N]			Force in Y-Axis [N]			Force in Z-Axis [N]		
Crew Activity	No. of Events	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	Max.
Foot Push-off	5	22	71	169	22	40	67	111	311	534
Foot Landing	1		67			31			200	
Hand Push-off	4	~0	49	111	~0	44	133	67	151	267
Hand Landing	3	22	31	44	31	36	44	36	102	178
"Baseline"	2	±9	±9	±9	±9	±11	±13	±13	±16	±18

As the data in Table 3.1 show the forces in the vertical (z) direction were quite high but the ability to perform specific motions in zero-g depends on the subject's prior experience in weightlessness. The subject with no prior experience in the KC-135 aircraft produced some of the largest forces during the activities. The experience on the KC-135 is not comparable to an actual sapce flight experience. During an ingress test for a proposed Assured Crew Return Vehicle for the space station on the KC-135, the performance of STS-40 astronauts was compared with that of experienced KC-135 fliers. The former were observed to move about much more easily in zero-g than the latter. This supports the hypothesis that veteran astronauts exert much smaller force on the spacecraft than unexperienced astronauts or subjects in ground tests [24].

CHAPTER

4

THE DLS/EDLS SPACEFLIGHT EXPERIMENTS AND THEIR TECHNOLOGY

This chapter deals with the Dynamic Load Sensors (DLS) experiment on the Space Shuttle and its successor the Enhanced Dynamic Load Sensors (EDLS) experiment on Mir. Section 4.1 introduces DLS and presents its key results. Section 4.2 addresses the motivation behind the EDLS experiment and its objectives. In addition, an description of how the experiment was conducted and a timeline are provided. However, the post-flight efforts to process and analyze the EDLS data is the focus of Chapter 5. DLS and EDLS made use of the same technology; Section 4.3 is devoted to the requirements for the DLS/EDLS systems and the design of the experimental hardware and software.

4.1 Dynamic Load Sensors Experiment on STS-62

The Dynamic Load Sensors (DLS) experiment was flown on the STS-62 Space Shuttle Mission (March 4–18, 1994) in conjunction with the re-flight of the Middeck 0-Gravity Dynamics Experiment. Known by the acronym MODE, the experiment was developed at MIT's Space Engineering Research Center to study the dynamics of large space structures and the gravity-dependent nonlinear behavior (sloshing) of contained fluids typical of spacecraft fuels. MODE's inaugural flight was on STS-48 in September 1991. It was the first university experiment to fly under NASA's In-Space Technology Experiments Program (IN-STEP) outreach effort to allow universities, industry, and the government to develop small, inexpensive flight experiments. The structural test article was a truss model whose response to vibrations in microgravity was examined [68], [70].

In the re-flight of MODE, the structural aspect of the experiment was kept but the fluids component was replaced with a new investigation—the Dynamic Load Sensors experiment proposed by Professor Dava Newman. The purpose of DLS was *to quantify the forces and moments exerted by the astronauts on the orbiter middeck as he/she is going about their normal on-orbit activities*. The key hardware component of DLS was a set of three sensors—a touchpad, a foot restraint, and a handhold—similar in

appearance to the devices found in the orbiter to assist the crew in moving about the vehicle (see Figure 1.1 on page 21). The electronics hardware, called Experiment Support Module (ESM), responsible for signal conditioning, data acquisition and storage was shared by MODE and DLS. The sensors were installed in a heavy crew traffic area—the orbiter middeck. The foot restraint was placed among the regular foot loops on the floor in front of the middeck lockers. The handhold was placed on the door of a middeck locker, and the touchpad on the middeck augmentation rack. The foot restraint and the handhold were six-degree-of-freedom sensors able to measure forces along three axes and moment about the three axes, while the touchpad was restricted to recording only the three force-components due to limitations in the signal conditioning hardware as will be explained later in more detail. The arrangement of the sensors in the middeck is illustrated in Figure 4.1.

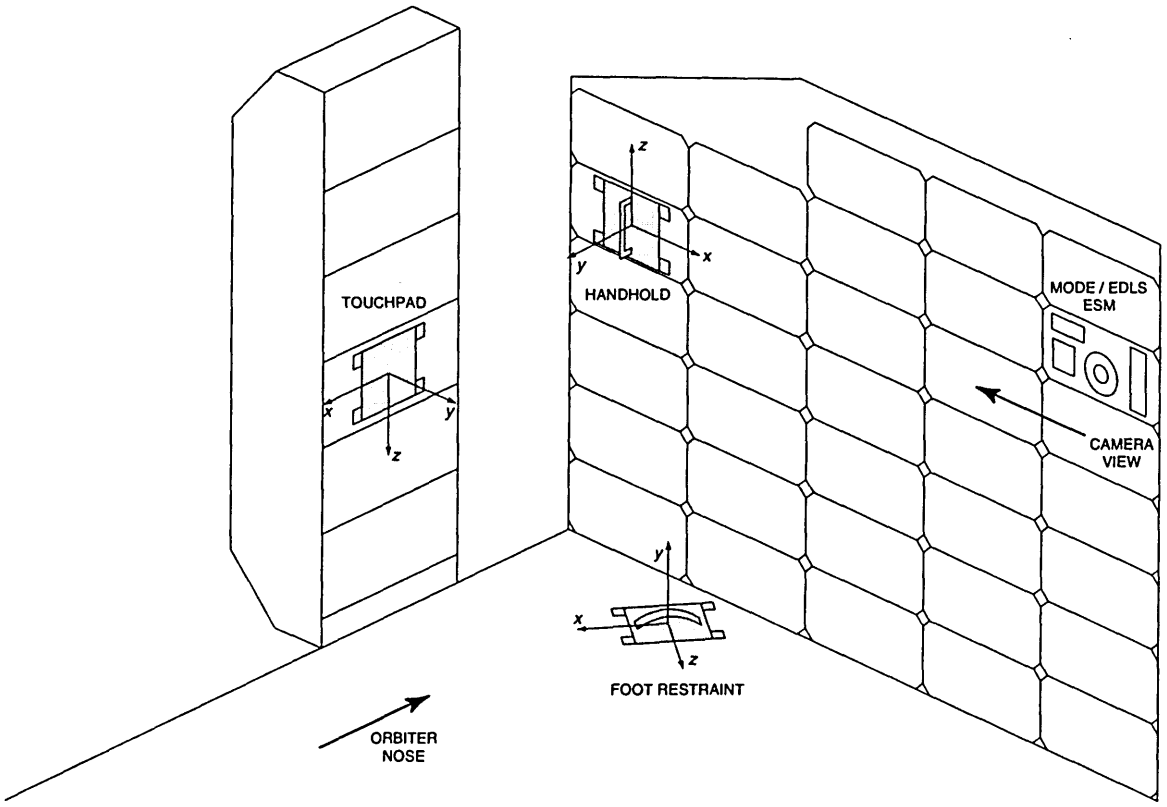


Figure 4.1: The drawing shows the location of the DLS sensors in the Space Shuttle orbiter middeck during mission STS-62. The direction of the orbiter nose and the orientation of the DLS axis systems are marked [27], [64].

DLS data was taken on Flight Day (FD) 7, 8, and 11 of the 14-day mission. The software of the ESM was designed in such a way that once it was activated, it recorded data continuously until manually terminated or all available disk space was taken.

After the flight a FORTRAN program filtered the data with a 3-pole elliptical filter (30 Hz corner frequency) and sweeped through the data searching for events. In DLS/EDLS terminology, an *event* is understood as either a specific crew motion typically lasting no more than a few seconds or several crew motions together without an interruption in the loading to a sensor. When the FORTRAN program detected that the force or moment magnitude first became larger and then smaller than certain force/moment magnitude levels, an event was located and the data snippet written to a file. The crew of STS-62 consisted of five astronauts and at least four of them used the sensors and hence contributed to the data collected.[1] Figure 4.2 shows astronaut Sam Gemar demonstrating the use of the foot restraint and handhold sensors.

Figure 4.2: This video footage from the Space Shuttle Mission STS-62 shows the DLS foot-restraint (left image) and the DLS handhold (right image) used by the astronauts. The sensors were mounted in the orbiter middeck; the rectangular "containers" visible in both images are the middeck lockers (NASA STS-62 Onboard Video ID#19).

From over 67-hours of DLS data more than xxx event files were extracted. Of these 1168 were further processed and in those 301 astronaut motions were found through a correlation with the videofootage recorded. However, correlating video and force/moment data was difficult because sometimes it was not possible to synchronize the video recording with the data precisely and at other times astrnauts

1. Since video was not recorded all times, it is impossible to say whether the fifth astronaut used the sensors or not.

were obscuring the view of the camera. The shape of the data plots helped the process of identifying specific crew motions. For example, a small steady load on a foot restraint followed by a short larger load and then just electronic noise is very likely to be a push-off from the sensor.

Overall seven characteristic motions that the astronauts performed in the weightless environment of the Shuttle were identified from an examination of the video tapes and the events sorted accordingly. All seven motions involved the hand hold and foot restraint and four of them included touchpad usage. The characteristic astronaut motions are summarized in Table 4.1. It is important to point out that the sensors listed for each type of motion in the table are those sensors that were used but not necessarily in the same event.

Table 4.1: Characteristic Astronaut Motion Identified in DLS Experiment

Characteristic Motion	Description	Sensor
Landing	Crew flying across middeck and landing on (utilizing) the sensor	HH, FR, TP
Push-off	Crew pushing off and flying	HH, FR, TP
Flexion / Extension	While using sensor, flexing or extending limb	HH, FR
Single Support	Using only one limb for support	HH, FR
Double Support	Using two limbs for support	HH, FR
Twisting	Twisting body motion	HH, FR
Orienting	Re-orienting oneself usually during posture control	HH, FR, TP

Analysis of the crew disturbances revealed that the average root-mean-square (rms) value for the reaction forces was 24 N and the average root-mean-square (rms) value for the reaction moments was 3.3 Nm as the astronauts performed their daily activities. The highest forces recorded by the foot restraint, handhold, and touchpad were 466 N, 75 N, and 153 N respectively [27]. Figure 4.3 summarizes the magnitude of the forces recorded in the DLS experiment. It shows a histogram of the peak force magnitudes for the 301 events that were correlated with the video footage from the STS-62 mission. [I WILL PERFORM MORE STATISTICAL ANALYSIS HERE. I SPOKE WITH DR. NATAPOFF ABOUT THAT. I WILL ALSO WRITE ABOUT THE 466 N PEAK. AND THE FACT THAT WE DO NOT REALLY KNOW WHAT IT IS.

Figure 4.4 shows a histogram of the rms forces recorded for the 301 events.

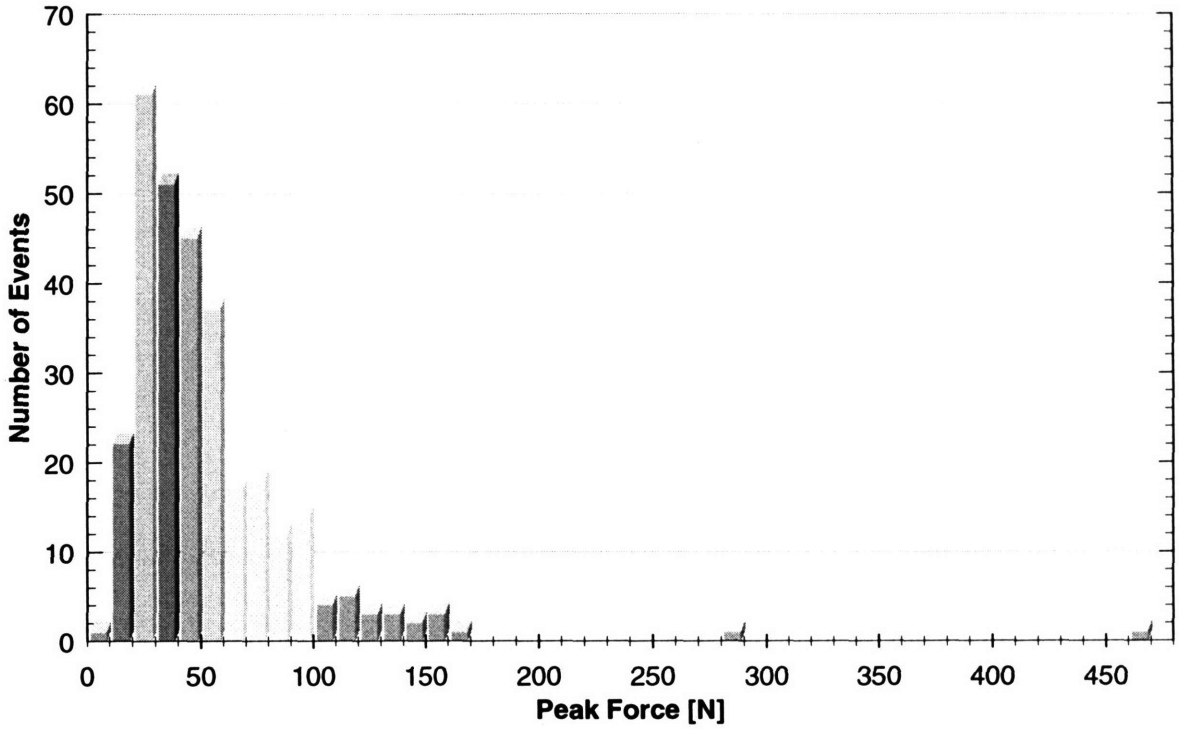


Figure 4.3: This histogram shows the peak forces recorded for 301 astronaut motions from all three flight days DLS data was recorded.

4.2 The Enhanced Dynamic Load Sensors Experiment on Mir

4.2.1 Motivation

The importance of a quiescent environment for microgravity research on the International Space Station and the resulting requirements have been discussed at length in Sections 2.3 and 2.4. The importance of investigating crew motion disturbances and the effects they have on a spacecraft was presented in Chapter 3. While measurements have been made to characterize the overall acceleratory environment on the Space Shuttle and later on Mir, there has been no quantification of crew-induced reaction forces and moments since the Skylab experiment until DLS was performed on STS-62 in 1994.

In designing the ISS, prime contractor Boeing and NASA adopted the maximum force level from the T-013 Skylab experiment as the peak load to be expected from the astronauts. From an engineering perspective, this was reflected on a large scale in the finite element models of ISS modules as well on a small scale in the design of individual components such as windows. From a scientific perspective, the

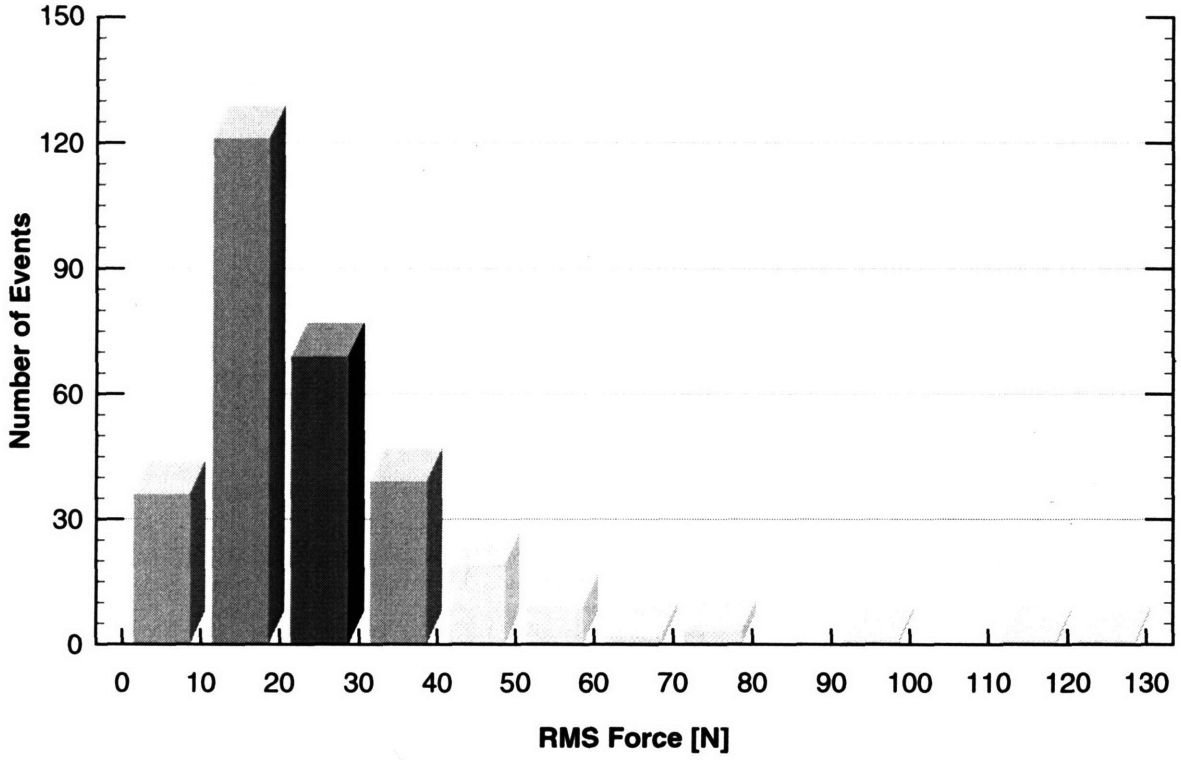


Figure 4.4: This histogram shows the root-mean-square forces recorded for 301 astronaut motions from all three flight days DLS data was recorded.

need of multi-billion dollar isolation systems for microgravity experiments was postulated. These systems would not only represent a substantial addition to the cost of an experiment but also to its complexity. For these reasons it was important to measure the crew effects and help station designers as well as payload planners.

Exclusive use of the Skylab data for the ISS was considered questionable for several reasons:

- The Skylab subjects executed a set of *prescribed* activities that did not reflect nominal crew motion.
- The peak loads of the experiment, which were in excess of 400 N, arose from soaring activities that were meant to measure the *worst* case scenario and did not reflect an activity that astronauts would perform in an quiescent environment.
- The entire data set was basically recorded on a *single day in a single session* lasting about 78 minutes. Two short additional sessions were conducted to address anomalies found after the first run.
- The data came from a *single* astronaut subject.[2]

The EDLS experiment[3] was proposed in May 1994 to expand upon the database acquired with DLS and by collecting crew forces and moments on the Russian Mir space station. The size and layout of the Space Shuttle middeck and the number of astronauts subjects available did not reflect the prevalent situation one would find in a long-duration space flight on a space station. Since Space Shuttle missions so far have never lasted more than 17 days[4], there is insufficient time to observe crew adaptation to weightlessness and learning effects as will occur on ISS. By recording EDLS data on Mir starting with the arrival of a new crew over the course of their entire stay, unique data on the development of zero-g adaptive strategies by crew members in modules with larger volumes would be obtained.

Since the Space Acceleration Measurement System (SAMS) was scheduled to fly on Mir, the opportunity existed to correlate the regular crew-induced forces and moments with the acceleration measurement of the Mir space station and establish a transfer function for a Mir module or even the entire complex.

4.2.2 Objectives

The motivation stated before led to the establishment of two objectives for the EDLS experiment. The primary objective was "to assess nominal crew-induced reactions for long-duration space station missions" with the ultimate goal "to provide a human factors assessment of crew reactions for engineering design requirements. [65]" The secondary objective of the research effort was "to provide a detailed model that identifies and characterizes the adaptive control strategies adopted by crew members in microgravity. [65]" The subordinate physiological control modeling endeavor is a scientific rather than an engineering effort and beyond the scope of this thesis.

Based on the above stated primary objective, three specific goals were set for EDLS:

1. To quantify the nominal crew-induced forces and moments during an extended stay in orbit.
2. To quantify changes in the forces and moments over time as the crew adapts to microgravity.
3. To characterize typical astronaut motion in microgravity.

2. While two astronauts performed the soaring exercises, force data was recorded for only of the two.
3. Originally, the experiment was called DLS on Mir ("DLS-Mir"). However, the name was later changed to Enhanced Dynamic Load Sensors (EDLS).
4. The Space Shuttle orbiters were upgraded into Extended Duration Orbiters (EDO) to prolong their on-orbit stay. While EDO's may one day extend Shuttle flights to as much as 30 days, the longest Shuttle mission so far was STS-78, which lasted 16 days and 21 hours.

4.2.3 Experiment Chronology

The EDLS experiment used the same hardware as DLS on the Shuttle. However, a second foot restraint (for a total of four sensors) was fabricated since it was the most frequently used sensor during DLS and necessary to conduct the "active" operations sessions. Because of the long-duration aspect of the EDLS experiment, the software of the ESM was modified to include *event detection*. By setting an appropriate threshold level, the data acquisition system would operate continuously once activated by the crew but record only useful events and not waste precious disk space on electronic noise.

In preparation for the experiment, U.S. astronauts Shannon Lucid and John Blaha were trained in June 1995 to run the EDLS experiment. Their collegues Jerry Linenger and Michael Foale were in December of 1995.

The EDLS hardware was delivered to the Russian Space Agency in and installed in the lockers of the Priroda module on the ground prior to launch on April 24, 1996. U.S astronaut Shannon Lucid (NASA 2 / Mir 21 / Mir 22 Mission), who had arrived about a month earlier on Mir than the module, set-up the EDLS hardware in Priroda. Default positions for the sensors had been determined prior to the launch of the Priroda module and labels were affixed to the floor and walls indicating where the EDLS sensors should be placed. However, the crew was encouraged to move the sensors around with them to the places where they work.

The astronauts were asked to conduct two types of sessions: "passive" and "active." In a passive session, data acquisition was activated and the crew went about its regular business and whenever, the forces/moments exceeded the threshold specified in the protocol, the data was recorded on the storage medium. In the active sessions, the astronauts used one or more foot restraints and conducted a throwing experiment. In the experiment the astronaut subjects were asked to throw a small ball at a target several meters away with their eyes either open or closed. The purpose of the experiment was to quantify the adaptation in human motor control due to the lack of gravity.

The first EDLS data was taken on May 23, 1996. Until August 1996, Shannon Lucid and her Russian colleagues Commander Yuri Onufrienko and Flight Engineer Yuri Usachyov performed about twenty EDLS sessions.[5]

5. The Mir 22 crew arrived on the station on August 19, 1996 but did not participate in EDLS.

After August 28, 1996, the MODE ESM failed to activate and prevented the continuation of the experiment during the NASA 3 mission of astronaut John Blaha. The cause of the failure was unknown but since the sensors were assumed to be fully intact, an ambitious workaround, using a different ESM, was proposed. The McDonnell Douglas Company (now part of The Boeing Company) was investigating the structural dynamic response of the Russian orbital complex in the Mir Structural Dynamics Experiment (MiSDE). The MiSDE hardware on Mir, which is collectively known as the Mir Auxiliary Sensor Unit (MASU), included an advanced version of the MODE ESM, built by the same manufacturer. An EDLS/MASU adapter was built that allowed connecting the EDLS umbilical (to which the sensors are connected) to the MASU ESM and the data acquisition software was modified to record both MiSDE and EDLS data. The adapter cable[6] as well as the new software was brought to Mir on STS-81 (12–22 January 1997) bringing the EDLS experiment back into operation in time for the NASA 4 mission of astronaut Jerry Linenger. From February to May 1997, Jerry Linenger and Russian cosmonauts Vasili Tsibliev (CDR) and Alexander Lazutkin (FE) performed about seventeen EDLS sessions.

The location and arrangement of the sensors depended on whether a session was active or passive and where the astronaut would work in the case of a passive session. A common configuration was the placement of two foot restraints in front of the microgravity glovebox or in front of a laptop computer which was mounted on a wall. Both set-ups are shown in Figure 4.5

The handhold remained mostly mounted next to the microgravity glovebox. Typically the astronauts used it to "guide" themselves along the module as is shown in Figure 4.6. Another common use of the handhold was as an aid to the astronaut to come to a stop. In this case, the handhold was grabbed with one hand and the feet swung down to the floor to come to a halt.

While the touchpad was intended to be mounted on a wall for push-offs, the Priroda module become cluttered so quickly that soon little space was available to mount the sensor to a wall. More importantly, however, since the touchpad had no extra functionality, the crew would use any little bit of wall surface available for pushing off. Jerry Linenger placed the sensor underneath a standard Priroda rail and so converted the touchpad in essence into a foot restraint, which he used in conjunction with a

6. The adapter is (jokingly) referred to as "Mr Ed" which stands "MiSDE Repair of EDLS."

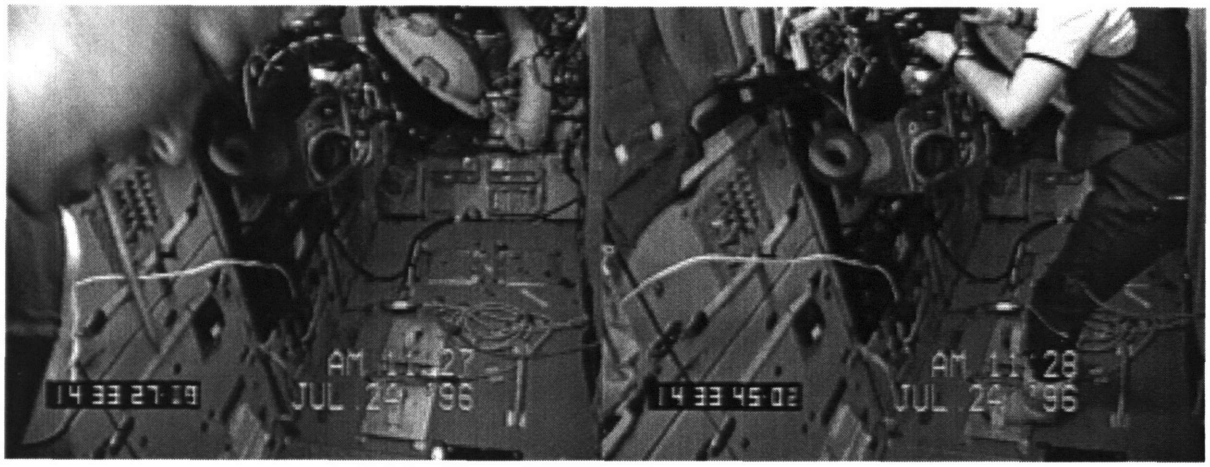


Figure 4.5: These to snapshots from a videotape recorded during NASA 2 / Mir 21 show the two EDLS foot restraints mounted in front of the microgravity glovebox. The astronaut whos just turns away from the camera in the left image and who uses the sensors in the right image is Shannon Lucid (NASA Mir-21/22 H-8mm Onboard Video ID#35).

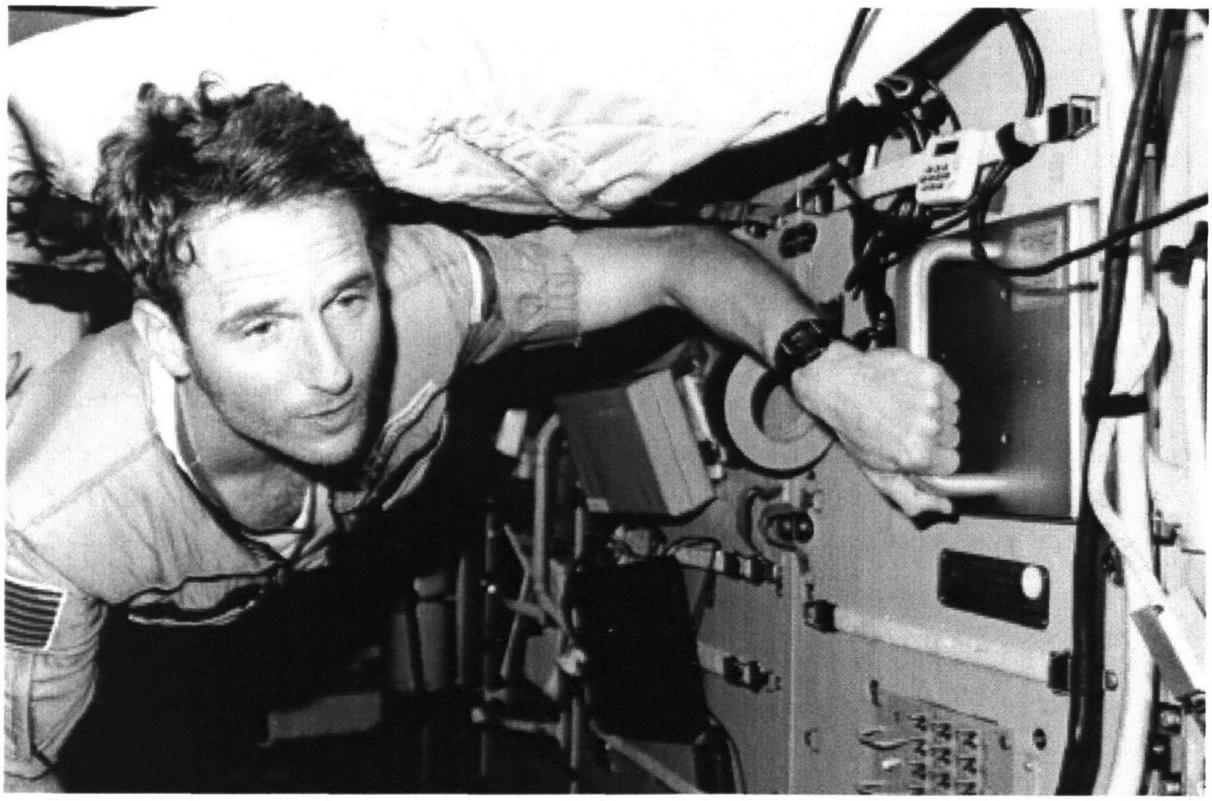


Figure 4.6: The image shows U.S. astronaut Jerry Linenger (NASA 4) mission use the EDLS handhold to guide himself in the Priroda module of Mir. (NASA Image NM23-01-013)

"real" EDLS foot restraint. He preferred this set up over a regular foot restraint because it kept his foot firmly in place as seen in Figure 4.7.

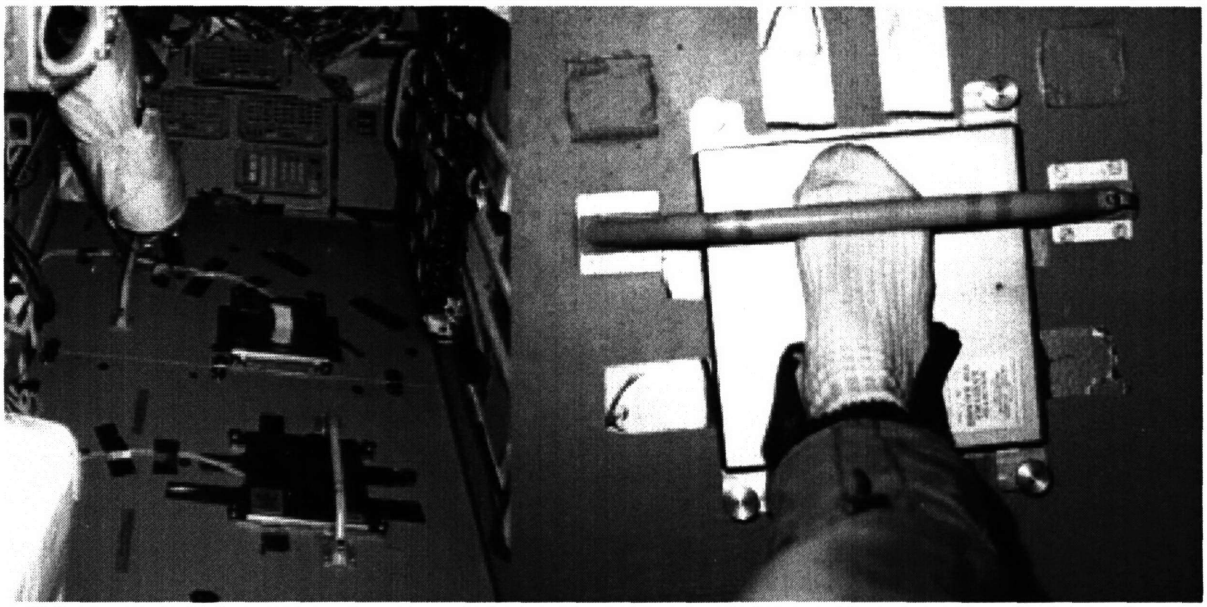


Figure 4.7: The left image shows foot restraint 1 (EDLS Sensor #2) as well as touchpad placed underneath a Priroda handrail. In the right image, Jerry Linenger's right foot is tightly secured in the "modified" touchpad (NASA Image NM22-38-037, NASA Image NM23-01-004).

Detailed information on when each EDLS session was held, what the sensor location and configuration was, is provided in Section 5.2.

In June of 1997, a *Progress* resupply vehicle collided with Mir's *Spektr* module. While much of the U.S. experimental hardware was located in that module, all EDLS equipment was in Priroda and hence not directly affected by the accident. However, due to the many technical difficulties as a result of the collision and others following it for unrelated reasons, no crew time was available to record additional EDLS data during NASA 5. The experimental hardware was returned to Earth on Shuttle flight STS-89 in January 1998 and returned to MIT in May 1998. Table 4.2 provides a timeline of the EDLS experiment.

4.3 DLS/EDLS Experimental Hardware and Software

The experimental hardware for MODE was designed and fabricated by Payload Systems Inc. (PSI), a small Cambridge, Massachusetts based company founded in 1984 to provide science and engineering services for spaceflight experiments. Because of their familiarity with the MODE hardware and the

Table 4.2: Timeline of the EDLS Spaceflight Experiment

Date(s)	Event(s)
10 May 1994	Proposal for EDLS submitted to NASA's Office of Life & Microgravity Sciences Applications, Life and Biomedical Sciences and Applications Division
29 September 1995	EDLS flight hardware acceptance test completed in Russia (Test performed included power consumption, functional, electrical and insulation.)
6–8 December 1995	EDLS Cosmonaut/Astronaut training for NASA 2 / Mir-21, Mir 23, and NASA 4 conducted in Russia.
23 April 1996	Mir Priroda Module with EDLS hardware (EDLS ESM, four sensors, one umbilical, 10 WORM disks) on board launched from Baikonur Cosmodrome in Kazakhstan.
26 April 1996	Priroda Module docks with Mir orbital complex.
23 May 1996	EDLS sensors set up in Priroda module
23 May through 27 August 1996	NASA 2 Astronaut Shannon Lucid performs about 20 EDLS sessions.
28 August 1996	MODE ESM fails to activate due to an unknown reason.
26 September 1996	Data from NASA 2 mission returned to ground on STS-79
15–20 January 1996	Adapter cable, PC Cards with an EDLS protocol for the MASU ESM transferred to Mir during the docking of orbiter Atlantis (STS-81).
25 February through 9 May 1997	NASA 4 Astronaut Jerry Linenger performs about 15 EDLS sessions.
24 May 1997	Data from NASA 4 mission returned to ground on STS-84
2 July 1997	MODE/EDLS ESM returned to MIT
31 January 1998	EDLS Hardware returned to ground on STS-89
21 April 1998	All remaining EDLS hardware returned to manufacturer.

successful execution of the experiment, PSI was selected by MIT as the primary subcontractor for DLS, responsible for the design, fabrication, modification, and integration of the hardware and software elements. It also aided in the crew training and performed mission support. PSI continued its role during EDLS.

4.3.1 Load Sensors

The key hardware component of the experiment is the set of four load sensors: a handhold, a touchpad, and two foot restraints. The sensors were designed to provide the same functionality as the foot loops and hand rails on the Space Shuttle orbiter for astronaut use and be used interchangeably. The touchpad's functionality was envisioned to be a that of a flat surface the astronaut would use to push them-

selves off using either their hands or their feet. All four sensors used for EDLS are shown in Figure 4.8.

