An Investigation of Space Suit Mobility with Applications to EVA Operations

by

Patricia Barrett Schmidt

S.B. Massachusetts Institute of Technology, 1996

S.M. Massachusetts Institute of Technology, 1998

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

Doctor of Philosophy in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August, 2001

Author .....
Department of Aeronautics and Astronautics
August 10, 2001

Certified by .....
Dava J. Newman, Thesis Committee Chair
Associate Professor of Aeronautics and Astronautics
MacVicar Faculty Fellow

Certified by .....
Laurence R. Young
Apollo Program Professor of Astronautics

Certified by .....
R. John Hansman
Professor of Aeronautics and Astronautics

Certified by .....
Carlos E. S. Cesnik
Associate Professor of Aeronautics and Astronautics

Accepted by .....
Wallace E. Vander Velde
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

An Investigation of Space Suit Mobility with Applications to EVA Operations
by

Patricia Schmidt

Submitted to the Department of Aeronautics and Astronautics on
August 10, 2001, in partial fulfillment of the requirements for the
degree of Doctor of Philosophy in Aeronautics and Astronautics

Abstract

The primary aim of this thesis is to advance the current understanding of astronauts' capabilities and limitations in space-suited extravehicular activity (EVA) by compiling a detailed database of the torques needed to bend the joints of a space suit, developing models of the mechanics of space suit joints based on experimental data, and utilizing these models to estimate a human factors performance metric, the work envelope for space suited EVA work. A detailed space suit joint torque-angle database is compiled in a novel experimental approach that uses space-suited human test subjects to generate realistic, multi-joint motions, which are used to drive an instrumented robot to measure the torques required to accomplish the motions in a pressurized space suit. Based on the experimental data, a mathematical model using the Preisach hysteresis modeling technique is developed to predict joint torque from the joint angle history. Two physics-based models describing the bending load-deflection characteristics of pressurized fabric cylinders were compared to the experimental space suit data. The beam model assumes that bending deflections are completely attributable to elongation of the fabric cylinder wall, while the membrane model assumes that the fabric never stretches. The experimental data corresponds closely with the membrane model, implying that space suit joint stiffness is primarily determined by volume changes as the joint bends and the resulting compression of the gas inside the space suit. The space suit models were applied in a computational work envelope analysis to determine the volume in which a space-suited astronaut can comfortably work. A new method that uses inverse kinematics and the space suit model to calculate a work envelope based on visibility constraints and human strength limits is developed. Sensitivity analysis of the work envelope indicates that improving shoulder mobility and upward and downward visibility enlarge the space-suited work envelope.

Thesis Supervisor: Dava J. Newman
Associate Professor of Aeronautics and Astronautics
MacVicar Faculty Fellow

Acknowledgements

First of all, I would like to thank my thesis committee: Dava Newman, Larry Young, John Hansman and Carlos Cesnik, for their efforts in helping me to plan and carry out this work. I would particularly like to thank Dava Newman for her support and mentoring over the years and for giving me the opportunity to work on an exciting, fun project. Special thanks to Ed Hodgson for all of his help in making this project work and his thorough reading of my thesis. Thanks to Walter Hollister for supplying the outside perspective at my general exams and my defense and to Jeff Hoffman for his helpful comments on my proposal.

A big thank you to Joe Saleh and Chris Carr for their help in preparing for my defense and for providing much-needed distractions. Amir Amir's help in the early stages of this project was essential. Best wishes to all of the MVL folks who helped me out and tolerated all of my noisy equipment. I would like to thank all of the UROP students who helped make this work possible: Alan Chen, Kamla Topsey, Dan Craig, Leah Soffer, and Blake Gottlieb. Thanks to Dick Perdichizzi, Paul Bauer, and Don Weiner for technical help and to Jennie Leith, Liz Zotos and Marsha Warren for administrative help.

Thanks to Joe Kosmo and Amy Ross, who made this research possible by their generous loan of the Robotic Space Suit Tester and to the engineers at Sarcos for giving me opportunities to develop my robot repair skills.

Finally, I would like to thank my family and friends for their support over the years, which made the grad school experience tolerable and even fun. Mom and Dad, Julie and Harry, you're going to have to go to one more graduation, but this is the last one, I promise! Thanks to Team Fresh, the Barrel of Monkeys and all the NH4 folks for all the fun times.

This work was supported by NASA grant NAG9-1089.

Table of Contents

List of Figures

Chapter 1 Introduction

Chapter 2 Background

Chapter 3 Space suit mobility database

Chapter 4 Modeling

List of Tables

Chapter 1 *Introduction*

1.1 Motivation

Astronauts and cosmonauts have made great accomplishments while working outside of their spacecraft, assembling and maintaining space stations, capturing and repairing satellites and even exploring the Moon. When astronauts exit their spacecraft to perform extravehicular activities, or EVA's, these operations carry extremely high costs in time, money, risks to personnel, and limited opportunities. Because of the high costs of EVA and the importance of accomplishing planned objectives, months of prior planning and hundreds of hours of rehearsal are required to prepare for each EVA. Planning for EVA's is complicated by the fact that it is not possible to exactly replicate the microgravity, vacuum environment in a single simulation environment on the ground. Thus, successful planning for EVA's requires accurate knowledge of the EVA environment and the capabilities of an astronaut wearing a space suit.

Since the first EVA's were performed by Alexei Leonov and Ed White in 1965, the capabilities of astronauts to do useful work outside of their spacecraft have steadily increased. Likewise, our understanding of EVA astronauts' capabilities and limitations have also progressed through inflight experience, experimentation in neutral buoyancy facilities and parabolic flight, and engineering tests of space suits and EVA tools. The purpose of this thesis is to further advance the current understanding of astronauts' capabilities and limitations in space-suited EVA.

The most important aspect of an EVA astronaut's capabilities is the ability to move his or her body while wearing the space suit. In every EVA scenario, astronauts physically manipulate objects to accomplish tasks. Two factors make these physical interactions strikingly different from those performed on the ground. First, the microgravity environment requires an astronaut to restrain his or her body in order to exert forces and moments on another object, and second, space suits constrain astronauts' body motions in significant and complicated ways.

Historically, feasible limits on planned EVA activities have been determined based on ground experimentation and in-flight experience. Data have been obtained experimentally on operational performance metrics such as joint ranges of motion, the volume in space that a space-suited astronaut can reach, known as the reach envelope, the subset of the

reach envelope in which a space-suited astronaut can comfortably work, known as the work envelope, and the strength of a suited astronaut. These performance measures are related to each other through the constitutive and compatibility relations that govern the space suit's behavior, but isolated experimental data does not allow one performance measure to be predicted from another or the same performance measure to be generalized to a different person or situation. Figure 1-1 depicts the human factors metrics in schematic form: performance measures are related centrally through the space suit's fundamental relations, but cross-links between performance measures do not currently exist.

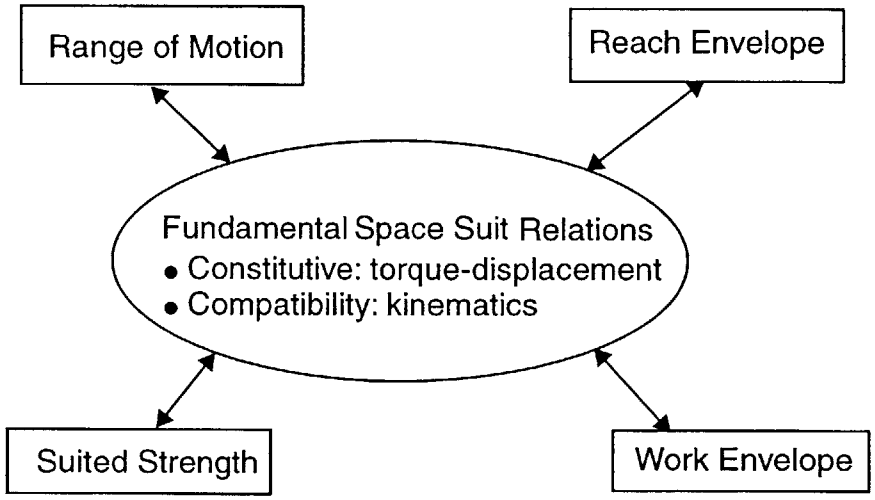


Figure 1-1. Schematic representation of the relationship between space suit mobility related human factors performance metrics.

The purpose of this thesis is to unify and generalize space suit mobility performance measures by modeling the fundamental relations that govern space suit performance, specifically, the constitutive relations between joint displacements and applied torques and the compatibility relations determined by the space suit joint geometry and kinematics. Understanding the fundamental relationships governing space suit performance provides a basis for predicting a global performance measure using information about another performance measure, for instance, predicting a reach envelope using range of motion data. This insight also allows performance measures to be generalized to alternative scenarios, such as determining work envelopes for different-sized crew members.

1.2 Previous research

The EMU space suit, which is currently used for NASA EVA's, has both hard fiberglass and soft fabric components. Mobility features, such as pleats that open as joints bend and rotational bearings, are built into all modern space suits. Without these mobility features, a person in a space suit would be virtually immobile. Even though space suits are designed to allow mobility, they restrict the wearer's motion in significant and complicated ways.[32]

Limited space suit joint torque-angle data has been reported in the literature. Studies that used human subjects wearing space suits reported higher torques than studies that measured torques on joints of empty, pressurized space suits. Higher torques with human subjects may be expected because contact between the space suit and the wearer's body affects the deformed shape of the space suit, but it is unclear whether the discrepancy in experimental results is due to shortcomings in experimental methods or actual differences in observed torques.[3, 15, 40, 41]

This thesis uses several mathematical and physical modeling techniques which were originally developed for other applications but are relevant to modeling the torque-angle characteristics of space suit joints. Mathematical modeling techniques allow torques to be predicted from angle histories with accuracy, but, since they do not incorporate physical principles, they do not contribute insights for designing space suit joints with better mobility characteristics. Three mathematical techniques for modeling hysteretic systems, the Krasnoselski-Pokrovski model[33], the Preisach model,[39] and the Tao and Kokotovic model[47], which were originally used for modeling magnetization and shape memory alloy actuators, were evaluated for possible use. The Preisach model was chosen because it can produce output curves similar in shape to the space suit torque vs. angle curves and its identification process is relatively simple.

Physical models, in contrast to mathematical models, can lead to insights into the physical processes that govern space suit mobility. Two physical models of the bending characteristics of inflated cylinders may be relevant to space suit joints. The beam model treats the inflated cylinder as a beam with a fabric wall that stretches, maintaining a constant internal volume. The mobility of the space suit joint is determined by elastic behavior of the fabric wall.[35-38] The membrane model treats the fabric shell as an inextensible membrane. Bending deflections of the cylinder result in shape and volume changes. The work required to bend the joint is entirely due to compression of the gas inside the tube [17,18] Differences in torque predictions between the beam model and the membrane model illustrate the relative importance of elasticity and gas compression in space suit joint mobility.

One application of space suit joint mobility models is in calculating human factors performance metrics, including the one-handed space suited work envelope, which indicates the volume within which a space-suited astronaut can comfortably work with one hand. The current NASA space-suited work envelope[2], which is used in planning EVA's, differs qualitatively from other work envelopes found in the literature.[46] A recently-developed computational technique allows suited work envelopes to be calculated using mathematical models of the torque-angle characteristics and kinematics of a space suit.[31, 28-30]

1.3 Objectives

The primary aim of this thesis is to advance the current understanding of astronauts' capabilities and limitations in space-suited EVA by developing models of the constitutive and

compatibility relations of a space suit, based on experimental data, and utilizing these fundamental relations to estimate a human factors performance metric for space suited EVA work. The three specific objectives are to:

1. Compile a detailed database of torques required to bend the joints of a space suit, using realistic, multi-joint human motions.
2. Develop a mathematical model of the constitutive relations between space suit joint torques and joint angular positions, based on experimental data and compare other investigators' physics-based models to experimental data.
3. Estimate the work envelope of a space suited astronaut, using the constitutive and compatibility relations of the space suit.

1.4 Overview

The body of work that makes up this thesis includes experimentation, empirical and physics-based modeling, and model applications. A detailed space suit joint torque-angle database is compiled with a novel experimental approach that uses space-suited human test subjects to generate realistic, multi-joint motions and an instrumented robot to measure the torques required to accomplish these motions in a space suit. Based on the experimental data, a mathematical model is developed to predict joint torque from the joint angle history. Two physics-based models of pressurized fabric cylinder bending are compared to experimental data, yielding design insights. The mathematical model is applied to EVA operations in an inverse kinematic analysis coupled to the space suit model to calculate the volume in which space-suited astronauts can work with their hands, demonstrating that operational human factors metrics can be predicted from fundamental space suit information.

1.5 Organization

Chapter 1 describes the motivation, objectives, and organization of this thesis. Chapter 2 presents prior research and background information on three topics: previous researchers' measurements of the torques required to bend space suit joints, physical and mathematical models that may be used to relate space suit joint torques and displacements, and an EVA human factors metric, the work envelope of an astronaut in a space suit, including the currently-used work envelope and an alternative method for work envelope estimation that is able to incorporate the constitutive and compatibility relations of a space suit. Chapter 3 details the compilation of the experimental space suit mobility database. Chapter 4 describes a mathematical model derived from experimental data that relates space suit joint position and torque and investigates structural mechanics-based techniques for modeling space suit joint mobility. In Chapter 5, a work envelope for a space-suited astronaut is calculated, based on the experimentally-determined constitutive and compatibility relations of a space suit. Chapter 6 summarizes conclusions and recommendations from Chapters 3, 4, and 5.

Chapter 2 *Background*

2.1 Overview

The space suit has an important effect on an astronaut's mobility during extravehicular activity (EVA). Understanding the mechanics of the space suit on a joint by joint level enables operational-level human factors performance metrics, such as the work envelope, to be predicted for a wide range of scenarios.

This literature review begins by describing the space suit currently used by the United States, which is known as the Extravehicular Mobility Unit or EMU. Then the torque-angle characteristics of space suit joints are discussed. Several researchers have collected data on space suit joint torque-angle relations, but they disagree on the magnitudes of the torques required to bend the same space suit's joints. The third section of the literature review discusses several techniques that may be used to model the torque-angle characteristics of a space suit joint, including both empirical modeling methods and physical modeling methods. In the last section, a human factors metric, the work envelope of a person wearing a space suit, is introduced and discussed. NASA's published suited work envelope, which differs qualitatively from unsuited work envelopes in the literature, is described. Finally, the literature review presents an alternative method for estimating work envelopes, based on robotics analysis methods, which enables the space suit models discussed earlier to be used to calculate work envelopes for astronauts in space suits.

2.2 The Extravehicular Mobility Unit space suit

The Extravehicular Mobility Unit (EMU), pictured in Figure 2-1, is the space suit used for all United States EVA's. This space suit is a self-sufficient miniature spacecraft for the EVA astronaut. The EMU's life support system provides a pure oxygen atmosphere pressurized to 4.3 psi (30 KPa), removes excess carbon dioxide, and rejects waste heat from the space suit. Consumables in the EMU's life support system last approximately 8 hours. A fully-charged flight EMU weighs approximately 260 lb. (118 kg).[32]

Cost estimates for a flight-qualified EMU, including the life support system, range from $15-40 million, which is roughly comparable to a modern fighter aircraft.The high cost of a space suit can be attributed to three factors. First of all, the high development costs of a space suit are amortized over the small number of space suits that are manufactured. Sec-

ond, the portable life support systems that are used in current space suits use very highly advanced and specialized technology, including some key components that are available only from a single source. Finally, the space suit is a spacecraft in itself; consequently, its function is critical to the EVA astronauts' safety, and large costs are incurred in documentation and certification that the space suit will function properly.

Figure 2-1. Extravehicular Mobility Unit (EMU).

A detailed drawing of the EMU is shown in Figure 2-2, with relevant portions of the space suit indicated by different colors. The hard upper torso (HUT), shown in yellow, is constructed of rigid fiberglass and covers the astronaut's torso, providing mechanical interfaces and structural support for the portable life support system backpack. The soft arms and legs of the space suit, shown in white, are composed of three fabric layers: the bladder, restraint layer, and thermal micrometeoroid garment (TMG). The innermost layer is the bladder, which is made of a gas-tight urethane-coated nylon material. The bladder layer retains the suit atmosphere, but, by itself, it would stretch and balloon outward when the suit is pressurized. The dacron restraint layer maintains the shape of the pressure suit, restricting radial expansion of the suit, providing joint articulations, and preventing the arms and legs from lengthening when the suit is pressurized. The TMG, which provides insulation and micrometeoroid protection, is composed of a neoprene-coated nylon liner, aluminized mylar insulation, and an ortho-fabric outer layer. The display and control module (DCM), which is shown in green in Figure 2-2, is located on the front of the HUT and houses all of the controls and displays for the space suit's life support system. Being able to reach and manipulate the controls on the DCM is an essential ability that an astronaut must have to be qualified to perform EVA's.[32]

The EMU splits into two segments for donning and doffing. The upper segment is made up of the HUT and arms and the lower segment, composed of the legs and torso below the HUT, is referred to as the lower torso assembly (LTA). The two segments join with an air-tight locking ring at the base of the HUT, known as the body seal closure. To don the space suit, a person puts on the LTA first. The HUT is mounted in a fixed position, either on a rack for ground operations or latched to a fixture in the airlock for space operations. The person gets into the HUT from the bottom, extends his or her arms above the head and into the arms of the space suit, then latches the LTA and HUT together at the body seal.The helmet and gloves are donned after the HUT and LTA are mated.

When a space suit is pressurized, the fabric portions become virtually rigid and, without special provisions for joints, an astronaut in a pressurized space suit would be essentially immobile. The joints on the EMU are articulated with both rotational bearings and bending joints built into the fabric. Bearings on the EMU are highlighted in color in Figure 2-2. Bearings on the EMU are located at the interface between the HUT and upper arm, which is called the scye, shown in red, on the upper arm, shown in blue, at the wrist, shown in orange, and at the waist, shown in purple.

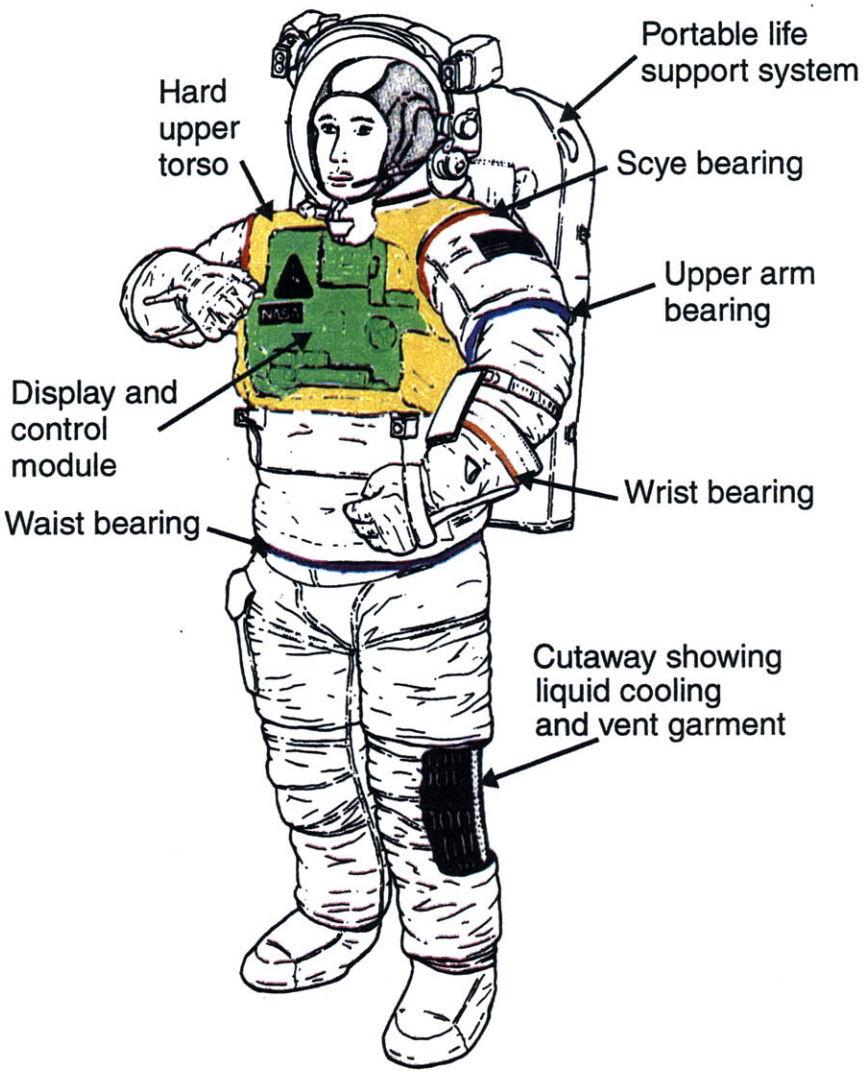


Figure 2-2. Drawing of the EMU, showing the hard upper torso in yellow, display and control module in green, scye bearing in red, arm bearing in blue, wrist bearing in orange and the waist bearing in purple.

Bending joints on the EMU are located at the shoulder between the scye bearing and upper arm bearing, elbow, wrist, hip, knee and ankle. The bending joints on the EMU, known as flat-pattern joints, are formed by sewing pleats into the fabric on the outer side of the joint. The pleats open as the joint bends, allowing the internal volume of the space suit to remain approximately constant.

A closer view of a flat panel joint is shown in Figure 2-3, where the TMG material is pulled down to expose the restraint layer at the knee. The five pleats at the front of the knee open as the knee is bent. Restraint cords that run down the left and right sides of the leg and arm carry the axial loads caused by the suit's internal pressure, preventing the arm and leg from lengthening when the suit is pressurized. The flexible axis of the joint goes

through the restraint cords, with the pleated surface forming the outside of the bent segment. Bending about this axis is relatively easy, while bending the joint perpendicular to the flexible axis is virtually impossible for a person in a pressurized space suit.

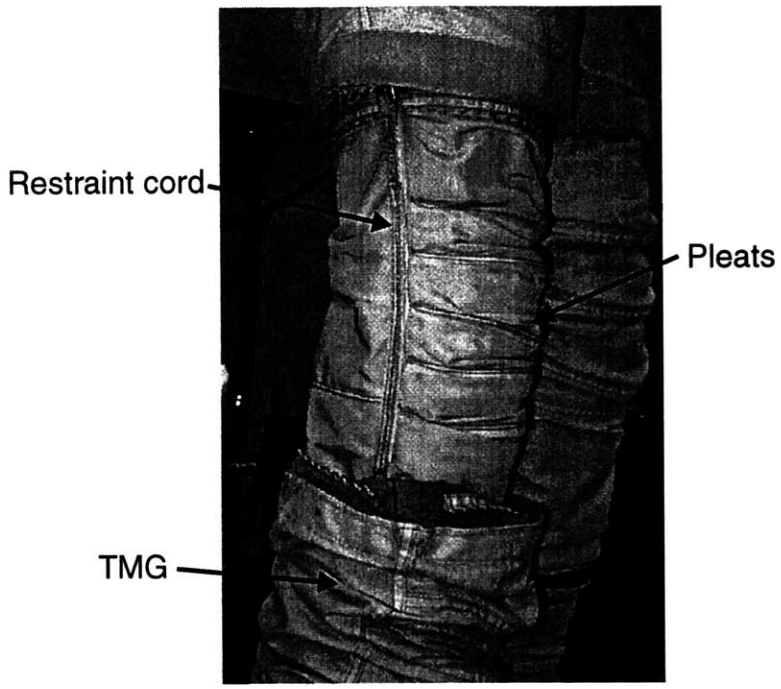


Figure 2-3. EMU knee joint, with TMG layer pulled down to expose restraint layer.

The joint design shown in Figure 2-3 bends only about a single axis, which is well-suited to single axis "hinge" joints like the knee, but poorly-suited to three-axis joints such as the shoulder and hip. Three-axis bending at the shoulder joint is accomplished by the combination of the two rotational bearings at the scye and upper arm and the fabric bending segment located between the two bearings. The fabric shoulder section only bends about a single axis, which is perpendicular to its restraint cords, but the fabric section is free to rotate about the scye and upper arm bearings so that the flexible axis may be aligned with the desired axis of arm motion. It is generally not necessary to manually position the shoulder segment; attempts to bend the shoulder segment about its stiff axis result in rotation of the shoulder segment so that the flexible axis is aligned with the bending axis of the astronaut's shoulder. The location and orientation of the scye bearing has an important impact on the range of motion at the shoulder joint. Because the areas that are important to reach with the hands, such as the controls on the DCM and most worksite locations, are located in front of the suited person, the scye bearing openings are shifted forward and towards the center of the HUT. As a result, in a relaxed body position, the arms extend forward, rather than out to the sides as they would without the space suit. The amount of forward shift of the arms varies from one person to another, depending on how closely their shoulder width fits the shoulder width of the HUT.

A multi-axis joint at the hip, which would allow realistic hip motion, was not built into the EMU. The space suit's hip joint is a single-axis fabric joint, similar to the knee, which is

flexible about a single axis, allowing hip flexion/extension, or forward-backward motion of the leg, only. It is very difficult to accomplish any hip abduction/adduction, or left-right motion of the leg, in a pressurized EMU, because this motion is perpendicular to the flexible axis of the joint. The rationale for this design decision was that NASA's EVA procedures call for EVA astronauts to do most of their work with their feet secured in foot restraints. This work arrangement calls for flexibility in the upper body, but only limited mobility in the legs, so that astronauts can use the stiffness of the space suit's lower body to aid in positioning themselves and transmitting forces and torques from their upper body to the foot restraint with minimum effort.

The EMU, like all space suits, is a complex machine that allows an astronaut to work outside his or her spacecraft. The mobility features that are required for the astronaut to do useful EVA work have to be specifically designed into the space suit. The EMU's mobility features, which include rotational bearings and fabric bending joints, were tailored to the work techniques that are used in EVA's. The next section focuses on measurements of the compliance of fabric bending joints used in space suits.

2.3 Torque-angle measurements on space suit joints

Several investigators have reported on torque and angle measurements taken on space suit fabric bending joints. Since it is impossible to measure torques directly in a human test subject's joints while he or she is wearing a space suit without either using invasive instrumentation or restricting the subject's motion, measurements of the torque required for a human to bend space suit joints have to be accomplished indirectly. Methods for torque measurement used by these investigators include measuring the torques required to bend an unoccupied, pressurized space suit by exerting torques on the outside of the suit and comparing the strength of human test subjects wearing the space suit and unsuited.[3, 15, 40, 41]

The majority of torque data collected on space suit joints comes from tests of empty, pressurized space suits.[15, 3, 40] The empty suit condition refers to a testing condition in which there is no person or support structure occupying the volume inside the space suit. Additionally, limited data has been collected on the torques required to bend space suit joints by human subjects wearing the space suit.[41] Two of the more extensive torque-angle data sets from empty space suits share common qualitative features. However, there are important differences in the magnitudes of torques measured with empty suits and with human subjects: torques measured with empty space suits are lower than those measured with human subjects wearing the space suits. Although there may be good reasons to expect that space suit joints will be stiffer when a person wears the space suit than when the space suit is empty, it is not clear whether the discrepancies in these investigators' torque magnitudes should be attributed to real differences in joint stiffnesses or shortcomings in the experimental methods. This unanswered question motivates the experimental portion of this thesis.

2.3.1 Empty Suit Data

A study comparing the EMU and two advanced technology demonstration suits obtained both subjective evaluations of the three space suits and extensive torque-angle data from the empty, pressurized suits.[15] Joints were bent by applying torques to the exterior of the pressurized space suit and measuring the resulting displacement. Plots of torque vs. angle for the EMU shoulder, elbow, and knee joints are shown in Figure 2-4.

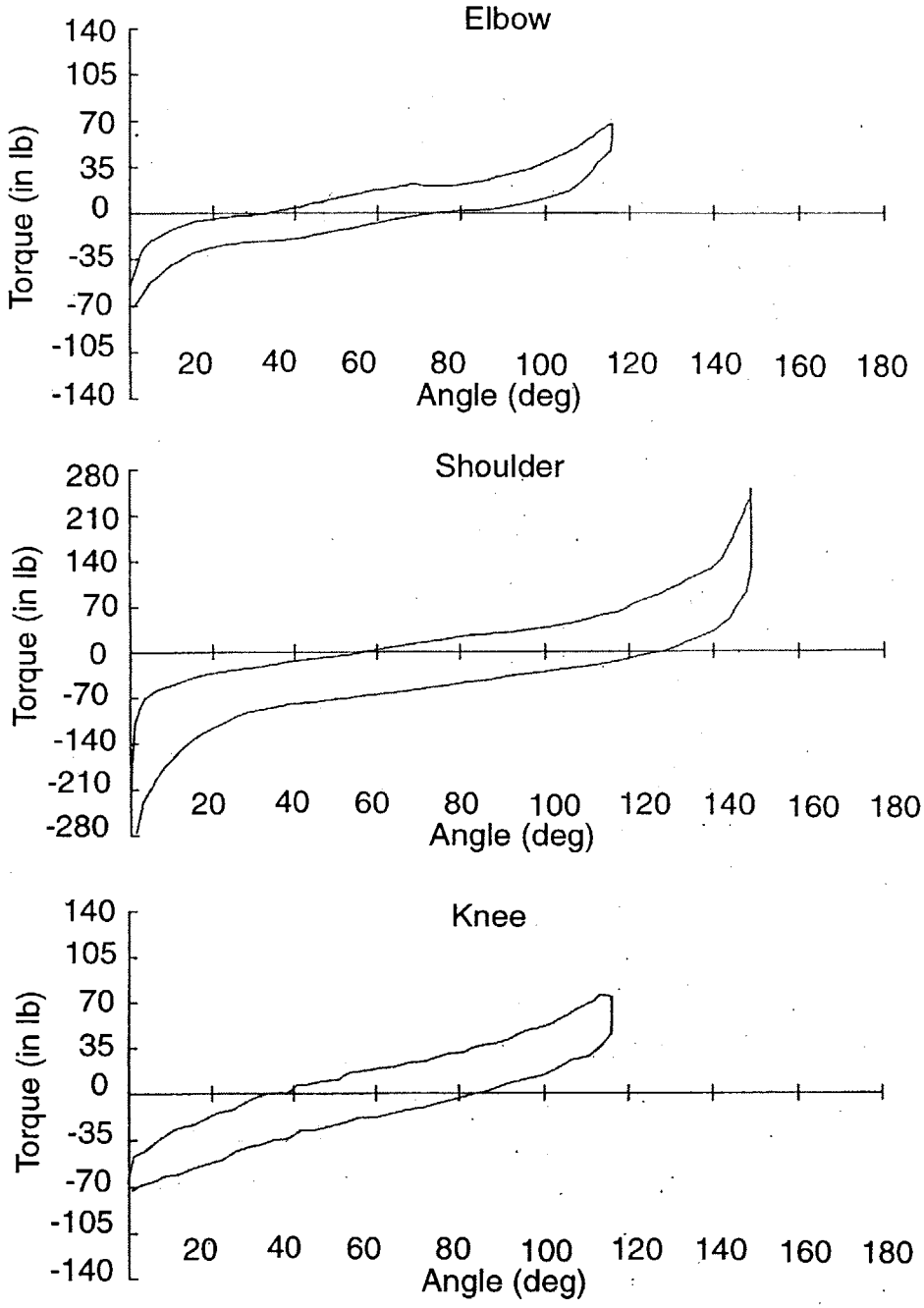


Figure 2-4. Torque vs. angle for the EMU shoulder, elbow, and knee. [15]

Joint bending torque data has also been published for the Russian Orlan DMA space suit.[3] As part of a design study of space suit operating pressure, the torque required to hold the knee or elbow in a bent position (72 deg or 80 deg) was measured at various suit pressures. At an operating pressure of 30 kPa, which is the same operating pressure as the EMU (but lower than the Orlan DMA's design pressure), a torque of 6.0 Nm was required to hold a knee flexion angle of 72 degrees and 2.2 Nm was required to hold the elbow at 80 degrees.

Torques required to hold the bent space suit in position increased linearly with operating pressure. For comparison, Dionne[15] reported torques of 3.2 Nm for the knee at 72 degrees and 2.0 Nm for the elbow at 80 degrees. These results indicate that the Orlan DMA knee joint is considerably stiffer than that of the EMU, while the elbow stiffnesses of the two space suits are similar.

The dependence of joint stiffness on operating pressure was also studied as part of an early design study for a European EVA suit concept.[40] Prototype fabric joints for the shoulder and elbow were built and integrated into an EVA suit demonstrator. The prototype joints were tested empty at several operating pressures, to assess the torque-angle characteristics as a function of pressure. A torque-angle plot for a prototype elbow joint is shown in Figure 2-5. The torque required to bend the joint was only weakly dependent on operating pressure near the joint's neutral position, but increased with increasing pressure near the joint's extremes.

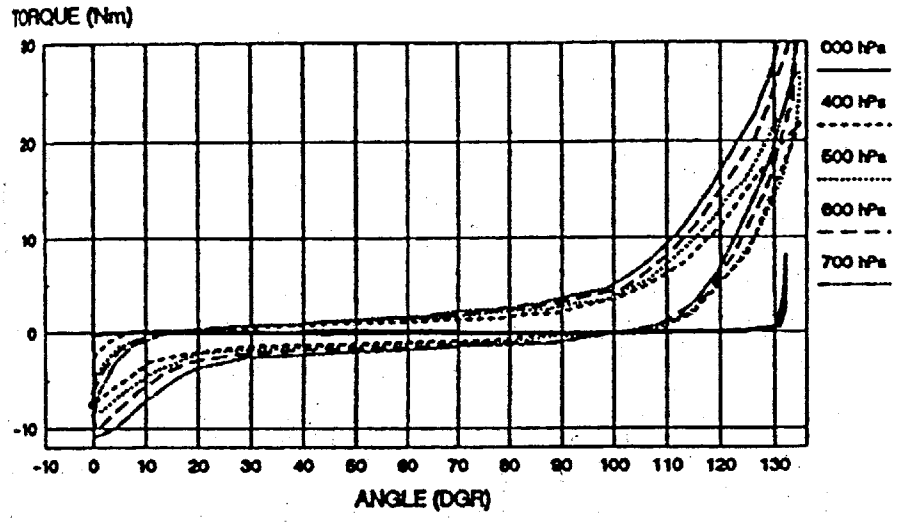


Figure 2-5. Torque vs. angle for a European EVA suit prototype elbow. [40]

2.3.2 Suit Torque Data from Human Subjects

Direct torque measurements from human joints would require strain gauges to be placed directly on bones. On rare occasions, bending moment data has been collected from human joints, using instrumented prosthetic hips.[9] Since these methods are impractical for human testing of space suits, indirect methods are used to estimate the torques exerted by human test subjects wearing space suits.

A study conducted by Morgan et al.[41] compared the strength of test subjects while wearing space suits and unsuited. While the investigators were primarily concerned with measuring and predicting the strength of a suited astronaut, the difference between suited and

unsuited torques may be interpreted approximately as the torque required to bend the suit joint. Strength measurements were taken using a dynamometer, while the test subjects made cyclic movements at constant velocity. Strength of the knee, elbow, wrist and shoulder were tested for 6 subjects under both suited and unsuited conditions. For elbow flexion, suited and unsuited maximum torques differed by 2.7 Nm - 11 Nm. In knee flexion, the suited and unsuited maximum torques differed by 6.8 Nm - 20 Nm. The motions allowed by the dynamometer were restricted to a single axis for each joint and limited in range. An important weakness of Morgan et al.'s study is that the suited and unsuited maximum torque differences can only be correctly interpreted as the torque required to bend the space suit's joints if the test subjects gave a consistent maximum effort all the time.

2.3.3 Discussion of space suit torque data

The angle and torque data collected by Dionne[15] and Menendez et al.[40] shows some important qualitative features that are similar across different joints. First, the torque-angle relationship is nonlinear. The slope of the torque-angle curves shown in Figure 2-4 and Figure 2-5 are flat over a large portion of the angle range, then increase dramatically near the ends of the range. This characteristic, known as hardening, is exhibited most notably in the shoulder and elbow joint data. The second common feature in these two data sets is hysteresis. The torque-angle curves follow a different path for bending the joint as for straightening. it. This response is typical of fabric structures and can be seen in prior experimental data in Figure 2-4 and Figure 2-5.[27]

A comparison of torque magnitudes collected by Dionne[15], Abramov et al.[3], Menendez et al.[40], and Morgan et al.[40], in the limited range where these data sets overlap, is shown in Table 2-1. Torques measured in Morgan et al.'s experiments with human subjects are greater than torques measured from empty space suits.

Table 2-1: Space suit joint torques

Study	Dionne	Menendez et al	Abramov et al.	Morgan et al.
Methods	EMU empty suit	Prototype joint segments, empty	Orlan-DMA, 4.3 psi (30 KPa) empty suit	EMU human subjects
Knee, 72 deg.	3.2 Nm	NA	6.0 Nm	8.1 Nm
Elbow, 80 deg	2.0 Nm	2 Nm	2.2 Nm	3.4 Nm

Although it would be easy to attribute the discrepancy between the results of Morgan et al.[41] and those reported by Dionne[15] to shortcomings of the suited/unsuited strength comparison method, there are good reasons to expect that torques required to bend the space suit joints are different when the suit is being worn. It is likely that when an empty, pressurized space suit joint is bent by applying a torque to the exterior of the space suit, the space suit has a different deformed shape than it would have if the bending torque were

applied by a person inside the suit. When a person wearing a space suit bends a joint, the space suit joint is forced to bend about an axis that is as near as possible to the person's joint axis. Furthermore, the radial inward deformations of the space suit fabric are constrained by contact with the wearer's arm or leg. When an empty suit is bent from the outside, these two constraints are not enforced. The bending axis of the joint may not correspond with the bending axis of the human joint, and the space suit material may deform radially inward by an unrealistic amount. Thus, it is unclear whether the differences in torque magnitudes measured with empty suits and human subjects are due to real torque differences or problems with the experiment methodology. This question motivates the experimental portion of this thesis: to collect torque and angle data on EMU space suit joints, using an instrumented robot as a surrogate for a human subject wearing the space suit, delivering an extensive torque-angle database on 11 space suit joints in both simple and complex motions.

2.4 Modeling space suit mobility

Models of space suit mobility are useful both in making numerical predictions of the torque that corresponds to a specific angular displacement of a space suit joint and in assessing the effects of design parameters on the compliance of a space suit joint. These two applications clearly require two different modeling approaches: first, a descriptive mathematical modeling technique based on experimental data and second, a theoretical model based on physical principles. This section reviews some techniques that may be used to develop descriptive models of systems that have input-output characteristics similar to the torque-angle relations of space suit joints and discusses several structural mechanics-based theoretical modeling approaches for inflatable structures that may be representative of the arms and legs of a space suit.

2.4.1 Descriptive models

The experimental data collected by both Dionne[15], in Figure 2-4 and Menendez et al[40] in Figure 2-5 indicate that the torque-angle characteristics of space suit joints are hysteretic. A hysteretic system has an output that depends on both the current value of the input and its history. An example of a hysteresis transducer and its output are shown in Figure 2-6. As the input, $u(t)$, increases, the output, $f(t)$, follows the lower branch of the curve, and as the input decreases, the output follows the upper branch. For every reachable point in the input-output space, there are two or more f-u curves that intersect it. [39]

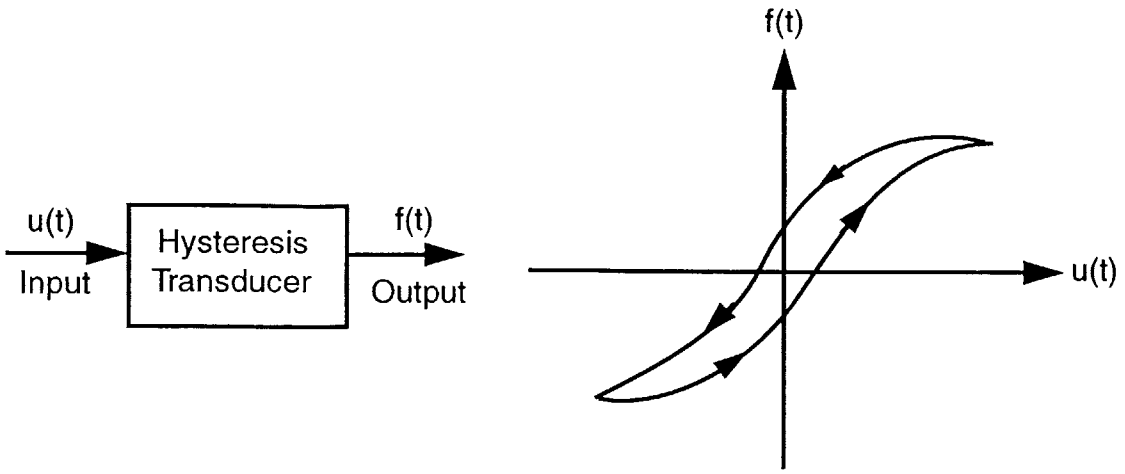


Figure 2-6. Hysteresis transducer and output.

Hysteresis is a common characteristic of fabric structures. When fabric is loaded, the individual yarns elongate and slide over each other, resulting in load-extension curves characterized by the yarn compliance and the friction between yarns. Frictional effects are particularly notable when fabrics are loaded biaxially and deform in shear, because this loading condition results in the most relative motion of the yarns.[27]

A simplified conceptual model of hysteresis in fabric shear, which is shown in Figure 2-7, serves to illustrate hysteretic characteristics of fabric shear. The model consists of two springs in series surrounding a static friction element. When a extensional force is applied at spring 1, spring 1 initially extends. This corresponds to a shear deformation of the fabric, without the yarns sliding over each other, and the force-displacement curve follows line OA. When the tension in spring 1 exceeds the static friction force, *F*, the static friction element slides and stretches spring 2. Sliding of the static friction element corresponds to yarns in the fabric sliding over each other. This phase corresponds to line AB on the force-displacement plot. When the direction of the force at spring 1 is reversed, initially, only spring 1 will compress, corresponding to elastic recovery of the fabric without yarn shifting, on line BC of the force-displacement plot. As in the extension case, when the compressive force in spring 1 exceeds the static friction force F, the friction element slides, compressing spring 2 along line CD. Finally, if the extensional force is applied again at spring 1, spring 1 will initially extend (line DE) and when the tension in spring 1 exceeds the static friction force F, the frictional element will slide and extend spring 2, along line EA of the force-displacement plot, completing the hysteresis loop.[27]

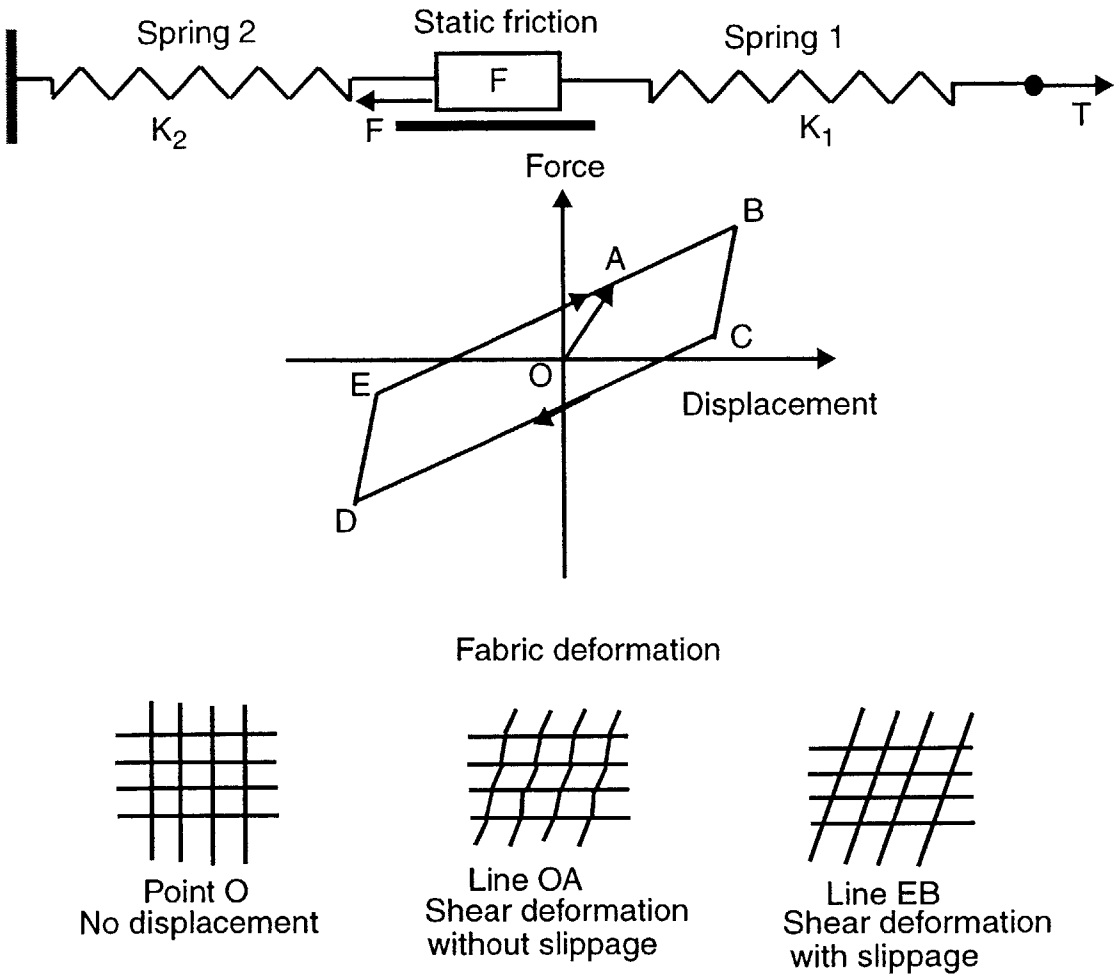


Figure 2-7. Simplified model of hysteresis in fabric shear.[27]

The spring-static friction model of hysteresis provides a conceptual explanation for hysteresis in fabric deformation on a micro scale, but, because the micro scale friction and yarn modulus properties cannot be reliably measured, it is too simplistic to accurately explain fabric deformations on a macro scale. However, other hysteresis modeling techniques, based on mathematics rather than physics, are available to construct input-output curves that reproduce those found in experimental data.

Mathematical hysteresis modeling techniques are useful in applications in which it is necessary to reproduce the input-output behavior of a system, but not to include a detailed representation of the physical processes that govern the system. An important example of this situation is the problem of designing a compensator for a control system that has a hysteretic actuator. Actuators that use smart materials, such as shape memory alloys or piezoceramics, exhibit hysteretic behavior, which must be compensated in the control system in order for the system to be stable and have satisfactory performance. A common approach in this situation is to design a linearizing compensator that incorporates a model of the hysteretic actuator.[20, 21, 51, 52, 47]

Control systems designers have used several hysteresis modeling techniques to design linearizing compensators for hysteretic actuators. Tao and Kokotovic[47] proposed a simplified hysteresis model to be used for adaptive control of hysteretic systems. The Tao and Kokotovic model assumes a simplified shape for the hysteresis curves that applies to their control application, but does not correspond with the shapes of hysteresis curves in fabric deformation.

Webb, Kurdila and Lagoudas designed and implemented a linearizing compensator for a shape memory alloy actuator that performs model identification in real time, tracking the time-varying hysteresis characteristics of the actuator.[51, 52] Their technique is based on the Krasnoselskii-Pokrovski (KP) hysteresis model.[33] The KP hysteresis model represents the output of a hysteresis transducer as the weighted sum of outputs from a finite number of simpler hysteresis transducers. The simplified hysteresis transducer that forms the basis of the KP model, referred to as a kernel, is shown in Figure 2-8. The KP kernel is parametrized by s_1 and s_2, which represent the width of the hysteresis, and a, which represents the slope of the increasing and decreasing curves. The kernel weightings are defined for evenly-spaced, discrete values of s_1 and s_2. The KP model is identified by setting the weighting value of each of the kernels. Banks, Kurdila and Webb[8] proved that for the proper choice of rise time a, the model can be identified in real time and furthermore, an exact inverse of the model exists. The KP model is well-suited to applications with discretized data, and its online identifiability and invertibility make it a powerful tool for compensating time-varying hysteretic systems, such as shape memory alloy actuators. However, the proof that an online identification solution for the KP model exists does not mean that a generally-applicable method for accomplishing KP model identification also exists. Moreover, there is no experimental or theoretical evidence that space suit joints have time-varying hysteretic behavior. Because a general solution to identifying the KP model does not exist, and the time-varying capabilities of the model are unnecessary, the KP model is too complex to be used for modeling space suit joint torque-angle data.

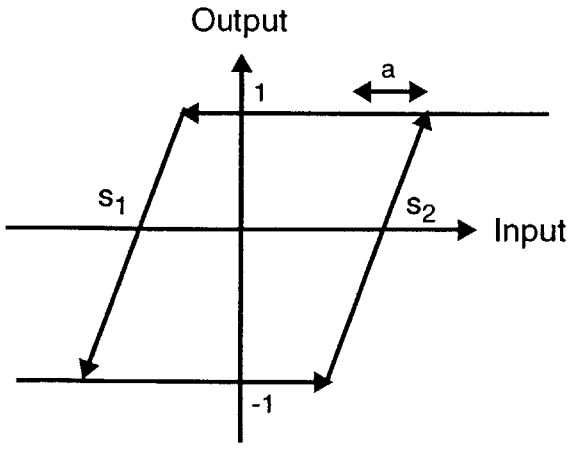


Figure 2-8. Krasnoselskii-Pokrovski (KP) hysteresis model kernel.[51]

The Preisach hysteresis model is less complicated mathematically and less accurate for discrete data than the KP model, but, with the proper formulation, the model identification step is straightforward.[39, 16, 20] The Preisach model originated as a physical model of magnetization. It was later extended to a purely mathematical form, which allowed the model to be applied to other hysteretic physical systems. Mayergoyz made extensive contributions to developing numerical implementations and alternative formulations of the Preisach model.[39, 16]

Similar to the KP model, the Preisach hysteresis model represents the output of a hysteresis transducer as a weighted sum of simple hysteresis operators, known as kernels. The Preisach model kernel, shown in Figure 2-9, is different from the KP model kernel, which is shown in Figure 2-8. Kernel weightings for the Preisach model are defined continuously in terms of the upward and downward switching coefficients α and β. The continuous formulation allowed Preisach to devise a graphical representation of the kernel weightings, which were plotted in three dimensions versus α and β and integrated to yield the model output.[39] The continuous nature of the Preisach model poses some difficulties when the model is applied to discrete data. Unlike the KP model kernel, which makes a smooth transition from its "down" state to its "up" state, the Preisach model kernel jumps from -1 to +1 instantaneously when the input value crosses the α and β thresholds. Consequently, when the Preisach model is discretized, the α and β resolution must be much higher than the resolution in s_1 and s_2 for the KP model, in order to prevent discontinuities in the model output. As a result, for discrete data, the Preisach model is less numerically efficient and less accurate than the KP model.[51] The identification process for the Preisach model is well-defined, however, making it a good choice for modeling space suit joint torque-angle relations. A detailed description of how the Preisach model is used in this thesis to model the torque-angle characteristics of space suit joints can be found in Chapter 4.

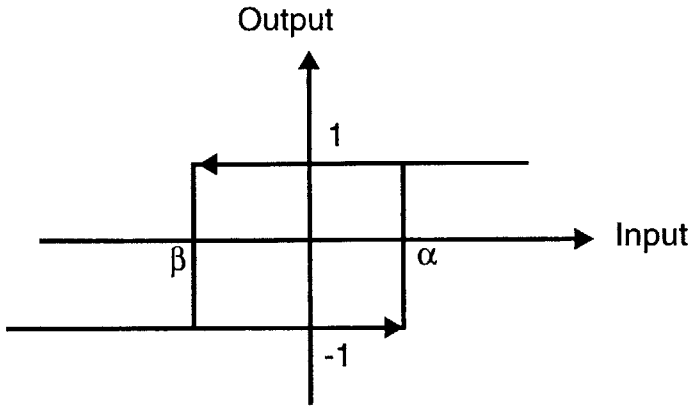


Figure 2-9. Preisach hysteresis model kernel.

Ge and Jouaneh[20, 21] used a Preisach model to compensate a piezoceramic actuator, using a formulation of the Preisach model developed by Doong and Mayergoyz.[16] The Doong and Mayergoyz formulation accomplishes the identification step while avoiding the

numerical differentiation of experimental data and double integrations that previous Preisach model formulation required.

Rahn applied the Preisach hysteresis modeling method to space suit mobility in a dynamic simulation of an astronaut moving a large, massive object.[45, 42] He used data reported by Dionne[15] to develop a Preisach model of the torque-angle relationships of several space suit joints. Rahn's simulation study demonstrated that torque contributions from the space suit modified the body positions and joint torques of the simulated astronaut, thus having an important impact on the results of EVA dynamic simulations.

Fabric structures have hysteretic behavior because of the combination of elastic and frictional forces that occur as the yarns in the fabric simultaneously elongate and slide over each other. Real fabrics are too complex to build up a physical model of this hysteresis process based on behavior at the yarn interaction level, but techniques do exist for reproducing hysteresis curves based on experimental data. Three hysteresis modeling techniques that have been used to model hysteretic actuators in control systems were reviewed. The Tao and Kokotovic model is not suitable for space suit modeling because it assumes a hysteresis curve shape that is inconsistent with space suit data. The Krasnoselskii-Pokrovski (KP) model is the most accurate and versatile of the three, but there is no generally-applicable method for identifying its coefficients, and developing a new KP model identification scheme is beyond the scope of this thesis. The Preisach model is less accurate than the KP model for discrete data, but its identification step is well-defined and straightforward. Thus, the Preisach model is chosen for modeling the torque-angle characteristics of space suit fabric bending joints.

2.4.2 Physics-based Models

While mathematical modeling approaches based on experimental data generate numerical predictions of the torque required to bend existing space suit joints, physical models allow designers to predict torque-angle characteristics of space suit joints that have not yet been built. Currently, little theory exists on the bending performance of space suit joints. Theoretical and experimental work on inflatable structures in other regimes may be relevant to space suit joint mobility. However, the mobility features built into space suit joints may be too complicated to be adequately described by a generic cylinder bending model. This thesis addresses physical modeling of space suit joints by comparing previous models' predictions to experimental data from space suits and evaluating methods of extending the previous models to explain the torque-angle characteristics of space suit joints.

A variety of inflatable structures have been analyzed, including large, complex structures such as air-supported stadium roofs and buildings and simple structures such as inflated columns and beams.[19] Inflatable structures are particularly attractive for space applications, since they can be launched in a low-volume, deflated configuration, then deployed in space. In the 1960's, several large inflatable space structures were proposed, including an inflatable airlock, a lunar habitation module, and an inflatable spacecraft for artificial gravity experimentation.[35] Recent interest in inflatable space structures has shifted to smaller,

less critical components, such as antennas and solar arrays, with the notable exception of the inflatable habitation module that was proposed for the International Space Station.

This literature review will focus on the two types of inflatable structure models that are most relevant to space suit joint mobility: the beam model and the membrane model. The beam model treats pressurized cylinders as long, slender members, loaded in a single plane, whose behavior is governed by elasticity and buckling phenomena.[35, 38] The membrane model treats pressurized cylinders as structures with inextensible walls whose cross-sectional shape and enclosed volume change with bending deflection.[17, 18, 34] Beams are designed for maximum rigidity, and consequently have high applied loads and low, distributed deflections, and fail suddenly in buckling.[38] In contrast, the membrane model encompasses very large deflections, possibly in a post-buckling regime. Deflections considered in the membrane model may be highly localized, as a kink or hinge in the fabric, while beam deflections are distributed over a finite length.

The beam and membrane models represent two bounding cases of the physical processes that govern space suit joint mobility. The beam model assumes that bending deflections are caused by extension of the fabric wall of the cylinder and that the gas inside the cylinder is never compressed. In contrast, the membrane model assumes that the fabric wall of the cylinder is inextensional and all of work done to bend the cylinder goes into compressing the gas inside it. It is reasonable to expect that the mechanics of space suit joints fall between the beam and membrane model regimes, with some contributions from both elasticity and gas compression. A comparison between the beam and membrane models' torque-angle predictions and experimental space suit joint torque-angle data serves to illustrate the relative contributions of elasticity and gas compression on space suit mobility.

The first beam model was developed by Comer and Levy.[12] Comer and Levy modeled the load-displacement characteristics of an inflatable cantilever beam with a tip load, considering the effects of pressurization and the fact that fabric cannot sustain compressional loads. They predicted that internal pressurization would give flexural rigidity to the inflated beam, but that as the beam is bent, wrinkles form at the root of the beam, decreasing its rigidity. When the wrinkled region extends completely around the beam, it fails in buckling. This work considered only the longitudinal stresses in the beam due to pressurization.[12] Experimental data reported by Webber showed that inflatable beams were less rigid and failed at lower applied loads than Comer and Levy's model predicted.[53] Main, Peterson, and Strauss[35, 38] extended Comer and Levy's model to consider the biaxial state of stress in the fabric, predicting that wrinkling would occur at lower applied moments and that stiffness would be less than that predicted by Comer and Levy's model. Main, Peterson, and Strauss's model agreed with experimental data for beams that had a length to diameter ratio greater than 10; the shorter beams they tested did not follow their model.[35]

Figure 2-10 shows the distribution of axial stresses in beams in pure bending in three configurations, illustrating the difference between unpressurized, solid beams and pressurized fabric beams. A is an unpressurized, hollow cylinder of solid material. The material is

stressed in extension on the outer half of the bent section and in compression on the lower half. B is a pressurized, fabric beam in an unwrinkled condition. Because of the extensional stresses due to pressurization, the axial stresses over the entire cross-section of the beam are extensional. C is a pressurized, fabric beam in a wrinkled condition. The region of the beam with an extensional stress lower than a specified limit is considered to be wrinkled and has a stress resultant of zero.[12, 35, 38]

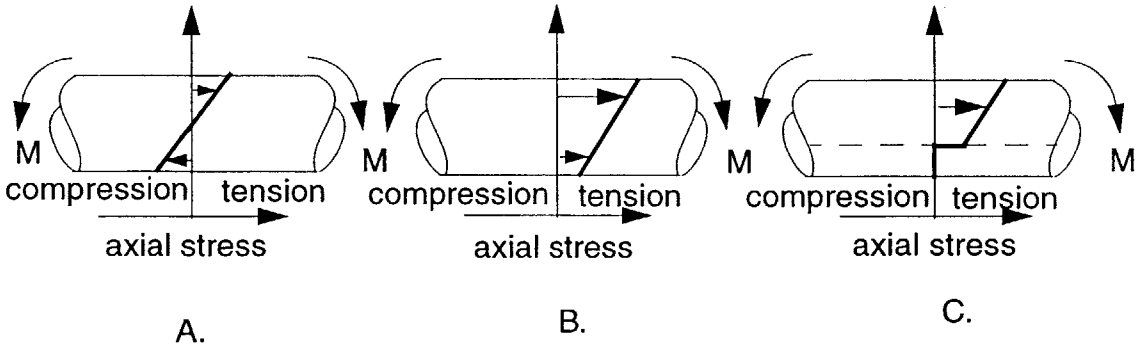


Figure 2-10. Stress resultants in three beam configurations: A.) Solid, unpressurized cylinder. B.) Pressurized, unwrinkled cylinder. C.) Pressurized, wrinkled cylinder.

Main, Peterson and Strauss assumed that, in the unwrinkled condition, case B, the axial stress varies linearly across the beam, from a maximum axial stress resultant N_{Lm} on the outside of the curved section to a minimum, N_{L0}, on the inside of the curved section. Using polar coordinates to represent locations along the circumference of the beam, they defined the angle θ to be zero at the center of the wrinkled region. The axial stress resultant N_L when the beam is unwrinkled is then given by Equation 2-1.[35, 38]

$$N_L = N_{L0}\left(\frac{1+\cos\theta}{2}\right) + N_{Lm}\left(\frac{1-\cos\theta}{2}\right) \qquad \text{Eq 2-1}$$

The next step is to define the minimum extensional stress that the fabric can sustain. According to the constitutive relations for fabric, which are given in Equation 2-2 in relation to the perpendicular warp and fill directions of fabric, fabric stresses along one axis result in strains in both the stressed direction and the orthogonal direction.[19]

$$\varepsilon_{warp} = \frac{N_{warp}}{E_{warp}} - \frac{\nu_{warp} N_{fill}}{E_{warp}} \qquad \varepsilon_{fill} = \frac{N_{fill}}{E_{fill}} - \frac{\nu_{warp} N_{warp}}{E_{warp}} \qquad \text{Eq 2-2}$$

Consequently, the circumferential stress in the fabric cylinder due to pressurization causes a small compressional strain on the fabric in the axial direction. For a cylinder of radius r, internal pressure p, and fabric Poisson ratio ν, the fabric wrinkles when the tensile axial

stress resultant is less than the axial compressive stress caused by pressurization loading in the circumferential direction. This condition is given in Equation 2-3.[35, 38]

$$N_L \leq \nu pr \qquad \text{Eq 2-3}$$

Using Equation 2-3 as a threshold for wrinkling, the axial stress resultant when the fabric between $\theta=0$ and $\theta=\theta_0$ is wrinkled is as follows.

$$N_L = \frac{(\cos\theta_0 - \cos\theta)(N_{Lm} - \nu pr)}{1 + \cos\theta_0} + \upsilon pr \qquad \text{for } \theta_0 < \theta < \pi$$
$$N_L = 0 \qquad \text{for } 0 < \theta < \theta_0 \qquad \text{Eq 2-4}$$

The relationship between axial stress and curvature of the beam is obtained using the moment-curvature relations for beams. The two equations relating applied moment to beam curvature and axial stresses are listed in Equation 2-5. $M(x)$ is the applied moment on the beam, K is the bending curvature, E is the fabric modulus, y is a linear vertical coordinate on the beam cross section and A is the beam cross sectional area.

$$M(x) = \int \sigma_x y dA = \int_0^{\pi} N_w r^2 \cos\theta d\theta$$
$$M(x) = \int -KEy^2 dA = \int_{\theta_0}^{\pi} -KEr^3(\cos\theta)^2 d\theta \qquad \text{Eq 2-5}$$

The moment-curvature relation for the beam is obtained by integrating both expressions for $M(x)$ with the appropriate N_w over the unwrinkled region of the beam. The two expressions for $M(x)$ are then set equal and solved for K as a function of applied moment yielding the following expression for the wrinkled case.

$$K = \frac{M - 2\nu pr^3 \sin\theta_0}{Er^3[(\pi - \theta_0) + \sin\theta_0 \cos\theta_0]} \qquad \text{Eq 2-6}$$

The remaining step is to determine θ_0, the wrinkle angle. Substituting Equation 2-4 into the stress-moment equation in Equation 2-5 results in an expression relating the maximum axial stress resultant N_{Lm}, θ_0, and the applied moment M. Integrating axial forces over the beam cross section as shown in Equation 2-7 gives another expression relating N_{Lm}, θ_0, and the axial force due to pressurization.

$$F_{axial} = p\pi r^2 = \int_0^{\pi} N_L r d\theta \qquad \text{Eq 2-7}$$

Equating the two expressions for N_{Lm} yields an equation relating the wrinkle angle θ_0 to the applied moment $M(x)$.

$$\frac{M}{pr^3} = \frac{\frac{\pi}{2}[(\pi-\theta_0)+\sin\theta_0\cos\theta_0]-\nu[(\pi-\theta_0)^2-(\pi-\theta_0)\sin\theta_0\cos\theta_0-(2\sin\theta_0)^2]}{\sin\theta_0+(\pi-\theta_0)\cos\theta_0} \quad \text{Eq 2-8}$$

The θ_0 obtained from Equation 2-8 is then substituted into Equation 2-6 to get an equation relating bending moment M and curvature K.

Setting $\theta_0=0$ in Equation 2-8 gives the moment at the start of wrinkling, which is dependent on the fabric Poisson ratio ν.[35, 38]

$$|M_{wrinkle}| = \frac{\pi pr^3}{2}(1-2\nu) \quad \text{Eq 2-9}$$

To calculate the y-axis deflection of the beam, v, Main, Peterson and Strauss used the approximation $K = \frac{d^2v}{dx^2}$ and solved the differential equation of bending numerically for cantilever beams. They constructed inflatable cantilever beams of various sizes and experimentally determined their load-deflection characteristics. The model predictions and experimental data are shown in Figure 2-11. For long, thin beams, as in Figure 2-11A, the model fit the data well, but for short beams, in Figure 2-11B, deflections are generally greater than those predicted by the model.[35, 38]

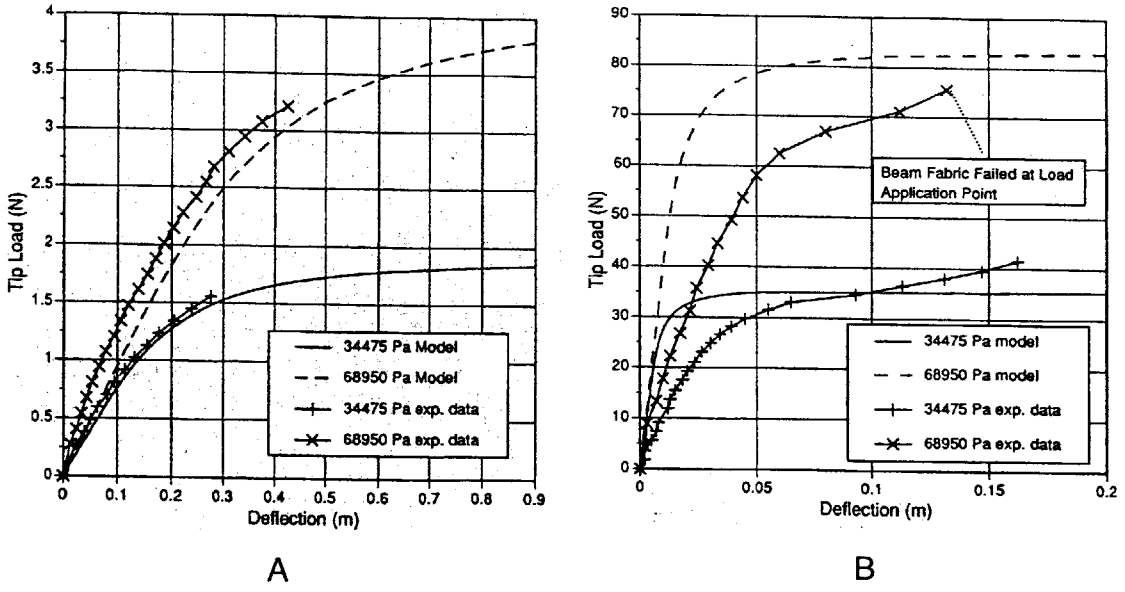


Figure 2-11. Main, Peterson, and Strauss's model predictions and experimental data. A.) Long beam (length=89.9 cm, diameter=5.0 cm). B.) Short beam (length=21.6 cm, diameter=8.0 cm).[35]

While the beam model assumes that the fabric stretches and the internal volume of the pressurized tube is constant, the membrane model makes the opposite assumptions: that the fabric tube wall is inextensible and bending the tube compresses the gas inside it. The membrane model treats the fabric tube wall as a rigid material that transmits forces along its surface, only in tension. Bending deflections of the tube result in changes in the tube's cross-sectional shape and the volume enclosed within it. The assumption of inextensibility of the fabric and the initial, undeformed cylindrical shape of the tube allow the deformed tube's shape to be calculated. The membrane model has been used to describe the load-deflection behavior of inflatable space structures in on-orbit deployment.[17, 18, 34]

Lukasiewicz and Glockner[34] modeled the stability of a cylindrical, pressurized fabric column with eccentric loads. The column, pictured in Figure 2-12A, is oriented vertically and has a hemispherical top. A vertical load, *P*, is applied to the top of the column, some distance away from its centerline. Lukasiewicz and Glockner identified two buckling modes of the column. Figure 2-12B shows the buckling mode of interest, in which the fabric cylinder bends at a point midway along its length. Lukasiewicz and Glockner used the assumption of isometric deformations to approximate the geometry of the deformed column and calculate load and length limits for inflated columns.

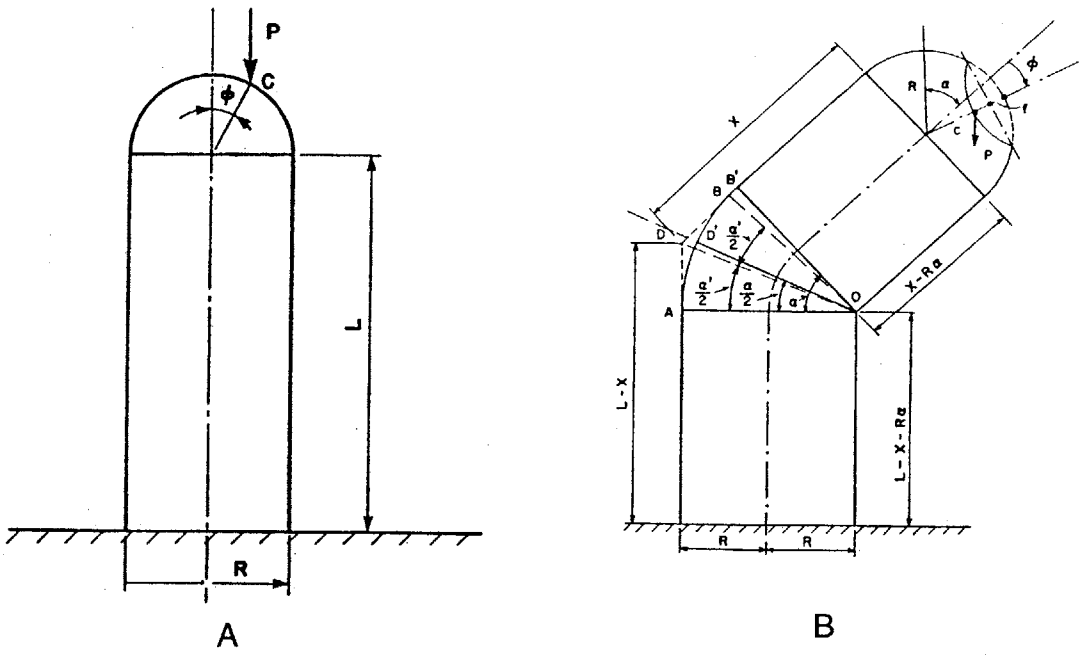


Figure 2-12. Cylindrical column. A.) Undeformed shape B.) Buckled shape.[34]

Fay and Steele[17, 18]built on Lukasiewicz and Glockner's work, making more refined approximations of the deformed cylinder's shape in static conditions, and considering more general loading conditions. Fay and Steele were concerned with the extensional loads and bending moments encountered when a rolled-up fabric tube unrolls as it is inflated, a situation that occurs when inflatable space structures are deployed. They mod-

eled both pinching the tube and bending it, although only the bending model will be discussed here.

Fay and Steele considered a pressurized cylinder of radius R_0, volume V, and internal pressure p. The cylinder may be loaded with an end force Q or bending moment M. They used a variational principle to determine equilibrium configurations for the cylinder. The total potential energy of the system Π, is defined by Equation 2-10 for linear displacement δ and bending angle ϕ.[18]

$$\Pi = -pV - Q\delta - M\phi \qquad \text{Eq 2-10}$$

In the case of pure bending, where $Q=0$, the equilibrium bending moment-bending angle relationship minimizes the potential energy as a function of the bending angle. This relation is given by

$$\frac{d\Pi}{d\phi} = -p\frac{dV}{d\phi} - M = 0$$
$$\Rightarrow M = -p\frac{dV}{d\phi} \qquad \text{Eq 2-11}$$

The relationship between the volume of the bent tube and the bending angle thus determines the moment-angle characteristic for the tube. Fay and Steele applied the following four principles of membrane shape to the bent tube to approximate its geometry.

1. An inextensional surface of zero Gaussian curvature must remain a surface of zero Gaussian curvature in a region of biaxial tension. Thus, the initially flat surface can have a curvature in one direction or the other, but not both, in a region of biaxial tension.
2. In a region of wrinkling, an "averaged" surface may be defined, for which the Gaussian curvature can be either positive or negative. This occurs in a region with a nonpositive component of principal stress.
3. For local equilibrium, constant pressure will be carried by constant curvature.
4. Discontinuity in slope occurs only when an external line load is present.[18]

Based on these four principles, the shape of the bent tube is depicted in Figure 2-13. The outer surface has a radius of $2R_0$, while the inner surface folds inward, forming a partition that extends into the tube. For bending angles less than 1 radian, the partition extends only partially across the tube, as shown in Figure 2-13A, but for bending angles greater than 1 radian, the partition contacts the opposite surface of the tube, as shown in Figure 2-13B. Because the fabric partition cannot withstand compressive stresses, it wrinkles or collapses to the side and does not affect the shape of the outer surface.[18]

It is interesting to note that when Webber[53] bent inflated cantilever beams, he observed that neighboring wrinkles joined together to form deeper wrinkles as loading on the beam

increased.The deepening wrinkles appear to be a transition from the wrinkling behavior predicted by the beam model to the partition predicted by the membrane model.

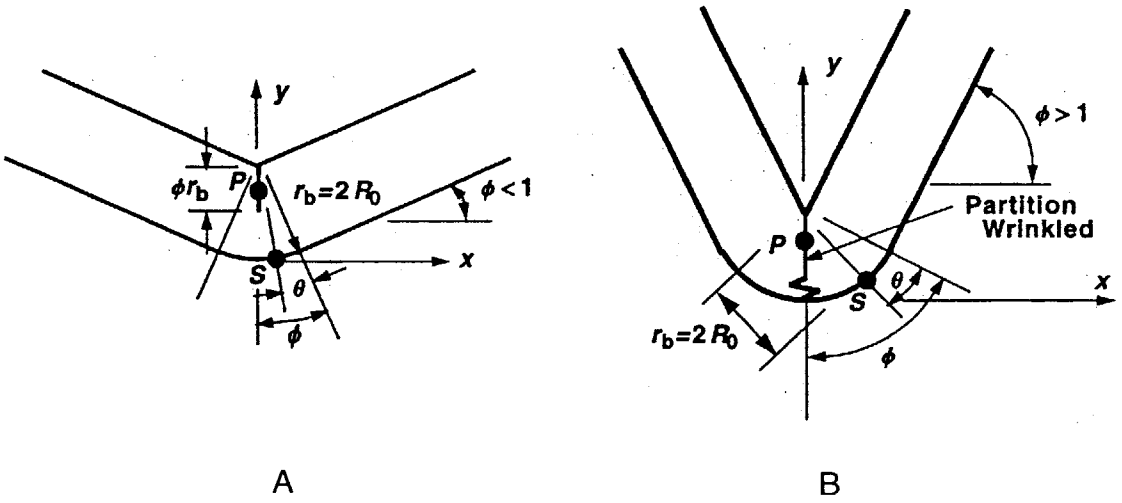


Figure 2-13. Geometry of bent pressurized tube. A.) Bending angle < 1 radian. B.) Bending angle > 1 radian.[18]

Using the membrane deflection principles listed above, the shape of the deformed tube can be calculated by specifying the positions of two points, P and S, on the upper and lower surfaces of the tube. Point P and point S are points on the tube that are equidistant from the end of the tube. Fay and Steele[18] give the following equations for the positions of points P and S, relative to the *x*-*y* coordinate frame of Figure 2-13. θ is defined as the angle between the plane of the tube's undeformed cross-section and point S.

$$\begin{aligned} x_P &= 0 \\ y_P &= 2R(1-\theta) \\ x_S &= 2R\sin(\phi-\theta) \\ y_S &= 2R(1-\cos(\phi-\theta)) \end{aligned} \qquad \text{Eq 2-12}$$

Using the coordinates of points P and S, the cross-sectional area of the tube can be calculated. The oval shape of the cross-section, shown in Figure 2-14, is determined by the four membrane shape principles listed above. Principle 1 states that biaxially stressed surfaces with zero Gaussian curvature retain zero Gaussian curvature as long as they remain biaxially stressed. This means that the cylinder's surfaces, which are initially curved in only one direction, remain curved in only one direction if they are stressed in both longitudinal and circumferential directions.[23] Consequently, in order for the outer surface of the bent tube to curve in the *x*-*y* plane, it has to be flattened in the *y*-*z* plane, forming the flat lower edge of the cross-section. According to principles 3 and 4, the radius of curvature remains constant for constant pressure, and there are no slope discontinuities without concentrated loads. Principles 3 and 4 dictate that the left and right sides of the cross-section are semi-

circles of radius *H/2*, where *H*, the deformed height of the cross-section, is given by the distance between point P and point S.[18] Numerous other investigators have also reported the oval cross-sectional shape of bent hollow cylinders.[10, 43, 49, 54, 55]

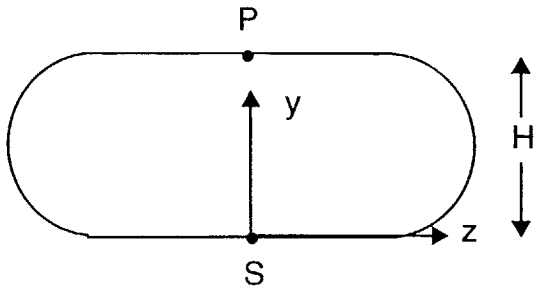


Figure 2-14. Cross-section of the bent tube.

Using the constraint that the perimeter of the deformed cross section is the same as the original circumference of the cylinder, the cross-sectional area is given by

$$A = \pi\left(R_0^{\,2} - \left(R_0 - \frac{H}{2}\right)^2\right). \qquad \text{Eq 2-13}$$

The equation for the cross-sectional area in Equation 2-13 can be combined with Equation 2-12 and integrated numerically along the length of the tube, yielding volume as a function of bending angle. The volume-bending angle relationship can then be used in Equation 2-11 to obtain the bending moment as a function of bending angle.[18]

Fay and Steele tested their model by bending long, thin, urethane-coated fabric tubes with an end force load. The comparison of model prediction and experimental results, shown in Figure 2-15, indicates that the model fits the data well for bending angles less than 1 radian. When the bending angle is greater than 1 radian, the predictions of the model using a wrinkled-partition assumption are closer to the data. It should be noted that, while the basis of the membrane model is not restricted to long, thin cylinders (unlike the beam model), Fay and Steele used only long, thin cylinders in their experiments, and consequently, there is no experimental data verifying the model for short or thick cylinders.[18]

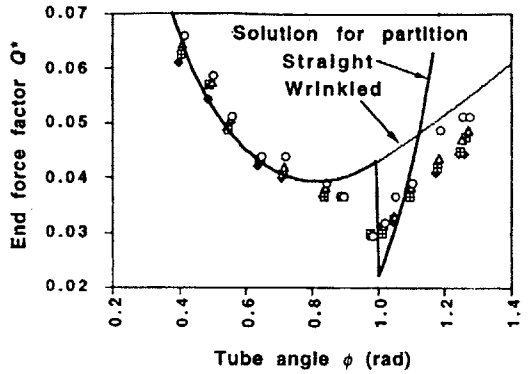


Figure 2-15. Load factor $Q^*=Q/(\pi pR_0^2)$ vs. bending angle ϕ for bending an inflated tube. Circles, triangles, squares and diamonds indicate pressures of 4905 Pa, 6900 Pa, 8800 Pa, and 11000 Pa, respectively. Lines indicate model predictions.[18]

The beam model and the membrane model represent two limiting cases in the physical processes that govern space suit joint mobility. The beam model assumes that elasticity dominates the pressurized cylinder bending process and gas compression plays no role, while the membrane model assumes that gas compression is dominant and the cylinder walls never stretch. Comparing these two models to experimental data illustrates the relative contributions of gas compression and elasticity to space suit joint mobility. Although space suit joints generally experience large deflections that are more consistent with the geometry predicted by the membrane model, space suit segments that are bent about their stiff axis, such as the hip abduction joint described in Section 2.2, may be consistent with the beam model. It is important to note, however, that the theoretical models discussed in this section deal only with uniform cylinders. Because space suit joints have complex, non-uniform structures, as discussed in Section 2.2, neither of these modeling approaches may be appropriate to describe space suit joints. This thesis addresses theoretical modeling questions by comparing the beam and membrane models' predictions to experimental data collected with a space suit, and investigates ways of extending or modifying the beam and membrane models to explain the torque-angle relationship for space suit joints.

2.5 Work Envelope

2.5.1 Work envelope overview

Because EVA operations carry a high cost in time, money, risks to personnel and limited opportunities, extensive prior planning, training, and rehearsal are done for each EVA. The EVA is then executed according to the detailed plans that were made beforehand. One of the critical issues that is addressed in planning for EVA's and evaluating EVA worksites is whether the EVA crew can reach and comfortably work in the designated worksite. Reach considerations are important because, in a microgravity environment, astronauts must restrain their bodies in order to exert forces or moments on other objects. Generally, body restraint in EVA is accomplished by foot restraints, particularly when tasks require apply-

ing significant forces. The articulated portable foot restraint (APFR) used on the space station can be adjusted to point in many orientations, but it can only be attached to the space station's structure at discrete locations which were set when the space station was designed. Because of the constraints on foot restraint placement, it is essential to determine in the planning process whether a work site is reachable from an available foot restraint location. Work envelope analysis is useful in addressing this issue.

Reach and work envelope analysis is an application of anthropometrics to workspace design. A reach envelope is the region in three-dimensional space that a person can reach. The work envelope is a subset of the reach envelope, representing the volume in which a person can comfortably work. Reach and work envelopes depend on the size and flexibility of the individual. A standard practice is to size workspaces to accommodate the reach and work envelopes of individuals at the extremes of the expected size range, for example, 5th percentile females and 95th percentile males.[46]

Most work envelopes are determined experimentally, by measuring how far people of different sizes can reach and obtaining subjective information about the difficulty of working with the hands in different locations. An alternative approach to reach envelope analysis has been developed recently, which uses robot kinematic analysis methods to determine the boundaries within which a person can reach with prescribed limits on joint ranges of motion.[28, 29, 30, 31, 50]

The work envelope analysis section of this literature review presents the NASA EVA work envelope, discusses the differences between the NASA work envelope and others in the literature and describes how the NASA work envelope is used in EVA worksite analysis. Lastly, a computational method for determining work envelopes is described. In Chapter 5 of this thesis, the computational reach envelope method discussed in the literature review is extended to include space suit properties and space suited work envelopes are calculated.

2.5.2 The NASA work envelope

The NASA EVA work envelope appears in NASA Standard 3000[2] and NSTS 07700[1]. Two-handed work envelopes were determined in neutral buoyancy simulation for 5th percentile males and 95th percentile males. The one handed and two handed work envelopes, shown in Figure 2-16[2], are cylinders centered on the body centerline at approximately shoulder height. Interestingly, NASA Standard 3000[2] states that the 5th percentile crewmember's two-handed work envelope should be used as the design two-handed work envelope, while the 95th percentile crewmember's two-handed work envelope should be used as the design one-handed work envelope.

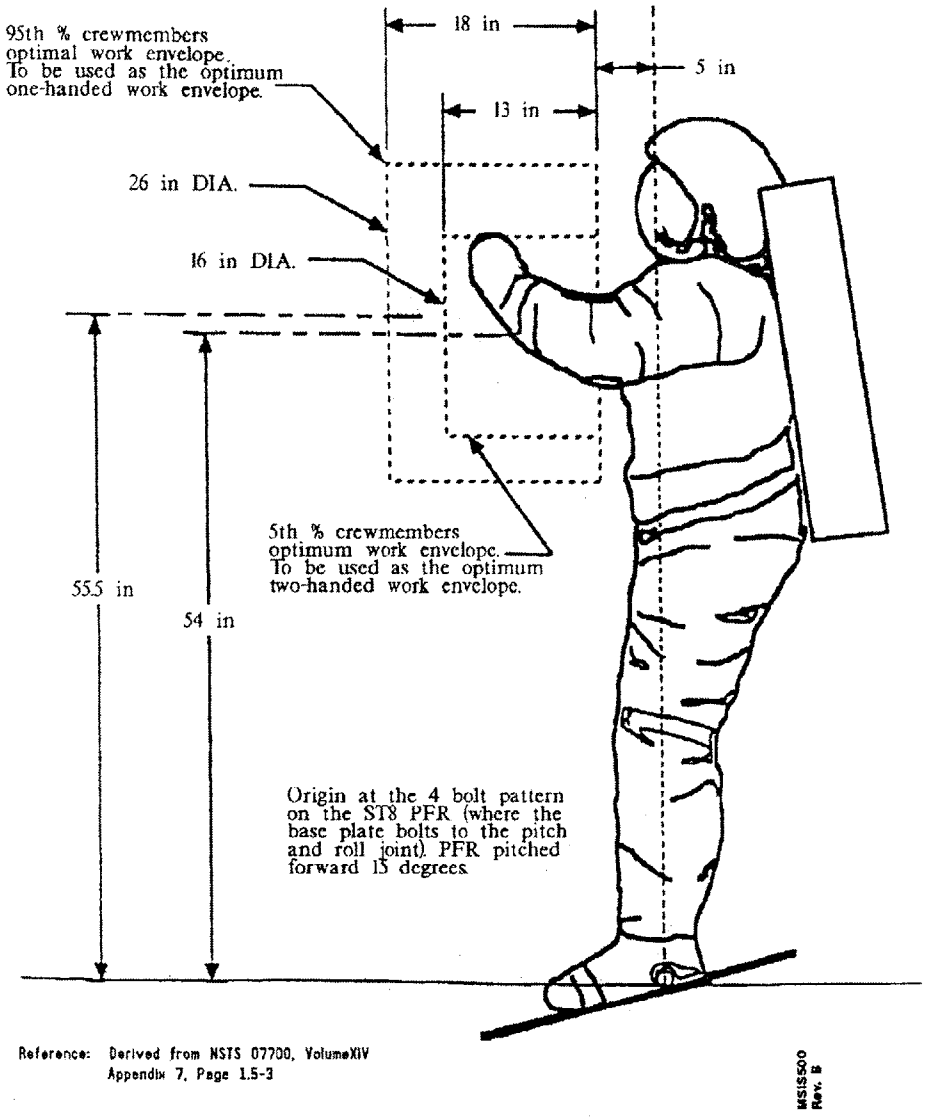


Figure 2-16. EVA work envelope, NSTS 07700. [1]

As the following two examples show, NASA's radially-symmetric one-handed work envelope shape is inconsistent with other one-handed reach and work envelopes found in the literature (Figure 2-17 and Figure 2-18)[46], which show inaccessible areas at the vertical extremes near the body centerline and extend much further laterally than vertically. The NASA one-handed work envelope probably does not include reachable areas beyond its the left and right extremes and may include non-reachable areas at the top and bottom near the body centerline.

An example of a standing reach envelope for a 5th percentile female wearing ordinary clothes is shown in Figure 2-17.[46] Contour lines are shown for areas that may be reached based on distance in front of the shoulders. In three dimensions, the reach envelope

resembles two partially-overlapping ellipsoids centered on the shoulders. The maximum vertical extent of the reach envelope is only about 60% of its maximum horizontal extent.

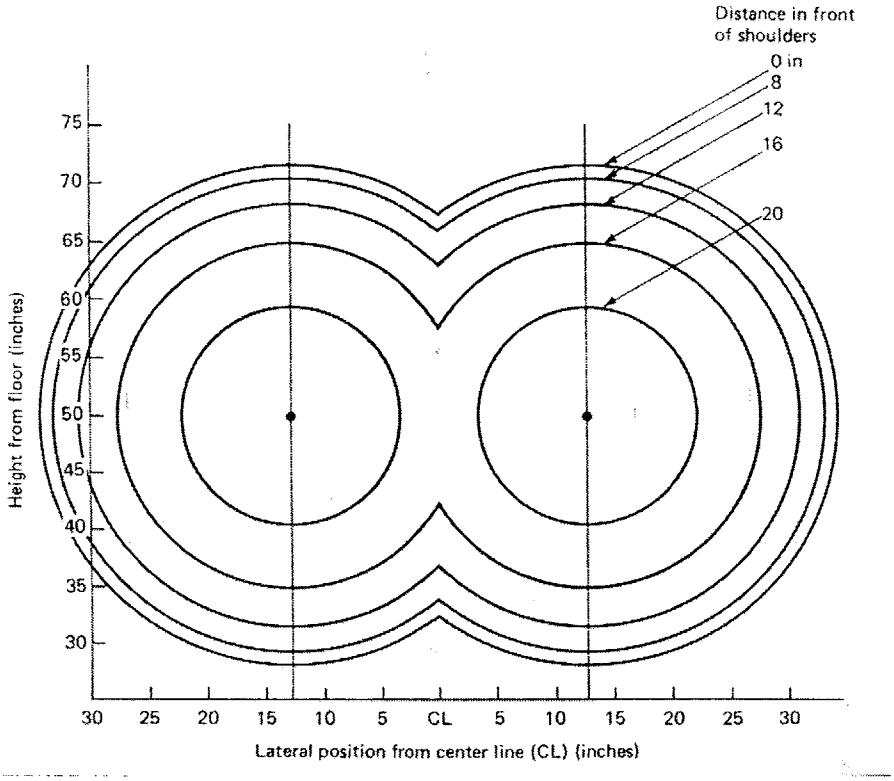


Figure 2-17. Standing reach envelope for 5th percentile female. [46]

An example of a seated work envelope in the horizontal plane is given in Figure 2-18.[46] The figure is a composite of work envelope recommendations from two investigators, showing both normal and maximum work envelope boundaries. The maximum work envelope represents areas in which it is possible, but undesirable to work. Similar to the reach envelope shown in Figure 2-17, the work envelope in Figure 2-18 is symmetrical about the body centerline and bounded by sections of partially-overlapping circles centered on the shoulders.

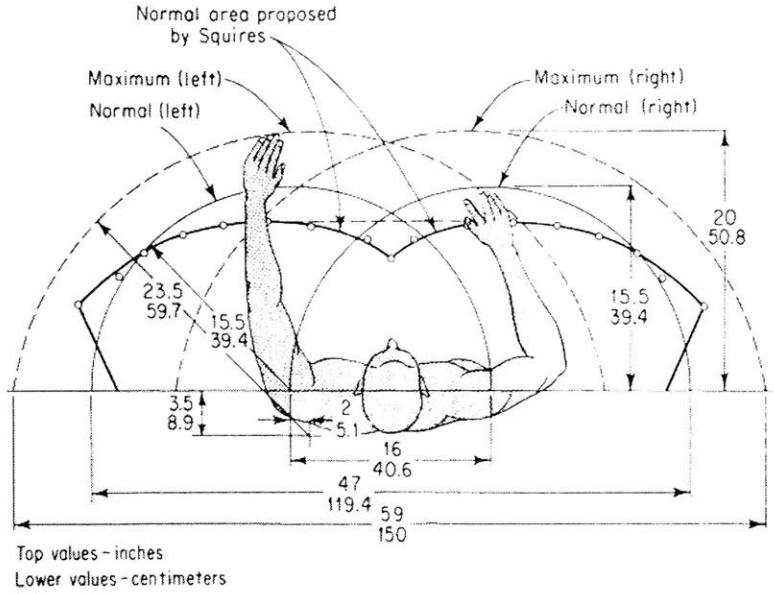


Figure 2-18. Sitting work envelope in the horizontal plane. [46]

One handed work envelopes in the literature have a characteristic shape, with their largest dimension in the left-right direction, corresponding to the span of a person's outstretched arms and inaccessible areas near the centerline and forward limit, at the bottom and top. The NASA work envelope has a markedly different shape. Based on qualitative features of work envelopes in the literature, it is probable that the NASA work envelope excludes some reachable areas on the left and right sides and includes some difficult to reach areas near the centerline, at the top and bottom.

EVA worksite analysis for the ISS is done by Boeing using a work envelope that closely follows the NSTS 07700 work envelope.[25, 4, 44] The primary purpose of Boeing's CAD worksite analysis is to certify that EVA worksites are accessible from the available foot restraint locations on the space station. Validation of the CAD models has been done in neutral buoyancy simulation, particularly in cases where the work area is near the limits of the work envelope. In 90% of the cases tested in neutral buoyancy simulation, it was possible for the suited test subject to reach the work site.[25] The high percentage of successful tests in borderline cases suggests that the NSTS 07700 work envelope is conservative.

2.5.3 Computational methods for work envelope analysis

A number of investigators have used inverse kinematic analysis to calculate reach and work envelopes.[28, 29, 30, 31, 50] Inverse kinematics refers to the process of determining the joint angles that are required to place an end effector at a specified point in space. An important advantage of inverse kinematics methods for determining reach and work envelopes is that anthropometric data such as body segment lengths and joint ranges can be used as inputs to the kinematic model, so that new experiments do not have to be con-

ducted for each population group. Jung et al.[28] were interested in inverse kinematics methods in reach envelope analysis, rather than empirical methods, because they wanted to develop an reach envelope database for a Korean population without having to conduct extensive experimentation.

One of the difficulties associated with inverse kinematics methods is that the human arm is kinematically redundant, or underdetermined. Three coordinates of wrist position determine four joint angles. As a result, the joint angle solution is not unique in most cases. Although both heuristic methods[28] and optimization methods[7] have been applied to resolving kinematic redundancy in human motions, there is no generally-accepted method to determine which of the possible body configurations a person will choose. Another issue that must be considered in developing an inverse kinematics method is that joint angle limits for the shoulder and other multi-degree of freedom joints are correlated.[31] A good inverse kinematics method should incorporate methods for resolving kinematic redundancies and determining whether proposed joint angles fit within correlated limits.

Korein[31] reported on a geometrical method for determining joint angles for a 4 degree of freedom arm as a function of wrist position. He pointed out an important simplification: the elbow angle depends only on the distance between the shoulder and target wrist position. With the elbow angle fixed and no limits on the other joints, the locus of possible elbow positions is a circle. Limits on shoulder flexion, abduction and twist angles reduce the elbow locus circle to one or more arcs on the circle, which represent configurations that are both consistent with the prescribed joint limits and place the wrist on target.

Later, other investigators implemented Korein's arm inverse kinematics model. Kamper and Rymer[30] used Korein's method to investigate the amount of kinematic redundancy the human arm has for normal and reduced joint ranges of motion. They determined that the locus of possible elbow positions usually spans 30 deg-80 deg, depending on the location of the target wrist position. Jung et al.[28, 29] used Korein's inverse kinematics method and chose configurations that minimized shoulder angle displacement from zero. After comparing the inverse kinematics results to their experimental data, they concluded that joint angles determined by inverse kinematics agreed with those chosen by human subjects.

2.6 Summary

The EMU space suit, which is currently used for NASA EVA's, has both hard fiberglass and soft fabric components. Mobility features, such as pleats that open as joints bend and rotational bearings, are built into all modern space suits. Without these mobility features, a person in a space suit would be virtually immobile. Even though space suits are designed to allow mobility, they restrict the wearer's motion in significant and complicated ways.

Limited space suit joint torque-angle data has been reported in the literature. Studies that used human subjects wearing space suits reported higher torques than studies that measured torques on joints of empty, pressurized space suits. Higher torques with human subjects may be expected because contact between the space suit and the wearer's body

affects the deformed shape of the space suit, but it is unclear whether the discrepancy in experimental results is due to shortcomings in experimental methods or actual differences in observed torques.

This thesis uses several mathematical and physical modeling techniques which were originally developed for other applications but are relevant to modeling the torque-angle characteristics of space suit joints. Mathematical modeling techniques allow torques to be predicted from angle histories with accuracy, but, since they do not incorporate physical principles, they do not contribute insights for designing space suit joints with better mobility characteristics. Three mathematical techniques for modeling hysteretic systems, originally used for modeling magnetization and shape memory alloy actuators, were evaluated for possible use. The Preisach model was chosen because it can produce output curves similar in shape to the space suit torque vs. angle curves and its identification process is relatively simple.

Physical models, in contrast to mathematical models, can lead to design insights. Two physical models of the bending characteristics of inflated cylinders may be relevant to space suit joints. Beam models treat the inflated cylinder as a beam with a fabric shell that stretches, resulting in small displacements with large loads. Membrane models treat the fabric shell as an inextensible membrane. Large bending deflections of the cylinder result in shape and volume changes. Space suit joint deflections fall between the load-displacement regions covered by these two models.

Finally, the literature review discusses an operational human factors performance metric, the space-suited work envelope and methods for predicting work envelopes based on space suit mechanics. The current NASA space-suited work envelope, which is used in planning EVA's, differs qualitatively from other work envelopes found in the literature. A recently-developed computational technique allows suited work envelopes to be calculated using mathematical models of the torque-angle characteristics and kinematics of a space suit.

Chapter 3 *Space suit mobility database*

3.1 Experiment overview

The objective of the experimental portion of this thesis was to obtain a quantitative database of joint angles and torques required to move a space suit's joints, under realistic conditions. The impossibility of directly measuring joint torques in suited humans necessitated an indirect measurement approach, using both human subjects and an instrumented robot to obtain torque data.

The human test subjects carried out arm and leg motions both wearing the space suit and not wearing the space suit, supplying realistic joint angle trajectories for each of 20 motions. The joint angle trajectories produced by the human subjects were then used as command inputs for the robot, so that the robot imitated the humans' motions while torques were measured at each of the robot's joints. Torques on the robot's joints due to the weight of the robot and space suit, were subtracted, resulting in a consistent set of joint angle and torque data.

3.2 Space suit description

The space suit used in these experiments was an Extravehicular Mobility Unit (EMU) supplied under contract by Hamilton Sundstrand. A size large hard upper torso (HUT) was used.The space suit that was tested was as close as feasibly possible to a space suit that would be used in flight, however there were some differences between this space suit and a flight-qualified space suit. The space suit tested was designated as Class III hardware, approved for demonstrations or non-hazardous testing. Class III space suits are known to be less stiff than Class I, or flight qualified suits, because they are generally older and have been used more. Additionally, the scye bearings on the space suit used in these experiments, which connect the arms to the HUT, were mounted on a single-fold bellows, which allows some additional shoulder range of motion. This bellows configuration, known as a pivoted HUT, is no longer used in flight.

Life support for the suited test subjects was provided by supplying breathing air on a non-recirculating vent loop. Air was supplied to the suit inlet by scuba tanks and exhausted from the suit to maintain a minimum air flow rate. One set of scuba tanks lasted approximately one hour. A volumetric mockup was substituted for the portable life support system

(PLSS), resulting in an overall weight for the space suit and PLSS mockup of 102 lb. (46.4 kg). In all cases, the space suit was pressurized to 4.3 psi (30 kPa) above ambient pressure.

3.3 Data collection with human subjects

3.3.1 Subjects

Four male subjects participated in the experiment; two of the subjects were experienced in 1-g testing of the EMU. Subjects performed arm and leg motions in two sessions. The first session was conducted without the space suit and the second session was performed with subjects wearing the space suit, one week later. One of the inexperienced test subjects did not conduct the simple motions described in Section 3.3.2 for space-suited data collection because of a scheduling constraint. The experiment was approved by MIT's Committee on the Use of Humans as Experimental Subjects and informed consent was obtained from each subject prior to each experimental session.

3.3.2 Motions

Human test subjects carried out arm and leg motions while wearing the space suit. Twenty motions were used for human and robot data collection. The motions, listed in Table 3-1, included 11 simple motions that isolated each of the robot's degrees of freedom (wrist rotation was not used) and 9 complex, free-form motions, which included reaching over the head, across the body and to an object 50 cm above the floor. Subjects also walked on a treadmill and traversed a 12 cm high step to simulate planetary EVA tasks. Trial ordering was randomized, with a first phase of simple motions followed by a second phase of complex motions. Subjects were instructed to move through the maximum range that they could comfortably reach. They were not instructed to attempt to achieve the maximum range of motion. Free-form motions were included for two reasons: to obtain data on multi-joint, non-planar motions and to compare the subjects' choices of arm and leg positions between suited and unsuited conditions.

Table 3-1: Arm and leg motions used in experiment

Simple Motions	Complex Motions
Shoulder flexion	Arm swing forward-backward
Shoulder abduction	Arm swing side to side
Humerus rotation	Leg swing forward backward
Elbow flexion	Leg swing side to side
Hip flexion	Overhead reach
Hip abduction	Cross-body reach
Thigh rotation	Low reach
Knee flexion	Locomotion over 12 cm step
Ankle rotation	Locomotion on treadmill

Table 3-1: Arm and leg motions used in experiment

Simple Motions	Complex Motions
Ankle flexion	
Ankle inversion	

Figure 3-1 shows an example of a subject performing the low reach trial. In the low reach trial, the subject's task is to reach down and to the left, touching the corner of the step with his right hand.

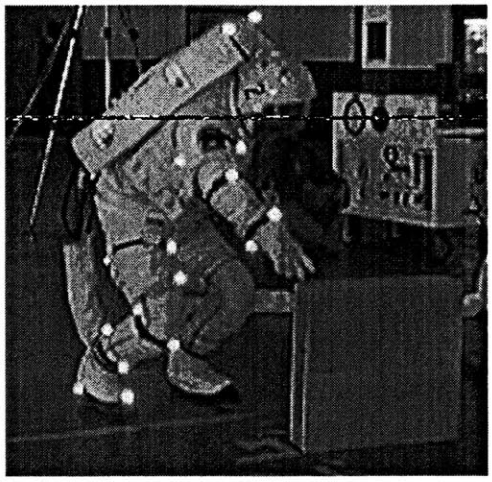


Figure 3-1. Low reach trial.

3.3.3 Data acquisition

Kinematic data was collected on the subjects' right arm and leg, using a Multitrax optical motion capture system (Adaptive Optics Associates, Cambridge, MA). Reflective markers were placed on the subject's right arm, right leg and the mockup PLSS, as shown in Figure 3-1. Outputs from the motion capture system were x, y, and z positions of each of

the markers in a laboratory-based coordinate system. Marker positions for suited and unsuited configurations are shown in Figure 3-2.

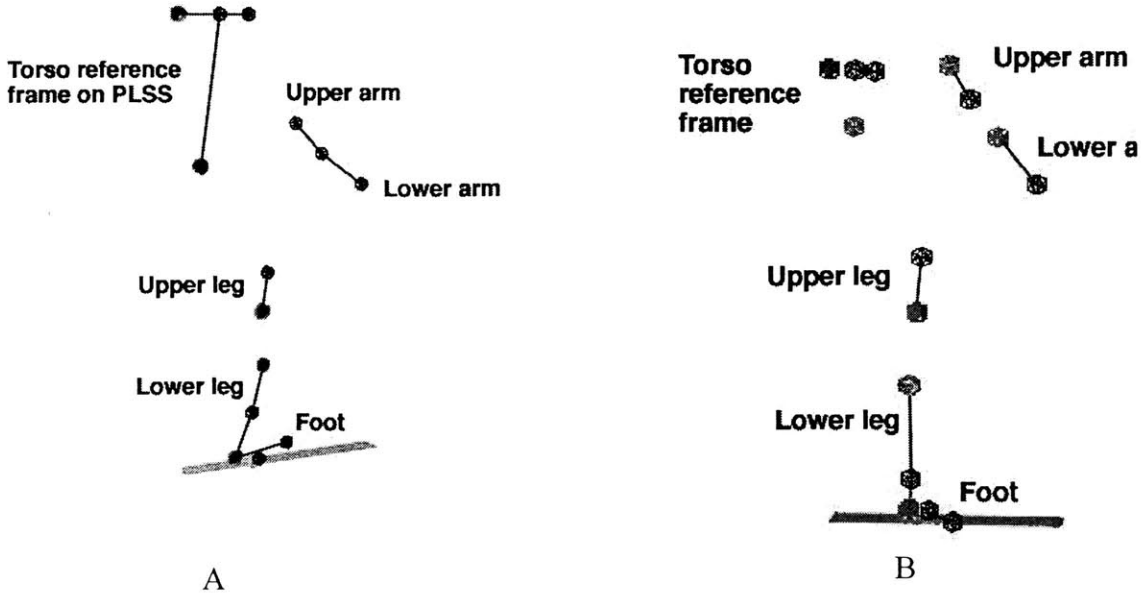


Figure 3-2. Marker configurations for A.) suited configuration and B.) unsuited configuration.

The Cartesian positions of the markers were converted into joint angles for 11 arm and leg joints, which completely describe the kinematics of the right arm, leg, and foot. Joint angles are measured with respect to the body segment closer to the torso or with respect to the torso in the case of the shoulder and hip joints. When all joint angles are zero, the arm and leg are both parallel to the torso and the palm of the hand and the toe both point forward. Each of the joint angles, with its zero position and positive direction, are shown in Figure 3-3. Figure 3-3A shows the five flexion angles: shoulder flexion (sf), elbow flexion (ef), hip flexion (hf), knee flexion (kf), and ankle flexion (af). Figure 3-3B shows shoulder abduction (sa), hip abduction (ha), and ankle inversion (ai). Figure 3-3C illustrates the thigh rotation angle (tr), when hf=90 degrees and kf=90 degrees. Figure 3-3D and Figure 3-3E are both overhead views, showing, in Figure 3-3D, the humerus rotation angle (hr), with ef=90 degrees, and in Figure 3-3E, the ankle rotation angle.

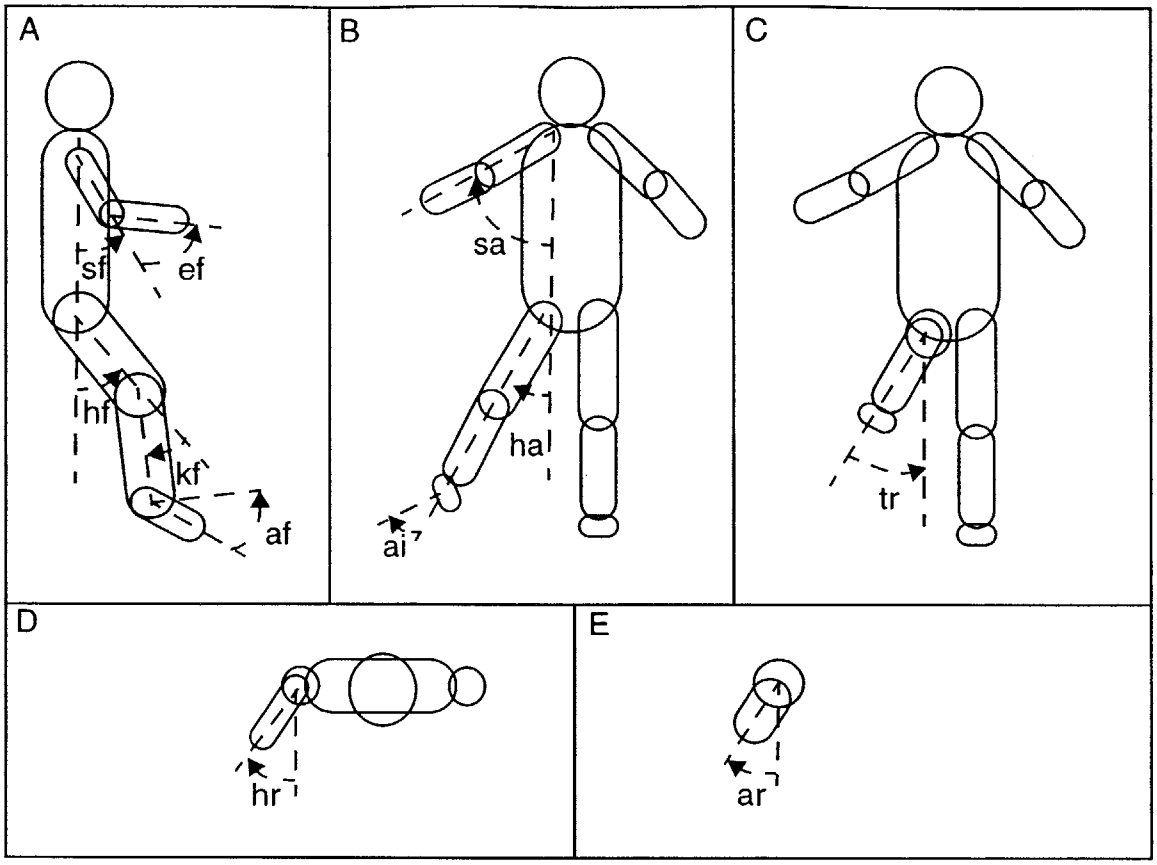


Figure 3-3. Joint angle conventions used for kinematic data for human test subjects and robot.

3.4 Robot data collection

3.4.1 Robotic Space Suit Tester

Sarcos, Inc. (Salt Lake City, UT) built a robot for space suit mobility testing under the Small Business Innovative Research (SBIR) program for NASA Johnson Space Center. The robot, which is called the Robotic Space Suit Tester (RSST), was loaned to MIT by NASA in 1998 for space suit mobility research.

The RSST has 12 hydraulically actuated joints on the right arm and leg and 12 posable joints on the left arm and leg. At each actuated joint, potentiometers measure joint deflection and strain gauge load cells measure torque. Locations of the robot's joints are indicated in Figure 3-4 and Table 3-2.

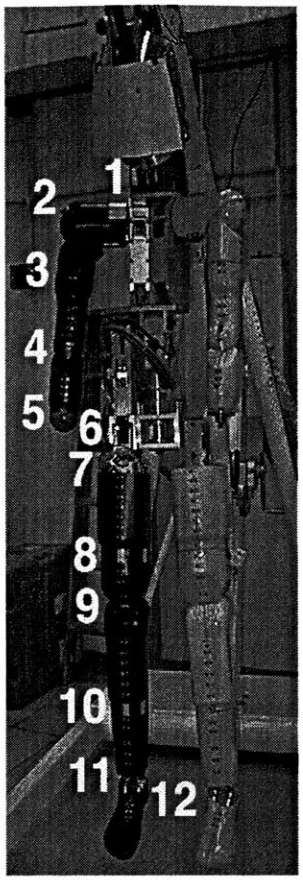


Figure 3-4. RSST joints.

Table 3-2: RSST joint degrees of freedom

Number	Joint
1	shoulder flexion
2	shoulder abduction
3	humerus rotation
4	elbow flexion
5	wrist rotation
6	hip flexion
7	hip abduction
8	thigh rotation
9	knee flexion
10	ankle rotation
11	ankle flexion
12	ankle inversion

The robot is suspended from a crane and supported at two points: a bolt at the "head" and a cable attached to the back, near the center of the torso. The torso is considered to be the ground segment and remains in a fixed position, while the arm and leg move with respect to the torso. Adjustment of the cable supporting the torso allows the robot's pitch angle to be adjusted and two positioning joints within the torso allow forward/backward and left/right bending of the torso segment.

The actuated joints of the robot are powered by an MTS model 510 hydraulic pump. Hydraulic fluid circulates in a loop from the pump, through each robot joint actuator, then back to the pump. All of the hydraulic lines and electrical cables exit the robot through a seal at the top so that a space suit may be pressurized on the robot.

Control of robot motions is accomplished through analog circuitry and two PC computers, as shown in Figure 3-5. An EPC-5 486 computer supplies the user interface, and an EPC-6 386 computer performs the lower-level control functions. Analog control loops are closed on position, velocity, and torque on circuit boards in the Advanced Joint Controller (AJC) cage, with control gains supplied by the EPC-6 through a digital interface. The AJC cage uses a modular architecture, with 12 boards, one for each joint. The EPC-5 computer runs a Microsoft Windows 3.1 based user interface, called the Robotic Space Suit Tester Application (RSSTA), which passes commands to the robot at an update rate of 5 Hz and records position and torque data to disk. At the time the robot was built, the analog/digital command and control architecture allowed for faster loop closure than would have been possible by closing control loops in the digital domain in a computer. Although the analog position, velocity, and torque servo loops are hard-wired on the circuit boards, the computer interface does allow for time-varying control gains, a capability which has not been implemented in this study.

The combination analog-digital control system is very robust to computer crashes, which happen periodically in Windows 3.1. If the EPC-6 computer stops receiving inputs from the EPC-5 computer, it holds the current commands to the robot's joints. All of the control loops remain closed, so stability and performance of the robot's control system are preserved when the EPC-5 computer crashes. In addition, the modular construction of the analog circuitry in the AJC cage aids in troubleshooting. Boards for different joints may be exchanged for diagnostic purposes by changing jumper settings and inserting the boards in different slots in the AJC cage.

The user interface program allows the robot to be commanded in two ways: joints can be manually positioned by clicking on arrows in the positioning window or multi-joint trajectories can be loaded from files and executed. Joint positions and torques, sampled at 5 Hz, can be plotted on the screen and saved to files. Trajectories for the robot to follow may be created interactively, by manually positioning the robot and saving a series of trajectory points. The list of robot positions can be saved as a file and run at user-set speeds. Trajectories can also be created outside of the RSSTA application, as described in Section 3.4.2.

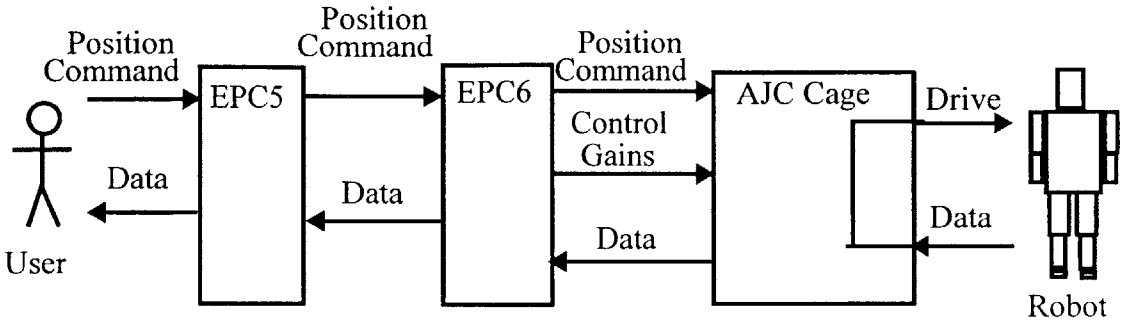


Figure 3-5. RSST command and control.

3.4.2 Robot operations

3.4.2.1 Trajectory generation

The RSST was designed to approximate the joint axes and range of motion of a human arm and leg, but due to geometrical constraints in its construction, the RSST has less range of motion in some joints than humans do. In particular, limits on two of the shoulder degrees of freedom and one of the hip joints are quite different from those of human joints. The shoulder flexion joint does not allow negative shoulder flexion angles, which bring the arm past vertical, behind the body. The shoulder abduction joint also does not allow negative shoulder abduction angles, which bring the arm towards the body. Like the shoulder flexion joint, the hip flexion joint does not allow the hip to be flexed beyond vertical, behind the body. A complete list of the robot's range of motion for each joint is given in Table 3-3, using the joint angle definitions from Figure 3-3.

The robot's limited range of motion prevents the raw kinematic data files collected from the human test subjects from being used directly to command the robot. Commanding the robot with a joint angle that is beyond its range causes the joint to move to its limit and abruptly stop. To avoid spurious torque data caused by sudden changes in robot joint velocity, the data files collected from the human subjects were pre-processed so that all of the commanded joint angles were within the robot's range of motion.

Table 3-3: Robot joint angle limits

Joint	Maximum angle (deg)	Minimum angle (deg)
Shoulder flexion	180	-15
Shoulder abduction	90	0

Joint	Maximum angle (deg)	Minimum angle (deg)
Humerus rotation	90	-90
Elbow flexion	130	0
Hip abduction	45	0
Hip flexion	100	0
Thigh rotation	22	-22
Knee flexion	130	0
Ankle rotation	20	-20
Ankle flexion	30	-45
Ankle inversion/eversion	20	-20

There are several methods that may be used to force commands to the robot to stay within the robot's limits. The simplest method is to uniformly reduce the amplitude of the entire joint trajectory so that the maximum displacement falls within the limits. The problem with this approach is that it modifies the joint angle values for the entire data file, even though the vast majority of joint angle values are permissible. A more complicated scheme that minimizes changes to valid joint angles and discontinuous velocities while forcing the output joint angle trajectory to remain within the robot's limits was implemented instead.

A nonlinear scaling method was used to eliminate robot joint commands that are outside the limits, while leaving non-extreme trajectory angles unchanged. This scheme forces the robot's commands to be in bounds by compressing the out-of-bounds angle values into a region near, but inside of, the joint angle limits. Compressing the out-of-bounds data rather than deleting it reduces velocity discontinuities, which may cause erroneous torque measurements.

The pre-processing scheme takes a desired robot command, $x(t)$, as an input, multiplies it by a scaling factor $f(t)$, and outputs $y(t)$, the modified command, which does not exceed the robot's limits for that joint. Two robot joint limit values were defined, as shown in Figure 3-6: b is the maximum possible robot joint displacement, and a is 90% of the way from the midpoint of the robot's joint range to b. Values of the input, $x(t)$, which exceed b are reduced to fall between a and b, while input values which are less than a are not changed. The maximum value of the input, $x(t)$, is called x_m. A scaling factor $f(t)$ that accomplishes this transformation is given in Equation 3-1, where k and c are constants whose values are set by matching boundary conditions.

$$x(t) < a \qquad f(t) = 1$$

$$x(t) > a \qquad f(t) = \frac{1}{1 + k\dfrac{(x-c)}{b}} \qquad \text{Eq 3-1}$$

The constants k and c are set so that at $x=a$, $y=x$ and at $x=x_m$, $y = b$. The $f(t)$ that satisfies these conditions is given by Equation 3-2.

$$y(t) = f(t)x(t)$$

$$x(t) < a \qquad f(t) = 1$$

$$x(t) > a \qquad f(t) = \frac{1}{1 + \dfrac{(x_m - b)}{(x_m - a)}\dfrac{(x(t) - a)}{b}} \qquad \text{Eq 3-2}$$

The results of applying this transformation are shown in Figure 3-6. For $0<t<t_1$, the input value, x, is less than a. The corresponding $f(0<t<t_1)$ is equal to 1 and consequently, the output $y(0<t<t_1)=x(0<t<t_1)$.When x is greater than a, for $t_1<t<t_2$, f is less than 1, and reaches a minimum of $f=b/x_m$ when x is at its maximum of x_m. Thus, the maximum output value is equal to b, the maximum allowable command. The plot of $y(t)$ vs. t illustrates that y is always less than b and y is equal to the input value, x, when x is less than a.

Equation 3-2 is valid only for non-zero, positive a and b. For negative or zero a and b, the same transformation can be accomplished by mirroring $x(t)$ about the joint range mid-point, m, using Equation 3-3 to obtain $x'(t)$. The mirrored input $x'(t)$ is then transformed according to Equation 3-2, and the output, $y'(t)$ is then mirrored back according to Equation 3-3.

$$x' = 2m - x \qquad y = 2m - y' \qquad \text{Eq 3-3}$$

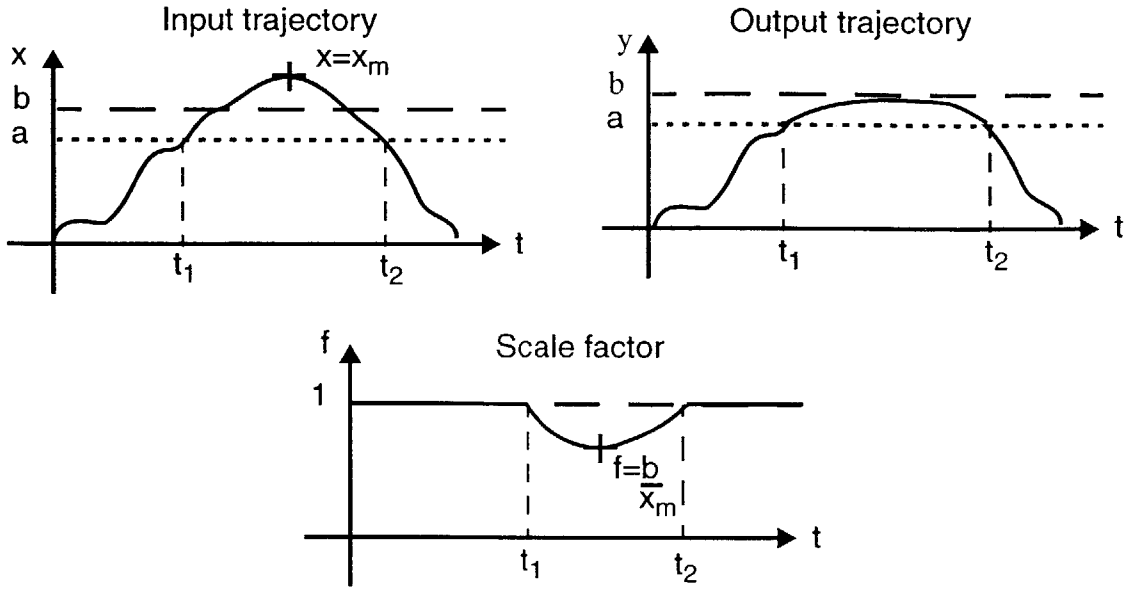


Figure 3-6. Trajectory motion limiting algorithm.

The pre-processed commands for the robot are then formatted into trajectory files that the RSSTA application can read. The robot's trajectory files are ascii text files, which are normally created in the RSSTA application when the user manually positions the robot and saves trajectory points. Trajectory files were created from pre-processed data with the Matlab script trajconvert.m, which emulated the file format of the trajectory files created in RSSTA. Each trajectory file begins with three lines of text. Each line ends in a line feed character (ctrl-j or ascii 10). Following the three text lines are the commanded joint positions. Each line contains joint positions for the 12 joints, with up to 3 digits before the decimal point and 1 digit after it. Numerical values are aligned horizontally on decimal points. Numbers are separated by spaces so that there are always 6 characters between each successive decimal point, 4 characters before the first decimal point in the line, and 2 characters after the last decimal point. Every line ends in a line feed (ctrl-j or ascii 10) and the end of file character is a carriage return (ctrl-m or ascii 13). If a trajectory file deviates from this format, either the RSSTA application will hang or it will load a short trajectory with incorrect joint angle values.

3.4.2.2 Trajectory speed setting

The speed at which the robot moves through a trajectory is set by the operator in RSSTA at the time the trajectory is executed, which allows the same trajectory to be run at various speeds. Speed setting is accomplished by entering a single velocity value in degrees per second, between 1 deg/sec and 50 deg/sec. The operator-entered velocity is assigned to the maximum velocity of the robot joint that has the largest angle range in that trajectory. For this experiment, the robot should move through the trajectory at the same speed as the human subject did. In order to accomplish this, the joint with the maximum range and its

maximum velocity would have to be determined beforehand, stored, and correctly entered by the robot operator each time the trajectory is run.

In order to reduce the likelihood of errors, a different method was used to set trajectory speeds. The wrist rotation joint mechanism had been removed from the robot, to allow the robot's arm to fit in the EMU sleeve, as described in Section 3.4.2.4. A dummy command to the wrist rotation joint is included in all trajectory files for motion from -90 degrees to +90 degrees at a constant velocity of 50 deg/sec. The wrist rotation joint then has the maximum range and its velocity is set by the operator's speed input. To run trajectories at the same speed as the human subjects moved, the operator enters 50 deg/sec for all trajectories.

3.4.2.3 Torque limits

Torque limits for each joint are set and enforced by the RSSTA software. When a motion trajectory is running and one or more of the joint torques exceed the limits, the robot's motion stops and the robot is commanded to return to the starting position of that trajectory. A dialog box is displayed indicating the joint that triggered the torque limit violation. Only one dialog box appears, even if multiple torque violations occur. Torque limits are not checked in manual operation. The joint torque limits used in the experiment are listed in Table 3-4. The limits used for wrist rotation are necessary because the wrist rotation load cell is disconnected and the open circuit returns a constant value of -819 in*lb for wrist rotation torque.

Table 3-4: Robot torque limits

Joint	Minimum Torque (in*lb)	Minimum Torque (Nm)	Maximum Torque (in*lb)	Maximum Torque (Nm)
Shoulder flexion	-400	-45.2	350	39.6
Shoulder abduction	-400	-45.2	350	39.6
Humerus rotation	-250	-28.3	250	28.3
Elbow flexion	-350	-39.6	275	31.0
Wrist rotation	-900	-102	500	56.5
Hip abduction	-400	-45.2	2000	226
Hip flexion	-1300	-147	1300	147
Thigh rotation	-500	-56.5	500	56.5
Knee flexion	-750	-84.8	750	84.8

Table 3-4: Robot torque limits

Joint	Minimum Torque (in*lb)	Minimum Torque (Nm)	Maximum Torque (in*lb)	Maximum Torque (Nm)
Ankle rotation	-500	-56.5	500	56.5
Ankle flexion	-700	-79.1	700	79.1
Ankle inversion/ eversion	-500	-56.5	500	56.5

3.4.2.4 Space suit installation

Although the robot had been designed for space suit mobility research, a space suit had never previously been installed. Several modifications were made to the robot to allow the EMU to fit properly and protect the space suit from damage. These included replacement of structural parts in the robot and removal of non-essential assemblies for ease in installing the space suit.

The structural component between the top of the robot's torso and the electrical-hydraulic feedthrough interface, referred to as the neck, was replaced to allow a gas-tight interface with the EMU space suit. The new neck has an angle of 25 degrees between its top and bottom surfaces, replacing the original neck which has a 55 degree angle between its top and bottom, as shown in Figure 3-7.

The neck piece provides the mechanical interface between the EMU and the robot. When the space suit is installed on the robot, an aluminum plate, called the neck plug, is mounted to the neck ring of the HUT, where the helmet attaches. The aluminum plate is bolted to the robot's neck piece, so that the space suit is supported by the neck ring, rather than the inner surface of the HUT at the shoulders. The neck plug has a hole in the center for hydraulic hoses and electrical cables to pass through. Sealing is accomplished with an airtight seal between the EMU neck ring and the neck plug and a rubber gasket between the neck plug and the robot's feedthrough interface, which is bolted directly above the neck plug.

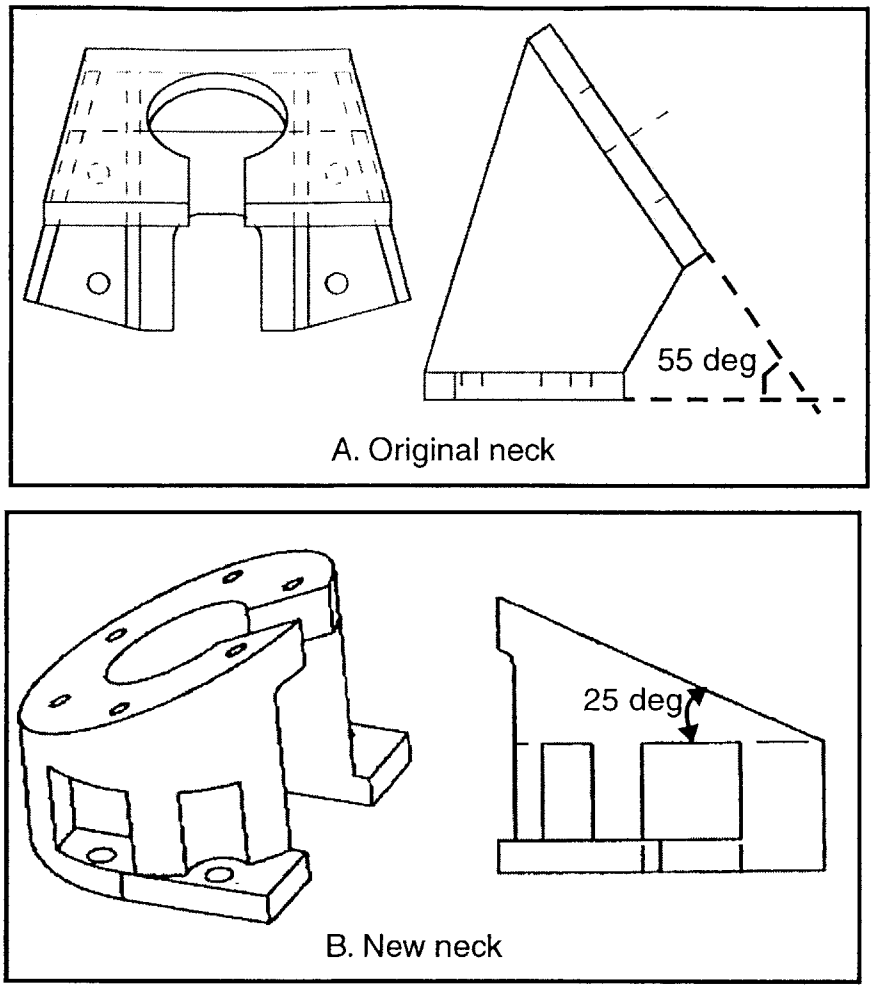


Figure 3-7. A.) Original neck with angle of 55 degrees. B.) New neck with angle of 25 degrees.

In its original configuration, the robot's torso was approximately 2.5 cm too long to allow the HUT and lower torso assembly (LTA) of the space suit to connect. The torso of the robot was shortened by replacing a torso structure piece with a piece that was 3.2 cm shorter. The location of the modified torso piece is shown in Figure 3-8. A screw at the bottom of the robot's torso was also replaced with a low-head version, to further shorten the torso.

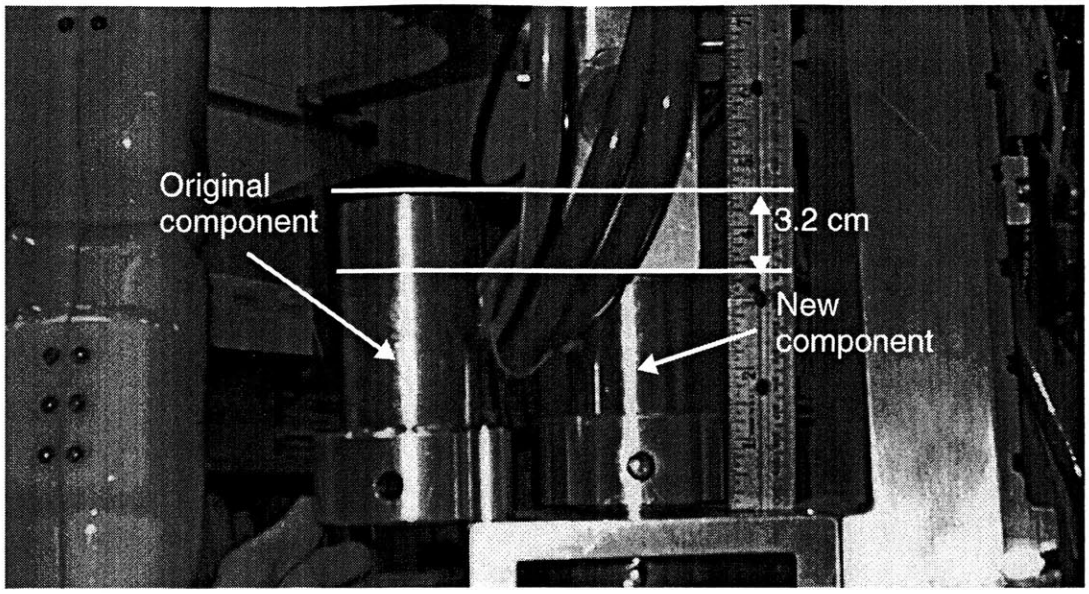


Figure 3-8. Modified torso structure piece, with original, larger torso structure piece at left.