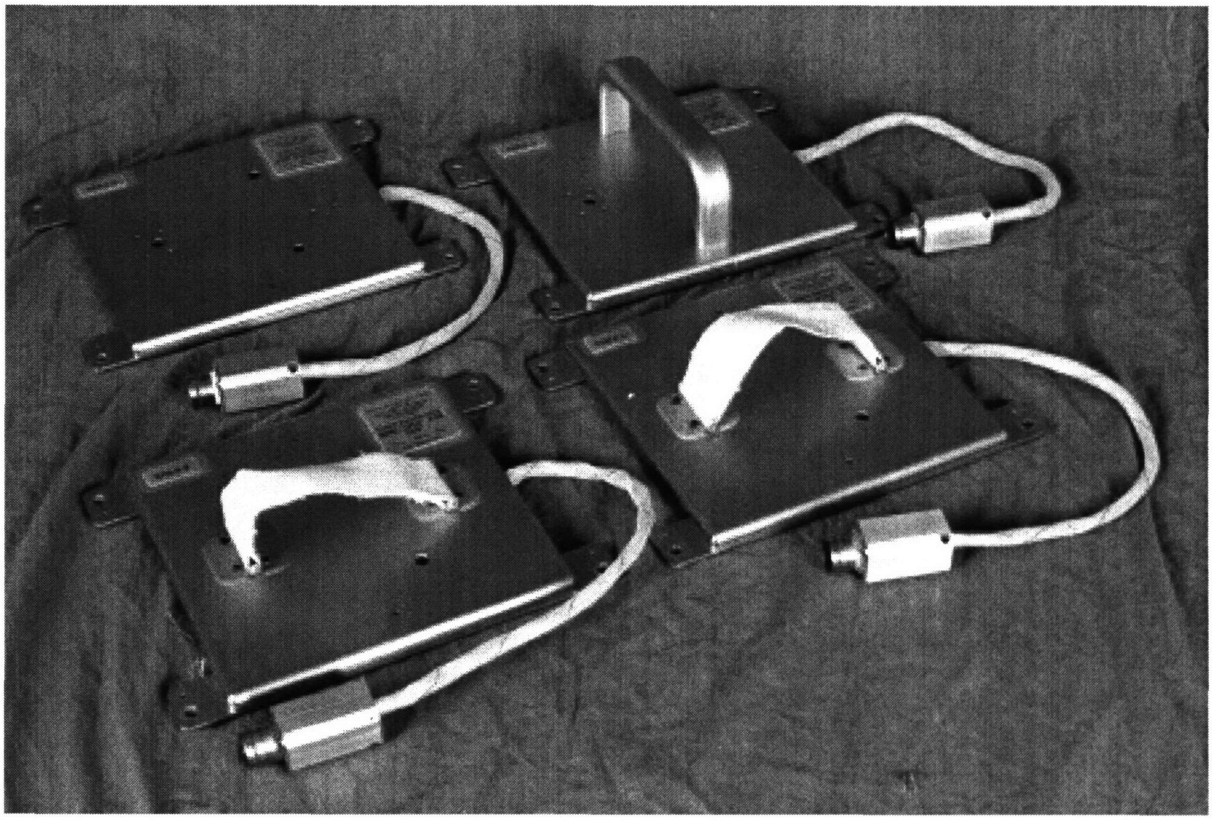


Figure 4.8: The photograph shows the four EDLS sensors: one touchpad, one handhold, and two foot restraints (NASA Image JSC S95 17030).

The sensors used for DLS and EDLS will be referred to as the *original* or *first-generation* sensors, while those designed as part of this thesis, will be called *advanced* or *second-generation* sensors. The rest of this section reviews the requirements and design of the original sensors. Requirements and specifications of the advanced load sensors are the subject of Section 6.4.

Requirements

In order to meet the objective of assessing crew disturbances, the load sensors were required to have six degrees of freedom (dof's) and thus be able to measure the applied forces in the x-, y-, and z-direction as well as the three moments about these directions. Due to the limited number of data acqui-

sition channels available on the MODE ESM, the touchpad sensor was restricted to measure only the three force components.

The requirement for the maximum force capability of the sensors was loosely based on the peak loads experienced in the Skylab experiment and the data gathered by Klute (see Table 3.1 on page 62). It was set at 400 N. The corresponding maximum moment was derived from the largest moment arm possible on the sensor from the sensor geometry and set at 50 Nm. These loads are the so-called *full scale* and measuring errors are referenced with respected to them. The minimum force resolution was determined by simply using Newton's second law for the space station. Based on a 1991 mass estimate of 227,000 kg (500,000 lb) for Space Station Freedom and a general microgravity requirement of 1 μg for experiments, the minimum force resolution for the sensors was calculated to be 2.2 N and conservatively set at 2 N. The moment resolution requirement was obtained by scaling the force resolution requirement of 2 N by factor of ten, which is a little bit more conservative than the ratio of full scale force to full scale moment. From the Skylab data, it was determined that most of the disturbances frequency content was below 60 Hz, which led to a requirement that lowest structural mode of the sensor had to be above that level and that the minimum sampling frequency had to be at least twice that[7] [65], [67].

The size of the sensors was limited to the footprint of the orbiter foot loops and the orbiter handholds (**NASA GSC-MDK-IDD xxx**) The requirement for the sensor height was 0.75" (19 mm). The load sensor requirements are summarized in Table 4.3.

Table 4.3: Requirements for DLS/EDLS Sensors [65]

Description of Requirement	Value
Maximum expected force in x, y, or z direction ("full scale load").	400 N
Maximum expected moment about either one of the 3 axes ("full scale load").	50 Nm
Minimum force resolution in x, y, or z direction.	2 N
Minimum resolution of applied moments about either one of three axes.	0.2 Nm

Specifications and Design

All three sensor share the same common design to simplify manufacturing, assembly, exchangeability, and reduce cost. The sensors are basically square in shape (measuring 9.5 by 9.5 inches or 24.1 by

7. The Nyquist Sampling Theorem states that the sampling frequency has to be higher than the Nyquist rate, which is twice the signal frequency, in order to capture the characteristics of the signal.

24.1 cm) and have small protrusion for screws to attach them to a floor/wall. They consist of six basic components, which are identical between all four sensors:

- Bottom plate (minimum thickness of 0.125" or 3.175 mm)
- Three loadcells (also called flexures)
- Top plate (0.125" or 3.175 mm thickness)
- Connector cable (approx. 37 cm in length; approx. 44 cm with connector)

The handhold has in addition a handle and the foot restraints cloth loops mounted on the top plate. The three load cells are secured to the bottom plate via screws. In addition, pockets are machined into the plate securing the load cells. This feature prevents a rotation of the individual load cell ("mechanical stop"). The sensors, including the loadcells, are made from aluminum AL 7075-T63 with an Iridite finish per Mil-C-5541, Class 3 specifications. AL 7075 is a NASA approved material and the T63 heat-treated form has a yield strength of 420 MPa. Machine shop drawings of the sensor components can be found in Appendix B.

A finite element model (FEM) was used to predict the modal characteristics of the sensors. For the analysis, it was assumed that the bottom plate was rigidly attached to the orbiter via screws in the protrusions. The lowest predicted modal frequency is 90 Hz and thus fifty percent above the required minimum. The results of the structural analysis are show in Table 4.4.

Table 4.4: Structural Design Specifications for EDLS Sensors

Specification	Vertical Direction	Rotation
Deflection Limit	1.00 mm	1.25 mm
Max. Deflection at design loads / moments	0.26 mm	0.81 mm
Von Mises Stress at Limit Deflection	240 MPa	284 MPa
Factor of Safety (σ_{yield} = 420 MPa)	1.7	1.5
Force that will result in limit deflection	1,515 N	617 N
Moment that will result in limit deflection	189 Nm	77 Nm
Lowest Predicted Structural Modes	90 Hz, 116 Hz, 129 Hz, 213 Hz, 217 Hz, 261 Hz,	

The sensing of the forces and moments is accomplished via strain gages on the loadcells. The geometry of the loadcell was determined iteratively by modeling the complete sensor with a finite element

model. Each sensor has three load cells arranged along a circle each 120° apart as is shown in Figure 4.9.

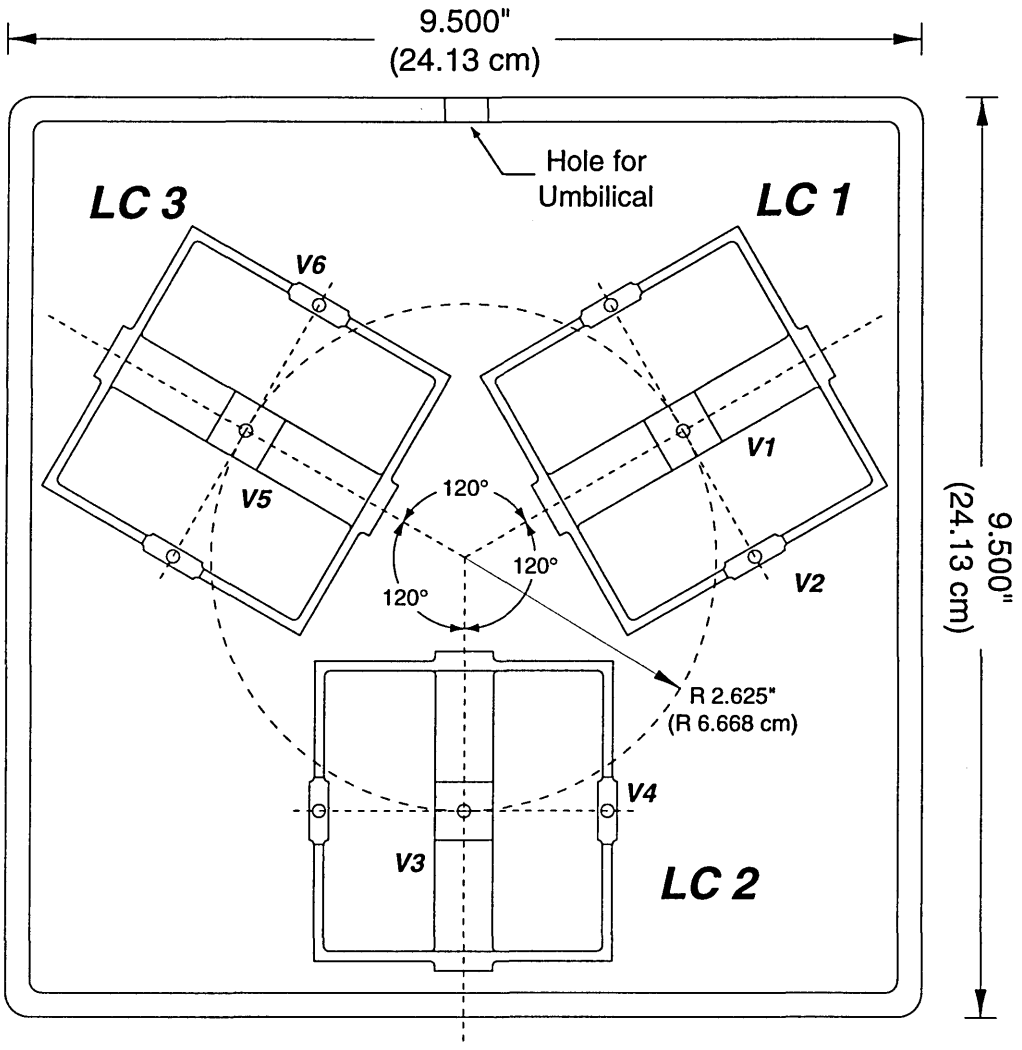


Figure 4.9: The figure shows the arrangement of the three load cells inside each sensor. V1 through V6 are the six voltages that are measured.

Each loadcell is equipped two *full wheatstone strain gage bridges* each consisting of four strain gages. Each full bridge provides one of the six signals. Signals V1, V3, V5 are from the bridges on the center beams and signals V2, V4, and V6 are from bridges on the side beams.

Full strain gage bridges are in general rare due to the high material and labor cost involved. Instead half or quarter bridges are used where simple resistors are used to complete the bridge. By using full bridges, the sensitivity is nearly doubled and most if not all temperature effects are compensated. The

gages were general-purpose foil-type strain gages widely used in experimental stress analysis. Manufactured by the Measurements Group, Inc.[8], the gages have a resistance of 350 Ω and a gage factor[9] of 2.155. Their position on the load cell is shown in Figure 4.10 and a photograph of the sensor interior is shown in Figure 4.11.

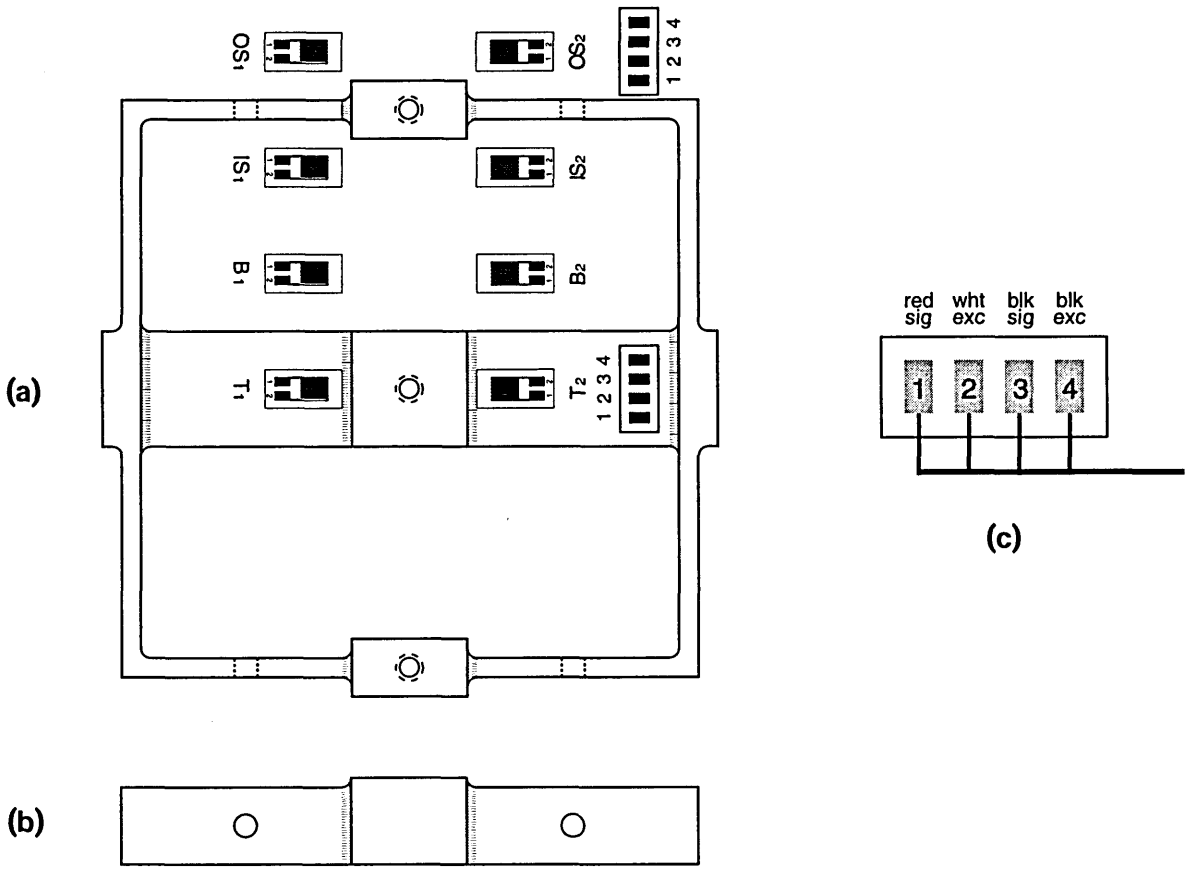


Figure 4.10: (a) A top view of a EDLS sensor load cell. Each load cell contains eight strain gages. Four are placed on the center beam (T1, T2, B1, B2) and make up the first full bridge. Another four strain gages are placed on the side beam (IS1, IS2, OS1, OS2) and make up the second full bridge. **(b)** A side view of the load cell. **(c)** Two wires carry the excitation voltage (#2, #4) for the strain gages and two wires carry the signal (#1, #3).

4.3.2 Experiment Support Module

As was explained earlier, two ESM's were used for EDLS on Mir. The second ESM, borrowed from the Mir Structural Dynamics Experiment (MiSDE), was the latest ESM type built by Payload Systems.

8. Micro-Measurements Division of the Measurements Group, Inc., Rayleigh, North Carolina.
9. The gage factor (GF) is the ratio of the relative resistance change in a strain gage to the unit strain causing the resistance change. A unit strain is defined as the ratio of the change in length to its initial length.

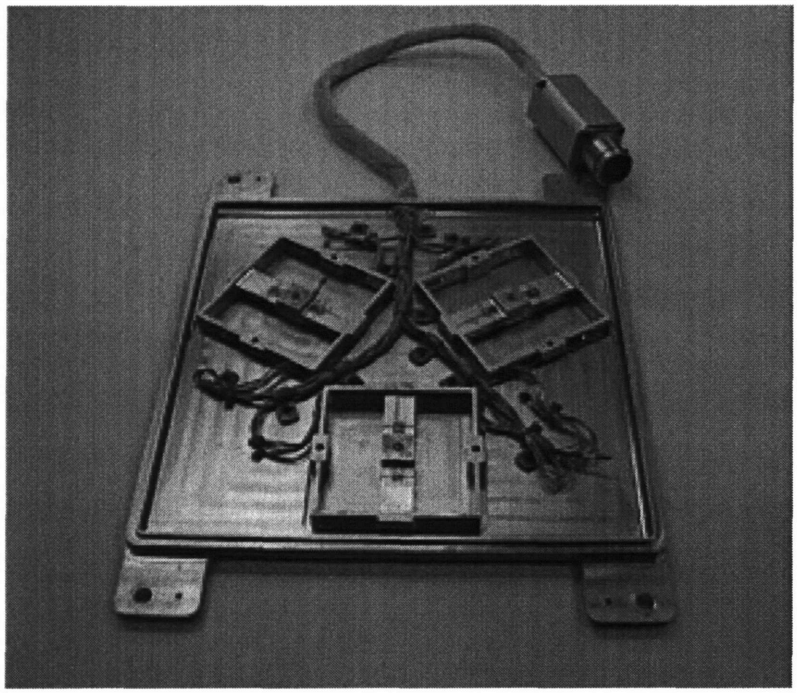


Figure 4.11: This photograph shows a load sensor with the top plate removed. Clearly visible are the three loadcells. All four sensors have the exact same interior arrangement.

It is a more advanced version of the ESM's used for MODE and the Middeck Active Control Experiment (MACE), which was an MIT spaceflight experiment that flew on the Space Shuttle in March of 1995. This section discusses first the MODE ESM and then the MASU ESM by pointing out the differences between the two models.

MODE ESM

The Experiment Support Module (ESM) developed for MODE houses all the electronics necessary for the DLS and EDLS experiment. While developed for a specific experiment, the ESM was designed to be generic enough to support a variety of Space Shuttle experiments. Very few modifications were necessary to adapt the ESM for EDLS on Mir. The description of the ESM provided herein applies to the configuration flown on Mir. Figure 4.12 shows the ESM ready for its flight on the Russian orbital complex.

The ESM consists of four key components: (1) Experiment computer, (2) signal conditioning system, (3) an optical disk drive, (4) a power supply. In addition, there is a cooling fan, circuit breakers, thermostats, air flow sensors, and associated cabling. All ESM elements are contained within a rigid alu-

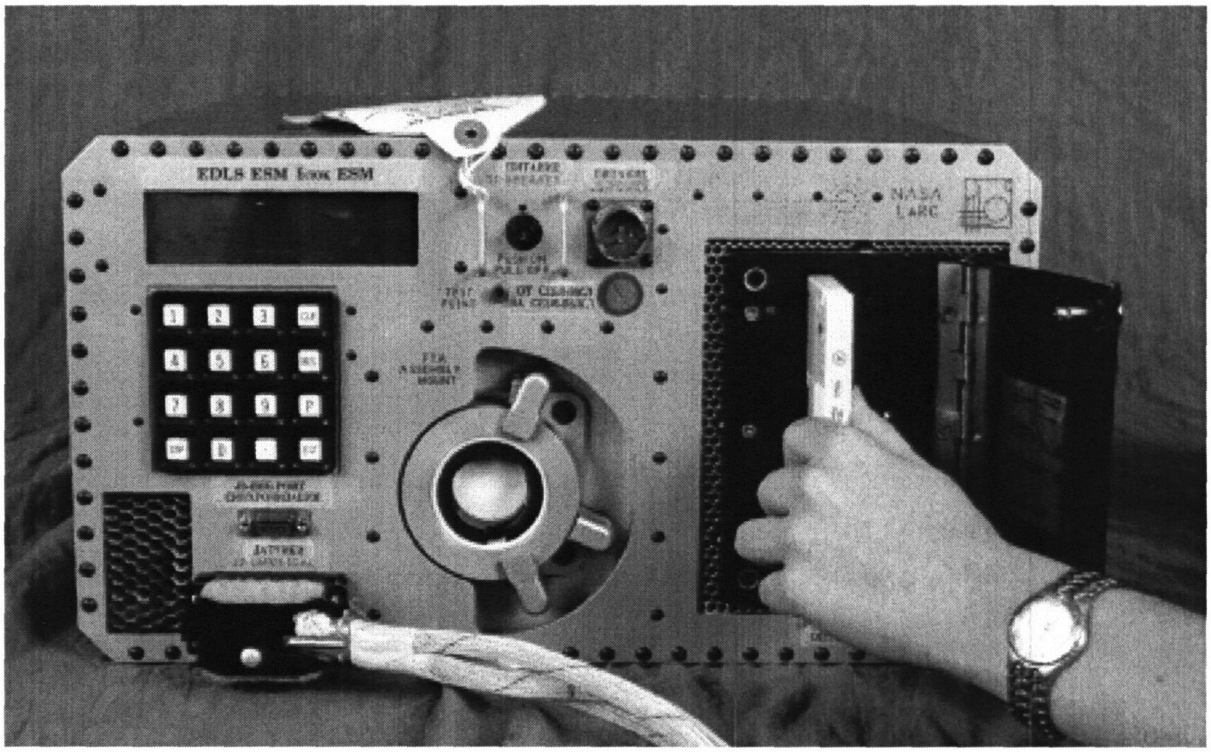


Figure 4.12: The figure shows the Experiment Support Module (ESM) used for the DLS and the EDLS experiment with a WORM optical disk being inserted in the drive. (NASA JSC S95 17024)

minum case that provides the containment of EMI emissions, and air distribution via internal air channels [70].

The ESM offers the following key features:

- DC to DC conversion of the supplied +28 V of power
- 20 channels of sensor powering and preamplification
- 16 channels of digitally controlled amplification and filtering (i.e., signal conditioning)
- 12-bit analog-to-digital conversion with an input range of ±10 Volts
- Automatic execution of experiment protocols stored in software
- Active cooling of electronics
- Operator interface through a 16-button keypad and a 2×16 characters alphanumeric display of experiment status, error messages, etc.

The four modules of the Experiment Computer are: (1) the Experiment Control Module, for digital input and output, (2) the analog Input Module for A/D conversion, (3) the Mass Storage Module for

data transfer and command to the disk drive, and (4) the Central Exchange for inter-processor communications management. The Central Exchange uses a V53 processor with an Intel 80186 instruction set operating at 16 MHz with 1 MByte of dedicated RAM. All the software was stored on EPROMs.

Four Signal Conditioning Cards were used to provide 16 channels of data recording. Each channel has software-selectable gain, followed by an 8th-order lowpass Bessel filter to avoid aliasing of the data [70]. During DLS and EDLS, a corner frequency of 125 Hz was used [27].

The ESM included a commercial Write-Once-Read-Many (WORM) optical disk drive manufactured by Mountain Optech, Inc. The drive uses 5.25" disk cartridges capable of storing 200 MBytes per side, giving a total storage capacity of 400 MBytes per disk. The disks are replacable on orbit and have a guaranteed data integrity, since once data is written it cannot be erased. The drive uses a SCSI bus to communicate with the Experiment Computer portion of the ESM.

The specifications of the MODE ESM are summarized in Table 4.5.

Table 4.5: Specification of the MODE / EDLS ESM [70], [71]

Width	49.10 cm
Height	22.80 cm
Depth	41.50 cm
Mass	26.59 kg
Power Consumption	105 W
Noise Produced (measured 1 m in front of the front panel)	20.7 dB_A

Failure of MODE ESM

The MODE ESM was returned from Mir to the ground to find the cause of the failure. Both of the exhaust air flow pressure sensors were "OFF," indicating insufficient exhaust air flow even though the fan was operating normally. It was found that one of the sensors was intermittent. Even when it functioned, it appeared less sensitive to inlet pressure than the other. The faulty sensor was compared to a spare secondary unit from the original MODE program and it appeared that the seal between the membrane and the sensor housing was not in tact around the entire circumference of the membrane due to loose rivets. The assembly tolerances for this device may not have been very tight or the rivets loosened over time with each high vibration exposure during testing, launch, etc. The sensor was a relatively inexpensive COTS item, built to "industry standards." [66]

Even though the ESM was designed to continue operating with one failed pressure sensor it did not and no explanation for that was found. Once the defective sensor was replaced, the ESM once again operated normally and was re-activated for post-flight calibration tests.

MASU ESM

The MASU ESM was part of the Mir Structural Dynamics Experiment (MiSDE). The primary purpose of the experiment was to determine the structural dynamic response of Mir and Mir with a Shuttle orbiter mated to it. The ESM and all the other hardware were brought to Mir on Shuttle Mission STS-79 in September 1996. [74] A photograph of the ESM is shown in Figure 4.13. Many of the systems and components of the MASU ESM are identical to those found in the MODE ESM. Like its predecessor, it occupies one standard middeck locker and so its overall dimensions are identical. However, a significant mass reduction was achieved. The MODE ESM had a mass of 26.6 kg, while the MASU ESM had a mass of 20.3 kg [40].

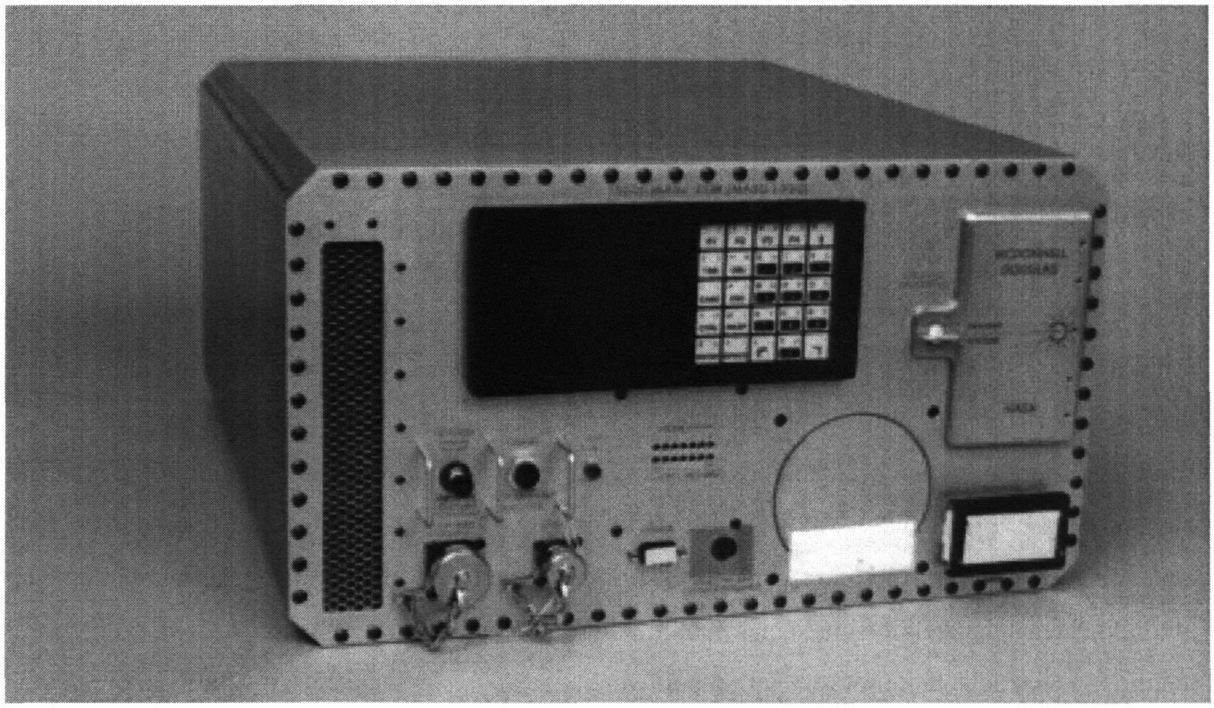


Figure 4.13: This figure shows the Mir Auxiliary Sensor Unit (MASU) ESM used for the MiSDE Experiment and for EDLS during the NASA 4 mission (NASA Image S96-07051).

The MASU ESM incorporates several noteworthy technical advancements [74]:

- CPU and Software

In essence, the MASU ESM is a modified Personal Computer. It uses a 33 MHz processor made by Ziatech that mimicks an Intel 486 CPU. The ESM software is written as a DOS application.

- Signal Conditioning

  While the MODE ESM provided only 16 A/D channels, the MASU ESM offeered 32 channels for the collection of time history data.[10] When the ESM was used to record both EDLS and MASU data, fifteen channels were used for EDLS data and seventeen channels for MiSDE data. A breakdown of the channel usage is provided in Table 5.4 on page 93.

- Data Storage Technology

  Instead of WORM disks, Type III PCMCIA cards[11] are used for data storage. PCMCIA or PC cards are "credit card-size" peripherals that add memory, mass storage, and I/O capabilities to computers in a compact form factor. The MASU ESM makes use of 260 MB hard disk PC Cards which measure only 85.6 by 54.0 by 10.5 millimeters and weigh 77 grams. Since the cards are solid state electronic devices, there are no moving mechanical parts. The ESM included *two* built-in PC card readers, which interface with the CPU through an IDE interface.

- Display and Keypad

  Much of the connectors on the front are identical or similar to those in older ESMs. However, the display and keypad unit are slighlty improved. The Termiflex Model CP/2501 keypad/display unit provides 25 sealed membrane keys and a 4 × 20 character supertwist backlit LCD.

4.3.3 Umbilical Cable / Adapter

The EDLS sensors do not connect directly to the ESM. Rather, a 25-feet (7.6 m) long umbilical plugs into the front port and splits into three cables with connectors at the ends to which the load sensors are attached. Therefore, no more than three sensors can be connected to the ESM at a time. The plugs on the umbilical cable include terminal caps to be put on unused connectors. As was mentioned already and as can be inferred from the technical description of the ESM, the number of sensors is limited by the number of channels of the signal conditioning hardware. Four sensors, each with 6 dof, would require 24 channels, three sensors with 6 dof, would need only 18 channels but still two more than the

10. The 32 channels were used originally as follows: 4 *triaxial* accelerometers with temperature recording = 24 channels, 4 *uniaxial* accelerometers with temperature recording = 8 channels.

11. The name "PCMCIA cards," is commonly used but the industry prefers to call products based on the technology as "PC Cards," and refers to the association as PCMCIA (Personal Computer Memory Card International Association), which is a non-profit trade association and standards body that promotes the PC Card technology by defining technical standards and educating the market.

16 available. As a result, only three sensors can be connected and one data from one of the three sensors would include force signals only (i.e., 6 + 6 + 3 = 15 channels).

To greatly simplify post-processing, each sensor would use only a specific group of channels and consequently only a specific plug on the umbilical. The two possible arrangements were:

Connector 1	Connector 2	Connector 3	Configuration
Foot restraint 1	Hand hold	Touchpad	FHT
Foot restraint 1	Foot restraint 2	Touchpad	FFT

In order to assure that each sensor is connected correctly, the pins on the plugs of the umbilical are such that only the correct sensor can be attached. In addition, each plug is color-coded on the umbilical and on the sensor.

Since the connector on the front of the MASU ESM was incompatible with the MODE/EDLS umbilical, an adapter was needed. The adapter was brought to Mir on STS-81 and allowed the MASU ESM to record data from the EDLS sensors.

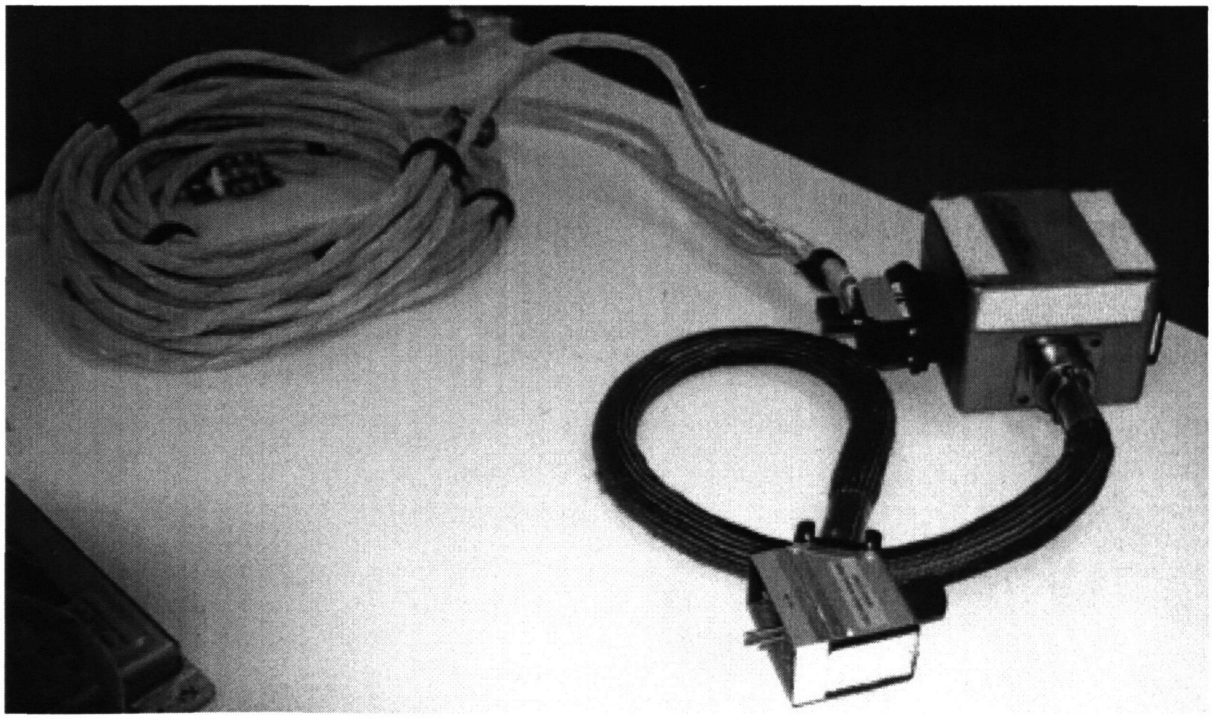


Figure 4.14: This photograph shows the white EDLS umbilical (on the left) through which the sensors are connected to the ESM as well as the MASU / EDLS adapter (on the right).

The MASU ESM was equipped with 31 channels and the EDLS signals were assigned to channels as to minimize the impact on the MiSDE experiment and are thus not in a logical order. Table 4.6 summarizes the "interface" between the sensors and the ESM. The table lists the color coding, possible ESM configurations, and the channel assignment for both the MODE and the MASU ESM.

Table 4.6: EDLS Sensor and Their Channel Arrangement

Nr.	Name	Color Coding of Connector	Possible Configurations with ESM	MODE ESM Channels[a, b]	MASU ESM Channels[a]
1	Hand hold	Light Blue	FHT	8–13	2, 3, 7, 25, 29, 30
2	Foot restraint 1	Turquoise	FHT or FFT	1–4 and 6	0, 1, 16, 17, 18, 19
3	Touchpad	White	FHT or FFT	14–16	31, 14, 15
4	Foot restraint 2	Light Blue	FFT	8–13	2, 3, 7, 25, 29, 30

a. The channel numbers are listed in the following order: V1 through V6 for the hand hold and the foot restraints and V1 through V3 in case of the touchpad.

b. Channel 5 was unused.

CHAPTER

5

PROCESSING AND ANALYSIS OF EDLS EXPERIMENTAL DATA

The previous chapter introduced the EDLS experiment and its technology. The development of the EDLS hardware as well as much of the payload support was carried out by the main subcontractor of the spaceflight experiment. The processing and analysis of the data gathered on Mir was done at MIT and is the topic of this chapter. Due to the failure of the original Experiment Support Module and the replacement with a different model on-orbit, the processing of the raw spaceflight data became a much more challenging task than was originally expected. Instead of a single data format, a known performance quality of the ESM with the sensors, and a preflight calibration, there were two (very different) data formats, no calibration data, and an unknown data quality.

5.1 Terminology and Notation

Some of the computer programming terminology in this chapter may be unfamiliar to the reader and hence a brief review is provided.

In this thesis a preceding dollar sign is used to indicate a hexadecimal number and a preceding percent sign a binary number. Computer variables or filenames are identified through the use of the fixed-width Courier font. The common units of "length" for binary information are the *bit*, *byte*, *word*, and *longword* (or *double word*). A byte is 8 bits long, a word consists of 2 bytes, and a longword consists of 4 bytes. When integer numbers are stored in binary form, the bytes, words, and longwords can be either signed or unsigned. For example, in a *signed byte*, the highest bit (i.e., the eight) is used to indicate whether the number is positive or not. When the bit is set, the number is positive and negative when the bit is not set. As a result, a signed byte can represent values from –127 to +127, while an unsigned bytes can stand for values from 0 to 255. Similarly, as signed word, can represent values from –32767 to +32767 and an unsigned word from 0 to 65535.

Modern microprocessors have either a 16-bit, 32-bit, or 64-bit data bus. The definition of a word and longword length is regardless of the processor. However, how bytes are interpreted as words and longwords varies by processor. In the so-called *little-endian* byte ordering, the lowest-order (rightmost) eight bits of the number are read in first. For example, an Intel processor reads $F3 and $FF as $FFF3. A Motorola CPU typically works with the *big-endian* byte ordering scheme in which the highest order (leftmost) eight bits of the number are read in first. Thus, the processor reads $F3 followed by $FF as $F3FF. In little-endian byte ordering, the signed word $FFF3 is interpreted as –13, while in big-endian byte ordering, $F3FF is understood as –3073. This difference between processors had to be accounted for to process EDLS data since MATLAB does not shield the user from this difference among processors.

In binary data, the term *offset* is used to indicate a position within the file. An offset of 0 means the very first byte in the file, an offset of 1 means the second byte, and so on.

An analog-to-digital (A/D) converter transforms an analog signal, voltage, into numbers. The MODE/EDLS ESM features 13-bit A/D board, giving it a range of $2^{13} = 8192$ *counts*. The MASU ESM had a 16-bit A/D board and thus had a range of $2^{16} = 65536$ counts. For the purpose of the discussion in this thesis, the term *data sample*, stands for the data collected at a point in time on all channels. The data recorded by the MODE/EDLS as well as the MASU ESM is referred to as *raw data*. Once converted into Matlab format, it is referred to as *processed data*.

5.2 EDLS Spaceflight Experiment Data

5.2.1 Data from NASA 2 Mission

As mentioned in Chapter 4, Shannon Lucid carried out about 20 EDLS sessions during the NASA 2 mission on Mir. Lucid and eight WORM disks with EDLS data were brought to Earth on the Space Shuttle Atlantis in Flight STS-79 in September 1996. When the original ESM was returned to the subcontractor PSI several months later, one WORM disk (Number 9) with one side full of data was found in the WORM drive of the ESM. Examination of the EDLS disks revealed that more than one thousand files totalling 2.46 Gbytes were recorded during NASA 2. Table 5.1 provides a break down of the EDLS data from the NASA 2 mission by disk.

Table 5.1: EDLS Data Returned from NASA 2 Mission

WORM Disk	Side	No. of Files	Size [MBytes]	Date(s) Recorded[a]	Config.[b]	Operation
1	A	416	186.400	24 May 1996; 4, 27, 28, 29 June 1996		Passive
1	B	98	20.365	1, 2, 3, 5, 8, July 1996		Passive
2	A	50	190.317	9 July 1996		Passive
2	B	141	190.211	10, 11, 12 July 1996	FHT	Passive
3	A	2	15.631	14, 16 July 1996	FFT	Active
3	B	61	190.118	18, 24, 30 July 1996	FFT	Passive
4	A	2	190.340	31 July 1996		Passive
4	B	8	3.806	1 August 1996		Passive
5	A	81	190.296	6, 7 August 1996		Passive
5	B	11	190.336	8, 9, 10 August 1996		Passive
6	A	58	190.313	9, 10 August 1996	FFT	Passive
6	B	5	190.339	12 August 1996		Passive
7	A	40	8.900	17 August 1996		Passive
7	B	3	190.337	23 August 1996		Passive
8	A	2	190.338	24 August 1996		Passive
8	B	2	190.276	29 August 1996		Passive
9	A	28	190.325	30 August 1996		Passive
9	B	1	0.123	30 August 1996		Passive
TOTAL:		**1009**	**2518.771**			

a. These dates are estimated based on the notes written on the WORM disks, Shannon's personal log, POSA reports, and video footage.

b. The configuration is indicated only when known.

The EDLS video recordings were made on Mir with a commercial Video Hi8 camcorder and then copied onto VHS tapes at the Johnson Space Center. According to the flight log, Shannon Lucid recorded seven video tapes but only copies of six tapes were received from NASA for data analysis. Experimental notes were scribbled down by Shannon Lucid directly on the plastic enclosure of the WORM disks as well on a few sheets of paper. Unfortunately, the notes were very limited and so it was difficult to piece together how the EDLS sessions were carried out. In addition, the video footage was not very useful.

The WORM disks do not employ a standard computer file system, and thus a special program is required to read the files. The WORM Access Utility (WAU) V3.2 was used to read the data files and

transfer the data to iomega 1GByte Jaz disks for further processing. Several sectors of the WORM disks were unreadable and the data contained in that sector marked. Table A.1 on page 145 lists the bad sector blocks and the affected files.

The naming convention of the raw data files is `PP_N.DAT` where `PP` is the two digit protocol number under which the data was recorded and `N` is a variable digit counter that numbers the files consecutively starting with one. So that for example, `99_5.DAT` is the fifth data file on the disk side. Each data file has a corresponding header files, which would have the name `99_5.HDR` for the previous example.

The header file contained information on the data file. Its structure is deciphered in Table 5.2.

Table 5.2: Data File Format of the NASA 2 Header File

Offset	Entry	Example
0	Validity String terminated by a zero byte.	`Payload System`
28	The name of the corresponding data file terminated by a zero byte.	`99_10.dat`
90	The time stamp in the form DDD:HH:MM:SS terminated by a period.	`143:16:19:30.`
115	The "gain indices" of the data. Each of the 16 bytes is the gain index, which can be converted to the actual gain by the following expression: Gain = $2^{\text{Gain Index}}$.	`04 04 04 04 03 03 03 03 03 03 03 03 03 03 03 03`

Of the 16 ESM channels, 15 were used to record EDLS data. The channel assignment for the MODE/ EDLS ESM grouped forces and moments together as shown below:

Channel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Sensor	Sensor 1				unused	Sensor 1		Sensor 2						Sensor 3		
Signal	*V1*	*V2*	*V3*	*V4*		*V5*	*V6*	*V1*	*V2*	*V3*	*V4*	*V5*	*V6*	*V1*	*V2*	*V3*

The file format of the data file is very simple. The data for each of the 16 channels in every sample has a length of one word. The sampled data is written consecutively from Channel 1 to Channel 16 followed by 16 words for the next sample, and so on. Hence, the first data sample is at offset 0, the next at offset 32, the third at offset 64, and so on.

5.2.2 Data from NASA 4 Mission

Following the replacement of the original ESM with the MASU ESM, NASA 4 astronaut Jerry Linenger was able to record EDLS data. From February 25 through May 9, 1997, Linenger executed eighteen sessions including six active ones. Of the twelve PC cards sent to Mir, ten contained EDLS data totalling 2.25 GBytes. One of the major changes made from DLS to EDLS, was the implementation of event detection software as was mentioned earlier. The successful implementation of event detection, however, relied on setting an appropriate threshold that needed to be exceeded to activate recording. Because of the on-orbit change of the ESM, it was not possible to determine a good threshold level on the ground prior to the experiment. A value of +/– 40 counts was set for the standard EDLS protocol (#99). Unfortunately, this threshold turned out to be too low due to a constantly large noise level. As a result, the more than two GBytes of data was contained in only 42 files of which some were as large as the entire free space on the PC Card. Table 5.3 provides a break down of the EDLS data from the NASA 4 mission by PC Card.

Table 5.3: EDLS Data Returned from NASA 4 Mission

PC Card	No. of Files	Size [MBytes]	Date Recorded	Config.[a]	Operation
1	12	246.773	10 March 1997	FHT / FFT	Active, Passive
2	1	246.799	12 March 1997	FHT	Passive
3	2	246.748	13 March 1997	FHT	Passive
4	3	246.784	25 February 1997	FHT / FFT	Calibration, Passive
5	3	79.431	26 March 1997	FFT / FHT	Active, Passive
6	1	246.770	9 May 1997	FHT / FFT	Passive
7	15	246.744	28 February 1997	FHT	Passive
8	2	246.774	27 March 1997	FHT	Passive, Active, Passive
9	2	246.742	6 March 1997	FHT / FFT	Active, Active, Passive
10	1	246.780	27 February 1997	FHT / FFT	Active, Passive
TOTAL:	**42**	**2300.345**			

a. Whenever Jerry Linenger conducted a passive session he used an FHT configuration and an FFT configuration for all active sessions.

Jerry Linenger recorded ten videotapes of the EDLS sessions and copies of nine of them were received from the Johnson Space Center.

Unlike the WORM disks, the PC Cards are directly readable on a PC or Macintosh since they use the common file system found on PC's (i.e., Microsoft's FAT). The naming convention for the files is the

same as for files on WORM disks: `PP_N.DAT` where `PP` is the protocol number and `N` is the counter that numbers the files on the PC Card starting with one. However, there are no separate header files.

Since the MASU ESM was shared between the MiSDE and the EDLS experiment, the data files contain data from both experiments. Of the 32 channels available on the ESM, 15 were used for EDLS and 17 for MiSDE. The breakdown of the channel usage is given in Table 5.4 on page 93. For convenience, the list is sorted by both channel number and sensor. Associated with each channel was also a system gain. The gains for the EDLS-related channels are shown Table 5.8.

To accommodate the EDLS data experiment, the MASU data file format was updated to V2.0. Each file contains a headersection of ASCII text followed by the data section, which contains the raw data in binary form. The header specifies the settings under which the data was recorded. Each entry in the header is one line long (separated by a carriage return) and consists of an identifying string followed by an equal sign and then the parameter. An exception is the Time Source entry which uses a colon instead of an equal sign.[1] The header of a typical data file is shown in Figure 5.1.

The meaning of the entries is explained below:

- **Validity String:** If the string is "Payload Systems," the data file is identified as valid.
- **Protocol Number:** Identifies the protocol that was used to record the data. In general the number is an integer from 1 to 65535. The EDLS PC Cards contained sixteen protocols—numbered 81 through 89 and 91 through 99. All data recorded during Jerry Linenger's NASA 4 mission used Protocol 99.
- **Sample Rate:** The data sampling rate had a possible range of 10 Hz to 1000 Hz in 1 Hz increments. All data recorded during the NASA 4 mission had a sampling frequency of 250 Hz.
- **Number of Channels:** The number of channels in which data was recorded. The possible range was 2 to 32 in 2 channel increments. During EDLS sessions data was recorded on all 32 channels. See Table 5.4 for the channel assignment.
- **Duration:** The length of time data is to be recorded. Possible durations were 00:00:01 seconds to 23:59:59 hours of which the *latter* was the default. A manual ending of the data-taking session would not affect the number.

1. This was not intentional but a programming error.

Table 5.4: Channel Assignment of MASU ESM Used During NASA 4 Mission

Sorted by Channel			Sorted by Sensor		
Channel	Sensor	EDLS #	Channel	Sensor	EDLS #
1	EDLS Sensor 1, V1	1	1	EDLS Sensor 1, V1	1
2	EDLS Sensor 1, V2	2	2	EDLS Sensor 1, V2	2
3	EDLS Sensor 2, V1	7	17	EDLS Sensor 1, V3	3
4	EDLS Sensor 2, V2	8	18	EDLS Sensor 1, V4	4
5	MiSDE Triaxial 1-X	—	19	EDLS Sensor 1, V5	5
6	MiSDE Triaxial 1-Y	—	20	EDLS Sensor 1, V6	6
7	MiSDE Triaxial 2-Z	—	3	EDLS Sensor 2, V1	7
8	EDLS Sensor 2, V3	9	4	EDLS Sensor 2, V2	8
9	MiSDE Triaxial 3-X	—	8	EDLS Sensor 2, V3	9
10	MiSDE Triaxial 3-Y	—	26	EDLS Sensor 2, V4	10
11	MiSDE Triaxial 4-Z	—	30	EDLS Sensor 2, V5	11
12	MiSDE Temperature T4	—	31	EDLS Sensor 2, V6	12
13	MiSDE Triaxial 5-X	—	32	EDLS Sensor 3, V1	13
14	MiSDE Triaxial 5-Y	—	15	EDLS Sensor 3, V2	14
15	EDLS Sensor 3, V2	14	16	EDLS Sensor 3, V3	15
16	EDLS Sensor 3, V3	15	5	MiSDE Triaxial 1-X	—
17	EDLS Sensor 1, V3	3	6	MiSDE Triaxial 1-Y	—
18	EDLS Sensor 1, V4	4	21	MiSDE Triaxial 1-Z	—
19	EDLS Sensor 1, V5	5	23	MiSDE Triaxial 2-X	—
20	EDLS Sensor 1, V6	6	24	MiSDE Triaxial 2-Y	—
21	MiSDE Triaxial 1-Z	—	7	MiSDE Triaxial 2-Z	—
22	MiSDE Temperature T1	—	9	MiSDE Triaxial 3-X	—
23	MiSDE Triaxial 2-X	—	10	MiSDE Triaxial 3-Y	—
24	MiSDE Triaxial 2-Y	—	25	MiSDE Triaxial 3-Z	—
25	MiSDE Triaxial 3-Z	—	27	MiSDE Triaxial 4-X	—
26	EDLS Sensor 2, V4	10	28	MiSDE Triaxial 4-Y	—
27	MiSDE Triaxial 4-X	—	11	MiSDE Triaxial 4-Z	—
28	MiSDE Triaxial 4-Y	—	13	MiSDE Triaxial 5-X	—
29	MiSDE Triaxial 5-Z	—	14	MiSDE Triaxial 5-Y	—
30	EDLS Sensor 2, V5	11	29	MiSDE Triaxial 5-Z	—
31	EDLS Sensor 2, V6	12	22	MiSDE Temperature T1	—
32	EDLS Sensor 3, V1	13	12	MiSDE Temperature T4	—

```
Validity String=Payload Systems
Protocol Number=00099
Sample Rate=0250
Number of Channels=32
Duration=23:59:59
Delay=00:00:00
Gains=10,10,10,10,10,10,10,10,10,10,01,10,10,10,10,10,10,10,10,01,10,10,10,10
,10,10,10,10,10,10
Filter Rolloff=075
Buffer Size=65024
PlusThreshold=00040
NegThreshold=-0040
Time Src: IRIGB DC
Time Stamp=072:10:15:22.721577
Time Offset=00030.6800
Offsets=001914,002088,002195,002192,002449,002948,001721,002149,001784,002161,00216
3,000000,000881,002033,002203,001996,002248,002167,001924,001892,000645,000000,0019
68,001794,001104,002400,002268,001199,001860,002121,002640,002276
AdOffsets=-00006,-00001,-00004,000000,-00090,000024,-00083,-00002,-00045,-
00018,000305,027970,000004,-00044,-00002,-00001,-00002,-00002,-
00006,000000,00025,024153,-00029,-00009,000004,-00002,-00018,-00049,000045,-00005,-
00001,000001
```

Figure 5.1: The figure shows the header of file 99_1.DAT from PC Card #3. It is an example of the text at the beginning of a typical raw EDLS data file from the MASU ESM. Note that the line wraps are artifacts of the line width available in the figure.

- **Delay**: The length of time data-taking was to be delayed after starting to run the protocol. Possible delays were 00:00:00 seconds to 23:59:59 hours of which the *former* was the default.
- **Gains**: The gains for each channel starting with the first channel.
- **Filter Rolloff**: The filter roll-off frequency. The range was 5 to 555 Hz.
- **Buffer Size**: The buffer size is the amount of data, including a buffer tag, written at a time to the data file. See further explanation below.
- **Positive Threshold**: The minimum amount of counts above 0 that data has to have to be recorded on the disk.
- **Negative Threshold**: The minimum amount of counts below 0 that data has to have to be recorded on the disk.
- **Time Source**: The time source indicates the clock that was used to synchronize with during data taken. The options are: IRIGB DC, RTC, and CPU.
- **Time Stamp**: The time stamp records the day of the year, hours, minutes, seconds, and microseconds at which the first data was recorded.
- **Time Offset**: The offset in seconds from the time stamp to give the actual time the event started.

- **Offsets:** The offsets from zero in counts for the channels of the D/A converter after auto-zeroing.
- **AdOffsets:** The offsets from zero in counts of the A/D channels after auto-zeroing.

The length of the header section is 1024 bytes. The end of the text is marked by ASCII code 26 and the rest is filled with zero bytes.

The actual data is stored in blocks written in binary form. The length of each block is determined by the buffer size, which is provided in the header. Each block begins with a buffer tag and continues with data samples of all channels. The length of the buffer tag and each data sample is twice as many bytes as channels are recorded. The normal case of 32 channels and a buffer size of 65024 bytes is illustrated in Figure 5.2. The first four bytes (a longword) of the buffer tag are the serial number of the buffer. However, this is only the case if the buffer is completely filled with data. If it is not, then the longword is a negative number whose absolute value indicates the number of data words in that buffer. For example, if the buffer tag is the hexadecimal number $80A8FFFF, it is read by Intel computer as $FFFFA880, which is in decimal notation –22400 if the highest bit is the sign bit. Thus, there are $(22400 \times 2) / 64 = 700$ data samples to read.

5.3 EDLSAP Software

The raw spaceflight data written by the two ESM's onto storage media is not in a format that is easily read by standard software nor useful for analysis and hence needs to be processed first. The analysis of the processed data is carried out at MIT but NASA requires also the unaltered data for their own examination. To simplify and automate the work MIT as well as to aid NASA in its future needs to analyze the data, EDLS data processing and analysis software was developed.

5.3.1 EDLSAP Requirements and Features

The software was named Enhanced Dynamic Load Sensors Analysis Package is referred to by its acronym EDLSAP. The requirements for the software were determined to be as follows:

- EDLSAP shall be able to convert raw EDLS spaceflight data from the MODE/EDLS ESM and the MASU ESM into Matlab data files.
- EDLSAP shall be able to edit (i.e., cut, scale, unbias, etc.) processed EDLS files.

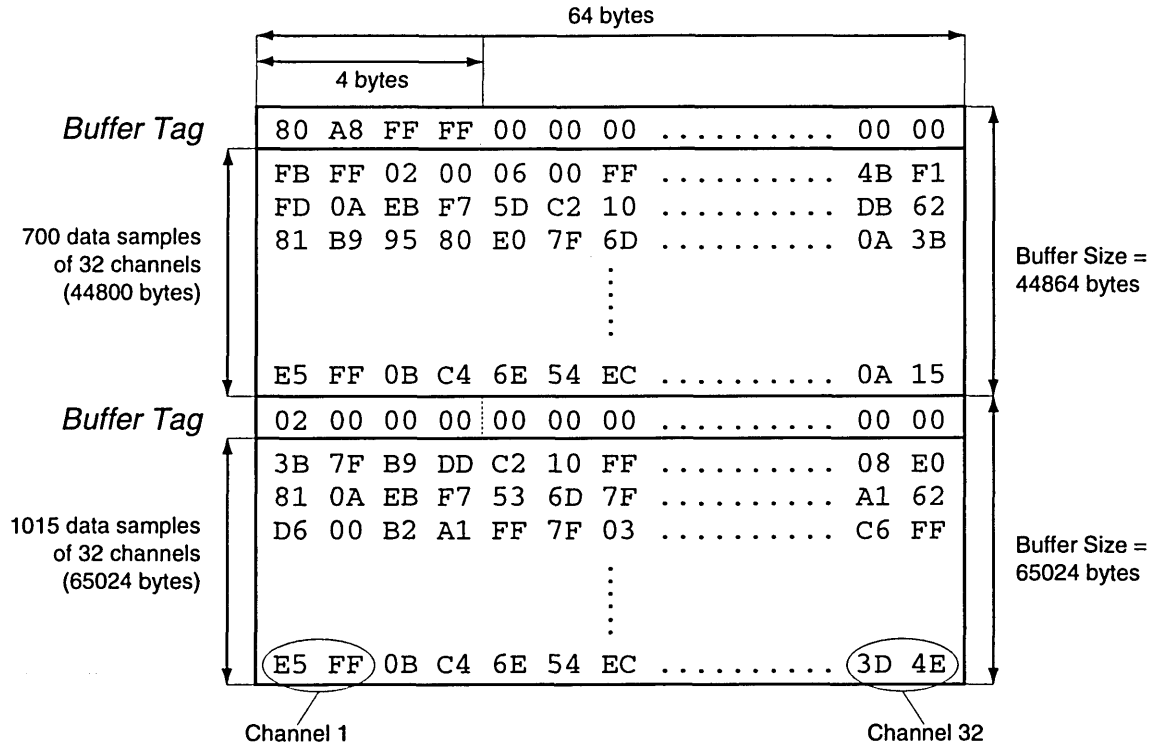


Figure 5.2: The figure shows the data section of a raw EDLS data file created by the MASU ESM. The first shown block is partially filled and the buffer tag indicates the size of the data (44800 bytes). The second block has the serial number 2 and is a full block with 1015 data samples, which corresponds to 4.06 seconds of data at a sampling rate of 250 Hz.

- EDLSAP shall be a cross-platform (i.e., run under Mac OS, Windows 95/NT, and various versions of Unix) and require, if any, software readily available at MIT, Payload Systems, NASA, and Boeing.
- EDLSAP shall make use of a graphical user interface for ease of use.
- EDLSAP shall have the capability to load and display SAMS data from Mir for correlation with the EDLS data.
- EDLSAP shall be easy to update, change, and expand.
- There shall be no legal restrictions on the distribution of the EDLSAP source code or the executable program.

Based on these requirements, the MATLAB® technical computing environment was chosen as the development tool. The current release of MATLAB 5.2 is available for all three common computer platforms and is in use in all organizations involved in EDLS. Since the code does not be compiled,

programs can be easily updated, changed, and expanded with the appropriate documentation of the source code. EDLSAP is a collection of 42 individual MATLAB scripts and encompasses about 5000 lines of code. Appendix C contains a print out of the most important scripts as well as a table with the names of all routines and the function calls of those scripts that arre functions.

The main user interface window of EDLSAP is shown in Figure 5.3. An extensive graphical user interface was implemented to simplify use of the program at MIT, NASA, and Boeing. While the MATLAB environment greatly reduced the complexity involved in writing a cross-platform application in comparison to C or C++, several differences exist between platforms. EDLSAP accounts such differences as different file path conventions from operating system to another. In the first section of the chapter it was mentioned that Intel and Motorola processor interpret two bytes into words differently. When EDLSAP processes raw data under the Mac OS, it swaps the bytes since the raw data was written under the little-endian byte ordering scheme.

As mentioned earlier, the two main tasks of EDLSAP are the processing of raw data and the analysis of processed data. The application's processing capabilities are discussed first.

5.3.2 Processing of Raw EDLS Data

EDLSAP converts raw data files generated by both the MODE/EDLS ESM and the MASU ESM into MATLAB data files. It provides the capability to convert filea manually (i.e., one file at a time) or automatically by reading in a "batch file" and converting a number of files at a time. Figure 5.4 shows the EDLSAP window that prompts the user to enter parameters to process the raw data. Batch files are text files that follow the EDLSAP specifications, which are provided in Table A.2 on page 146.

The characteristics of the data, such as sampling frequency, as well as the actual data are extracted from the raw data files and stored in global MATLAB variables and then saved as regular (binary) MATLAB Version 5 data files[2]. Following good programming practice, the global variables use uppercase letter for differentiation from regular variables. The complete list of variables in a processed EDLS data file is shown in Table 5.5.

To simplify working with the processed data, the time the data was recorded is used a the filename. Consequently, the filename follows the following structure `DDD_HH_MM_SS.mat`, where `DDD` stands

2. By convention, MATLAB data files have the extension "mat" irregardless whether they are binary or text files.

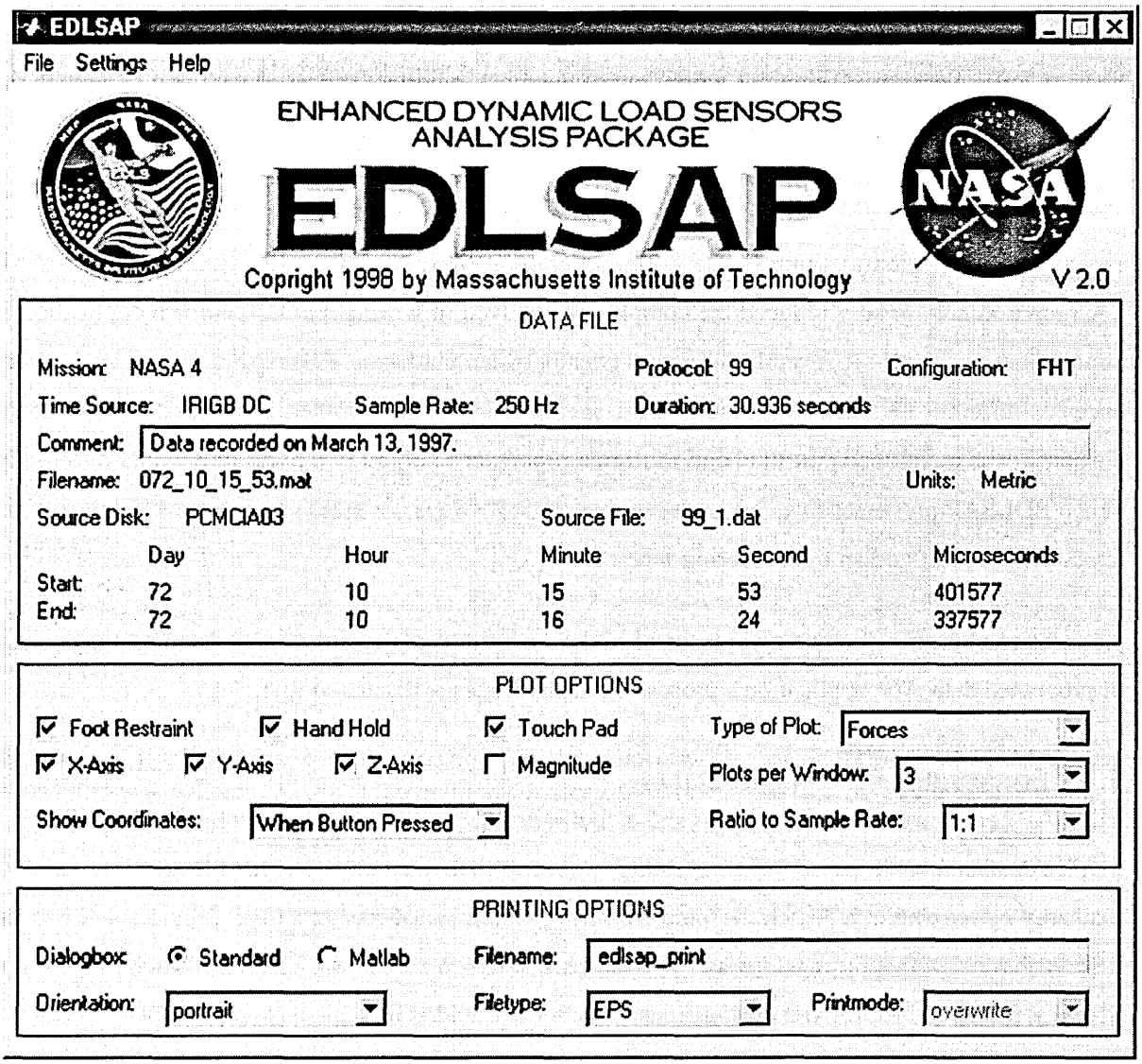


Figure 5.3: The figure shows the main window of the Enhanced Dynamic Load Sensors Analysis Package (EDLSAP). EDLSAP is a collection of Matlab 5 routines than were designed to run without changes under Windows, Mac OS, and Unix.

for the day of the year, and `HH`, `MM`, `SS` for the hour, minute, and second, respectively.[3] As was mentioned earlier, the threshold set for the EDLS data recording was set too low and so, some of the raw data file are more than 200 Mbytes long. When processing, the raw data files are cut into smaller data files, whose length (from 10 seconds to 60 minutes) can be specified. The time stamp and the filename are adjusted for the splitting of the file accordingly.

3. Some data files lack a time stamp and for those the raw data filename without the extension "dat" is added in front of the time in the processed data filename (e.g., `99_04_000_16_23.mat`).

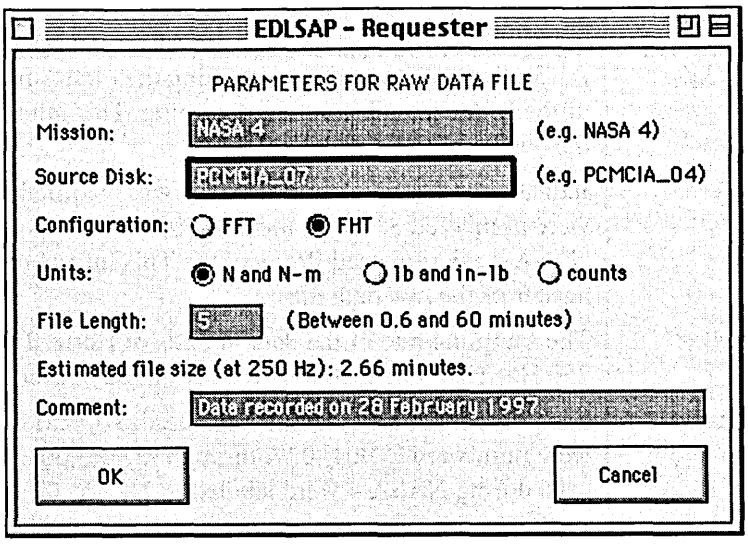


Figure 5.4: The EDLSAP application allows the user to enter several parameters on how the raw EDLS data should be processed. The key parameters are the configuration (either FFT or FHT), the desired units, and the file length. The file length is specified by the number of minutes of sampling it should include. The "Mission," "Source Disk," and "Comment" entries are optional. configuration, the units, and the file length

Table 5.5: Variables Contained in EDLS Data Files

Variable	Description
`EDLS_AdOffsets`	A 15-element row vector containing the offsets from zero, in counts, of the A/D Channels after auto-zeroing. This information is read from the header of the raw data file.
`EDLS_Configuration`	This variable indicates whether the sensor configuration was: `FHT` or `FFT`. This information has to be provided by the user.
`EDLS_Comment`	Contains a comment on the data added by the user during the processing of the raw data.
`EDLS_Data`	The data is stored in a matrix with 15 columns and n rows, where each column corresponds to one channel and n is the number of data points per channel.
`EDLS_Filename`	The name of the data file indicates when the data taking was started. The format of the name is `DDD_HH_MM_SS.mat`, where `DDD` stands for the mission day in 3-digit form, `HH` for the hour, `MM` for the minutes, and `SS` for seconds. The extension `.mat` indicates that the data is stored as a Matlab data file.
`EDLS_Mission`	Indicates during which mission or situation data was recorded; e.g. `EDLS_Mission` = NASA 4. This information has to be provided by the user.

Table 5.5: Variables Contained in EDLS Data Files *(continued)*

Variable	Description
`EDLS_Offsets`	A 15-element row vector containing the offsets from zero, in counts, of the D/A channels after auto-zeroing. This information is read from the header of the raw data file.
`EDLS_Protocol`	Indicates under which protocol data was recorded. Available protocols were numbered 81 to 89 and 91 to 99. All data recorded during the NASA 4 mission used Protocol 99. This information is read from the header of the raw data file
`EDLS_SampleRate`	The sampling rate of the data in units of Hertz. The typical value is 250 Hz. This information is read from the header of the raw data file.
`EDLS_SourceDisk`	Each disk for storing data is numbered. WORM disks on NASA 2 were numbered `WORM_01A` through `WORM_09B` and PCMCIA disks used during NASA 4 were labeled `PCMCIA_01` through `PCMCIA_11`. This information has to be provided by the user.
`EDLS_SourceFile`	The name of the raw data file which contained the data (e.g. `99_15.dat`).
`EDLS_TimeSource`	The possible sources for the time are: `IRIGB DC`, `RTC`, and `CPU`. Which system supplied the time is read from the header file.
`EDLS_TimeStamp`	This 10-element row vector contains the mission day, hour, minute, second, and microsecond at which the data collection began and mission day, hour, minute, second, and microsecond when it ended.
`EDLS_Units`	The units of the forces and torques in the matrix EDLS_Data are either Newtons and Newton-Meters (`EDLS_Units = 1`), pounds and inch-pounds (`EDLS_Units = 2`), or in counts (`EDLS_Units = 3`). The units are selected by the user during processing.

5.3.3 Analysis of EDLS Data

The application makes it easy to edit the processed data. Data segments, such as noise, can be removed from the data by *cutting* unwanted portions, the signal can be *unbiased*, so that when no loads are applied the average signal is at zero. The data can also be *scaled* (i.e., multiplied with an arbitrary multiplication factor) and *filtered* with a low-pass digital elliptic filter (with a variable number of poles and corner frequency). Once a specific event has been located, EDLSAP can visualize the force vector through a 3-D animation.

EDLSAP incorporates several signal processing functions to analyze the force and moment data. Because the application is required to run on MATLAB installations without the Signal Processing Toolbox, EDLSAP's routines make no use of these specialized functions from that toolbox. Instead, the signal processing functions are implemented from scratch using basic MATLAB commands.

Before presenting, the implementation in EDLSAP, some important signal processing concepts and the underlying equations are presented. Throughout this section the word *signal* implies either force or moment traces over time. If the signal is continuous, it is denoted by $x(t)$. If the signal is discrete, it is denoted by $x[n]$, where n is the index.

An important average of a signal is the *average power* in the signal. In the continuous case, the average power in the signal between time 0 and T is given by

$$\langle x^2(t)\rangle = \frac{1}{T}\int_0^T x^2(t)dt \tag{5.1}$$

As implemented in EDLSAP, the average power for a discrete signal $x[n]$ consisting of N samples is calculates as follows:

$$\langle x^2[n]\rangle = \frac{1}{N}\sum_{n=0}^{N} x^2[n] \tag{5.2}$$

The square root of the average power, is the *root-mean-square* (or *rms*) value of the signal and is a useful measure of the amplitude of the signal.

$$x_{\text{rms}} = \sqrt{\langle x^2(t)\rangle}\text{, or } x_{\text{rms}} = \sqrt{\langle x^2[n]\rangle} \tag{5.3}$$

If one calculates the average of $x(t)x(t+\tau)$, where $x(t+\tau)$ is the same signal just shifted in time, one obtains the *autocorrelation function*. This function of a signal measures the rate at which the signal $x(t)$ changes and is defined as

$$R_x(\tau) \equiv \langle x(t)x(t+\tau)\rangle = \frac{1}{T}\int_0^T x(t)x(t+\tau)dt \tag{5.4}$$

In other words, $R_x(\tau)$ measures the extent to which $x(t)$ and the same signal shifted in time, $x(t-\tau)$, are the same. In the discrete case, Eqn. (5.4), becomes

$$R_x[m] = \frac{1}{N}\sum_{n=0}^{N-1-m} x[n]x[m+n] \qquad 0 \le m \le N-1 \tag{5.5}$$

If one takes the Fourier transform of the autocorrelation function, one obtains the power spectral density (PSD):

$$S_x(f) = \int_{-\infty}^{\infty} R_x(\tau) e^{-j2\pi f\tau} d\tau \tag{5.6}$$

The PSD measures the distribution of power in $x(t)$ as a function of frequency f. This can be seen when one integrates $S_x(f)$ over all frequencies f:

$$\langle x^2(t) \rangle = R_x(0) = \int_{-\infty}^{\infty} S_x(f) df \tag{5.7}$$

Because of this relation, a more suitable but less common, name for PSD is *power density spectrum*.