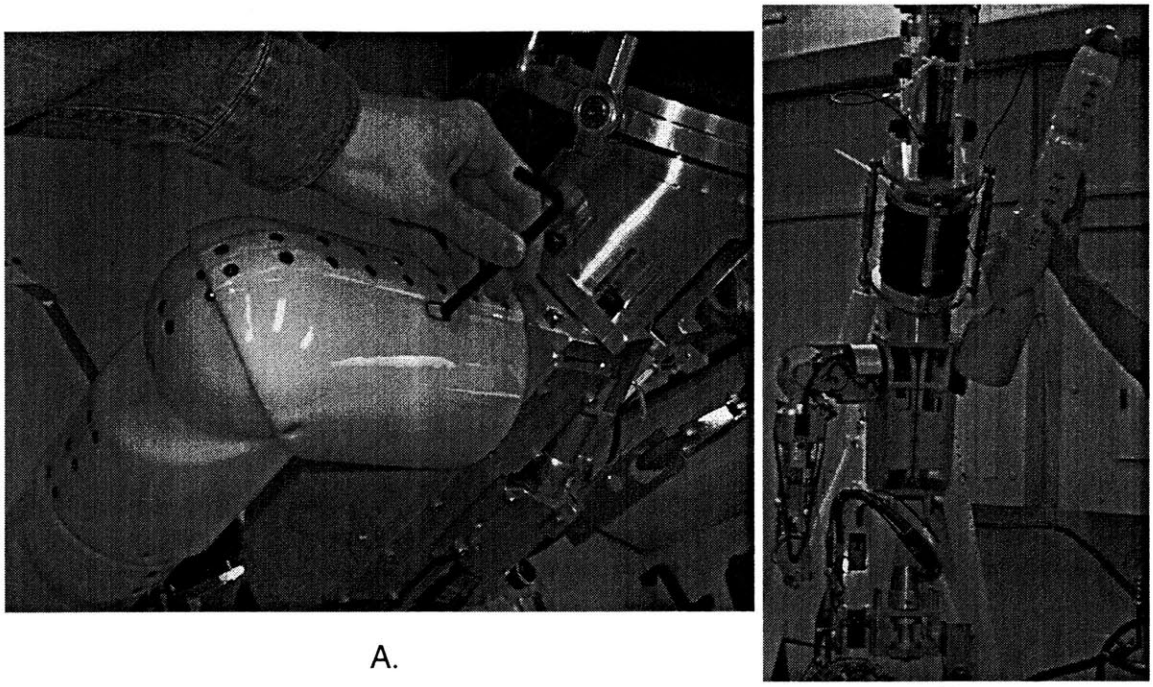
The robot's lower arm also did not fit inside the space suit. The wrist rotation shaft and wrist disk were removed, shortening the lower arm length by 8.9 cm and reducing the arm end diameter from 7.6 to 6.4 cm. As a result, the wrist rotation joint was not functional, but it was never intended to be used in the experiment.

In order to ease installation of the space suit on the robot, and eliminate the need to accurately size both arms and legs of the space suit, the robot's non-actuated left arm was removed at the shoulder flexion pivot. The left foot was also removed at the ankle inversion pivot.

To protect the space suit from sharp edges and pinch points on the robot, a wet suit was placed on the robot before the space suit was installed. Foam padding was also used at the lower end of the plastic shell on the lower leg to prevent the space suit's bladder material from being pinched between the foot shell and the lower leg during ankle flexion motion. A plastic cover was placed over the exposed end of the wrist rotation shaft.

To install the space suit on the robot, the hydraulic lines and electrical cables were disconnected and the robot was disconnected from the crane. The robot's feedthrough interface was then removed, down to the neck piece. With the robot held in a sitting position on the floor, the HUT was first installed on the robot. The arm was brought above the head and the shoulder latch was released by inserting an allen wrench through the access hole in the shell and pushing the latch mechanism, as shown in Figure 3-9. When the shoulder latch is released, the upper portion of the arm, which is normally horizontal, can be rotated to a vertical position so that the arm can be brought through the sleeve of the HUT. Care should be taken to ensure that the shoulder latch is engaged properly following installation of the HUT. The HUT was attached to the robot by bolts that pass through the robot's feedthrough interface and the space suit's neck plug into the robot's neck piece. Using

small lengths of threaded rod to align the bolt holes while the three parts are assembled aids this process. The robot was then reconnected to the crane, hydraulic lines, and electrical cables and raised to its normal position. The LTA, with the right boot removed, was pulled up over the legs and latched to the HUT at the body seal closure and the right boot was replaced.



A. B.

Figure 3-9. Shoulder latches. A.) Latch location. B.) Arm position with latch released.

3.4.3 Position and torque data collection

The EMU used in the human subjects phase of the experiment was installed on the robot, as described in Section 3.4.2.4, and pressurized to 30 kPa (4.3 psi), as shown in Figure 3-10. Air was supplied from a scuba tank to maintain the 30 kPa (4.3 psi) operating pressure in the space suit. Due to a high leakage rate in the robot-space suit interface, one scuba tank lasted approximately 30 minutes. The high leakage rate required a suit depressurization and repressurization to replace the air tank every 30 minutes while data was being collected.

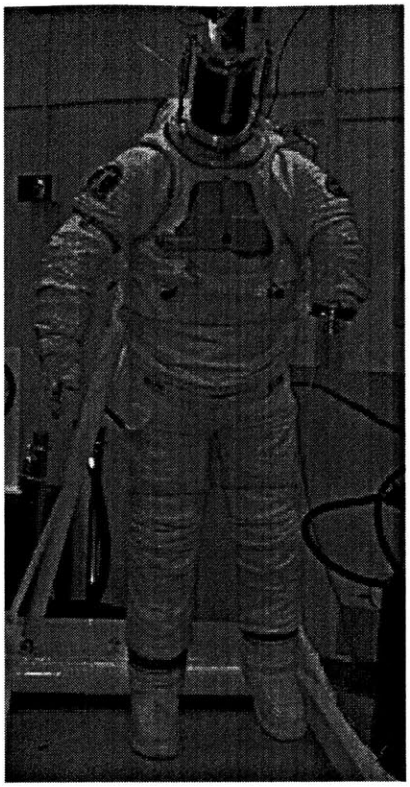


Figure 3-10. EMU installed on robot.

The space-suited robot performed 80 different motions that had been generated by the human test subjects. Robot joint angle and torque data for the 11 robot joints was obtained at both the speed that the human test subjects performed the motions and at half speed, to improve the robot's trajectory-following performance. A total of 247 torque-angle data sets were obtained from the suited robot, including several replications of each motion. In addition to the robot-generated angle and torque data, motion capture data was acquired for at least one replication of each motion, using the Adaptive Optics Associates optical motion capture system that had been used for data collection with human subjects.

The robot also performed the above-listed motions without the space suit under two conditions: wearing the wet suit that had been used in space-suited data collection to protect the space suit from sharp edges on the robot and not wearing the wet suit. The unsuited torque and angle data was collected at full speed and half speed, the same speeds at which the suited data had been collected. Over 300 torque-angle data sets were collected from the robot without the space suit. Figure 3-11 shows in schematic form the nine motion capture and robot-generated data files that originate from one motion of the space-suited human subject. This thesis presents only the space-suited human motion capture data and the robot-generated torque and angle data under suited and unsuited conditions.

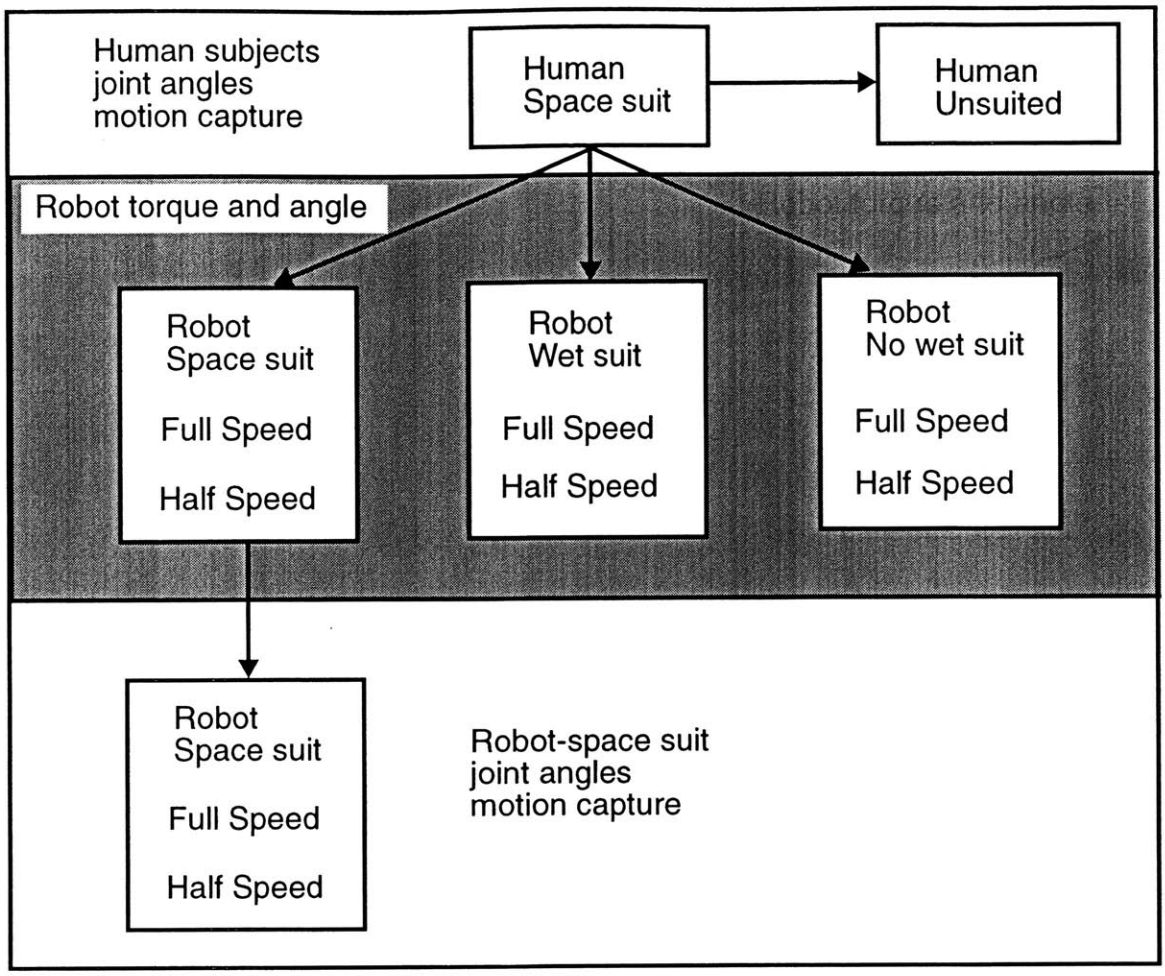


Figure 3-11. Schematic of data collected from human subjects and robot.

3.5 Data reduction

The torque data that was recorded by the robot is a composite of torques required to support the weight of the robot's limb, support the weight of the space suit arm or leg, and bend the joints of the space suit. Only the torques required to bend the space suit's joints are relevant to the space suit model. Both experimental data and kinematic analysis were used to calculate the gravity-induced torques due to the robot and space suit. The gravity-induced torques were then subtracted from the joint torques recorded by the robot.

3.5.1 Robot weight removal

An empirical method was chosen for estimating torques induced by the robot's weight because the robot's mass properties are not documented. Torques caused by the robot's weight were measured by recording torque data while the robot performed all of the same motions that were used in suited robot data collection. The torques caused by the robot's weight were eliminated from the measured torque data by manually aligning the two files in time and subtracting the unsuited torque data from the suited torque data. Data col-

lected with the robot wearing the wet suit only was used as the unsuited torque data. All data presented here has been corrected to remove torques induced by the weight of the space suit and robot.

3.5.2 Space suit weight removal

Torques caused by the space suit's weight were estimated using kinematic analysis techniques, based on both the known mass properties of the space suit and the robot dimensions and joint axes. The kinematic analysis yielded the positions and orientation of each of the robot and space suit segments. Joint torques were calculated based the relative positions of the space suit segment centers of mass and the robot joints.

The geometry of the torque calculation is shown in Figure 3-12. The vector from joint i to the center of mass of segment j is $\underline{r}_{ij}$, $\underline{z}_i$ is the axis of joint i, m_j is the mass of segment j, and g is the acceleration due to gravity. The torque caused by the weight of segment j about a point on $\underline{z}_i$ is given by:

$$\underline{T} = \underline{r}_{ij} \times m_j \underline{g}. \qquad \text{Eq 3-4}$$

Only the torque along $\underline{z}_i$ is measured by the robot's load cell at that joint, so the torque measured by the load cell on joint i is given by Equation 3-5.

$$T_{ij} = (\underline{r}_{ij} \times m_j \underline{g}) \bullet \underline{z}_i \qquad \text{Eq 3-5}$$

The torque on each joint due to the suit weight is the sum of the T_{ij} torques for each segment that is further away from the torso than joint i is.

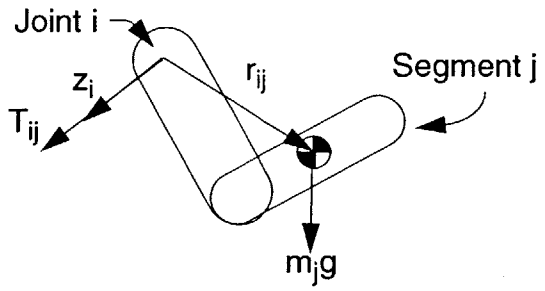


Figure 3-12. Gravity-induced torque T_{ij} of segment j on joint i.

Using the Denavit-Hartenburg notation for robot kinematics, the position, $\underline{r}_i$, and orientation, $\underline{z}_i$, of each of the robot's joints and the position of the center of mass of each space suit segment, $\underline{r}_j$, were calculated, giving the $\underline{z}_i$ and $\underline{r}_{ij}$ vectors for a given set of joint angles. Coordinate axes used on each of the robot's arm and leg segments are shown in Figure 3-13. The Denavit-Hartenburg notation prescribes the locations of the x and z axes, based on

the joint axes of the current segment and the previous segment.[5, 14] The *y* axes are chosen to form a right-handed set of coordinates.

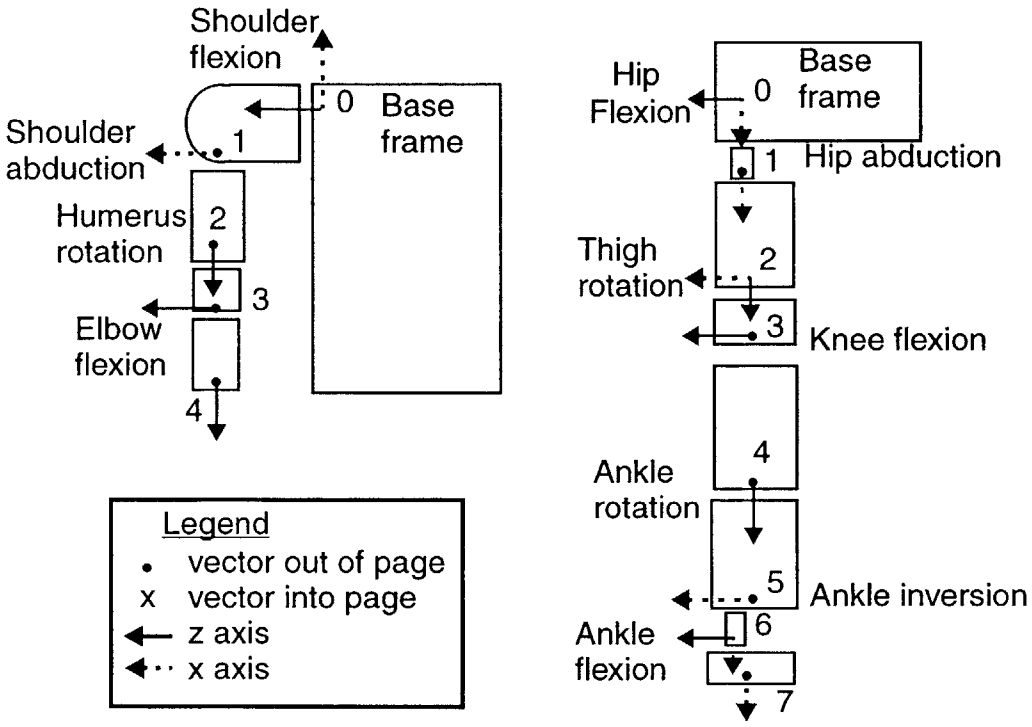


Figure 3-13. Coordinate axes for robot's arm and leg.

Based on the coordinate axes shown in Figure 3-13 and the joint angle definitions from Figure 3-3, the four Denavit-Hartenberg link parameters, a, d, α, and θ, for each robot segment are given in Table 3-5 for the leg and Table 3-6 for the arm.

Table 3-5: Leg link parameters

Link number	a (m)	d (m)	α (deg)	θ(deg)
1	0.0699	0	-90	hf
2	0	0	-90	-ha-90
3	0	0.40	-90	tr-90
4	0	0	90	kf
5	0	0.43	90	ar+90
6	0.0338	0.0157	90	ai+90
7	0.0845	0	-90	af

Table 3-5: Leg link parameters

Link number	a (m)	d (m)	α (deg)	θ(deg)
8	0	0.220	0	0

Table 3-6: Arm link parameters

Link number	a (m)	d (m)	α (deg)	θ (deg)
1	0	0.127	90	sf
2	0	0	-90	-sa+90
3	0	0.318	-90	-hr-90
4	0.140	0	90	ef-90

The position and orientation of each segment are represented by 4x4 homogeneous matrices. The homogeneous matrix that transforms coordinate frame *i-1* to coordinate frame *i* with rotation *R* and translation by vector $\underline{x}$ is given by:

$$A_{(i-1)i} = \begin{bmatrix} R & x \\ 0\ 0\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\alpha_i\cos\theta_i & -\sin\alpha_i\cos\theta_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq 3-6}$$

Positions and orientations are propagated down the kinematic chain by multiplying the $A_{i,\,i+1}$ matrices. The homogeneous matrix representing the transformation from the base coordinate frame to the frame of segment *n* is given by:

$$T_{0n} = \prod_{1}^{n} A_{(i-1)i} \quad \text{Eq 3-7}$$

By multiplying the $A_{i,\,i+1}$ matrices corresponding to each segment, the position and orientation of each of the robot's joints can be calculated. Positions of space suit segments are calculated by considering each space suit segment to be a shorter version of one of the robot's segments, with a length equal to the distance from the next robot joint closer to the torso to the space suit segment center of mass. To calculate a space suit center of mass

position, the space suit segment is used as the terminal segment of the chain. Link parameters and masses of the space suit segments are given in Table 3-7.

Table 3-7: Space suit link parameters

Space suit link	Corresponding robot link	a(m)	d(m)	a(deg)	q(deg)	mass (kg)
upper arm	3	0	0.225	-90	-hr-90	0.677
lower arm and glove	4	0.196	0	90	ef-90	2.25
upper leg	3	0	0.239	-90	tr-90	0.741
lower leg	5	0	0.252	90	ar+90	1.18
boot	8	0	0.0668	0	0	2.24

For each time step in the robot suited data file, the robot joint angle data is used to calculate the $A_{i,\,i+1}$ matrix for each robot and space suit segment. The $A_{i,\,i+1}$ matrices are multiplied to obtain the $\underline{r}_i$ vectors that go from the origin of the base coordinate frame to the origin of the segment i coordinate frame, the $\underline{z}_i$ vectors that indicate the axis orientation of joint i, and the $\underline{r}_j$ vectors that go from the origin of the base coordinate frame to the center of mass of space suit segment j. The robot and space suit link position and orientation vectors are substituted into Equation 3-5 to obtain the torque caused by the weight of space suit segment j on joint i. The space suit weight-induced torques are then summed for each joint to obtain the torque caused by the space suit's weight. Torques due to the space suit's weight are then subtracted from the robot's torque data.

3.6 Results: Experiment

3.6.1 Example data

After both the robot's weight and the space suit's weight are eliminated from the recorded torque data, the remaining torques indicate the torque necessary to bend the joints of the space suit. Examples of the space suit-induced torque plotted vs. joint angles are shown in Figure 3-14- Figure 3-18. Each of the example plots represents one test subject and one experimental trial, which includes several repetitions of the motion. Joint angles are measured according to the definitions provided in Figure 3-3.

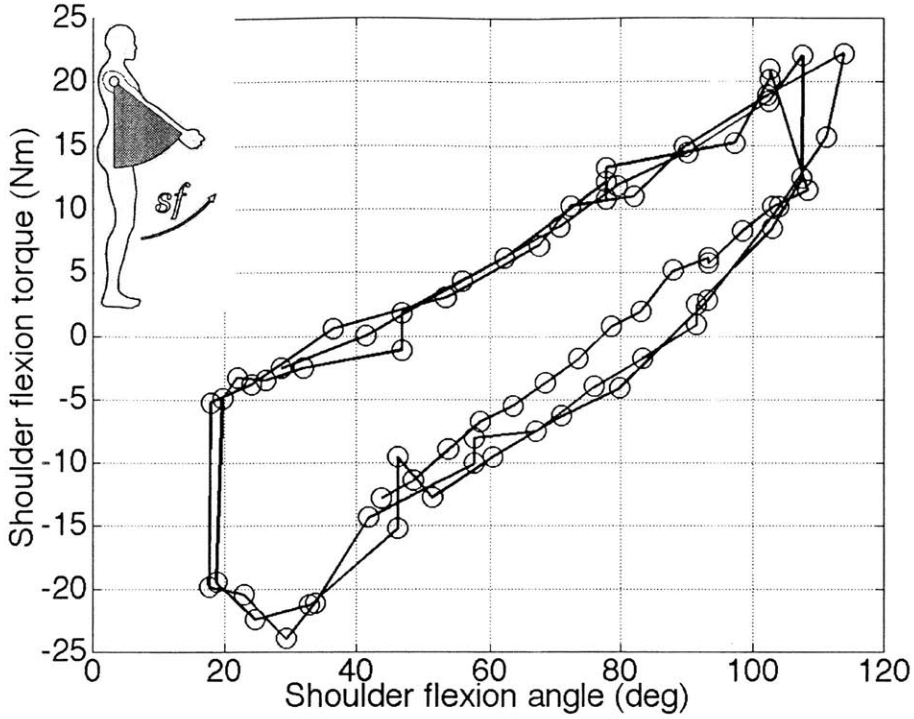


Figure 3-14. Shoulder flexion torque vs. angle, 3 repetitions.

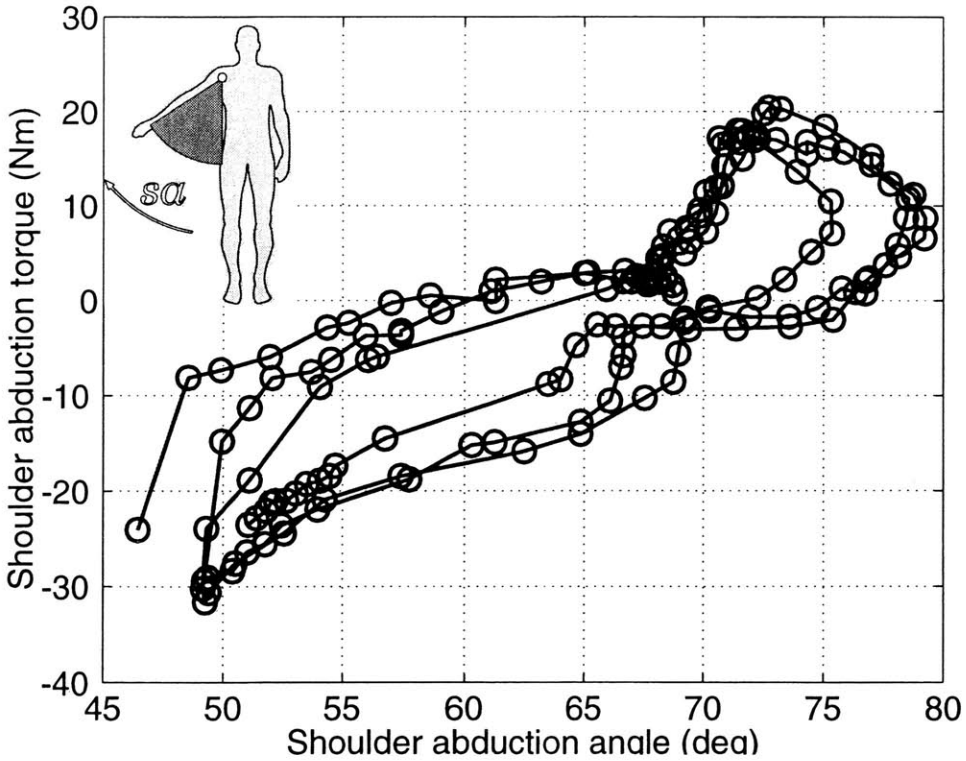


Figure 3-15. Shoulder abduction torque vs. angle, 3 repetitions.

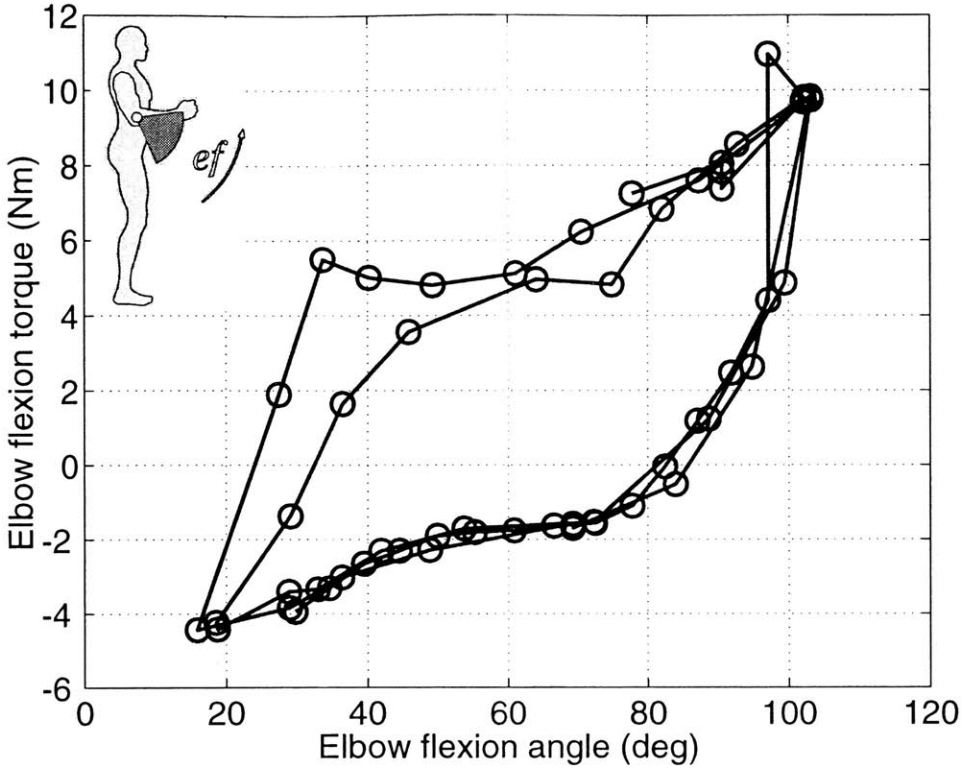


Figure 3-16. Elbow flexion torque vs. angle, 2.5 repetitions.

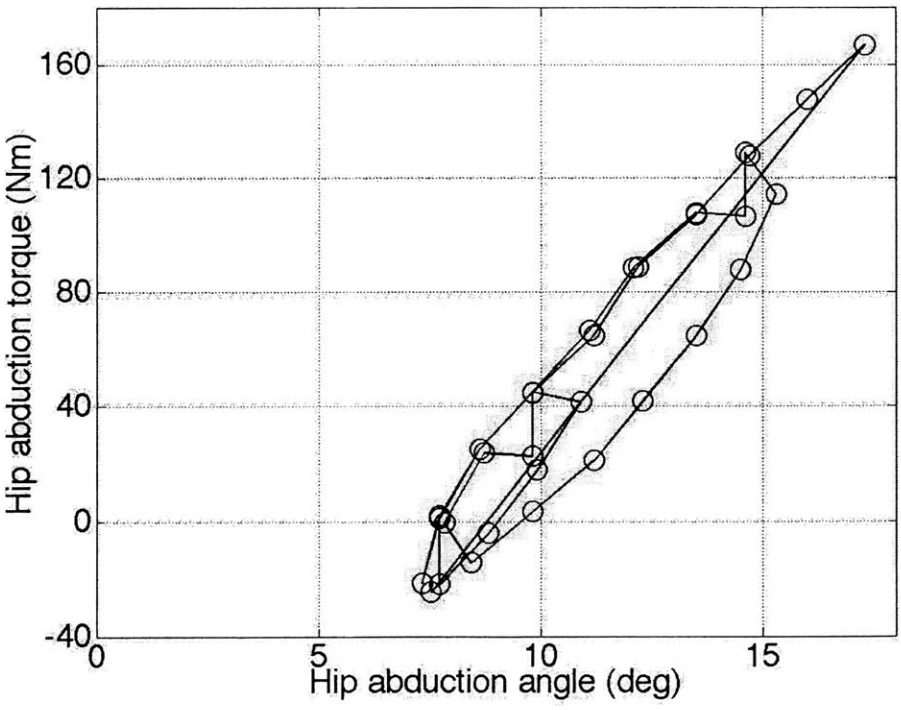


Figure 3-17. Hip abduction torque vs. angle, 3 repetitions.

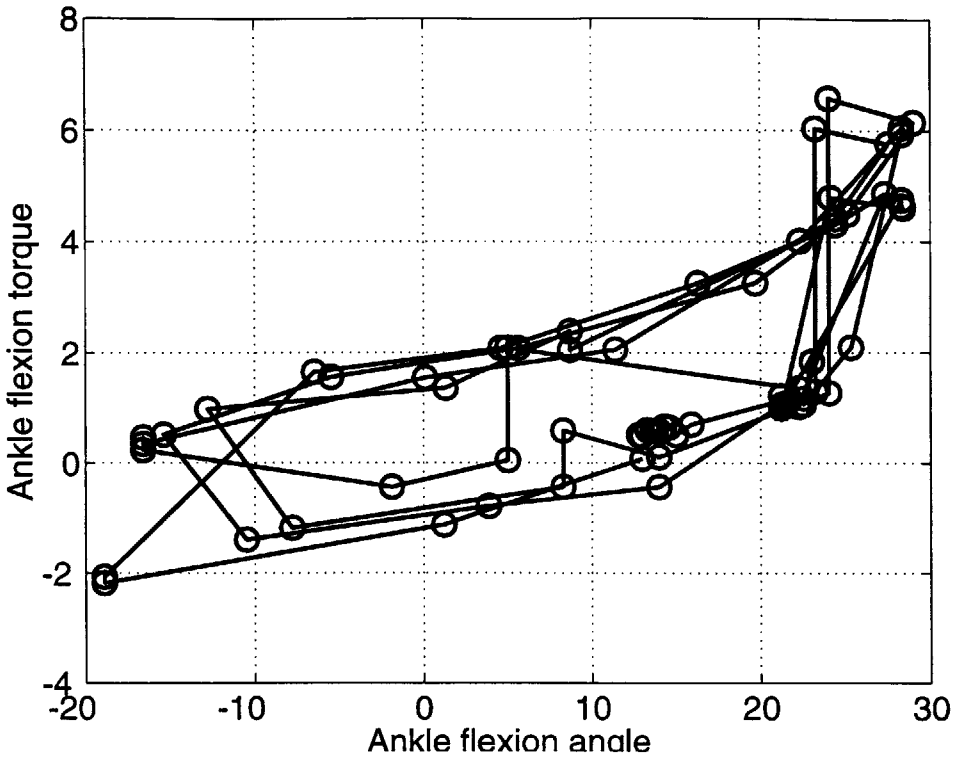


Figure 3-18. Ankle flexion torque vs. angle, 4 repetitions.

3.6.2 Database coverage

The space suit database compiled in this work covers a larger range of joint angles than any other previous study on space suit joint torques. Angle ranges for which angle and torque data was obtained are shown in Figure 3-19. The gray rectangles in Figure 3-19 indicate the range of motion that was specified for the design of the EMU, or the robot's range of motion for the humerus rotation, ankle rotation and ankle inversion joints, where the EMU range of motion was unspecified. The robot's range of motion coincides with the EMU design specification in most joints, except the shoulder abduction joint. The robot's shoulder abduction joint has a range of 0 deg-90 deg, which is less than the EMU design specification. The robot can, however, position its arm above its shoulder level by using the shoulder flexion joint, but it cannot measure true shoulder abduction joint torques in this configuration. Another instance in which the space suit design range of motion is much greater that the database range is the humerus rotation joint. The human test subjects did not exceed 50 deg of positive humerus rotation angle, even though the robot can accomplish a positive humerus rotation of 90 deg. Published data on the range of motion of unsuited individuals for humerus rotation motions indicates that the 50th percentile humerus rotation limit in the positive direction is approximately 60 deg, and the 95th percentile positive limit is 97 deg.[2] It is likely, then, that a 90 degree humerus rotation would not have been achievable by the test subjects, even if they had not been wearing the space suit.

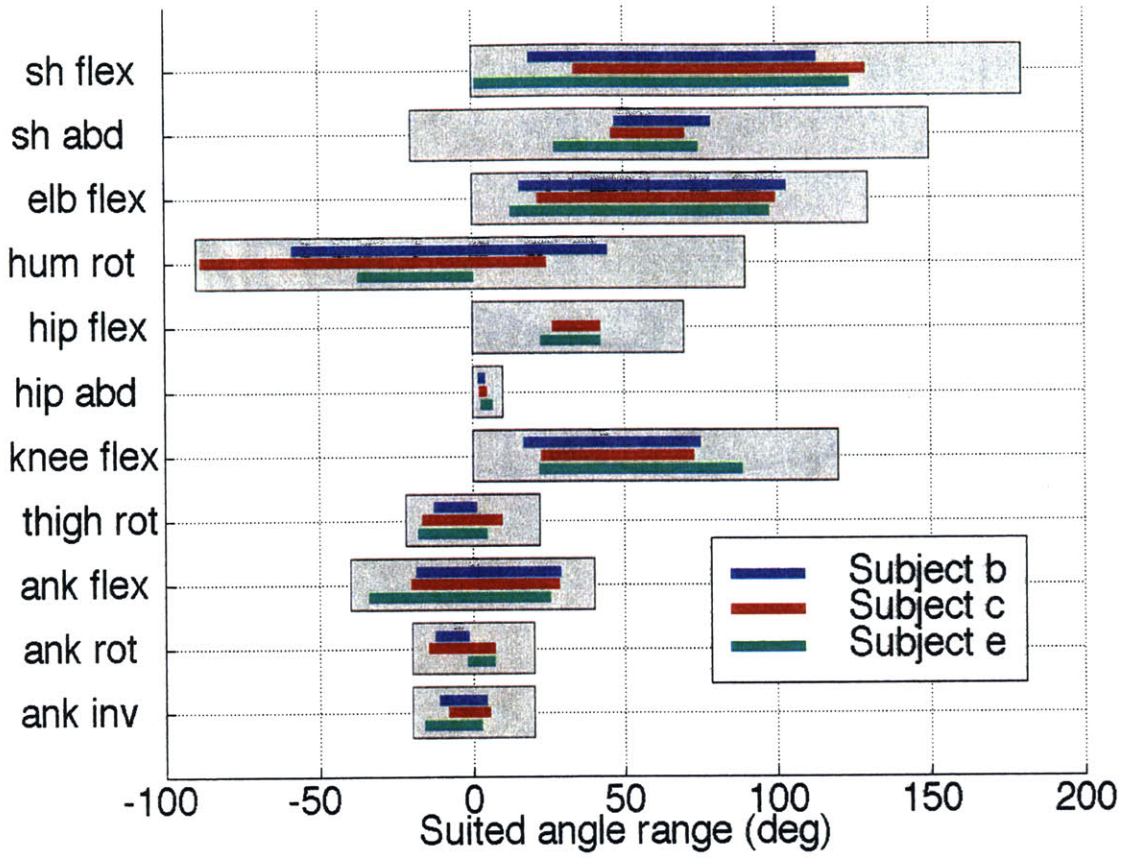


Figure 3-19. Angle range of space suit database compared to space suit design specification for joint range of motion.

A total of 80 motion capture data sets were obtained from the space-suited test subjects, 247 torque data sets were obtained from the suited robot and 337 data sets obtained from the unsuited robot, resulting in over 580 robot torque data sets.

3.6.3 Error analysis

Errors in the torque and angle values come from three sources:

- Motion capture system errors in estimating the human subjects's joint angles
- Robot joint position errors due to imperfect tracking of the trajectory
- Robot torque measurement errors.

3.6.3.1 Motion capture errors

Estimating the accuracy of off-the-shelf motion capture systems, such as the one used in this study, is difficult. Manufacturers make optimistic accuracy claims and sufficient computational detail is usually not available to the end user to make a detailed error analysis.

putational detail is usually not available to the end user to make a detailed error analysis. This analysis is an effort to make a reasonable, but somewhat conservative, estimate of the motion capture system's accuracy in the configuration used in this study.

The motion capture system measures the positions of reflective markers attached to the human test subjects' arm, leg, and torso. Using the Cartesian positions of the markers, the angles between arm and leg segments are calculated. Random errors in measuring marker positions can be propagated through the angle calculations to produce estimates of the angle errors that result from position measurement errors.

The angle calculation is shown in Figure 3-20. The motion capture system outputs the *x*, *y*, *z* position of four points, A, B, C, and D, on two successive arm or leg segments. Using the positions of the four points, vectors r_1 and r_2, along the two body segments, are calculated. The angle between these two body segment vectors, θ, is the joint angle.

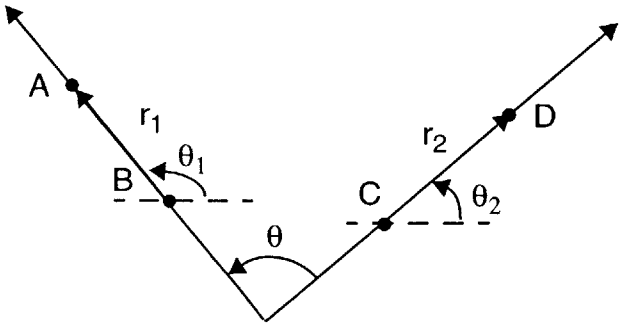


Figure 3-20. Joint angle calculation used for motion capture data.

As shown in Figure 3-20, the vectors r_1 and r_2 are at angles θ_1 and θ_2 with respect to an arbitrary coordinate reference. The joint angle θ is the difference between θ_1 and θ_2. Focusing on a single body segment vector illustrates how errors in measuring the positions of points A-D relate to angle errors. Figure 3-21 shows body segment vector r_1, with point A' displaced from point A by an error *d*, perpendicular to r_1.

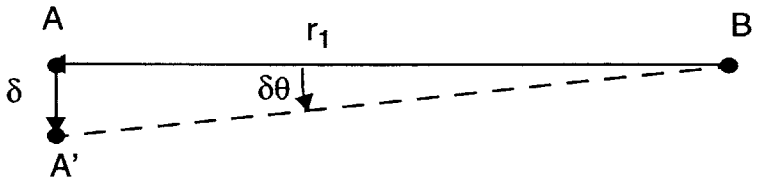


Figure 3-21. Joint segment vector with measurement error δ.

The difference in angle between segment BA and segment BA' is given by

$$\delta\theta \cong \sin(\delta\theta) = \frac{|\underline{\delta}|}{|\underline{r}_1 + \underline{\delta}|} = \sqrt{\frac{\delta^2}{r_1^2 + \delta^2}} \qquad \text{Eq 3-8}$$

If d_2 is the variance of the error in measuring r_1, then

$$var(\delta\theta_1) = \frac{\delta^2}{r_1^2 + \delta^2}. \qquad \text{Eq 3-9}$$

By the same reasoning, the variance of the error in θ_2 is

$$var(\delta\theta_2) = \frac{\delta^2}{r_2^2 + \delta^2}. \qquad \text{Eq 3-10}$$

Therefore, the variance of the error in measuring θ is given by

$$var(\delta\theta) = var(\delta\theta_1) + var(\delta\theta_2) = \frac{\delta^2}{r_1^2 + \delta^2} + \frac{\delta^2}{r_2^2 + \delta^2}. \qquad \text{Eq 3-11}$$

Equation 3-11 shows that errors in joint angle estimation increase when marker position estimation errors increase and decrease when the markers on a body segment are separated by a greater distance.

The manufacturer of the motion capture system claims that marker positions are determined with errors on the order of millimeters. The high accuracy claim may apply to the most favorable configurations, but the motion capture data acquired in this study is not likely to be that accurate. The markers used in this study were hemispheres of diameter 1 cm. It is reasonable to assume that, under good visibility conditions, the markers are accurately located to within one diameter, or 1 cm.

Because the markers were blocked from the camera's view at times, redundant markers were used. Each of the points A, B, C, and D in Figure 3-20 is a position calculated by taking the means of the positions of two markers that were placed on a band around a test subject's arm or leg. Averaging marker positions improves accuracy, but, since markers were sometimes obscured, the full eight-marker accuracy was not obtained at all times. When two markers are available to be averaged, the variance of the error in calculating the position of point A, B, C or D is 0.5 cm^2, while when only one marker is available, the position error variance is 1 cm^2. Table 3-8 shows the angle error standard deviation for r_1 and r_2 dimensions typical of several joints, for different numbers of visible markers. It is

assumed that r_1 and r_2 span 0.75 of the length of the arm or leg segment. At least four markers must be visible in order to calculate an angle between the two segments.

Table 3-8: Angle error estimates for different numbers of visible markers

	Joint angle error standard deviation (deg)				
Joint	8 markers	7 markers	6 markers	5 markers	4 markers
Elbow	3.7	4.2	4.6	5.0	5.3
Knee	2.8	3.3	3.5	3.8	4.0

3.6.3.2 Robot trajectory tracking errors

The motion capture data from the human subjects was used as trajectories for the robot to follow in both suited and unsuited torque data collection. Errors in trajectory following were caused by loads imposed by the space suit and timing issues with the RSSTA software application.

The space suit imposed significant loads on the robot as it attempted to follow the trajectories. As a result, the amplitudes of some of the motions were reduced on the robot, particularly in the shoulder abduction and hip flexion joints, where the loads due to the robot and space suit's weight and the space suit's stiffness were largest. The lack of stiffness of the robot's joints effectively limited the range over which the joint could be positioned when the space suit was on the robot. Because the RSSTA application does not allow the user to command joint deflections that are outside the robot's limits, the actual joint angle that was obtained when the maximum joint angle was commanded was the effective limit. This problem can be remedied in the future by tuning the servo gains on the robot's shoulder and hip joints to have higher position error gains. Effects of tracking errors when the robot was wearing the space suit are shown in Figure 3-22, where the percentage of unsuited joint angle range that was achieved in suited data collection is plotted for each of the 11 simple motions.

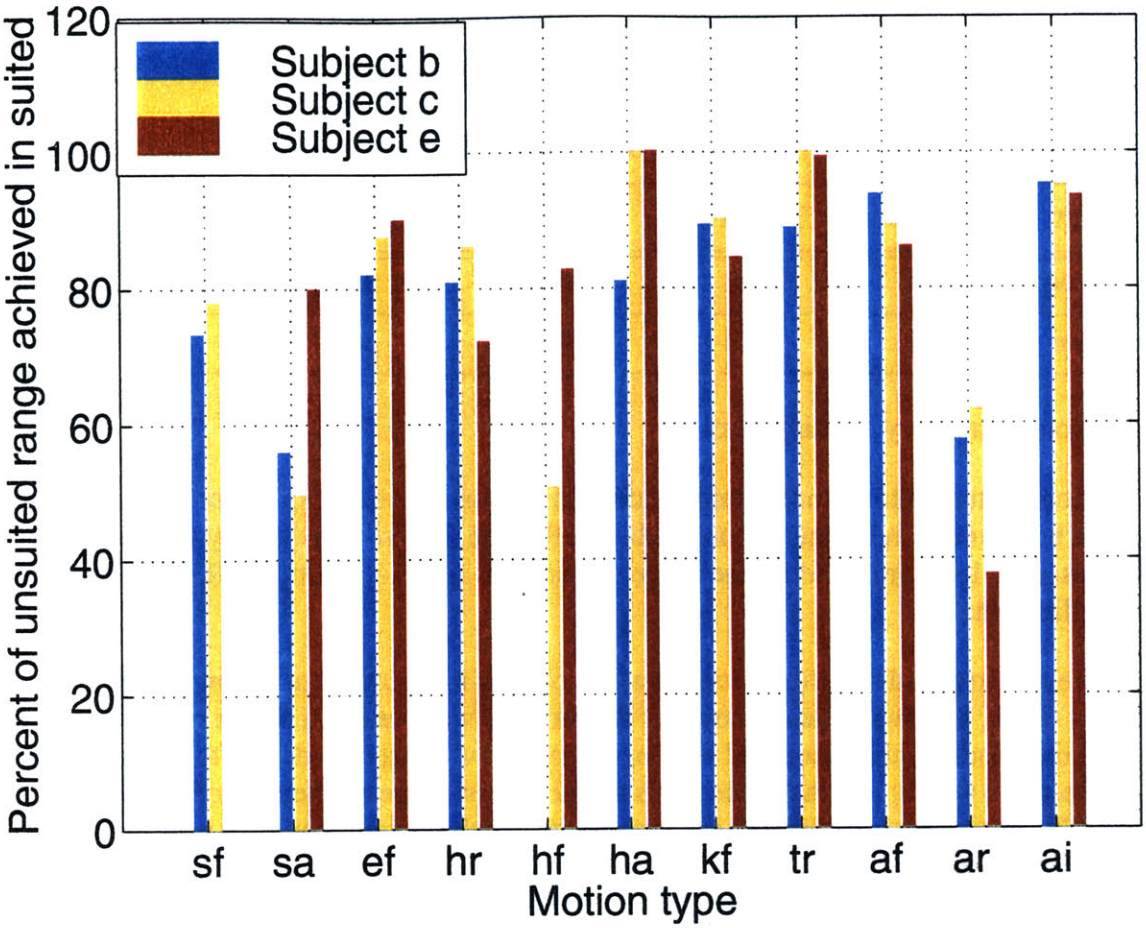


Figure 3-22. Percentage of unsuited range achieved in suited robot trials.

An additional source of error in robot trajectory following arose from timing issues. When the robot was programmed to follow motion capture data trajectories at full speed, the joint angle data recorded by the robot indicated in many cases that the robot actually moved slower than the command trajectory. The duration of the data file exceeded the duration of the trajectory file and it appeared that samples were being dropped while the robot was performing motions. The robot would hold all joints in the same positions for 2-3 samples at a time, then proceed with the motion. The number of "held" samples corresponded to the excess time taken in executing the motions. Less sample dropping occurred when trajectories were run at lower speeds. To produce better matching of the robot's motions to the command trajectories, suited and unsuited data was collected for all trajectories at both full speed and half speed.

The angle data in the space suit torque-angle database is the robot's actual position, not the commanded position, so tracking errors have the effect of slightly distorting the motions that the robot performed; tracking errors do not cause errors in the measured angles in the space suit database. Figure 3-23 shows root mean square (RMS) differences between suited and unsuited robot joint angles, plotted by trial type. The RMS errors, which range

from 0.5 deg to 3 deg, are of the same order of magnitude as the motion capture data errors listed in Table 3-8.

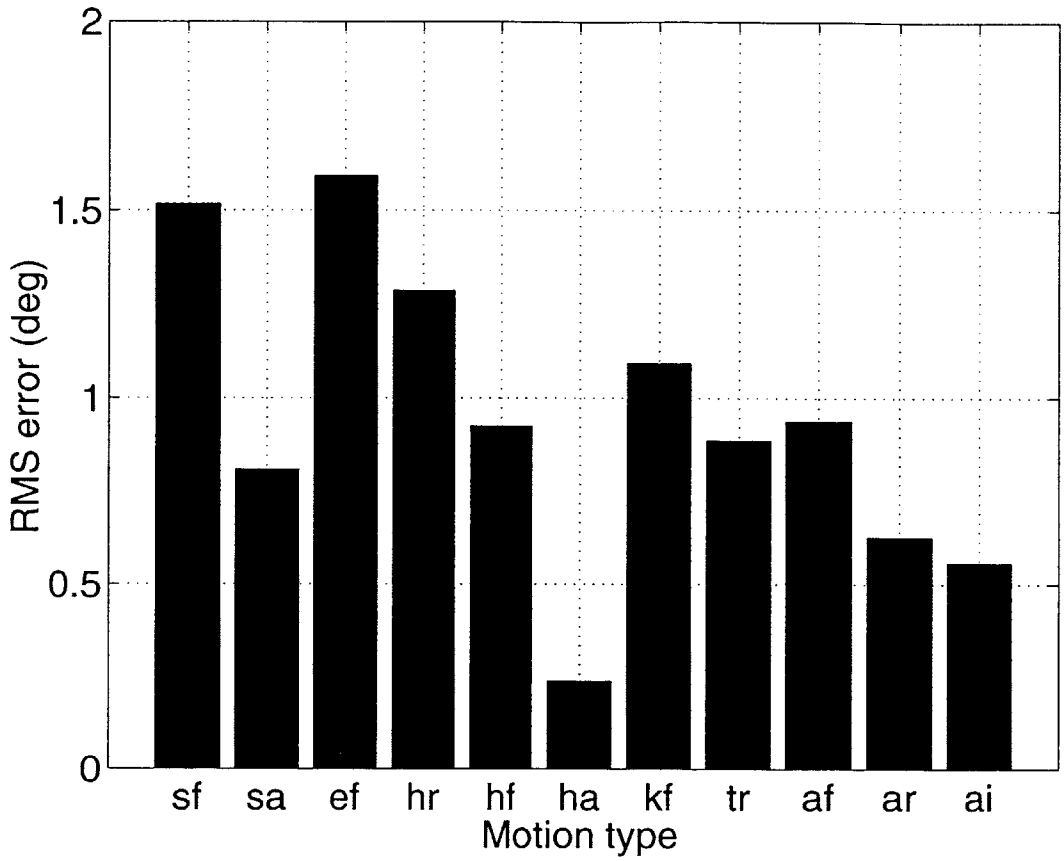


Figure 3-23. Mean RMS angle differences between suited and unsuited robot motions.

3.6.3.3 Torque measurement errors

Errors in torque measurement arise from bias, noise, and quantization. The robot's load cells were calibrated when it was originally set up at Johnson Space Center. These calibration factors were in use when suited and unsuited data was collected with the robot. The load cell calibrations were again evaluated after the data collection by applying known torques to each load cell and comparing the RSSTA torque output to the known applied torque. The later assessment indicated that calibration factor errors were all less than 4%.[13]

Noise and quantization effects interact to result in a random error of approximately 0.113 Nm (1 inch pound) in all joints. Torque values are represented in the RSSTA software as integers in the units of inch pounds, reducing the resolution by as much as a factor of 10, compared to the hardware-imposed resolution limits, in some cases. As a result, the resolution of the torque data depends on the calibration factor for each joint. The resolution is the larger of the torque corresponding to 1 A/D count or 1 inch pound (0.113 Nm). For all

joints except the hip flexion joint, the torque resolution is 0.113 Nm; the torque resolution for the hip flexion joint is 0.226 Nm.

3.7 Database contributions

The objective of the experimental portion of this thesis was to compile an extensive database of the torques required to bend the joints of a space suit, in realistic, human-generated motions. The experiments were carried out in two phases, with both human test subjects and an instrumented robot. Human test subjects wearing the space suit performed 20 simple and complex motions, while joint angles of their arm and leg were recorded using video motion capture. The space suit was then installed on an instrumented robot, which was driven to reproduce the human subjects motions while joint torques were recorded. Because the robot had never been used with a space suit, several modifications were made to the robot to allow the space suit to fit on the robot and protect both from damage. Contributions due to the weight of the space suit and robot were subtracted from the torque data, resulting in a consistent set of space suit joint torques and joint angles in realistic motions, with angles accurate to approximately 2 deg-5 deg and torques accurate to approximately 0.1 Nm.

The database compiled in this work is more extensive than any other published space suit torque-angle database, covering 11 joints over a large range of angles. Realistic, three-dimensional human-generated motions were used for data collection. Because torque data was collected by the robot as a surrogate for a human occupant of the space suit, the torques measured in this study are more representative of realistic conditions than data from previous studies that measured joint stiffnesses for empty, pressurized space suits. The space suit torque-angle database serves as a basis for developing and validating both mathematical and physical models of space suit joint mobility characteristics.

Chapter 4 *Modeling*

4.1 Introduction

The space suit torque-angle database described in Chapter 3 presents a unique opportunity to develop models of space suit mobility and verify them against experimental data. Models of space suit mobility are useful in two applications: numerically predicting the torque required to bend a space suit's joints and understanding the physical processes that determine how mobile a space suit's joints are. These two applications clearly require two different modeling approaches: first, a descriptive mathematical modeling technique based on experimental data and second, a theoretical model based on physical principles. Chapter 4 describes two modeling efforts: a mathematical model based on empirical data that predicts the torque required to bend the space suit's joints and a comparison between two physics-based models of bending pressurized cylinders and the space suit torque angle database compiled in Chapter 3. The opportunity to validate the mathematical and physical models against experimental data is a unique aspect of this thesis.

4.2 Mathematical model

The mathematical modeling section first describes the theoretical conception of the Preisach model, including its mechanism for storing input minima and maxima and a graphical representation of the model implementation. Numerical methods for identifying the Preisach model coefficients and implementing the identified model are then discussed. The Preisach model coefficients were identified for the torque-angle characteristic of six of the space suit's joints. The model was then used to predict the torque necessary to bend the space suit joints for motions that were generated by space-suited human test subjects. Finally, the predicted torques are compared to the torques measured experimentally in Chapter 3.

4.2.1 The Preisach hysteresis model

The purpose of the mathematical model is to make dynamic numerical predictions of the torques required to bend the space suit joints. The space suit joint torque-angle data collected in the experimental portion of this thesis, shown in Figure 3-14-Figure 3-18, along with previous investigators' space suit joint torque-angle data, depicted in Figure 2-4 and

Figure 2-5, illustrate that the torque-angle relationship for space suit joints is hysteretic. A hysteretic system has an output that is determined by both the input and the input history. As the data in Chapter 2 and Chapter 3 show, the space suit joint torque-angle curves depend on whether the torque is increasing or decreasing and the history of torque minima and maxima.

Hysteresis is a common characteristic of fabrics. When fabrics are loaded, the individual yarns not only stretch, but also slide over each other. The combination of elastic and viscous forces within the fabric results in the large-scale hysteretic behavior, as described in detail in Section 2.4.1. Numerous other physical systems exhibit hysteretic behavior, including shape memory alloys, piezoceramics, and magnetic materials. Several techniques, described in Section 2.4.1, that have been developed to model other hysteretic systems can be used to model space suit joints. Of these, the Preisach hysteresis model was chosen to model space suit joints because it can reproduce the shape of the space suit torque-angle curves and a straightforward method exists for identifying the model coefficients.

4.2.1.1 Preisach model overview

The Preisach hysteresis model reproduces a hysteresis curve by summing contributions from the simplest possible hysteresis transducers. The primitive hysteresis transducer, $\hat{\gamma}(\alpha, \beta)$, is shown in Figure 4-1. For ascending inputs, the output follows the path *abcde*; for descending inputs, the output follows *edfba*. The two parameters that may be set are the ascending and descending switching values, α and β. The output of the transducer is always either -1 or +1.

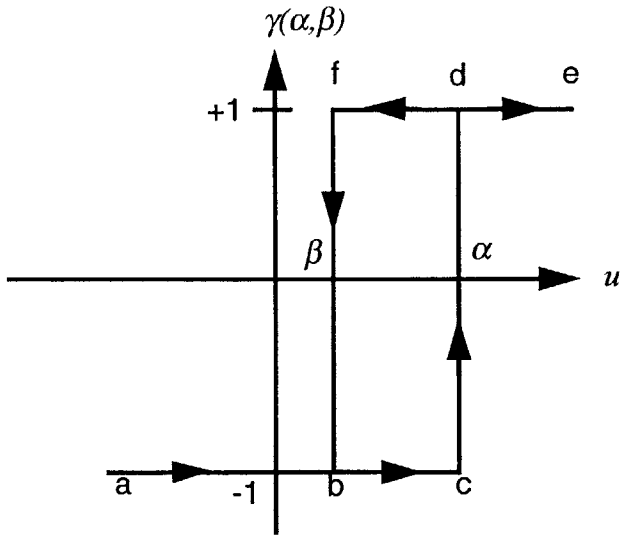


Figure 4-1. Simplest hysteresis transducer

To construct more complicated hysteresis transducers with continuous, non-unity outputs, the Preisach model uses a weighted sum of simple hysteresis operators. The weighting

function $\mu(\alpha,\beta)$ is defined as a function of the combination of upward and downward switching values, α, β, of the hysteresis transducer. The Preisach function, $\mu(\alpha,\beta)$, is defined for all $\alpha > \beta$, $-\alpha_0 < \alpha < \alpha_0$, and $-\alpha_0 < \beta < \alpha_0$, forming the triangle in α–β space shown in Figure 4-2.

Construction of the output of the composite hysteresis transducer is done by integrating the individual $\hat{\gamma}_{\alpha,\beta}$ values, as shown in Equation 4-1.

$$f(t) = \iint_{(\alpha > \beta)} \mu(\alpha, \beta)\hat{\gamma}_{\alpha,\beta}u(t)d\alpha d\beta \qquad \text{Eq 4-1}$$

Calculation of $f(t)$ is aided by the graphical representation of the α-β space shown in Figure 4-2. The weighting function $\mu(\alpha,\beta)$ is defined over the triangle that is bounded by the $\alpha=\beta$ line and the maximum and minimum values of α and β, $\pm\alpha_0$, which is defined by the saturation limits of the output. To obtain $f(t)$, Equation 4-1 is integrated over the triangle shown in Figure 4-2.

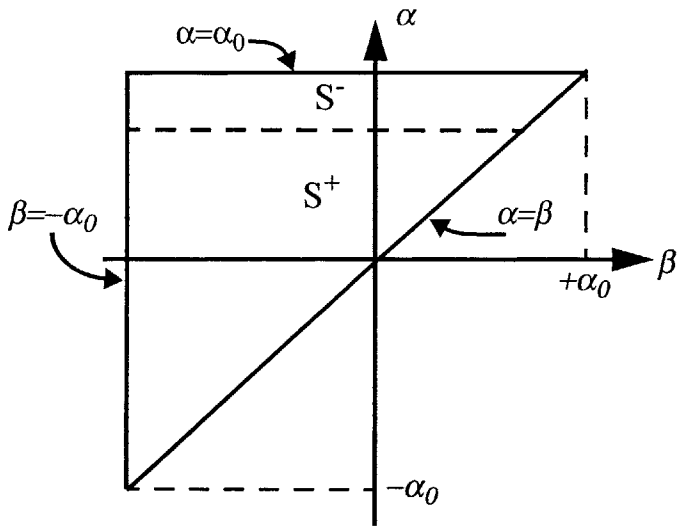


Figure 4-2. Triangle in the α–β space over which $\mu(\alpha,\beta)$ is defined.

To perform the integration in Equation 4-1,the triangle may be subdivided into two sets:

- Region S+, where corresponding $\hat{\gamma}_{\alpha,\beta}$ operators are "up", or equal to +1
- Region S-, where corresponding $\hat{\gamma}_{\alpha,\beta}$ operators are "down", or equal to -1

Substituting +1 for $\hat{\gamma}_{\alpha,\beta}$ in the S^+ region and -1 for $\hat{\gamma}_{\alpha,\beta}$ in the S^- region, Equation 4-1 simplifies to

$$f(t) = \iint_{S^+} \mu(\alpha, \beta) d\alpha d\beta - \iint_{S^-} \mu(\alpha, \beta) d\alpha d\beta . \qquad \text{Eq 4-2}$$

Equation 4-2 can be integrated if the boundary between the S^+ and S^- regions is known.

The input history is stored by drawing the boundary between the S^+ and S^- regions. The boundary is drawn by constructing line segments in a staircase pattern within the triangle based on the value of the input *u(t)* and whether the input is increasing or decreasing. The boundary is drawn according to the following rules:

1. The boundary starts on the $\alpha = \alpha_0$ segment if the initial input is descending and the $\beta = -\alpha_0$ segment if the initial input is ascending.
2. Subsequent boundary segments are drawn horizontally or vertically depending on whether *u(t)* is increasing or decreasing:

- Increasing *u(t)*: horizontal line segment at $\alpha = u$
- Decreasing *u(t)*: vertical line segment at $\beta = u$

3. A boundary segment is obsolete if its α value is less than the α value of a later segment having the same β value or if its β value is less than the β value of a later segment that has the same α value.
4. The last line segment ends on the $\alpha = \beta$ line.

An example of boundary drawing is shown in Figure 4-3. The input *u* initially increases, resulting in a horizontal segment at $\alpha = u$, moving upwards at point A. After *u* reaches its maximum and begins to decrease, the boundary link is a vertical line at $\beta = u$, at point B. Finally, when *u* increases again, the latest boundary segment is a horizontal line at $\alpha = u$. Previous maxima and minima are stored as the vertices between previous segments of the boundary.

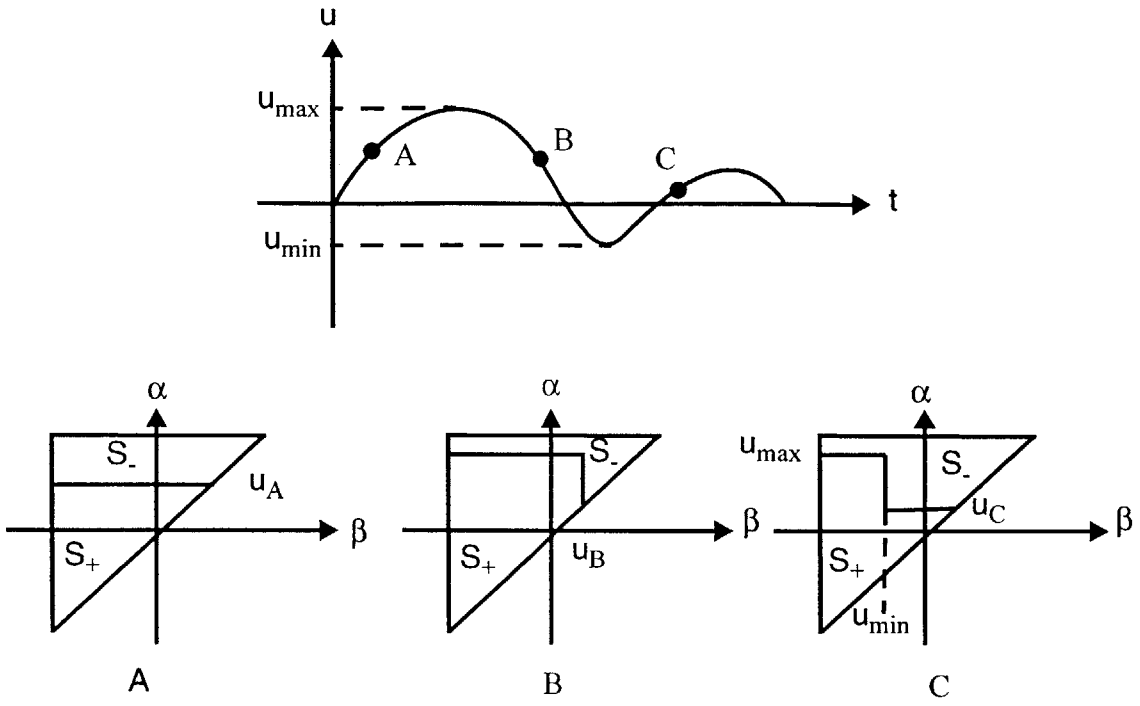


Figure 4-3. Boundary drawing for Preisach model

If the boundary between the S^+ and S^- regions is known, the output value T can be calculated by integrating μ over S^+ and S^-, as shown in Equation 4-2, or an equivalent expression:

$$T = \iint_{T} \mu(\alpha, \beta) d\alpha d\beta - 2 \iint_{S-} \mu(\alpha, \beta) d\alpha d\beta = T_{max} - 2 \iint_{S-} \mu(\alpha, \beta) d\alpha d\beta . \qquad \text{Eq 4-3}$$

Using Equation 4-3, the output T can be calculated from the maximum possible output T_{max} and the integral of $\mu(\alpha,\beta)$ over the region S-.

4.2.1.2 Exploration of the Preisach hysteresis model

If $\mu(\alpha,\beta)$ is specified in a functional form and the input u is known, the output of the Preisach model can be determined analytically as a function of u. Looking at the input-output curves of some examples of $\mu(\alpha,\beta)$ illustrates properties of the Preisach model. The first two examples demonstrate implementation of the Preisach model to determine the outermost curves of the hysteresis loop. The next three examples illustrate the following properties of the outermost hysteresis curves, relating these characteristics to $\mu(\alpha,\beta)$.

- Hysteresis loop direction: clockwise or counterclockwise
- Number of loops present
- Separation between increasing and decreasing curves
- dT/du of the increasing and decreasing curves

Example 1. Simplest case.

The simplest $\mu(\alpha,\beta)$ is an impulse at $\alpha=\alpha_1$, $\beta=\beta_1$. The input-output relation corresponding to this $\mu(\alpha,\beta)$ is a simple hysteresis loop switching from -1 to 1 at $u=\alpha_1$ for increasing u and switching from 1 to -1 at $u=\beta_1$ for decreasing u, as shown in Figure 4-4.