In the discrete case, the power spectral density is the discrete Fourier Transform (DFT) of the autocorrelation function. MATLAB's basic functions include a DFT which uses a fast Fourier transform (FFT) algorithm whenever the number of samples is a power of two.

$$S_x[f] = \sum_{n=0}^{N} R[m] e^{\frac{-j2\pi fm}{N}} \qquad 0 \le f \le N \tag{5.8}$$

In the discrete case, the frequency f ranges from 0 Hz to the half of the sampling frequency, which is the Nyquist frequency (see footnote 7 on page 76).

For practical applications, there are two techniques to compute the PSD, the *Correlation Method* and *Welch's Method of Modified Periodograms* as described in [75]. EDLSAP uses latter method to estimate the power spectral density. The signal $x[n]$ is decomposed into subsequences $x_r[n]$ of length L samples such that r rankes from 1 to K. For each of the subsequences $x_r[n]$ the windowed FFT $X_r[k]$ is computed as

$$X_r[k] = \sum_{n=0}^{L-1} x_r[n] w[n] e^{-\frac{j2\pi nk}{L}}. \tag{5.9}$$

The function $w[n]$ is an appropriate window function—used in EDLSAP is a simple boxcar window function.

The periodogram $I_r[f_k]$ is computed with the following expression

$$I_r[f_k] = \frac{1}{U}|X_r[k]|^2 \tag{5.10}$$

where $f_k = (k / L)$ is the discrete Fourier transform frequency and the quantity U is the energy in the window calculated as follows

$$U = \sum_{n=0}^{L-1} (w[n])^2. \tag{5.11}$$

The PSD estimate is then the weighted sum of the periodograms of each of the individual subsequences.

$$S_x[f_k] = \frac{1}{K}\sum_{r=0}^{K} I_r[f_k] \tag{5.12}$$

With the PSD distribution known, it is possible to estimate the peak PSD value and the frequency below which a certain percentage of the power is contained. The default value is 95% (two standard deviations) of the power. EDLSAP uses the trapezoidal rule to estimate the area underneath the PSD curve and a second order interpolation to obtain the frequency below which the specified percentage power lies. There is no need to filter the data prior to the PSD computation. Since the PSD is in the frequency domain, a corner frequency above which the power density is ignored can be specified. Figure 5.5 shows the EDLSP window in which the PSD calculation parameters can be entered.

5.4 Postflight Calibration of EDLS Hardware

After the sensors and the MASU ESM were returned to subcontractor PSI, it became possible to examine the condition of the sensors after 21 months on Mir as well as perform a postflight calibration using for the first time the replacement ESM.

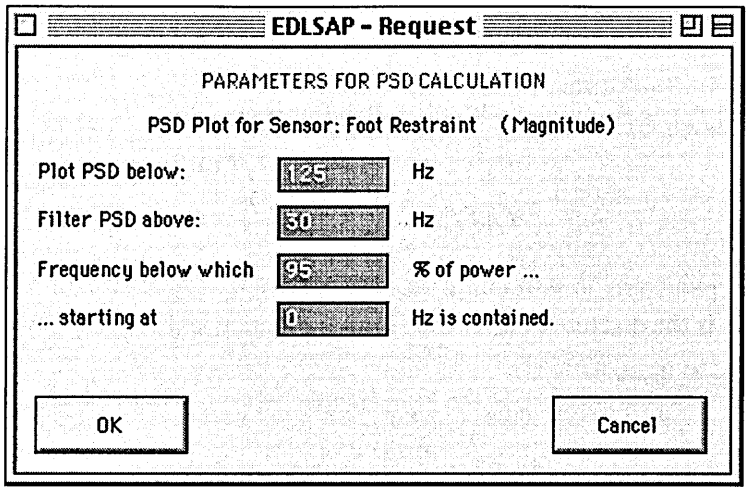


Figure 5.5: EDLSAP provides considerable flexibility in the PSD computation. The upper limit of the plot range can be specified (default is the Nyquist frequency), the corner frequency above which the power is content is ignored, as well as the percentage of the power to and the starting frequency from where the total power should be computed. The default values for the last three parameters are as shown.

Examination of the sensors, showed that the sensors were in excellent conditions of the long-duration stay on Mir. The top surface on the foot restraint and the touchpad showed little wear from extensive usage and no permanent deformation. The screws connecting the top plate to the rest of sensors had corroded significantly. The environmental control do not function perfectly; the astronauts reported that on occasions they needed to wipe off lots of water that had condensed on equipment. As a result, the corrosion observed is not unexpected. In the inside of the sensor, the strain gages remained well bonded to the flexures and so it was expected that the preflight and postflight calibration results should agree quite well.

There is no inherent axis system associated with the design of the sensors and thus can arbitrary selected. During EDLS, a right-handed orthogonal axis system with the z axis point out of the sensor was chosen as is shown in Figure 5.6.

The calibration procedure carried out post-flight was similar to the one conducted prior to the Mir flight. However, the post-flight calibration was more comprehensive and detailed and is thus explained herein. The procedure was identical for the hand hold and the two foot restraints. On the touchpad moment testing was omitted since no moments were recorded. Figure 5.7 shows how the sensors were set-up for calibration when loads in the $-y$-direction were applied. The entire procedure was as follows:

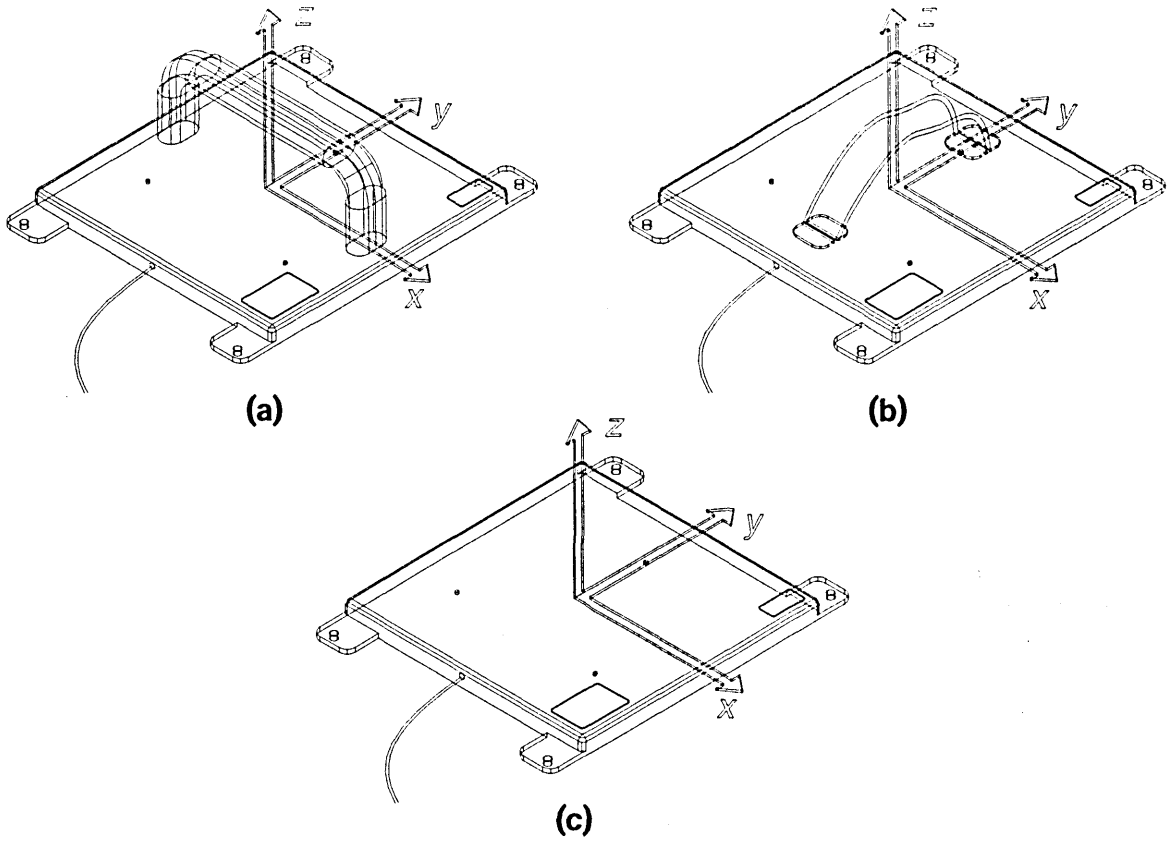


Figure 5.6: The figure shows orientation of the right-handed orthogonal axis-system used in the EDLS experiment for (a) the handhold, (b) the foot restraints, and (c) for the touchpad. During DLS a different axis system, a left-handed one, was used as is shown in Figure 4.1 on page 64.

- The top plate of the sensor was removed and replaced with an aluminum plate of approximately the same size with an L-shaped aluminum bar on top of it.
- The sensor with the L-shaped bar was mounted vertically on a rack, so that the gravity would act either in the $-y$ direction or in the $-x$ direction.
- Thirteen weights ranging from 1.00 to 40.13 lb (0.455–18.20 kg) were suspended from the bar. There were four possible locations on the bar from which the weights were hung: (1) Flushed against the plate to create no moment, (2) in the middle of the long-section of the L-shaped bar to create a small moment, (3) at the end of the long-section of the L-shaped bar to create a large moment, and (4) at the end of the short section of the L-shaped bar to create moments about two axes.
- The sensor was then rotated on the rack as to apply forces and moments about another axis.

- To record forces in the z-direction, the sensor was placed on a flat surface, the L-shaped bar and its plate removed, and the weights placed at the center of the sensor, resting directly on the flexures.

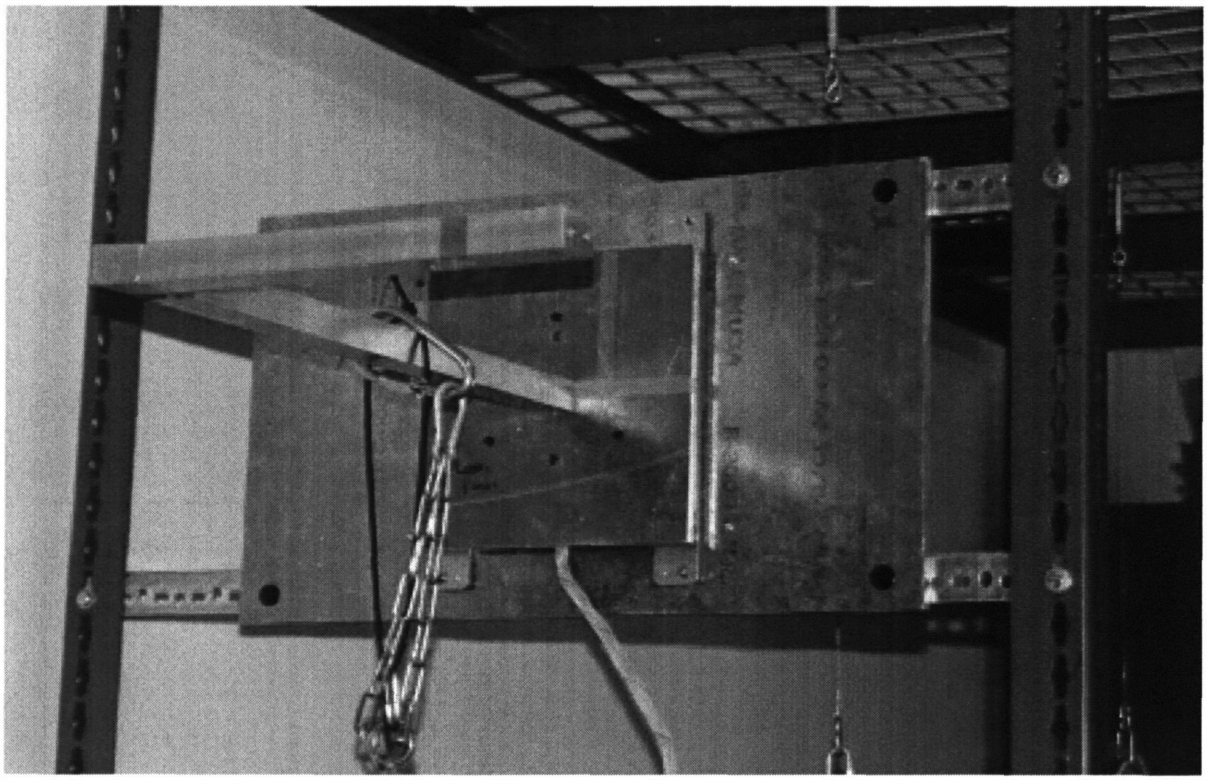


Figure 5.7: This photograph shows the setup for the postflight calibration of the sensors. The sensor is mounted on a rack and a plate with an L-shaped bar is attached to it. Weights are hung from the bar with moment arms as needed. Note that the weights were not suspended from the big chain in the but rather on the thin plastic band behind it. The chain was there in case the plastic band failed.

The data was recorded with the MASU ESM under the same protocol (#99) as the spaceflight data and PC Card 11, returned blank from space, was used to store the calibration data. To test the reliability of the sensors and the repeatability of the data collected, the same loads were applied under slightly different conditions and in different time intervals.

The raw calibration data files were converted into Matlab files with EDLSAP but with the force and moment readings remaining in counts as in the raw data. Figure 5.8 shows the force measurements and Figure 5.9 the torques from a calibration session of the second foot restraint (EDLS Sensor #4). The nearly staircase shaped function shows how higher and higher loads were applied. The no loading case and the excessive loading cases occurred when weights were put on or removed.

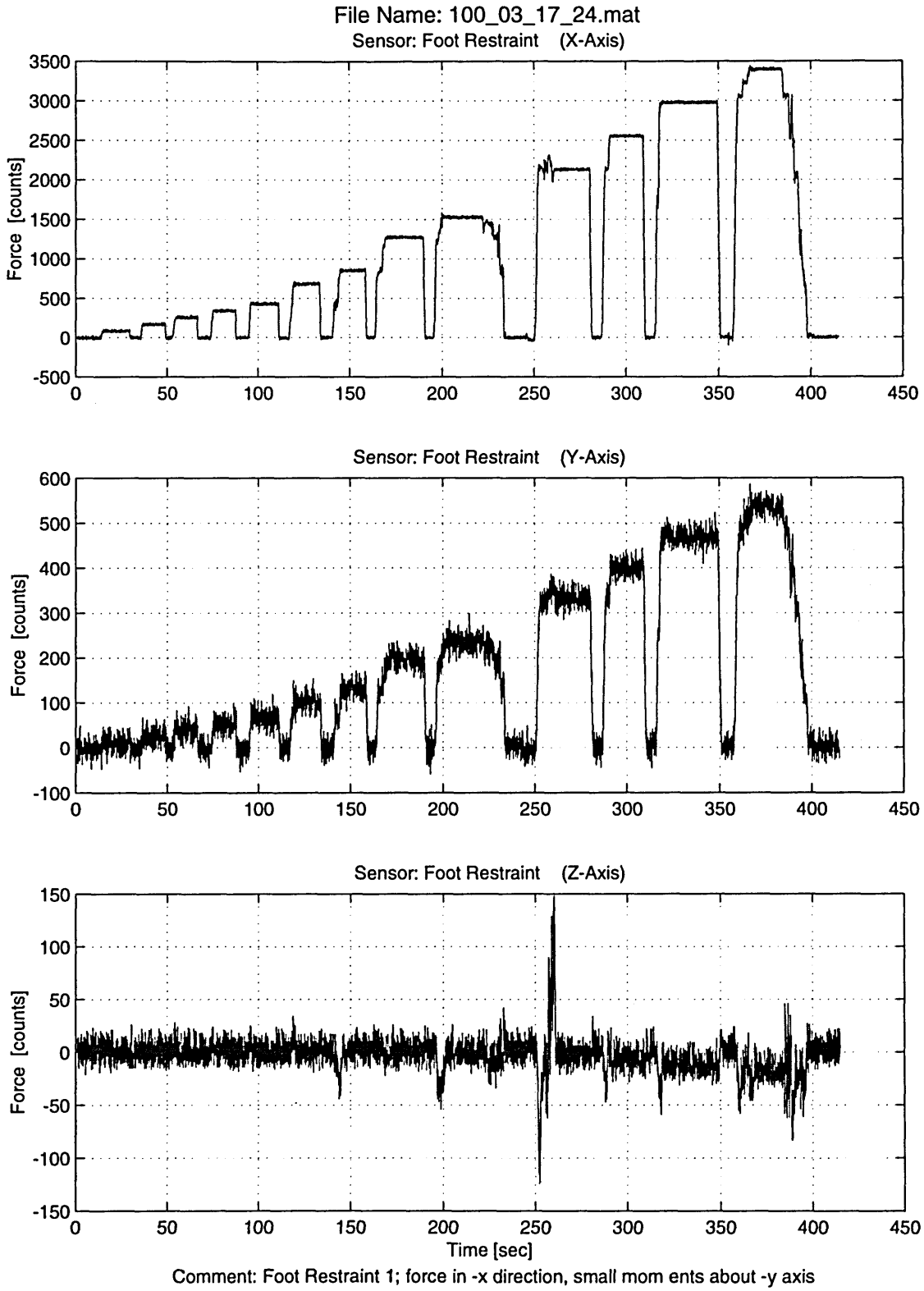


Figure 5.8: The plots show the measured "forces," in counts, during the calibration of Sensor 2.

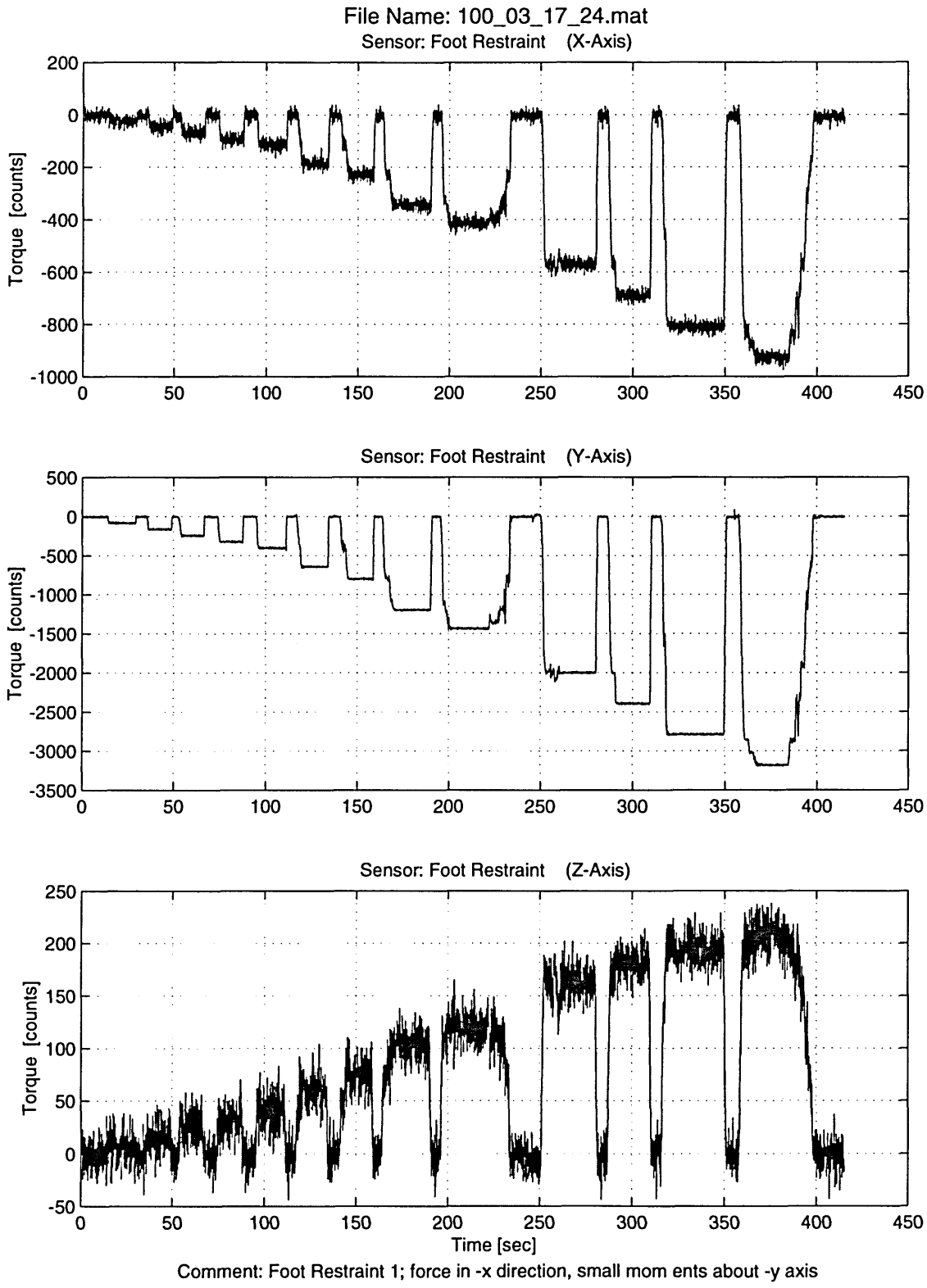


Figure 5.9: The plots show the measured "torques," in counts, during the calibration of Sensor 2.

EDLSAP's signal average value function was used to determine the average number of counts measured under the load condition. The applied forces and moments (in Newtons and Newton-meters, respectively) as well as the average measured forces and moments (in counts) were entered into a spreadsheet as well as the average number of counts that were recorded while the load was applied. The applied loads and the counts measured for the data shown in Figure 5.8 and Figure 5.9 is shown in Table 5.6. The table lists thirteen of the over 90 load cases that were recorded for this sensor.

Table 5.6: Sample Calibration Data for the Foot restraint 2 Sensor

Applied Forces [N]			Applied Torques [Nm]			Average Measured Forces [counts]			Average Measured Torques [counts]		
X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
-4.5	0	0	-1.2	0	0	83.1	10.3	0.8	-24.7	-83.3	5.8
-8.9	0	0	-2.5	0	0	169.0	24.2	0.8	-47.2	-163.6	14.6
-13.3	0	0	-3.7	0	0	254.0	37.7	0.8	-69.9	-243.7	22.9
-17.8	0	0	-5.0	0	0	339.2	51.9	0.8	-92.8	-324.0	31.4
-22.2	0	0	-6.2	0	0	425.2	66.3	0.8	-114.4	-403.9	40.5
-35.6	0	0	-9.9	0	0	679.5	104.6	0.8	-184.0	-642.7	60.6
-44.5	0	0	-12.4	0	0	849.2	132.4	0.8	-227.2	-801.2	75.6
-66.7	0	0	-18.6	0	0	1272.2	195.2	1.2	-343.8	-1197.0	103.2
-80.1	0	0	-22.3	0	0	1526.2	235.2	-0.2	-412.3	-1434.7	121.0
-111.8	0	0	-31.2	0	0	2130.7	332.3	-2.4	-571.9	-2001.4	162.7
-134.1	0	0	-37.4	0	0	2554.6	400.2	-6.4	-689.0	-2396.4	181.5
-156.3	0	0	-43.6	0	0	2977.9	469.1	-11.5	-808.7	-2790.1	194.5
-178.6	0	0	-49.8	0	0	3396.0	536.2	-16.5	-928.2	-3183.8	206.5

The calibration matrix relates measured loads to applied loads for each sensor. It is calculated by the method of least squares as follows: Given a 6-element vector **a** consisting of the applied forces (in [N]) in the three axes and the applied moments (in [Nm]) about three axes and a vector **m** of measured forces along the three axes and moments about the three axes (both in [counts]), the relation between the two quantities is given by

$$\mathbf{a} = C\mathbf{m}, \tag{5.1}$$

where C is the so-called calibration matrix. Since there were some 90 calibration load cases, the quantities **a** and **m** must be written as $6 \times n$ matrices A and M, so that Eqn. (5.1) is rewritten as

$$CM = A. \tag{5.2}$$

Taking the transpose and multiplying by M yields,

$$MM^TC^T = MA^T, \tag{5.3}$$

Taking the pseudo-inverse of MM^T and then the transpose of the entire equation, yields the following expression for the calibration matrix C:

$$C = ((MM^T)^{-1}MA^T)^T \tag{5.4}$$

EDLSAP computes the calibration matrix for each sensor and compares the accuracy of the measured loads (using the calibration matrix found by the method of least squares) with the actually applied loads. This is illustrated in Figure 5.10, which shows the accuracy of the Fx measurements for the handhold sensor. Similarly, Figure 5.11 shows the accuracy of the M_z measurements for the first foot restraint.

Not every load case that was recorded was used in the computation of the calibration matrix. If an individual measurements (versus a large error of several measurements in a row) resulted in a very high error, it was eliminated, so were the cases in which more than 30 lb were suspended such as to create large moments about two axes, where the bending of the L-shaped aluminum bar was to large for an accurate measurement. The accuracy of the measurements determined in the calibration is summarized in Table 5.7.

As was mentioned earlier, EDLS data was collected under two configurations:

- FHT Configuration: Footrestraint 1, Hand hold, Touchpad
- FFT Configuration: Foot restraint 1, Foot restraint 2, Touchpad

As a result, the calibration matrices for the four individual sensors are combined into two block-diagonal 15×15 matrices, one for the FHT configuration and one for the FFT configuration. By multiplying the raw data with the appropriate 15×15 calibration matrix, the signals from the six channels (in counts) are converted into three forces (in Newtons) and three moments (in Newton-meters).

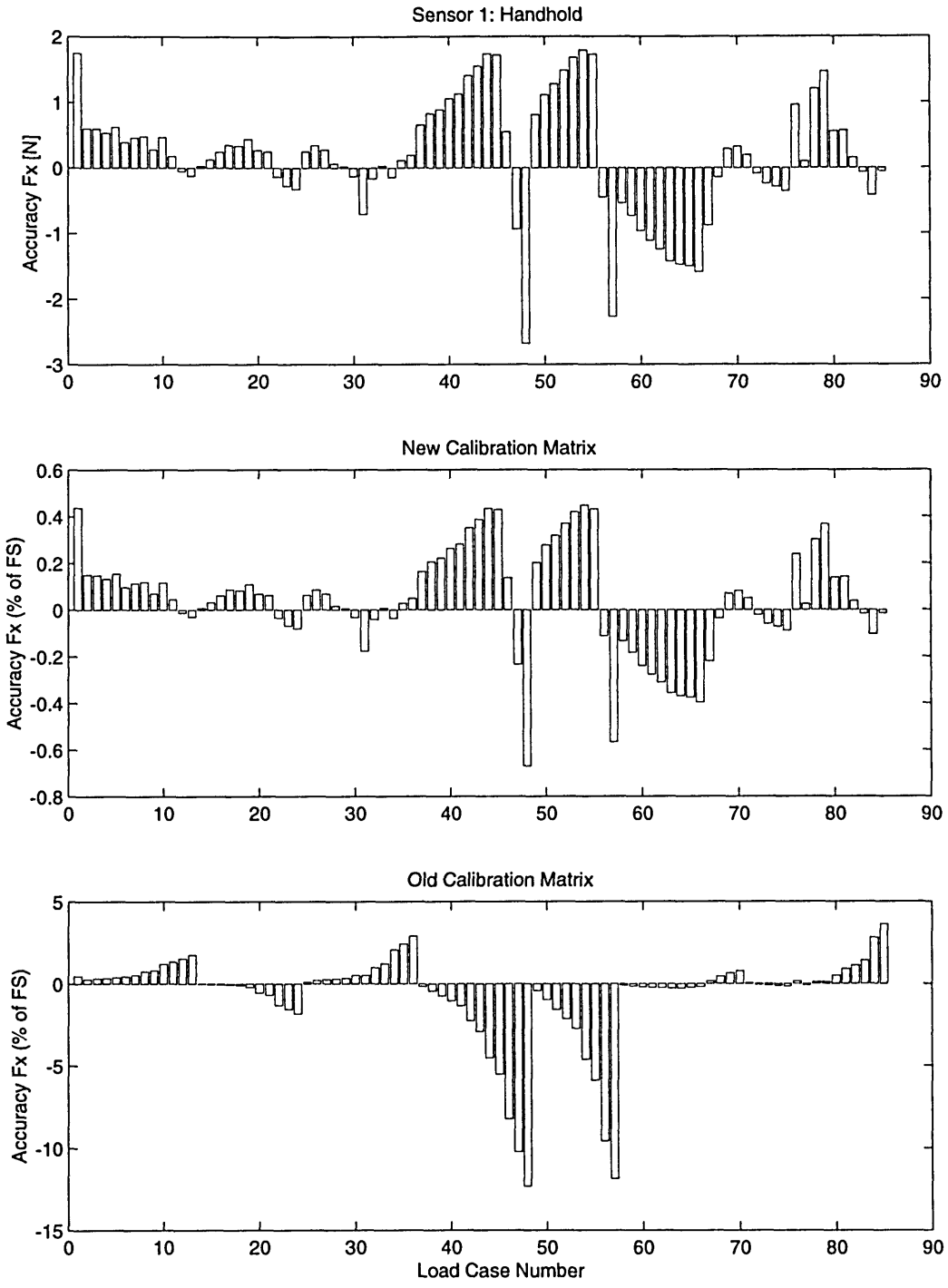


Figure 5.10: The plots in this figure show the accuracy of force measurements along the x-axis of the handhold sensor. Each bar represents one load case. The top plot gives the error between the applied and the measured load in Newtons. The middle plot shows the accuracy of the measured load as a percentage of the full scale load (400 N) using the new calibration matrix and the bottom plot shows the same accuracy using the calibration matrix computed for the MODE/EDLS ESM.

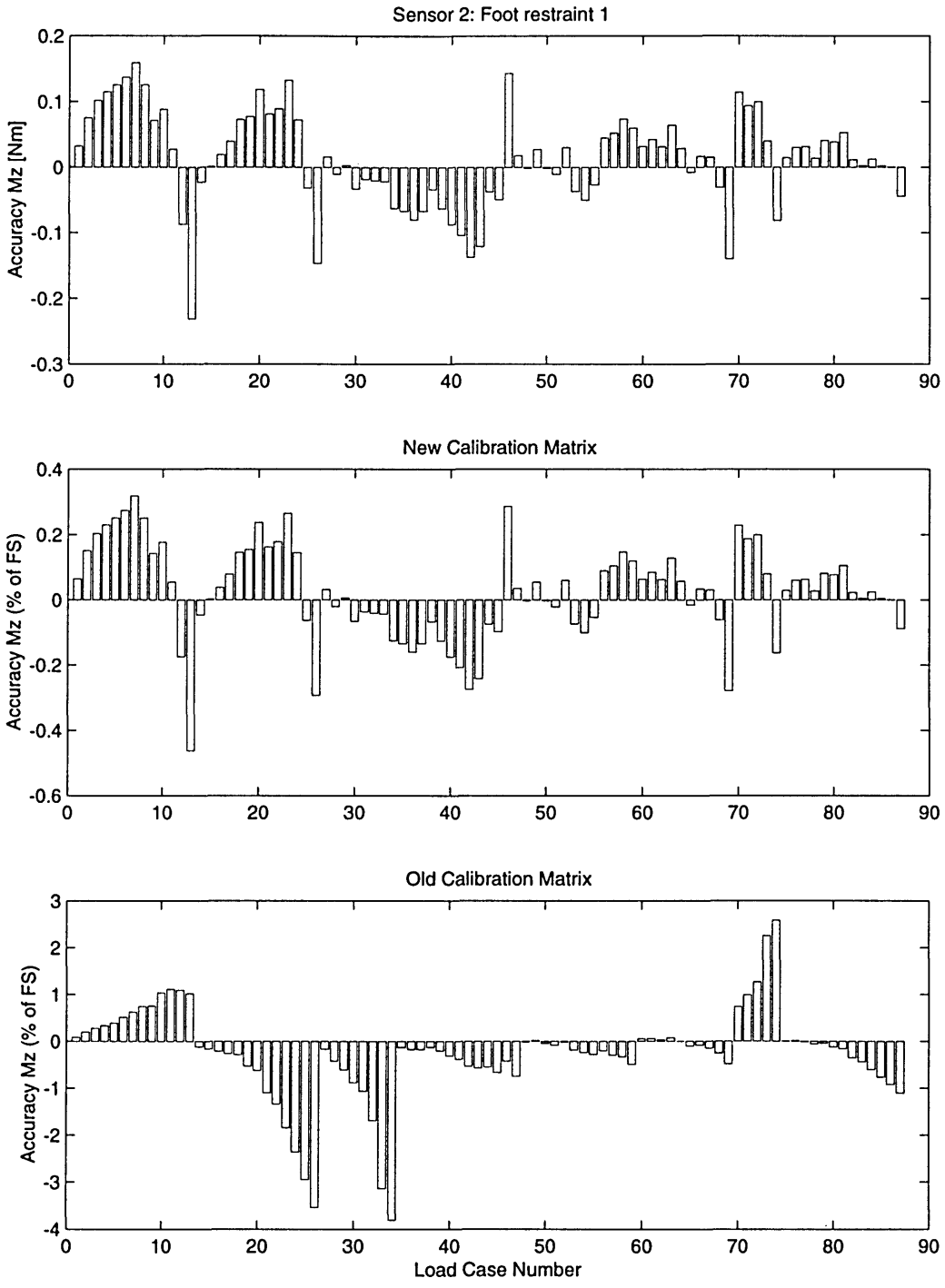


Figure 5.11: The plots in this figure show the accuracy of moment measurements about the z-axis of the foot restraint sensor. Each bar represents one load case. The top plot gives the error between the applied and the measured load in Newton-Meters. The middle plot shows the accuracy of the measured load as a percentage of the full scale load (50 Nm) using the new calibration matrix and the bottom plot shows the same accuracy using the calibration matrix computed for the MODE/EDLS ESM.