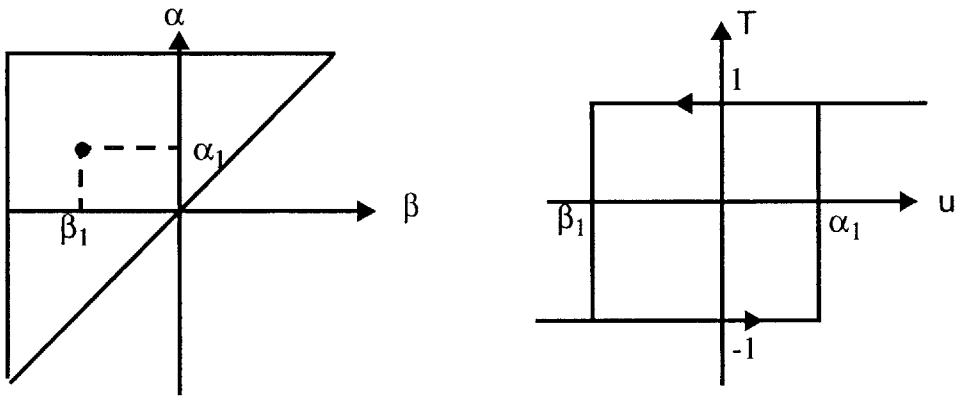


Figure 4-4. μ plotted vs. α and β and T plotted vs. u for $\mu(\alpha,\beta)=\delta(\alpha_1,\beta_1)$.

Example 2. Constant μ

The input-output curve for $\mu(\alpha,\beta)=k$, where k is constant, demonstrates the effect the $\alpha>\beta$ constraint has on the shape of the output curve. As u nears the $+\alpha_0$ and $-\alpha_0$ limits, the area within the triangle being integrated shrinks, and consequently dT/du approaches zero. The constraint also insures that the increasing and decreasing curves always intersect at $u=\alpha_0$ and $u=-\alpha_0$. For u increasing from below $-\alpha_0$, to α_0 then decreasing, the outputs T_{inc} and T_{dec}, are determined by integrating Equation 4-3.

$$T_{\text{inc}} = k(u^2 + 2\alpha_0 u - \alpha_0^2)$$
$$T_{\text{dec}} = k(\alpha_0^2 + 2\alpha_0 u - u^2) \qquad \text{Eq 4-4}$$

Equation 4-4 is plotted in Figure 4-5, for $k=1/2$, $\alpha_0=1$.

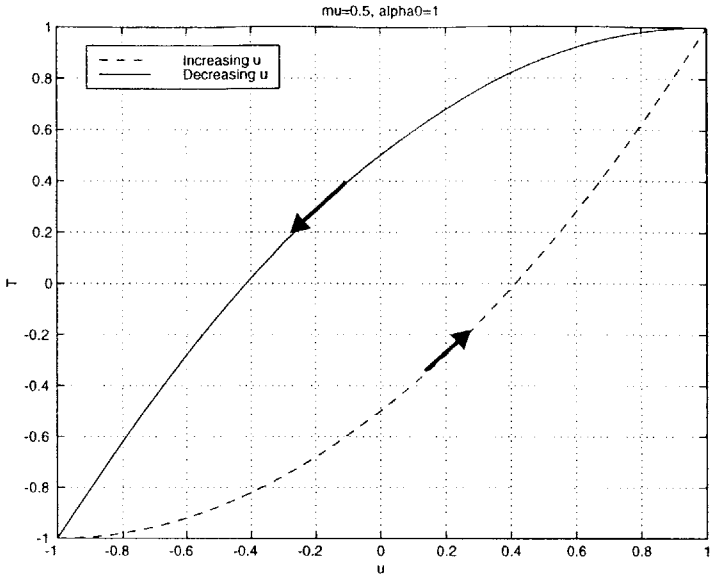


Figure 4-5. Input-output curve for constant μ: $\mu(\alpha,\beta)=k$

The next three examples illustrate relationships between the outermost *T-u* curves and the functional form of $\mu(\alpha,\beta)$. To obtain a $\mu(\alpha,\beta)$ that agrees with experimental data, at least 4 characteristics of the *T-u* curve shape must match:

- Hysteresis loop direction
- Number of loops present
- Separation between increasing and decreasing curves
- *dT/du* of the increasing and decreasing curves.

If the essential features of the modeled hysteresis curve agree with the experimental data, the curve can be transformed by scaling, shifting and rotating it to line up with the data.

The hysteresis loop direction, number of loops present, and separation between increasing and decreasing curves can be determined from the difference between the increasing output curve and the decreasing output curve. If the increasing curve is always greater than the decreasing curve, the loop direction is clockwise; if the increasing curve is always less than the decreasing curve, the loop direction is counterclockwise. The number of loops present is indicated by the number of input values for which the increasing and decreasing curves are equal. The triangular shape of the integration region forces the increasing and decreasing curves to intersect at $u=\alpha_0$ and $u=-\alpha_0$. Any additional intersections of the increasing and decreasing curves indicate the presence of more than one hysteresis loop. Finally, the separation between increasing and decreasing curves can be determined analytically, if $\mu(\alpha,\beta)$ is known.

The difference between the outermost increasing output curve and the outermost decreasing output curve can be obtained using Equation 4-3

$$T_{inc} - T_{dec} = \iint_{T} \mu(\alpha, \beta) d\alpha d\beta - 2 \int_{(S\text{-})\text{inc}} \int \mu(\alpha, \beta) d\alpha d\beta + \\ -\left(\iint_{T} \mu(\alpha, \beta) d\alpha d\beta - 2 \int_{(S\text{-})\text{dec}} \int \mu(\alpha, \beta) d\alpha d\beta \right) \quad \text{Eq 4-5}$$

The result of Equation 4-5 simplifies to

$$T_{inc} - T_{dec} = -2 \int_{(S\text{-})\text{inc}} \int \mu(\alpha, \beta) d\alpha d\beta + 2 \int_{(S\text{-})\text{dec}} \int \mu(\alpha, \beta) d\alpha d\beta \quad \text{Eq 4-6}$$

The difference of the two integral terms is shown graphically in Figure 4-6. The S- region for the increasing curve is the trapezoid bounded by $u<\alpha<\alpha_0$ and $-\alpha_0<\beta<\alpha$. The S- region for the decreasing curve is the triangle bounded by $u<\alpha<\alpha 0$ and $u<\beta<\alpha$, which is included within the S- region for the increasing curve.

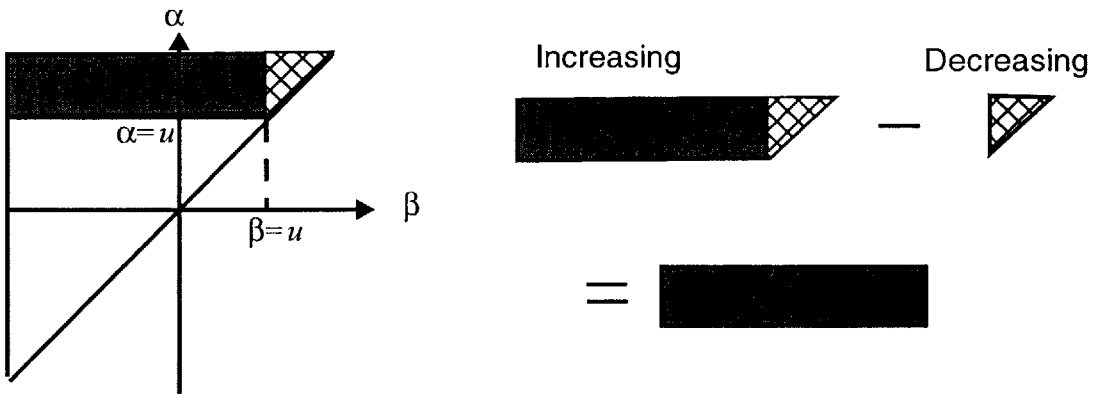


Figure 4-6. Difference between increasing and decreasing outputs.

Since the integrands are equal and only the integration limits are different, Equation 4-6 can be simplified to a single integral over a rectangular region bounded by $u<\alpha<\alpha_0$ and $-\alpha_0<\beta<u$, as Figure 4-6 shows.

$$T_{inc} - T_{dec} = -2 \int_{-\alpha_0}^{u} \int_{u}^{\alpha_0} \mu(\alpha, \beta) d\alpha d\beta \quad \text{Eq 4-7}$$

Intuitively, this means that $\mu(\alpha,\beta)$ functions that have high amplitudes in the top left corner of the α–β space will have large separations between increasing and decreasing curves.

Equation 4-7 can be integrated analytically for the restricted class of $\mu(\alpha,\beta)$ functions where $\mu(\alpha,\beta)=f(\alpha)+g(\beta)$. That is, $\mu(\alpha,\beta)$ is not a function of products of α and β. In that case, the $f(\alpha)$ and $g(\beta)$ functions can be integrated separately to find *Tinc-Tdec*.

$$T_{inc} - T_{dec} = -2 \int_{-\alpha}^{u} \int_{u}^{\alpha_0} (f(\alpha) + g(\beta)) d\alpha d\beta + \\ -2\left[(u + \alpha_0) \int_{u}^{\alpha_0} f(\alpha) d\alpha - (u - \alpha_0) \int_{-\alpha_0}^{u} g(\beta) d\beta \right] \qquad \text{Eq 4-8}$$

To determine the hysteresis loop direction, we first find out how many hysteresis loops are present. If only one loop is present, evaluating T_{inc}-T_{dec} at one u value is enough to determine the direction of the loop. Equation 4-8 can be used to determine how many hysteresis loops are present. Equation 4-8 shows that T_{inc}-T_{dec} is always equal to zero when $u=\alpha_0$ and $u=-\alpha_0$, but if T_{inc}-T_{dec} is equal to zero for any other $-\alpha_0<u<\alpha_0$, then the number of hysteresis loops is greater than 1 and T_{inc}-T_{dec} should be evaluated within each loop to determine the direction of each loop. Equation 4-8 can also be used to find the separation between the increasing and decreasing curves as a function of u.

The shape of the T-u curve is indicated by dT/du. With an expression for dT/du, it is possible to find where the slope of the T-u curve is zero and whether the slope increases or decreases with increasing u. For the outermost increasing curve, where u increases from below α_0, dT/du is given by $dT/d\alpha$, shown in Equation 4-9.

$$\frac{dT}{du} = \frac{dT}{d\alpha} = \frac{d}{d\alpha}\left(\int_{u}^{\alpha_0} \int_{-\alpha_0}^{\alpha} (f(\alpha) + g(\beta)) d\beta d\alpha \right) = (u + \alpha_0) f(u) + \int_{-\alpha}^{\alpha} g(\beta) d\beta \qquad \text{Eq 4-9}$$

dT/du for the outermost decreasing curve is given by $dT/d\beta$, as shown in Equation 4-10.

$$\frac{dT}{du} = \frac{dT}{d\beta} = \frac{d}{d\beta}\left(\int_{u}^{\alpha_0} \int_{\beta}^{\alpha_0} (f(\alpha) + g(\beta)) d\alpha d\beta \right) = \int_{\beta}^{\alpha_0} f(\alpha) d\alpha + (\alpha_0 - u) g(u) \qquad \text{Eq 4-10}$$

Example 3. Linear μ

Example 3 illustrates how the number and direction of hysteresis loops and the slope of the increasing and decreasing input-output curves can be predicted from $\mu(\alpha,\beta)$. In this example, μ varies linearly with α or β. Two cases are considered: $\mu(\alpha,\beta)=k\alpha$, and $\mu(\alpha,\beta)=k\beta$, where k is constant. Integrating to find the input-output curve as in Example 2, we obtain for $\mu(\alpha,\beta)=k\alpha$,

$$T_{inc} = -\frac{2}{3}k\alpha_0^3 - 2k\left(\frac{5}{6}\alpha_0^3 - \frac{u^3}{3} - \frac{u^2\alpha_0}{2}\right)$$

$$T_{dec} = -\frac{2}{3}k\alpha_0^3 - 2k\left(\frac{\alpha_0^3}{3} - \frac{\alpha_0^2 u}{2} + \frac{u^3}{6}\right)$$

Eq 4-11

and for $\mu(\alpha,\beta)=k\beta$, we obtain

$$T_{inc} = \frac{2}{3}k\alpha_0^3 - 2k\left(-\frac{\alpha_0^3}{3} - \frac{u^3}{6} + \frac{\alpha_0^2 u}{2}\right)$$

$$T_{dec} = \frac{2}{3}k\alpha_0^3 - 2k\left(\frac{\alpha_0^3}{6} + \frac{-u^2\alpha_0}{2} + \frac{u^3}{3}\right)$$

Eq 4-12

The results are plotted in Figure 4-7

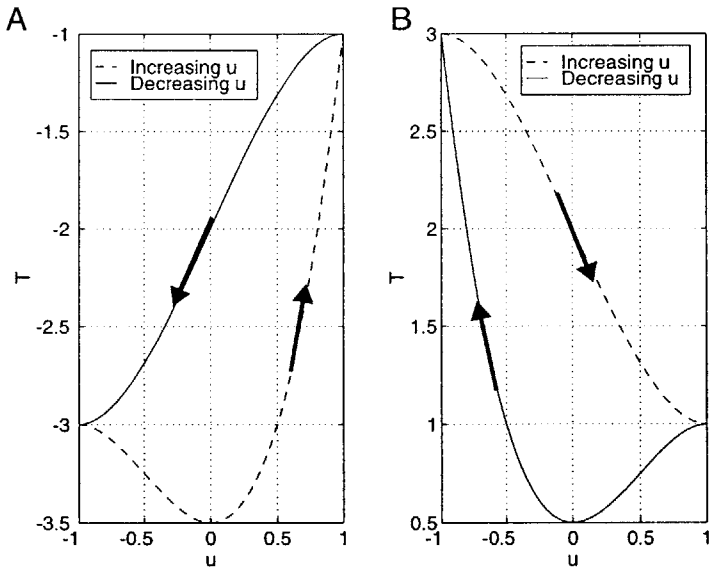


Figure 4-7. Input-output curves for A.)$\mu=k\alpha$, and *B.)*$\mu=k\beta$

Figure 4-7 shows that both μ functions give single hysteresis loops, but the $\mu=k\alpha$ loop is counterclockwise, while the $\mu=k\beta$ loop is clockwise. The $\mu=k\alpha$ curves have a net positive slope, while the net slope of the $\mu=k\beta$ curves is negative. In addition, the shapes of the increasing and decreasing curves are different, with the $\mu=k\alpha$ increasing curve and the $\mu=k\beta$ decreasing curve having zero slope at $u=0$.

Substituting $\mu=k\alpha$ and $\mu=k\beta$ into Equation 4-8 and setting the result equal to zero verifies that, in both cases, the only intersections between the increasing and decreasing curves occur at $u=\alpha_0$ and $u=-\alpha_0$, thus there is only one hysteresis loop in each case.

$$T_{inc} - T_{dec} = -2\left[(u+\alpha_0)\int_{u}^{\alpha_0} f(\alpha)d\alpha - (u-\alpha_0)\int_{-\alpha_0}^{u} g(\beta)d\beta\right]$$

$$T_{inc} - T_{dec} = -2k(u+\alpha_0)(\alpha_0^2 - u^2) \qquad \mu(\alpha,\beta) = k\alpha$$

$$T_{inc} - T_{dec} = 2k(u-\alpha_0)(u^2 - \alpha_0^2) \qquad \mu(\alpha,\beta) = k\beta$$

Eq 4-13

Since there is only one hysteresis loop in both cases, the direction of the loop can be determined by evaluating Equation 4-13 at one point. Evaluating Equation 4-13 at $u=0$ indicates that, for $\mu=k\alpha$, the loop is counterclockwise, while for $\mu=k\beta$, the loop is clockwise.

$$T_{inc} - T_{dec} = -2k\alpha_0^3 \qquad \mu(\alpha,\beta) = k\alpha, u = 0$$

$$T_{inc} - T_{dec} = 2k\alpha_0^3 \qquad \mu(\alpha,\beta) = k\beta, u = 0$$

Eq 4-14

Finally, we can look at the shape of the *T-u* curves for $\mu=k\alpha$. From Equation 4-9, *dT/du* for the increasing curve is given by

$$\frac{dT}{du} = (u+\alpha_0)f(u) + \int_{-\alpha}^{\alpha} g(\beta)d\beta = ku(u+\alpha_0)$$

Eq 4-15

Equation 4-15 indicates that the slope of the increasing curve is zero at $u=0$ and at $u=-\alpha_0$, as shown in Figure 4-7.

dT/du for the decreasing curve is given by

$$\frac{dT}{du} = \int_{\beta}^{\alpha_0} f(\alpha)d\alpha + (\alpha_0 - u)g(u) = \frac{k}{2}(\alpha_0^2 - u^2)$$

Eq 4-16

According to Figure 4-7 and Equation 4-16, for the decreasing curve, $dT/du=0$ only at $u=-\alpha 0$ and $u=\alpha_0$.

Example 4. Multiple hysteresis loops.

Example 4 demonstrates multiple hysteresis loops. In this example, $\mu(\alpha,\beta)=\alpha+\beta$. Integrating as before, to find *T(u)* for the outermost curves, we get:

$$T_{\text{inc}} = \frac{1}{2}(-u^3 - \alpha_0 u^2 + \alpha_0^2 u + \alpha_0^2)$$

$$T_{\text{dec}} = \frac{1}{2}(u^3 - \alpha_0 u^2 + -\alpha_0^2 u + \alpha_0^2)$$

Eq 4-17

T_{inc} and T_{dec} are plotted vs. u in Figure 4-8. The increasing and decreasing curves cross at $u=0$, resulting in two hysteresis loops in opposite directions.

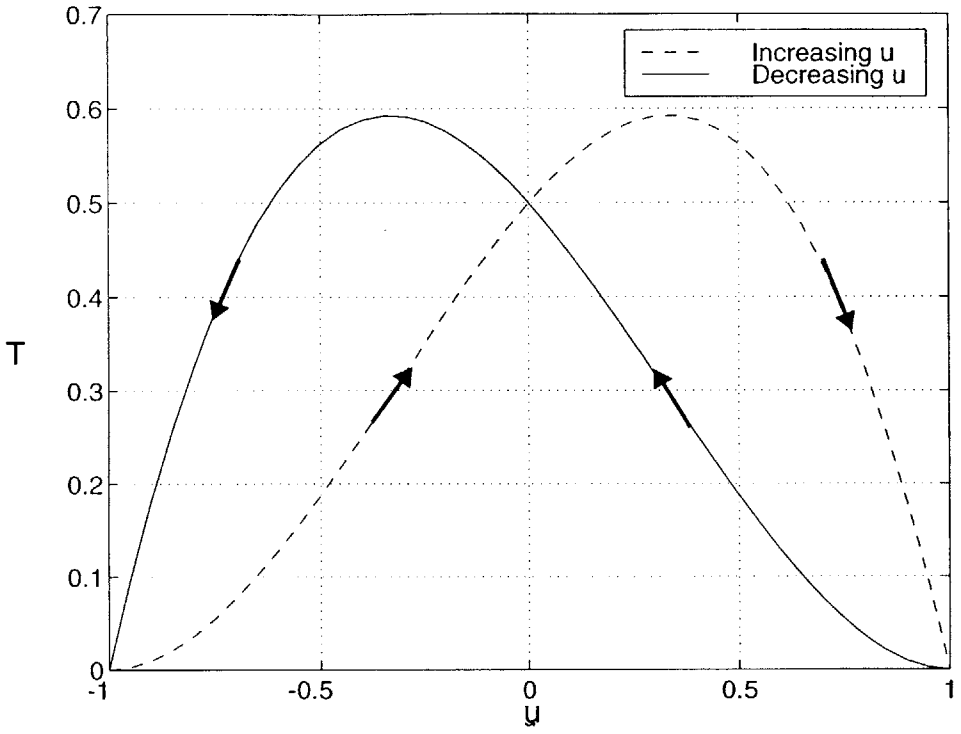


Figure 4-8. *T-u* curves for $\mu=\alpha+\beta$

Equation 4-8 predicts the double hysteresis loop. Substituting for $\mu(\alpha,\beta)$ in Equation 4-8, we get

$$T_{inc} - T_{dec} = -2\left[(u+\alpha_0)\int_{u}^{\alpha_0} f(\alpha)d\alpha - (u-\alpha_0)\int_{-\alpha_0}^{u} g(\beta)d\beta\right] = 2u(\alpha_0^2 - u^2) \qquad \text{Eq 4-18}$$

T_{inc}-T_{dec} is equal to zero at three points: $u=-\alpha_0$, $u=0$, and $u=\alpha_0$, which agrees with Figure 4-8. By substituting u values corresponding to each hysteresis loop in Equation 4-18, it is possible to determine the direction of each loop. If T_{inc}-T_{dec} is positive, the loop is clockwise; if T_{inc}-T_{dec} is negative, the loop is counterclockwise. Substituting $u=-\alpha_0/2$ and $u=\alpha_0/2$ into Equation 4-18 shows that the loop for $-\alpha_0<u<0$ is counterclockwise and the loop for $0<u<\alpha_0$ is clockwise, as shown in Figure 4-8.

$$u = \frac{-\alpha_0}{2} \qquad T_{inc} - T_{dec} = -\frac{3}{4}\alpha_0^3 < 0 \quad \text{counterclockwise}$$

$$u = \frac{\alpha_0}{2} \qquad T_{inc} - T_{dec} = \frac{3}{4}\alpha_0^3 > 0 \quad \text{clockwise} \qquad \text{Eq 4-19}$$

Effects of μ on hysteresis curve shape

In fitting the Preisach hysteresis model to data, the most important feature for the model to capture is the shape of the outer and inner hysteresis curves. Once the essential features of the experimental data curve shape have been reproduced, the model output can be scaled, shifted, or rotated as needed to provide the proper numerical match for the data. The exploration of the Preisach hysteresis model discusses four of the important features of the outer hysteresis curve shape: number of loops, direction of loops, separation between increasing and decreasing curves and *dT/du*, resulting in several important insights.

The proper choice of $\mu(\alpha,\beta)$ can produce single or multiple hysteresis loops, in either clockwise or counterclockwise directions. The separation between increasing and decreasing curves can be written as an integral of $\mu(\alpha,\beta)$ over a region in the top left corner of the α–β space, as shown in Figure 4-6 and Equation 4-7. Thus, high amplitudes of $\mu(\alpha,\beta)$ in the top left corner of the α–β space, where α is near α_0 and β is near $-\alpha_0$, result in large separations between the increasing and decreasing output curves.

It should be noted that in Examples 3 and 4, $\mu(\alpha,\beta)$ was restricted to the case where $\mu(\alpha,\beta)=f(\alpha)+g(\beta)$, thus having no dependence on products of α and β. The restriction is arbitrary, but it simplifies analytical calculation of model outputs and aids in understanding how the functions specified for $\mu(\alpha,\beta)$ impact the separation between increasing and decreasing output curves and the slope of the output curve, *dT/du*.

4.2.2 Numerical implementations of the Preisach hysteresis model

4.2.2.1 Preisach hysteresis model identification

The central premise of the Preisach model identification methods is that the difference between two output values is equal to the integral of $\mu(\alpha,\beta)$ over a region whose bounds are known from the input history. Calculating the integrated $\mu(\alpha,\beta)$ for a sufficient number of regions allows the Preisach model to be implemented to predict the hysteretic system's output as a function of its input and input history. The following derivation shows how the integral of $\mu(\alpha,\beta)$ over a region with known bounds can be calculated from the difference between two output values.

The input to the hysteresis transducer begins below the low input limit, $-\alpha_0$, and increases to a value u_1, which is below the high input limit. The corresponding output, when $u=u_1$, is $T=T_1$, as shown in Figure 4-9*A*. In the α-β plane, the S^+/S^- boundary lies at $\alpha=u_1$, as shown in Figure 4-9B. The input then decreases from $u=u_1$ to $u=u_2$, and a vertical segment is added to the S^+/S^- boundary at $\beta=u_2$, as shown in Figure 4-9C. The output when $u=u_2$ is $T=T_2$.

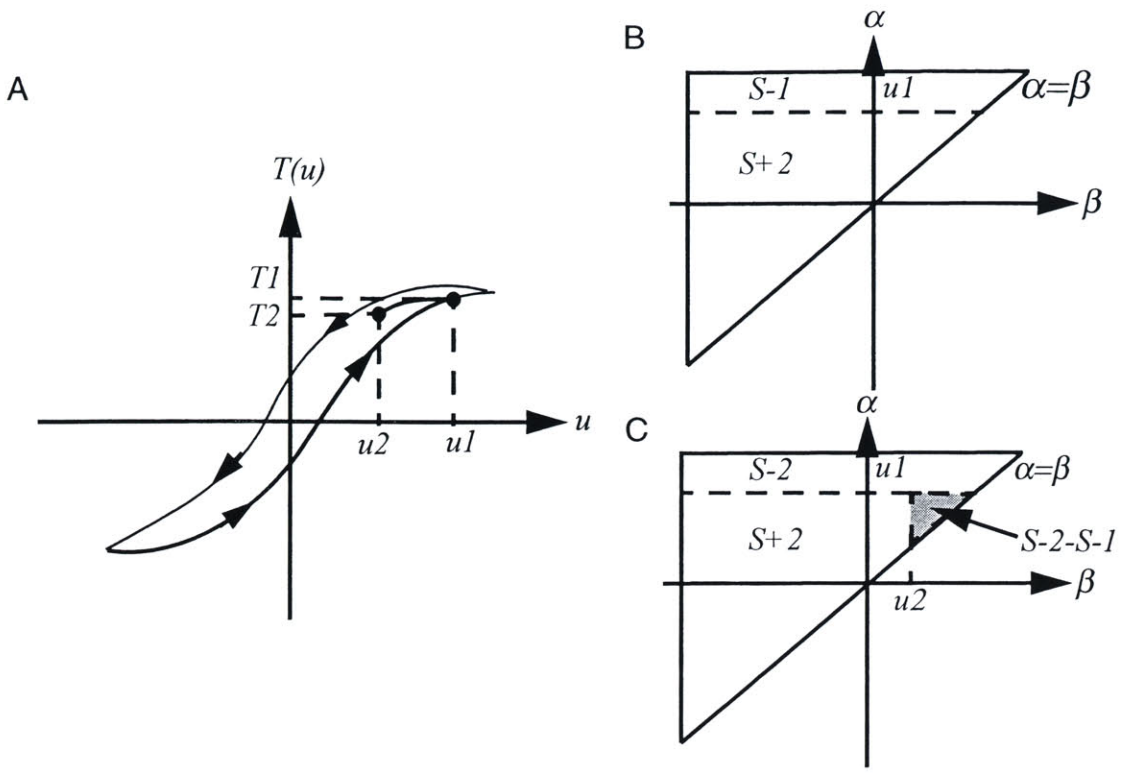


Figure 4-9. Calculation of the integral of $\mu(\alpha,\beta)$ over a small triangle based on output differences

According to Equation 4-2, the output $T(u)$ is equal to the integral of $\mu(\alpha,\beta)$ over the S^+ region minus the integral of $\mu(\alpha,\beta)$ over the S^- region. Using Equation 4-2 to calculate the difference between T_1 and T_2 results in the following expression for the output difference, in terms of the S^+ and S^- regions at $u=u_1$ and $u=u_2$.

$$T_1 - T_2 = T_{max} - 2\int_{S^-_1}\int \mu(\alpha,\beta)d\alpha d\beta - \left[T_{max} - 2\int_{S^-_2}\int \mu(\alpha,\beta)d\alpha d\beta\right] \qquad \text{Eq 4-20}$$

$$T_1 - T_2 = 2\int_{S^-_2 - S^-_1}\int \mu(\alpha,\beta)d\alpha d\beta$$

The difference between integration areas, S^-_2-S^-_1, for $u=u_2$ and $u=u_1$, is the shaded triangle shown in Figure 4-9C which has vertices at (u_1,u_2), (u_1, u_1) and (u_2, u_2). When the input decreases from u_1 to u_2, the difference in output values is equal to the integral of $\mu(\alpha,\beta)$ over the small shaded triangle shown in Figure 4-9C.

To determine $\mu(\alpha,\beta)$ from output differences, Mayergoyz[39] suggests defining the quantity

$$F(u_1, u_2) = T_1 - T_2 = 2 \int_{S^-_2 - S^-_1} \int \mu(\alpha, \beta) d\alpha d\beta . \qquad \text{Eq 4-21}$$

Then $\mu(\alpha,\beta)$ is given by

$$\mu(u_1, u_2) = \frac{1}{2} \frac{\partial^2}{\partial u_1 \partial u_2} F(u_1, u_2) . \qquad \text{Eq 4-22}$$

Taking two derivatives of measured data values would amplify random noise to unacceptable levels, so the method that Mayergoyz[39] recommends is not practical.

An alternative Preisach model identification scheme, developed by Doong and Mayergoyz[16] and further explained by Ge and Jouaneh[20], is based on the same premise, but avoids differentiating data in the identification step and double integrations in the implementation step. This method avoids differentiation and integration by calculating the integral of $\mu(\alpha,\beta)$ over a collection of triangles in α–β space from output differences, then uses sums and differences of the triangle integrals to construct the output for any input history.

According to Equation 4-20, the integral of $\mu(\alpha,\beta)$ over a triangle bounded by the $\alpha=\beta$ line and the $\alpha=u_1$, $\beta=u_2$ point is equal to T_1-T_2, as long as the input increased from its lower limit to u_1, then reversed direction and decreased to u_2, with no other direction reversals. This relationship allows the integral of $\mu(\alpha,\beta)$ to be calculated based on the appropriate output differences for triangles bounded by the $\alpha=\beta$ line and any point inside the large triangle over which $\mu(\alpha,\beta)$ is defined. The quantity $X(\alpha_1,\beta_1)$ is defined as the integral of $\mu(\alpha,\beta)$ over a triangle bounded by α_1, β_1 and the $\alpha=\beta$ line. If u increases from its low limit, reverses direction at $u=\alpha_1$, then decreases to β_1, $X(\alpha_1,\beta_1)$ is given by

$$X(\alpha_1, \beta_1) = T(u = \alpha_1) - T(u = \beta_1) = 2 \int_{\beta_1}^{\alpha_1} \int_{\beta}^{\alpha_1} \mu(\alpha, \beta) d\alpha d\beta \qquad \text{Eq 4-23}$$

For input data similar to that shown in Figure 4-10A, $X(\alpha_1,\beta_1)$ can be calculated for a grid of α_1,β_1 points that cover the entire $-\alpha_0<\alpha<\alpha_0$, $-\alpha_0>\beta<\alpha_0$, $\alpha>\beta$ triangle. Figure 4-10A shows the input $u(t)$ increasing from a value below α_0 to three different maximum values: α_1, α_2, and α_3. The input then descends from the maximum to intermediate values β_1, β_2, and β_3. The difference between the output at $u=\alpha_1$ and $u=\beta_1$ is equal to $X(\alpha_1,\beta_1)$, which is also equal to the integral of $\mu(\alpha,\beta)$ over the small triangle bounded by α_1, β_1 and the $\alpha=\beta$ line. Likewise, $T(u=\alpha_2)$-$T(u=\beta_2)$ and $T(u=\alpha_3)$ -$T(u=\beta_3)$ are equal to $X(\alpha_2,\beta_2)$ and $X(\alpha_3,\beta_3)$. Using this method, $X(\alpha,\beta)$ can be calculated for a mesh of α, β points.

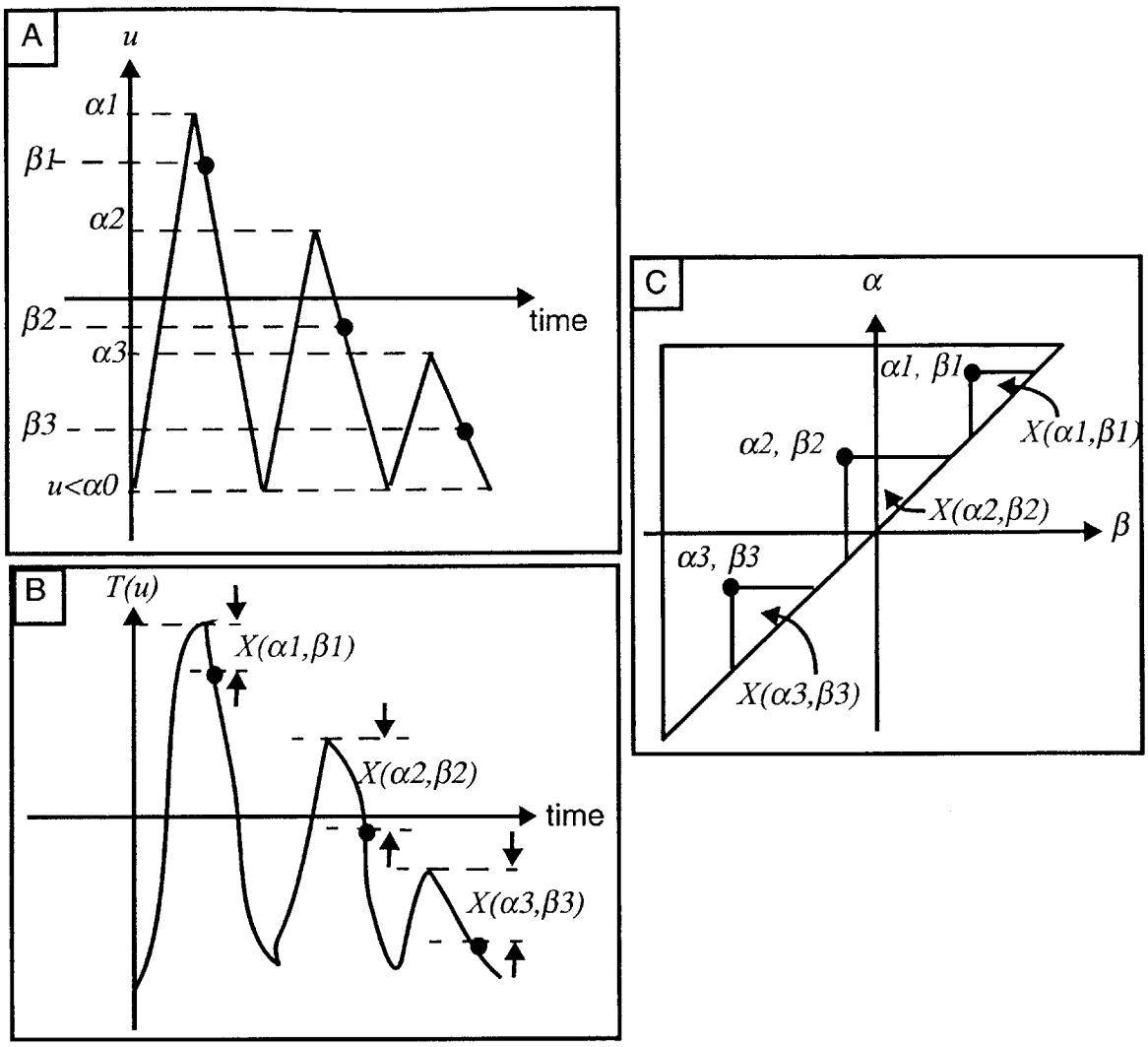


Figure 4-10. Calculation of $X(\alpha,\beta)$ from output differences. A.) Input $u(t)$. B.) Output $T(t)$. C.) Regions in the α–β plane where $\mu(\alpha,\beta)$ is integrated to get $X(\alpha,\beta)$.

4.2.2.2 Identification procedures

The identification process for the Preisach hysteresis model is the determination of the weighting coefficients $\mu(\alpha,\beta)$ that multiply the primitive hysteresis transducers, based on experimental data. Because the outputs of a hysteretic system depend on the input history, the experimental data that is used for hysteresis model identification should include transitions between several different minima and maxima. According to Mayergoyz[39], identifying the Preisach model requires a "complete" set of first order transition curves. First order transition curves are the input-output curves that occur after a single direction reversal in the input. The completeness of a set of input-output curves depends on the desired resolution in α and β for the model coefficients.

The space suit torque-angle database described in Chapter 3 is based on joint angle trajectories that were generated by human test subjects. While this experimental method insures that torque-angle data is collected using motions that are realistic for humans to perform, it precludes the systematic variations in joint angle amplitudes that are needed to generate an extensive set of first order transition curves. To provide suitable data for identifying a hysteresis model, torque and angle data was collected on the space suit for a set of first order transition curves for several joints, by commanding the robot to move between several angle minima and maxima. The transition curve data was obtained for elbow flexion, hip flexion, hip abduction, knee flexion, ankle rotation and ankle flexion motions. An example of the transition curve data is shown in Figure 4-11. The elbow flexion data increases to 7 different maxima, then decreases to zero in Figure 4-11A. The torque-angle hysteresis curves that result from these motions are shown in Figure 4-11B.

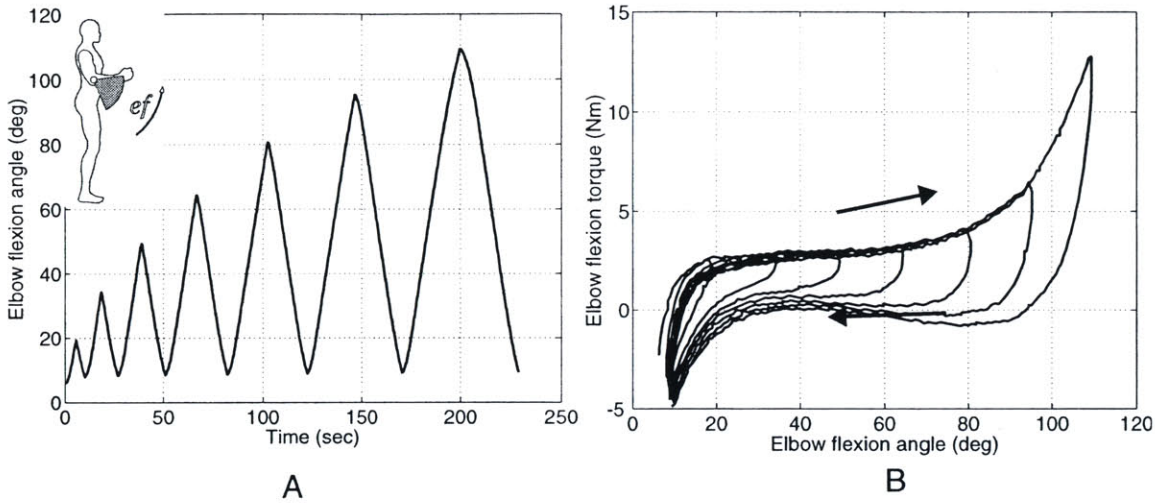


Figure 4-11. Transition curve data for space suit elbow joint. A.) Elbow flexion angle increases to 7 different maxima and returns to 0. B.) Hysteresis curve generated by the angle trajectory in A.

The choice of independent and dependent variables in the torque-angle database differs from that used in other hysteresis modeling work in the literature[8, 16, 20, 21, 33, 39, 47, 51, 52]. Because the robot followed joint angle trajectories supplied by the human subjects and recorded the joint torques necessary to perform the motions, angle is the independent variable and torque is the dependent variable in the data set. Conventionally, torque is considered to cause changes in angular position, so torque is usually the independent variable and position is the dependent variable. Examples of hysteresis model identification in the literature use the conventional causality relationship, with voltage as an input and strain as an output for piezoceramic actuators or magnetic field strength as an input and magnetization as an output for magnetization processes. In the conventional case, input and output values travel around the hysteresis loops in a counterclockwise direction, because of the choice of independent and dependent variables.The primitive hysteresis transducer used in the Preisach model, shown in Figure 4-1, is also counterclockwise. For the space suit data, however, the hysteresis loops are clockwise, when torque is plotted vs. angle. Rahn dealt

with this issue in fitting a Preisach model to space suit torque angle data by inverting the direction of the primitive hysteresis transducer, so that the upward switching threshold was less than the downward switching threshold and the hysteresis loop was counterclockwise, to agree with experimental data. However, the reversed-transducer implementation gives ambiguous results when the input direction reverses for input values between the switching thresholds. Even though the hysteresis curves in the space suit data are clockwise, it is not necessary to modify the Preisach model, because Examples 3 and 4 in Section 4.2.1.2 demonstrate that the conventional Preisach model can reproduce clockwise hysteresis curves.

A detailed, but not exhaustive, set of first order transitions curves in joint torque and angle, which were obtained by moving the robot's joints between several minima and maxima, are used to identify the Preisach model's weighting coefficients. Although the choice of independent and dependent variables is unusual, the examples derived in Section 4.2.1.2 demonstrate that no modifications to the Preisach model are necessary to apply it to the space suit joint torque-angle data set.

The Preisach model identification process is accomplished using the Doong and Mayergoyz[16] method in the Matlab function idx.m. A torque-angle data file similar to the one shown in Figure 4-11 is loaded. The idx.m function identifies the maxima and minima of the input angle data. The maxima of the angle data are the α values. One α value exists for each angle maximum that is followed by decreasing angle values. As the angle decreases from each maximum, the angle values in the descending segments are the β values corresponding to the α value of the maximum. X is calculated by differencing torque values. $X(\alpha,\beta)$ is the torque at the previous maximum, α, minus the torque at the current, decreasing, angle value, β. The list ofα values is assembled into a column vector with n rows. The β values are assembled into a matrix with one row for each a value and enough columns to accommodate the longest list of β values. $X(\alpha,\beta)$ values are put into a matrix that has the same dimensions as the β matrix. Because $\alpha>\beta$, the β and X matrices are triangular, and filler values are inserted to occupy the remaining places in the rectangular matrices. The resulting a vector and β and X matrices are set up so that for row i and column j, $X(i,j)=X(\alpha(i),\beta(i,j))$. The α vector and β and X matrices are saved in a file which is used in implementing the model.

Four replications of the first order transition curves were obtained for each joint, and idx.m was used to fit model coefficients to one replication only, so that the other three replications could be used for model validation.

4.2.2.3 Implementing the identified Preisach model

If $X(\alpha,\beta)$ is known for all $-\alpha_0<\alpha<\alpha_0$, $-\alpha_0<\beta<\alpha_0$, $\alpha>\beta$, then $X(\alpha,\beta)$ values can be added and subtracted to construct any Preisach model output, provided that the boundary between S^+ and S^- is drawn according to the rules described in Section 4.2.1.1. The rules for boundary drawing force the S^+/S^- boundary to be piecewise linear, with only horizontal and vertical links. These stairstep contours can be constructed from triangles, as Figure 4-12 shows.

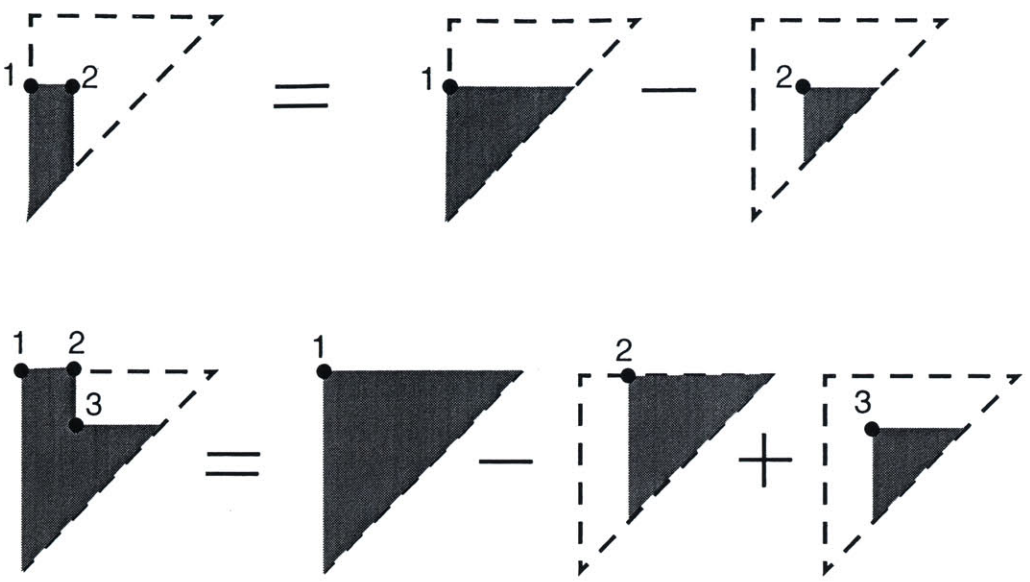


Figure 4-12. Stairstep contours reconstructed from triangles bounded by the α=β line and a third vertex within the bounding triangle.

As Figure 4-12 shows, if $X(\alpha,\beta)$ is known for all allowable α and β, and the S^+/S^- boundary has n vertices: (α_1, β_1), (α_2, β_2),...(α_n,β_n), then the integral of $\mu(\alpha,\beta)$ over S^+ is given by:

$$\int_{S^+}\int \mu(\alpha,\beta)d\alpha d\beta = \frac{1}{2}\sum_{i=1}^{n}(-1)^{i+1}X(\alpha_i,\beta_i) \qquad \text{Eq 4-24}$$

Since the maximum output T_{max} is known and

$$T_{max} = \int_{S^+}\int \mu(\alpha,\beta)d\alpha d\beta + \int_{S^-}\int \mu(\alpha,\beta)d\alpha d\beta = X(\alpha_0,-\alpha_0), \qquad \text{Eq 4-25}$$

the Preisach model output for any input history can be calculated from the boundary vertex locations (α_i, β_i) and the small triangle integrals $X(\alpha_i,\beta_i)$ evaluated at the vertices.

$$T = -X(\alpha_0,-\alpha_0) + \sum_{i=1}^{n} -1^{n+1}X(\alpha_i,\beta_i) \qquad \text{Eq 4-26}$$

The $X(\alpha,\beta)$ values are calculated from experimental data. To assemble a grid of α, β points, the input increases from the low limit to a series of maxima, then decreases back to the low limit, as shown in the example data in Figure 4-11 and schematically in Figure 4-10A. The α values that this experiment supplies are the input maxima, and the β values are the input values as the input decreases. $X(\alpha,\beta)$ is equal to the output at the last maximum minus the current output. The scheme shown in Figure 4-10A for generating first-order transition curves should result in a finer resolution in β points than in α points, because each cycle of input generates only one α value and many β values. The β resolution is limited only by the measurement resolution, sampling rate, and speed, while α resolution is limited by the spacing of maxima in the input. Additionally, different β values are generated for each α value, so that the grid of α, β values is not evenly spaced.

$X(\alpha,\beta)$ can be interpolated from the unevenly-spaced grid using a nearest-neighbors interpolation method. $X(\alpha,\beta)$ is interpolated between the four experimentally-determined α_j, β_j points surrounding α and β using a weighted average based on the distance, dj, between the desired α, β and the α_j, β_j points. When α, β is close to the $\alpha=\beta$ line, only three surrounding points may exist; in this case, $X(\alpha,\beta)$ is interpolated between the three existing points. Figure 4-13A shows the four points surrounding α, β and Figure 4-13B shows the case when only three points exist.

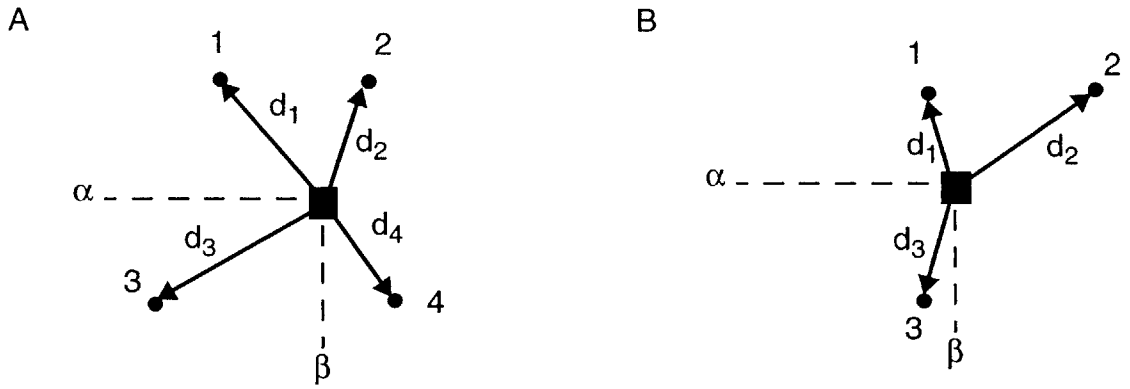


Figure 4-13. Interpolation geometry for $X(\alpha,\beta)$. A) Four point case B) Three point case.

For the four-point interpolation, the interpolated $X(\alpha,\beta)$ is given by

$$X(\alpha,\beta) = \frac{(d_2d_3d_4)x_1 + (d_1d_3d_4)x_2 + (d_1d_2d_4)x_3 + (d_1d_2d_3)x_4}{d_2d_3d_4 + d_1d_3d_4 + d_1d_2d_4 + d_1d_2d_3}. \qquad \text{Eq 4-27}$$

In the three-point case, $X(\alpha,\beta)$ is given by

$$X(\alpha,\beta) = \frac{(d_2d_3)x_1 + (d_1d_3)x_2 + (d_1d_2)x_3}{d_2d_3 + d_1d_3 + d_1d_2}. \qquad \text{Eq 4-28}$$

Using the interpolation process described above, $X(\alpha,\beta)$ can be calculated for any α, β within the bounding triangle. If the contour between the S^+ and S^- regions is drawn according to Section 4.2.1.1, then the contour vertices α_i and β_i are known. Using Equation 4-26, the output for any set of α_i and β_i can be calculated.

4.2.2.4 Preisach hysteresis model error analysis

Because the hysteresis model coefficients are determined from experimental data, random errors in the experimental data lead to random errors in the model output whose statistical properties can be predicted. The error analysis provides a method for generating confidence intervals for the Preisach model output if the model is identified and implemented according to the Doong and Mayergoyz[16] method.

According to Equation 4-26, the Preisach model output is the sum of positive and negative $X(\alpha,\beta)$ values at the vertices of the S^+/S^- contour. Assuming that the errors in the experimental torque and angle data that was used to calculate $X(\alpha,\beta)$ are random, white noise, errors in $X(\alpha,\beta)$ should be uncorrelated with errors in X at other values of α and β. Thus, the variance of the model output T is equal to the sum of the variances of the individual $X(\alpha,\beta)$ values that were summed to obtain T.

$$var(T) = \sum_{i=1}^{n} var(X(\alpha_i, \beta_i)) \qquad \text{Eq 4-29}$$

This relation implies that the variance of T increases when the number of vertices in the S^+/S^- boundary increases. The next step is to determine the variance of $X(\alpha,\beta)$. Random errors in $X(\alpha,\beta)$ come from two sources: errors in measured torques, which lead to errors in X and errors in measured angles, which lead to errors in α and β.

The $X(\alpha,\beta)$ values are calculated from the difference of two measured torque values, according to Equation 4-23. Consequently, the variance in X due to torque errors, $var(X_T)$, is given by:

$$var(X_T) = var(\text{torque1} - \text{torque2}) = 2var(\text{torque}) \qquad \text{Eq 4-30}$$

Errors in X due to errors in angle measurements vary with α and β. The change in X due to changes in α and β is approximately given by

$$\Delta X = \Delta\alpha\frac{\partial X}{\partial \alpha} + \Delta\beta\frac{\partial X}{\partial \beta} \qquad \text{Eq 4-31}$$

If $\Delta\alpha$ and $\Delta\beta$ are both equal to the standard deviation of angle measurement errors, then the variance of errors in X due to angle errors, $var(X_A)$, is

$$var(X_A) = var(\text{angle})\left(\frac{\partial X}{\partial \alpha} + \frac{\partial X}{\partial \beta}\right)^2 \quad \text{Eq 4-32}$$

Then the variance of X due to both angle and torque errors is

$$var(X) = 2var(\text{torque}) + var(\text{angle})\left(\frac{\partial X}{\partial \alpha} + \frac{\partial X}{\partial \beta}\right)^2 \quad \text{Eq 4-33}$$

Summing the variances of the X values at the n S^+/S^- boundary vertices results in the variance of the model output T.

$$var(T) = (2n)var(\text{torque}) + var(\text{angle}) \sum_{i=1}^{n} \left(\frac{\partial X_i}{\partial \alpha_i} + \frac{\partial X_i}{\partial \beta_i}\right)^2 \quad \text{Eq 4-34}$$

If the errors in torque and angle measurements are Gaussian, then 95% of the data points are expected to fall within 2 standard deviations of the mean. The 95% confidence interval is given by

$$\text{confidence interval} = T \pm 2\sqrt{var(T)} \quad \text{Eq 4-35}$$

4.2.2.5 Preisach model implementation procedures

The Preisach model is identified and implemented using Matlab functions specifically written for this purpose. The Matlab function xmodel.m, listed in Appendix B, implements the Preisach model. The xmodel.m function takes the angle trajectory as an input and outputs the torque value and torque variance predicted by the model at each time step. Xmodel.m loads the file with the X, α, and β values determined from the experimental data by idx.m, then calls the function ab.m to determine the vertices of the S^+/S^- contour at each time step, according to the boundary-drawing rules discussed in Section 4.2.1. It then calculates $X(\alpha,\beta)$ at each of the vertices, interpolating between the experimentally-determined α, β values using Equation 4-27 and Equation 4-28 in interpx.m. Finally, the $X(\alpha,\beta)$ values at the boundary vertices are summed according to Equation 4-26 to obtain the model output torque. The model output variance is calculated in errx.m at each time step according to Equation 4-34, using a torque error variance of 1 Nm^2 and an angle error variance of 4 deg^2.

The Preisach model identification scheme is based on the idea that the integral of the weighting coefficients $\mu(\alpha,\beta)$ can be calculated based on output differences over regions whose bounds are known from the input minima and maxima. Although Mayergoyz recommends differentiating the input differences to calculate $\mu(\alpha,\beta)$ directly, then integrating $\mu(\alpha,\beta)$ to obtain the model output, this method is impractical. Doong and Mayergoyz, as well as Ge and Jouaneh describe an alternative method for Preisach model identification and implementation that avoid both differentiation and integration. The Doong and Mayergoyz method, which is implemented in the Matlab function idx.m to identify the model

coefficients from experimental data, calculates the integral of the weighting function $\mu(\alpha,\beta)$ over triangles bounded by the $\alpha=\beta$ line and any allowable point in the $\alpha-\beta$ space. These triangle integrals, which are obtained from experimental data, can be summed to calculate the Preisach model output for any input history.

4.2.3 Results: Hysteresis modeling

Preisach model coefficients $X(\alpha,\beta)$ were fitted to the first-order transition curves that were produced for space suit torque-angle data, as discussed in Section 4.2.2.1 and Section 4.2.2.1. Model coefficients were obtained for elbow flexion, hip abduction, hip flexion, knee flexion, ankle rotation, and ankle flexion motions. These experimentally-determined model coefficients were then used to predict the torques required to bend the space suit joints for motions produced by space-suited human subjects, then the torque predictions generated by the model were compared to experimental data.

The Preisach model coefficients are defined in terms of the upward and downward switching thresholds, α and β, of the primitive hysteresis transducers. The model coefficients, $X(\alpha,\beta)$, can be plotted in three dimensions vs. α and β, over the triangular region for which α and β are defined: $-\alpha_0<\alpha<\alpha_0$, $-\alpha_0<\beta<\alpha_0$, $\alpha>\beta$. The experimentally-determined model coefficients for the elbow flexion, hip abduction, hip flexion, knee flexion, ankle rotation, and ankle flexion joints are plotted in Figure 4-14 - Figure 4-19.

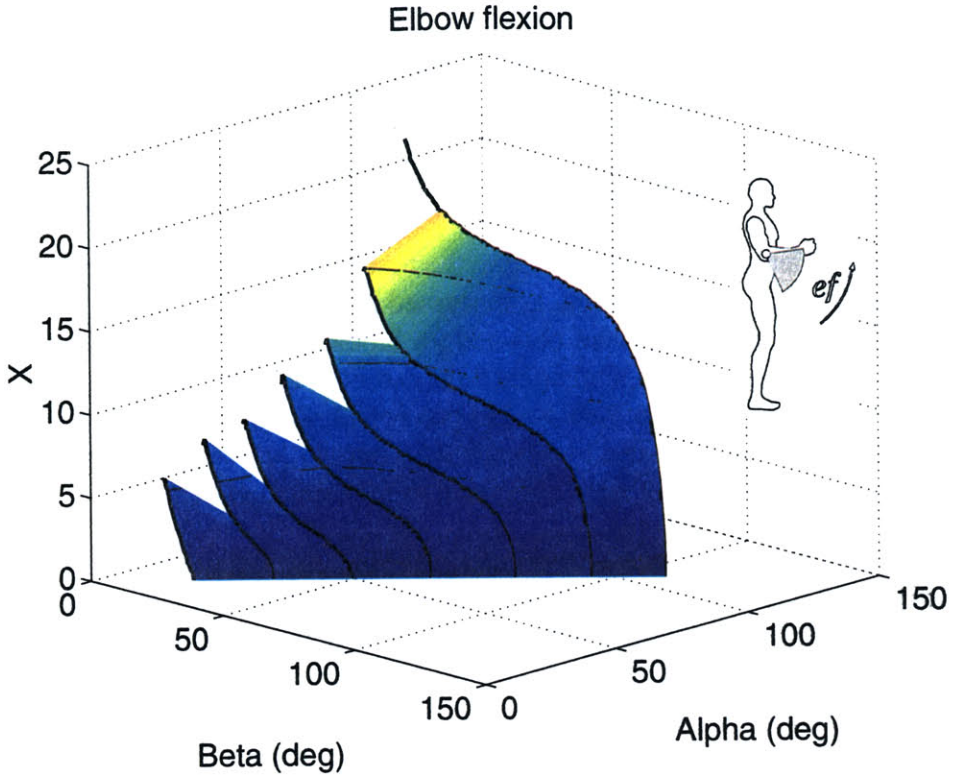


Figure 4-14. Preisach model coefficients plotted vs α and β for the elbow flexion joint.

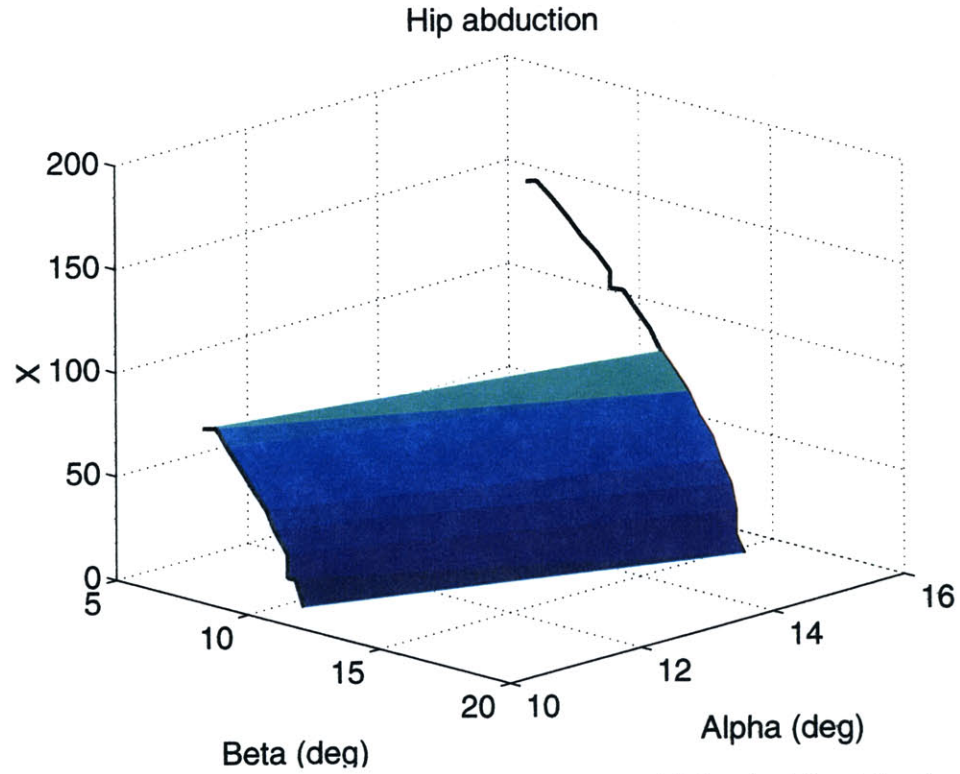


Figure 4-15. Preisach model coefficients plotted vs α and β for the hip abduction joint.

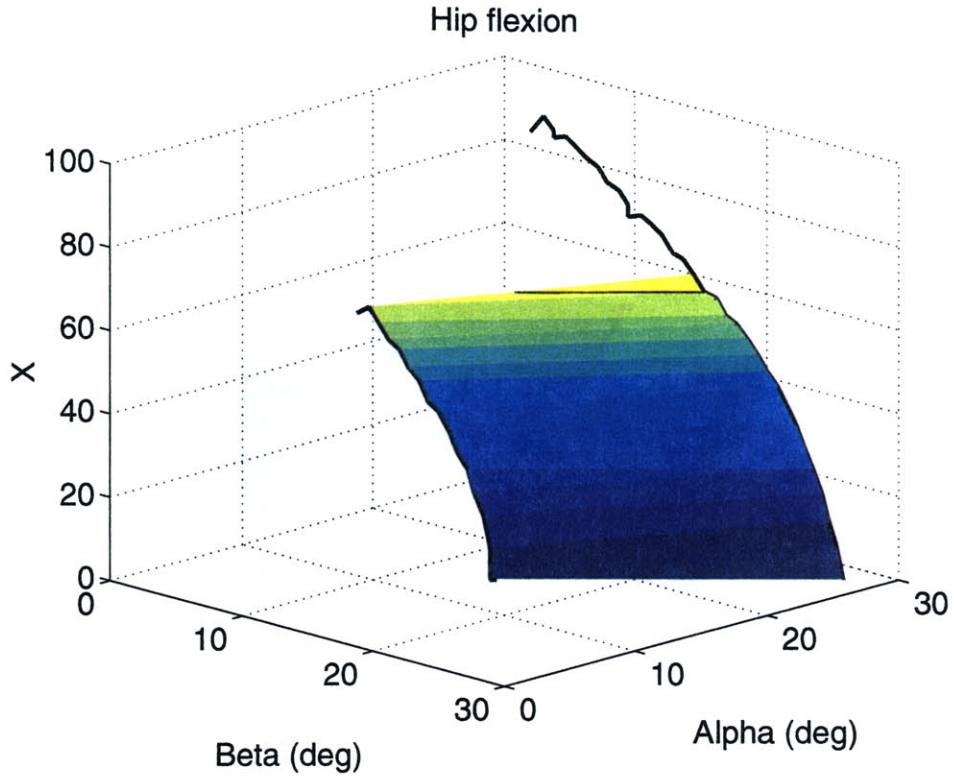


Figure 4-16. Preisach model coefficients plotted vs α and β for the hip flexion joint.

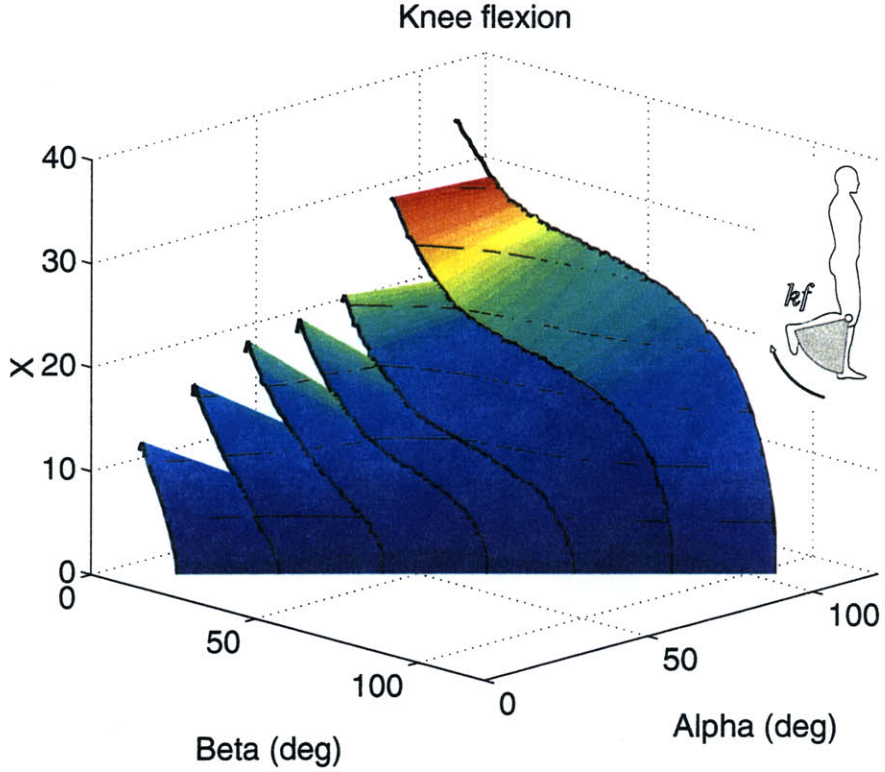


Figure 4-17. Preisach model coefficients plotted vs α and β for the knee flexion joint.