Table 5.7: Accuracy of Post-flight Sensor Calibration

Properties	Max Load [N] or [Nm]	OLD CALIBRATION MATRIX Accuracy as a % of Full Scale		NEW CALIBRATION MATRIX Accuracy as a % of Full Scale	
		Worst Positive	Worst Negative	Worst Positive	Worst Negative
Sensor 1: Handhold					
Fx	178.567	3.64%	-12.30%	0.45%	-0.67%
Fy	178.567	4.63%	-3.22%	0.57%	-0.81%
Fz	178.567	15.60%	0.22%	0.57%	-0.55%
Mx	49.821	13.02%	-2.66%	1.57%	-0.94%
My	46.092	5.26%	-0.28%	0.43%	-0.45%
Mz	11.373	4.61%	-2.12%	0.60%	-0.56%
Sensor 2: Foot restraint 1					
Fx	178.567	6.31%	-12.79%	0.69%	-0.50%
Fy	178.567	1.29%	-4.13%	0.63%	-0.89%
Fz	178.567	1.65%	-11.25%	0.20%	-0.37%
Mx	64.35	3.99%	-1.84%	1.26%	-0.79%
My	49.821	1.50%	-1.99%	0.27%	-0.54%
Mz	11.373	2.58%	-3.81%	0.32%	-0.46%
Sensor 3: Touchpad					
Fx	178.567	9.31%	-18.02%	1.37%	-1.49%
Fy	178.567	14.22%	-0.22%	0.52%	-0.71%
Fz	134.07	6.73%	-2.59%	0.87%	-0.82%
Sensor 4: Foot restraint 2					
Fx	178.567	6.08%	-12.25%	0.42%	-0.39%
Fy	178.567	4.17%	-2.91%	1.06%	-1.24%
Fz	178.567	15.57%	-0.52%	0.81%	-0.94%
Mx	49.821	4.38%	-3.81%	1.48%	-1.38%
My	49.821	8.13%	-0.11%	0.45%	-0.47%
Mz	17.059	2.24%	-3.47%	0.50%	-0.38%
Fx, Fy, Fz, Mx, My, Mz		**15.60%**	**-12.79**	**1.57%**	**-1.49%**

The post-flight calibration matrix for the handhold (Sensor 1) using the MASU ESM was found to be:

$$C_1 = \begin{bmatrix} -47.06 & -416.16 & -119.65 & 831.48 & -21.77 & -415.56 \\ -53.32 & 723.59 & -9.07 & 4.54 & 4.29 & -717.59 \\ -688.04 & -7.55 & -704.51 & -18.57 & -717.21 & 27.16 \\ 28.90 & 18.00 & -69.65 & 2.16 & 31.39 & -16.47 \\ -53.74 & 10.73 & 2.94 & -18.97 & 55.15 & 9.56 \\ -0.55 & -36.71 & 2.25 & -37.31 & 1.99 & -40.16 \end{bmatrix} \tag{5.1}$$

The calibration matrix for the first foot restraint (Sensor #2) is given by:

$$C_2 = \begin{bmatrix} 24.99 & -392.80 & -21.94 & 847.87 & 52.81 & -417.04 \\ -121.92 & 691.61 & -77.19 & 4.85 & -69.10 & -728.01 \\ -664.59 & -5.41 & -699.75 & -1.36 & -705.76 & 5.63 \\ 24.62 & 16.20 & -67.11 & 0.18 & 28.03 & -14.33 \\ -53.77 & 9.31 & 0.10 & -20.03 & 53.28 & 10.04 \\ 2.41 & -34.99 & 5.60 & -39.20 & 5.38 & -37.06 \end{bmatrix} \tag{5.2}$$

The calibration matrix for the touchpad is of size 3×3 since only three channels were sampled:

$$C_3 = \begin{bmatrix} -2386.90 & -1273.83 & 2839.81 \\ -1280.39 & 644.59 & 1443.92 \\ -683.29 & 364.01 & -1461.35 \end{bmatrix} \tag{5.3}$$

The calibration matrix for the second foot restraint (Sensor #4) is given by:

$$C_4 = \begin{bmatrix} -21.53 & -444.77 & -84.42 & 824.37 & -1.43 & -404.31 \\ -78.29 & 765.83 & -47.84 & 17.88 & -26.22 & -719.28 \\ -712.66 & 765.83 & -47.84 & 17.88 & -26.22 & -719.28 \\ 31.17 & 18.76 & -65.88 & 4.18 & 31.84 & -15.53 \\ -56.49 & 11.72 & 2.07 & -19.65 & 55.16 & 10.36 \\ 0.51 & -37.66 & 2.41 & -36.17 & 2.63 & -41.01 \end{bmatrix} \tag{5.4}$$

The calibration matrices for all four sensors are assembled into two large calibration matrices—one for each configuration:

$$C_{\text{FHT}} = \begin{bmatrix} C_2 & 0 & 0 \\ 0 & C_1 & 0 \\ 0 & 0 & C_3 \end{bmatrix} ; \; C_{\text{FFT}} = \begin{bmatrix} C_2 & 0 & 0 \\ 0 & C_4 & 0 \\ 0 & 0 & C_3 \end{bmatrix} \tag{5.5}$$

The calibration matrices in Eqn. (5.5) assume that the 15 data channels are sorted according to the list in the right part of see Table 5.4. However, since this is not automatically the case (see left part of Table 5.4), during processing the data columns have to be switched appropriately before multiplying the raw data with the calibration data. The calibration matrix is left-multiplied with the matrix containing the raw data:

$$D_{\text{processed, FHT}} = \left(\frac{10}{65536} C_{\text{FHT}}\right) D_{\text{raw}} \tag{5.6}$$

$$D_{\text{processed, FFT}} = \left(\frac{10}{65536} C_{\text{FFT}}\right) D_{\text{raw}} \tag{5.7}$$

The factor of (10 / 65536) accounts for the A/D converter. An input voltage range from –5V to +5V and is sampled with a 16-bit resolution and to produces counts in the range from –32768 to +32768. Each channel of the MASU ESM had a system gain as listed in Table 5.8 and so one can normalize the calibration matrix to a gain of one. The resulting four matrices are shown in (5.7) through (5.4).

Table 5.8: System Gains of the MASU ESM

EDLS Number	A/D Channel	Purpose	System Gain
1	1	Sensor 1, V1	1.83
2	2	Sensor 1, V2	1.72
3	17	Sensor 1, V3	1.82
4	18	Sensor 1, V4	1.72
5	19	Sensor 1, V5	4.88
6	20	Sensor 1, V6	4.58
7	3	Sensor 2, V1	5.26
8	4	Sensor 2, V2	4.45
9	8	Sensor 2, V3	4.35
10	26	Sensor 2, V4	4.09

Table 5.8: System Gains of the MASU ESM *(continued)*

EDLS Number	A/D Channel	Purpose	System Gain
11	30	Sensor 2, V5	4.72
12	31	Sensor 2, V6	4.69
13	32	Sensor 3, V1	4.83
14	15	Sensor 3, V2	4.46
15	16	Sensor 3, V3	4.20

The handhold matrix:

$$C_1 = \begin{bmatrix} -8.95 & -93.52 & -27.51 & 203.30 & -4.61 & -88.60 \\ -10.14 & 162.60 & -2.09 & 1.11 & 0.91 & -153.00 \\ -130.81 & -1.70 & -161.96 & -4.54 & -151.95 & 5.79 \\ 5.50 & 4.04 & -16.01 & 0.53 & 6.65 & -3.51 \\ -10.22 & 2.41 & 0.68 & -4.64 & 11.96 & 2.04 \\ -0.10 & -8.25 & 0.52 & -9.12 & 0.42 & -8.56 \end{bmatrix} \tag{5.1}$$

The foot restraint 1 matrix

$$C_2 = \begin{bmatrix} 13.66 & -228.37 & -12.05 & 492.95 & 10.82 & -91.06 \\ -66.62 & 402.10 & -42.41 & 2.82 & -14.16 & -158.95 \\ -363.17 & -3.14 & -384.48 & -0.79 & -144.62 & 1.23 \\ 13.46 & 9.42 & -36.87 & 0.11 & 5.74 & -3.13 \\ -29.38 & 5.41 & 0.05 & -11.64 & 10.92 & 2.19 \\ 1.32 & -20.34 & 3.08 & -22.79 & 1.10 & -8.09 \end{bmatrix}, \tag{5.2}$$

the touchpad matrix

$$C_3 = \begin{bmatrix} -494.18 & -285.61 & 676.15 \\ -265.09 & 144.53 & 343.79 \\ -141.47 & 81.62 & -347.94 \end{bmatrix}, \tag{5.3}$$

and finally the foot restraint 2 matrix

$$C_4 = \begin{bmatrix} -4.09 & -99.95 & -19.41 & 201.56 & -0.30 & -86.21 \\ -14.88 & 172.10 & -11.00 & 4.37 & -5.56 & -153.36 \\ -135.49 & 0.48 & -162.01 & 0.62 & -148.80 & 2.01 \\ 5.93 & 4.22 & -15.14 & 1.02 & 6.75 & -3.31 \\ -10.74 & 2.63 & 0.48 & -4.80 & 11.69 & 2.21 \\ 0.10 & -8.46 & 0.55 & -8.84 & 0.56 & -8.74 \end{bmatrix} \tag{5.4}$$

To examine both the validity of the calibration matrix as well as a drastic change in the signal output from the strain gages, the post-flight calibration matrices from the MASU ESM can be compared with the preflight calibration matrices obtained with the original MODE/EDLS ESM. In these matrices, the "volts per count" factor of (20 / 8192) is not included. Since the MODE/EDLS ESM had a unity gain for all channels, no normalizing of the matrices is necessary. The handhold matrix is

$$C_1 = \begin{bmatrix} -4.46 & -46.47 & -26.64 & 132.32 & -15.41 & -50.93 \\ -8.05 & 89.02 & 5.23 & 0.53 & 1.19 & -92.14 \\ -93.54 & 1.73 & -94.42 & -0.75 & -90.08 & -1.03 \\ 3.49 & 2.24 & -8.52 & 0.12 & 3.87 & -2.26 \\ -6.96 & 1.26 & 0.7 & -2.17 & 7.24 & 1.28 \\ 0.48 & -4.24 & 0.93 & -4.24 & 0.54 & -5.22 \end{bmatrix}, \tag{5.5}$$

the foot restraint 1 matrix is

$$C_2 = \begin{bmatrix} -8.92 & -116.22 & -48.53 & 332.34 & -14.02 & -50.75 \\ -24.87 & 229.79 & -9.99 & -2.08 & -1.06 & -91.23 \\ -165.31 & 1.7 & -177.51 & -0.33 & -69.81 & 0.19 \\ 8.73 & 6.56 & -21.62 & 0 & 4.29 & -2.62 \\ -16.7 & 3.15 & 0.88 & -5.13 & 7.18 & 1.3 \\ 2.35 & -12.41 & 2.44 & -10.92 & 0.58 & -4.5 \end{bmatrix}, \tag{5.6}$$

the touchpad matrix is

$$C_3 = \begin{bmatrix} -200.28 & -95.99 & 186.32 \\ -311.28 & 56.14 & 290.55 \\ -137.49 & 22.95 & -129.97 \end{bmatrix}, \tag{5.7}$$

and the second foot restraint matrix is

$$C_4 = \begin{bmatrix} 8.68 & -47.36 & -6.78 & 130.58 & 0.51 & -49.08 \\ -17.77 & 96.11 & -9.81 & -1.83 & -7.36 & -92.98 \\ -97.5 & 1.14 & -93.23 & 1.12 & -90.27 & -0.93 \\ 3.71 & 1.91 & -9.02 & -0.08 & 3.81 & -1.77 \\ -7.71 & 1.39 & 0.22 & -1.42 & 6.9 & 1.32 \\ 0.68 & -5.69 & 0.83 & -4.82 & 0.43 & -4.69 \end{bmatrix}. \quad (5.8)$$

The two set of matrices agree well in so far that the order of magnitude of the numbers match and most signs are the same. The next step undertaken was to re-process the calibration data files with the newly obtained matrices and compare the results with the loads applied. Figure 5.12 and Figure 5.13 show the same calibration data as Figure 5.10 and Figure 5.11. This time, however, the calibration matrices were used during the processing to obtain (real) forces in Newtons and Newton-meters respectively.

A comparison with the applied loads verifies the accuracy precision of the sensors. The error for the handhold and the two foot restraints is less than 2%, as is consistent with Table 5.7. The largest error, as is expected, occurs in the touchpad data. While its hardware is identical to that of the other sensors, only 3 signals instead of six are available to compute the forces applied. A comparison showed that the touchpad force data tends to be higher than the actually applied loads. The error can be as high as 20%. While the touchpad was little used during STS-62 and NASA 2, Jerry Linenger made heavy use of it by modifying it into a foot restraint (see Figure 4.7 on page 73).

All raw data files from both the NASA 2 and NASA 4 mission were batch-processed with EDLSAP. Since most raw data files were very large and included hours of data, the processed data files are limited to 5 minutes of data. For example, the NASA 4 data from May 9th, 1997 on PC Card 6 was in a single data file 246.770 MBytes in length. After processing, there were 42 MATLAB data files, each holding 5 minutes of EDLS data (except for the last one which has less). The 5-min. length was selected based on the capability of the computers to process the raw data[4] and to display and edit the processed data.

4. The processing of raw EDLS data with EDLSAP requires a lot of computer memory. Not because the data is so large but because MATLAB's memory handling is not the best and short spikes and memory requirements occur.

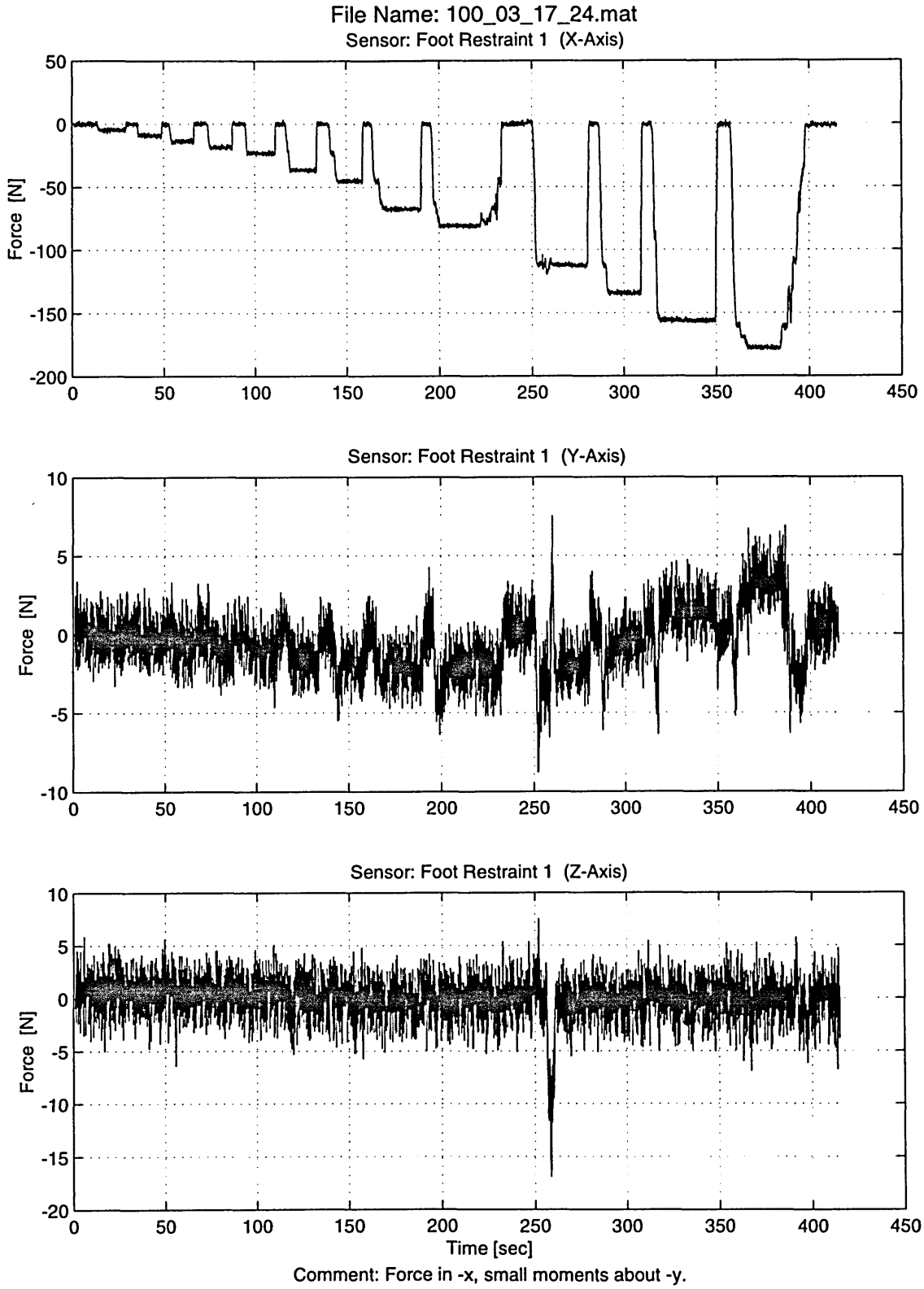


Figure 5.12: The plots show the same calibration data as Figure 5.11 but this time in Newtons.

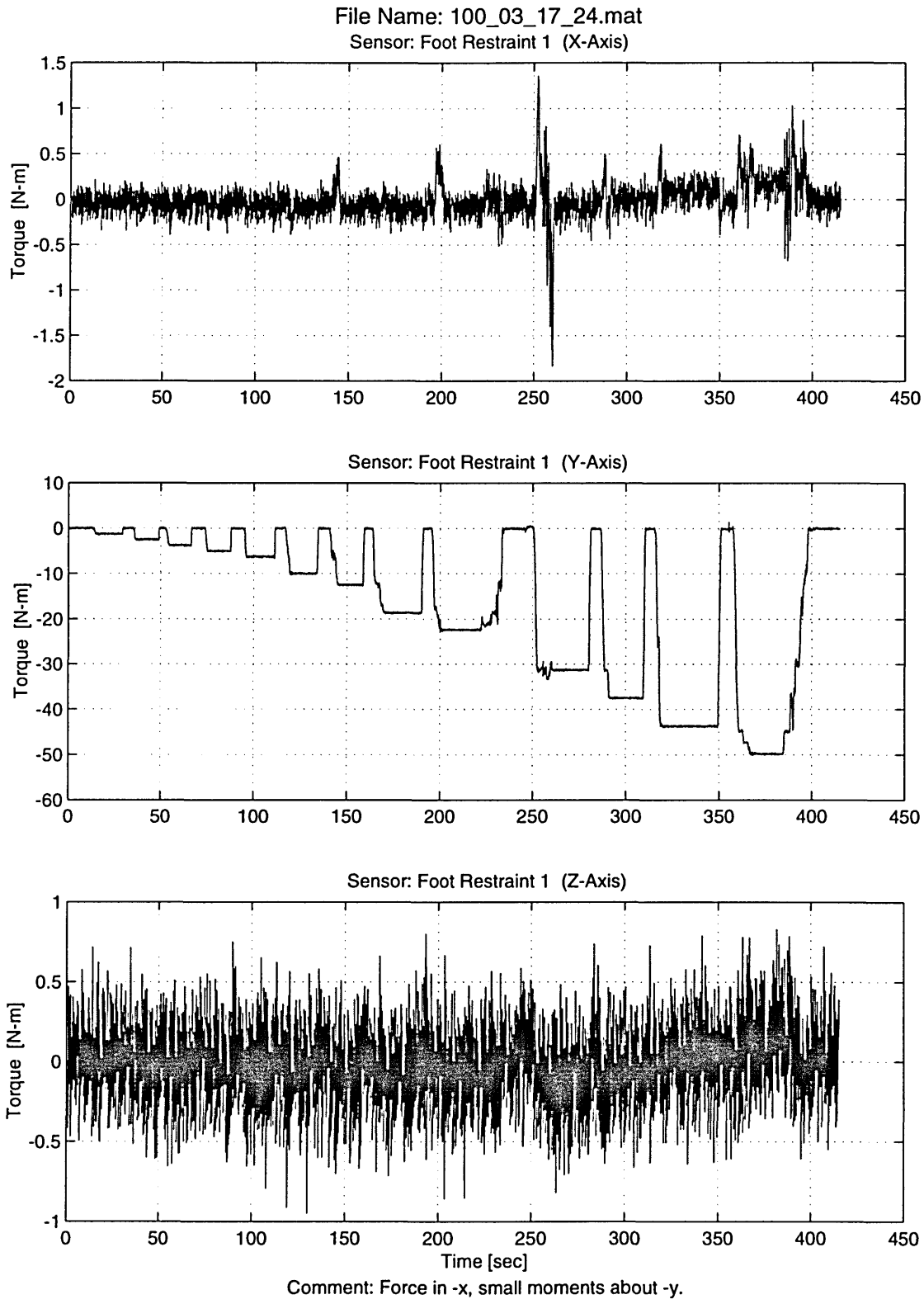


Figure 5.13: The plots show the same calibration data as Figure 5.12 but this time in Newton-meters.

The data file 99_1.dat on PC Card 02 (from 12 March 1997) was slighlt corrupted. More precisely, the second buffer contained bytes from an unknown sources (for example, the computer's memorry, other portions of the disk). According to the buffer tag of buffer 2, the third buffer should begin at Offset 121,984 instead it is found at Offset 265,344. The rest of the file seemed completely normal.

In processing the fraw data ile, the first two buffer containing 7.552 seconds of data samples were not included in the MATLAB file and the timestamp adjusted accordingly. As a result the timestamp is only approximate since it is unknown how long it took the computer to insert the erronous data in the EDLS data file. The error in time could range from a few microseconds to a few seconds with the former more likely.

The raw and processed EDLS data from all PC Cards was burned on CD-ROMs and is being distributed to NASA and Boeing as a contract deliverable. How the actual data analysis will be conducted is discussed in Chapter 7. The chapter includes also data from a typical event.

CHAPTER

6

DESIGN AND DEVELOPMENT OF ADVANCED LOAD SENSORS

During the first incremental design review for the International Space Station in the spring of 1995, the following controlling parameters for the microgravity environment on ISS were identified: (1) Crew activity, (2) Acoustics, (3) Equipment vibrations, (4) Station configuration, (5) Scheduled activities, (6) Payload locations, and (7) Transient disturbances [42].

The purpose of the advanced load sensors is to help control the issue of crew disturbances by measuring the astronaut-induced loads. The importance of making such measurements and possible mitigation approaches are addressed in the Microgravity Control Plan for the entire ISS Program. Document SSP 50036, Revision A from February 1996 states on page 3-3:

> While the effects of crew activity on the microgravity environment are not the contractual obligation of any program entity, crew induced acceleration disturbances may be significant and must be considered in any comprehensive microgravity control plan and spacecraft design approach. Crew activity disturbances need to be characterized and quantified, and appropriate design and operational solutions developed and integrated with the hardware control effort. Crew translation aids and awareness training, and active and passive rack level isolation techniques are among the mitigation strategies available. [61]

So far no progress has been made to implement the strategies suggested. An advanced version of the DLS/EDLS sensors could fill the existing void in this area. However, they would not only help in the operation of the ISS during microgravity mode and at other times, but also serve as a standard experimental hardware for purely scientific experiments. Existing plans for such experiments are discussed in Chapter 7.

Sections 6.1 and 6.2 provide information that guided the preliminary design of the advanced sensors. They discuss the restraints and mobility aids that will be available on the International Space Station

and the technical lessons that were learned from the EDLS experiment. Section 6.3 lists the objectives and the requirements for the sensors as well as the design approach taken. The final section of the chapter presents the actual preliminary design.

6.1 ISS Restraints and Mobility Aids

The DLS/EDLS sensors were designed to have the same functionality and "feel" as the foot loops and hand rails installed on the Space Shuttle orbiter for the IVA of the crew. The advanced sensors will reflect the design of the analogous aids on the ISS. The mechanisms are formally known as *Restraints and Mobility Aids* (R&MAs) and are a subsystem of the Flight Crew Systems (FCS). R&MAs are provided to support IVA personnel restraint, IVA equipment restraint, and IVA personnel mobility. They can be moved around to meet the changing work and translation needs and removed when necessary for maximum aisle clearance. By default, the R&MAs are located at frequently used workstations and at paths most frequently travelled. Figure 6.1 shows a photograph of all flight crew restraint and mobility aids as will be available after Flight 2A.

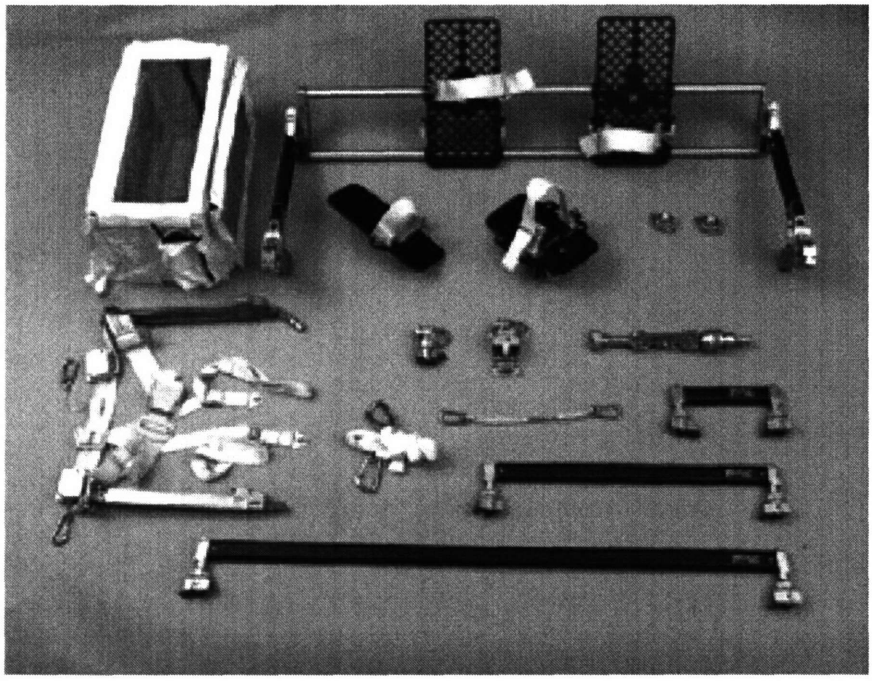


Figure 6.1: This photograph shows the restraint and mobility aids for the International Space Station after delivery from the contractor to NASA's JSC (NASA Image S97-13524).

A common attachment mechanism, called *seat track*, is used for all portable R&MAs. It is a standard interface between R&MAs hardware and all U.S.-, Russian, ESA-, and Japanese ISS components. The seat track simply extruded aluminum and part of the rack structure and all racks have seat track mounted on their front (i.e., aisle-facing) side. This can be seen on the station in the U.S. Laboratory mock-up in Figure 2.5 on page 37. Small sections of seat track, called *seat track buttons*, are provided throughout modules to support maintenance and other activities. A seat track button has a length of 4.11 cm (1.62"). A seat track is shown in Figure 6.2

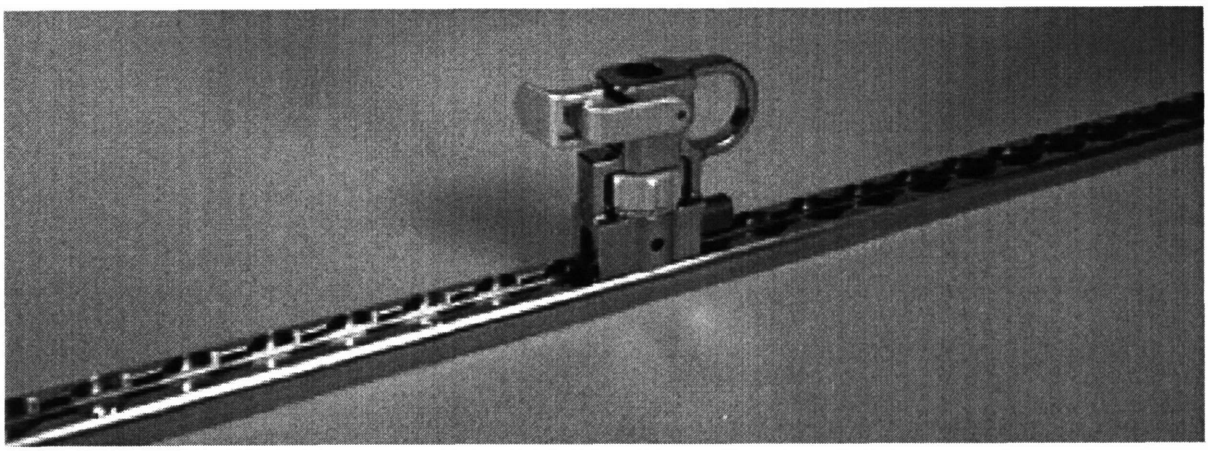


Figure 6.2: A seat track with a Seat Track Equipment Anchor Assembly (STEA) attached to it (NASA Image S97-13410).

There are two types of mobility aids: *handrails* and *rack handle assemblies* (RHA's). The removable handrails serve as convenient handholds, attachment points for various crew and equipment anchors, tethers, etc. Handrails are offered in three lengths, 21.6 cm, 54.6 cm, and 104.5 cm (8.5", 21.5", and 41.5", respectively), and attach to rack fronts through the seat track interface. The 104.5 cm handrail is long enough to span the width of a rack and thus can be mounted perpendicular to the two seat tracks along the rack or along a single track. The middle-size handrails are located along primary crew paths and around hatches. The short haindrails are mounted along less traveled crew paths and also used as equipment handles. Rack Handle Assemblies are simple removable handles used to move racks. When not in use they are stowed away.

The RHA's are standardized on the USOS, the European, and Japanese ISS components. The Russian On-orbit Segment will have the same type of handrails as those found on Mir. Because the cross-section is circular rather than oval, U.S. attachments for the rails will be incompatible to the Russian handrails.

There are five types of personnel restraint devices:

1. Short-Duration Foot Restraint (SDFR)
2. Long-Duration Foot Restraint (LDFR)
3. Anchor Foot Restraint (AFR)
4. Torso Restraint
5. Long Duration Crew Restraint (LDCR)

The SDFR consists of a single metal foot plate, cloth foot loop, and clamping mechanism for connecting to a handrail. The device allows easy ingress/egress while providing sufficient restraint to prevent the crewmember from floating away. An SDFR is depicted in Figure 6.2.

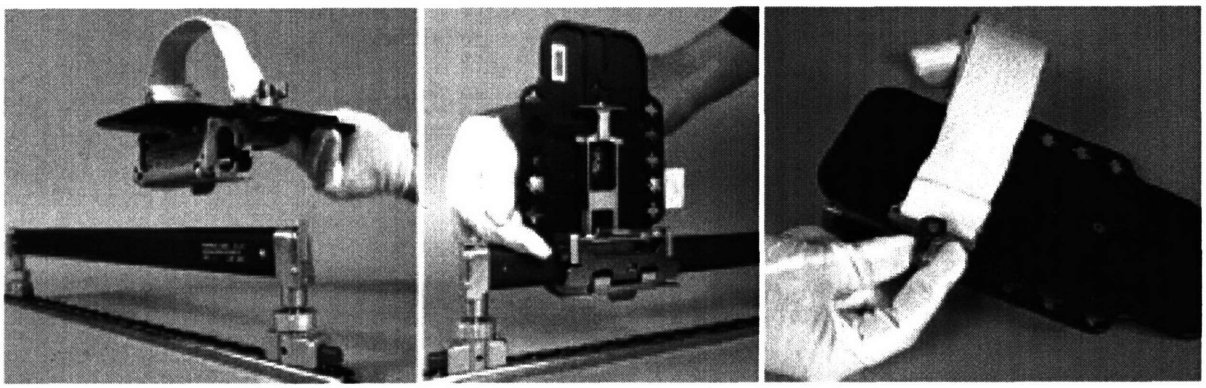


Figure 6.3: The photographs show the Short Duration Foot Restraint. The first image shows the interface to a handrail for a horizontal attachment; the second image shows the SDFR mounted in vertical position, and the third image is a close-up view of the foot plate with the loop (NASA Images S97-13453, S97-13556, and S97-13451).

The LDFR is similar to an SDFR. It consists of two instead of one metal foot plate with foot loop and is attached to a seat track instead of a handrail. It is intended for extended length operation such as working on a workstation or heavy-duty tasks (requiring extensive or forceful motions). A schematic of an LDFR is shown in Figure 6.4. An AFR is made of the same foot plate / cloth loop as the other foot restraints but can be attached to a single seat track. The Torso Restraint is in essence a rope put around the waist and attached to a rack via extension rods. The rods are attached to a seat track with an anchor (known as a Seat Track Equipment Anchor or STEA). The Torso Restraint would be used alone or in conjunction with an LDFR. The fifth aid is the LDCR which serves the same role an LDFR but uses foam rollers (similar to those found in weight lifting machines) between legs to keep the astronaut in place.

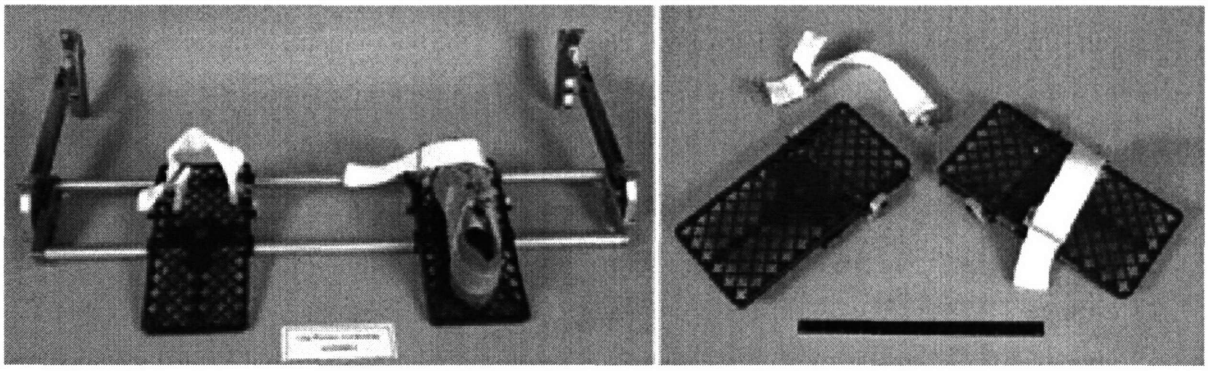


Figure 6.4: The Long Duration Foot Restraint consists of two foot plates with loops, a rail assembly, and two braces to mount to a seat track (NASA Images S97-13592 and S97-13517).

As is evident from this description, ISS R&MAs are more sophisticated than any previous system. Velcro or tape are no longer the universal solutions for attaching items. The refined and standardized aids provide a lot of opportunities in the design of the advanced load sensors.

6.2 Lessons Learned from EDLS

This section summarizes the experiences made with the first-generation hardware that was flown on the Shuttle and on Mir. The lessons from DLS and EDLS were learned through an examination of the spaceflight hardware, video tape recordings of the astronauts using the sensors, and direct feedback from the crew. It is important to keep in mind that neither the first nor the second ESM used during DLS and EDLS were designed solely for these experiments, and thus the designers of both the hardware and software were restricted in the modifications that they incorporated. In addition, the replacement ESM, had to support not just EDLS but also the MiSDE experiment. Henceforth, some of the design decisions made were known to be sub optimal. In addition, Jerry Linenger was trained with the original ESM but then faced a different ESM and flight procedures that were a poor mixture of those written specifically for the previous system. While well-written flight procedures were prepared and translated, they were never brought to Mir.

- When the MASU ESM is booted up, the system checks the status of the various hardware components. Each item tested is listed on the screen followed by the words "PASSED" or "FAILED." Since there are many components and few lines of text displayed on the screen, the information scrolls by too quickly if one is unfamiliar with the system. The ESM featured two PC Card drives, a primary and a secondary one and could be used simultaneously to copy files. When recording

data, the astronauts were instructed to insert a blank PC card into the primary drive and leave the second drive empty. If the second drive was empty during the boot-up phase, the message "Secondary drive... FAILED" would appear. The word "failure" by itself raised unnecessary concerns each time it appeared. In addition, due to the quick scrolling and the crew member's focus on reading the flight procedures, the astronaut noticed that something "failed" but had missed what item was the source of the warning.

- The three DLS or EDLS sensors were connected to the ESM via an umbilical cable (see Section 4.3.3). Before data is acquired, the system performs an "auto-zeroing" so that when no load is applied to the sensor the signal is zero. Since the sensors are fairly delicate instruments, the auto-zeroing of one sensor does not apply to another. In addition, electrical power is being transmitted during all times when the ESM is acquiring data. Because of these two reasons, it is not permissible to hot-swap sensors (i.e., exchange two sensors while the power is on). However, this was done by a crew member, causing no physical damage, but affecting the signal quality and therefore making it difficult to process the raw data.
- During an EDLS session, the ESM display shows a timer counting down from 23:59:59 hours (default setting) to zero. The timer confirmed that the system was running but not whether a signal, exceeding the threshold level specified in the protocol, was being recorded. An *unlabeled* LED lit up whenever data was written to the storage medium. The presence of data buffer, made it difficult for the astronauts to conclude whether the system was recording properly and whether the threshold level needed adjustment.
- The ESM's internal clock is not battery-buffered and is completely dependent on external time synchronization, which is important to correlate two different experiments (e.g., EDLS and SAMS). If no time synchronization signal is received, the ESM, once powered up, maintains an internal clock starting time at 0:00:00. During the NASA 2 mission the external time source, MIPS, failed, and so one cannot even tell well on which day, data was recorded. Also the time kept by the astronauts (on their watches or on the camcorders) often varies with the "official" time associated with the data (when time synchronization worked). During the activation of the sensors, the time the computer uses (from whatever source) is not displayed and neither can the astronaut enter the time her/she uses as a reference.
- While the two foot restraints look identical they are *not* interchangeable. Although labeled, the labels are usually too small to read on a video tape recording. If the two sensors were easier to distinguish (e.g., different color of cloth for the loops), the data processing would have been simpli-

fied. However, the optimal solution would be an "auto-detect" feature in the ESM, that would determine which sensors are attached.