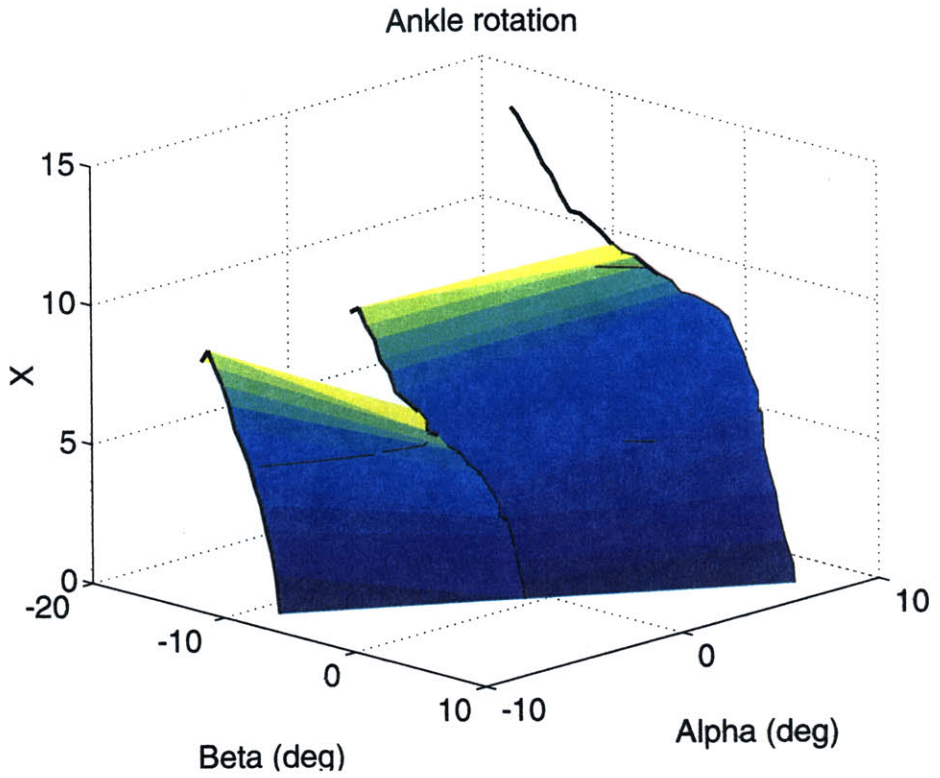


Figure 4-18. Preisach model coefficients plotted vs α and β for the ankle rotation joint.

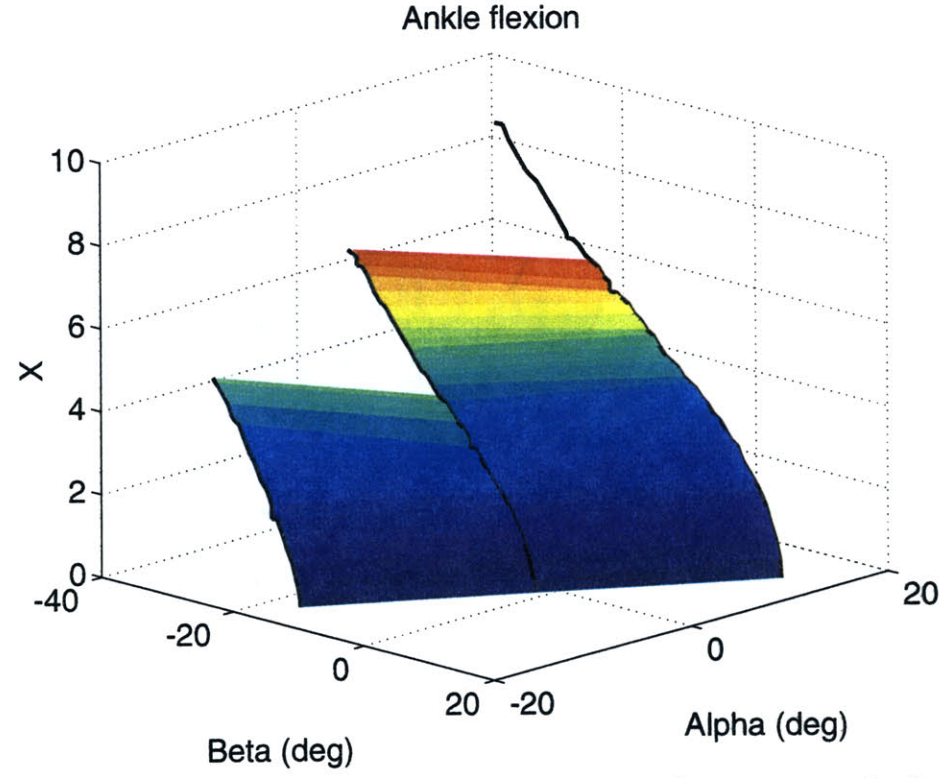


Figure 4-19. Preisach model coefficients plotted vs α and β for the ankle flexion joint.

The hysteresis models were implemented for each joint and compared to experimental torque-angle data for motions that were generated by space-suited human subjects, as described in Chapter 3. Joint angle data from the human subjects was used as an input for the hysteresis model, which generated a torque prediction as well as a 95% confidence interval on the prediction. The experimental joint torque data is compared to the model predictions and confidence intervals in Figure 4-20 - Figure 4-24. The hip abduction model-data comparison is shown in Figure 4-21 for hip abduction first-order transition curve data files, rather than data from human subjects, because the human subjects were not able to move the hip abduction joint more than a few degrees. The hip flexion model could not be compared to data because 80% of the angle data generated by human subjects was outside the angle range for the model. R^2 values for the model-data comparisons are listed in Table 4-1.

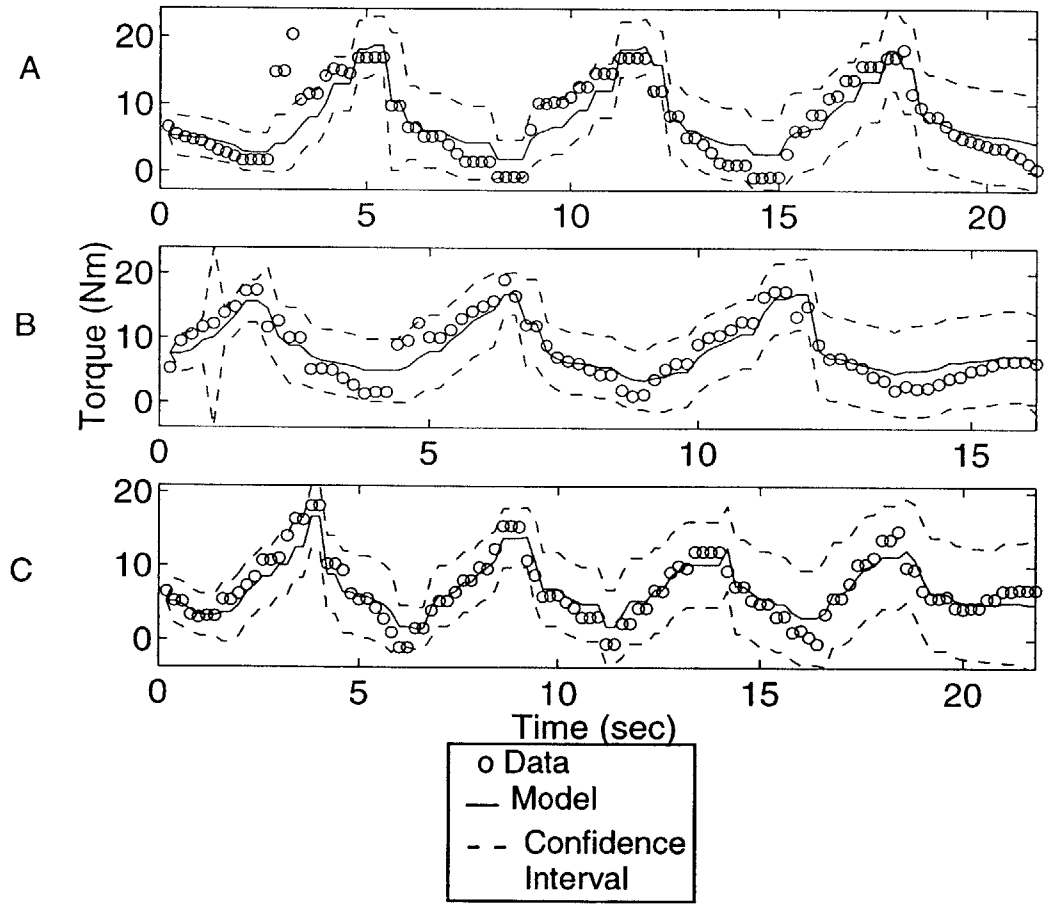


Figure 4-20. Elbow flexion torque data compared to hysteresis model output for A.) Subject B. B.) Subject C. C.) Subject E.

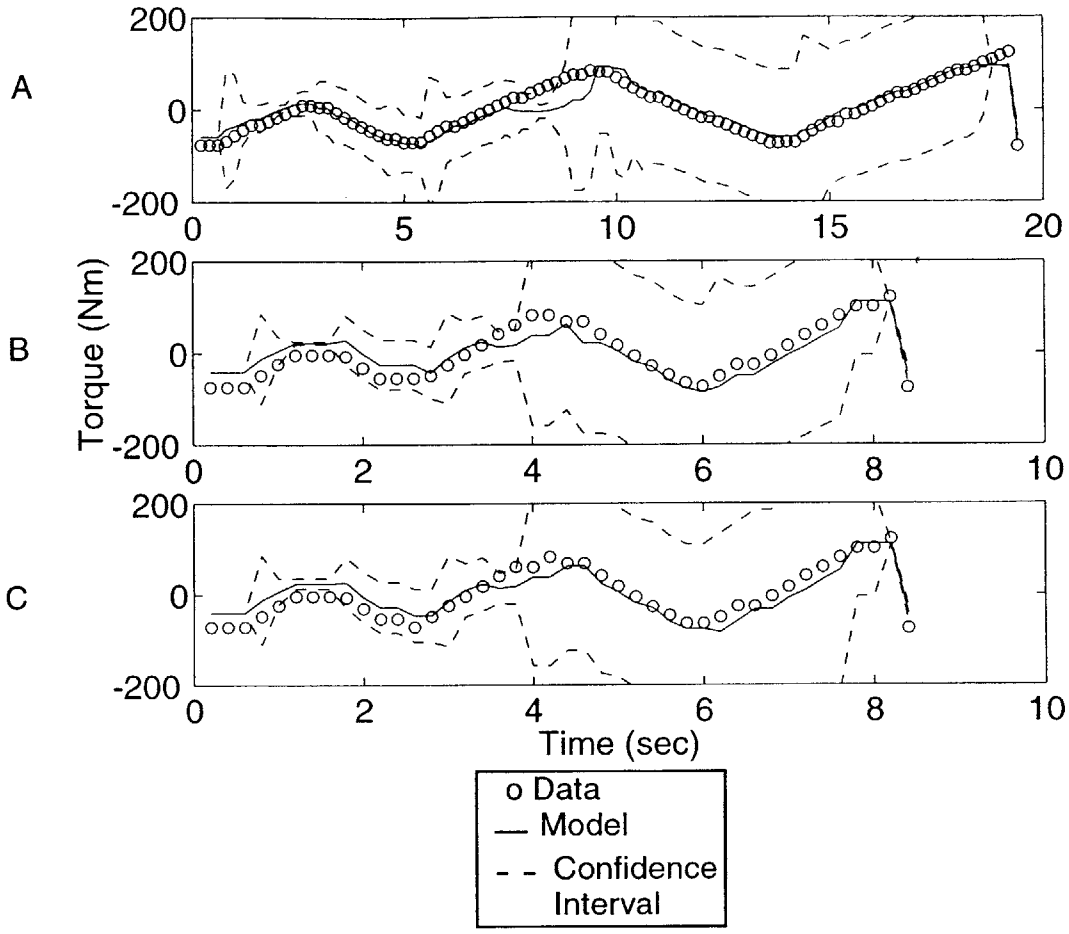


Figure 4-21. Hip abduction torque data compared to hysteresis model output for A.) File 2 B.) File 3. C.) File 4.

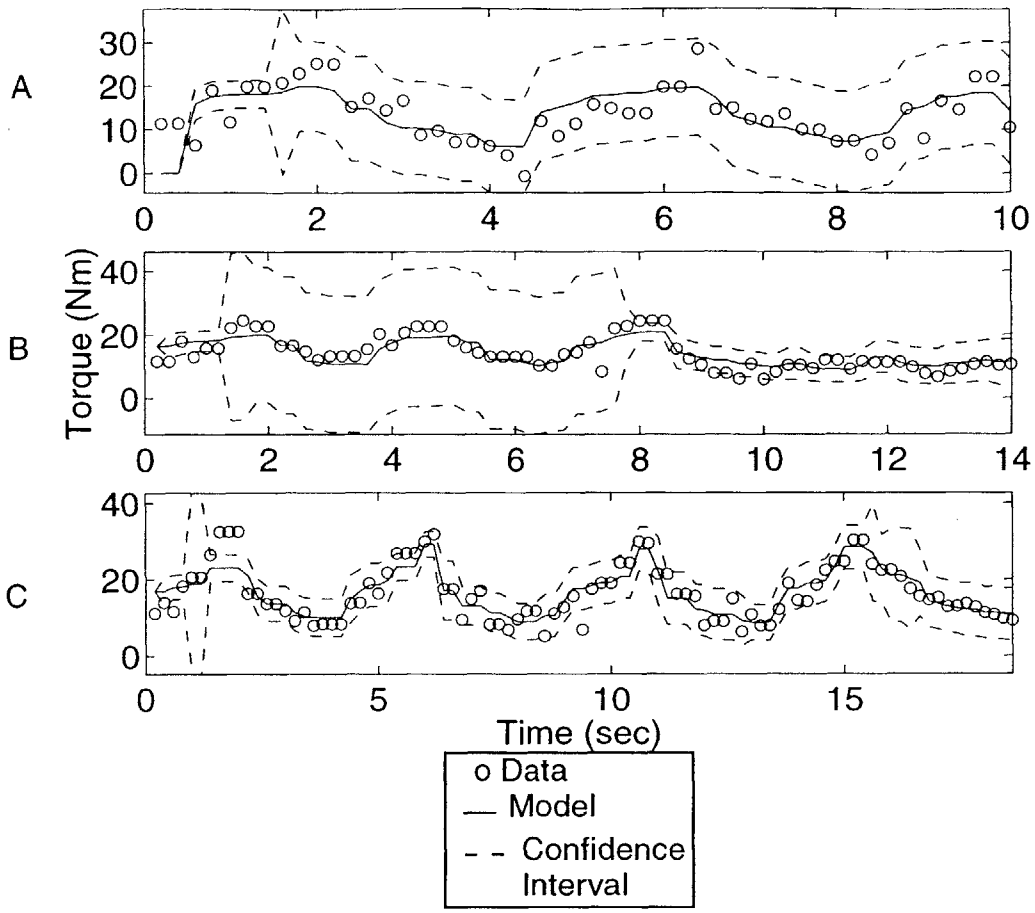


Figure 4-22. Knee flexion torque data compared to hysteresis model output for A.) Subject B. B.) Subject C. C.) Subject E.

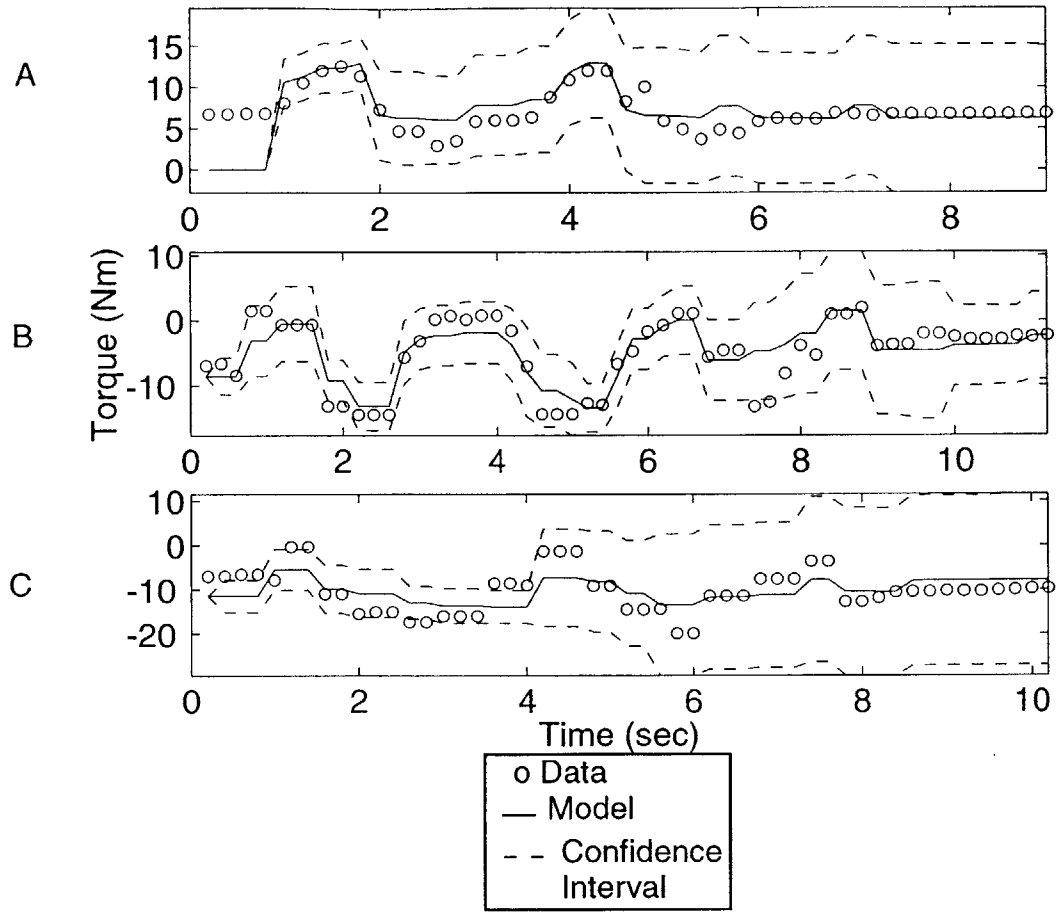


Figure 4-23. Ankle rotation torque data compared to hysteresis model output for A.) Subject B. B.) Subject C. C.) Subject E.

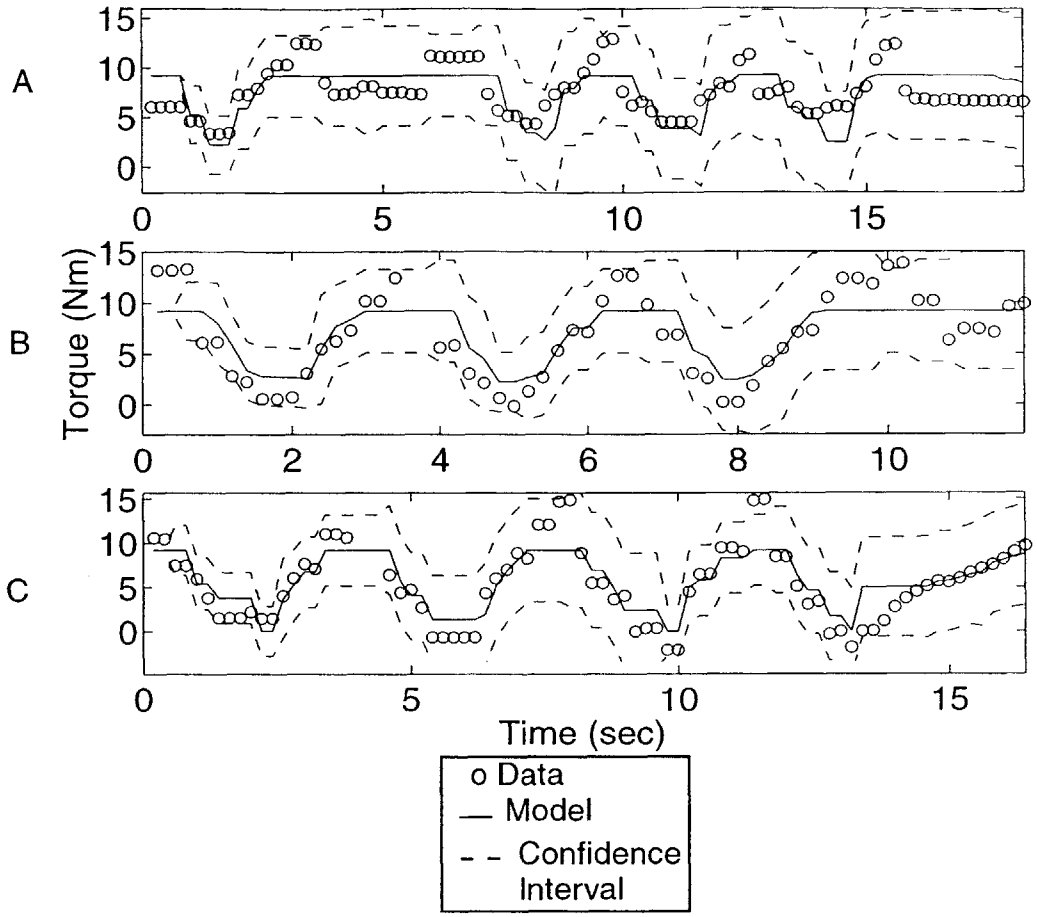


Figure 4-24. Ankle flexion torque data compared to hysteresis model output for A.) Subject B. B.) Subject C. C.) Subject E.

Table 4-1: R^2 values for model-data comparison

Joint	R^2
Elbow flexion	0.629
Hip abduction	0.776
Knee flexion	0.631
Ankle rotation	0.494
Ankle flexion	0.471

4.2.4 Discussion: Hysteresis modeling

The Preisach model outputs match well with data for elbow flexion, hip abduction and knee flexion, with $R^2 > 0.6$ for these joints. A comparison between the model output and data for ankle flexion in Figure 4-24 shows that the model output saturates and does not follow the torque data for torques above 9 Nm. This occurs because angles in the human-generated data files exceeded those in the first order transition curve data that was used to identify the Preisach model coefficients, according to the process described in Section 4.2.2.2. Saturation also occurred in the hip flexion data. More than 80% of the human-generated hip flexion angles exceeded the range of the model coefficients.

The error estimates generated appear to be a reliable, or at least conservative, estimate of the error standard deviations. In some cases, particularly in Figure 4-21 and Figure 4-22, the model's estimate of error variance suddenly increases. This occurs when the joint angle data momentarily reverses direction, adding at least two vertices to the S^+/S^- boundary. According to Equation 4-34, the model output error variance depends on the number of vertices in the boundary. Since there are more $X(\alpha,\beta)$ values summed to obtain the model's output, it is logical to expect that errors will increase when the number of vertices in the boundary increases.

The Preisach hysteresis model is a useful tool for predicting the torque needed to bend space suit joints as a function of time for arbitrary angle time histories. The ability to reproduce realistic space suit torques dynamically makes the Preisach model well-suited for use in dynamic simulation of EVA tasks. The experimental data reported in Chapter 3 as well as other investigators' data indicate that the torque-angle relationship for space suit joints is markedly hysteretic. Consequently, reproducing dynamic behavior of space suit joints requires a model that captures hysteresis; other modeling techniques such as linear regressions on angle and angular velocity will not accurately predict the effects of the angle history on the torque required to bend the space suit's joints.

The Preisach hysteresis model has some limitations in modeling space suit joints, however. To identify the model coefficients, it is necessary to vary the input between several distinct minima and maxima. Because of this, the model identification process is not well-suited to human-generated motions, although characteristics of human-generated motions, such as range and speed, can be incorporated into the input data for realism. Another limitation of the Preisach model is that the maximum input and output are set when the model coefficients are identified. When the model is implemented later, if the input exceeds the previously-set maximum, the output saturates and errors between the model prediction and actual torques may become large, as shown in Figure 4-24.

Modeling the hysteretic characteristics of space suit joints is essential for making accurate predictions of the torques required to bend the space suit joints in dynamic situations. This work demonstrates a new application of hysteresis modeling techniques to space suit joint mobility characteristics, resulting in the capability to accurately predict the torques needed to bend the space suit joints, for arbitrary joint angle time histories. The results of the

mathematical modeling work show that the Preisach model accurately reproduces the torques needed to bend the space suit joints in human-generated motions.

4.3 Physics-based models

While the mathematical modeling approach of Section 4.2, which is based on experimental data, generate numerical predictions of the torque required to bend existing space suit joints, physics-based models enable prediction of the torque-angle characteristics of space suit joints that have not yet been built. Currently, little theory exists on the bending performance of space suit joints. Theoretical and experimental work on other inflatable structures may explain space suit joint mobility. On the other hand, it is possible that the mobility features built into space suit joints are too complicated to be adequately described by a generic cylinder bending model. This physics-based modeling section addresses physical modeling of space suit joints by comparing two approximate models of bending pressurized cylinders to experimental data from space suits.

Section 4.3 describes the beam and membrane models, and compares the two approximate models in conceptual terms, then discusses common characteristics of the two models' implementations for space suit joints, including loading conditions and space suit parameter values. Torque predictions from the two models are compared to experimental space suit torque-angle data. Finally, the agreement between the two models and data is evaluated and implications for space suit design and recommendations for future research are discussed.

4.3.1 Beam model

The beam model, which was developed by Comer and Levy[12] and extended by Main, Peterson, and Strauss[35, 36, 37, 38], treats a pressurized cylinder as a long, slender member, loaded in a single plane, whose behavior is governed by elasticity and buckling phenomena. The central idea of the beam model is that fabric can sustain only extensional stresses; when the fabric wall of the beam is compressed, it wrinkles and does not contribute to the stiffness of the beam. The beam model predicts the extent of the wrinkled portion of the beam and integrates the extensional stress resultants over the tensioned portion of the beam to obtain a moment-curvature relationship for the pressurized beam.[12, 35, 38] Stress resultants are normalized by the thickness of the fabric, and expressed in terms of load per unit length in units of N/m, because fabric stress-strain behavior is not highly correlated with fabric thickness.

Stresses on the fabric wall of the cylinder are caused by internal pressurization and bending loads. For a cylinder with internal pressurization p, the stress on the fabric in the circumferential, or hoop, direction, σ_H, which is due to internal pressurization, is

$$\sigma_H = pr. \qquad \text{Eq 4-36}$$

The longitudinal stress, σ_L, on the fabric, which is due to both pressurization and applied bending moment M, varies with y, the transverse coordinate on the beam. For a beam of modulus E and moment of inertia I, σ_L is given by

$$\sigma_L = 2pr - \frac{M}{EI}y \qquad \text{Eq 4-37}$$

The applied loads given in Equation 4-36 and Equation 4-37 and the constitutive relations of fabric, given in Equation 4-38, which relate stresses and strains, determine the extent of wrinkling in the fabric. In general, fabric is assumed to be orthotropic, with Poisson ratio ν and different moduli, E_H and E_L in the hoop and longitudinal directions.

$$\varepsilon_L = \frac{\sigma_L}{E_L} - \frac{\nu\sigma_H}{E_L} \qquad \varepsilon_H = \frac{\sigma_H}{E_H} - \frac{\nu\sigma_L}{E_L} \qquad \text{Eq 4-38}$$

The beam model predicts that the fabric wall of the beam wrinkles when the local ε_L becomes negative. Equation 4-38 shows that the cross-coupling component in the constitutive relations makes the fabric wrinkle at positive values of σ_L when σ_H is positive. The hoop loading due to pressurization thus makes the fabric wrinkle sooner as the beam is bent. The longitudinal stress at which wrinkling occurs is given by:

$$\sigma_L = \nu pr \qquad \text{Eq 4-39}$$

When the fabric is wrinkled, its stress resultant is considered to be zero, because it cannot resist compressive loads. The longitudinal stress resultants when the beam model's wrinkling conditions are applied to a bent pressurized fabric cylinder are shown in Figure 4-25. The longitudinal stress is maximum on the outside of the bend and decreases linearly until it reaches the wrinkling limit, then the wrinkling condition sets it equal to zero.

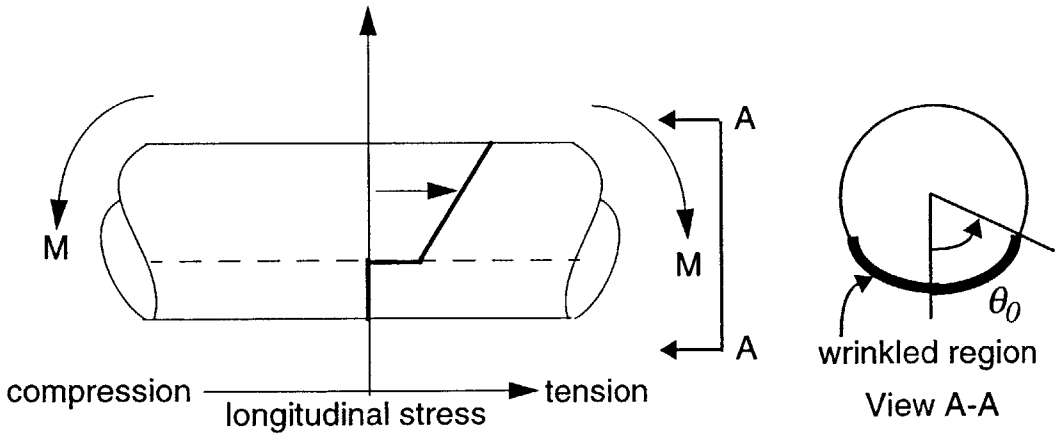


Figure 4-25. Longitudinal stress resultants in a bent pressurized fabric beam.

Using polar coordinates, with $\theta=0$ on the inside of the bend, at the center of the wrinkled region, and $\theta=\theta_0$ at the boundary of the wrinkled region, the longitudinal stress resultant reaches a maximum σ_{Lm} at $\theta=\pi$, and can be written as

$$\sigma_L = \frac{(\cos\theta_0 - \cos\theta)(\sigma_{Lm} - \nu pr)}{1 + \cos\theta_0} + \nu pr \qquad \text{for } \theta_0 < \theta < \pi$$
$$\sigma_L = 0 \qquad \text{for } 0 < \theta < \theta_0 \qquad \text{Eq 4-40}$$

Integrating the longitudinal stresses over the beam cross-section and applying force and moment balance, as derived in Chapter 2, results in the following moment-curvature relation for the partially-wrinkled beam.

$$K = \frac{M - 2\nu pr^3 \sin\theta_0}{Er^3[(\pi - \theta_0) + \sin\theta_0\cos\theta_0]} \qquad \text{Eq 4-41}$$

To determine the curvature of the beam, it is first necessary to find θ_0, using the relation derived in Chapter 2 between applied moment, M, internal pressure, p, and beam radius.

$$\frac{M}{pr^3} = \frac{\frac{\pi}{2}[(\pi - \theta_0) + \sin\theta_0\cos\theta_0] - \nu[(\pi - \theta_0)^2 - (\pi - \theta_0)\sin\theta_0\cos\theta_0 - (2\sin\theta_0)^2]}{\sin\theta_0 + (\pi - \theta_0)\cos\theta_0} \qquad \text{Eq 4-42}$$

The θ_0 that is calculated by solving Equation 4-42 numerically for specified M/pr^3 can then be substituted into Equation 4-41 to obtain the curvature resulting from the applied moment.

4.3.2 Membrane model

In contrast to the beam model, the membrane model treats the fabric cylinder wall as an inextensible material that transmits forces only along its surface and only in tension. Bending deflections of the structure result in changes in its cylinder's cross-sectional shape and the volume that it encloses. The cross-sectional shape of the fabric tube is determined by the assumptions of inextensibility and exclusively tensile loading. These assumptions result in reliable approximations of the tube's shape, even when deflections are large.

The membrane model uses a variational principle to relate applied bending moment, force applied at the tube's ends, and the bending angle of the pressurized fabric tube. The potential energy of a tube with volume V, internal pressure p, end force Q, bending moment M, bending angle ϕ and linear displacement δ is given by:

$$\Pi = -pV - Q\delta - M\phi \qquad \text{Eq 4-43}$$

When the system is at equilibrium, the potential energy is minimized. For the case of pure bending, $Q=0$ and $\delta=0$. The moment-angle relationship that minimizes the potential energy is calculated by holding the bending moment, M, fixed and differentiating the potential energy, Π, with respect to the bending angle ϕ. When $\frac{d\Pi}{d\phi} = 0$, the potential energy is minimized.

$$\frac{d\Pi}{d\phi} = -p\frac{dV}{d\phi} - M = 0$$

$$\Rightarrow M = -p\frac{dV}{d\phi} \qquad \text{Eq 4-44}$$

Equation 4-44 shows that the equilibrium bending moment-bending angle relationship is set by the relationship between the internal volume of the tube and the bending angle. The assumptions of inextensibility and tensile loading set the shape of the tube as a function of bending angle. Based on the four principles of membrane shape listed in Section 2.4.2, the shape of the bent tube can be determined. The geometry of a bent pressurized tube is shown in Figure 4-26. For a tube of radius R, the radius of the bent outer surface of the tube is $2R$. The inner surface at the bend folds inward, towards the center of the tube, forming a kink in the tube wall. Away from the kinked region, the tube is straight. Points P and S, which lie on the tube wall, are defined to aid in calculating the tube's cross-sectional area. Point P is located on the kink at the inside of the bend, and point S is the corresponding point on the outside of the bend. Point P and point S are the same distance away from the end of the tube. θ is the angle between a line drawn from the inside of the bend to point S and another line that is normal to the straight sides of the tube. The length of the bent portion of the tube on the outer edge is $4R\phi$.

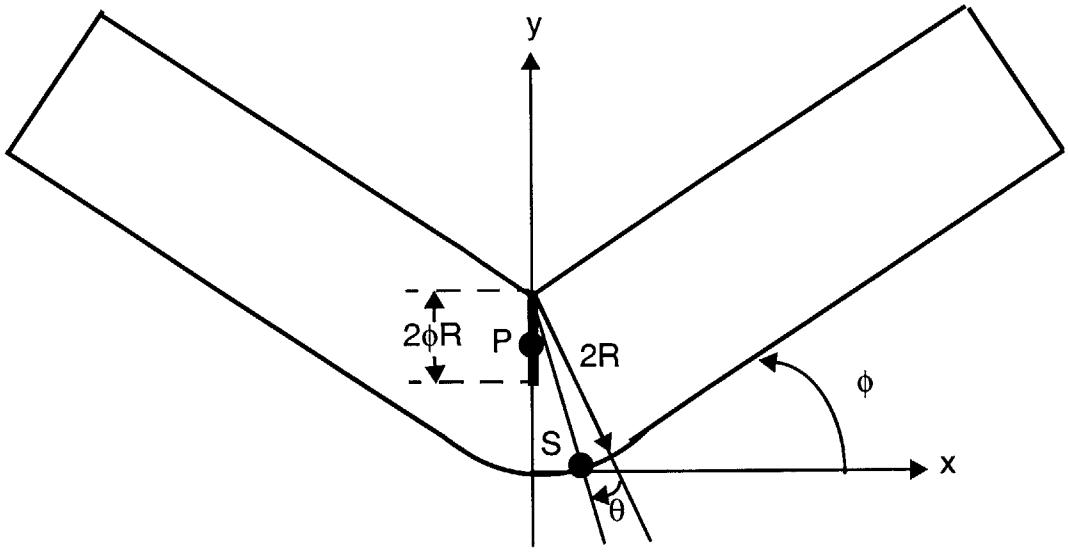


Figure 4-26. Bent tube geometry

The locations of points P and S are defined as follows.

$$\begin{aligned} x_P &= 0 \\ y_P &= 2R(1+\theta) \\ x_S &= 2R(1-\cos(\phi-\theta)) \\ y_S &= 2R\sin(\phi-\theta) \end{aligned} \qquad \text{Eq 4-45}$$

The cross-sectional shape and cross-sectional area of the tube can be determined based on the principles of membrane shape and the definitions of points P and S. The membrane shape principles define a rounded-rectangle shape for the tube's cross-section, shown in Figure 4-27. The distance between points P and S, which can be calculated from Equation 4-45, and the continuity in slope between the rectangular portion and the circular portion of the cross-section allow the cross-sectional area to be calculated as a function of θ.

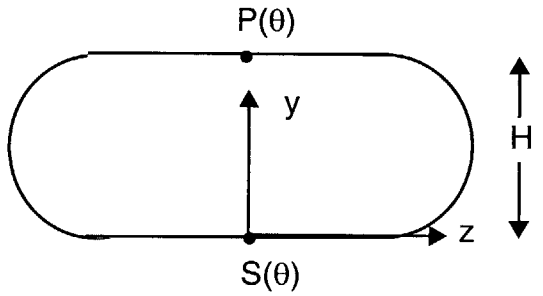


Figure 4-27. Cross-section of the bent tube.

The cross-sectional area of the tube is given by

$$A(\theta) = \pi\left(R^2 - \left(R - \frac{H(\theta)}{2}\right)^2\right) \qquad \text{Eq 4-46}$$

Equation 4-45 and the cross-sectional shape from Figure 4-27 allow the overall shape of the bent tube to be calculated as a function of bending angle. Figure 4-28 shows the tube's longitudinal and cross-sectional shape for bending angles from 10 degrees - 40 degrees. The shape of the tube is determined by the inextensibility constraint and is independent of the internal pressure.

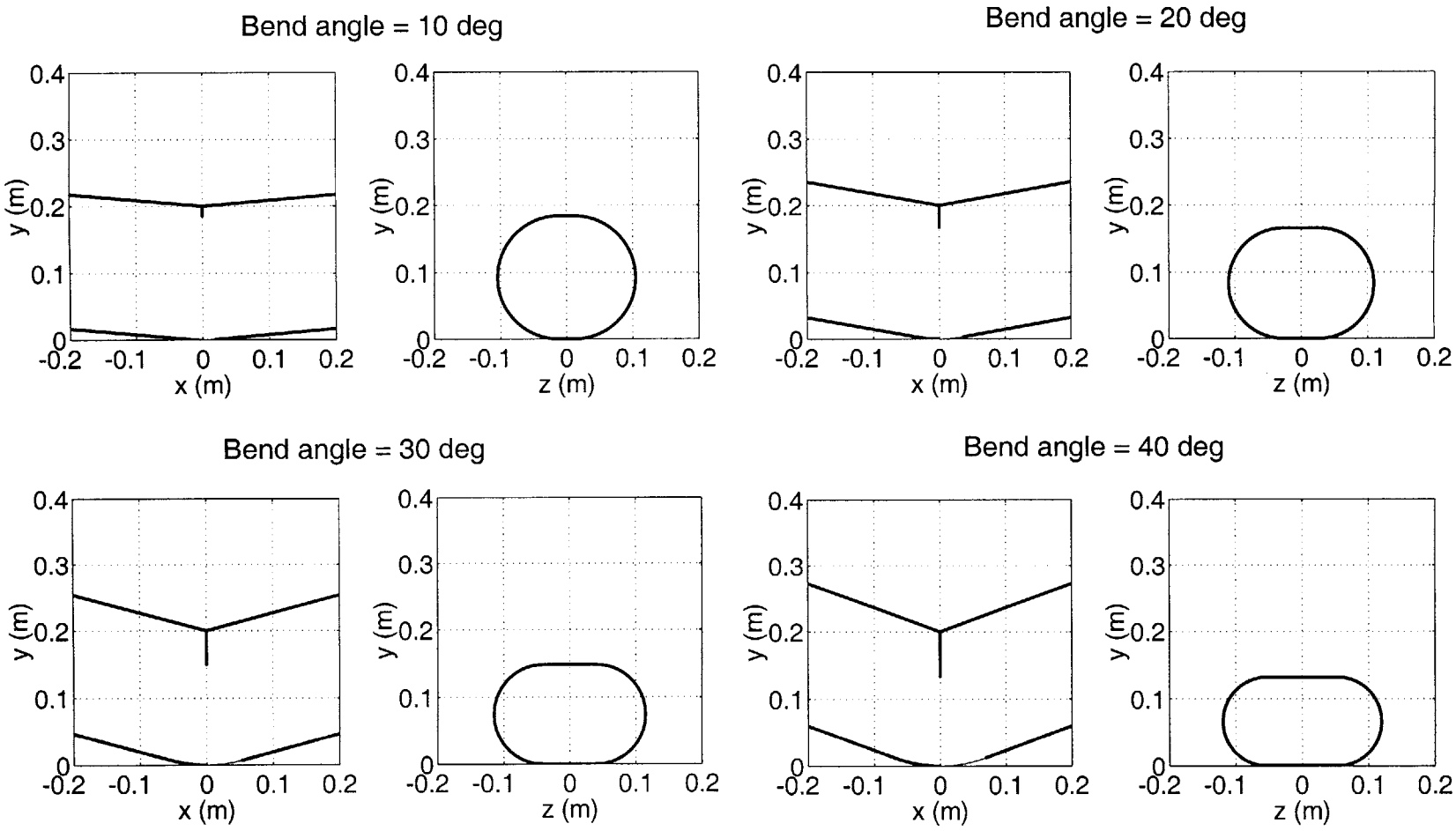


Figure 4-28. Membrane model predicts the shape of the bent pressurized tube

The volume of the tube, including both straight and curved parts, is calculated by numerical integration. The volume of the curved portion of the tube, V_c, is obtained by integrating

$$V_c = \int_0^{\phi} ARd\theta = \int_0^{\phi} \pi\left(R^2 - \left(R - \frac{H}{2}\right)^2\right)Rd\theta . \qquad \text{Eq 4-47}$$

Because the length of the curved part of the tube increases as the bend angle increases, the volume of the straight portion of the tube must be added. The volume of the straight part, V_s, decreases with increasing ϕ.

$$V_s = \pi R^2(L - 2R\phi) \qquad \text{Eq 4-48}$$

The tube volume is calculated for a range of ϕ values, then V is differentiated numerically with respect to ϕ, to obtain $\frac{dV}{d\phi}$. The calculated $\frac{dV}{d\phi}$ is then substituted into Equation 4-44 to obtain the bending moment M as a function of tube bending angle ϕ.

A dummy value for tube length, L, is used in Equation 4-48, but the value chosen for L does not affect the calculated bending moment, as shown by differentiating V with respect to ϕ.

$$M = -p\frac{dV}{d\phi} = -p\left(\frac{d}{d\phi}V_c + \frac{dV_s}{d\phi}\right) = -p\left(\frac{d}{d\phi}V_c - 2\pi R^3\right) \qquad \text{Eq 4-49}$$

4.3.3 Conceptual differences between beam and membrane models

A conceptual comparison between the beam and membrane models illustrates the space suit joint behavior that each of the models could be expected to capture. The two approximate models represent two limiting cases. The beam model assumes that all deflections are caused by extension of the fabric wall of the cylinder and that the gas inside the cylinder is never compressed. By contrast, the membrane model assumes that the fabric wall of the tube is inextensible and all of the work that is required to bend the tube goes into compressing the gas enclosed within it.

The beam model predicts only lateral deflections of the pressurized tube and does not consider changes in the shape of the tube's cross-section. Because it does not include the cross-sectional shape of the tube, the beam model cannot capture changes in the internal volume of the tube when it is bent. Since the beam model assumes that beam deflections are caused entirely by elastic behavior of the fabric, it does capture effects of the fabric's material properties, although it does not include any viscous or hysteretic effects. The valid range of load-deflection behavior predicted by the beam model is limited to small deflections for two reasons. Large deflections result in reductions in the volume enclosed by the cylinder that are not included in the model, and the work that would be required to compress the gas is not included in the moment-curvature relation. In addition, the beam

model's assumptions about wrinkle propagation around the circumference of the cylinder break down when the wrinkles extend around a large portion of the cylinder. Experimental results indicate that, when deflections are high, the small wrinkles combine to form larger wrinkles, transitioning to the membrane model's single crease, rather than extending further around the cylinder's circumference.[53]

The membrane model describes the three-dimensional shape of the bent, pressurized tube, including both transverse deflections and changes in the cross-sectional shape of the tube. Because it describes the cross-sectional shape of the tube, it can include changes in the internal volume of the tube. The membrane model assumes that the fabric wall of the tube is inextensible, so material properties are not included in the model. Like the beam model, the membrane model does not include any hysteretic behavior. Unlike the beam model, the membrane model has a realistic approximation of the bent tube geometry at large deflection angles.

The beam and membrane models encompass different aspects of pressurized tube bending behavior. The beam model captures the effects due to material properties of the fabric, while the membrane model is based entirely on geometry. The rigid membrane assumption considers the fabric modulus to be effectively infinite, so material selection does not affect the output of the membrane model. The membrane model has a more realistic assumption about the geometry of the bent tube for large deflections than the beam model does. Neither model can predict hysteresis effects. The beam model may be more useful than the membrane model for assessing the effects of space suit joint material selection and construction on mobility, because it includes material properties, while the membrane model can address geometrical issues such as the scaling of joint stiffness with radius, and the effects of contact between the space suit and the body or structure inside it.

4.3.4 Loading conditions

To compare the physics-based models to experimental data, it is necessary to determine the manner in which loads were applied and deflections were measured in the experiment. The experimental data was collected as described in Chapter 3, by installing the space suit on an instrumented robot, pressurizing the space suit, then performing prescribed arm and leg motions with the robot while measuring the deflection angles of the joints and the torques required to produce the deflections. In this experiment, the specified joint deflection was the independent variable and the torque required to achieve the specified deflection was the dependent variable.

The loading conditions on the space suit determine the deformed shape of the space suit arm or leg. The interface between the space suit arm and leg and the space suit torso is considered to be a clamp, with reaction forces and moments that fix the arm or leg to the torso. At the far end of the arm or leg, no reaction forces or moments are applied, so the hand or foot may be considered free. This cantilevered loading condition is depicted in Figure 4-29. A concentrated bending moment, M, is applied to the joint by the wearer of the suit, at the location of the suit wearer's joint axis, which is a distance a away from the clamp location. As a result, the space suit deforms only in the region between the clamp

and the location of the applied moment, and is straight elsewhere. The space suit joint bends by an angle ϕ, over a distance a, resulting in a curvature $K=\phi/a$, over the region $0<x<a$. The joint deflection angle measured in the experiment,ϕ, is given by

$$\phi = aK \qquad \text{Eq 4-50}$$

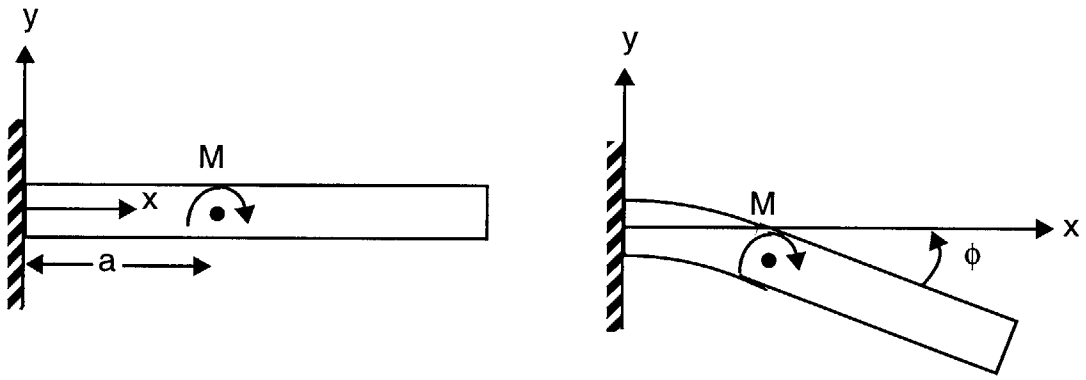


Figure 4-29. Loading conditions for space suit experiment. A.) Undeflected. B.) Deflected.

4.3.5 Parameter values

The beam model and membrane model each require parameter values for implementation. Both models use the space suit joint radius and internal pressure. In addition, the beam model uses the fabric modulus E, Poisson ratio ν, and bending region length a. The parameters common to all joints are listed in Table 4-2.

Table 4-2: Common parameters

Parameter	Value
Restraint layer fabric modulus	$1.4 \times 10^{10} N/m^2$
Restraint cord modulus	$1.73 \times 10^{11} N/m^2$
Poisson ratio	0.5
Pressure	30 kPa

Geometrical parameter values for the elbow flexion, hip flexion, hip abduction and knee flexion joints are given in Table 4-3.

Table 4-3: Joint geometry parameters

Joint	Space suit radius (cm)	Bending region length (cm)
Elbow flexion	6.95	20
Hip flexion	12.4	20
Hip abduction	12.4	20
Knee flexion	8.6	20

4.3.6 Results: Physics-based models

4.3.6.1 Beam model

The beam model was implemented in the Matlab script beammodel.m, in a three-step precess, to calculate the beam model's prediction of the bending moment, M, needed to produce the joint bending angle measured in the experiment, ϕ. The beam model is formulated with bending moment as the independent variable and curvature as the dependent variable. First, the circumferential extent of the wrinkles on the tube that result from an input bending moment M are calculated, using a look-up table method to obtain θ_0 from Equation 4-42. A range of θ_0 values are substituted into Equation 4-42, and the resulting M/pr^3 values are compared to the input M/pr^3. The θ_0 value that results in the M/pr^3 value closest to the input value is selected. Second, the curvature K is calculated from Equation 4-41. Finally, Equation 4-50 applies the loading conditions to obtain the bending angle ϕ corresponding to the input M.

Predictions from the beam model are compared first to the elbow and knee joints. The elbow and knee joints are designed with mobility features, such as pleats on the outer surface of the joint that open when the joint is bent and axial restraint cords that carry pressurization loads close to the joint axis. Because these joints differ significantly from the uniform cylinder that underlies the beam model, it is unsurprising that the beam model does not predict the elbow and knee torque-angle curves. Figure 4-30 shows a comparison between torque-angle data for the knee and elbow and the torque-angle predictions generated by the beam model, in order to illustrate the qualitative features of the data and model predictions. The beam model output shown in Figure 4-30 uses an unrealistically low E and is scaled to roughly correspond to the magnitude of the torque data.

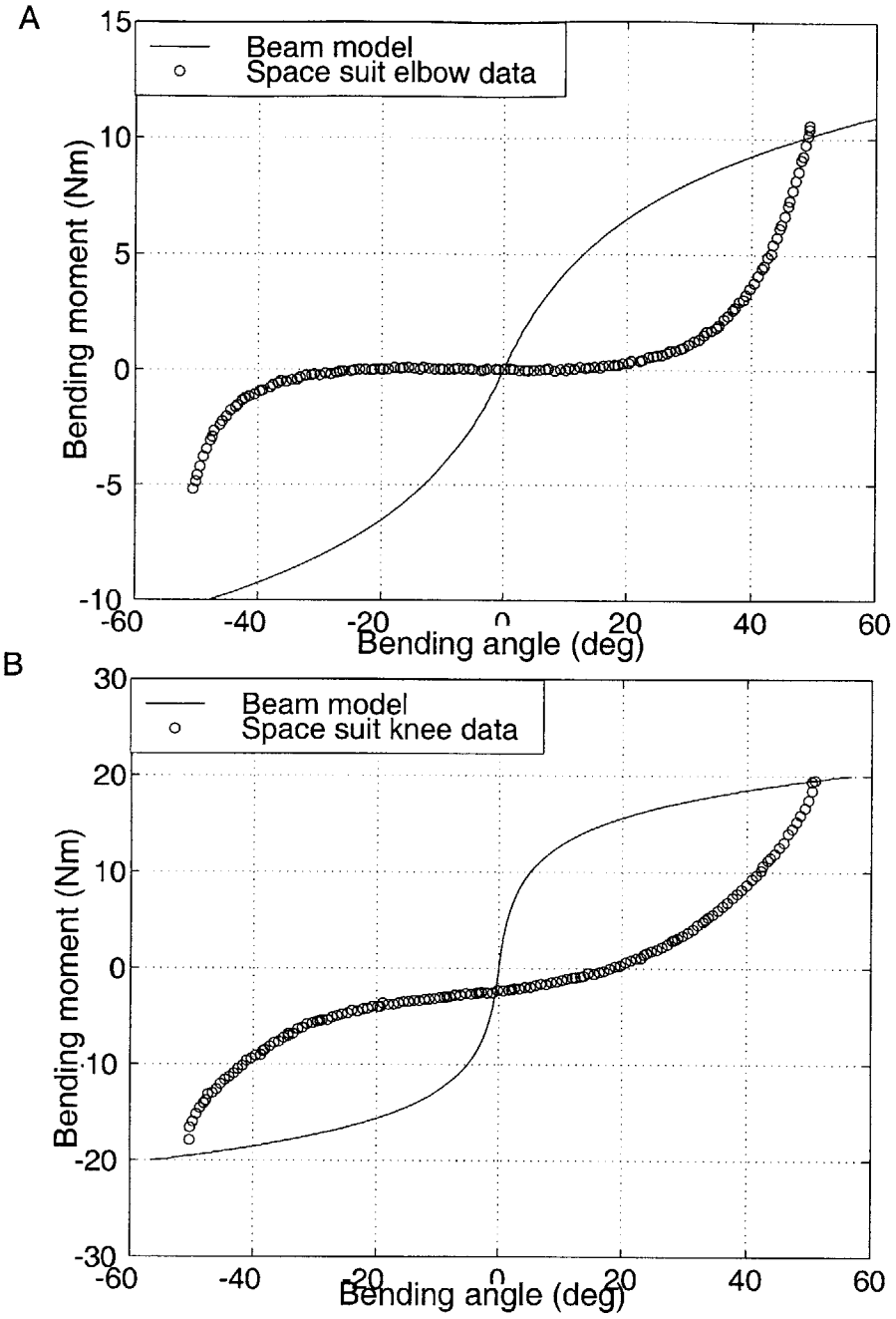


Figure 4-30. Comparison of beam model curve shape to experimental data. A.) Elbow. B.) Knee

Figure 4-30 shows that the shape of the torque-angle curves predicted by the beam model differs dramatically from the torque-angle data for both the knee and the elbow. The beam model predicts softening, or decreasing stiffness with increasing deflection. As the applied bending moment increases, the extent of the wrinkled region of the beam wall increases, and the beam's stiffness decreases. In contrast, the experimental data indicates hardening, an increase in stiffness with increasing deflection. Other investigators' data, shown in

Figure 2-4 and Figure 2-5, corroborates the increase in stiffness with deflection, which is contrary to the beam model's predictions.

The magnitudes of the torques predicted by the beam model, using realistic dimensions and modulus values, are compared in Figure 4-31. The torque magnitudes that the beam model predicts are higher than the actual torques measured, by approximately a factor of 5.

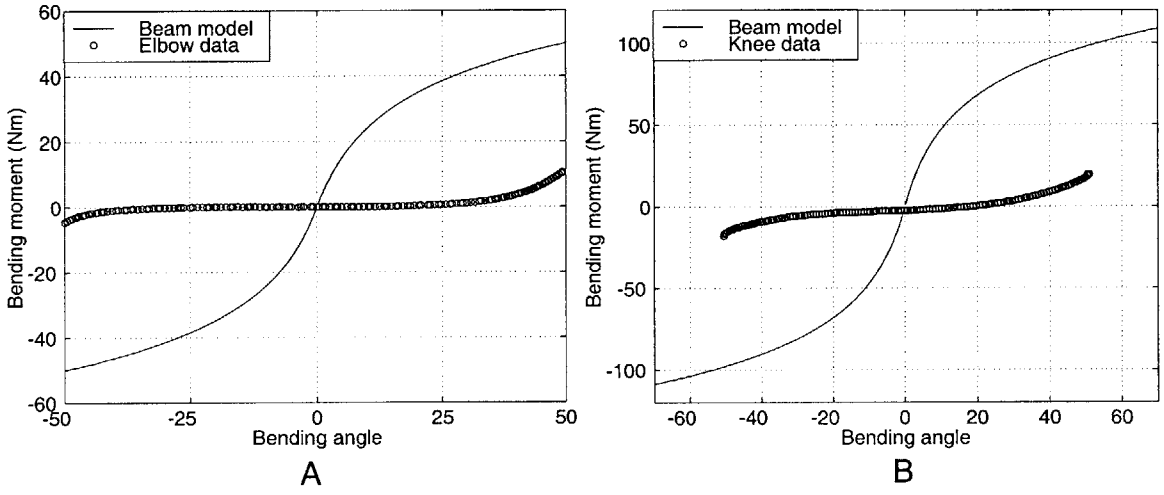


Figure 4-31. Comparison of beam model torque magnitude to experimental data. A.) Elbow. B.) Knee

Although Figure 4-30 and Figure 4-31 show that the beam model does not describe the torque-angle characteristics of the elbow and knee joints, it may, however, describe the behavior of the hip abduction joint. The hip joint on the space suit is designed for maximum mobility in the flexion, or front-back direction by placing the pleats on the front and back and the restraint cords on the left and right sides, in an arrangement similar to that shown in Figure 2-3. Maximizing mobility in the flexion direction means that abduction, or left-right bending, cannot take advantage of the mobility features, and is consequently very stiff. Bending the space suit hip joint in abduction may require stretching the restraint cords, so there is reason to expect that the beam model may describe the hip abduction joint's behavior.

Figure 4-32 shows a comparison between hip abduction torque-angle data and the output of the beam model, indicating that the beam model can describe the hip joint's behavior. Limits on torques that can be safely applied to the space suit restrict the angle range in the data to 0-10 degrees, so the softening behavior that is the key qualitative characteristic of the beam model is not seen over the angle range in the data.

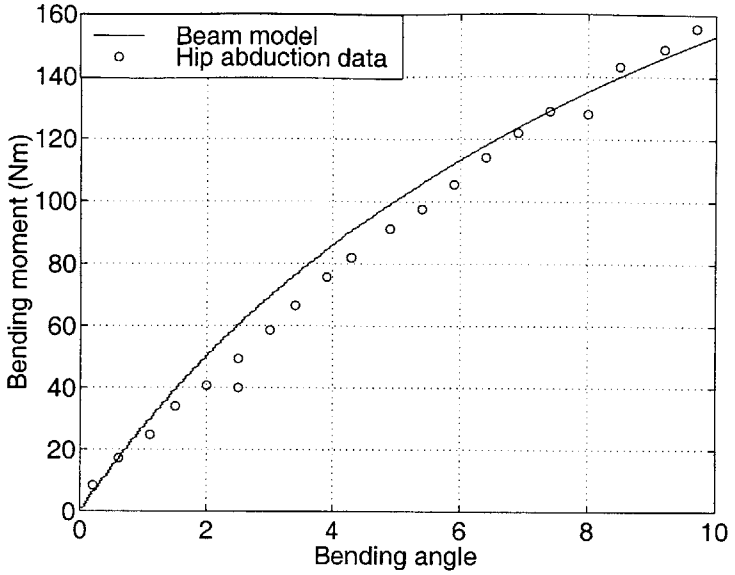


Figure 4-32. Beam model compared to hip abduction data.

Despite Main, Peterson, and Strauss's[35, 36, 37]claims that the beam model describes the load-deflection behavior of space suit joints, a comparison between the beam model's predictions and experimental data for the elbow and knee reveals that the beam model does not reproduce either the magnitude of torques or the shape of the torque-angle curve for the elbow and knee joints. The experimental data indicates that space suit elbow and knee joints are much less stiff than the beam model predicts and the stiffness increases with deflection, rather than decreasing, as the beam model predicts. The empirical torque-angle curve for the hip abduction joint is consistent with the beam model, however, the angular range that is attainable in operation is too small to display the important qualitative features of the beam model's torque-angle curves.

4.3.6.2 Membrane model

The membrane model was implemented as described in Section 4.3.2. The general shape of a torque-angle curve from the membrane model is shown in Figure 4-33. Like the experimental data collected from the space suit, but in contrast to the beam model, the torque-angle slope is low near the zero torque angle, then increases as the bend angle increases.

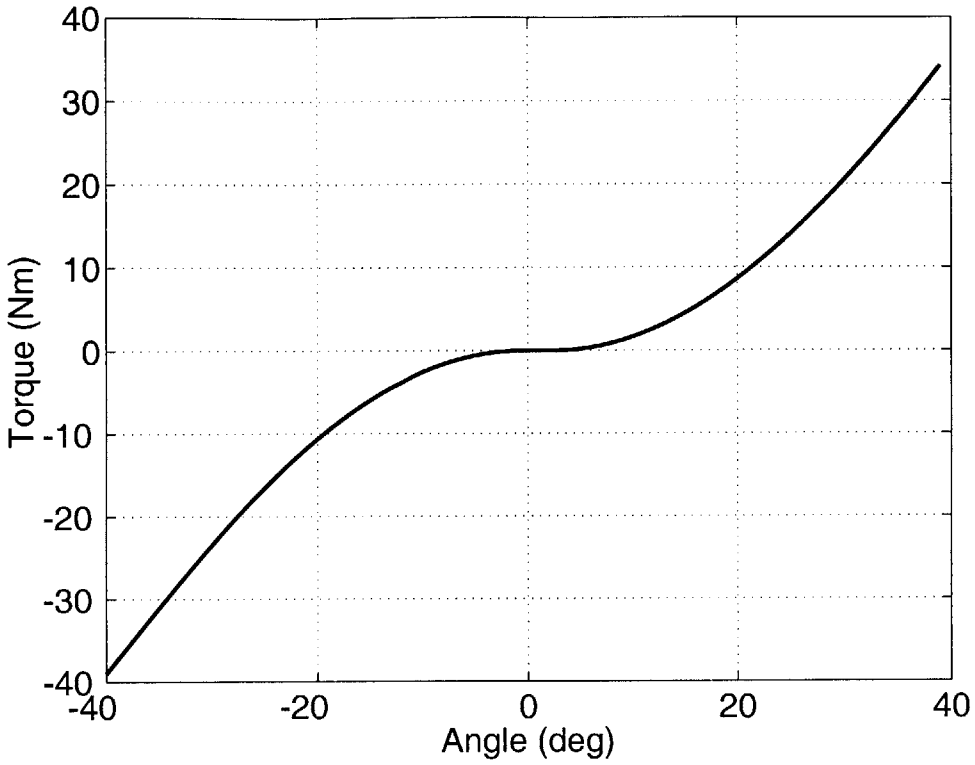


Figure 4-33. General shape of membrane model output.

A nonlinear least squares algorithm was used to find the equilibrium angle values that best fit the elbow and knee torque-angle data. The equilibrium angle was fit to the first of four data files used in the hysteresis model fitting. To obtain an overall curve shape without hysteresis effects, the two torque values that correspond to the same angle for the largest-amplitude ascending and descending curves were averaged. The output of the membrane model is then compared to the remaining three data files. The membrane model output, using an equilibrium angle of 50 degrees, is compared to elbow data in Figure 4-34, showing a close correspondence between the membrane model and experimental data for elbow flexion angles from 20 degrees to 100 degrees. Outside of the 20-100 degree range, the torque data is higher than the model prediction. Figure 4-35 shows a similar result for the knee, using an equilibrium angle of 48 degrees. The model matches the knee torque data for knee flexion angles of 20-80 degrees and, similar to the elbow joint, the knee torque data is higher than the model predictions for angles outside of the 20-80 degree range.

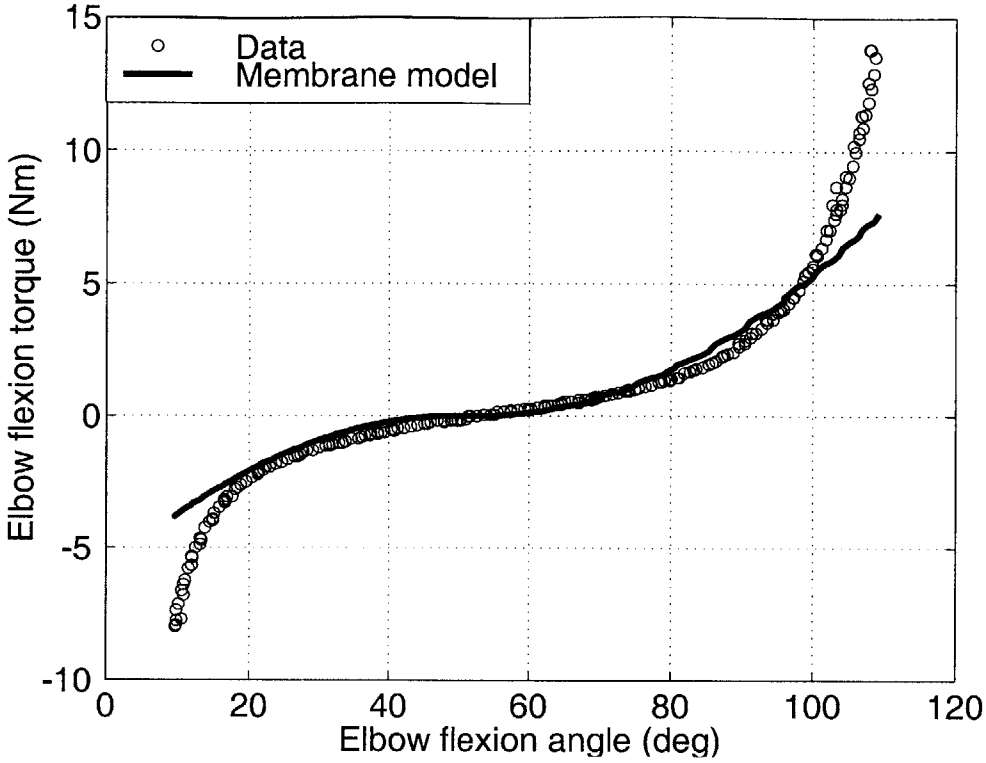


Figure 4-34. Comparison between membrane model and elbow data

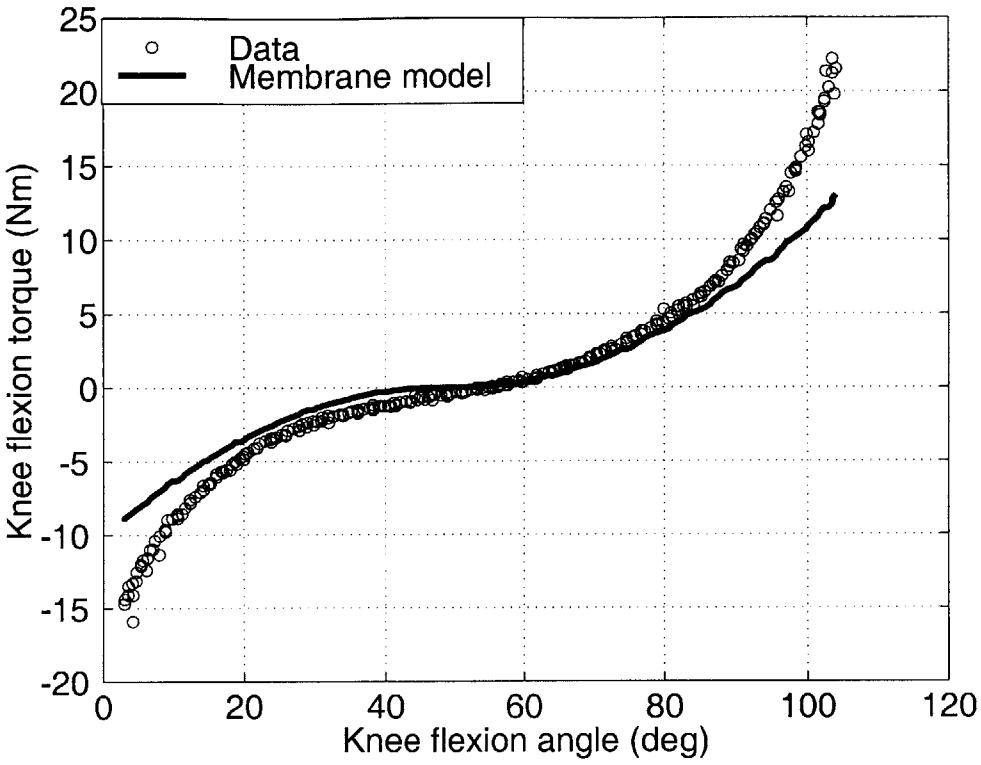


Figure 4-35. Comparison between membrane model and knee data.