- Because of the high workload that the astronauts face in orbit and the lengthy flight procedures, astronauts tend not to keep as good of an experimental log as investigators would like. If information is written down, there is no assurance that it is legible (it is difficult to find a flat surface to write on) or that it will find its way through the various bureaucracies to the researchers. It is thus prudent to query important or at least critical, information from the astronaut performing the experiment in the start-up procedure and thus have the information electronically stored and as part of the data. In the case of EDLS, the configuration, either FFT or FHT, is a critical knowledge for processing the data but did not have to be entered into the ESM during activation and so had to be obtained through detective work from various sources.
- Before collecting the first data of the NASA 4 mission using the "borrowed" ESM, astronaut Jerry Linenger was asked to perform an on-orbit "calibration." This consisted of tapping the sensors at specific points in a specific sequence. Such a procedure had very little use. This was all that could be done since there was no device to apply a known force on the sensors in orbit. Incidentally, light tapping with a finger or with a pen (as Jerry did sometimes for time synchronization) produced a force smaller than the electronic noise-level on the MASU ESM was not visible in the data plots.
- On Mir, some of the astronauts wear only socks, while other prefer wearing light boots which provide a feeling of support on the sole. Those that wear socks had the option of using stirrup type attachment to their pants to supply some pressure on the foot. In general the foot loops were too large for the astronauts and so they constantly moved up and down in the loop. On the positive side though, the size and flexibility of the cloth loop allowed the astronaut to stay in the FR but still make large rotations about his/her vertical axis or lean on either side.

6.3 Objectives, Requirements, and Constraints for Advanced Sensors

Primary Objective:

To record and store for later analysis the forces and moments applied by the crew onto the space station during nominal intra-vehicular activities and to provide real-time feedback to the crew on the magnitude of the applied loads.

Secondary Objective:

To record and store force and moment data of human subjects for scientific experiments.

Functional Requirements

1. Each sensor shall record the force components along the x-, y-, and z-axis and the moments about these axes.
2. The minimum force and moment resolutions shall be 2 N and 0.2 Nm, respectively.
3. The maximum allowable load shall be 400 N and 50 Nm.
4. The sampling frequency shall be no less than 120 Hz.
5. The signal conditioning shall include a low-pass filter of adjustable corner frequency.
6. The system interface shall be simple enough such that the required crew time for set-up should be no more than 3 minutes.

Operational Requirements

1. Each sensor shall be able to serve as a handhold, foot restraint, or touchpad and be modifiable by the crew on-orbit.
2. The system shall provide immediate feedback on the magnitude of an applied load and a visual and (optional) auditory warning when a specified value is exceeded.
3. The system shall be able to synchronize its internal time with the ISS-wide time system.
4. The sensor shall have a variable threshold event detection.
5. The sensor shall be able to be re-calibrated on-orbit.
6. The data shall be recorded on common, commercially available, storage media.
7. The system shall be able to accept power from standard ISS utility control panels providing 124 Volts dc (available in the US, European, and Japanese segments).

6.4 Design of Advanced Load Sensors

As the postflight calibration has shown, the quality and accuracy of the sensors is excellent and those there is no need to make an improvement in the design of the load cells and/or their arrangement. However, the top plate of the sensor with the handle or the loop attachment has been redesigned. As has the ESM been.

Instead of a central computer supporting three sensors, each advanced sensor will be incorporate its own electronics and thus become a self-sufficient unit. Fundamentally, it consists of two parts: (1) the Sensor Restraint Unit (SRU) and (2) the Sensor Electronics Unit (SEU). Former contains, as the name

implies, the IVA restraint device (i.e, handle/foot loop) as well as the load cells. The latter unit incorporates most of the electronics that the original ESM contained. The two units can be attached to each to form a single and compact "box" but during operation that two units would mostly be connected via an umbilical.

Designing the next generation sensors, or any kind of similar equipment, for the ISS is a more challenging task than for an existing system such as the Shuttle since the requirements and designs of all subsystems are completed or documentation is difficult to obtain. Since the assembly process itself will take six or more years and lifetime is expected to be 15 or more years, the selection of the technologies to be incorporated into the sensors must be forward looking. This can be accomplished by making use of the latest commercial off-the-shelve systems as was done and described this section. The consequence of that decision is that few components have space-flight qualification. However, the cost for spaceflight qualification can be shared with other users easier in COTS than in customized hardware and so the design objective has been to maximize COTS hardware for the advanced sensors.

The current technological level in solid-state electronics would likely permit to miniaturize most of the electronics in the MASU ESM into a case with the same width and length as an EDLS sensor and no more than 3 centimeters in height. This would require the development of an application specific integrated circuit (ASIC) and the development cost of such a system is prohibitive for the funding available. Instead of extreme down scaling, the technology used for *embedded systems* was examined as a basis for the advanced sensors.

Motorola's Semiconductor Group is the leading provider of embedded systems and offers a range products. The latest offering is the PowerPAQ—a reference design that can be used to develop handheld systems. The PowerPAQ was developed to support image capture, wireless connectivity, speech recognition, Global Positioning Systems receivers, etc.

The PowerPAQ design consists of three Printed Circuit Boards (PCB) each measuring 10.2 x 7.4 x 3.2 cm. The three cards have the following functions:

- The CPU Card

  This card incorporates either a Motorola MPC821 or MPC823 processor. These processor belong the PowerPC chip family based on RISC technology. They share the core architecture and instruction set with the PPC750 (or G3) processor found in Apple Macintosh computers. The MPC8xx family offers up to 100 MIPS at a very low power consumption and heat output. The CPU Card

includes the main power management circuitry and the main memory consisting of a 4 MByte flash memory and 4 MByte DRAM (upgradeable to 32 MB).

- The I/O Card has connectors for a headset, microphone, speakers, a Color TFT Active Matrix Display, a Touchscreen, a VGA monitor, a keyboard, and RS-232 connection.
- The PCMCIA Card accepts Type II PC Cards (5 mm height) internally and on the PowerPAQ with MPC823 offers a USB connection.

The EDLS sensors consist of a bottom plate, three load cells, a top plate, a cable with a connector, and an optional attachment (the foot loop or the handrail). The load cell arrangement and the bottom and top plate have worked very well and no area for improvement has been identified. The advancement lies in the electronics, the attachments, and how the sensors are mounted.

The sensor electronics is housed in the so-called Sensor Electronics Unit (SEU), which has the same footprint (24 x 24 cm) as the sensor and a height of 10 centimeters. The SEU consists of two component layers. The bottom layer contains the three PowerPAQ cards side by side connected by a ribbon cable instead of stacked together (as in the original Motorola arrangement). The Type II PC Card slot is filled with a commercial IOtech 16-bit, 100 kHz A/D converter.

Power to the system is provided through an Intel/Duracell battery (common in laptop computers). As on the MASU ESM, data storage is provided through Type III PC Card hard drives from Calluna. Each Card has a capacity of 540 MBytes. The PC Card reader/writer is connected and powered via the USB connection from the PowerPAQ. Cooling is provided through a small fan as found in laptop computers. The sensor's top plate providing slits for air circulation.

The second layer is occupied to 70% by the 6-channel signal conditioning card. This would be a modified version of the commercial IOtech 8-channel card (DBK-43A) in which two channels are eliminated to conserve space and fit the card inside the SEU. The top layer has also a mechanism to hold a a space PC Card Calluna hard drive in place.

The SEM is connected via custom-built umbilical to the sensor. The top surface of the SEM contains a Sharp 6.4" VGA TFT Active Matrix Display and a keyboard identical to that in the MASU ESM.

THERE ARE APPROX. 15 MORE DESIGN PAGES THAT WILL BE ADDED. INCLUDING DRAWINGS, PHOTOGRAPHS AND DETAILED SPECS.

CHAPTER

7

FUTURE DEVELOPMENT

7.1 Data Analysis

After the raw data has been processed, it is necessary to locate and identify individual events for detailed analysis. The DLS data was expected to be continues and MATLAB routines were written that searched automatically through hours of data. When a certain threshold was crossed from one "direction" and then later crossed from the other "direction," an event was found. With EDLS the same approach will not work so automated. Since the handhold and foot restraint #2 sensors were occasionally hot-swapped, there are very large spikes in the signal followed by a an unloaded level that is far from zero. Since the noise level is not constant and quite high (sometimes as much as +/– 10 N), the process is further complicated. As a result, much of the cutting out of noise, unbiasing if necessary, and saving of events in individual small data files has to be done manually with EDLSAP.

From the video footage a new list of typical astronaut motion has been assembled. The list differs somewhat from that of DLS in Table 4.1 on page 66 because the geometry of the Mir modules is quite different than that of the Space Shuttle middeck, the on-orbit stay is much longer, and the restraint aids are not identical. These EDLS list of motions has to be finalized and can then be used for data analysis.

Events found in the EDLS data will be grouped into two fundamental categories. Those that can be correlated to video footage and those that cannot be. For all events, the maximum force magnitude, maximum rms force, and 95% power spectral density will be computed. Those events that can be associated with a certain type of motion will be sub-categorized appropriately, so that average forces, and spectral densities for each type of motions can be computed. As is evident from this description this effort is a very labor-intensive and lengthy.

As has been mentioned before, the EDLS risk mitigation experiment involves also modeling of human motion in space and comparing to the EDLS data. This research effort has been undertaken since Sep-

tember of 1997. Because of the much larger amount of video footage available from the NASA 4 mission, only Jerry Linenger's could be used for comparison with the models. To allow the modeling efforts to proceed without a completed post-flight calibration, the original MODE/EDLS ESM calibration data (for NASA 2) was used to process the raw data and obtain approximate data for NASA 4. With the final data available those events used in the modeling effort will be revisited and the quality of the models determined.

Data snipped containing just events and grouped by category and sub-category will be recorded onto CD-ROMs and distributed along with the raw data and the entire processed data. The final deliverable to NASA will be a contractor report. The report will draw upon large portions from this thesis, present the results of the data analysis, the modeling efforts, and include the CD-ROMs with the various sets of EDLS data.

7.2 Further Development of the Advanced Sensors

The advanced sensors are in a preliminary design phase. The next step involves the integration of Motorola's PowerPAQ with the off-the-shelve version of the IOtech signal conditioning and A/D converter, screen, and keyboard, to create a breadboard prototype of the Sensor Electronics Module. This phase will also include the manufacturing of the case for SEM. The hardware will be tested to determine the feasibility of the SEM design for spaceflight.

Modifications to the hardware design will be made as necessary following the tests. Customized versions of those hardware components needing adjustments will be procured from the manufactures and integrated into a prototype. This prototype will undergo lengthy tests to examine compliance with the requirements set forth for ISS payloads and equipment. Which tests and how they are to be performed as well as the requirements to be met are specified in ISS documents, such as SSP 30237 "Space Station Electromagnetic Emission and Susceptibility Requirements" and many others.

Most likely the largest effort will involve the writing and debugging of software drivers for the various hardware components and the development of a user interface. Drivers for the IOtech equipment are of course already written and integrated into IOtech's DaqView software but, modifications and adjustments will be necessary and thus requiring collaboration and sharing of source code with IOtech.

The development of the user interface for the sensors will be done on Mac OS / Windows desktop machines with a standard application development environment for the PowerPAQ.

REFERENCES

[1] Baugher, Charles R., Gary L. Martin, and Richard DeLombard, "Low-Frequency Vibration Environment for Five Shuttle Missions" NASA Technical Memorandum 106059, January 1993.

[2] Conway, Bruce A., "Mathematical Crew Motion Disturbance Models for Spacecraft Control System Design," S.M. Thesis, Virginia Polytechnic Institute, August 1974.

[3] DeLombard, Richard, "Compendium of Information for Interpreting the Microgravity Environment of the Orbiter Spacecraft," NASA Technical Memorandum 107032, August 1996.

[4] DeLombard, Richard, "Microgravity Environment Countermeasures Panel Discussion," 35th Aerospace Sciences Meeting & Exhibit, AIAA Paper 97-0351, January 1997.

[5] DeLombard, Richard, "SAMS Acceleration Measurements on Mir From November 1995 to March 1996," NASA Technical Memorandum 107435. April 1997.

[6] DeLombard, Richard, B. D. Finley, and C. R. Baugher, "Development of and Flight Results from the Space Acceleration Measurement System (SAMS)," 30th Aerospace Sciences Meeting & Exhibit, AIAA Paper 92-0354, January 1992.

[7] DeLombard, Richard and Finley, Brian D., "Space Acceleration Measurement System Description and Operations on the First Spacelab Life Sciences Mission," NASA Technical Memorandum 105301, November 1991.

[8] DeLombard, Richard and Kevin McPherson, "Comparison Tools for Assessing the Microgravity Environment of Missions, Carriers and Condition," NASA Technical Memorandum 107446, April 1997.

[9] DeLombard, Richard and et al., "An Overview of the Microgravity Environment and Its Effects on Science," AIAA Paper 96-0401, 34th Aerospace Sciences Meeting & Exhibit, January 1996.

[10] DeLombard, Richard. et al., "Further Analysis of the Microgravity Environment on Mir Space Station During Mir-16," NASA Technical Memorandum 107239, June 1996.

[11] Ellison J., G. Ahmadi, and C. Grodsinsky, "Evaluation of Passive and Active Vibration Control Mechanisms in a Microgravity Environment," *Journal of Spacecraft*. Vol. 32, No. 2., 1994.

[12] Feonychev, A. I., B. J. Tillotson, L. P. Torre, and H. J. Willenberg, "Characterization of the Microgravity Environment on the Salyut and Mir Space Stations," SPIE Vol. 2220, 1994.

[13] Finley, B. D., C. M. Grodsinsky and R. DeLombard, "Microgravity Environment of STS-57 and Summary Comparisons with Selected Previous Missions," AIAA Paper 94-0435, 32nd Aerospace Sciences Meeting & Exhibit, January 1994.

[14] Graham, S. J. and R. C. Rhome, "Achievements in Microgravity: Ten Years of Microgravity Research," 32nd Aerospace Sciences Meeting & Exhibit, AIAA Paper 94-0344, January 1994.

[15] Grodsinsky, Carlos M. and Gerald V. Brown, "Low Frequency Vibration Isolation Technology for Microgravity Space Experiments," NASA Technical Memorandum 101448, September 1989.

[16] Harman, P. and D. Rohn, "The Impact of an IVA Robot on the Space Station Microgravity Environment," AIAA Paper 89-0596, 27th Aerospace Sciences Meeting, January 1989.

[17] Hendricks T. C. and Johnson C. H., "Stochastic Crew Motion Modeling," *Journal of Spacecraft*. Vol. 8, No. 2, February 1971.

[18] Hoyt B, and G. H. Kitmacher, Priroda Interface Control Document (PICD), Report JSC-27027/ SM-P-Eng-01, July 1995.

[19] —, "International Space Station Assembly Sequence (5/31/98: Revision D)," NASA Facts. NASA Johnson Space Center, June 1998.

[20] —,"International Space Station Creating a World-Class Orbiting Laboratory," NASA Facts.

[21] —, "International Space Station: Assembly Complete," NASA Facts LG-1997-07-456-HQ, NASA Headquarters, July 1997.

[22] —, "International Space Station, U.S. Space Station History," NASA Johnson Space Center, IS-1995-08-SS004JSC, August 1995.

[23] —, "International Space Station, Russian Space Stations," NASA Johnson Space Center, IS-1995-08-SS005JSC, August 1995.

[24] Klute, Glenn K., "Pilot Investigation: Nominal Crew Induced Forces in Zero-G, "SAE Technical Paper 921155, July 1992.

[25] Kullas, Conlon M., "Handbook on Astronaut Crew Motion Disturbances for Control System Design," NASA Reference Publication 1025, 1979.

[26] Moskowitz, Milton E., "SAMS Acceleration Measurements on Mir From March to September 1996," NASA Technical Memorandum 107524, August 1997.

[27] Newman, Dava J., Michail Tryfonidis, and Marthinus C. van Schoor, "Astronaut-Induced Disturbances in Microgravity," Journal of Spacecraft and Rockets. Vol. 34, No. 2, March–April 1997.

[28] Poli, Corrado R. "Effect of Man's Motion on the Attitude of a Satellite," *Journal of Spacecraft*, Vol. 4, No. 1, January 1967.

[29] Roberson, Robert E., "Comments on the Incorporation of Man into the Attitude Dynamics of Spacecraft," *The Journal of the Astronautical Sciences*. Vol. X, No. 1, Spring 1963.

[30] Robey, J., "Planning for Microgravity Science Research on the International Space Station," 35th Aerospace Sciences Meeting & Exhibit, AIAA Paper 97-0106, January 1997.

[31] Rochon, Brian V. and Steven A. Scheer, "Crew Activity & Motion Effects on the Space Station," Presented at the Workshop on Structural Dynamics and Control Interaction of Flexible Structures, Gerorge C. Marshall Space Flight Center, April 22–24, 1986.

[32] Rogers, M. J. B., et al., "A Comparison of Low-gravity Measurements On-board *Columbia* During STS-40." *Microgravity Science & Technology.* Vol. VI, No. 3, 1993.

[33] Thomson, W. T. and Y. C. Fung, "Instability of Spinning Space Stations Due to Crew Motion." AIAA Journal, Vol. 3, No. 6, June 1965.

[34] Conway, Bruce A., "Investigation of Crew Motion Disturbances on Skylab-Experiment T-013," *Advances in the Astronautical Sciences*, Vol. 31, Part 1, Proceedings of the AAS 20th Annual Meeting, 1974.

[35] Conway, Bruce A. and T. C. Hendricks, *A Summary of the Skylab Crew/Vehicle Disturbance Experiment T-013*, NASA Technical Note D-8128, March 1976.

[36] Bizony, Piers, *Island in the Sky: Building the International Space Station*, Aurum Press, 1996.

[37] Rose, David, International Space Station Familiarization, NASA Space Flight Training Division, Mission Operations Directorate, ISS FAM C 21109, December 1997.

[38] Sutliff, Thomas J., "An Acceleration Measurement Capability On International Space Station Supporting Microgravity Science Payloads," 35th Aerospace Sciences Meeting & Exhibit, AIAA Paper 97-0349 / NASA Technical Memorandum 107415, January 1997.

[39] Del Basso, S., "The International Space Station Microgravity Environment," 34th Aerospace Sciences Meeting & Exhibit, AIAA Paper 96-0402, January 1996.

[40] Moore, Cherice et al., "Phase 1: A Journey to Mir 1994–98" CD-ROM SM-G-472, Lockheed Martin, 1998.

[41] Newman, Dava J., "Crew Force Measurements: Dynamic Load Sensors (DLS) on Mir," Proposal to NASA Office of Life & Microgravity Sciences & Applications Life and Biomedical Sciences and Applications Divisio, May 1994.

[42] Plang, Ronnie D., et al., "International Space Station Alpha Reference Guide, Incremental Design Review #1," Boeing Defense & Space Group, March 29, 1995.

[43] Tryggvason, B. V., B. Y. Stewart, and J. DeCarufel, "The Microgravity Vibration Isolation Mount: Development and Flight Test Results," IAF-97-J.2.04, 48th International Astronautical Congress, October 1997.

[44] Sutliff, Thomas J., "Requirements and Development of an Acceleration Measurement System for International Space Station Payloads Microgravity Science Payloads," NASA Technical Memorandum 107484, May 1997.

[45] Web Site http://www.mcs.net/~rusaerog/asifs/asif2.html#anchor39451

[46] Web Site http://nauts.com/histpace/vehiclesNT/histmirNT.html

[47] Web Site http://shuttle-mir.nasa.gov/shuttle-mir/ops/crew/resident.html

[48] Portree, David S. F., "Mir Hardware Heritage," NASA RP 1357, March 1995.

[49] Web Site http://station.nasa.gov/gallery/lineart/index.html

[50] Buckingham, Bruce, "International Space Station Partners Adjust Target Dates For First Launches, Revise Other Station Assembly Launches", Kennedy Space Center Release No. 66-98, May 31, 1998.

[51] Web Site http://station.nasa.gov/partners/index.html

[52] —, "International Space Station: Benefits from the Shuttle-Mir Program," NASA Facts, NASA Lyndon B. Johnson Space Center, March 1998.

[53] —,"Sign Agreement For International Space Station," NASA Press Release.

[54] Info Sheet on SAMS by LeRC

[55] Unity Connecting Module

[56] Control Module FGB

[57] ISS Reference: station.nasa.gov/reference/statistics/index.html

[58] http://www.hq.nasa.gov/osf/mir/priroda.html

[59] http://station.nasa.gov/reference/factbook/stats.html

[60] Boeing Defense & Space Group, "System Specifications for the International Space Station," SSP 41000, Revision H (Advanced Copy), Type 1, NASA, 25 May 1998.

[61] Johnson Space Center, "Microgravity Control Plan," SSP 50036, Revision A, Type 2, NASA, 29 February 1996.

[62] —. "ISS: Continuing Journey," *Microgravity News*, Vol. 5, No. 1, Spring 1998.

[63] Larson, Wiley J., and James R. Wertz (eds.), *Space Mission Analysis and Design*, 2nd Edition, Microcosm, 1992.

[64] —, "Middeck 0-Gravity Dynamics Experiment (MODE)," Interface Control Document ICD A-21148, Revision B, NASA JSC Space Shuttle Program Office, January 1994.

[65] Newman, D. J., Marthinus van Schoor, and Javier de Luis, "Crew Force Measurements: Dynamic Load Sensors (DLS) on Mir," Proposal to the NASA Office of Life & Microgravity Sciences & Applications, May 1994.

[66] Bokhour, Edward, "EDLS ESM Failure" Memorandum for the Record, Payload Systems, December 4, 1997.

[67] van Schoor, Marthinus C. and Dava J. Newman, "Dynamic Load Sensor Experiment: Background, Science Objectives, Requirements, and Preliminary Design," November 18, 1992.

[68] Crawley, E. F., M. C. van Schoor, and E. Bokhour, *The Middeck 0-Gravity Dynamics Experiment*, NASA CR-4500, March 1993.

[69] http://www.hq.nasa.gov/osf/slide_show/pdf/allslides.pdf

[70] Bokhour, E., R. Grimes, M. Hachkowski, C. Krebs, and J. Zapetis, "Middeck 0-Gravity Dynamics Experiment Flight Systems," *Acta Astronautica*, Vol. 29, No. 10/11, IAF Paper 92-0776, 1992.

[71] Payload Systems, "Safety Report and Conclusions for the Enhanced Dynamic Load Sensor (EDLS)," Document USA-EDLS/95-107, 11 June 1995.

[72] Beckwith, T. G., R. D. Marangoni, and J. H. Lienhard V, *Mechanical Measuements*, 5th Edition, Addison-Wesley, 1993.

[73] DeLombard, Richard, personal e-mail message, August 14, 1998.

[74] Kim, Hyoung M. and Edward B. Bokhour, "Mir Structural Dynamics Experiment: A Flight Experiment Development," AIAA-97-1169.

[75] Rabiner, Lawrence R. and Bernard Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, 1975.

APPENDIXES

APPENDIX A

REFERENCE TABLES

This appendix provides various information that is too detailed to be included in regular chapters but is useful for referencing purposes.

A.1 Damaged Sectors in WORM Disks

Table A.1: Damaged Data Files from NASA 2 Mission

Disk and Side	Files	Bad Sector Blocks
WORM 1A	99_166.DAT	2394
WORM 2B	99_109.DAT	652
WORM 3B	99_2.DAT	84234
WORM 4A	99_2.DAT	168009, 389522
WORM 5A	99_44.DAT	300785
WORM 5B	99_11.DAT	163152, 163739
WORM 6A	99_42.DAT	20902, 59952, 60167, 61712, 62526, 62577, 135544, 255100
WORM 6B	99_5.DAT	115364, 380791
WORM 7B	99_3.DAT	378209
WORM 8A	99_1.DAT, 99_2.DAT	24, 389554
WORM 8B	99_2.DAT	389410
WORM 9A	99_28.DAT	380325
WORM 9B	99_22.DAT ?????	261

A.2 EDLSAP Batch File Format

Raw EDLS data files can be processed either manually by selecting each individual file and specifying the desired parameters such as configuration, length of the processed files, etc. However, due to the

large number of files, EDLSAP can process multiple files at once when a batch file with the necessary information is provided. The structure of the EDLSAP batch file is given in Table A.2. The only difference between batch files for MODE/EDLS ESM raw data and MASU ESM raw data is that for the former the header files must be specified (`.HDR` extension) instead of the data files (`.DAT` extension).

Table A.2: Format of an EDLSAP Batch File for Processing Raw Data

Line	Entry	Example(s)
1	Validity String	`EDLSAP BATCH FILE`
2	Mission	`NASA 4`
3	Source Disk	`PCMCIA_11`
4	Input Filename (including complete path from root level)	For MODE/EDLS ESM MacOS: `MacHD:EDLSAP:RawData:99_5.DAT` Windows: `E:\Edlsap\RawData\99_5.DAT` Unix: `/mit/edls/EDLSAP/RawData/99_5.DAT` For MASU ESM MacOS: `MacHD:EDLSAP:RawData:99_5.HDR` Windows: `E:\Edlsap\RawData\99_5.HDR` Unix: `/mit/edls/EDLSAP/RawData/99_5.HDR`
5	Path of directory to write data files into	MacOS: `MacHD:EDLSAP:DataFiles:` Windows: `E:\Edlsap\DataFiles\` Unix: `/mit/edls/EDLSAP/DataFiles/`
6	Sensor Configuration	`FHT` or `FFT`
7	Desired Units of the Data (1 = Metric, 2 = English, and 3 = counts)	`1`, `2`, or `3`
8	Maximum Length of Processed Data File in Minutes	`10`
9	Comment	`This data was recorded on May 17, 1996.`
	Entries of lines 2 through 9 are repeated for every single raw data file to be processed.	
n	End of batch entries	Two or more carriage return characters (ASCII code 13).

A.3 Ground Sensors to IOtech Signal Conditioning Interface

Table A.3: Cable Connecting EDLS Ground Sensors to IOtech A/D Hardware

DB25 Pin Number	EDLS SENSOR DB25 Wire Color	Description	Channel or Mini-DIN6 Connector Number	Mini-DIN6 Pin Number	Description
1	White	Excitation 1	1	5	Excitation +
1	White	Excitation 1	1	6	Sense +
2	Black	Excitation return 1	1	1	Sense –
2	Black	Excitation return 1	1	2	Excitation –
3	Red	V1 +	1	4	V in +
4	Black	V1 –	1	3	V in –
5	White	Excitation 2	2	5	Excitation +
5	White	Excitation 2	2	6	Sense +
6	Black	Excitation return 2	2	1	Sense –
6	Black	Excitation return 2	2	2	Excitation –
7	Red	V2 +	2	4	V in +
8	Black	V2 –	2	3	V in –
9	White	Excitation 3	3	5	Excitation +
9	White	Excitation 3	3	6	Sense +
10	Black	Excitation return 3	3	1	Sense –
10	Black	Excitation return 3	3	2	Excitation –
11	Red	V3 +	3	4	V in +
12	Black	V3 –	3	3	V in –
13	Brown	Chassis / Shield	3	3	V in –
14	White	Excitation 4	4	5	Excitation +
14	White	Excitation 4	4	6	Sense +
15	Black	Excitation return 4	4	1	Sense –
15	Black	Excitation return 4	4	2	Excitation –
16	Red	V4 +	4	4	V in +
17	Black	V4 –	4	3	V in –
18	White	Excitation 5	5	5	Excitation +
18	White	Excitation 5	5	6	Sense +
19	Black	Excitation return 5	5	1	Sense –
19	Black	Excitation return 5	5	2	Excitation –
20	Red	V5 +	5	4	V in +
21	Black	V5 –	5	3	V in –
22	White	Excitation 6	6	5	Excitation +
22	White	Excitation 6	6	6	Sense +

Table A.3: Cable Connecting EDLS Ground Sensors to IOtech A/D Hardware

DB25 Pin Number	EDLS SENSOR DB25 Wire Color	Description	Channel or Mini-DIN6 Connector Number	Mini-DIN6 Pin Number	Description
23	Black	Excitation return 6	6	1	Sense –
23	Black	Excitation return 6	6	2	Excitation –
24	Red	V6 +	6	4	V in +
25	Black	V6 –	6	3	V in –

APPENDIX

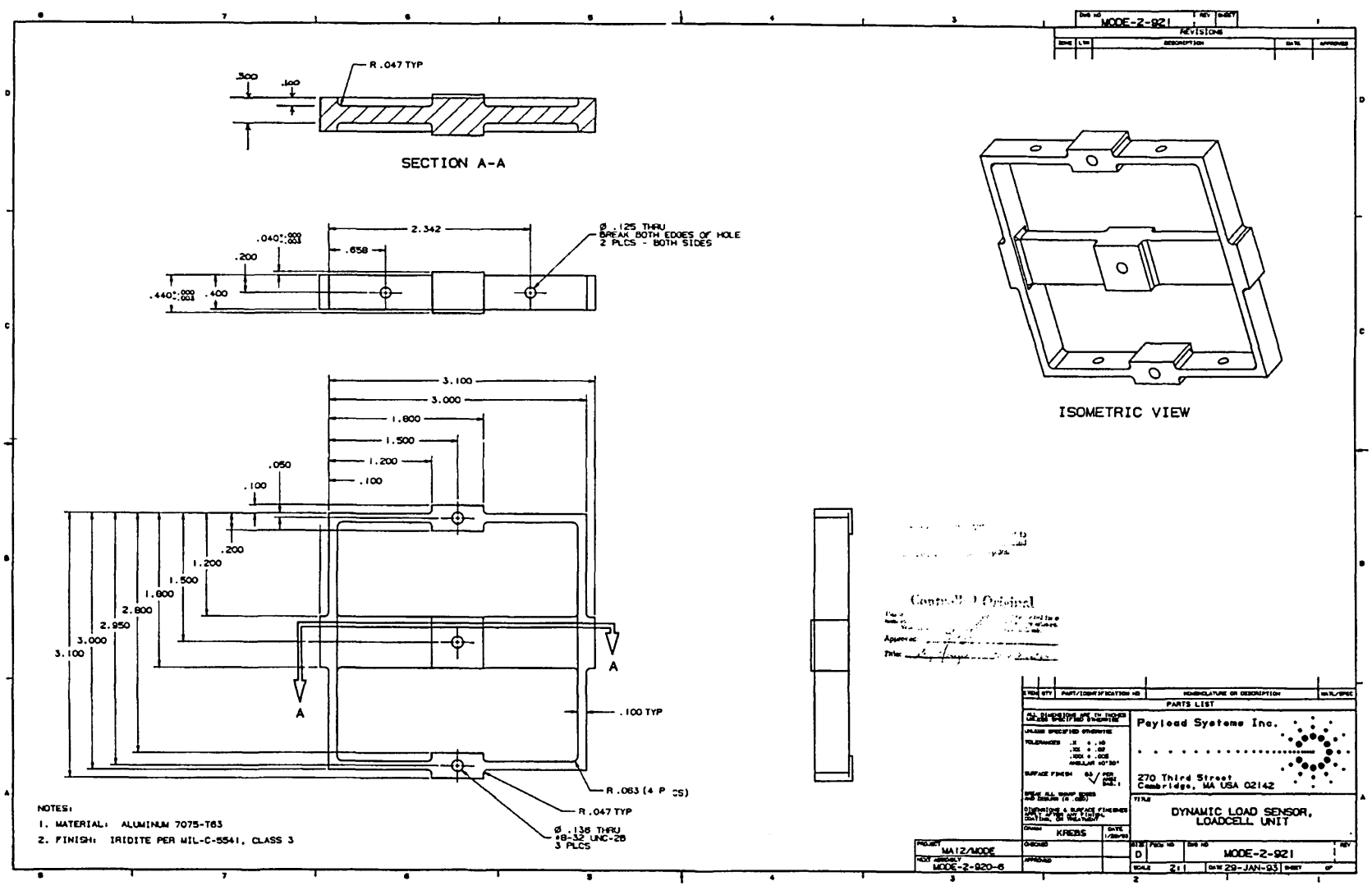
B

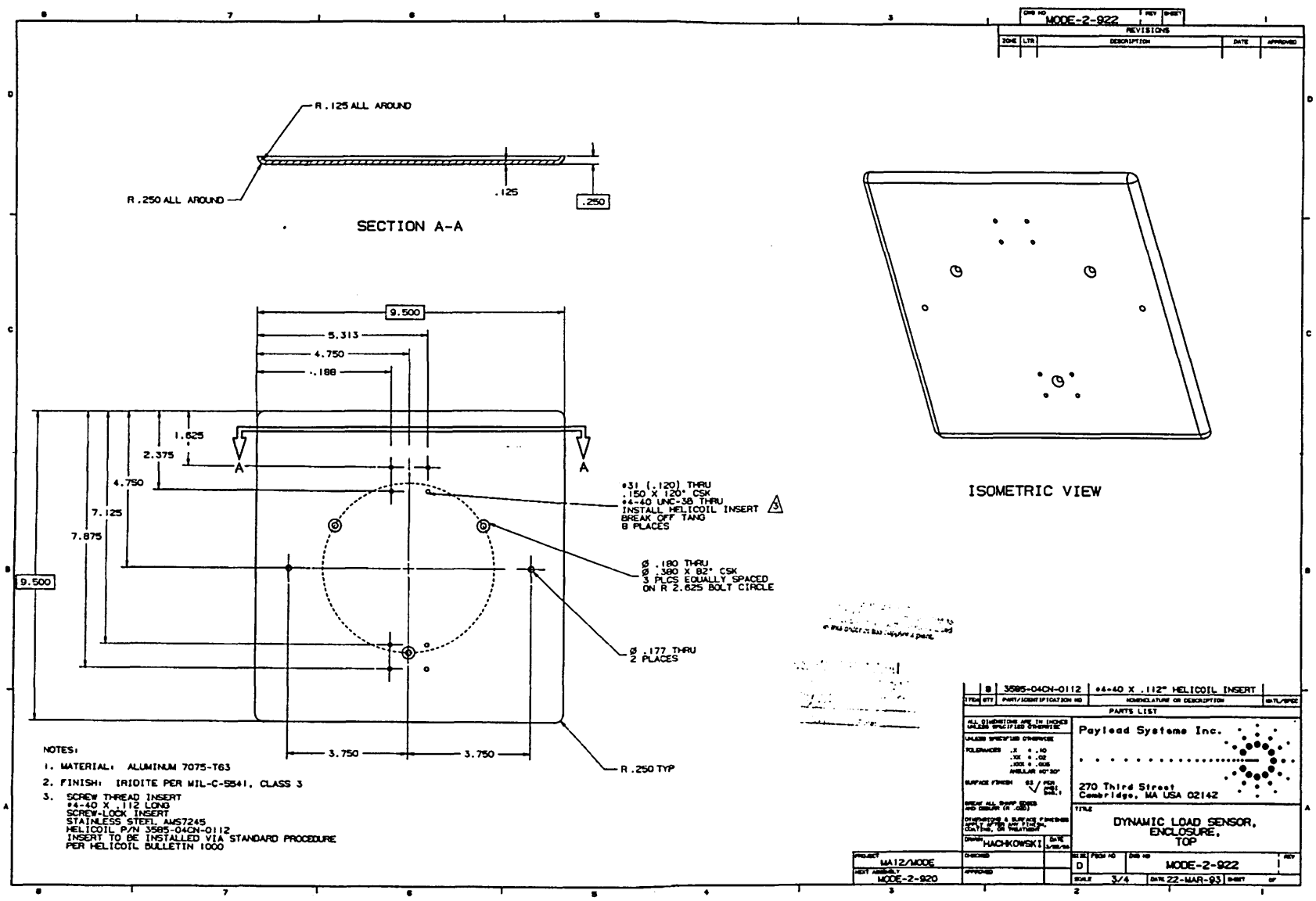
EDLS SENSOR MACHINE SHOP DRAWINGS

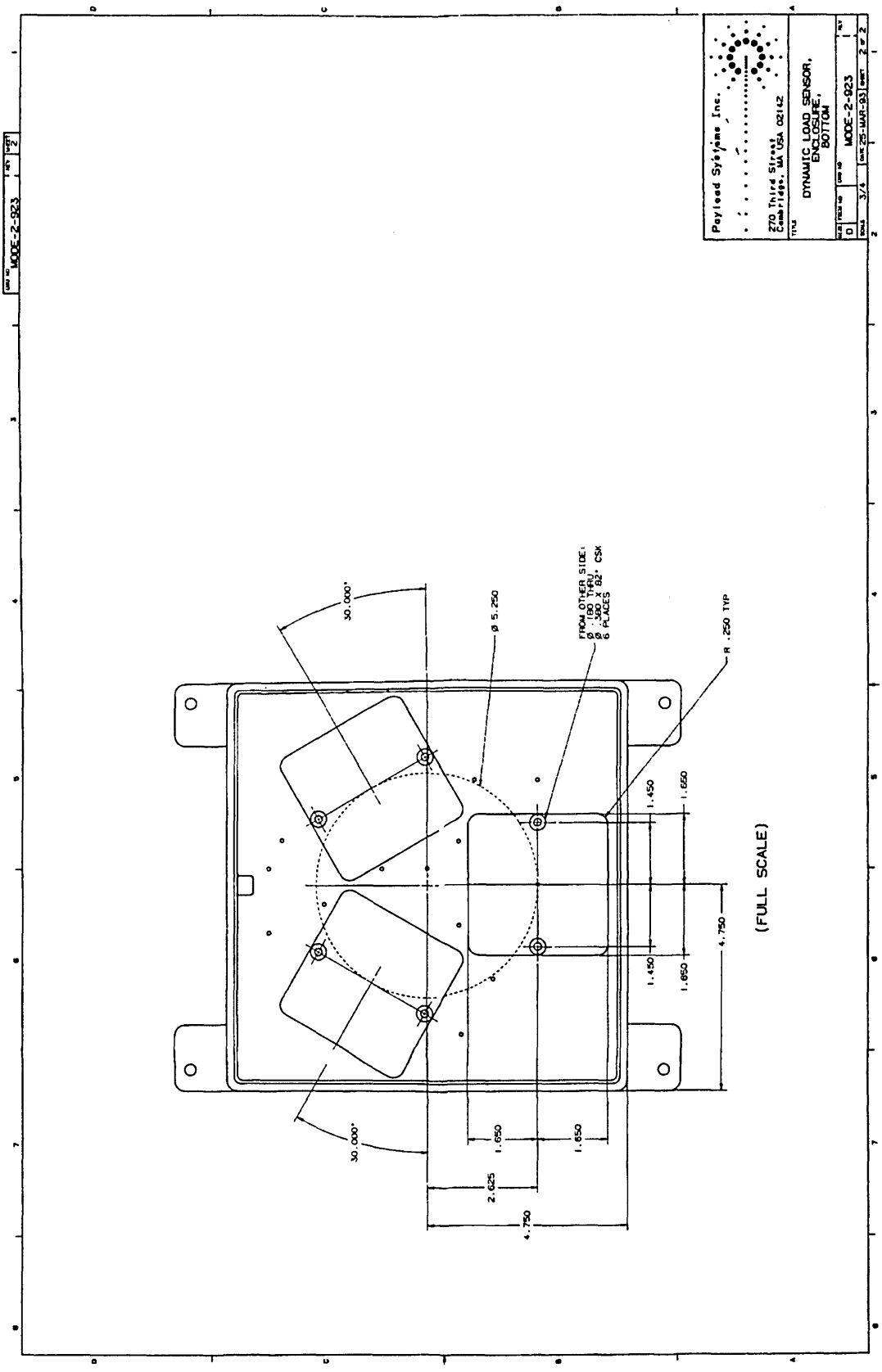
This appendix contains the machine shop drawings used for manufacturing the original DLS / EDLS sensors. The drawings were scanned in electronically at 300 dpi. The imprinted stamp "Uncontrolled Copy" was removed from the scans since there were no changes from the controlled originals.