4.3.7 Discussion: Physics-based models

4.3.7.1 Agreement between models and data

The beam model and membrane model represent opposite extremes of the processes governing space suit joint mobility. The beam model assumes that the space suit joint torque-angle relationship is determined solely by elastic deformations of the fabric wall of the space suit, with no changes in the internal volume of the space suit segment. The membrane model makes the opposite assumptions: that the space suit fabric does not stretch, although the cross-sectional shape deforms, resulting in reduction in internal volume as the joint is bent. According to the beam model, bending moments only stretch the space suit fabric; according to the membrane model, bending moments only compress the gas inside the space suit.

It is reasonable to expect that the actual behavior of a space suit falls between these two extremes, that both elasticity and volume changes determine space suit joint mobility. Comparing these two approximate models to experimental data illustrates where on the continuum between elasticity and volume change the space suit joint behavior falls and indicates whether the space suit behavior can be approximated by one model or the other. Understanding the physical processes that govern space suit joint performance leads to insights into ways of designing improved-mobility joints. If elastic behavior is dominant,

details of material selection and construction of the joints are drive mobility, while if volume changes are dominant, geometrical factors in the joint design are more important.

As Figure 4-30 and Figure 4-31 show, the beam model is inconsistent with experimental data for the elbow and knee joints, both in the bending moment magnitudes predicted and the trend in bending moment with increasing deflection. The beam model predicts bending moments for the elbow and knee joints that exceed the experimental data by at least a factor of 5. In addition, the beam model predicts a decreasing slope of the moment-angle curve as bending angle increases, while the data shows a increasing slope with increasing bending angle. The beam model is consistent with experimental data for the hip abduction joint, which does not utilize the mobility features of the elbow and knee joints, although the limited angle range for which data is available prevents the beam model from showing its characteristic softening behavior.

The shape of the torque-angle curves generated by the membrane model is consistent with the experimental data, because the model's torque-angle curves are flat near the equilibrium position of the joint and increase in slope as deflection increases. The membrane model follows the experimental data over a bending angle range approximately 30-50 degrees from the equilibrium angle. Outside of this range, the amplitude of the experimental torque data is greater than the membrane model's predictions. Although the membrane model can predict the torque needed to bend the joints over about half of the available range of motion, the model's underprediction of torques near the extremes of the angle range prevent it from being used to predict a joint's angular range of motion. Because the membrane model underpredicts torques near the ends of the range, relaxing the fabric inextensibility constraint and allowing a combination of fabric stretching and gas compression would not make the model agree more closely with data, because it would decrease the model's prediction of the torque needed to bend the joint. Other physical processes, including fabric bunching, sideways compression of the space suit arm or leg, and friction between fabric layers, which were not included in the membrane model, likely play an important role in determining the torque needed to bend the space suit joints at high bending angles.

Experimental data indicates that the membrane model, is a good approximation to the elbow and knee joint torque-angle performance, near the joint's equilibrium angle, but underpredicts torque at the ends of the range of motion. It follows from this conclusion that elastic deformations of the space suit fabric do not contribute significantly to the torque needed to bend the elbow and knee joints near the equilibrium angle.

4.3.7.2 Implications for space suit design

Because the torque-angle performance of the space suit joints within 30-50 degrees of their equilibrium angles can be accounted for by changes in internal volume without stretching of the space suit fabric, the mobility of the EMU elbow and knee joints near their equilibrium angles is not sensitive to changes in the material properties of the fabric. The membrane model predicts that internal pressurization and joint geometry determine mobility.

According to the membrane model, the torque needed to bend the space suit joints is directly proportional to internal pressure. A space suit that has an operating pressure of 60 kPa, or twice the operating pressure of the EMU, would be useful because astronauts would not be required to spend several hours breathing pure oxygen before EVAs to avoid decompression sickness. However, if the higher-pressure space suit used the same joint designs as the EMU, the torque required to bend the joints would increase by at least a factor of two. Modifications to the joint designs would be necessary for a high pressure space suit to be practical.

Useful mobility of the EMU joints could be improved in three ways. Because the joint stiffness increases with radius, the radius of the space suit segment should be the minimum possible. The length of the pleats in the joint should allow for the most change in length along the outside surface of the joint without compromising the shape of the joint. Finally, the zero-torque positions of the joints, including the position and orientation of the arm openings on the HUT should be set near the midpoint of the desired range and so that the relaxed overall body position of an astronaut in a space suit in microgravity corresponds to a convenient working position.

Agreement between experimental data and the membrane model implies that elastic deformations of the space suit fabric do not determine the torque-angle performance of space suit joints. Consequently, the design parameters that affect space suit mobility are the internal pressure, radius of the space suit segment, and the pleat design that allows the joint to be bent without reducing the internal volume.

4.3.7.3 Future research

Interesting future research on physics-based modeling of space suit mobility includes both theoretical and experimental work. The modeling work in this thesis is based on two theoretical models in the literature on transverse bending of pressurized cylinders. Additional theoretical work could build on these uniform cylinder models, resulting in theoretical models that are more representative of space suit joints. Areas for additional theoretical work include a more sophisticated representation of the geometry of space suit joints with non-zero equilibrium bending angles, inclusion of sideways compression of the space suit's arm or leg for large bending angles, fabric bunching effects, and effects of friction between the layers of the space suit.

Experimental work on bending joints could evaluate the key assumptions of the two models, such as whether the space suit fabric stretches, assess specific model predictions, and determine the effects of unmodeled physical processes on space suit mobility. Tests conducted on isolated joint mockups may answer the following questions.

- Does the space suit fabric stretch when the joint is bent?
- How does the internal volume of the joint vary with bending angle?
- Is the joint stiffness directly proportional to internal pressure?
- How does the presence of the body inside the space suit affect the joint's mobility?

Additional testing with human subjects wearing the space suit would address the relevance of model predictions to practical situations. Measuring contact forces on the inside surface of the bending joint and measuring the cross-sectional deformations of the joint as it is bent would illustrate the effects of contact between the body and the space suit.

The physics-based modeling work in this thesis leads to specific questions that may be answered by experiments with isolated space suit segment mockups and human subjects wearing space suits. Further experimental work would clarify the relative roles of elasticity and volume loss in space suit mobility, leading to insights into how to improve joint mobility.

4.4 Conclusions: Modeling

4.4.1 Mathematical model

The Preisach hysteresis model is a useful tool for predicting the torque needed to bend space suit joints as a function of time for complicated angle trajectories. The ability to reproduce realistic space suit torques dynamically makes the Preisach model well-suited for use in dynamic simulation of EVA tasks. An exploration of the properties of the Preisach model shows that the model will fit space suit torque-angle curves, even though angle is the independent variable and torque is the dependent variable, which is an unconventional arrangement. A new method for estimating the variance of the error in the Preisach model's torque prediction was developed and applied.

The Preisach hysteresis model was fitted to torque-angle data from six space suit joints, for data in which the input joint angle was varied between a series of minima and maxima. The model outputs were then compared to data for human-generated motions. The Preisach model matched torque data well for four joints: elbow flexion, hip abduction, knee flexion, and ankle rotation. The ankle flexion and hip flexion angles in the human-generated data were outside the valid range for the fitted model coefficients, so the model did not follow the human-generated data for those two joints. The error estimates generated by the Preisach model are valid, but possibly conservative.

Modeling the hysteretic characteristics of space suit joints is essential for making accurate predictions of the torques required to bend the space suit joints in dynamic situations. This work demonstrates that the Preisach hysteresis model accurately reproduces torques needed to bend space suit joints.

4.4.2 Physics-based models

Two approximate physics-based models of pressurized tube bending, the beam model and the membrane model, were compared to experimental torque-angle data from space suit joints. The beam model assumes that, when the joint bends, the space suit fabric deforms elastically, with no loss of internal volume. Conversely, the membrane model assumes that the space suit fabric is inextensional, and all of the work required to bend the joint goes

into compressing the gas inside the space suit. Comparing these two limiting-case models to experimental data illustrates the relative importance of elasticity and volume changes on space suit mobility.

The beam model was found to be fundamentally inconsistent with the performance of the elbow and knee joints, predicting torques much higher than the experimental data, with a torque-angle curve markedly different from the data. Torque-angle data from the hip abduction joint, which is much stiffer than any other space suit joint, is consistent with the predictions of the beam model, although the limited bending angle range of the data prevents a comparison of the model and data torque-angle curve shape.

The shape of the torque-angle curves output from the membrane model are similar to the shape of the experimental data curves, with a flat region near the zero-torque angle and increasing slope with increasing deflection. Comparisons to experimental torque-angle data from the elbow and knee joints indicate a good correspondence between the membrane model and data, for bending angles within 30-50 degrees of the equilibrium angle, which covers approximately half of the available joint angle range. For more extreme bending angles, the membrane model underpredicts torques needed to bend the space suit joints. Thus, the membrane model is a good representation of space suit mobility for non-extreme bending angles.

Close correspondence between the membrane model and experimental data indicates that elasticity is not a major contributor to space suit mobility. As a result, it is not likely that modifications to the material properties or elasticity-related design parameters of the space suit will improve mobility. Instead, geometrical properties play a larger role in mobility. Improvements in useful mobility could be accomplished by sizing the radius of suit segments to be as small as possible, modifying the pleat design to maximize the constant-volume joint bending range, and locating the equilibrium bending angles of the joints to result in a convenient neutral body position.

Chapter 5 *The work envelope: Applying space suit modeling to EVA operations*

5.1 Introduction

A critical issue that is addressed in planning for extravehicular activities (EVA's) and evaluating EVA worksites is whether the EVA crew can reach and comfortably work in the designated worksite. Reach considerations are important because, in microgravity, astronauts must restrain their bodies in order to exert forces or moments on other objects. Body restraint during EVA is usually accomplished by foot restraints, particularly when tasks require applying significant forces. The articulated portable foot restraint (APFR) used on the space station can be adjusted to point in many orientations but can only be attached to the space station's structure at discrete locations that were set when the space station was designed. Because of the constraints on foot restraint placement, it is essential to determine in the planning process whether a work site is reachable from an available foot restraint location. Although foot restraints are not the only available method of providing body restraint, the necessity of restraining the body for microgravity work makes the problem of providing both adequate restraint and access to the worksite a universal one in microgravity. A work envelope analysis that incorporates the mechanics of the space suit is a useful method for both assessing potential worksite locations and evaluating the functional significance of modifications to space suit mobility and visibility.

A reach envelope is the region in three-dimensional space that a person can reach. The work envelope is a subset of the reach envelope, representing the volume in which a person can comfortably work. Reach and work envelopes depend on the size and flexibility of the individual. A standard practice is to size workspaces to accommodate the reach and work envelopes of individuals at the extremes of the expected size range, for example, 5th percentile females and 95th percentile males.[46] Most work envelopes are determined experimentally, by measuring how far people of different sizes can reach and obtaining subjective information about the difficulty of working with the hands in different locations. The results of empirical work envelope studies are typically a set of boundaries within which a specified percentage of the intended population can work comfortably. An alternative approach to reach envelope analysis has been developed recently, which uses robot kinematic analysis methods to determine the boundaries within which a person can reach with prescribed limits on joint ranges of motion.[28, 29, 30, 31, 50] Work envelopes that are determined computationally can be tailored to specific people and can also indicate

regions within the work envelope that are easier to reach and should be preferred in worksite placement.

A new work envelope analysis method, which combines inverse kinematics and the space suit model, enables the work envelope of a space-suited person to be assessed in greater detail and with more generality than the current, experimental, methods allow. The computational work envelope analysis developed in this thesis extends the inverse kinematics reach envelope work by prior investigators[28, 29, 30, 31, 50] to determine feasible working positions, a more restrictive category than reachable positions, incorporating human strength limits and visibility considerations. For any proposed hand position, the inverse kinematics analysis determines the arm configurations that place the hand on target. Based on the arm joint angles, the joint torques required to hold a specific arm position are determined by the space suit model. Knowledge of joint torques as a function of hand position allows a detailed analysis of worksite placement: areas requiring excessive torques at any joint can be eliminated from the feasible work space, while areas that are particularly easy to reach can be preferred. Furthermore, the inverse kinematics method allows the work envelope analysis to be customized for individuals of different sizes.

Work envelope analysis represents one area in which fundamental models of space suit mechanics are useful in predicting and understanding a large-scale EVA human factors performance metric.

5.2 Work envelope criteria

Three criteria determine whether a proposed hand position is in the work envelope: torques needed to hold the required arm configuration, visibility, and the shape of the resulting work envelope boundaries. The shape of the work envelope is considered because highly convoluted work envelope boundaries are difficult to use, since they are excessively sensitive to the relative positions of the EVA astronaut and the worksite. However, adding the shape criterion overconstrains the problem; as a result, a single torque limit cannot be enforced uniformly. To deal with this issue, a smoothing algorithm is used to systematically trade off the joint torque and work envelope shape criteria.

The volume surrounding the astronaut is evaluated point-by-point to determine whether each point in a three-dimensional mesh is inside or outside of the work envelope. Points are placed in three categories, based on their visibility and the joint torques required to place the hand on that point. Points that are not visible, or require excessive torques, as defined in Section 5.2.1, go into the "always exclude" category. Points that are visible and require only low joint torques, defined in Section 5.2.1, go into the "always include" category. The remaining points, which are visible and require torques higher than the low limit, but lower than the excessive limit, go into the "possibly include" category. The smoothing algorithm then draws the work envelope boundaries to include all of the "always include" points, none of the "always exclude" points, and then draws a limited number of points from the "possibly include" category to smooth the shape of the work envelope.

5.2.1 Joint torques

In order for a person to work for several minutes in a particular arm configuration, the torques needed at all of the arm joints should be low enough that fatigue does not prevent the person from maintaining the desired arm position. Many strength measures are found in the anthropometry and biomechanics literature. The strength measure of interest in a work envelope analysis is the maximum torque that can be sustained for several minutes in a static position. In a survey of research on muscle fatigue, Astrand[6] reports that the maximum force in static, isometric (constant position) muscle contractions can be held for only a short period of time, but 15% of the maximum force can be held for at least 10 minutes, as shown in Figure 5-1. According to the data reported by Astrand[6], 30% of the maximum torque can be held for approximately 3 minutes, 50% can be held for 1.5 minutes, and 100% can be held for less than 1 minute.

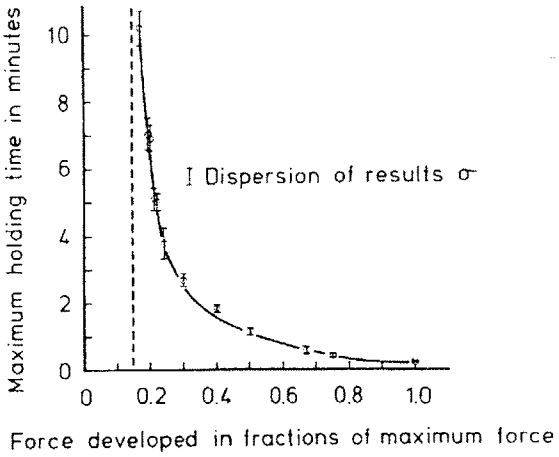


Figure 5-1. Maximum holding time vs. isometric maximum force[6]

Reports on maximum isometric strength for isolated joints are somewhat rare in the human factors and biomechanics literature. Most of the published static strength data is related to specific multi-joint motions, such as lifting or pushing objects. In one exception, Chaffin[11] reported isometric strength data for single joints for a population of industrial workers. Chaffin's strength data for shoulder flexion, shoulder abduction, humerus rotation, and elbow flexion motions in both positive and negative directions (defined in

Figure 3-3 and Figure 5-3) are shown in Table 5-1. Torque limits used in the work envelope analysis are based on the limits in Table 5-1.

Table 5-1: Isometric strength limits for shoulder and elbow.[11]

Motion	50th percentile male (Nm)	95th percentile male (Nm)	50th percentile female (Nm)	95th percentile female (Nm)
Shoulder flexion (+)	71	101	37	57
Shoulder flexion (-)	67	115	30	54
Shoulder abduction (+)	67	103	33	57
Shoulder abduction (-)	92	119	40	60
Humerus rotation (+)	33	51	19	28
- Humerus rotation (-)	52	83	21	33
Elbow flexion (+)	77	111	41	55
Elbow flexion (-)	46	67	27	39

Based on the relationship between maximum holding time and percentage of maximum isometric torque, the low torque and excessive torque limits are set. An arm position for which no joint torque exceeds 15% of the maximum torque is considered "low torque". An arm position which requires more than 30% of the maximum torque from any joint is considered "excessive torque".

5.2.2 Visibility

The space suit helmet and the display and control module (DCM), which is located on the front of the space suit, limit a space-suited person's field of view. A suited person's field of view is limited in the upward and left-right directions by the space suit's helmet and visor and in the downward direction by the DCM and the mini-workstation, a tool rack located on the front of the hard upper torso (HUT). (See Section 2.2 for a description of the EMU space suit) Only points that are within the field of view are included within the work envelope.

Field of view limits for the EMU space suit are published in NSTS-07700[1]. Additionally, Graziosi, Stein and Kearney (GSK)[24], who conducted a study evaluating possible modifications to the EMU helmet to improve visibility, reported data on upward and downward field of view limits. Both data sets are shown in Figure 5-2. Graziosi, Stein and Kearney[24] concluded that modifications to the helmet alone would not improve downward visibility because of the presence of the DCM. The NASA field of view limits were used in the work envelope analysis. Although the mini-workstation obstructs downward visibility, the NSTS-07700 visibility limits do not include the presence of the mini-workstation. Conse-

quently, the analysis presented herein does not consider the downward visibility limitations imposed by the mini-workstation.

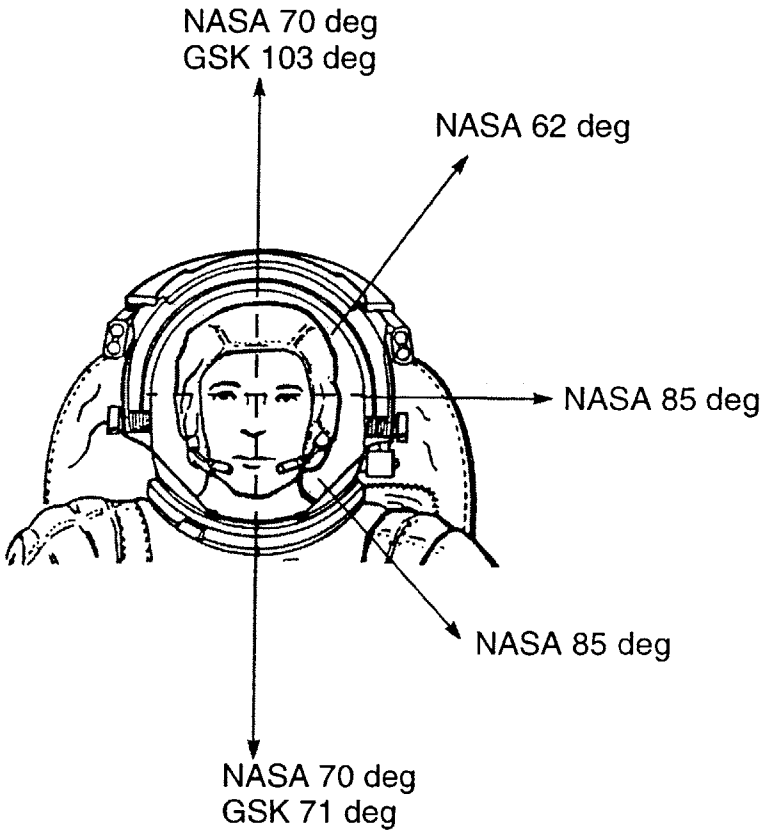


Figure 5-2. EMU field of view[1, 24]

5.2.3 Work envelope shape

Using an arbitrary torque limit level to determine whether points are inside the workspace results in a work envelope with a complex and convoluted shape. A work envelope with such a complicated shape is difficult to implement in planning EVAs or designing EVA workstations. Small changes in the relative positions of the EVA astronaut and worksite may shift the worksite into or out of the work envelope. Excessive sensitivity to positioning is unwarranted because the appropriate maximum torque levels depend on the strength and fatigue level of the astronaut, the duration of the task, and the accuracy of arm motions required in the task. Furthermore, a complicated work envelope boundary requires storing a large number of boundary vertices for computational implementation, and is difficult to visualize. Therefore, the boundaries of the work envelope should be as simple as possible, while including the maximum number of low-torque points and excluding all excessive-torque points.

The shape of the work envelope is set by categorizing possible points, based on visibility and required torques, as "always include", "always exclude", and "possibly include".

Points that are visible and require torques that do not exceed a specified low torque limit are in the "always include" category, points that are not visible or require any joint torque to exceed a specified excessive torque limit are in the "always exclude" category, and points that are visible and require torques that fall between the low torque and excessive torque limits are in the "possibly include" category. The initial boundary is drawn to include only the "always include" points. Then the boundary is smoothed by adding a limited number of points that are in the "possibly include" category. Work envelope boundary smoothness is assessed by determining how many vertices are needed to draw a piecewise-linear boundary around the work envelope region. Minimizing the number of vertices needed to draw the boundary results in the smoothest boundary.

5.2.4 Arm model

The arm model chosen for the inverse kinematics calculations is a simplified approximation of the complex mechanics of the human arm. The number and type of articulations of the model arm were chosen to coincide with the joints included in the space suit model and to limit the kinematic redundancy of the model arm. The model arm includes four joints: shoulder flexion (sf), shoulder abduction (sa), humerus rotation (hr), and elbow flexion (ef). The joint angles and sign conventions are shown in Figure 5-3.

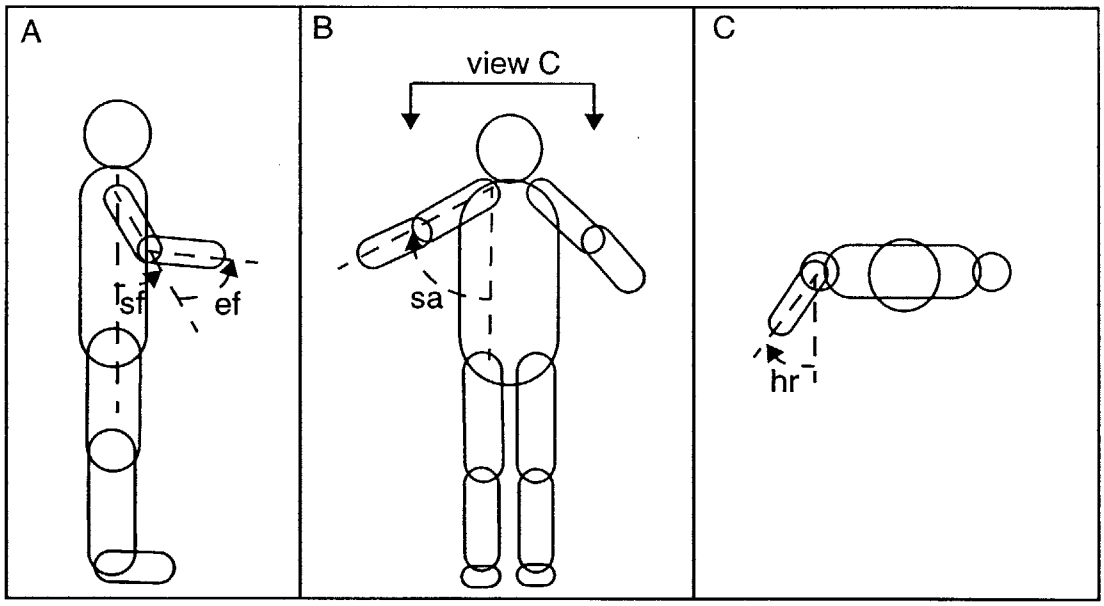


Figure 5-3. Joints included in arm model. A.) Shoulder flexion (sf) and elbow flexion (ef). B.) Shoulder abduction (sa). C.) Overhead view with elbow flexion=90 deg, showing humerus rotation (hr)

The wrist joint was not included in the arm model for two reasons: lack of a model of the space suit wrist joint and kinematic redundancy. Because wrist torque-angle data was not collected, the space suit wrist joint was not modeled. The number of joints that are included in the arm model affects the computational efficiency and reliability of the

results. Three coordinates of hand position are used to determine the angles of all of the arm joints. Since the number of arm joints is at least 4, the system is underdetermined, or kinematically redundant. There are multiple arm configurations that place the hand at the same location, and no universally accepted method for predicting which of these configurations a person will choose. Previous investigators have used both heuristic[28, 29] and optimization approaches[7] to resolve kinematic redundancy in the arm, generating results that appear realistic. However, the true basis for a choice of arm position is still an open research topic. The uncertainty associated with resolving kinematic redundancy in an arbitrary way motivates a choice of arm model of limited redundancy, which has the minimum number of joints for realism.

An important drawback associated with using an arm model without a wrist joint is that hand orientation cannot be included in the model. Therefore, even though the hand could be located at any point included in the work envelope, the arm model does not guarantee that the hand could be oriented in every direction specified.

The arm model assumes that the joint rotation axes are fixed in relation to the appropriate arm segments. Physically, this means that the joints are assumed to pivot about a fixed axle or spindle, in pure rotational motion, similar to the joints on a robot. The kinematics of human joints is somewhat different, with the ends of the bones both pivoting and sliding over each other, resulting in movement of the joint rotation axes that cause both rotation and small amounts of translation. The difference between the arm model and real human joint motions leads to small errors in the inverse kinematics analysis. The real hand position would be several centimeters away from the hand position calculated from a set of arm joint angles. Kinematic errors due to the constant rotation axis assumption have negligible effects on the results of the work envelope analysis because the kinematic errors are small (1-2 cm) compared to the 6.24 cm mesh spacing in the inverse kinematics analysis and the torques output from the space suit model are smooth functions of joint angle, with no large changes for small angle differences.

The lengths of the arm segments are set based on anthropometric data[2], which is listed in Table 5-2. The lower arm segment length is the elbow to wrist distance plus half the hand length. Arm segment lengths can also be customized to a specific person.

Table 5-2: Arm segment lengths[2]

Segment	Male, 50th percentile (cm)	Male, 95th percentile (cm)	Female, 50th percentile (cm)	Female, 95th percentile (cm)
Upper arm	36.6	39.4	29.8	32.4
Lower arm	38.3	41.6	33.1	35.3

5.3 Work envelope methods

The visibility and joint torque limitations are applied in a five-step process depicted in Figure 5-4 and implemented in the Matlab script arminvkinbatch.m, in Appendix C. First,

visibility is evaluated: if the target point is outside of the field of view, it is excluded from the work envelope. Second, inverse kinematics is used to determine the joint angles at the shoulder and elbow that are needed to place the hand on target. The inverse kinematics solution is not unique, because three coordinates of hand position determine four joint angles. The inverse kinematics analysis outputs a set of several arm configurations that put the hand on target. The space suit torque-angle model, implemented in the Matlab function suittorques.m, then accepts these joint angles as inputs and calculates the joint torques required to maintain the specified arm configuration. The joint torques are compared to the pre-set torque limits and the arm configurations for which no joint torques violate the torque limits are returned as feasible arm positions. If at least one feasible arm position exists for a target hand position, the target point is considered to be within the work envelope. Finally, the overall difficulty of reaching the target point is assessed by calculating a reach difficulty metric, based on the required joint torques. The feasible arm configuration that has the lowest reach difficulty metric is recorded for each target position in the work envelope.

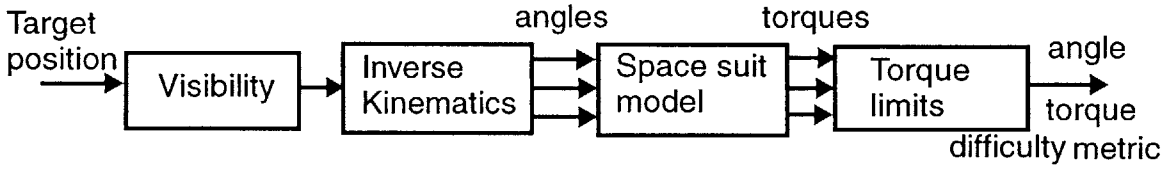


Figure 5-4. Work envelope analysis technique block diagram

5.3.1 Visibility

Visibility of the target point is evaluated first, because the visibility constraint eliminates a large portion of the possible target points. A suited person's field of view is limited in the upward and lateral directions by the space suit's helmet and visor and in the downward direction by the DCM and the mini-workstation, which are both located on the front of the HUT. This analysis used the visibility limits for the EMU space suit published in NSTS-07700[1], which are shown in Figure 5-2. The NSTS-07700 visibility data was empirically determined, and, while details of the experimental methods were not supplied, it is most likely that test subjects were allowed to move their heads within the space suit helmet and only monocular vision was measured.

Visibility is evaluated by first locating the suited person's eyepoint relative to the right shoulder. A single point that represents the eyepoint position is located on the body centerline, halfway between the eyes. The eyepoint coordinates that were used for 50th and 95th

percentile males and females are listed in Table 5-3.[2] No anthropometric data was available for the position of the eyes forward of the shoulder, so 10 cm was used in all cases.

Table 5-3: Eyepoint position relative to the right shoulder[2]

Percentile	M 50th	M 95th	F 50th	F 95th
Vertical (cm)	16.5	16.9	15.9	16.5
Left-right (cm)	20.6	22.2	18.7	20.3

The field of view limits are used to set the vertices of an 8-sided polygon centered on the horizontal and vertical eyepoint position. The forward distance between the target hand position and the eyepoint position sizes the polygon, so that the vertices of the polygon represent the angular field of view limits. If the target position falls within this polygon, it is within the field of view; if the target position is outside the polygon, it is outside of the field of view and consequently out of the work envelope. An example of the field of view boundaries 30 cm in front of the shoulder is shown in Figure 5-5.

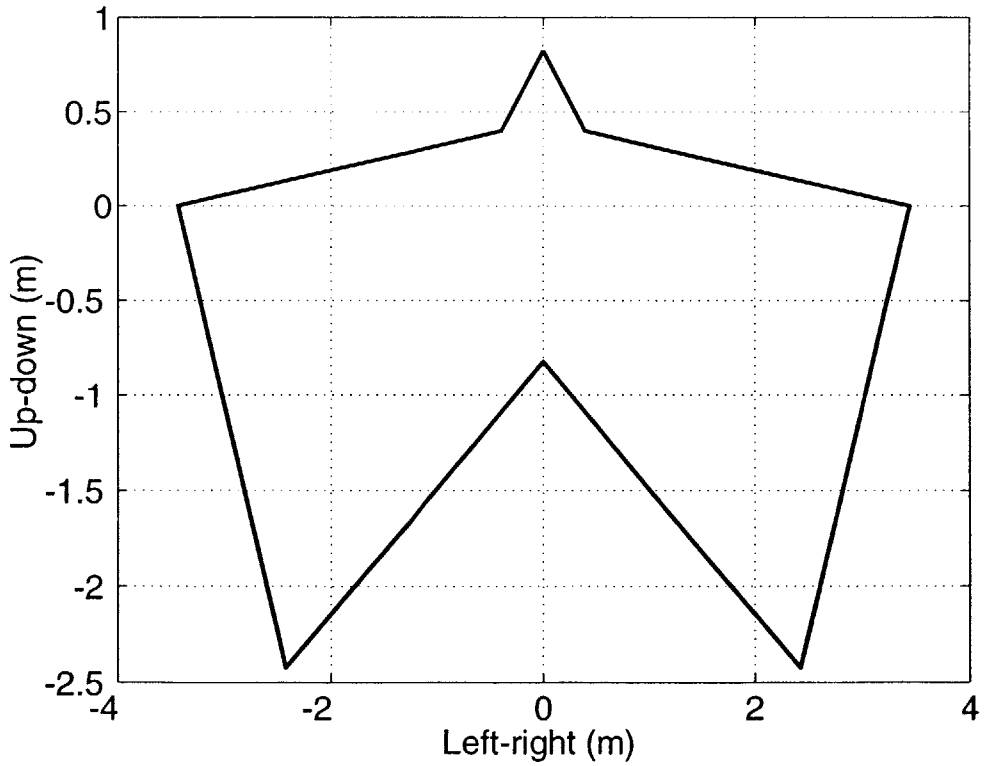


Figure 5-5. Field of view boundaries 20 cm in front of the eyepoint position.

5.3.2 Inverse kinematics

Inverse kinematics refers to the process of determining the joint angles that are required to place an end effector at a specified point in space. Inverse kinematic analysis of the arm is calculated using a geometric method described by Korein[31] for a four degree of freedom arm. Korein pointed out an important simplification, that the elbow angle depends only on the distance between the shoulder and the target hand position. Using this result, the elbow angle can be decoupled from the shoulder angles, allowing the three shoulder angles to be calculated separately from the elbow angle.

The 4 degree of freedom arm is shown in Figure 5-6. Points S, E, and W represent the shoulder, elbow, and wrist joint locations. Coordinates are centered on point S, which is at the center of the right shoulder, and coordinate directions are defined in relation to the body. The z axis points down, the y axis points forward, and the x axis points to the left.

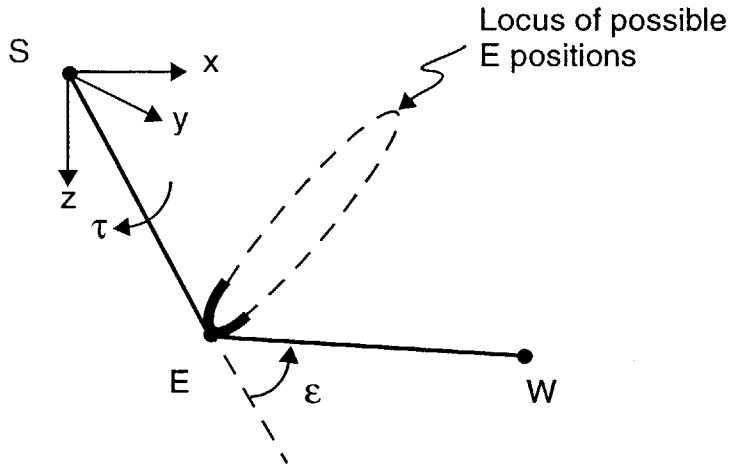


Figure 5-6. Two-link arm, showing the shoulder (S), elbow (E) and wrist (W) locations.

The hand target position uniquely determines the elbow angle, while coordinated displacements of the shoulder flexion, shoulder abduction, and humerus rotation joints allow the arm to assume multiple configurations that all place the hand on the target. If 360 degree rotation of all joints were allowed, the locus of possible elbow positions that place the hand on target would be the dashed circle shown in Figure 5-6. Since the joint ranges are limited, the elbow position locus forms arcs on the circle, like the solid arc in Figure 5-6. In virtually all cases, there are many possible arm configurations that place the hand at its target position.

The following procedure, adapted from Korein's[31] arm inverse kinematics method, is used to determine if a proposed hand target point W lies within the arm reach space and calculate the shoulder flexion, shoulder abduction, humerus rotation and elbow flexion angles required to place the hand at the target position. Rot(x,θ) indicates the rotation matrix required to rotate a vector about an axis described by vector x by angle θ.[26]

1. Determine the elbow angle ε.

The elbow angle ε is determined uniquely by the distance between the shoulder joint and the hand target point, $|\underline{w}|$. Based on $|\underline{w}|$ and the lengths of the upper and lower arm L_u and L_l, ε is calculated from the law of cosines, as shown in Figure 5-7 and Equation 5-1.

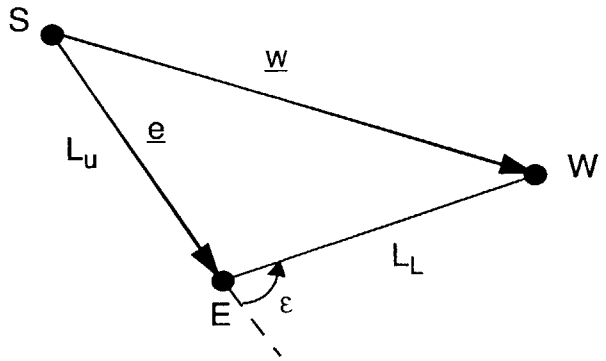


Figure 5-7. The vector w and lengths L_u and L_L determine the elbow angle ε

$$\varepsilon = \pi - \text{acos}\left[\frac{(L_u^2 + L_L^2 - |\underline{w}|^2)}{2L_u L_L}\right] \qquad \text{Eq 5-1}$$

The hand position $\underline{w}_1$ that results from the elbow joint rotation alone is determined by Equation 5-2. The hand position $\underline{w}_1$ is a three-element vector in the coordinate frame given in Figure 5-6.

$$\underline{w}_1 = L_u \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + L_L \begin{bmatrix} \sin(\pi - \varepsilon) \\ 0 \\ -\cos(\pi - \varepsilon) \end{bmatrix} \qquad \text{Eq 5-2}$$

2. Prescribe a humerus rotation angle, τ.

A range of humerus rotation angles, τ, distributed evenly over the 0-180 degree range, is chosen. The hand position that results from each of the humerus rotation angles is then calculated.

$$\underline{w}_2 = \text{Rot}(\hat{\underline{z}}, \tau)\underline{w}_1 \qquad \text{Eq 5-3}$$

Any choice of τ that results in permissible values for the other three arm joint angles (shoulder flexion, shoulder abduction, and humerus rotation) is an allowed arm configuration. To resolve the kinematic redundancy, the inverse kinematics calculations were carried out in parallel for multiple values of τ, then one arm configuration was chosen to minimize the torques required at the joints to maintain the arm configuration.

3. Rotate the shoulder to bring $\underline{w}_2$ to $\underline{w}$ without changing the humerus rotation angle.

The shoulder flexion and shoulder abduction rotations are performed simultaneously by rotating vector $\underline{w}_2$ about the shoulder so that the hand moves to its target location. In order to preserve the τ specified in step 2, the rotation axis must be chosen to avoid rotations about the z axis. The rotation axis must also be equidistant from $\underline{w}_2$ and $\underline{w}$, in order to rotate $\underline{w}_2$ to $\underline{w}$. The unit vector in the x-y plane that is equidistant from $\underline{w}_2$ and $\underline{w}$ is defined as $\hat{\underline{a}}$ and is given by:

$$\hat{\underline{a}} = \frac{\begin{bmatrix} (w_y - w_{y2}) \\ -(w_x - w_{x2}) \\ 0 \end{bmatrix}}{\left\| \begin{bmatrix} (w_y - w_{y2}) \\ (w_x - w_{x2}) \\ 0 \end{bmatrix} \right\|}. \qquad \text{Eq 5-4}$$

The shoulder rotation angle required, η, is the angle between the projections of $\underline{w}_2$ and $\underline{w}$ on the plane perpendicular to $\hat{\underline{a}}$. To determine η, the projections of $\underline{w}_2$ and $\underline{w}$ on the plane perpendicular to $\hat{\underline{a}}$ are defined as follows.

$$\begin{aligned} \underline{\mathrm{pw}} &= (\hat{\underline{a}} \times \hat{\underline{w}}) \times \hat{\underline{a}} \\ \underline{\mathrm{pw}}_2 &= (\hat{\underline{a}} \times \hat{\underline{w}}_2) \times \hat{\underline{a}} \end{aligned} \qquad \text{Eq 5-5}$$

The angle between $\underline{pw}$ and $\underline{pw}_2$ determines η.

$$\eta = \mathrm{acos}\left[\frac{\underline{\mathrm{pw}} \bullet \underline{\mathrm{pw}}_2}{|\underline{\mathrm{pw}}||\underline{\mathrm{pw}}_2|} \right] \qquad \text{Eq 5-6}$$

Since the numerator and denominator in Equation 5-6 are both always positive, the sign of η is determined according to Equation 5-7, so that the rotation is right-handed with $\hat{\underline{a}}$.

$$\begin{aligned} &\eta > 0, \underline{w} \times \underline{w}_2 \bullet \hat{\underline{a}} > 0 \\ &\eta < 0, \underline{w} \times \underline{w}_2 \bullet \hat{\underline{a}} < 0 \end{aligned} \qquad \text{Eq 5-7}$$

This single rotation of angle η about axis $\hat{\underline{a}}$ will later be decomposed into shoulder flexion and shoulder abduction angles.

4. Determine the elbow position.

The new elbow position $\underline{e}_1$ that results from the shoulder rotation is determined by rotating the vector from point S to point E by angle η about axis $\hat{\underline{a}}$.

$$\underline{e}_1 = \underline{e}\mathrm{Rot}(\hat{\underline{a}}, \eta) \qquad \text{Eq 5-8}$$

5. Determine the shoulder flexion and shoulder abduction angles.

The shoulder flexion and shoulder abduction angles can be determined by the vector from the shoulder S to the elbow $\underline{e}_1$. The shoulder flexion and shoulder abduction degrees of freedom are assumed to be successive rotations about perpendicular axes. The shoulder flexion (SF) rotation, which is performed first, is a rotation about the negative x axis. The shoulder abduction (SA) rotation, performed second, is a rotation about the a negative y axis that is attached to the upper arm segment and rotated with the shoulder flexion rotation. While the order of the two shoulder rotations must be specified and applied uniformly, there is no inherently preferred ordering. The RSST shoulder mechanism performs shoulder flexion rotation first, so that ordering is preserved in the inverse kinematics calculations.

$$\mathrm{SF} = \mathrm{atan}\left(\frac{e_{1y}}{e_{1z}}\right) \qquad \mathrm{SA} = \begin{cases} \mathrm{atan}\left(\dfrac{e_{1x}}{e_{1y}}\sin(\mathrm{SF})\right), & \mathrm{SF} \neq 0 \\ \mathrm{atan}\left(\dfrac{e_{1x}}{e_{1y}}\right), & \mathrm{SF} = 0 \end{cases} \qquad \text{Eq 5-9}$$

At this point in the procedure, angles for the four arm joints have been calculated for 10 arm configurations corresponding to the 10 prescribed values of humerus rotation angle. All of these arm configurations place the hand on the target, although some of them are unattainable because they exceed the available joint ranges of motion. The space suit model is used next to determine the torque required to maintain each of these arm configurations. A single arm configuration is then chosen as a unique solution, based on the torques needed to hold the configuration.

5.3.3 Required joint torques

The joint torques required to maintain the hand in a specific position depend on the torques needed to hold the space suit joints in their bent positions. These static torques can be derived from the experimental torque-angle data collected in Chapter 3. Only the static torques that are required to hold the space suit joint in a bent position are considered, rather than the torques needed to move from one arm position to another, for two reasons. First, the reaching motion from an initial hand position to a final hand position is a short-duration motion. In motions that last only several seconds, it is acceptable to exert torques that are substantially higher than the low torque limits that are imposed for torques that are held for several minutes. Second, the torques required to move the hand from an initial position to a final position are highly dependent on the initial position and the path taken

between the initial and final positions, neither of which can be determined in general for typical EVA work situations.

Articulations in the space suit's arm are either rotational bearings that are at equilibrium at any rotation angle or bending joints that tend to return to an equilibrium angle. While torque is required to rotate the bearing joints, no torque is required to hold the joint in a static position. The humerus rotation joint is an instance where a space suit bearing coincides with a human joint degree of freedom. Consequently, no torque is required at the humerus rotation joint to maintain any desired humerus rotation angle that is within the unsuited human range of motion. Limits on humerus rotation are therefore implemented as angle limitations, rather than torque limits. The humerus rotation angle range that was used in all anthropometric cases was -113 deg to 70 deg.[2]

In contrast to the humerus rotation joint, the elbow and shoulder joints require transverse bending of the space suit arm. Although the shoulder joint does incorporate bearings, it is considered a bending joint in this analysis because most shoulder positions require bending of the segment between the scye bearing and the arm bearing.

Torques required to maintain the space suit shoulder flexion, shoulder abduction, and elbow flexion joints were calculated using a simplified version of the space suit joint stiffness model, which predicts torque as a function of joint angle and angular velocity, as shown in Equation 5-10.

$$\text{Torque} = k_1 + k_2 \times \text{angle} + k_3 \times \text{angular velocity} \qquad \text{Eq 5-10}$$

Because the arm position is considered to be static in the work envelope configuration, the angular velocity term in Equation 5-10 is set to zero, to isolate the torque needed to hold a specific joint angle. Values of k_1, k_2, and k_3 and the regression statistics R^2 and standard error of the fitted torque are given in Table 5-4.

Table 5-4: Torque model coefficients

Joint	k_1 (Nm)	k_2 (Nm/ deg)	k_3 (Nm/ (deg/sec))	R^2	standard error (Nm)
Shoulder flexion	-23.4	0.222	0.120	0.777	4.39
Shoulder abduction >50 deg	-75.5	1.20	0.372	0.743	6.42
Shoulder abduction <50 deg	-27.2	0.303	0.110	0.899	0.553
Elbow flexion	-9.04	0.155	0.0380	0.823	2.12

The model outputs for each joint are compared to experimental data in Figure 5-8-Figure 5-10. The experimental joint torque data shown includes all of the available data from human-generated motions for that joint, including data from three test subjects. Torque and angle data from multiple test subjects was concatenated, not averaged.

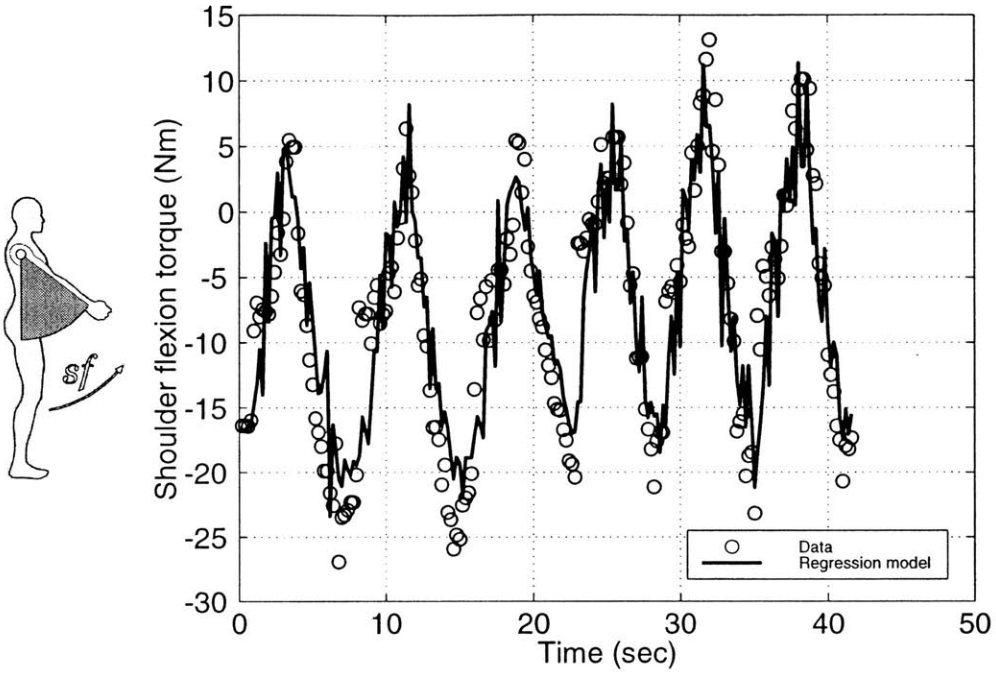


Figure 5-8. Shoulder flexion torque and regression model output.

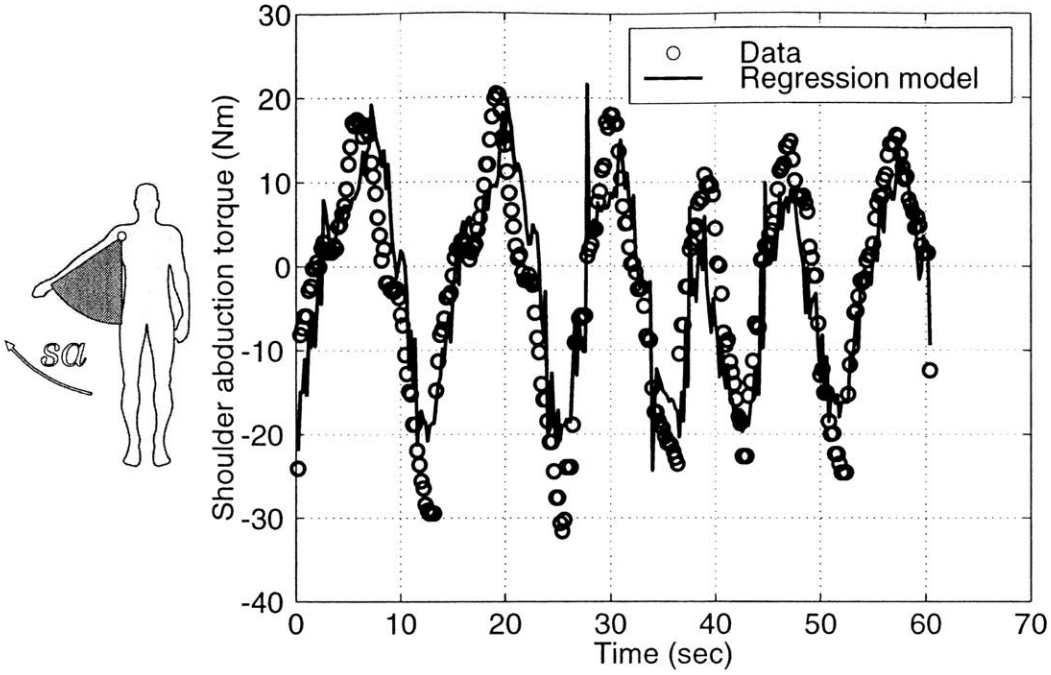


Figure 5-9. Shoulder abduction torque and regression model output.

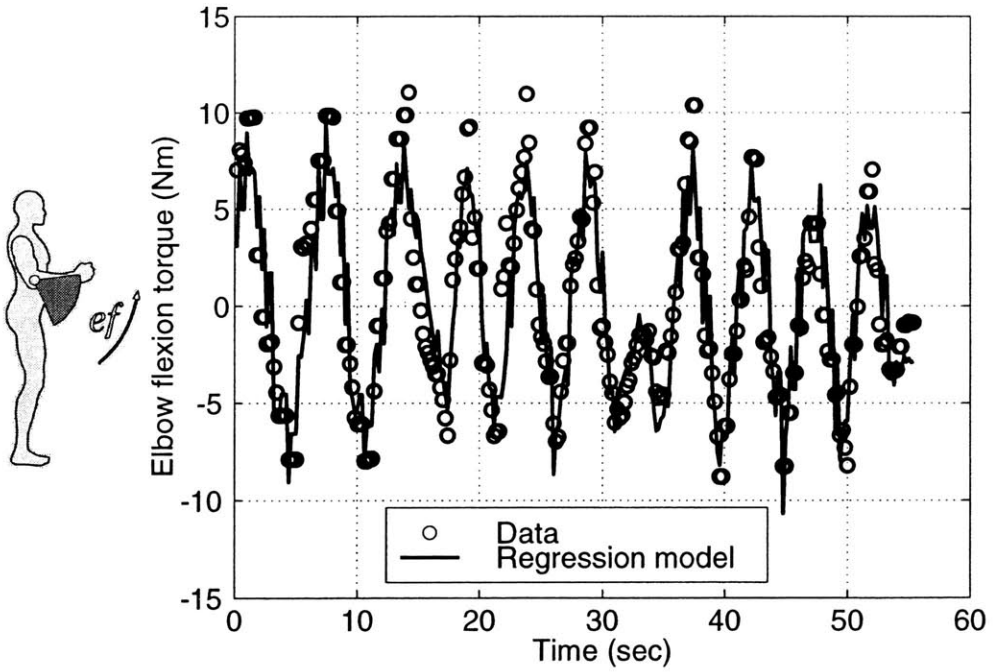


Figure 5-10. Elbow flexion torque and regression model.

5.3.4 Reach difficulty metric

The difficulty of reaching a specific location is indicated by the reach difficulty metric, which is a nondimensional, positive scalar corresponding to each proposed arm configuration. A low reach difficulty metric means that the arm configuration is easy to maintain; a high reach difficulty metric means that the arm configuration is difficult to maintain. The reach difficulty metric is the sum over the four arm joints of the required joint torque divided by either the positive or negative torque limit for that joint, as shown in Equation 5-11.

$$\text{metric} = \sum_{\text{joint 1}}^{\text{joint 4}} \left(\frac{\text{torque}}{\text{torque limit}} \right) \qquad \text{Eq 5-11}$$

For each proposed hand location, the inverse kinematics analysis outputs several arm configurations that place the hand at the target location. The torques required to maintain these arm configurations are compared to the torque limits. If the torque at any joint exceeds its limit, that arm configuration is eliminated from consideration. When more than one arm configuration is allowable, the one with the lowest reach difficulty metric is chosen as the single solution for that hand location. As a result of the choice of a single arm configuration for each proposed hand location, one value of the reach difficulty metric corresponds to each point in the proposed workspace. There are no constraints applied to the reach difficulty metric; the torque limits that determine whether a point is within the work envelope apply only to torques required from individual joints.

5.3.5 Boundary smoothing

The boundaries of the work envelope regions are smoothed using image processing techniques to set the simplest work envelope boundaries that include as many low-torque points as possible and exclude excessive-torque and nonvisible points. The smoothing technique chosen is objective and systematic and uses standard Matlab library functions, although other smoothing techniques may be used to accomplish similar results. A binary image is formed from a horizontal or vertical planar section of the workspace, with points in the "always include" category set to be black and points outside that category set to be white. Points in the "always include" category, the region of points that is visible and requires no more than 15% of the maximum torque for each joint, which is used as a starting point, are shown in Figure 5-11A, for a horizontal slice 25 cm above the shoulder. Additional points from the "possibly include" category are added to this region, or colored black, using morphological operators to fill interior gaps in the black region, as shown in Figure 5-11B. A gaussian low-pass filter is then applied to the image, resulting in a blurred, grayscale version of the image, in Figure 5-11C. The grayscale image is converted back to binary form, using a variable threshold value. The threshold is chosen to meet two conditions:

1. The smoothing process may add no more than 30% new points to the image
2. A piecewise linear boundary drawn around the white region in the smoothed image

should require the minimum number of vertices.

The result of the smoothing process for a horizontal slice is shown in Figure 5-11D. In the case shown in Figure 5-11, the smoothing operation increased the number of points considered inside the workspace to 80 points from 64 points in the "always include" region, and reduced the number of vertices in the boundary to 16 from 22. The smoothing algorithm was carried out first for horizontal slices, then for vertical slices through the workspace region.

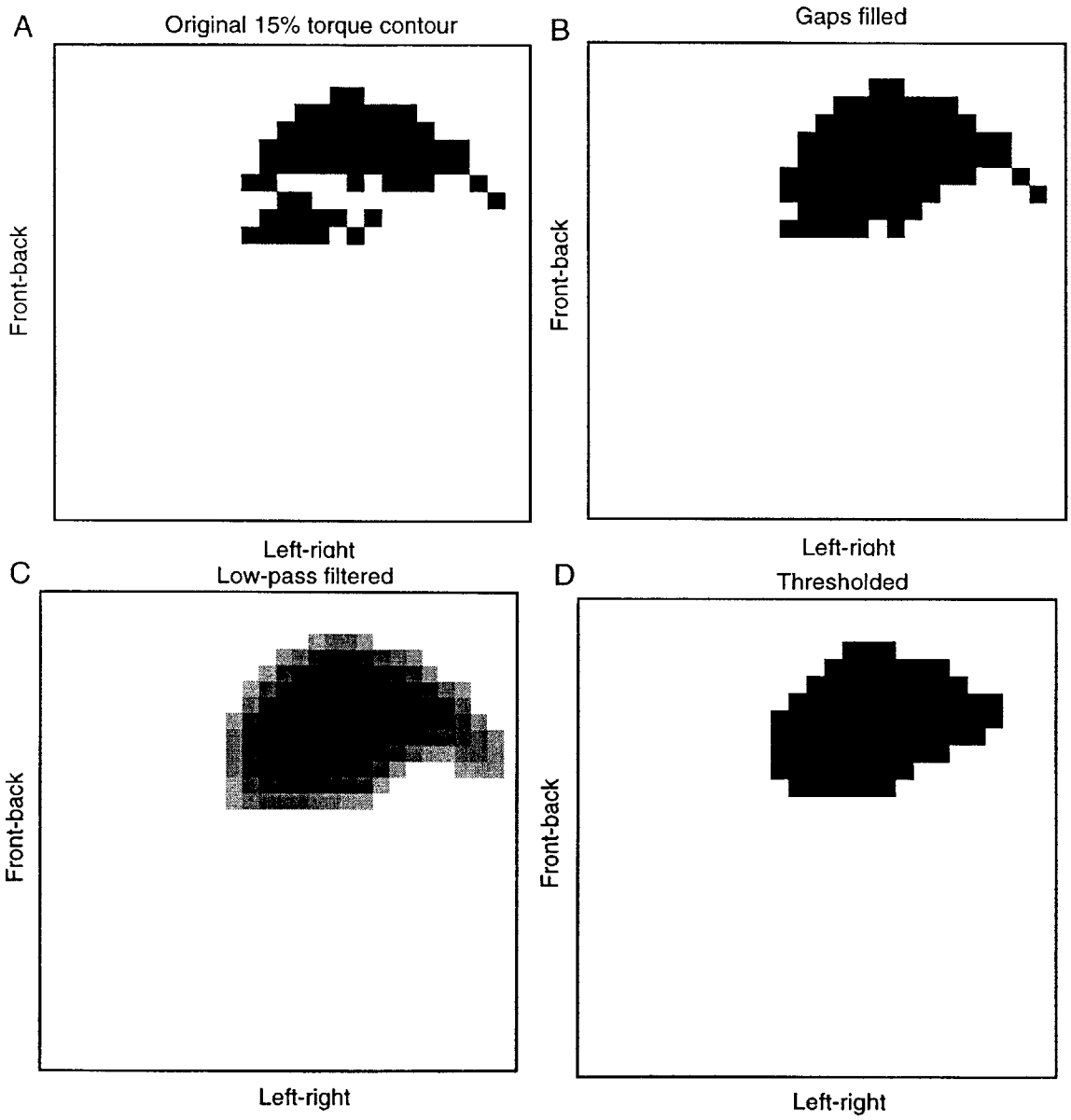


Figure 5-11. Intermediate results of work envelope boundary smoothing. A.) Image formed from "always include" region. B.) Gaps filled by adding points from the "possibly include" region. C.) Low-pass filtered image. D.) Final result of smoothing process.

5.4 Work envelope results

Work envelopes were calculated for eight cases, encompassing variability in size and strength for both male and female anthropometric data, as listed in Table 5-5.

Table 5-5: Work envelope cases evaluated

Male/Female	Size percentile	Strength percentile
M	50	50
M	50	95
M	95	50
M	95	95
F	50	50
F	50	95
F	95	50
F	95	95

5.4.1 Joint angle ranges

The space suit torque model, arm segment lengths, and joint torque limits determine the angle range that is generated by the work envelope analysis for each joint. Ranges of motion of each of the four arm joints for torque limits of 15%, 30%, 50% and 100% of the maximum isometric strength are shown in Figure 5-12-Figure 5-19, for each of the cases listed in Table 5-5. These ranges of motion are the maximum and minimum joint angles selected in the inverse kinematics analysis. While they do span the vast majority of available joint angles, they are not mathematically guaranteed to cover the entire range of feasible joint angles predicted by the space suit torque model and the strength limits. Consequently, small numerical differences can be seen between the angle ranges for cases that have the same strength limits but different arm lengths.

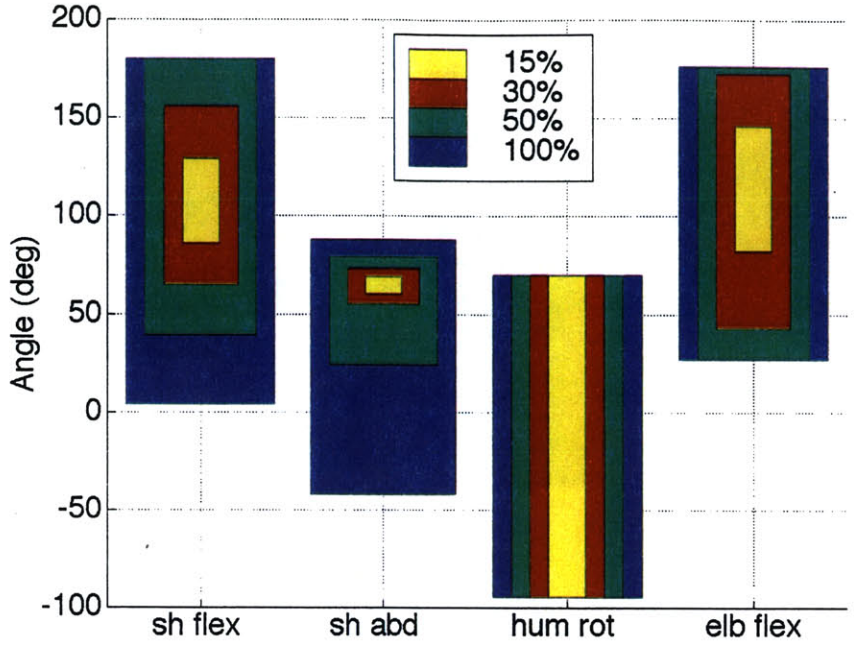


Figure 5-12. Joint angle ranges for female, 50th percentile size, 50th percentile strength deduced from work envelope analysis.

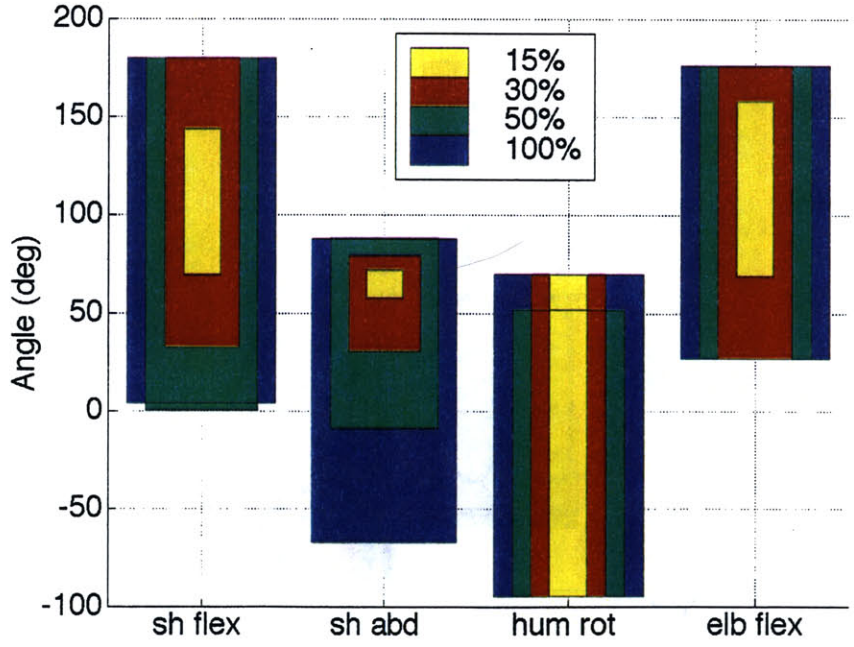


Figure 5-13. Joint angle ranges for female, 50th percentile size, 95th percentile strength deduced from work envelope analysis.

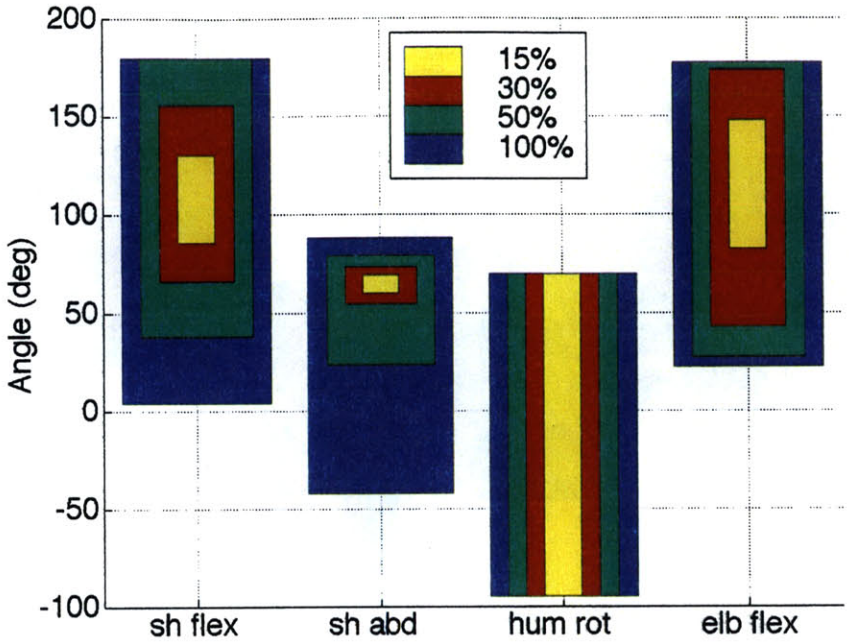


Figure 5-14. Joint angle ranges for female, 95th percentile size, 50th percentile strength deduced from work envelope analysis.