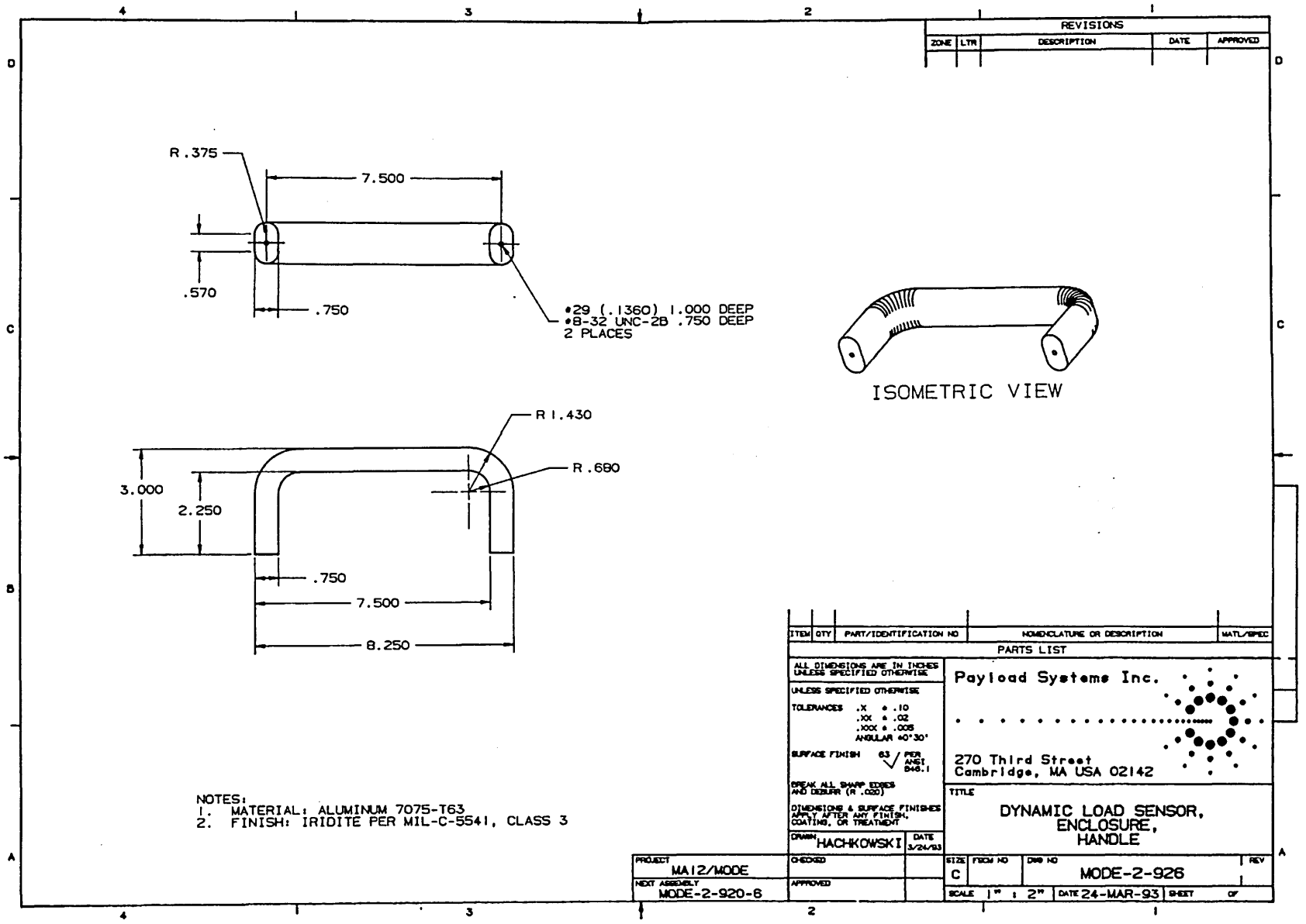
The machine shop drawings for the following components are included:

Part Number	Description
MODE-2-921	Loadcell Unit
MODE-2-922	Enclosure, Top
MODE-2-923	Enclosure, Botton
MODE-2-926	Enclosure, Handle


MODE-2-921
REVISIONS
R .047 TYP
.500
.100
SECTION A-A
2.342
.658
.040
.200
.440
.400
Ø .125 THRU
BREAK BOTH EDGES OF HOLE
2 PLCS - BOTH SIDES
ISOMETRIC VIEW
3.100
3.000
1.800
1.500
1.200
.100
.050
.200
1.200
1.500
1.800
2.800
2.950
3.000
3.100
A
A
.100 TYP
R .063 (4 PLCS)
R .047 TYP
Ø .138 THRU
#8-32 UNC-2B
3 PLCS
NOTES:
1. MATERIAL: ALUMINUM 7075-T63
2. FINISH: IRIDITE PER MIL-C-5541, CLASS 3
PARTS LIST
Payload Systems Inc.
270 Third Street
Cambridge, MA USA 02142
DYNAMIC LOAD SENSOR,
LOADCELL UNIT
KREBS
MA12/MODE
MODE-2-920-6
D
MODE-2-921
2:1
29-JAN-93


R .125 ALL AROUND
R .250 ALL AROUND
.125
.250
SECTION A-A
ISOMETRIC VIEW
9.500
5.313
4.750
.188
1.625
2.375
4.750
7.125
7.875
9.500
3.750
3.750
A
A
#31 (.120) THRU
.150 X 120° CSK
#4-40 UNC-3B THRU
INSTALL HELICOIL INSERT
BREAK OFF TANG
8 PLACES
Ø .180 THRU
Ø .380 X 82° CSK
3 PLCS EQUALLY SPACED
ON R 2.625 BOLT CIRCLE
Ø .177 THRU
2 PLACES
R .250 TYP
NOTES:
1. MATERIAL: ALUMINUM 7075-T63
2. FINISH: IRIDITE PER MIL-C-5541, CLASS 3
3. SCREW THREAD INSERT
#4-40 X .112 LONG
SCREW-LOCK INSERT
STAINLESS STEEL, AMS7245
HELICOIL P/N 3585-04CN-0112
INSERT TO BE INSTALLED VIA STANDARD PROCEDURE
PER HELICOIL BULLETIN 1000
MODE-2-922
REVISIONS
ZONE LTR DESCRIPTION DATE APPROVED
B 3585-04CN-0112 #4-40 X .112" HELICOIL INSERT
ITEM QTY PART/IDENTIFICATION NO NOMENCLATURE OR DESCRIPTION MATL/SPEC
PARTS LIST
Payload Systems Inc.
270 Third Street
Cambridge, MA USA 02142
TITLE
DYNAMIC LOAD SENSOR,
ENCLOSURE,
TOP
DRAWN HACHKOWSKI
PROJECT MA12/MODE
NEXT ASSEMBLY MODE-2-920
SIZE D
DWG NO MODE-2-922
SCALE 3/4
DATE 22-MAR-93


MODE-2-923
Payload Systems Inc.
270 Third Street
Cambridge, MA USA 02142
DYNAMIC LOAD SENSOR, ENCLOSURE, BOTTOM
MODE-2-923
25-MAR-93
30.000°
Ø 5.250
FROM OTHER SIDE:
Ø .180 THRU
Ø .300 X 82° CSK
6 PLACES
R .250 TYP
1.450
1.650
4.750
(FULL SCALE)
30.000°
1.650
1.650
2.625
4.750


R .375
7.500
.570
.750
#29 (.1360) 1.000 DEEP
#8-32 UNC-2B .750 DEEP
2 PLACES
ISOMETRIC VIEW
R 1.430
R .680
3.000
2.250
.750
7.500
8.250
NOTES:
1. MATERIAL: ALUMINUM 7075-T63
2. FINISH: IRIDITE PER MIL-C-5541, CLASS 3
REVISIONS
ZONE
LTR
DESCRIPTION
DATE
APPROVED
ITEM
QTY
PART/IDENTIFICATION NO
NOMENCLATURE OR DESCRIPTION
MATL/SPEC
PARTS LIST
ALL DIMENSIONS ARE IN INCHES UNLESS SPECIFIED OTHERWISE
UNLESS SPECIFIED OTHERWISE
TOLERANCES .X ± .10
.XX ± .02
.XXX ± .005
ANGULAR ±0°30'
SURFACE FINISH 63 PER ANSI B46.1
BREAK ALL SHARP EDGES AND DEBURR (R .020)
DIMENSIONS & SURFACE FINISHES APPLY AFTER ANY FINISH, COATING, OR TREATMENT
DRAWN HACHKOWSKI
DATE 3/24/93
CHECKED
APPROVED
Payload Systems Inc.
270 Third Street
Cambridge, MA USA 02142
TITLE
DYNAMIC LOAD SENSOR, ENCLOSURE, HANDLE
PROJECT MA12/MODE
NEXT ASSEMBLY MODE-2-920-6
SIZE C
FSCM NO
DWG NO MODE-2-926
REV
SCALE 1" : 2"
DATE 24-MAR-93
SHEET
OF

APPENDIX C

MATLAB CODE

C.1 EDLSAP Scripts

All EDLSAP script files are listed in Table C.1. The script files are either *commands* or *functions*. A command is simply called by its file name (without the "m" extension), while functions are called with a one or more paramters and may return values. The EDLSAP script files that are functions have their call syntax included in Table C.1. Since there are 42 routines, only the most important are included in this appendix—their names are set in italics.

Table C.1: EDLSAP Script Files and Function Syntax

Script File Name	Call Syntax of Functions
about_edlsap.m	N/A
analysis_commands.m	`analysis_commands(action)`
clear_edlsap.m	N/A
calibration_menu.m	N/A
convert_seconds.m	`Time_Vector = convert_seconds(Total_Seconds)`
create_calib_matrix.m	N/A
difference_time.m	N/A
difference_value	N/A
edit_axes.m	`edit_axes(action)`
edlsap.m	N/A
edlsap_help.m	`edlsap_help(action)`
edlsap_message.m	`edlsap_message(msg_type,message)`
filter_parameters.m	`filter_parameters(action)`
init_var.m	N/A
load_data.m	`load_data(pathname,filename)`
main_menu.m	`status = main_menu(action)`
main_window.m	`status = main_window(action)`
path_setting.m	N/A

Table C.1: EDLSAP Script Files and Function Syntax *(continued)*

Script File Name	Call Syntax of Functions
plot_calib_results.m	`plot_calib_results(Measured,Applied, Predicted,Predicted_Old,Cases,dof,Title_str)`
plot_commands.m	`plot_commands(action)`
plot_data.m	N/A
plot_menu.m	N/A
plot_text.m	`plot_text(action)`
plot_text_window.m	`plot_text_window(text_string,Font_Align, Font_weight,Font_size,Font_rotation,Text_Position)`
plot_vector_3D.m	`plot_vector_3D(action,sensor)`
plot_window.m	N/A
print_data.m	`print_data(action)`
process_rawdata_masu.m	`process_rawdata_masu(raw_data,Gains, raw_data_filename,file_counter)`
process_rawdata_mode.m	`process_rawdata_mode(raw_data,Gains, raw_data_filename,file_counter)`
progress_msg.m	`progress_msg(action,argument)`
psd_parameters.m	`Result_str = psd_parameters(action,Title_handle)`
rawdata_parameters.m	`rawdata_parameters(action)`
read_rawdata_masu.m	`read_rawdata_masu(pathname,filename, Length_in_Minutes)`
read_rawdata_mode.m	`read_rawdata_mode(pathname,filename, Length_in_Minutes)`
save_data.m	N/A
saveas_data.m	N/A
scale_factor.m	`scale_factor(action)`
select_data.m	N/A
select_rawdata_masu.m	N/A
select_rawdata_mode.m	N/A
set_paths.m	N/A
verify_paths.m	N/A

Script File "analysis_commands"

```
function analysis_commands(action)
% ------------------------------------------------------------------------
% analysis_commands(action)                                  EDLS Analysis Package
%
% Various functions to analyze a data segment.
%
% Copyright (c) 1998 by Massachusetts Institute of Technology
% $Revision: 2.0 $  $Date: 1998/08/27 22:26:00 $
% ------------------------------------------------------------------------

% Share EDLS data variables
globalEDLS_AdOffsets EDLS_Configuration EDLS_Comment EDLS_Data EDLS_Filename ...
       EDLS_Mission EDLS_Offsets EDLS_Protocol EDLS_SourceDisk EDLS_SourceFile ...
      EDLS_SampleRate EDLS_TimeSource EDLS_TimeStamp EDLS_Units ...
       EDLS_DataPSD


% Obtain variables which are needed
vars = ['default_Back |Value '
          'default_Fore |Value '
          'inactive_Fore|Value '
          'highlite_Fore|Value '
          'error_Fore   |Value '
          'edittext_Back|Value '
          'edittext_Fore|Value '];
[l,c] = size(vars);
s = findstr(vars(1,:),'|');
for i = 1:l
   var  = deblank(vars(i,1:s-1));
   type = deblank(vars(i,s+1:length(vars(i,:))));
   eval([var '=get(findobj(''Tag'',''' var '''),''' type ''');']);
end
mainwin_handle = findobj('Name','EDLSAP');
delete(findobj('Tag','Info_Text'));

% Determine data segment
% Note: The values returned by ginput are the x- and y-coordinates
% from the current axes.
[Pos1_X,Pos1_Y] = ginput(1);
PlotAxes = axis;
if (Pos1_Y < PlotAxes(3) | Pos1_Y > PlotAxes(4))
   plot_win_handle = findobj('Name','EDLSAP - Plot Window');

uicontrol(plot_win_handle,'Tag','Info_Text','Style','text','Units','normalized',...
   'Position',[0.01 0.01 0.98 0.02],'String','Canceling
Command','Fore',default_Fore);
   return% If the y-Position of the mouse click is outside the y-limits => Cancel
command
end

Y(1) = PlotAxes(3);
Y(2) = PlotAxes(4);
if Pos1_X < PlotAxes(1)
   Pos1_X = PlotAxes(1);
elseif Pos1_X > PlotAxes(2)
   Pos1_X = PlotAxes(2);
```

```
else
   X(1) = Pos1_X;
   X(2) = Pos1_X;
   handle1 = line(X,Y);
   set(handle1,'Color',[0 0 0]);
end

[Pos2_X,Pos2_Y] = ginput(1);
if Pos2_Y < PlotAxes(3)
   Pos2_Y = PlotAxes(3);
end
if Pos2_Y > PlotAxes(4)
   Pos2_Y = PlotAxes(4);
end
if Pos1_X > Pos2_X
   Start_Data = round(Pos2_X*EDLS_SampleRate);
   End_Data = round(Pos1_X*EDLS_SampleRate);
else
   Start_Data = round(Pos1_X*EDLS_SampleRate);
   End_Data = round(Pos2_X*EDLS_SampleRate);
end
if Start_Data < 1
   Start_Data = 1;
end
[data_rows data_cols] = size(EDLS_Data);
if End_Data > data_rows
   End_Data = data_rows;
end
X(1) = Pos2_X;
X(2) = Pos2_X;
Y(1) = PlotAxes(3);
Y(2) = PlotAxes(4);

handle2 = line(X,Y);
set(handle2,'Color',[0 0 0]);
Signal = str2num(get(gca,'UserData'));
YLabel_handle = get(gca,'YLabel');
Title_handle = get(gca,'Title');

switch action

   % ==========================
   % Start of 'average' action
   % ==========================
   case 'average',
   ave_value = sum(EDLS_Data(Start_Data:End_Data,Signal)) /
length(EDLS_Data(Start_Data:End_Data,Signal));
   Info_text_str = ['Average Value of ' get(YLabel_handle,'String') ...
      ': ' sprintf('%0.5g',ave_value)];

   % ==========================
   % Start of 'minimum' action
   % ==========================
   case 'minimum',
      min_value = min(EDLS_Data(Start_Data:End_Data,Signal));
      Info_text_str = ['Minimum Value of ' get(YLabel_handle,'String') ...
         ': ' sprintf('%0.5g',min_value)];
```

```
    % ==========================
    % Start of 'maximum' action
    % ==========================
    case 'maximum',
        max_value = max(EDLS_Data(Start_Data:End_Data,Signal));
        Info_text_str = ['Maximum Value of ' get(YLabel_handle,'String') ...
            ': ' sprintf('%0.5g',max_value)];

    % ==========================
    % Start of 'avgpower' action
    % ==========================
    case 'avgpower',
        Average_power = 0;
        for i = Start_Data:End_Data
           Average_power = Average_power + ((EDLS_Data(i,Signal))^2);
        end
         Average_power = Average_power / (End_Data - Start_Data);
        if strcmp(get(YLabel_handle,'String'),'Force [N]') == 1
            powerunits = ' N^2';
        else
            powerunits = ' (Nm)^2';
        end
        Info_text_str = ['Average power of signal is: ' ...
            sprintf('%0.5g',Average_power) powerunits];


    % ==========================
    % Start of 'rms' action
    % ==========================
    case 'rms',
        Average_power = 0;
        for i = Start_Data:End_Data
           Average_power = Average_power + ((EDLS_Data(i,Signal))^2);
        end
        Average_power = Average_power / (End_Data - Start_Data);
        RMS_magnitude = sqrt(Average_power);
        Info_text_str = ['The Root-Mean-Square ' get(YLabel_handle,'String') ' is: '
...
            sprintf('%0.5g',RMS_magnitude)];


    % ==========================
    % Start of 'psd' action
    % ==========================
    case 'psd',
       EDLS_DataPSD = EDLS_Data(Start_Data:End_Data,Signal);
       if exist('handle1') == 1
        delete(handle1);
        end
        if exist('handle2') == 1
        delete(handle2);
        end
       psd_parameters('init',Title_handle);
       return

end
```

```
plot_win_handle = findobj('Name','EDLSAP - Plot Window');
uicontrol(plot_win_handle,'Tag','Info_Text','Style','text','Units','normalized',...
   'Position',[0.01 0.01 0.98 0.02],'String',Info_text_str,'Fore',default_Fore);
disp(Info_text_str);

% Delete the data selection markers
if exist('handle1') == 1
   delete(handle1);
end
if exist('handle2') == 1
   delete(handle2);
end
```

Script File "create_calib_matrix"

```
% --------------------------------------------------------------------------------
% create_calib_matrix                                    EDLS Analysis Package
%
% This command selects the tab delimited text file with the calibration data
% and calculates the new calibration matrix. It also compares the result
% with a different calibration matrix.
%
% Copyright (c) 1998 by Massachusetts Institute of Technology
% $Revision: 2.0 $  $Date: 1998/08/27 22:12:00 $
% --------------------------------------------------------------------------------

% Share EDLS data variables
globalEDLS_AdOffsets EDLS_Configuration EDLS_Comment EDLS_Data EDLS_Filename ...
       EDLS_Mission EDLS_Offsets EDLS_Protocol EDLS_SourceDisk EDLS_SourceFile ...
       EDLS_SampleRate EDLS_TimeSource EDLS_TimeStamp EDLS_Units

% Obtain paths needed
rawdata_path = get(findobj('Tag','rawdata_path'),'String');
data_path = get(findobj('Tag','data_path'),'String');

% Load old calibration matrix
[filename,pathname] = uigetfile('*.mat','Please Select A Previous Calibration MAT
File');
if filename == 0 & pathname == 0% If the user presses the Cancel button
   return
elseif upper(filename((length(filename) - 3):length(filename))) == '.MAT'
   fullname = [pathname filename];
   eval(['load ''' fullname '''']);
else
   % File does not have the proper extension
   edlsap_message('error','This file does not have the extension ".mat". Please
select an appropriate file.');
   return
end
if exist('sensor1_cal_matrix') ~= 1
   edlsap_message('error','This is not a valid calibration file!');
   if strcmp(get(findobj('Name','EDLSAP'),'UserData'),'CANCEL') == 1
   return
   else
      create_calib_matrix;;
   end
end

S1CM_old = sensor1_cal_matrix;
S2CM_old = sensor2_cal_matrix;
S3CM_old = sensor3_cal_matrix;
S4CM_old = sensor4_cal_matrix;


% Select and load the calibration Data
% The file must have the extension ".txt"
eval(['cd ' rawdata_path]);
[filename,pathname] = uigetfile('*.TXT','Please Select A New Calibration Text
File');
if filename == 0 & pathname == 0% If the user presses the Cancel button
   return
```

```
elseif upper(filename((length(filename) - 3):length(filename))) == '.TXT'
   fullname = [pathname filename];
   eval(['load ''' fullname '''']);
else
   % File does not have the proper extension
   edlsap_message('error','This file does not have the extension ".TXT". Please
select an appropriate file.');
   return
end


% Extract data from calibration TXT file
eval(['counts_matrix = ' filename(1:length(filename) - 4) ';']);
[n,m] = size(counts_matrix);

S1M = [];
S1A = [];
S2M = [];
S2A = [];
S3M = [];
S3A = [];
S4M = [];
S4A = [];

i1 = 0;
i2 = 0;
i3 = 0;
i4 = 0;

for i = 1:n
   if (counts_matrix(i,1) == 1)
      i1 = i1 + 1;
      S1A(i1,1:6) = counts_matrix(i,2:7);
      S1M(i1,1:6) = counts_matrix(i,8:13);
   end

   if (counts_matrix(i,1) == 2)
      i2 = i2 + 1;
      S2A(i2,1:6) = counts_matrix(i,2:7);
      S2M(i2,1:6) = counts_matrix(i,8:13);
   end

   if (counts_matrix(i,1) == 3)
      i3 = i3 + 1;
      S3A(i3,1:3) = counts_matrix(i,2:4);
      S3M(i3,1:3) = counts_matrix(i,8:10);
   end

   if (counts_matrix(i,1) == 4)
      i4 = i4 + 1;
      S4A(i4,1:6) = counts_matrix(i,2:7);
      S4M(i4,1:6) = counts_matrix(i,8:13);
   end
end
```

```
% Calculate calibration matrix for Sensor #1 (Handhold)
S1M = S1M';
S1A = S1A';
[dummy Cases1] = size(S1M);
sensor1_cal_matrix = (inv(S1M*S1M')*S1M*S1A')';
S1P = sensor1_cal_matrix * S1M;

% Predict loads for Sensor #1 (Handold) using old calibration matrix
Gains = [10 10 10 10 10 10 10 10 10 10];
System_Gains = [5.26 4.45 4.35 4.09 4.72 4.69];
for i = 1:6
   S1M_temp(i,:) = S1M(i,:) * (10 / 65536) * 16 * System_Gains(i) / Gains(i);
end
S1P_old = S1CM_old * S1M_temp;

% Calculate calibration matrix for Sensor #2 (Foot restraint 1)
S2M = S2M';
S2A = S2A';
[dummy Cases2] = size(S2M);
sensor2_cal_matrix = (inv(S2M*S2M')*S2M*S2A')';
S2P = sensor2_cal_matrix*S2M;

% Predict loads for Sensor #2 (Foot restraint 1) using old calibration matrix
Gains = [10 10 10 10 10 10];
System_Gains = [1.83 1.72 1.82 1.72 4.88 4.58];
for i = 1:6
   S2M_temp(i,:) = S2M(i,:) * (10 / 65536) * 16 * System_Gains(i) / Gains(i);
end
S2P_old = S2CM_old * S2M_temp;


% Calculate calibration matrix for Sensor #3 (Touchpad)
S3M = S3M';
S3A = S3A';
[dummy Cases3] = size(S3M);
sensor3_cal_matrix = (inv(S3M*S3M')*S3M*S3A')';
S3P = sensor3_cal_matrix*S3M;

% Predict loads for Sensor #3 (Touchpad) using old calibration matrix
Gains = [10 10 10];
System_Gains = [4.83 4.46 4.20];
for i = 1:3
   S3M_temp(i,:) = S3M(i,:) * (10 / 65536) * 16 * System_Gains(i) / Gains(i);
end
S3P_old = S3CM_old * S3M_temp;


% Calculate calibration matrix for Sensor #4 (Foot restraint 2)
S4M = S4M';
S4A = S4A';
[dummy Cases4] = size(S4M);
sensor4_cal_matrix = (inv(S4M*S4M')*S4M*S4A')';
S4P = sensor4_cal_matrix*S4M;

% Predict loads for Sensor #4 (Foot restraint 2) using old calibration matrix
Gains = [10 10 10 10 10 10];
System_Gains = [5.26 4.45 4.35 4.09 4.72 4.69];
```

```
for i = 1:6
   S4M_temp(i,:) = S4M(i,:) * (10 / 65536) * 16 * System_Gains(i) / Gains(i);
end
S4P_old = S4CM_old * S4M_temp;


% Plot accuracy of new and old calibration matrix
plot_calib_results(S1M,S1A,S1P,S1P_old,Cases1, 6, 'Sensor 1: Handhold');
plot_calib_results(S2M,S2A,S2P,S2P_old,Cases2, 6, 'Sensor 2: Foot restraint 1');
plot_calib_results(S3M,S3A,S3P,S3P_old,Cases3, 3, 'Sensor 3: Touchpad');
plot_calib_results(S4M,S4A,S4P,S4P_old,Cases4, 6, 'Sensor 4: Foot restraint 2');

% Set up FFT and FHT matrices and save matrices in a file
% Convert calibration matrices from [N/counts] into [N/Volts]
% The MASU ESM had +/- 5V and a 16-bit (=65536 counts) A/D board
sensor1_cal_matrix = sensor1_cal_matrix * (65536 / 10)
sensor2_cal_matrix = sensor2_cal_matrix * (65536 / 10)
sensor3_cal_matrix = sensor3_cal_matrix * (65536 / 10)
sensor4_cal_matrix = sensor4_cal_matrix * (65536 / 10)

fft_cal_matrix(1:6,1:6)     = sensor2_cal_matrix;
fft_cal_matrix(7:12,7:12)   = sensor4_cal_matrix;
fft_cal_matrix(13:15,13:15) = sensor3_cal_matrix;

fht_cal_matrix(1:6,1:6)     = sensor2_cal_matrix;
fht_cal_matrix(7:12,7:12)   = sensor1_cal_matrix;
fht_cal_matrix(13:15,13:15) = sensor3_cal_matrix;

eval(['cd ' data_path]);
save cal_matrices fft_cal_matrix fht_cal_matrix sensor1_cal_matrix
sensor2_cal_matrix sensor3_cal_matrix sensor4_cal_matrix

% Close calibration plot Window
delete(findobj('Name','EDLSAP - Calibration'));
```

Script File "process_rawdata_masu"

```
function process_rawdata_masu(raw_data,Gains,raw_data_filename,file_counter)
% --------------------------------------------------------------------------------
%                                                                  EDLS Analysis Package
%
% process_rawdata_MASU(raw_data,Gains,raw_data_filename,file_counter)
%
% Process raw EDLS flight data from the MASU ESM and save it.
%
% Copyright (c) 1998 by Massachusetts Institute of Technology
% $Revision: 2.0 $  $Date: 1998/08/28 01:22:00 $
% --------------------------------------------------------------------------------

% Share EDLS data variables
globalEDLS_AdOffsets EDLS_Configuration EDLS_Comment EDLS_Data EDLS_Filename ...
       EDLS_Mission EDLS_Offsets EDLS_Protocol EDLS_SourceDisk EDLS_SourceFile ...
       EDLS_SampleRate EDLS_TimeSource EDLS_TimeStamp EDLS_Units

data_path = get(findobj('Tag','data_path'),'String');
rawdata_path = get(findobj('Tag','rawdata_path'),'String');
data_temp = [];
EDLS_Configuration = upper(EDLS_Configuration);

% Eliminates the non-EDLS channels and sort the EDLS channels
% in the typical order: Fx,Fy,Fz,Mx,My,Mz, Fx, Fy, etc.
raw_data = raw_data([1 2 17 18 19 20 3 4 8 26 30 31 32 15 16],:);


% Process the data according to the Units selected
if EDLS_Units == 1 | EDLS_Units == 2

   % Throughout the calibration a gain of 10 was used for all
   % EDLS Channels. By dividing by 10, the gain will be 1 for
   % all data files which also had the gain set to 10.
   % This is the case for most if not all files.
   Gains = Gains([1 2 17 18 19 20 3 4 8 26 30 31 32 15 16]) / 10;
   for i = 1:15
   raw_data(i,:) = raw_data(i,:) / Gains(i);
   end

   % Load calibration matrices and multipy raw data
   load masu_cal_matrices
   switch EDLS_Configuration
       case 'FFT'
       EDLS_Data = (fft_cal_matrix * (10 / 65536)) * raw_data;
    case 'FHT'
          EDLS_Data = (fht_cal_matrix * (10 / 65536)) * raw_data;
   otherwise
       edlsap_message('error','Internal Error: Incorrect calibration type!');
   end
end

% Convert from SI to English units
% 1 Newton = 0.22481 lb, 1 Newton-Meter = 8.85075 in-lb
if EDLS_Units == 2
   Conversion_vector = [0.22481 0.22481 0.22481 8.85075 8.85075 8.85075 ...
                              0.22481 0.22481 0.22481 8.85075 8.85075 8.85075...
```

```
                                0.22481 0.22481 0.22481]';
    [rows cols] = size(EDLS_Data);
    for i = 1:cols
        EDLS_Data(:,i) = Conversion_vector .* EDLS_Data(:,i);
    end
end

if EDLS_Units == 3
   EDLS_Data = raw_data;
end

EDLS_Data= EDLS_Data';

% The string for 'Day' must always have 3 digits
% The strings for 'Hour', 'Min', and 'Sec' must always have 2 digits
Day_str = ['00' num2str(EDLS_TimeStamp(1))];
Day_str = Day_str(length(Day_str)-2:length(Day_str));
Hour_str = ['0' num2str(EDLS_TimeStamp(2))];
Hour_str = Hour_str(length(Hour_str)-1:length(Hour_str));
Min_str = ['0' num2str(EDLS_TimeStamp(3))];
Min_str = Min_str(length(Min_str)-1:length(Min_str));
Sec_str = ['0' num2str(EDLS_TimeStamp(4))];
Sec_str = Sec_str(length(Sec_str)-1:length(Sec_str));

% Store the filename of the raw data file
sep_char = '/';
if strcmp(computer,'PCWIN')
   sep_char = '\';
end
if strcmp(computer,'MAC2')
   sep_char = ':';
end
char_occur = findstr(raw_data_filename,sep_char);
if length(char_occur) ~= 0
    final_char_occur = char_occur(length(char_occur));
    EDLS_SourceFile =
raw_data_filename(final_char_occur+1:length(raw_data_filename));
else
   EDLS_SourceFile = raw_data_filename;
end

% Calculate the end time of the data file
[samples dummy]= size(EDLS_Data);
Total_Seconds = samples / EDLS_SampleRate + (EDLS_TimeStamp(5) / (10^6)) +
EDLS_TimeStamp(4) + EDLS_TimeStamp(3)*60 + EDLS_TimeStamp(2)*3600 +
EDLS_TimeStamp(1)*86400;
EDLS_TimeStamp(6:10) = convert_seconds(Total_Seconds);

% Save data file in Matlab format
% NOTE: If the day in the time stamp is 0, then EDLSAP assumes that there was no
valid time stamp
EDLS_Filename = [Day_str '_' Hour_str '_' Min_str '_' Sec_str '.mat'];
if Day_str == '000'
    progress_msg('notimestamp');
    EDLS_Filename = [EDLS_SourceFile(1:length(EDLS_SourceFile)-4) '_' Day_str '_'
Hour_str '_' Min_str '_' Sec_str '.mat'];
end
```

```
eval(['cd ''' data_path '''']);
eval(['save ' EDLS_Filename ' EDLS_AdOffsets EDLS_Configuration EDLS_Comment
EDLS_Data EDLS_Filename EDLS_Mission EDLS_Offsets EDLS_Protocol EDLS_SourceDisk
EDLS_SourceFile EDLS_SampleRate EDLS_TimeSource EDLS_TimeStamp EDLS_Units ']);
eval(['cd ''' rawdata_path '''']);
progress_msg('writefile',EDLS_Filename);
progress_msg('filecounter',num2str(file_counter));


% This is necessary because of the shredder we will cut big files
% into several smaller files and will adjust the time stamp accordingly
EDLS_TimeStamp(1:5) = EDLS_TimeStamp(6:10);
EDLS_TimeStamp(6:10) = 0;
```