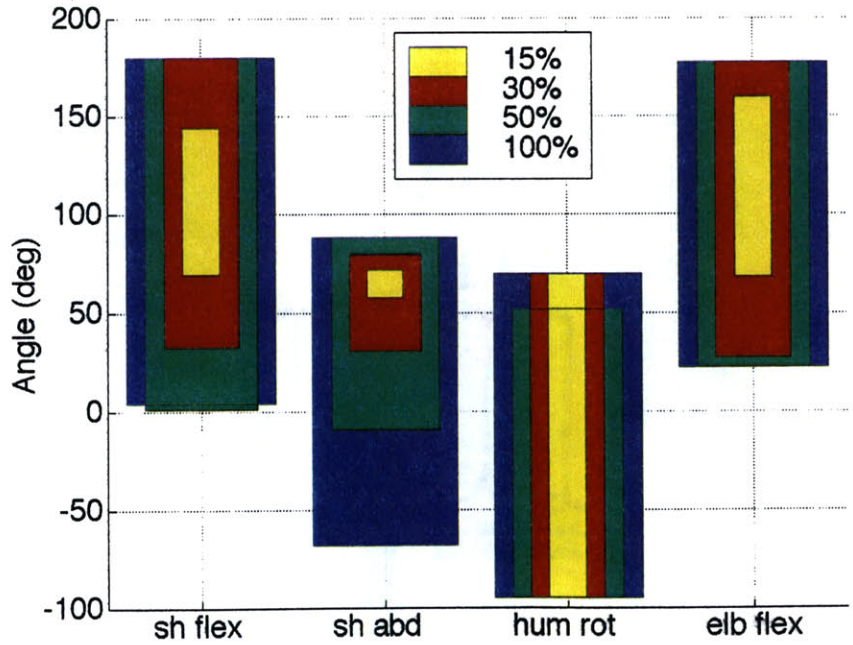


Figure 5-15. Joint angle ranges for female, 95th percentile size, 95th percentile strength deduced from work envelope analysis.

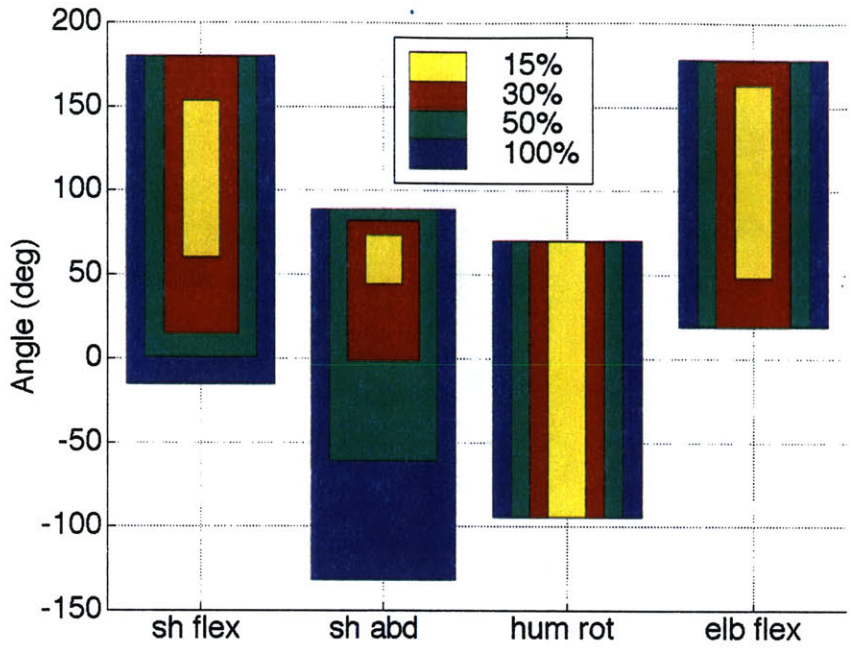


Figure 5-16. Joint angle ranges for male, 50th percentile size, 50th percentile strength deduced from work envelope analysis.

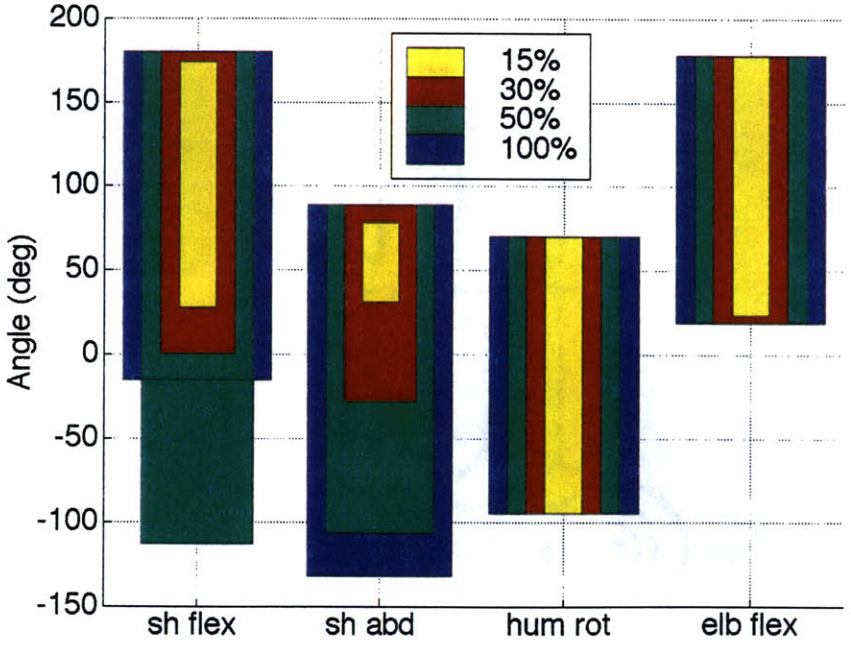


Figure 5-17. Joint angle ranges for male, 50th percentile size, 95th percentile strength deduced from work envelope analysis.

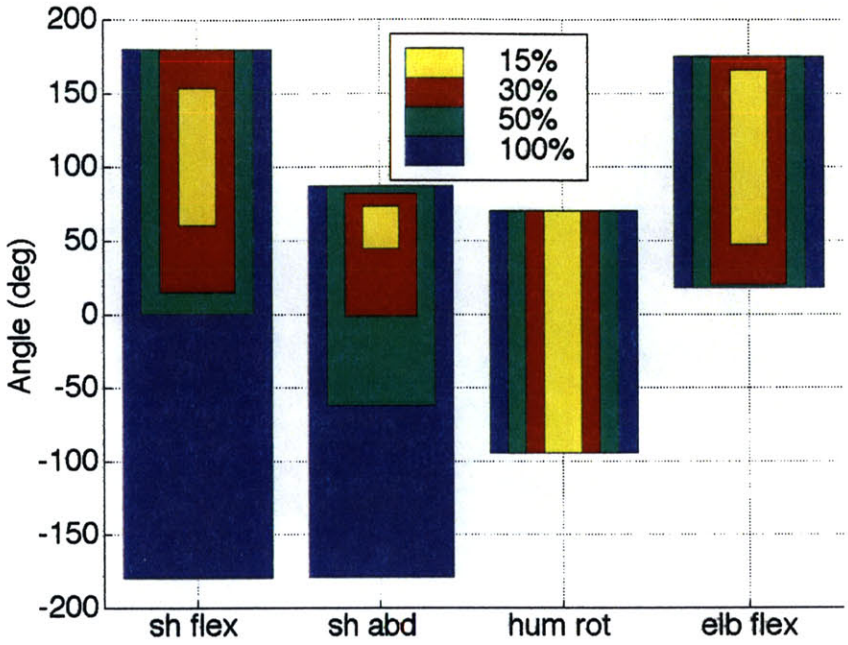


Figure 5-18. Joint angle ranges for male, 95th percentile size, 50th percentile strength deduced from work envelope analysis.

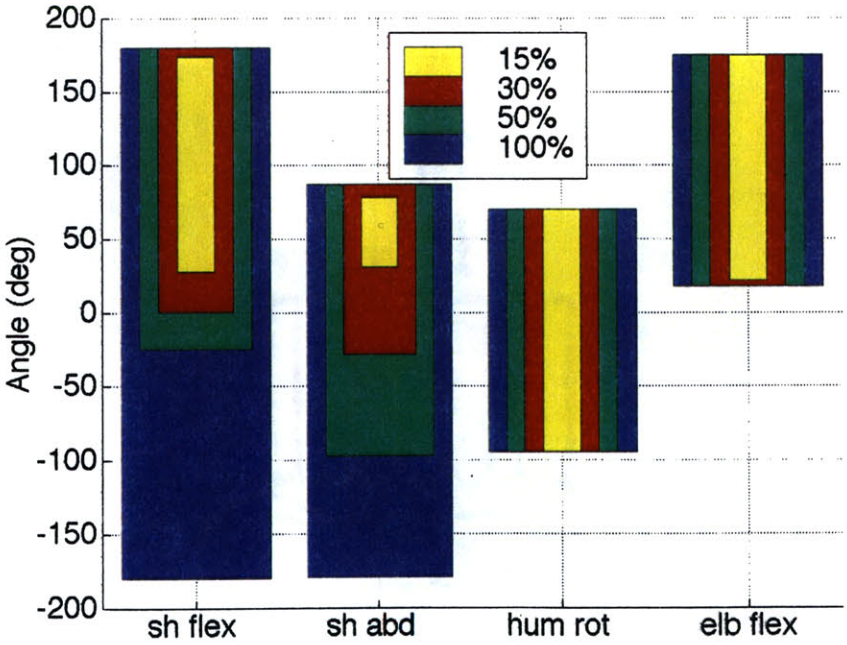


Figure 5-19. Joint angle ranges for male, 95th percentile size, 95th percentile strength deduced from work envelope analysis.

5.4.2 Boundaries for 15% of isometric strength

The boundaries for torque limits of 15% of the maximum isometric strength are shown in Figure 5-20-Figure 5-27 for all of the cases listed in Table 5-5. Contours drawn in the horizontal plane indicate the boundaries in which a space-suited person can place their hand while not using more than 15% of the maximum torque in any arm joint. The 15% torque limit contour represents the region in which a space-suited astronaut can hold his or her hand for approximately 10 minutes, without being forced to change hand positions due to fatigue in any joint. The contours shown in Figure 5-20-Figure 5-27 indicate the boundaries of the "always include" category of points, that is obtained from enforcing a single-valued torque limit. These boundaries constitute the starting point for the smoothing algorithm that determines the work envelope boundaries.

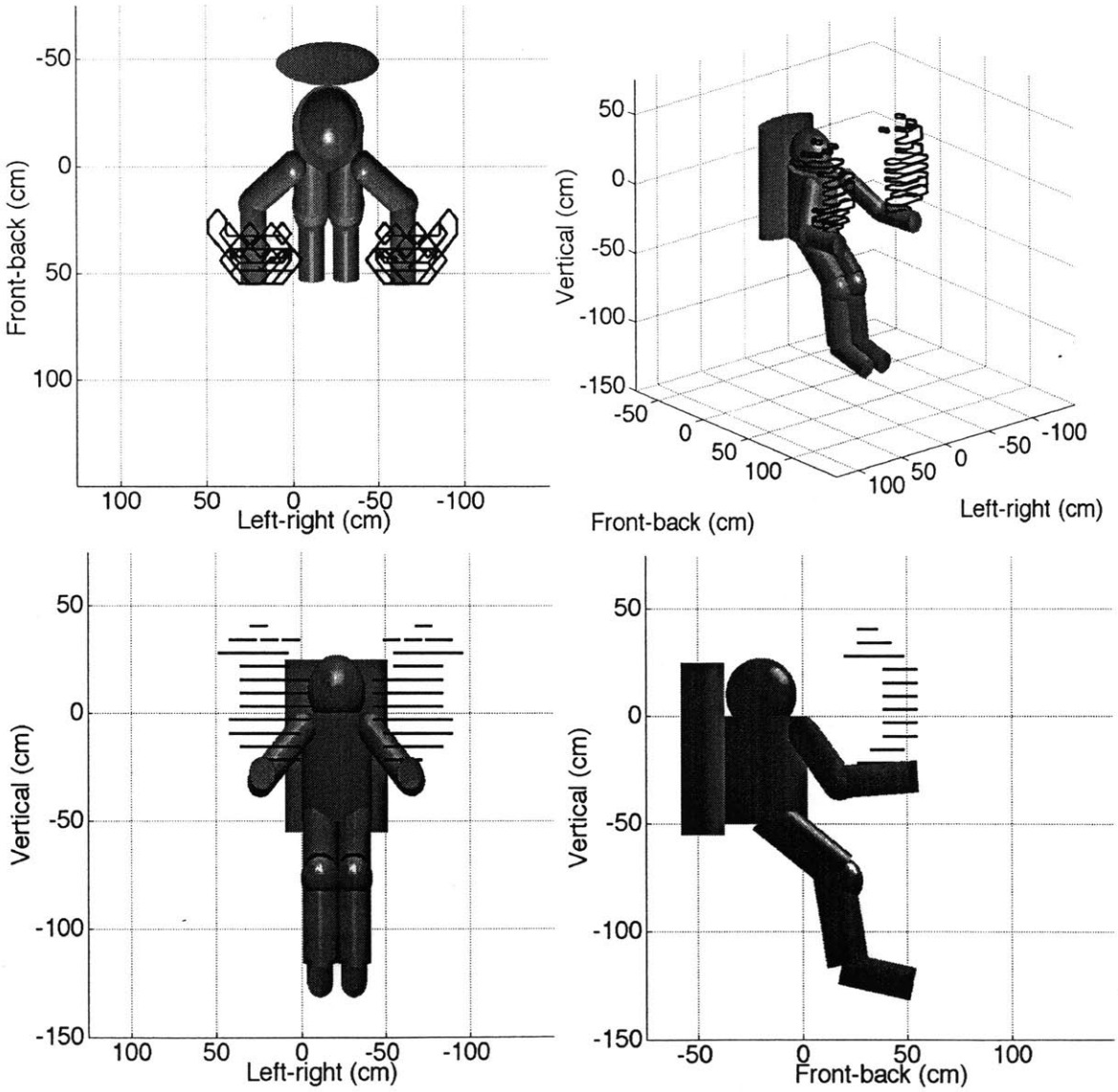


Figure 5-20. 15% torque boundaries for work envelope: female, 50th percentile strength, 50th percentile size

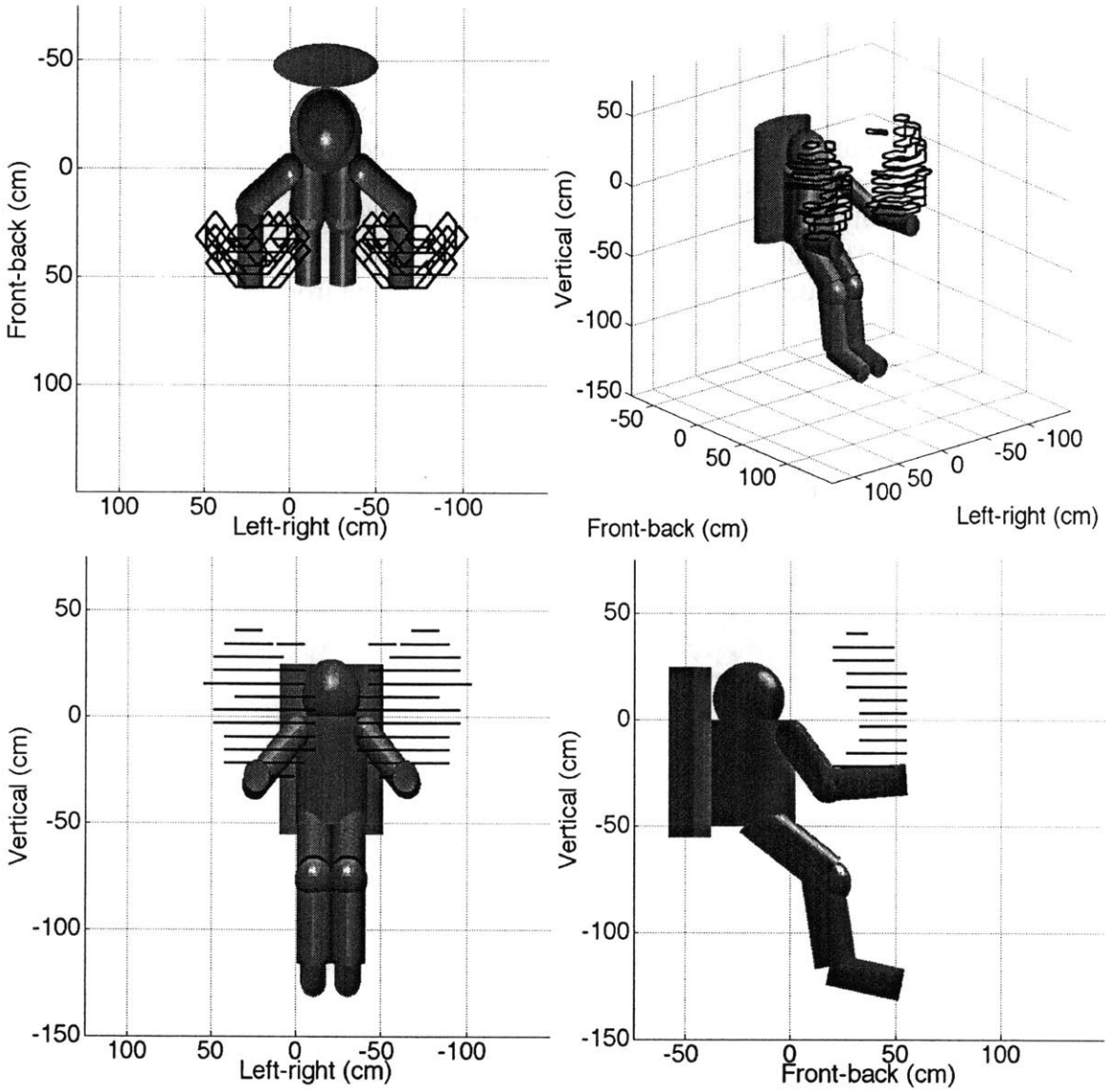


Figure 5-21. 15% torque boundaries for work envelope: female, 95th percentile strength, 50th percentile size

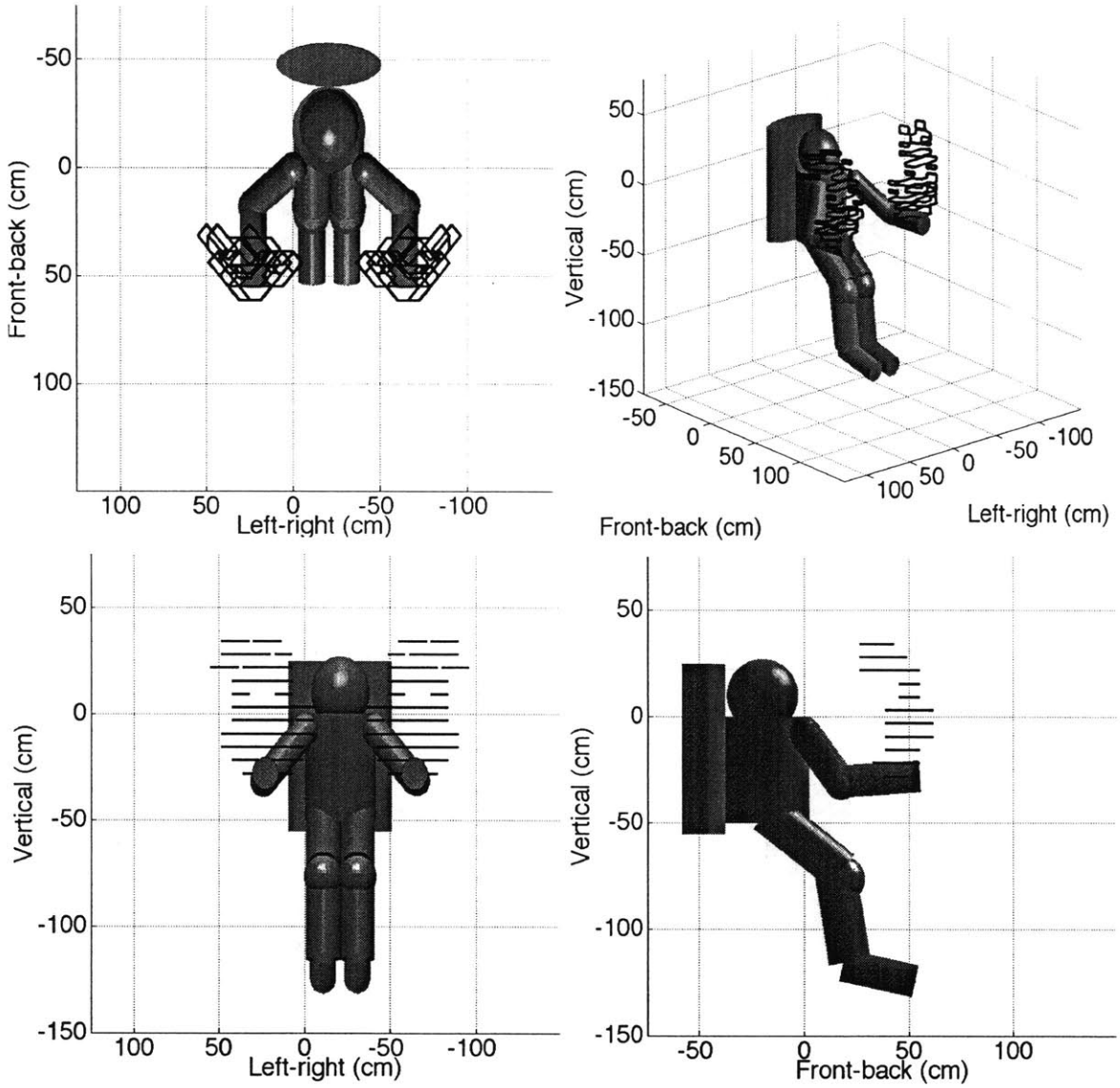


Figure 5-22. 15% torque boundaries for work envelope: female, 50th percentile strength, 95th percentile size

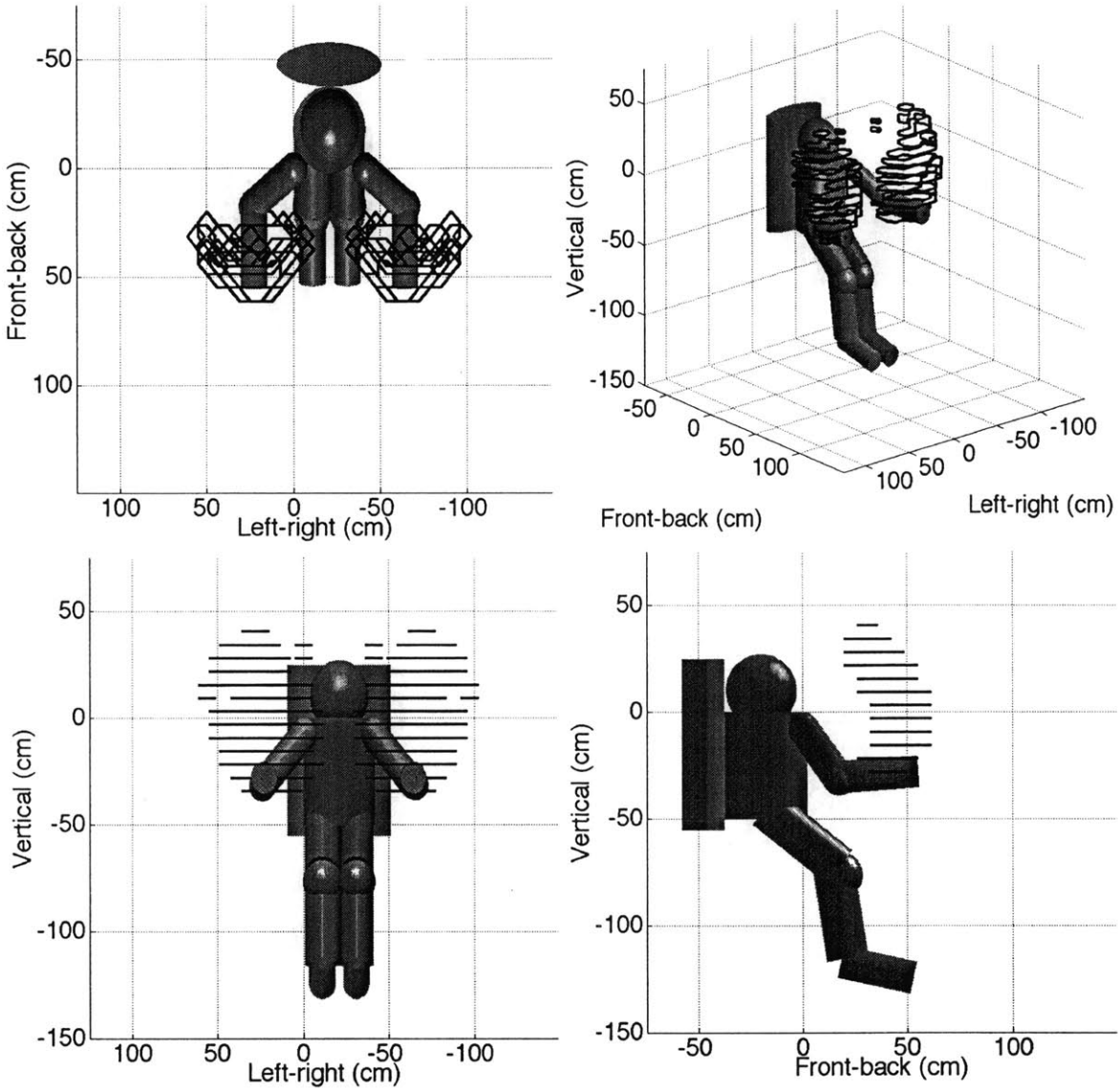


Figure 5-23. 15% torque boundaries for work envelope: female, 95th percentile strength, 95th percentile size

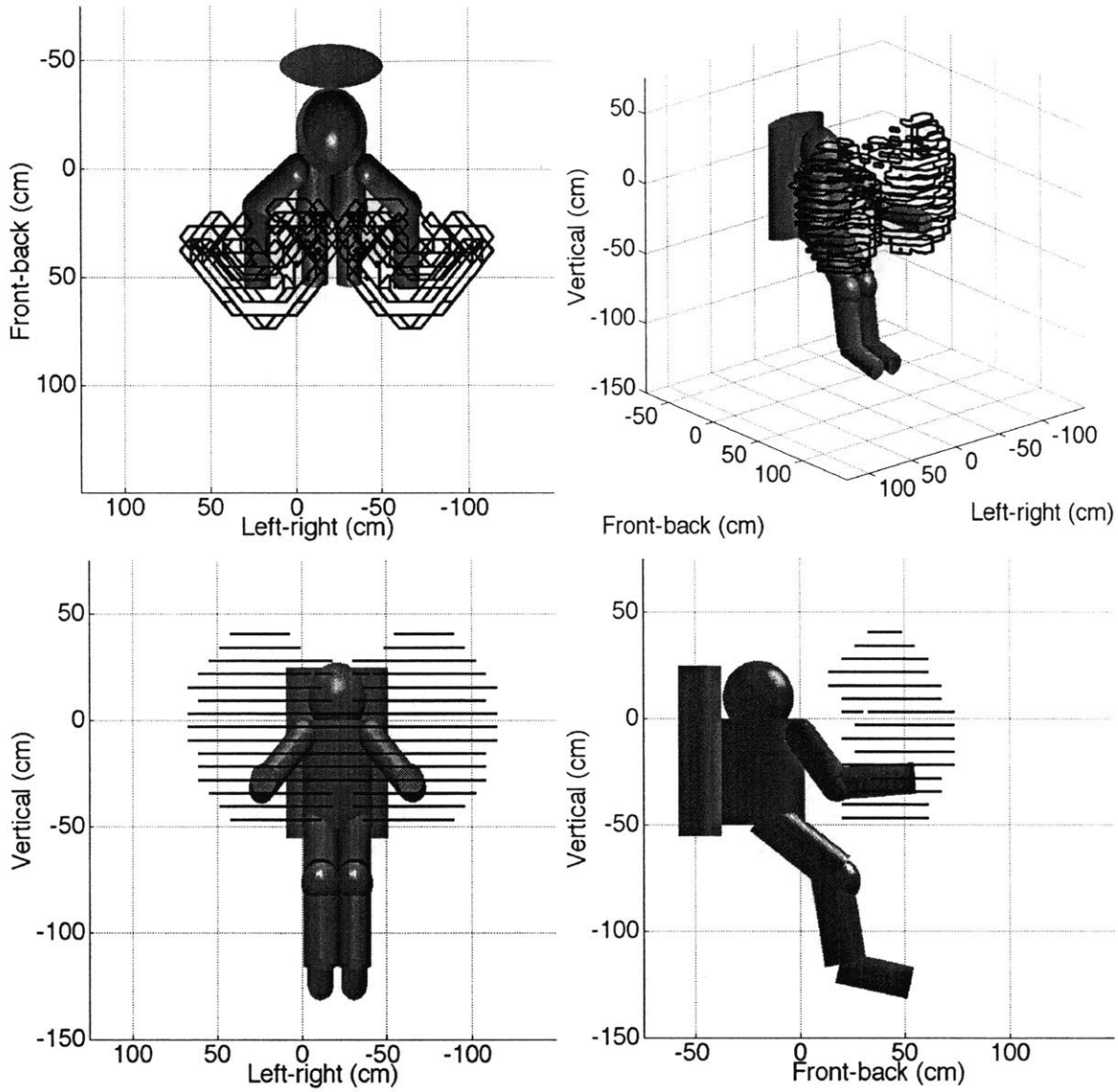


Figure 5-24. 15% torque boundaries for work envelope: male, 50th percentile strength, 50th percentile size

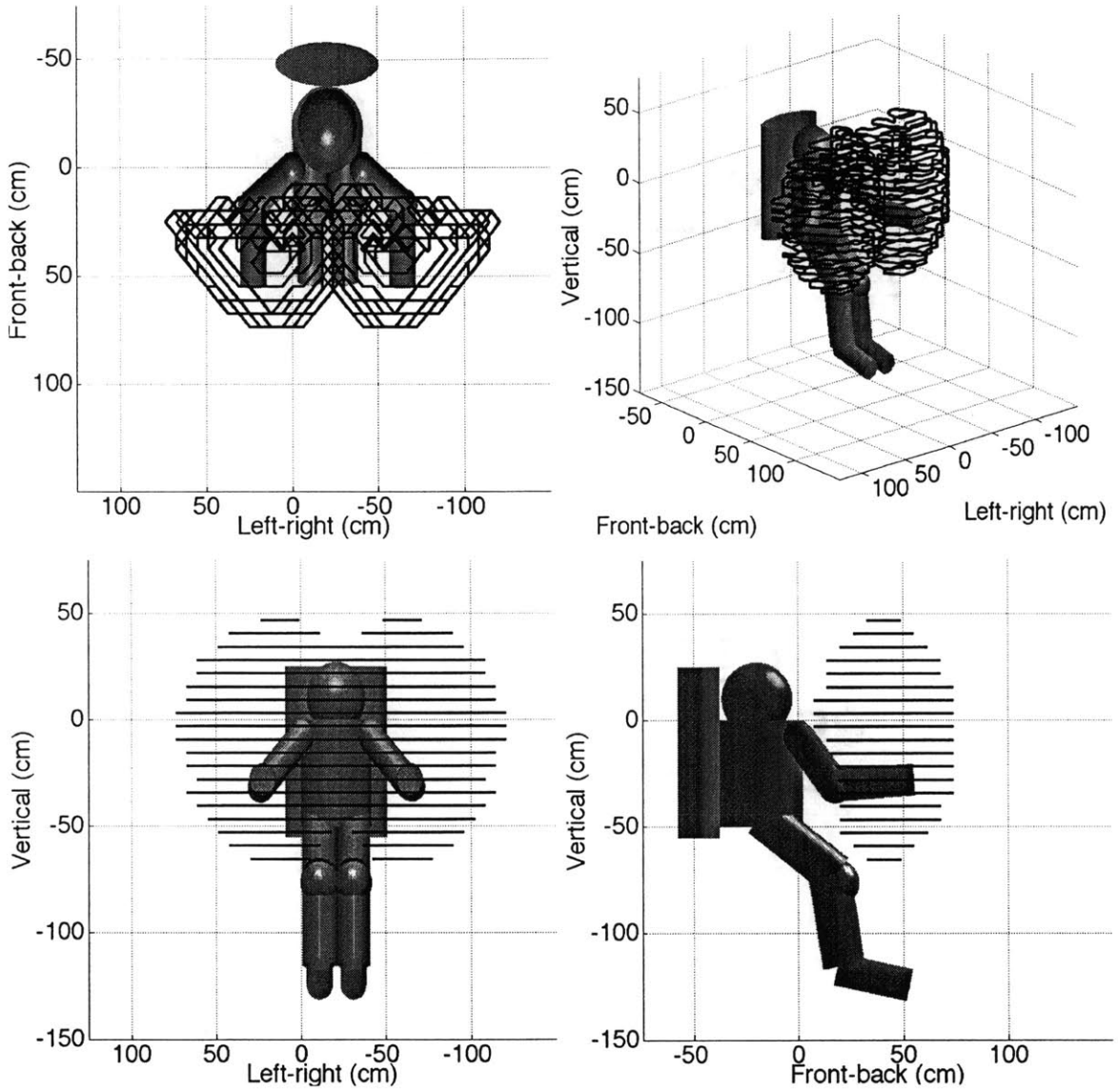


Figure 5-25. 15% torque boundaries for work envelope: male, 95th percentile size, 50th percentile strength

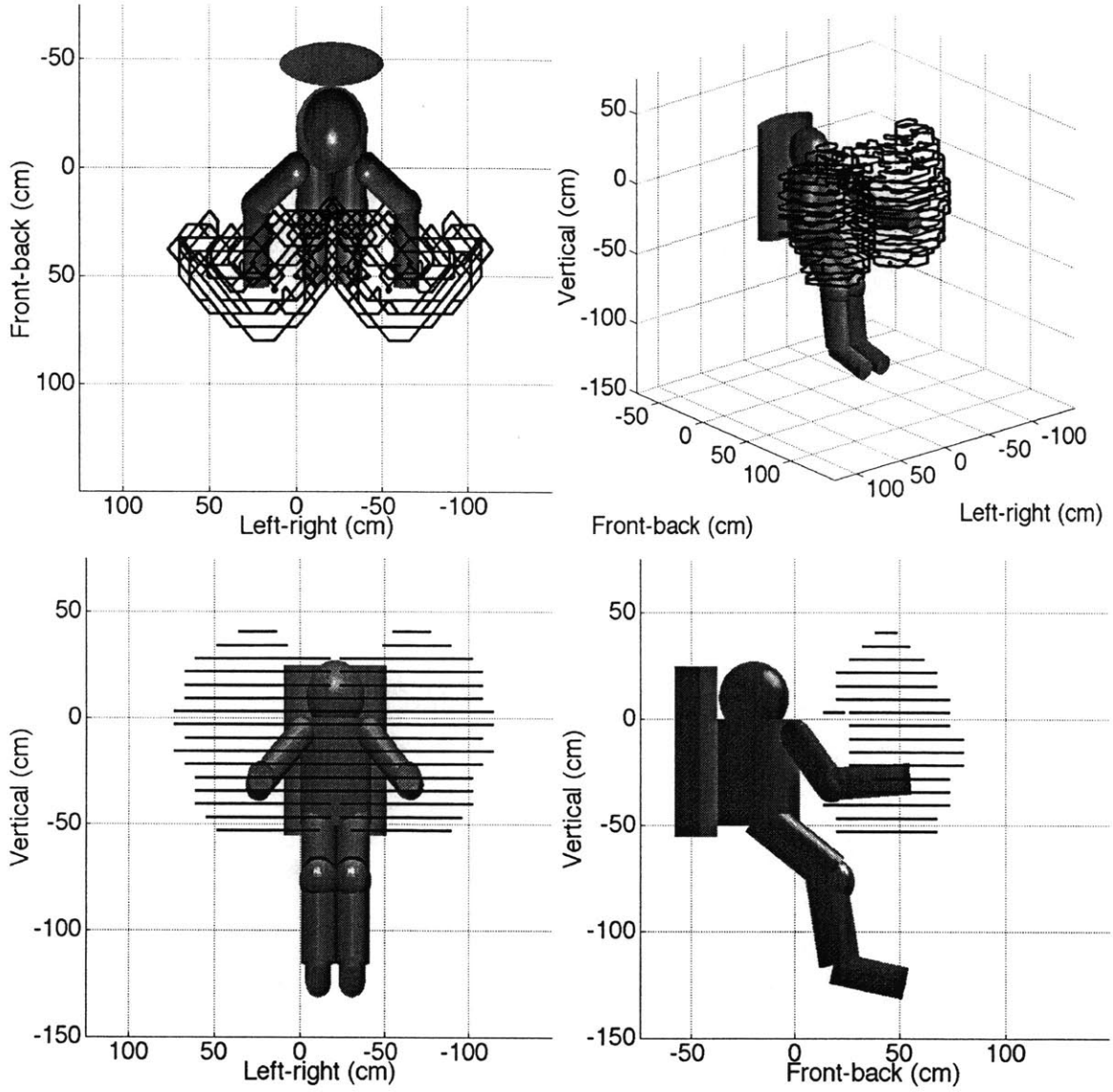


Figure 5-26. 15% torque boundaries for work envelope: male, 50th percentile strength, 95th percentile size

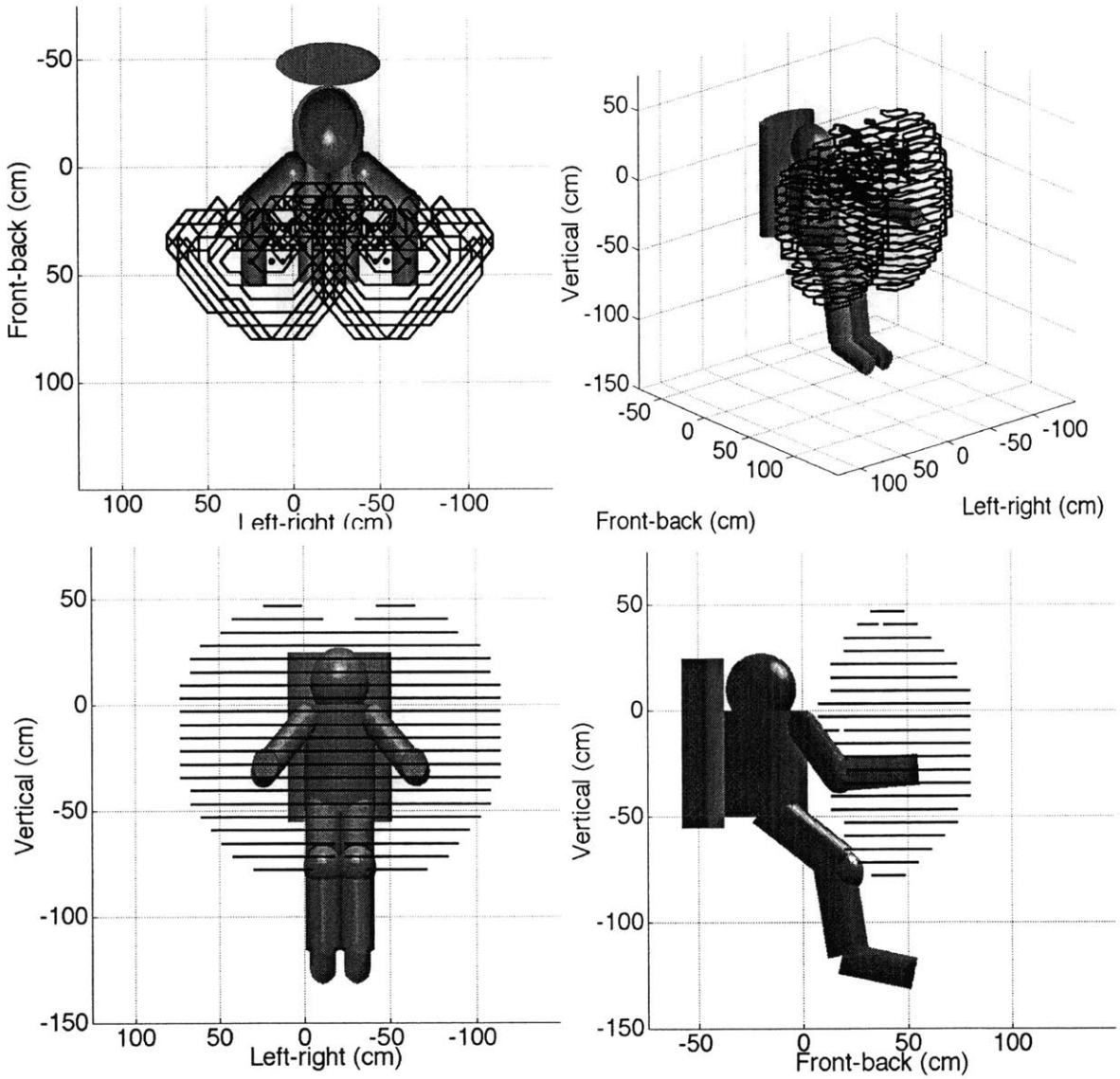


Figure 5-27. 15% torque boundaries for work envelope: male, 95th percentile strength, 95th percentile size

5.4.3 Reach difficulty metric and torque limit contours

The work envelopes can be studied in more detail by looking at horizontal slices through the workspace at various heights. Two aspects of reach difficulty are of interest: the reach difficulty metric, which indicates the overall difficulty of reaching a point which is indicated in the figures by color and the area that falls within a maximum torque limit, which indicates the region that can be reached without exceeding a specified torque limit at any single joint and is indicated in the figures by boundary lines. Figure 5-28 and Figure 5-29 show the reach difficulty metric and torque limit contours at 15%, 30%, 50%, and 100% of the strength limits for two design points: female, 50th percentile size and 95th percentile

strength and male, 50th percentile size and 95th percentile strength. Reach difficulty metric is indicated by color, with easy to reach areas shown in blue and difficult to reach areas shown in red. The torque limit contours indicate the boundaries within which the hand can be placed without exceeding the indicated percentage of the strength limits at any joint.

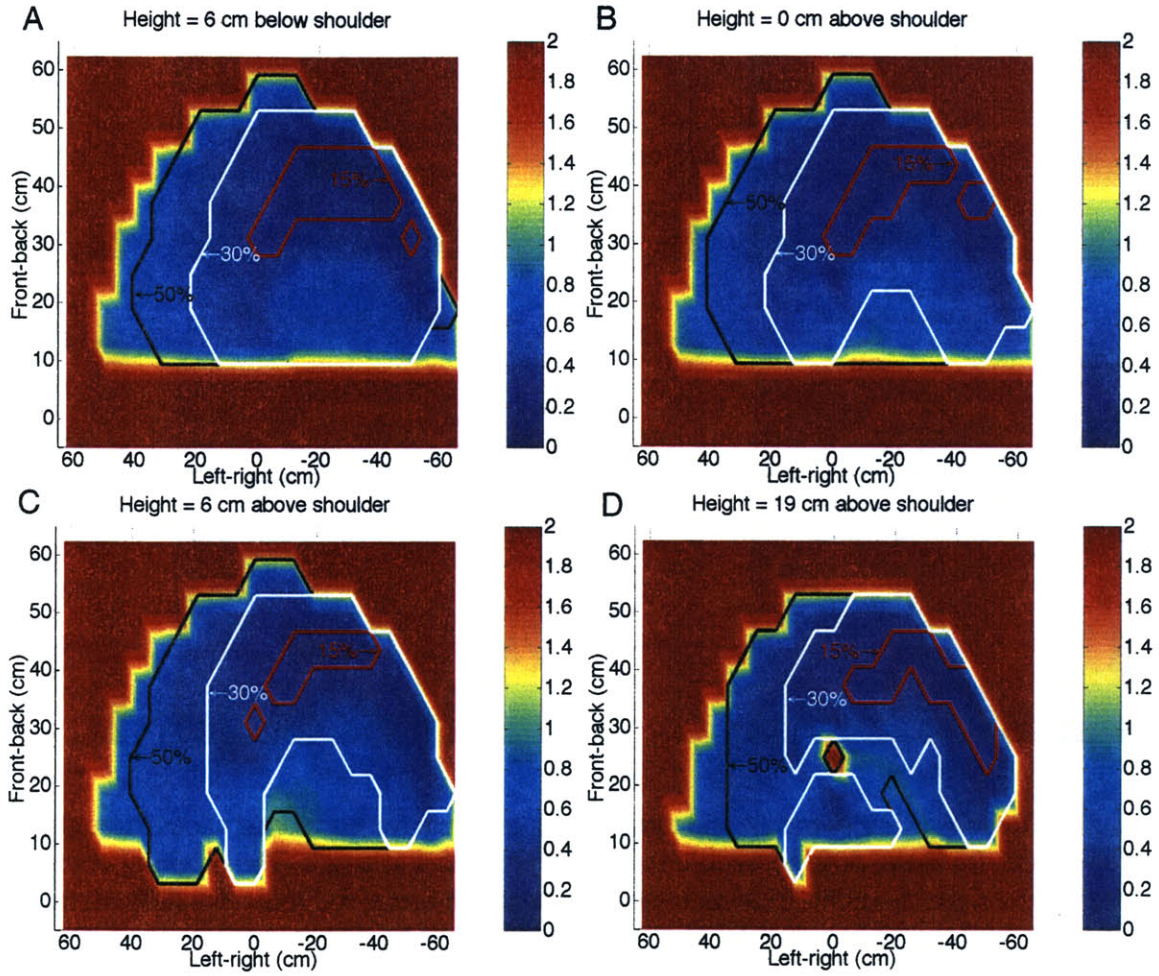


Figure 5-28. Reach difficulty metric and torque limit contours for horizontal slices through reach envelope, for female, 50th percentile size, 95th percentile strength. The origin of coordinates is at the shoulder. Blue indicates easy to reach areas; red indicates difficult to reach areas.A.) 6 cm below shoulder. B.) At shoulder level. C.) 6 cm above shoulder. D.) 19 cm above shoulder.

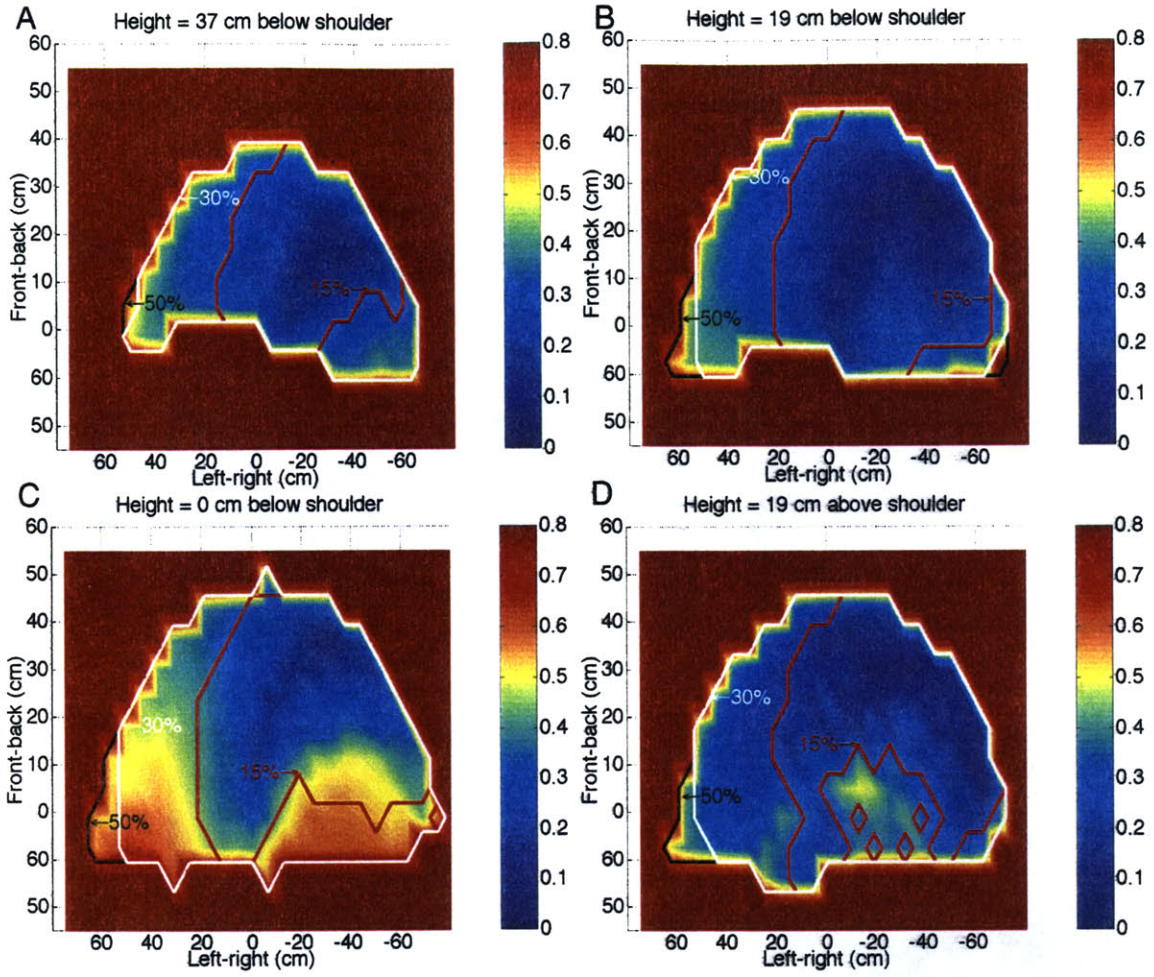


Figure 5-29. Reach difficulty metric and torque limit contours for horizontal slices through reach envelope, for male, 50th percentile size, 95th percentile strength. The origin of coordinates is at the right shoulder. Blue indicates easy to reach areas; red indicates difficult to reach areas.A.) 37 cm below shoulder. B.) 19 cm below shoulder. C.) at shoulder level. D.) 19 cm above shoulder.

5.4.4 Smoothed work envelope boundaries

Boundaries for the work envelope were set using the smoothing algorithm described in Section 5.3.5. Three-dimensional plots of the work envelope boundaries for both the female and male 50th percentile size, 95th percentile strength cases are shown in Figure 5-30 and Figure 5-31. The smoothing process increases the volume of the male reach envelope by 21% over the 15% torque limit contour and the female reach envelope by 35% over the 15% torque limit contour.

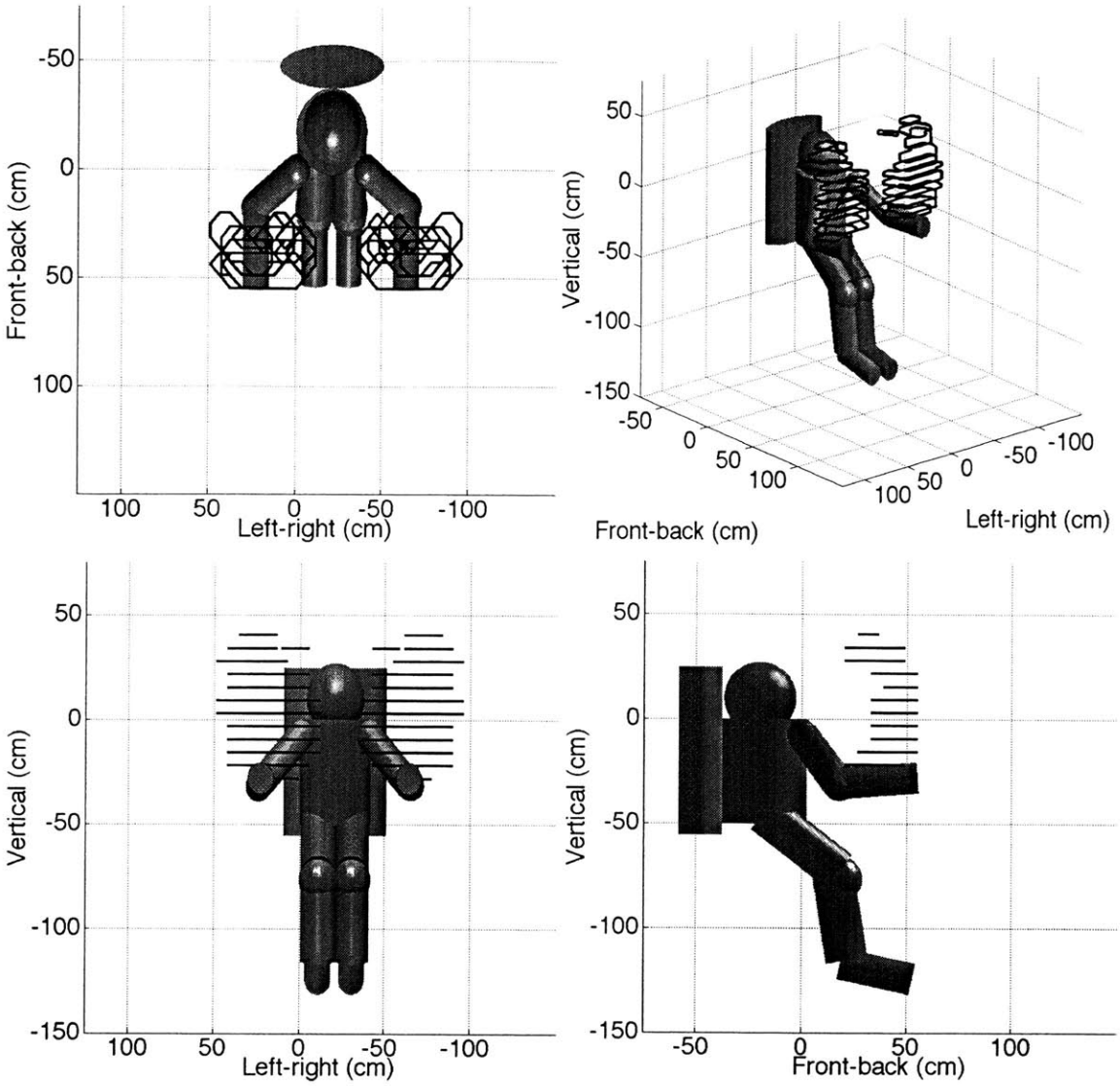


Figure 5-30. Smoothed work envelope boundaries for female, 50th percentile size, 95th percentile strength.

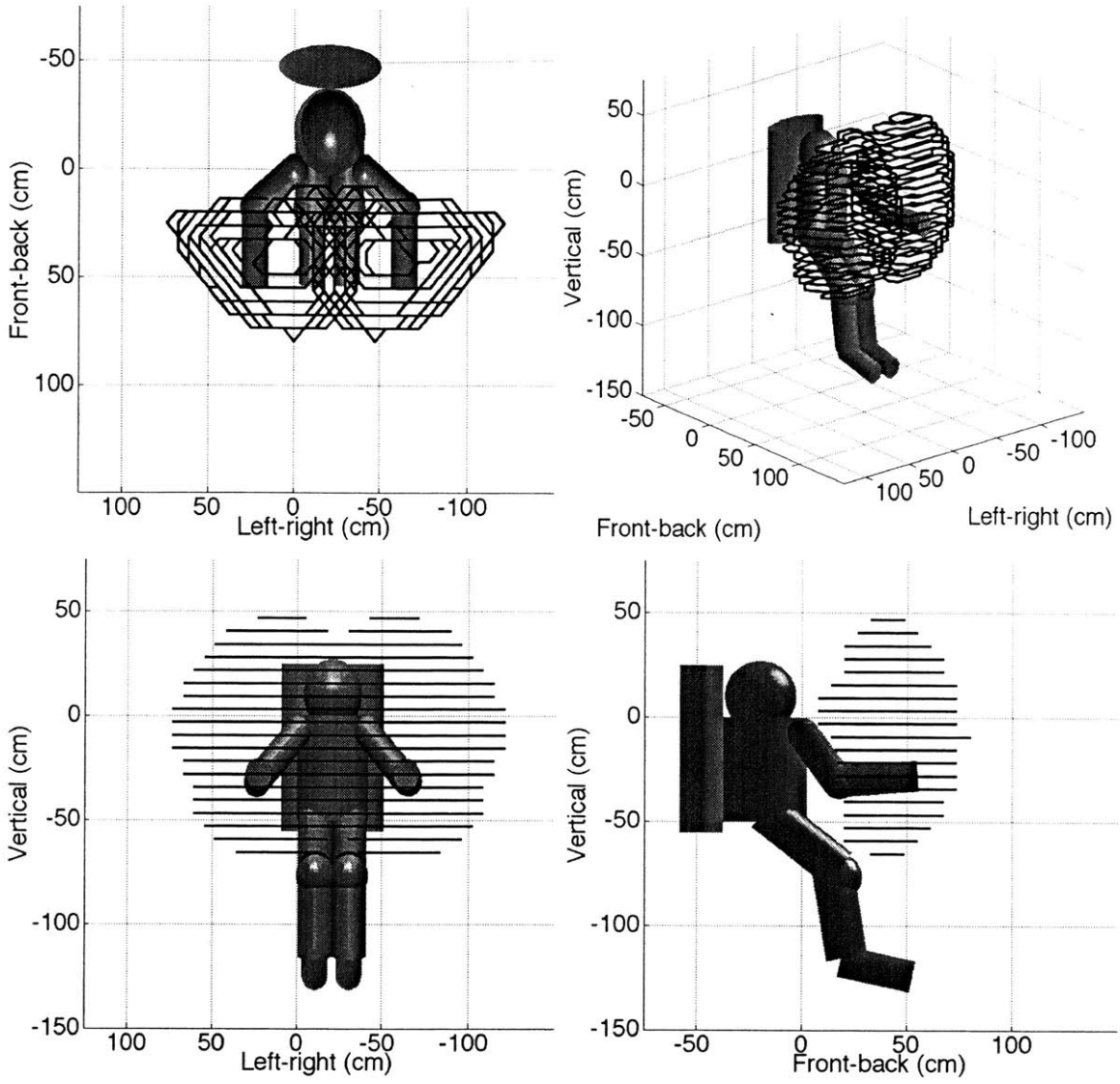


Figure 5-31. Smoothed work envelope boundaries for male, 50th percentile size, 95th percentile strength.

Horizontal slices through the work area, illustrating the reach difficulty metric and the smoothed work envelope boundaries are shown in Figure 5-32 for the female 50th percentile size, 95th percentile strength case and Figure 5-33, for the male, 50th percentile size, 95th percentile strength case. Figure 5-32 and Figure 5-33 can be compared directly to Figure 5-28 and Figure 5-29, which shows the reach difficulty metric and 15%, 30%, 50% and 100% torque limit contours for horizontal slices at the same heights.

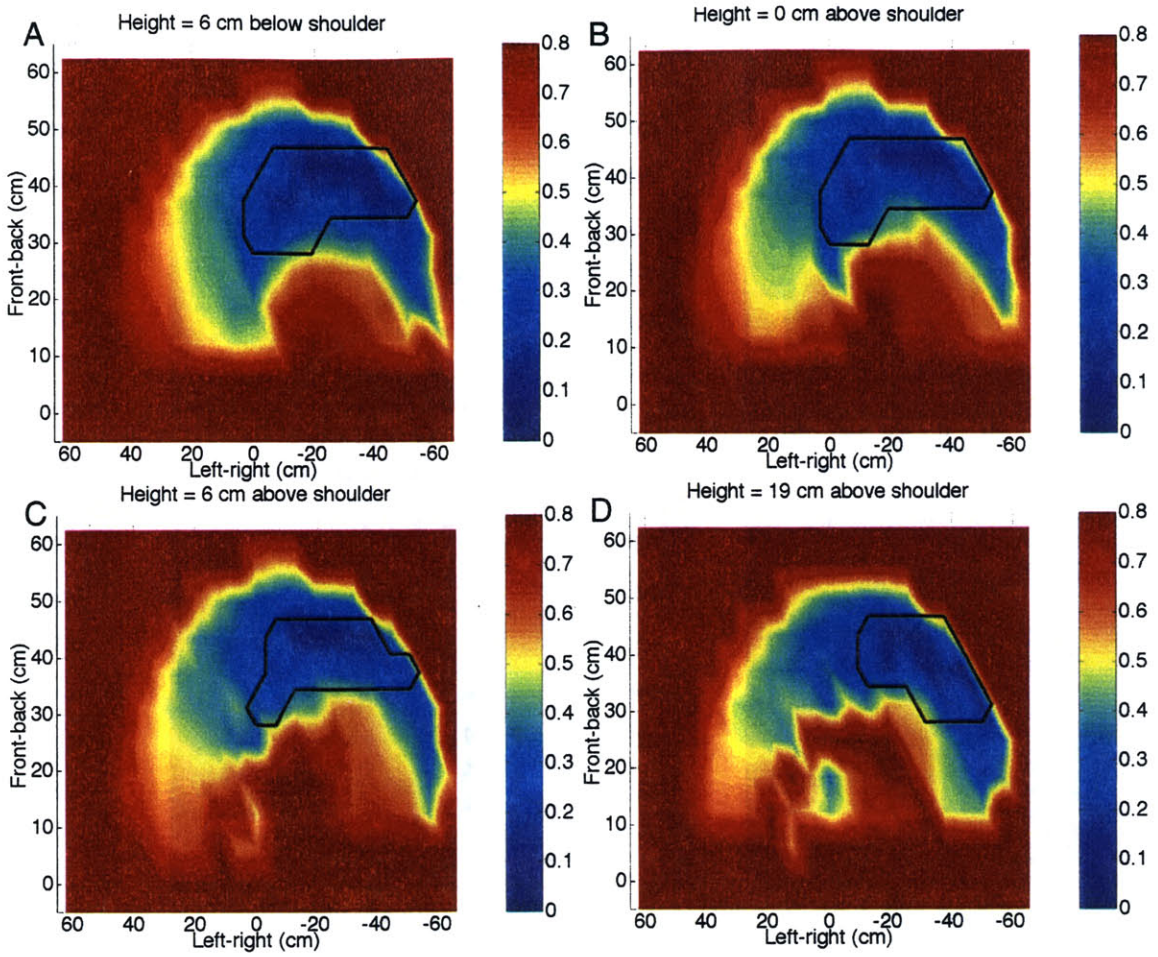


Figure 5-32. Reach difficulty metric shown by color and work envelope boundaries shown by black line for female, 50th percentile size and 95th percentile strength. A.) 6 cm below shoulder. B.) At shoulder level. C.) 6 cm above shoulder. D.) 19 cm above shoulder.

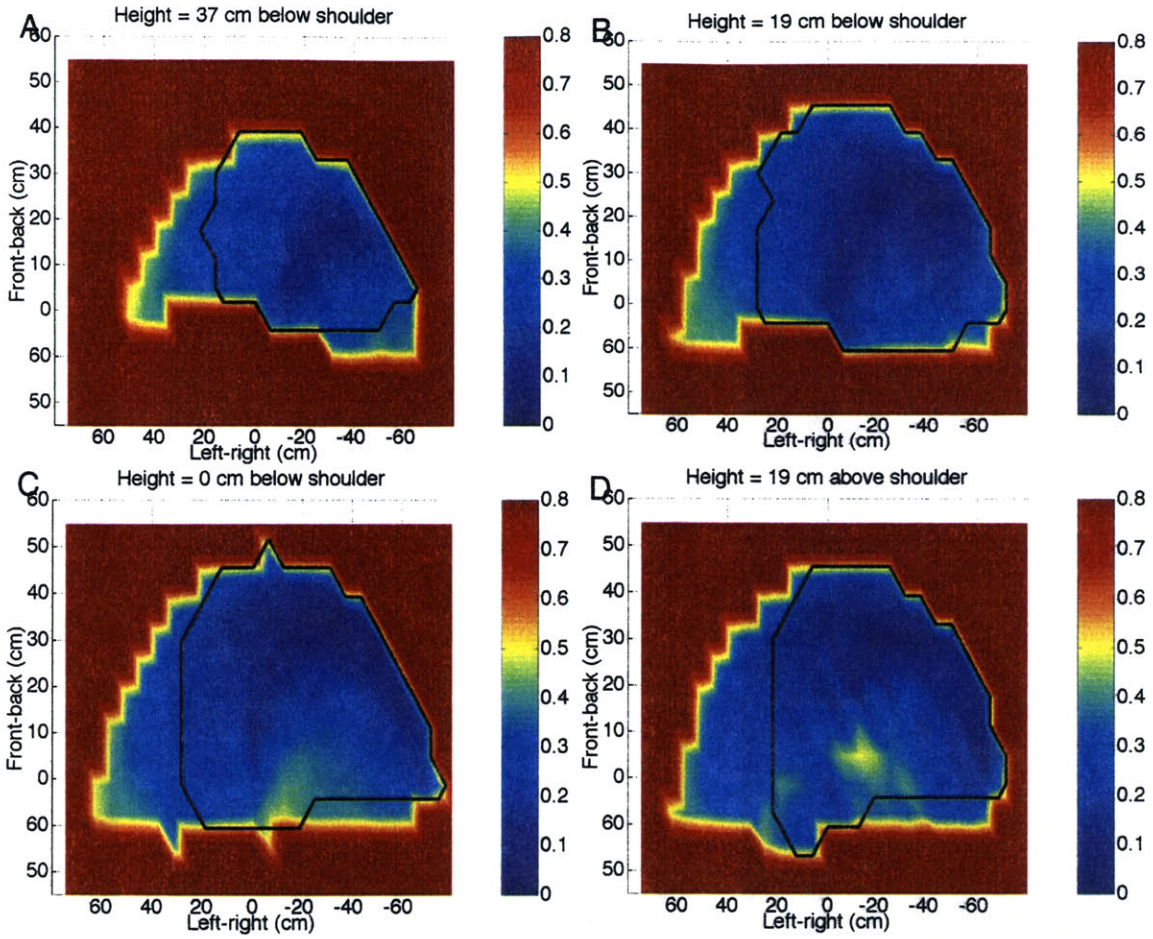


Figure 5-33. Reach difficulty metric shown by color and work envelope boundaries shown by black line for male, 50th percentile size and 95th percentile strength.A.) 37 cm below shoulder. B). 19 cm below shoulder. C.) At shoulder level. D.)19 cm above shoulder.