Script File "process_rawdata_mode"

```
function process_rawdata_mode(raw_data,Gains,raw_data_filename,file_counter)
% --------------------------------------------------------------------------------
%                                                                  EDLS Analysis Package
%
% process_rawdata_mode(raw_data,Gains,raw_data_filename,file_counter)
%
% Process raw EDLS flight data from the MODE ESM and save it.
%
% Copyright (c) 1998 by Massachusetts Institute of Technology
% $Revision: 2.0 $  $Date: 1998/08/28 01:24:00 $
% --------------------------------------------------------------------------------

% Share EDLS data variables
globalEDLS_AdOffsets EDLS_Configuration EDLS_Comment EDLS_Data EDLS_Filename ...
       EDLS_Mission EDLS_Offsets EDLS_Protocol EDLS_SourceDisk EDLS_SourceFile ...
       EDLS_SampleRate EDLS_TimeSource EDLS_TimeStamp EDLS_Units

data_path = get(findobj('Tag','data_path'),'String');
rawdata_path = get(findobj('Tag','rawdata_path'),'String');
data_temp = [];
EDLS_Configuration = upper(EDLS_Configuration);

% Eliminate the unused channel (Channel # 5)
raw_data = raw_data([1 2 3 4 6 7 8 9 10 11 12 13 14 15 16],:);

% Process the data according to the units selected
if EDLS_Units == 1 | EDLS_Units ==2

   % Eliminate unused channel and apply Gains
   Gains = Gains([1 2 3 4 6 7 8 9 10 11 12 13 14 15 16]);
   for i = 1:15
   raw_data(i,:) = raw_data(i,:) / Gains(i);
   end

   % Load calibration matrices and multipy raw data
   load mode_cal_matrices
   switch EDLS_Configuration
       case 'FFT'
       EDLS_Data = (fft_cal_matrix * (20 / 8192)) * raw_data;
    case 'FHT'
          EDLS_Data = (fht_cal_matrix * (20 / 8192)) * raw_data;
   otherwise
          edlsap_message('error','Internal Error: Incorrect calibration type!');
   end
end

% Convert from SI to English units
% 1 Newton = 0.22481 lb, 1 Newton-Meter = 8.85075 in-lb
if EDLS_Units == 2
   Conversion_vector = [0.22481 0.22481 0.22481 8.85075 8.85075 8.85075 ...
                              0.22481 0.22481 0.22481 8.85075 8.85075 8.85075...
                              0.22481 0.22481 0.22481]';
   [rows cols] = size(EDLS_Data);
   for i=1:cols
      EDLS_Data(:,i) = Conversion_vector .* EDLS_Data(:,i);
   end
```

```
end

if EDLS_Units == 3
   EDLS_Data = raw_data;
end

EDLS_Data = EDLS_Data';

% The string for 'Day' must always have 3 digits
% The strings for 'Hour', 'Min', and 'Sec' must always have 2 digits
Day_str = ['00' num2str(EDLS_TimeStamp(1))];
Day_str = Day_str(length(Day_str)-2:length(Day_str));
Hour_str = ['0' num2str(EDLS_TimeStamp(2))];
Hour_str = Hour_str(length(Hour_str)-1:length(Hour_str));
Min_str = ['0' num2str(EDLS_TimeStamp(3))];
Min_str = Min_str(length(Min_str)-1:length(Min_str));
Sec_str = ['0' num2str(EDLS_TimeStamp(4))];
Sec_str = Sec_str(length(Sec_str)-1:length(Sec_str));

% Store the filename of the raw data file
sep_char = '/';
if strcmp(computer,'PCWIN')
   sep_char = '\';
end
if strcmp(computer,'MAC2')
   sep_char = ':';
end
char_occur = findstr(raw_data_filename,sep_char);
if length(char_occur) ~= 0
   final_char_occur = char_occur(length(char_occur));
   EDLS_SourceFile =
raw_data_filename(final_char_occur+1:length(raw_data_filename));
else
   EDLS_SourceFile = raw_data_filename;
end

% Calculate the end time of the data file
[samples dummy]= size(EDLS_Data);
Total_Seconds = (samples / EDLS_SampleRate) + EDLS_TimeStamp(4) +
EDLS_TimeStamp(3)*60 + EDLS_TimeStamp(2)*3600 + EDLS_TimeStamp(1)*86400;
EDLS_TimeStamp(6:10) = convert_seconds(Total_Seconds);

% Save data file in Matlab format
% NOTE: If the day in the time stamp is 0, then EDLSAP assumes that there was no
valid time stamp
EDLS_Filename = [Day_str '_' Hour_str '_' Min_str '_' Sec_str '.mat'];
if Day_str == '000'
   progress_msg('notimestamp');
   EDLS_Filename = [EDLS_SourceFile(1:length(EDLS_SourceFile)-4) '_' Day_str '_'
Hour_str '_' Min_str '_' Sec_str '.mat'];
end

eval(['cd ''' data_path '''']);
eval(['save ' EDLS_Filename ' EDLS_AdOffsets EDLS_Configuration EDLS_Comment
EDLS_Data EDLS_Filename EDLS_Mission EDLS_Offsets EDLS_Protocol EDLS_SourceDisk
EDLS_SourceFile EDLS_SampleRate EDLS_TimeSource EDLS_TimeStamp EDLS_Units ']);
EDLS_Data = [];
```

```
eval(['cd ''' rawdata_path '''']);
progress_msg('writefile',EDLS_Filename);
progress_msg('filecounter',num2str(file_counter));

% This is necessary because of the shredder we will cut big files
% into several smaller files and will adjust the time stamp accordingly
EDLS_TimeStamp(1:5) = EDLS_TimeStamp(6:10);
EDLS_TimeStamp(6:10) = 0;
```

Script File "psd_parameters"

```
function Result_str = psd_parameters(action,Title_handle)
% ------------------------------------------------------------------------------
% psd_parameters(action)                                    EDLS Analysis Package
%
% Displays a window in which the user is asked to enter
% parameters for computing the power spectral density
%
% Copyright (c) 1998 by Massachusetts Institute of Technology
% $Revision: 2.0 $  $Date: 1998/08/29 12:46:00 $
% ------------------------------------------------------------------------------

% Share EDLS data variables
globalEDLS_AdOffsets EDLS_Configuration EDLS_Comment EDLS_Data EDLS_Filename ...
       EDLS_Mission EDLS_Offsets EDLS_Protocol EDLS_SourceDisk EDLS_SourceFile ...
      EDLS_SampleRate EDLS_TimeSource EDLS_TimeStamp EDLS_Units ...
      EDLS_DataPSD

% Obtain the variables needed
vars = ['default_Back |Value '
          'default_Fore |Value '
          'inactive_Fore|Value '
          'highlite_Fore|Value '
          'error_Fore   |Value '
          'edittext_Back|Value '
          'edittext_Fore|Value '];
[l,c] = size(vars);
s = findstr(vars(1,:),'|');
for i = 1:l
   var  = deblank(vars(i,1:s-1));
   type = deblank(vars(i,s+1:length(vars(i,:))));
   eval([var '=get(findobj(''Tag'',''' var '''),''' type ''');']);
end

set(findobj('Name','EDLSAP'),'UserData','CANCEL');

switch action

   % ==========================
   % Start of 'init' action
   % ==========================
   case 'init',
      Title_str = get(Title_handle,'String');

      % Create the window if it does not exist
      Handle = figure('Name','EDLSAP -
Request','NumberTitle','off','Color',default_Back,...
      'Position',[440 350 370 220],'Resize','off','Menubar','none');

      uicontrol(Handle,'Style','text','Position',[  0 190 370
20],'String','PARAMETERS FOR PSD CALCULATION',...
      'Hori','center','Back',default_Back,'Fore',highlite_Fore);
      uicontrol(Handle,'Style','text','Position',[ 10 167  360  20],'String',['PSD
Plot for ' Title_str],...

'Hori','center','Tag','PSD_Plot_Title','Back',default_Back,'Fore',default_Fore);
```

```
      uicontrol(Handle,'Style','text','Position',[ 10 143  130  20],'String','Plot
PSD below:',...
      'Hori','left','Back',default_Back,'Fore',default_Fore);
      uicontrol(Handle,'Style','text','Position',[205 143  130
20],'String','Hz',...
      'Hori','left','Back',default_Back,'Fore',default_Fore);
      uicontrol(Handle,'Style','text','Position',[ 10 118  130
20],'String','Filter PSD above:',...
      'Hori','left','Back',default_Back,'Fore',default_Fore);
      uicontrol(Handle,'Style','text','Position',[205 118  130
20],'String','Hz',...
      'Hori','left','Back',default_Back,'Fore',default_Fore);
      uicontrol(Handle,'Style','text','Position',[ 10  92  130
20],'String','Frequency below which',...
      'Hori','left','Back',default_Back,'Fore',default_Fore);
      uicontrol(Handle,'Style','text','Position',[205  92 200   20],'String','% of
power ...',...
      'Hori','left','Back',default_Back,'Fore',default_Fore);
      uicontrol(Handle,'Style','text','Position',[ 10  67  130  20],'String','...
starting at',...
      'Hori','left','Back',default_Back,'Fore',default_Fore);
      uicontrol(Handle,'Style','text','Position',[205  67  130  20],'String','Hz is
contained.',...
      'Hori','left','Back',default_Back,'Fore',default_Fore);

      uicontrol(Handle,'Style','edit','Position',[135  145  60
20],'String',(EDLS_SampleRate/2),...

'Hori','left','Tag','Max_Freq_req','Back',edittext_Back,'Fore',edittext_Fore);
      uicontrol(Handle,'Style','edit','Position',[135  120  60
20],'String','30',...

'Hori','left','Tag','Corner_Freq_req','Back',edittext_Back,'Fore',edittext_Fore);
      uicontrol(Handle,'Style','edit','Position',[135  95  60
20],'String','95',...

'Hori','left','Tag','Percent_Power_req','Back',edittext_Back,'Fore',edittext_Fore);
      uicontrol(Handle,'Style','edit','Position',[135  70  60
20],'String','0',...

'Hori','left','Tag','Min_Freq_req','Back',edittext_Back,'Fore',edittext_Fore);

      uicontrol(Handle,'Style','push','String','OK','Position',[ 10 10 80 30],...
      'CallBack','psd_parameters(''OK'');',
'Back',default_Back,'Fore',default_Fore);
      uicontrol(Handle,'Style','push','String','Cancel','Position',[280 10 80
30],...
      'CallBack','psd_parameters(''CANCEL'');','Back',default_Back,'Fore',default_Fore);
      waitfor(Handle);

   % ====================
   % Start of 'OK' action
   % ====================
   case 'OK',
      Title_str = get(findobj('Tag','PSD_Plot_Title'),'String');
      Corner_Freq =round(str2num(get(findobj('Tag','Corner_Freq_req'),'String')));
      Max_Disp_Freq =round(str2num(get(findobj('Tag','Max_Freq_req'),'String')));
```

```
      Min_Freq =round(str2num(get(findobj('Tag','Min_Freq_req'),'String')));
      Percent_Power =
round(str2num(get(findobj('Tag','Percent_Power_req'),'String')));
      if Percent_Power > 100
         Percent_Power = 100;
      end
      if Percent_Power < 1
         Percent_Power = 1;
      end
      if Max_Disp_Freq > (EDLS_SampleRate/2)
         Max_Disp_Freq = (EDLS_SampleRate/2);
      end
      if Max_Disp_Freq < 1
         Max_Disp_Freq = 1;
      end
      if Corner_Freq > (EDLS_SampleRate/2)
         Corner_Freq = (EDLS_SampleRate/2);
      end
      if Corner_Freq < 1
         Corner_Freq = 1;
      end
      if Min_Freq < 0
         Min_Freq = 0;
      end
      if Min_Freq > Corner_Freq
         Min_Freq = Corner_Freq;
      end

      EDLS_DataPSD = EDLS_DataPSD(:);
      N = length(EDLS_DataPSD);
      index = 1:N;
      KMU = norm(boxcar(N))^2; % Normalizing scale factor; asymptotically unbiased

      EDLS_DataPSD_W = boxcar(N).*(EDLS_DataPSD(index));
      p_s_d = abs(fft(EDLS_DataPSD_W,N)).^2;% Compute PSD

      % Select the first half of the frequencies
      if rem(N,2) == 0
         select = (1:N/2+1)';% if N is even
      else
         select = (1:(N+1)/2)';% if N is odd
      end
      p_s_d = p_s_d(select);
      p_s_d = p_s_d / KMU;    % Normalizing PSD
      freq_vector = ((select-1) * 2 / N) * (EDLS_SampleRate / 2);

      % Eliminate power content below the specified Corner Frequency
      freq_index_end = min(find(freq_vector > Corner_Freq));

      % Eliminate any remaining noise floor
      p_s_d = p_s_d - p_s_d(end);

      % Calculating the freq. below which a specified percentage
      % of the power is contained. The default is 95% of PSD
      % after filtering out the power above 30 Hz.
      freq_index_start = min(find(freq_vector >= Min_Freq));
      total_area = trapz(p_s_d(freq_index_start:freq_index_end));
```

```
      for i = freq_index_start:freq_index_end
         partial_area = trapz(p_s_d(freq_index_start:i));
         if (partial_area/total_area) > (Percent_Power/100)
            break
        end
      end

      % Use quadratic interpolation to determine the
      % Percentage area more precisely
      psd_value1 = p_s_d(i-1);
      psd_value2 = p_s_d(i);
      area1 = trapz(p_s_d(freq_index_start:i-1));
      freq1 = freq_vector(i-1);
      freq2 = freq_vector(i);

      area_diff = ((Percent_Power/100) * total_area) - area1;
      Freq_Power = roots([((psd_value2 - psd_value1) / (freq2-freq1))
(2*psd_value1) (-2*area_diff)]);
      if Freq_Power(1) < (freq2 - freq1) & Freq_Power(1) >= 0
         Freq_Power = Freq_Power(1);
      else
         Freq_Power = Freq_Power(2);
      end
      Freq_Power = freq_vector(i-1) + Freq_Power;

      % Find the peak of the PSD plot and where it occurs
      peak_PSD = max(p_s_d);
      peak_PSD_index = find(p_s_d == peak_PSD);
      peak_PSD_freq = freq_vector(peak_PSD_index);

      % Create the PSD Plot Window
      PSD_Plot_handle = findobj('Name','EDLSAP - PSD Plot Window');
      if PSD_Plot_handle > 1
         figure(PSD_Plot_handle);
         clf;
      else
         PSD_Plot_handle = figure('Name','EDLSAP - PSD Plot
Window','NumberTitle','off',...
         'Color',default_Back,'Resize','on','MenuBar','none');
      end
      plot(freq_vector,p_s_d);
      grid on;
      axis([0 Max_Disp_Freq -inf inf]);

      % Determine the appropriate units
      if isempty(findstr(Title_str,'Force')) ~= 1
         if EDLS_Units == 1
            PSD_Units = 'N^2/Hz';
         end
         if EDLS_Units == 2
            PSD_Units = 'lb^2/Hz';
         end
         if EDLS_Units == 3
         PSD_Units = 'counts^2/Hz';
         end
      else
```

```
            if EDLS_Units == 1
               PSD_Units = '(Nm)^2/Hz';
            end
            if EDLS_Units == 2
               PSD_Units = '(lb-in)^2/Hz';
            end
            if EDLS_Units == 3
            PSD_Units = 'counts^2/Hz';
            end
         end

         % Label the plot and output the results in the
         % MATLAB command window
         xlabel('Frequency [Hz]');
         ylabel(['Power Spectral Density [' PSD_Units ']']);
        title_str = get(findobj('Tag','PSD_Plot_Title'),'String');
         title(title_str);

         disp(title_str)
         disp(['The maximum PSD value is ' sprintf('%0.5g',peak_PSD) ' ' PSD_Units ...
         ' at a frequency of approx. ' sprintf('%0.5g',peak_PSD_freq) ' Hz.'])
         disp([sprintf('%0g',Percent_Power) '% of the power is below approx. ' ...
         sprintf('%0.5g',Freq_Power) ' Hz starting at ' sprintf('%0.5g',Min_Freq) '
Hz.'])

         close(findobj('Name','EDLSAP - Request'));
         set(findobj('Name','EDLSAP'),'UserData','OK');
         status = main_window('update');
         figure(PSD_Plot_handle);

      % ============================
      % Start of 'CANCEL' action
      % ============================
      case 'CANCEL',
         Result_str = 'CANCEL';
         close(findobj('Name','EDLSAP - Request'));
         set(findobj('Name','EDLSAP'),'UserData','CANCEL');


      % ============================
      % Start of otherwise action
      % ============================
      otherwise
         edlsap_message('error','An internal EDLSAP error occurred.');
         set(findobj('Name','EDLSAP'),'UserData','CANCEL');
end
```

Script File "read_rawdata_masu"

```
function read_rawdata_masu(pathname,filename,Length_in_Minutes)
% ------------------------------------------------------------------------------
%                                                              EDLS Analysis Package
% read_rawdata_masu(pathname,filename,Length_in_Minutes)
%
% Reads in raw EDLS flight data from the MASU ESM.
%
% Copyright (c) 1998 by Massachusetts Institute of Technology
% $Revision: 2.0 $  $Date: 1998/08/28 02:13:00 $
% ------------------------------------------------------------------------------

% Share EDLS data variables
globalEDLS_AdOffsets EDLS_Configuration EDLS_Comment EDLS_Data EDLS_Filename ...
       EDLS_Mission EDLS_Offsets EDLS_Protocol EDLS_SourceDisk EDLS_SourceFile ...
       EDLS_SampleRate EDLS_TimeSource EDLS_TimeStamp EDLS_Units

% Retrieve paths needed
data_path = get(findobj('Tag','data_path'),'String');
rawdata_path = get(findobj('Tag','rawdata_path'),'String');

if exist(data_path) ~= 7
   edlsap_message('error',['Output directory' data_path 'does not exist!']);
   return
end

Buffer_Counter = 1;
file_counter = 0;
data = [];
data_temp = [];
raw_data = [];
Error_Flag = 0;

fid = 0;
[fid,message] = fopen([pathname filename],'r');
if fid == -1
   File_str = ['File:  "' pathname filename '" '];
   Error_String = [File_str message];
   edlsap_message('error',Error_String);
   return
end


status = fseek(fid,0,'eof');
file_size = ftell(fid);
frewind(fid);
progress_msg('init',file_size);
% Read the entire header including fill bytes
Header_bytes = fread(fid,1024);
Header_str = char(Header_bytes');
Header_str = strrep(Header_str,'Time Src:','Time Src=');
Header_str = strrep(Header_str,char([ 61 0 0]),'=');
Header_str = strrep(Header_str,char([ 61 0]),'=');
Header_str = strrep(Header_str,char([13 10]),char(10));
Header_str = strrep(Header_str,char([10 10]),char(10));
Char1_Pos = findstr(Header_str,'=');
Char2_Pos = findstr(Header_str,char(10));
```

```
if length(Char1_Pos) ~= length(Char2_Pos)
   edlsap_message('error','Errors in the header text.');
   return
end

progress_msg('readfile',filename);
Char2_Pos = [ [0] Char2_Pos];
for i = 1:length(Char1_Pos)
   Marker_str = Header_str(Char2_Pos(i) + 1 : Char1_Pos(i) -1);

   switch Marker_str

      case 'Validity String'
      Validity_str = Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1);

      case 'Protocol Number'
      Protocol = str2num(Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1));

      case 'Sample Rate'
      SampleRate = str2num(Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1));

      case 'Number of Channels'
      NumChannels = str2num(Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1));

      case 'Duration'
      Duration_str = Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1);

      case 'Delay'
      Delay_str = Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1);

      case 'Gains'
      Gains_str = Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1);
       Gains = eval(['[',Gains_str,']']);

      case 'Filter Rolloff'
      FilterRolloff = str2num(Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1));

      case 'Buffer Size'
      BufferSize = str2num(Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1));

      case 'PlusThreshold'
      PlusThreshold = str2num(Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1));

      case 'NegThreshold'
      NegThreshold = str2num(Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1));

      case 'Time Src'
         TimeSource_str = Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1);
         TimeSource_str = deblank(TimeSource_str);

      case 'Time Stamp'
         TimeStamp_str = Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1);
         TimeStamp_str = [TimeStamp_str '000000'];

      case 'Time Offset'
        TimeOffset = str2num(Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1));
```

```
        case 'Offsets'
          Offsets_str= Header_str(Char1_Pos(i) + 1 : Char2_Pos(i+1) - 1);
          Offsets = eval(['[ ', Offsets_str ,' ]']);

        case 'AdOffsets'
        AdOffsets_str = Header_str(Char1_Pos(i) + 2 : Char2_Pos(i+1) - 1);
        AdOffsets = eval(['[', AdOffsets_str ,']']);
    end

end


% Extract EDLS related offsets
EDLS_Offsets = Offsets([1 2 17 18 19 20 3 4 8 26 30 31 32 14 15]);
EDLS_AdOffsets= AdOffsets([1 2 17 18 19 20 3 4 8 26 30 31 32 14 15]);

EDLS_TimeStamp(1) = str2num(TimeStamp_str(1:3));% Days
EDLS_TimeStamp(2) = str2num(TimeStamp_str(5:6));% Hours
EDLS_TimeStamp(3) = str2num(TimeStamp_str(8:9));% Minutes
EDLS_TimeStamp(4) = str2num(TimeStamp_str(11:12));% Seconds
EDLS_TimeStamp(5) = str2num(TimeStamp_str(14:19));% Microseconds

Delay_seconds = str2num(Delay_str(1:2))*3600 + str2num(Delay_str(3:4))*60 +
str2num(Delay_str(3:4));
if isempty(Delay_seconds) == 1
    Delay_seconds = 0;
end
Total_Seconds = (EDLS_TimeStamp(5) / (10^6)) + EDLS_TimeStamp(4) +
EDLS_TimeStamp(3)*60 + EDLS_TimeStamp(2)*3600 + EDLS_TimeStamp(1)*86400;
Total_Seconds = Total_Seconds + Delay_seconds + TimeOffset;
EDLS_TimeStamp(1:5) = convert_seconds(Total_Seconds);

% Set the global data variables
EDLS_Protocol = Protocol;
EDLS_SampleRate = SampleRate;
EDLS_TimeSource = TimeSource_str;


% Read buffers until the end of the file
while ftell(fid) < file_size
    progress_msg('percentage',sprintf('%0.3g', (ftell(fid)*100/file_size)));
    % Read tag of buffer
    tag = fread(fid,NumChannels*2);
    tag_num = tag(1) + tag(2)*(2^8) + tag(3)*(2^16) + tag(4)*(2^24);
    if tag_num > hex2dec('7FFFFFFF'),
       tag_num = -(hex2dec('FFFFFFFF') - tag_num + 1);
        words_to_read = abs(tag_num);
    else
        words_to_read = (BufferSize/2 - NumChannels);
    end

    if strcmp(computer,'MAC2') == 1
       bytes_read = fread(fid,[2 words_to_read]);
       reverse_bytes = [];
       reverse_bytes(1,:) = bytes_read(2,:);
       reverse_bytes(2,:) = bytes_read(1,:);
       eval(['cd ''' data_path '''']);
```

```
      [fid_temp,message] = fopen(['EDLSAPtemp.dat'],'w');
      if fid_temp == -1
         edlsap_message('error',message);
      end
      counter = fwrite(fid_temp,reverse_bytes);
      fclose(fid_temp);
      [fid_temp,message] = fopen(['EDLSAPtemp.dat'],'r');
      if fid_temp == -1
         edlsap_message('error',message);
      end
      words_read = fread(fid_temp,[NumChannels words_to_read/NumChannels],'int16');
       raw_data = [raw_data words_read];
      fclose(fid_temp);
      delete('EDLSAPtemp.dat');
   else
      words_read = fread(fid,[NumChannels words_to_read/NumChannels],'int16');
      raw_data = [raw_data words_read];
   end

   [RD_rows RD_columns] = size(raw_data);
   if RD_columns >= (Length_in_Minutes*60*EDLS_SampleRate)
      file_counter = file_counter + 1;
      process_rawdata_masu(raw_data,Gains,filename,file_counter);
      raw_data = [];
   end
end


% Process the raw data
file_counter = file_counter + 1;
process_rawdata_masu(raw_data,Gains,filename,file_counter);
raw_data = [];
eval(['cd ''' data_path '''']);
progress_msg('close',num2str(fid));
```

Script File "read_rawdata_mode"

```
function read_rawdata_mode(pathname,filename,Length_in_Minutes)
% ------------------------------------------------------------------------
%                                                          EDLS Analysis Package
% read_rawdata_mode(pathname,filename,Length_in_Minutes)
%
% Reads in raw EDLS flight data from the MODE ESM.
%
% Copyright (c) 1998 by Massachusetts Institute of Technology
% $Revision: 2.0 $  $Date: 1998/08/28 02:14:00 $
% ------------------------------------------------------------------------

% Share EDLS data variables
globalEDLS_AdOffsets EDLS_Configuration EDLS_Comment EDLS_Data EDLS_Filename ...
       EDLS_Mission EDLS_Offsets EDLS_Protocol EDLS_SourceDisk EDLS_SourceFile ...
       EDLS_SampleRate EDLS_TimeSource EDLS_TimeStamp EDLS_Units

% Retrive paths needed
data_path = get(findobj('Tag','data_path'),'String');
rawdata_path = get(findobj('Tag','rawdata_path'),'String');

if exist(data_path) ~= 7
   edlsap_message('error',['Output directory' data_path 'does not exist!']);
   return
end

Buffer_Counter = 1;
file_counter = 0;
data = [];
data_temp = [];
raw_data = [];

% Open an EDLS header file (with extension .HDR)
fid = 0;
[fid,message] = fopen([pathname headerfilename],'r');
if fid == -1
   File_str = ['File:  "' pathname headerfilename '" '];
   Error_String = [File_str message];
   edlsap_message('error',Error_String);
   return
end

Validity_Bytes= fread(fid,15);
Validity_String = char(Validity_Bytes');

if Validity_String ~= 'Payload Systems'
   edlsap_message('Problems reading the raw data header file. Either the file is
corrupt a wrong file has been selected.');
   return
end


% Read the time stamp and put into a global variable
% The last character has to be a period.
status = fseek(fid,90,'bof');
TimeStamp_bytes= fread(fid,13);
if max(TimeStamp_bytes) > 58 | min(TimeStamp_bytes) < 46;
```

```
   EDLS_TimeStamp = [0 0 0 0 0];
else
   TimeStamp_str = char(TimeStamp_bytes');
   EDLS_TimeStamp(1) = str2num(TimeStamp_str(1:3));% Days
   EDLS_TimeStamp(2) = str2num(TimeStamp_str(5:6));% Hours
   EDLS_TimeStamp(3) = str2num(TimeStamp_str(8:9));% Minutes
   EDLS_TimeStamp(4) = str2num(TimeStamp_str(11:12));% Seconds
   EDLS_TimeStamp(5) = 0;            % Microseconds
end


% Read gain indices and calculate gains
status = fseek(fid,115,'bof');
Gain_Index = fread(fid,16);
Gains = 2.^Gain_Index;
if (Gain_Index < 1) | (Gain_Index > 4)
   edlsap_message('error','An incorrect Gain Index was read in the file.');
   return
end


% Close header file
fclose(fid);

% Open data file (with extension .DAT)
datafilename = [headerfilename(1:length(headerfilename)-3) 'DAT'];

fid = 0;
[fid,message] = fopen([pathname datafilename],'r');
if fid == -1
   File_str = ['File:  "' pathname datafilename '"' blanks(100)];
   message = [message blanks(100)];
   Str_lengths(1) = length(File_str);
   Str_lengths(2) = length(message);
   MLen = max(Str_lengths)-100;
   Error_String = [File_str(1:MLen); message(1:MLen)];
   edlsap_message('error',Error_String);
   return
end

status = fseek(fid,0,'eof');
file_size = ftell(fid);
frewind(fid);
progress_msg('init',file_size);
progress_msg('readfile',datafilename);

% Set the global data variables
EDLS_Protocol = 0;
EDLS_SampleRate = 250;
EDLS_TimeSource = 'unknown';

% Set parameters
Total_Samples = file_size/32;
Samples_Read = 0;
Samples_to_Read = Length_in_Minutes*60*EDLS_SampleRate;
if Samples_to_Read > 10000
   Samples_to_Read = 15000;% Note: 1 min. * 60 sec/min * 250 Hz=15000
```

```
end
if Samples_to_Read > Total_Samples
   Samples_to_Read = Total_Samples;
end

% Read in data
while Samples_Read < Total_Samples
   progress_msg('percentage',sprintf('%0.3g', (ftell(fid)*100/file_size)));

   if strcmp(computer,'MAC2') == 1
   bytes_read = fread(fid,[2 16*Samples_to_Read]);
      reverse_bytes = [];
      reverse_bytes(1,:) = bytes_read(2,:);
      reverse_bytes(2,:) = bytes_read(1,:);
      eval(['cd ''' data_path '''']);
     [fid_temp,message] = fopen(['EDLSAPtemp.dat'],'w');
     if fid_temp == -1
     edlsap_message('error',message);
     end
      counter = fwrite(fid_temp,reverse_bytes);
      fclose(fid_temp);
      [fid_temp,message] = fopen(['EDLSAPtemp.dat'],'r');
      if fid_temp == -1
         edlsap_message('error',message);
      end
      words_read = fread(fid_temp,[16 Samples_to_Read],'int16');
      raw_data = [raw_data words_read];
     fclose(fid_temp);
     delete('EDLStemp,dat');
   else
     words_read = fread(fid,[16 Samples_to_Read],'int16');
      raw_data = [raw_data words_read];
   end

   [RD_rows RD_columns] = size(raw_data);
   if RD_columns >= (Length_in_Minutes*60*EDLS_SampleRate)
      file_counter = file_counter + 1;
      process_rawdata_mode(raw_data,Gains,datafilename,file_counter);
      raw_data = [];
   end
   Samples_Read = Samples_Read + Samples_to_Read;
   if (Samples_Read + Samples_to_Read) > Total_Samples
      Samples_to_Read = Total_Samples - Samples_Read;
   end
end

% Process the raw data
file_counter = file_counter + 1;
Gains = Gains';
process_rawdata_mode(raw_data,Gains,datafilename,file_counter);
raw_data = [];
eval(['cd ''' data_path '''']);
progress_msg('close',num2str(fid));
```

C.2 DLS to EDLSAP Conversion Scripts

The data collected during the DLS experiment on Mission STS-62 was stored in MATLAB files. However, its data format was different. The variables contained in these files and their meaning is explained in Table C.2.

Table C.2:

Matlab Variable	Explanation
s1	The force component in the x-direction of the foot restraint.
s2	The moment component in the x-direction of the foot restraint.
s3	The force component in the y-direction of the foot restraint.
s4	The moment component in the y-direction of the foot restraint.
s5	The force component in the z-direction of the foot restraint.
s6	The moment component in the z-direction of the foot restraint.
s7	The force component in the x-direction of the hand hold.
s8	The moment component in the x-direction of the hand hold.
s9	The force component in the y-direction of the hand hold.
s10	The moment component in the y-direction of the hand hold.
s11	The force component in the z-direction of the hand hold.
s12	The moment component in the z-direction of the hand hold.
s13	The force component in the x-direction of the touchpad.
s14	The force component in the y-direction of the touchpad.
s15	The force component in the z-direction of the touchpad.
days	A numerical variable containing the MET day the raw data was recorded.
hours	A numerical variable containing the MET hour the raw data was recorded.
minutes	A numerical variable containing the MET minutes the raw data was recorded.
seconds	A numerical variable containing the MET seconds the data was recorded.
t	A vector whose length is equal to the number of sample points. The time vector's first element is 0 and every successive element is incremented by 0.004 since the sampling frequency was 250 Hz.
tfile	The timestamp contained in the variables days, hours, minutes, seconds is that of the raw data file. The processed data file is usually a portion of the raw data file. The first element of the time vector tfile contains the offset in seconds between the timestamp and the actual time the data in the file was recorded. All the remaining elements of vector are increments by 0.004 seconds.
good_noise	

Two MATLAB scripts were written to convert the DLS data into a format readable by EDLSAP. These routines are printed on the next pages.

Script File "DLS_to_EDLSAP_Converter.m"

```
% ------------------------------------------------------------
% This MATLAB script converts processed DLS data
% files into a format readable by EDLSAP
% Written by Amir R. Amir
% Copyright (c) 1998 by Massachusetts Institute of Technology
% $Revision: 1.0 $  $Date: 1998/08/12 15:30:00 $
% ------------------------------------------------------------

clear
clc

disp('This program converts DLS data files into a format readable by EDLSAP.')
disp('Written by Amir R. Amir. Version 1.0')
disp(' ')
disp('The current directory is:')
cd
disp('The directory contents is:')
dir
filename = input('Please enter the file name or "*" to process all DLS files:
',’s’);

if filename == '*'
   action = 'all';
elseif exist(filename) == 2
   action = 'single';
else
   action = 'error';
end

switch action

   case 'single'
    convert_dls(filename)


   case 'all'
   allfiles = what;
   matfiles = getfield(allfiles,'mat');
   for i=1:length(matfiles)
      filename = char(matfiles(i))
      convert_dls(filename);
   end


   case 'error'
   disp('Sorry. An error occurred. Please press a key to try again.')

end
```

Script File "convert_dls.m"

```
function convert_dls(filename)
clear s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 s14 s15
clear t tfile days hours minutes seconds
clear EDLS_AdOffsets EDLS_Configuration EDLS_Comment EDLS_Data EDLS_Filename
EDLS_Mission
clear EDLS_Offsets EDLS_Protocol EDLS_SourceDisk EDLS_SourceFile EDLS_SampleRate
clear EDLS_TimeSource EDLS_TimeStamp EDLS_Units

eval(['load ' filename]);

if exist('s1') ~= 1
   disp('Invalid DLS file')
   return
end


EDLS_AdOffsets = zeros(1,15);
EDLS_Configuration = 'FHT';
EDLS_Comment = 'Data from Flight Day 7  on Video 19.';
Samples_Count = length(s1);
EDLS_Data = zeros(Samples_Count,15);
EDLS_Data(:, 1) = s1;% Sensor FR, Force x
EDLS_Data(:, 2) = s3;% Sensor FR, Force y
EDLS_Data(:, 3) = s5;% Sensor FR, Force z
EDLS_Data(:, 4) = s2;% Sensor FR, Torque x
EDLS_Data(:, 5) = s4;% Sensor FR, Torque y
EDLS_Data(:, 6) = s6;% Sensor FR, Torque z

EDLS_Data(:, 7) = s7;% Sensor HH, Force x
EDLS_Data(:, 8) = s9;% Sensor HH, Force y
EDLS_Data(:, 9) = s11;% Sensor HH, Force z
EDLS_Data(:,10) = s8;% Sensor HH, Torque x
EDLS_Data(:,11) = s10;% Sensor HH, Torque y
EDLS_Data(:,12) = s12;% Sensor HH, Torque z

EDLS_Data(:,13) = s13;% Sensor TP, Force x
EDLS_Data(:,14) = s14;% Sensor TP, Force y
EDLS_Data(:,15) = s15;% Sensor TP, Force z

name_len = length(filename);
EDLS_Filename = filename(1:name_len-4);
EDLS_Mission = 'DLS on STS-62';
EDLS_Offsets = zeros(1,15);
EDLS_Protocol = 'Unknown';
EDLS_SampleRate = 1/(t(2) - t(1));
EDLS_SourceDisk = 'Unknown';
EDLS_SourceFile = 'Unknown';
EDLS_TimeSource = 'Unknown';

% Calculate the start and end time
% NOTE: Using an EDLSAP routine for that !!
Total_seconds = days*86400 + hours*3600 + minutes*60 + seconds + tfile(1);
EDLS_TimeStamp(1:5) = convert_seconds(Total_seconds);
EDLS_TimeStamp(6:10) = convert_seconds(Total_seconds + Samples_Count/
EDLS_SampleRate);
```

```
EDLS_Units = 1;
fullname = [EDLS_Filename];
eval(['save ''' fullname ''' EDLS_AdOffsets EDLS_Configuration EDLS_Comment
EDLS_Data EDLS_Filename EDLS_Mission EDLS_Offsets EDLS_Protocol EDLS_SourceDisk
EDLS_SourceFile EDLS_SampleRate EDLS_TimeSource EDLS_TimeStamp EDLS_Units ']);
```