5.4.5 Work envelope volume

The volume enclosed within the work envelope boundaries serves as one global measure of work envelope size, for purposes of comparison. The eight cases tested in the work envelope analysis produced different volumes for the boundaries enclosing the 15%, 30%, 50% and 100% strength contours. Volumes enclosed by these boundaries are shown in Figure 5-34. Work envelope volume is shown in Figure 5-35, normalized by the volume of a hemisphere whose radius is equal to the arm length for the particular case tested.

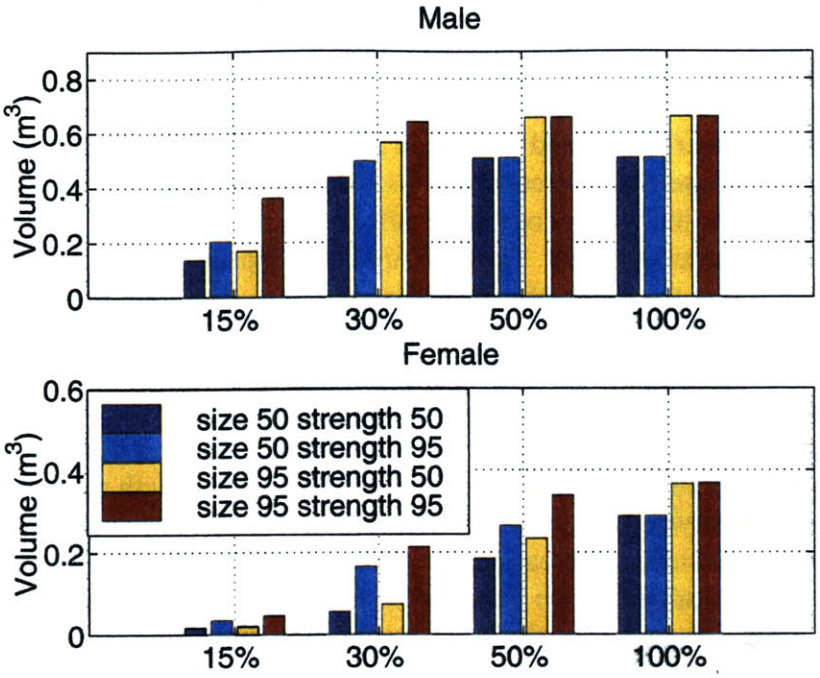


Figure 5-34. Volume enclosed within the 15%, 30%, 50%, and 100% torque limit boundaries for different anthropometric conditions.

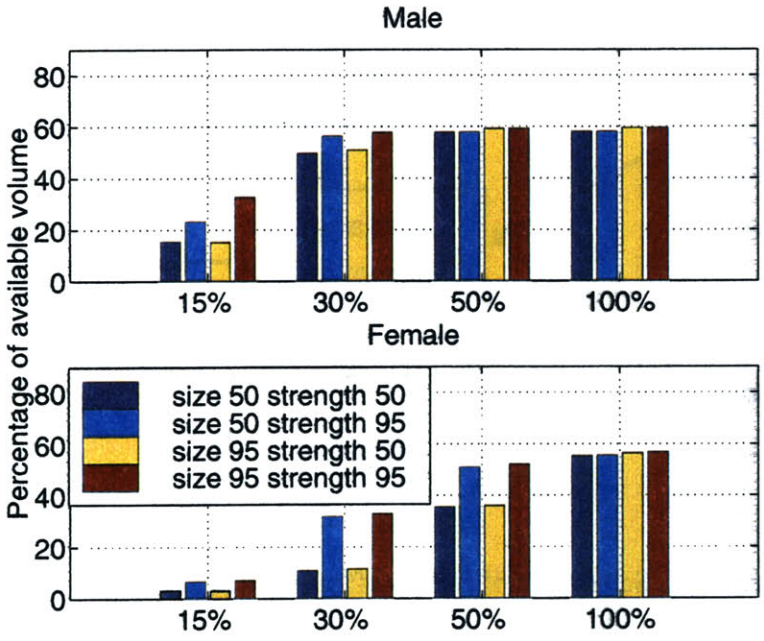


Figure 5-35. Volume enclosed within the 15%, 30%, 50%, and 100% torque limit boundaries for different anthropometric conditions, normalized by available volume.

5.5 Work envelope discussion

5.5.1 Factors affecting work envelope size and shape

All of the work envelope criteria listed in Section 5.2 contribute to the size and shape of the work envelope. Because multiple constraints set the boundaries of the work envelope, it is informative to determine which constraints affect the outcome of the work envelope analysis and which constraints do not. The combination of space suit joint stiffness and human strength limits sets the range of motion for each joint. The joint ranges of motion, combined with the lengths of the arm segments, determine the region that a person can reach. These relationships are shown schematically in Figure 5-36. In contrast to Figure 5-4, which illustrates the work envelope analysis technique and data flow, Figure 5-36 depicts the relationships between the applied constraints and work envelope analysis results. Using anthropometric data that represents the size and strength of real people makes the relationship between the work envelope criteria and the resulting work envelope more complicated. For instance, increasing a high, but realistic, torque limit may not increase the size of the work envelope if the joint can already traverse its full range with the lower torque limit. The results of the work envelope analysis illustrate how the astronaut's size, strength and visibility and the space suit's stiffness affect the size and shape of the work envelope.

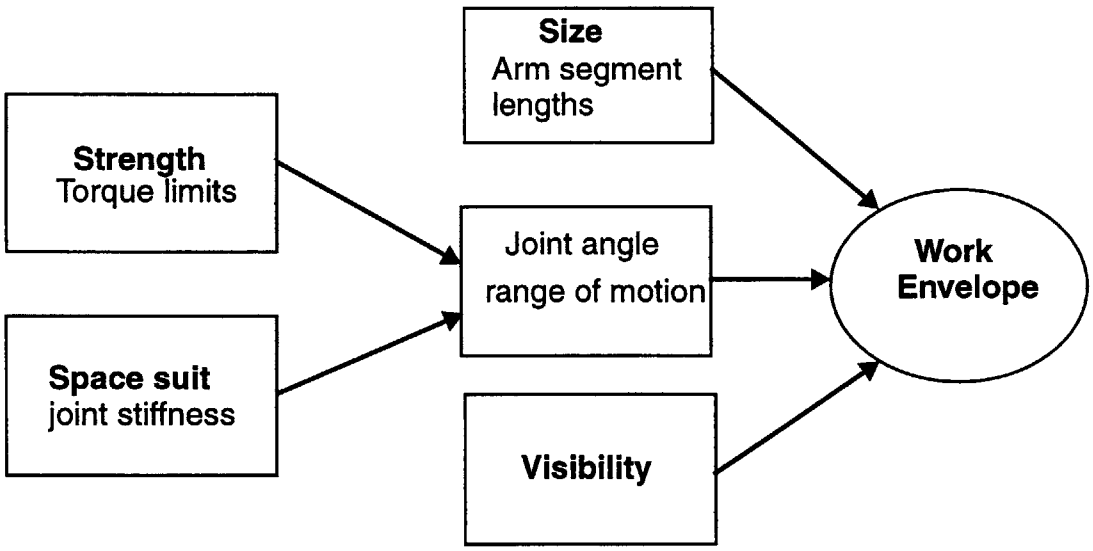


Figure 5-36. Schematic of factors affecting work envelope size

5.5.1.1 Human body size

Body size enters the inverse kinematics analysis through the lengths of the upper and lower arm segments. Total arm length ranges from 63 cm for the 50th percentile female to 81 cm for the 95th percentile male. The joint angle ranges of motion shown in Figure 5-16-Figure 5-15 indicate that body size had only a very small effect on the ranges of motion

of the joints. The small changes that are seen in joint ranges of motion between 50th percentile size and 95th percentile size people occur because the inverse kinematics analysis does not guarantee use of all of the available range of the joints. Because changing the arm segment lengths changes the required joint angles for a given hand position, it is expected that the joint angles output from the inverse kinematics analysis will be slightly different for different arm lengths.

The work envelope volume, plotted in Figure 5-34, and the percentage of available volume occupied by the work envelope, shown in Figure 5-35, give an overall picture of the work envelope size. At the 50% and 100% torque limits for the male anthropometric data and the female 95th percentile strength anthropometric data, the work envelope volume is determined solely by the arm length. The dominant effect of arm length on work envelope volume at the female 95th percentile strength and male 50% and 100% torque limits can be seen in Figure 5-35, where the percentage of available volume occupied by the work envelope is virtually the same in all cases. The uniformity of work envelope volumes for the male, 50% and 100% torque limits is explained by Figure 5-16-Figure 5-19, which indicate that the joint ranges of motion for 50% and 100% torque limits vary only slightly over all four male size and strength combinations. A different trend is seen for the female, 50th percentile strength, anthropometric data, where work envelope volume consistently increases as torque limits increase.

When the torque limits are high enough that the full ranges of the joints are available, the arm length determines the work envelope volume. This is the case for the 50% and 100% male torque limits and the 50% and 100% torque limits for female, 95th percentile strength, but it is not true for the female 50th percentile strength data, because the female 50th percentile strength torque limits are lower. Both arm length and strength limits affect the work envelope size for the female 50th percentile strength anthropometric data, at all torque limit levels.

5.5.1.2 Human strength limits and space suit joint stiffness

Human strength limits and the torques required to bend the space suit joints together determine the available range of motion of each joint. As the strength limits increase or the space suit joints requires less torque, the range of motion expands. Benefits associated with increasing torque limits or decreasing space suit stiffness are exhausted when the joints achieve their full range. When the full range of a joint is available, increasing torque limits further does not increase the size of the work envelope, and the work envelope size and shape are limited by the visibility constraint and arm segment lengths. Changes in work envelope volume due to changes in torque limit levels (15%, 30%, 50% or 100%) and strength percentile values (50th percentile or 95th percentile) illustrate the effects of across-the-board reductions in space suit stiffness, while a more detailed analysis of the effects of increasing individual joint torque limits shows the effects of improving mobility in single joints.

Figure 5-34 shows the volume occupied by the work envelope for 15%, 30%, 50% and 100% torque limits. For the male, 50th percentile size, 95th percentile strength case, increasing the torque limits from 15% to 30% results in an increase in work envelope vol-

ume of 270%, while increasing the torque limits further to 50% results in an additional gain in work envelope volume of only 5%. Therefore, reducing the stiffness of all space suit joints by a factor of two results in a large increase in work envelope volume, while further reductions in space suit stiffness produce little gain.

Looking at the effects of increasing torque limits for individual joints indicates which joints limit the overall size of the work envelope and where each joint's limits bound the work envelope. Individual joints were analyzed using the female, 50th percentile size, 95th percentile strength case. The baseline case uses torque limits of 30% of the maximum strength for all joints. Six variations on the baseline case were analyzed, varying the positive and negative torque limits on the shoulder flexion, shoulder abduction, and elbow flexion joints. The humerus rotation joint was not considered because the space suit's arm bearing allows it to traverse its full unsuited range of motion without reaching even 15% torque limits. In each variation evaluated in the sensitivity analysis, the positive or negative torque limit for one joint was increased from 30% of the strength limit to 50% of the strength limit. The work envelope volume in the baseline case and the six variations, normalized by the volume of a hemisphere with radius equal to the total arm length, are shown in Figure 5-37. The largest gain in work envelope volume is associated with an increase in the negative shoulder abduction torque limit. Increasing the negative shoulder flexion torque limit also increases the work envelope volume somewhat, while positive shoulder flexion and positive and negative elbow flexion do not limit the work envelope volume.

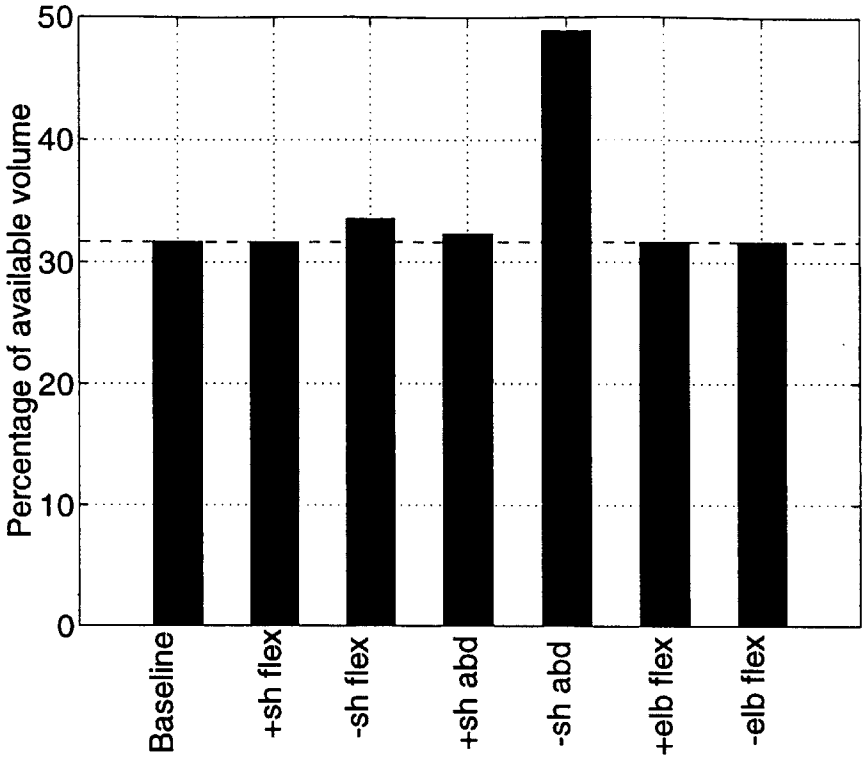


Figure 5-37. Percentage of available volume occupied by work envelope when torque limits are increased from 30% to 50% for single joints. The baseline case uses torque limits of 30% of the strength limits for all joints.

Increasing joint torque limits individually also demonstrates which joints set which portions of the work envelope boundaries. Figure 5-38 shows horizontal slices through the work envelope at four heights, with the reach difficulty metric indicated by color and work envelope boundaries shown for two cases: the baseline case and the higher 50% torque limit on negative shoulder flexion with 30% torque limits on all other joints. Increasing the negative shoulder abduction limit adds to the work envelope close to the body centerline, at all heights.

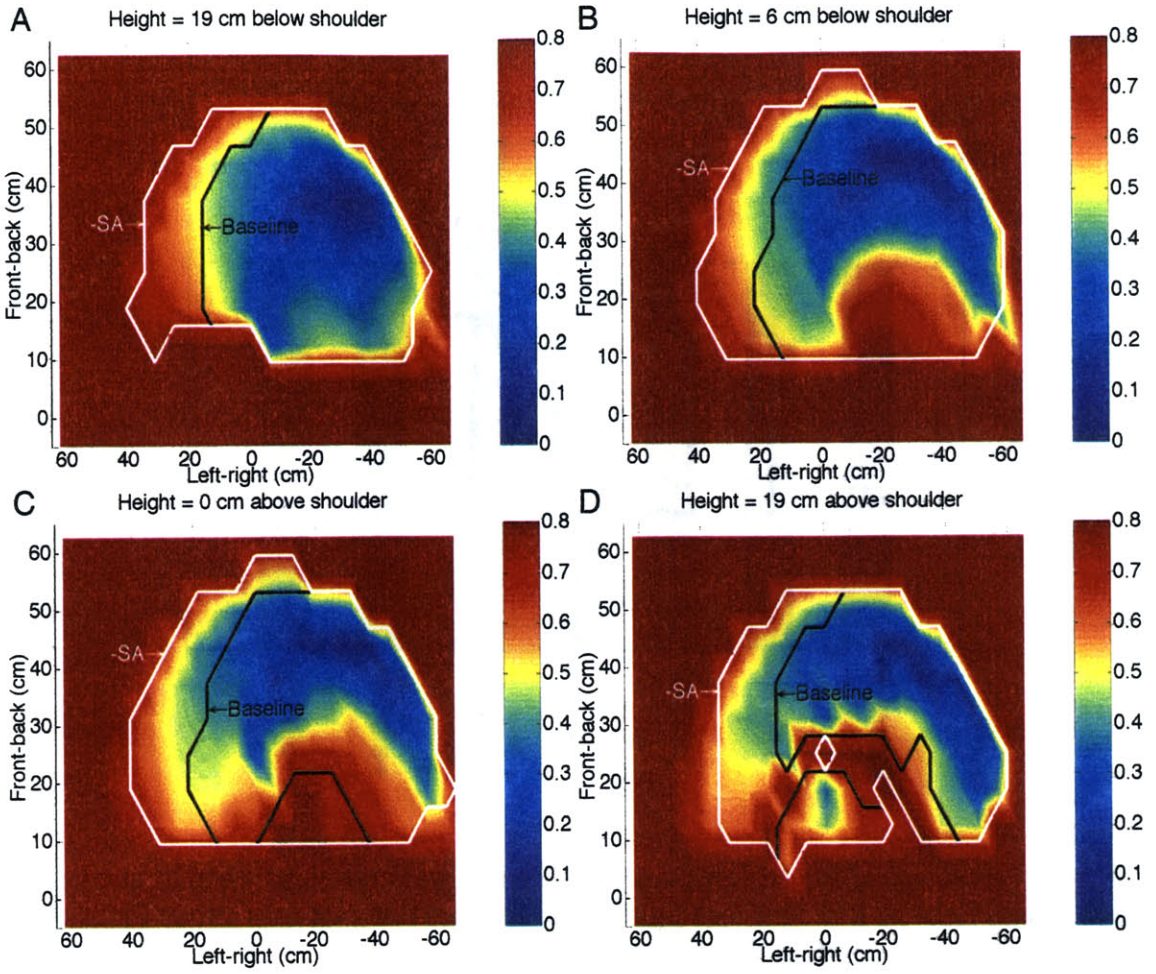


Figure 5-38. Reach difficulty metric and torque limit contours for horizontal slices through reach envelope, comparing baseline 30% torque limits for all joints to cases where the negative shoulder abduction torque limit is individually raised to 50%. A.) 19 cm below shoulder. B.) 6 cm below shoulder. C.) At shoulder level. D.) 19 cm above shoulder.

5.5.1.3 Visibility

The visibility constraint excludes areas close to the body, at the top and bottom of the work space, from the work envelope. The 30% torque contours calculated using field of view restrictions were compared to boundaries calculated using the same torque limits but no field of view restrictions for male 50th percentile size, 95th percentile strength and female 50th percentile size,95th percentile strength. Slices through the work envelopes at different heights are shown in Figure 5-39 for male data and Figure 5-40 for female data, showing the 30% torque contours when the field of view is restricted to the visibility limits described in Section 5.2.2 and when the field of view is not restricted.

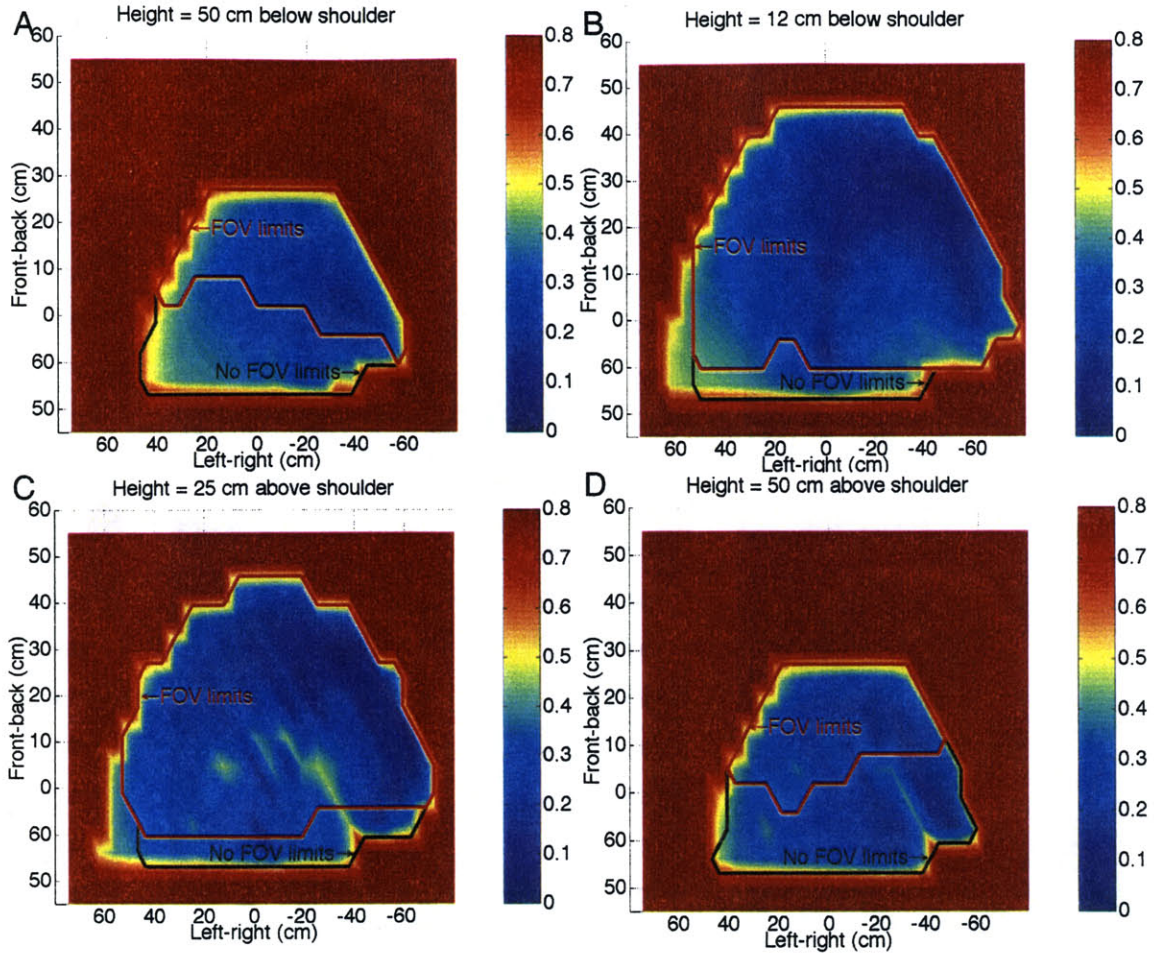


Figure 5-39. Comparison of 30% torque contours with and without visibility limits, for male, 50th percentile size, 95th percentile strength. A.) 50 cm below shoulder. B.) 12 cm below shoulder. C.) 25 cm above shoulder. D.) 50 cm above shoulder.

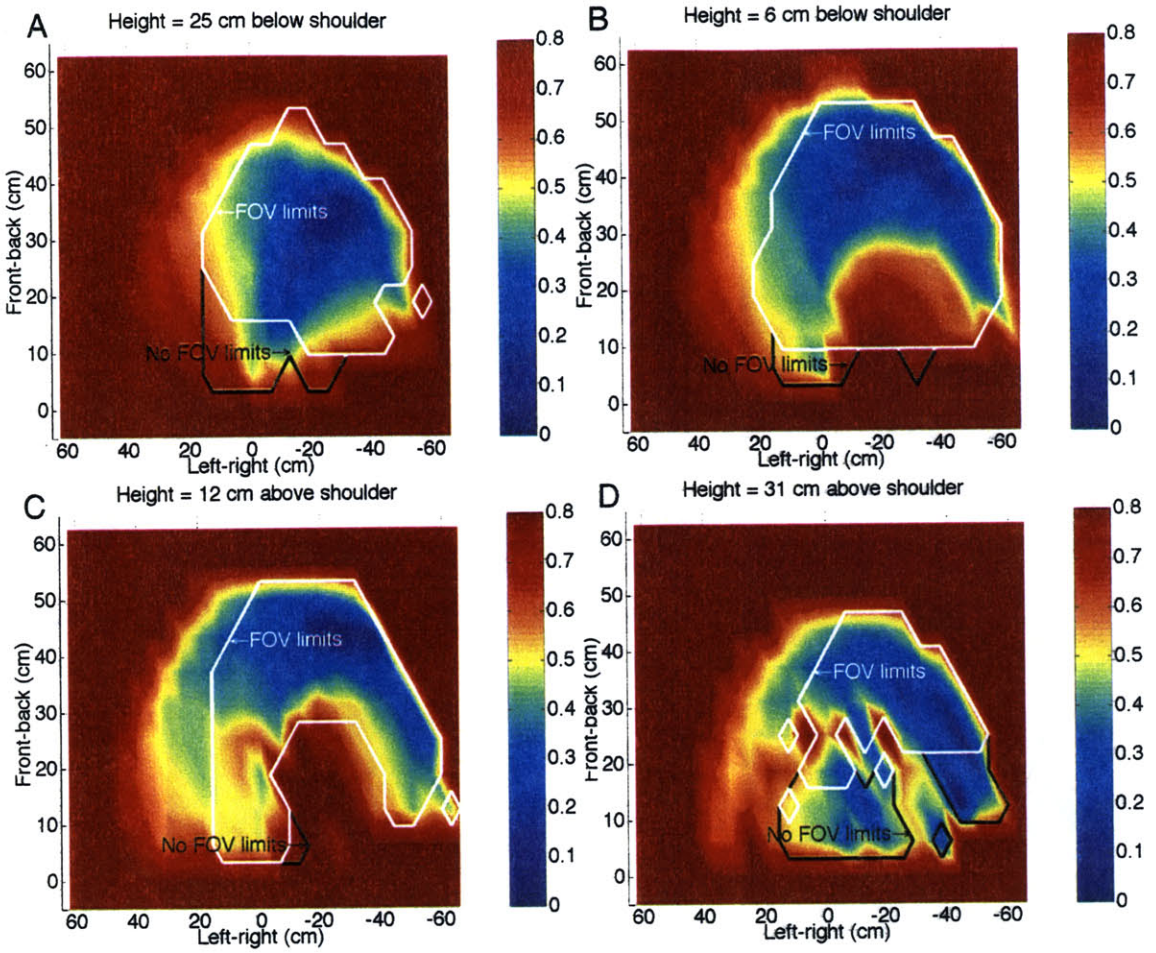


Figure 5-40. Comparison of 30% torque contours with and without visibility limits, for female, 50th percentile size, 95th percentile strength. A.) 12 cm below shoulder. B.) 19 cm above shoulder. C.) 31 cm above shoulder. D.) 44 cm above shoulder.

As shown in Figure 5-39 and Figure 5-40, visibility limits the field of view close to the body. This occurs at both the top and bottom for both the male 50th percentile size, 95th percentile strength case and the female 50th percentile size, 95th percentile strength case. Removing the visibility constraint increases the volume occupied by the work envelope by 33% in the male case studied and 22% in the female case.

5.5.2 Comparison to NASA work envelope

5.5.2.1 Agreement with the NASA work envelope

The NASA work envelope, which is a cylindrical region, centered on the body centerline at the height of the shoulders, is described in detail in Section 2.5.2. A comparison between the smoothed work envelope boundaries for the male 50th percentile size, 95th percentile strength case and the NASA work envelope is shown in fig.

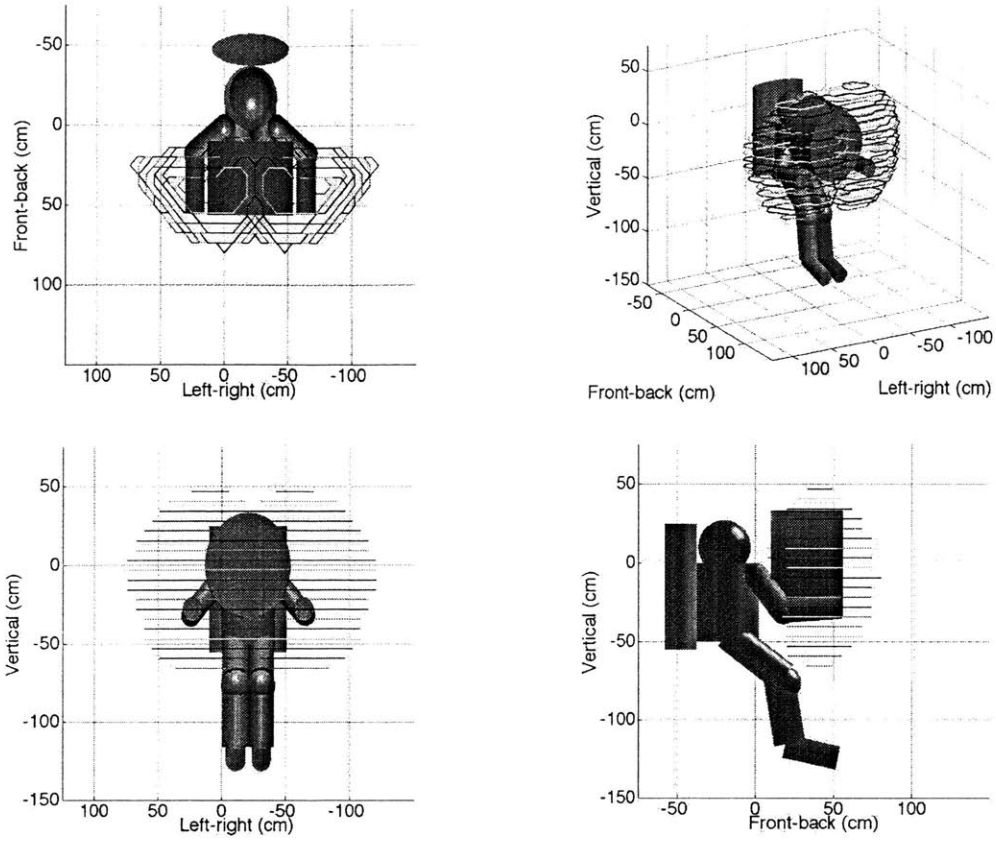


Figure 5-41. Comparison between calculated work envelope boundaries and cylindrical NASA work envelope.

A detailed comparison between the NASA work envelope, the smoothed work envelope boundaries and the 15% and 30% torque limit boundaries calculated for the male 50th percentile size, 95th percentile strength illustrates the extent of agreement between the NASA work envelope and the calculated work envelopes. Table 5-6 shows the volume occupied by the male 50th percentile size, 95th percentile strength work envelope, the volume occupied by the right half of the NASA work envelope, and the volume that is included in both the calculated and NASA work envelopes, for 15% and 30% torque limits and the smoothed work envelope boundaries. Only the right half of the NASA work envelope is considered, because the calculated work envelopes only consider the right arm. Although the calculated work envelope does extend into the left half of the NASA work envelope,

considering these areas in the comparison would count them twice when the left arm's work envelope is considered.

Table 5-6: Work envelope comparison

	15% torque limits	30% torque limits	Smoothed work envelope boundaries
Calculated work envelope volume (m^3)	0.2835	0.4940	0.3552
NASA work envelope volume (m^3)	0.0194	0.0194	0.0194
Volume in both (m^3)	0.0083	0.0146	0.0136

The NASA work envelope is smaller than the calculated work envelopes, although it is not simply a subset of the calculated work envelopes. In spite of the discrepancies between the three sets of work envelope boundaries, 70% of the volume included in the NASA work envelope is also included in the smoothed work envelope. Figure 5-25 and Table 5-6 both show that the NASA work envelope is considerably smaller than the calculated work envelopes, representing 7% of the volume of the 15% torque limit contours, 4% of the volume of the smoothed work envelope boundaries, and 5% of the volume of the 30% torque contours. A sizable region on the side away from the body centerline is included in the calculated work envelopes, but not the NASA work envelope, as illustrated in Figure 5-24-Figure 5-23. Despite its much smaller size, the NASA work envelope still contains regions that are not included in the calculated work envelopes, mainly because the visibility constraint is applied to the work envelope calculations but not to the NASA work envelope. In addition to the areas excluded from the calculated work envelopes because of the visibility constraints, a region at the body centerline, near the forward edge of the NASA work envelope is excluded from the calculated work envelopes, illustrated in the overhead views in Figure 5-25 and Figure 5-31. The area near the body centerline is excluded because of the shoulder abduction torque limits, as discussed in Section 5.5.1.2.

5.5.2.2 Visibility and the NASA work envelope

The NASA work envelope is based solely on acceptable work areas determined in neutral buoyancy simulation and does not consider the visibility limitations imposed by the space suit's helmet and the DCM. An analysis of the visibility of areas within the NASA work envelope indicates that a portion of the region inside the NASA work envelope, at the bottom, close to the body, is not visible. Figure 5-42A shows the field of view boundaries and the NASA work envelope boundaries at the inner edge of the NASA work envelope. At the inner edge, 13 cm in front of the shoulders, more than half of the area of the work enve-

lope is not visible, illustrated in Figure 5-42A. 11 cm further forward, the visible areas is greatly expanded to include the entire NASA work envelope, as shown in Figure 5-42B.

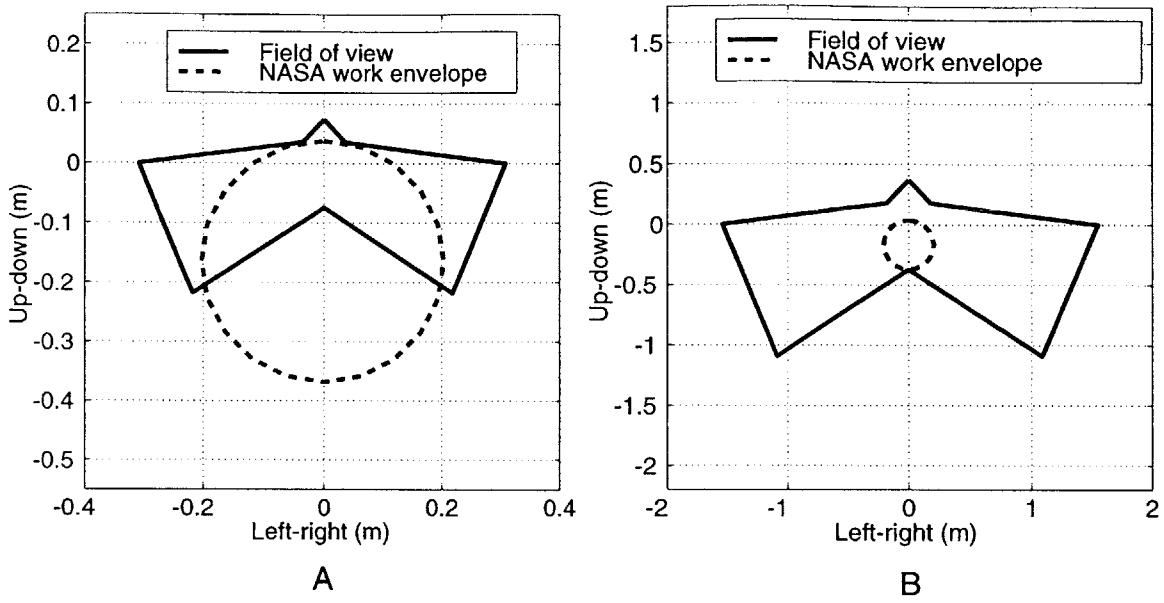


Figure 5-42. NASA work envelope and field of view. A.) Inner edge of work envelope, 13 cm in front of the shoulders. B.) 23.5 cm in front of the shoulders.

Although there may be situations in which it is not essential to see the entire work area, for instance, grasping an object and maintaining a grip on it while moving to a different position where it is out of view, the visibility limitations should be considered when using the NASA work envelope for EVA planning purposes.

5.5.3 Limitations

5.5.3.1 Use of anthropometric data

The work envelope analysis described in this chapter uses data reported from anthropometric studies to supply values for arm segment lengths, offsets between the shoulder and eyepoint, and maximum strength values. Anthropometric data is essential to the work envelope analysis because it supplies realistic parameter values for size and strength. Because the inverse kinematics calculations are inherently nonlinear and multiple constraints are applied, it is not possible to predict trends or generate meaningful results without reasonably accurate parameter values. One limitation of this study is the use of several anthropometric data measures in combination, which, in some cases, limits the statistical interpretations of the results. Several pieces of anthropometric data, such as arm segment lengths and individual joint strengths, are combined for each case listed in Table 5-5 and tested in the work envelope analysis. When several pieces of anthropometric data of the same percentile value are combined, the resulting configuration is considerably rarer than the percentile value for a single data element, particularly when the percentile values are near the extremes. Although the different anthropometric measures are not independent, the portion of the population that has several body dimensions greater than or equal to the

95th percentile values is substantially less than 5%. The use of anthropometric data adds value to this study because it supplies realistic parameter ranges for sizes and strengths. Because several pieces of anthropometric data are combined to form each case tested, statistical properties should not be inferred from the work envelope results. In other words, it is not correct to consider the work envelope outputs 50th percentile work envelopes or 95th percentile work envelopes because the statistical meaning of the percentile values is distorted when the anthropometric data measures are combined.

5.5.3.2 Extrapolation of space suit joint angle-torque data

The space suit joint angle-torque data that was the basis of the space suit joint stiffness models was obtained from the experiments described in Chapter 3. Because it is not possible to measure joint torques in humans without restricting their motion, joint torques were measured using an instrumented robot that was wearing the space suit. Although the robot provides a good approximation to realistic human motion, there are some human shoulder joint positions that the robot is unable to reproduce. These include extending the upper arm back, behind the body and forward, towards the body centerline. Consequently, in order to represent realistic human shoulder positions, the space suit joint torque-angle models were applied to joint angles outside of the range for which torque data exists. Extrapolation beyond the range of the experimental torque-angle data was unavoidable because restricting the inverse kinematics model to the range of motion of the robot's arm would result in an unrealistic representation of human arm mobility.

5.5.3.3 Choice of arm joints

A simplified model of the human arm was used in this analysis, considering the shoulder and elbow joints, but not the wrist. This choice was made primarily because torque-angle data for the wrist was not available. The lack of a wrist joint in the modeled arm means that hand orientation is not considered in the work envelope analysis. Hand orientation would be an interesting element to include in a work envelope analysis, particularly in answering focused questions about specific tasks, such as the feasibility of reaching around obstacles, grasping objects, or determining the maximum forces and torques that could be exerted by the space suited astronaut at a specific location within the workspace. The choice of arm model used in this analysis, which only captures hand location, generates results that are useful in understanding the overall size and shape of a space suited astronaut's workspace.

5.5.3.4 Verification

Verification of the predictions made in the work envelope analysis would be a necessary next step in implementing the calculated work envelopes for EVA planning and workstation design. Experiments in neutral buoyancy simulation to determine work envelope boundaries for individuals of different sizes and strengths would be an interesting area for future research.

5.6 Conclusions

The work envelope analysis demonstrates the practical utility of modeling the space suit mechanics on a fundamental level, yielding both operational and design recommendations. Using models of the torque-angle relations of the space suit joints and space suit kinematics, data on visibility in the space suit, and a spatial smoothing algorithm, a rapid, systematic, and customizable method was developed to determine work envelope boundaries for a space-suited astronaut.The method was used to calculate work envelope boundaries for two cases, using both male and female anthropometric data. In addition to boundaries on suitable locations of work areas, the work envelope analysis also indicates the relative difficulty of holding the hand in different locations.

A sensitivity analysis of the work envelope boundaries, which varied joint torque limits, arm length, and visibility limits, yielded several space suit design recommendations. Increasing mobility of the space suit's shoulder joint to allow the arm to be brought closer to the body centerline in front of the body results in the largest gains in work envelope size. An across-the board reduction in space suit joint stiffness by a factor of 2 results in large gains in work envelope size, while further reductions in space suit joint stiffness only yield small gains. Visibility improvements in the upward and downward directions increase the work envelope size, while visibility does not limit work envelope size in the left and right directions.

Chapter 6 *Conclusions*

The primary aim of this thesis is to advance the current understanding of astronauts' capabilities and limitations in space-suited EVA by compiling a detailed database of the torques needed to bend the joints of a space suit, developing models of the mechanics of space suit joints based on experimental data, and utilizing these models to estimate a human factors performance metric, the work envelope for space suited EVA work.

6.1 Space suit mobility database

The space suit torque-angle database was collected using an Extravehicular Mobility Unit space suit, in a novel measurement approach which used both human subjects and an instrumented robot to collect joint angle and joint torque data for realistic human motions. Human test subjects carried out arm and leg motions both wearing the space suit and not wearing the space suit, supplying realistic joint angle trajectories for each of 20 motions. The joint angle trajectories produced by the human subjects were then used as command inputs for the instrumented robot, so that the robot imitated the humans' motions while torques were measured at each of the robot's joints. Torques on the robot's joints due to the weight of the robot and space suit were subtracted from the torques that the robot measured, resulting in a consistent set of joint angle and torque data.

The database compiled in this work is more extensive than any other published space suit torque-angle data set, covering 11 joints over a large range of angles, including multiple degrees of freedom at the shoulder, hip, and ankle. Realistic, three-dimensional human-generated motions were used for data collection. Because torque data was collected by the robot as a surrogate for a human occupant of the space suit, the torques measured in this study are more representative of realistic conditions than data from previous studies[3, 15, 40] that measured joint stiffnesses for empty, pressurized space suits. A limited comparison of torques required to bend the elbow and knee to two angles, shown in Table 6-1, illustrates that the torques measured in this study are higher than the torques measured with empty

space suits. The space suit torque-angle database serves as a basis for developing and validating both mathematical and physical models of space suit joint mobility characteristics.

Table 6-1: Space suit joint torques

Study	Dionne	Menendez et al	Abramov et al.	Morgan et al.	Current Study
Methods	EMU empty suit	Prototype joint segments, empty	Orlan-DMA, 4.3 psi (30 KPa) empty suit	EMU human subjects	EMU, human subjects and robot
Knee, 72 deg.	3.2 Nm	NA	6.0 Nm	8.1 Nm	3.74±.0676 Nm
Elbow, 80 deg	2.0 Nm	2 Nm	2.2 Nm	3.4 Nm	14.6±.136 Nm

6.2 Modeling

The space suit torque-angle database presents a rare opportunity to develop models of space suit mobility and verify them against experimental data. Models of space suit mobility are useful in two applications: first, numerically predicting the torque required to bend a space suit's joints and second, understanding the physical processes that determine the stiffness of a space suit's joints. These two applications clearly require two different modeling approaches: a descriptive mathematical modeling technique based on experimental data and a theoretical model based on physical principles. Chapter 4 describes two modeling efforts: a mathematical model based on empirical data that predicts the torque required to bend the space suit's joints and a comparison between two physics-based models of bending pressurized cylinders and the space suit torque angle database.

Numerical predictions of the torque required to bend space suit joints were generated using a mathematical hysteresis modeling technique. The experimental data reported in Chapter 3, as well as other investigators' data, indicate that the torque-angle relationship for space suit joints is markedly hysteretic. The Preisach hysteresis model, which has been used to model other hysteretic mechanical systems, is used to mathematically model the torque required to bend the space suit joints as a function of bending angle history. The Preisach model coefficients were identified for the torque-angle characteristic of six of the space suit's joints: elbow flexion, hip flexion, hip abduction, knee flexion, ankle rotation and ankle flexion. In addition, a new method was developed for estimating the variance of the error in the Preisach model's torque predictions. Comparisons between the Preisach model's output and experimental data indicate that the Preisach hysteresis model accurately reproduces torques needed to bend space suit joints.

The physics-based modeling effort compares two approximate models of pressurized fabric cylinder bending to experimental data. The two models, the beam model and the membrane model represent limiting cases on the physical processes that govern space suit joint mobility. The beam model assumes that space suit joint bending occurs through elastic deformations of the space suit fabric, with no change in internal volume when joints are

bent. In contrast, the membrane model assumes that the space suit fabric never stretches, and the work done to bend the space suit joint goes entirely into compressing the gas inside the space suit. Comparing these two approximate models to experimental data illustrates the relative importance of elasticity and volume loss on space suit joint mechanics and indicates whether the space suit behavior can be approximated by one model or the other. Understanding the physical processes that govern space suit joint performance leads to insights into how space suit joints would perform if their construction or operating conditions were changed.

Comparisons between the two models and experimental data show that the membrane model matches the elbow and knee joint torque-angle data well over a bending angle range 30-50 degrees from the equilibrium bending angle and underestimates the torque required to bend the space suit joints for more extreme bending angles. The beam model is inconsistent with both the magnitude and curve shape of the elbow and knee joint torque-angle data. Data from the hip abduction joint, which has no mobility features and is consequently very stiff, is consistent with the beam model, although the limited angle range of the data prevents the beam model from showing important qualitative characteristics.

Because the torque-angle performance of the space suit joints can be accounted for by changes in internal volume without stretching of the space suit fabric, the mobility of the EMU elbow and knee joints is not sensitive to changes in the material properties of the fabric. The membrane model predicts that the design parameters that affect space suit mobility are the internal pressure, radius of the space suit segment, and the pleat design that allows the joint to be bent without reducing the internal volume.

The physics-based modeling work in this thesis provides insights into the physical processes that govern space suit mobility, points out promising methods of improving space suit mobility and provides a theoretical basis for further experimentation to develop more sophisticated physics-based models of space suit mobility.

6.3 The work envelope: Applying space suit modeling to EVA operations

The work envelope analysis demonstrates the practical utility of modeling the space suit mechanics on a fundamental level, yielding both operational and design recommendations. Using models of the torque-angle relations of the space suit joints and space suit kinematics, data on visibility in the space suit, and a spatial smoothing algorithm, a rapid, systematic, and customizable method was developed to determine work envelope boundaries for a space-suited astronaut.The method was used to calculate work envelope boundaries for two cases, using both male and female anthropometric data. In addition to boundaries on suitable locations of work areas, the work envelope analysis also indicates which areas are best for workstation placement.

A sensitivity analysis of the work envelope boundaries, which varied joint torque limits, arm length, and visibility limits, yielded several space suit design recommendations. Increasing mobility of the space suit's shoulder joint to allow the arm to be brought closer to the body centerline in front of the body results in the largest gains in work envelope size. An across-the board reduction in space suit joint stiffness by a factor of 2 results in

large gains in work envelope size, while further reductions in space suit joint stiffness only yield small gains. Visibility improvements in the upward and downward directions increase the work envelope size, while visibility does not limit work envelope size in the left and right directions.

6.4 Recommendations for future work

6.4.1 Experimental work

Joints in the space suit include both bending joints and rotational bearing joints, to allow motion in multiple degrees of freedom in the arm. While this thesis focuses mostly on the performance of the bending joints, experimental work to quantify the motion of the space suit shoulder segment between the two bearings in the arm, as distinguished from the joint motion of the person wearing the space suit, would lead to a better understanding of the multiple degree of freedom motion at the shoulder joint.

Future experimental work would also contribute to the physics-based modeling efforts. The membrane model, which agrees closely with experimental data for the elbow and knee, predicts that joint stiffness increases linearly with space suit pressure. This result could easily be tested by measuring joint torque vs. angle at a range of pressures. Other predictions that the membrane model makes, including inextensibility, the amount of internal volume change, and the bent shape of the space suit joint could be tested in experiments with pressurized mockups of individual space suit joints.

The work envelope analysis makes detailed predictions of easily-reachable locations based on body size and strength. For these predictions to have practical value, the work envelope analysis should be experimentally validated. Experimental work to validate the work envelope predictions should use arm length, viewpoint location, and static joint strength data from the test subjects to predict work envelopes tailored to those individuals. Validation of the calculated work envelope should be performed in neutral buoyancy simulation, rather than a 1-g lab environment, so that the test subjects are not required to supply additional joint torques to support the weight of the space suit's arm. Test subjects would attempt to hold one hand at pre-determined locations for several minutes and subjectively rate the difficulty of holding the hand position, as well as indicating whether the point is visible. Testing points should be chosen to fall both inside and outside of the calculated work envelope, with particular attention paid to points that are inside the calculated work envelope but outside of the NASA work envelope. Data collected should include whether the subject was successful in holding the hand at the test point for the required duration, the length of time that the test subject was able to maintain the required hand position, the subjective difficulty ratings and whether the point was visible. This information should then be compared to the calculated work envelope's boundaries, the reach difficulty metric, and the visibility predictions. Modifications to the computational work envelope based on these evaluations may include changes to the percentages of maximum torque that are used as torque limits and more detailed visibility information.

6.4.2 Analysis

Future analytical work should include implementation of existing space suit models and development of new models, based on experimental data. Rahn[45, 42] demonstrated in a simulation study that torques induced by the space suit have an important impact on body positions and loads when astronauts move large objects in EVA. Because of Rahn's result and the availability of the hysteresis models developed in this work, future EVA dynamic simulation studies should include the torques needed to bend the space suit joints.

New, physics-based models of space suit joint mobility should address joint design issues by including details of space suit joint construction. Work should also be done to asses the proper way of relaxing the inextensibility constraint of the membrane model to include the limited fabric stretching that likely occurs when space suit joints are bent. Hysteresis in fabrics is generally thought to be caused by the fabric threads sliding over each other as the fabric stretches[27]. Although the space suit joints show significant hysteresis, implying that there is some internal relative motion in the fabric, the membrane model's agreement with experimental data implies the contrary, that the space suit fabric is inextensible. It would be particularly interesting to account for the hysteresis mechanism in light of this apparent contradiction.

6.4.3 EVA standards and requirements

The NSTS-07700 EVA requirements document[1] specifies NASA's requirements for the space suited work envelope (Figure 2-16) and the field of view for an EVA astronaut (Figure 5-2). The current requirements are simply stated, requiring 4 numbers for the work envelope (3 coordinates of cylinder center location and cylinder diameter) and 5 numbers for the field of view (angular field of view in 5 directions). The simple requirements are easy to understand and visualize, lending themselves to qualitative evaluation and application to worksite design. However, much more powerful methods, including CAD modeling, are currently available for EVA worksite analysis and have the capability to enforce complicated requirements quantitatively.[4, 25] Now that advanced techniques exist for evaluating work envelope and visibility requirements, the work envelope and visibility standards should be updated to a higher accuracy and level of detail.

The work envelope standard in NSTS-07700 is significantly smaller than the work envelope calculated based on space suit mobility and human strength limits. In particular, there are large areas beyond the left and right boundaries of the NSTS-07700 work envelope that are inside the calculated work envelope. Also, a visibility analysis of the NSTS-07700 work envelope using the visibility requirements from the same standards document indicates that a portion of the NSTS-0770 work envelope is not visible. These disagreements should spur a re-evaluation of the NSTS-07700 work envelope, including visibility, body size differences, and, at the least, enlarging its left-right dimension.

The visibility standard in NSTS-07700 specifies the angular field of view in 5 directions. The lack of spatial resolution prevents this guideline from accurately specifying the visi-

bility limitations caused by the shape of the helmet and visors and the geometry of other obstructions, including the space suit's Display and Control Module (DCM), and the mini-workstation, which both block downward visibility. Since powerful analytical techniques currently exist for assessing adherence to these requirements, the visibility requirement should be updated to improve its accuracy and make full use of the available analytical techniques for assessing compliance to requirements.

6.4.4 Applications to current and advanced EVA

The analytical techniques and space suit mobility database developed in this thesis can be applied to current EVA planning, training and hardware design and future planetary EVA system design. The computational work envelope method can be used for EVA and neutral buoyancy simulation planning and EVA hardware design evaluations. A larger and more realistic work envelope estimate would make more efficient use of time and resources in EVA work than the current work envelope estimate. The mathematical model developed in this thesis is implementable in real time and suitable for incorporation into EVA dynamic simulations to evaluate large mass handling situations. The mathematical model can also be incorporated into virtual reality training to supply the space suit's contribution to the virtual reality system's force feedback. Physics-based models of space suit joint mobility will aid in designing joints for planetary space suits that have the proper combination of mobility and stability for best performance in walking, running and traversing rough terrain. An analysis combining kinematics, space suit mobility models, and human strength could be used to assess the lower body mobility of planetary space suits, relating individual joint mobility to terrain-traversing capabilities, such as the maximum height of an object that can be stepped over, aiding in requirement definition for advanced space suits. The space suit mobility database and analytical techniques that were developed in this thesis can be applied to both near-term and advanced EVA planning and design by supplying realistic benchmarks on the mobility performance of the EMU space suit and enabling designers and planners to assess practical performance measures based on small-scale, joint level mobility models.

6.5 Summary of contributions

This thesis advances current knowledge of astronauts' capabilities and limitations in space-suited EVA in three areas: by gathering data on the torques needed to bend space suit joints, developing both mathematical and physics-based models relating joint torques to angles, and using the joint torque-angle data to predict and analyze the work envelope of a space suited astronaut.

The space suit torque-angle database compiled in this thesis is more extensive in the number of joints and angular range than any other published space suit mobility data set. Furthermore, the data was collected under realistic conditions, using joint angle trajectories that were generated by space-suited human subjects and torques measured using an instrumented robot as a surrogate for a person in the space suit. This database provides the basis for developing and validating models of space suit mobility.

The modeling work in this thesis contributes both numerical predictions of the torques needed to bend the space suit joints and insight into the physical processes that govern space suit joint mobility. Preisach hysteresis model coefficients were identified from experimental space suit torque-angle data and a new method was developed for estimating the variance of the error in the Preisach model's torque predictions. The Preisach model was then used to generate numerical predictions of the torque needed to bend the space suit joint as a function of time. The physics-based modeling work compared two approximate models which describe pressurized fabric cylinder bending to experimental data. The beam model assumes that space suit joint bending occurs through elastic deformations of the space suit fabric, while the membrane model assumes that space suit fabric never stretches. The experimental data agrees most closely with the membrane model, indicating that elasticity is not an important contribution to space suit joint bending performance and efforts to improve space suit joint mobility should focus on geometrical aspects of joint design, rather than the material properties of the fabric.

The work envelope analysis demonstrates the usefulness of modeling space suit mobility in predicting a global human factors metric. The work envelope prediction method that this thesis developed is rapid and easily reconfigurable for people of different sizes and strengths. It generates not only boundaries on acceptable work sites, but also predicts the locations of desirable work sites. A sensitivity analysis on the work envelope revealed that improvements in shoulder mobility and upward and downward visibility would be most effective in enlarging the space suited astronaut's work envelope.

References

1. "Space Shuttle System Payload Accommodations, NSTS 07700," NASA, 1988.
2. "Man-Systems Integrations Standards, NASA-STD-3000," NASA, 1995.
3. Abramov, I. P., Stoklitsky, A. Y., Barer, A. S., and Filipenkov, S. N., "Essential Aspects of Space Suit Operating Pressure Trade-off," Proceedings of the International Conference on Environmental Systems, Society of Automotive Engineers, 1994.
4. Anderson, D., "EVA Worksite Analysis--Use of Computer Analysis for EVA Operations Development and Execution," Acta Astronautica, Vol. 44, No. 7-12, 1999, pp. 593-606.
5. Asada, H. and Slotine, J. J. E., Robot Analysis and Control. John Wiley and Sons, New York, 1986.
6. Astrand, P. Textbook of Work Physiology: Physiological Bases of Exercise, McGraw-Hill, New York, 1977.
7. Badler, N. I., Phillips, C. B., and Webber, B. L., Simulating Humans: Computer Graphics Animation and Control, Oxford University Press, New York, 1993.
8. Banks HT, Kurdila AJ, and Webb GV "Identification of hysteretic control influence operators representing smart actuators, part II: Convergent approximations," Journal of Intelligent Material Systems and Structures Vol 8, p.536, 1997.
9. Bergmann, G., Graichen, F., and Rohlman, A., "Hip joint loading during walking and running, measured in two patients," Journal of Biomechanics, Vol. 26, No. 8, 1993, pp. 969-990.
10. Brazier, L. G., "On flexure of thin cylindrical shells and other thin sections," Proceedings of the Royal Society of London, Series A, Vol. 116, 1927, pp. 104-114.
11. Chaffin, DB. Occupational Biomechanics, Wiley, New York, 1984.
12. Comer, RL and Levy, S, "Deflections of an inflated circular cylindrical cantilever beam," AIAA Journal, Vol 1, No 7, July, 1963, pp. 1652-1655.
13. Craig, D and Soffer, L." Robot Load Cell Calibration Report," Unpublished report. June, 2000.
14. Critchlow, AJ. Introduction to Robotics, Macmillan Publishing Co., New York, 1985.
15. Dionne, S., "AX-5, Mk III, and Shuttle Space Suit Comparison Test Summary," NASA, Moffett Field, CA, EVA Technical Document 91-SAE/SD-004, August, 1991.
16. Doong, T. and Mayergoyz, I. "On numerical implementations of hysteresis models" IEEE Transactions on Magnetics, Vol 21 p.1853-1855, 1985.
17. Fay, JP and Steele CR, "Forces for rolling and asymmetric pinching of pressurized cylindrical tubes" Journal of Spacecraft and Rockets, Vol. 36, No. 4, July-August

1999, pp. 531-537.

18. Fay, JP and Steele CR, "Bending and symmetric pinching of pressurized tubes", International Journal of Solids and Structures, Vol 37, 2000, pp. 6917-6931.

19. Firt, V. Statics, Formfinding and Dynamics of Air-supported Membrane Structures, Martinus Nijhoff Publishers, The Hague, 1983.

20. Ge, P. and Jouaneh, M. "Modeling hysteresis in piezoceramic actuators," Precision Engineering, Vol 17, p. 211-221, 1995.

21. Ge, P. and Jouaneh, M. "Tracking control of a piezoceramic actuator," IEEE Transactions on Control Systems Technology, Vol.4, No.3, 1996, p.209.

22. Gere, J. M. and Timoshenko, S. P., Mechanics of Materials, 4th ed, PWS Publishing Company, Boston, 1997.

23. Gray, A. Modern differential geometry of curves and surfaces, CRC Press, Boca Raton, 1993.

24. Graziosi, D., Stein, J., and Kearney, L., "Space Shuttle Small EMU Development," Proceedings of the International Conference on Environmental Systems, Society of Automotive Engineers, Toulouse, France, 2000.

25. Hagale, T. J. and Price, L. R., "EVA Design, Verification, and On-Orbit Operations Support Using Worksite Analysis," Proceedings of the International Conference on Environmental Systems, Society of Automotive Engineers, Toulouse, France, 2000.

26. Hearn, D. and Baker, M. P., Computer Graphics: C Version, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 1997.

27. Hearle J.W.S., Grosberg P., Backer S., Structural Mechanics of Fibers, Yarns and Fabrics Vol. 1., Wiley-Interscience, New York, 1969.

28. Jung, E. S. and Choe, J., "Human reach posture prediction based on psychophysical discomfort," International Journal of Industrial Ergonomics, Vol. 18, 1996, pp. 173-179.

29. Jung, E. S., Kee, D., and Chung, M. K., "Upper body reach posture prediction for ergonomic evaluation model," International Journal of Industrial Ergonomics, Vol. 16, 1995, pp. 95-107.

30. Kamper, D. G. and Rymer, W. Z., "Effects of geometric joint constraints on the selection of final arm posture during reaching: a simulation study," Experimental Brain Research, Vol. 126, 1999, pp. 134-138.

31. Korein, J. U., A Geometrical Investigation of Reach. MIT Press, Cambridge, MA, 1984.

32. Kosloski, L. D., US Space Gear: Outfitting the Astronaut, Smithsonian Institution Press, Washington, DC, 1994.

33. Krasnoselskii M, Pokrovskii, A. Systems with hysteresis. Springer-Verlag, Berlin, 1989.

34. Lukasiewicz, S. and Glockner, P.G., "Stability of lofty air-supported cylindrical

membranes," Journal of Structural Mechanics, Vol.12, No. 4 1984-1985, pp. 543-555.

35. Main, JA. Analysis and design of inflatable aerospace structures. PhD. Thesis, Vanderbilt University, 1993.

36. Main, J. A., Peterson, S. W., and Strauss, A. M., "Material Selection for Highly Mobile Space Suit Gloves Based on Fabric Mechanical Properties," Proceedings of the 43rd IAF Congress, International Astronautical Federation, 1993.

37. Main, J. A., Peterson, S. W., and Strauss, A. M., "Design and Structural Analysis of Highly Mobile Space Suits and Gloves," Journal of Spacecraft and Rockets, Vol. 31, No. 6, 1994, pp. 1115-1122.

38. Main, J. A., Peterson, S. W., and Strauss, A. M, "Beam-type bending of space-based inflatable structures," Journal of Aerospace Engineering, Vol 8, No 2, April, 1995, pp.120-125.

39. Mayergoyz, I. D., Mathematical Models of Hysteresis, Springer-Verlag, New York, NY, 1991.

40. Menendez, V., Diener, M., and Baez, J. M., "Performance of EVA Suit Soft Flat Pattern Mobility Joints," Proceedings of the International Conference on Environmental Systems, Society of Automotive Engineers, 1994.

41. Morgan, D. A., Wilmington, R. P., Pandya, A. K., Maida, J. C., and Demel, K. J., "Comparison of Extravehicular Mobility Unit (EMU) Suited and Unsuited Isolated Joint Strength Measurements," NASA TP-3613, Houston, TX, June, 1996.

42. Newman, D. J., Schmidt, P. B., Rahn, D. B., Badler, N., and Metaxas, D., "Modeling the Extravehicular Mobility Unit Space Suit: Physiological Implications for Extravehicular Activity," Proceedings of the International Conference on Environmental Systems, Society of Automotive Engineers, Toulouse, France, 2000.

43. Pan, K. and Stelson, K. A., "On the Plastic Deformation of a Tube During Bending," Journal of Engineering for Industry, Vol. 117, No. 4, 1995, pp. 494-500.

44. Price, L. R., Fruhwirth, M. A., and Knutson, J. A., "Computer Aided Design and Graphics Techniques for EVA Analysis," Proceedings of the International Conference on Environmental Systems 1994, Society of Automotive Engineers, 1994.

45. Rahn, D. B., "A Dynamic Model of the Extravehicular Mobility Unit (EMU): Human Performance Issues During EVA," Master of Science Thesis, Massachusetts Institute of Technology, Cambridge, MA, 1997.

46. Sanders, M. S. and McCormick, E. J., Human Factors in Engineering and Design, 7th ed. McGraw-Hill, New York, 1993.

47. Tao, G and Kokotovic, P,. Adaptive Control of Systems with Actuator and Sensor Nonlinearities. Wiley, New York, 1996.

48. Thuot, P. J. and Harbaugh, G. J., "Extravehicular Activity Training and Hardware Design Consideration," Acta Astronautica, Vol. 36, No. 1, 1995, pp. 13-26.

49. Timoshenko, S. P. and Gere, J. M., Theory of Elastic Stability, 2nd ed. McGraw-Hill, New York, 1961.

50. Wang, X. and Verriest, J. P., "A Geometric Algorithm to Predict the Arm Reach Posture for Computer-aided Ergonomic Evaluation," The Journal of Visualization and Computer Animation, Vol. 9, 1998, pp. 33-47.

51. Webb G.V., Kurdila A.J. and Lagoudas D.C., "Adaptive Hysteresis Model for Model Reference Control with Actuator Hysteresis," Journal of Guidance, Control, and Dynamics, Vol. 23, No. 3 p.459, 2000.

52. Webb G.V., Lagoudas D.C., and Kurdila A.J., "Hysteresis Modeling of SMA Actuators for Control Applications," Journal of Intelligent Material Systems and Structures, Vol. 9, p.432, 1998.

53. Webber, J.P.H. "Deflections of inflated cylindrical cantilever beams subjected to bending and torsion," Aeronautical Journal, Vol. 86, No. 858, October, 1982, pp 306-312.

54. Wierzbicki, T. and Sinmao, M., "A simplified model of Brazier effect in plastic bending of cylindrical tubes," International Journal of Pressure Vessels and Piping, Vol. 71, No. 1, 1997, pp. 19-28.

55. Zhang, L. C. and Yu, T. X., "An Investigation of the Brazier Effect of a Cylindrical Tube under Pure Elastic-Plastic Bending," International Journal of Pressure Vessels and Piping, Vol. 30, 1987, pp. 77-86.

Appendix A

Experiment Matlab Scripts

```
%  maketraj.m
% script to generate trajectory file from excel out-
put
% this is early test version--trying to see what sam-
pling rate to use in robot
% trajectory file

% load excel file--must be exported as text and
header removed

clear
inpath='PowerMac8100:Data:SubjectC:';
infile='csb10r'
eval([ 'load ',inpath,infile,'.txt'] )
eval([ 'in=',infile,';'] );


% remove first column (times)
out=in(:,2:13);
%out =in;

% fix shoulder flexion angle wraps
out(find(out(:,1)<-160),1)=out(find(out(:,1)<-
160),1)+360;

% do constraint check
outlc=limitcheck(out);

% make sure file is long enough for bogus wr--if not,
pad it
% do this before resample so that it gets low-passed

lout=size(outlc);
lout=lout(1);
reqlen=37;

for index=lout:(reqlen*12)
      outlc=[ outlc;outlc(lout,:)] ;
end %for

size(outlc)

% downsample to 5 hz
for i=1:12
      % downsample in stages, using decimateb, which
is butterworth filter
      temp=decimateb(outlc(:,i),3);
      out5(:,i)=decimateb(temp,4);
end %for

% don't do constraint check again
out5=limitcheck(out5);


% for index=lout:19
%     out5=[ out5;out5(lout,:)] ;
% end %for


%
% if length(out5)<19
%     out5=[ out5;out5(length(out5),:)*ones(19-
length(out5),12)] ;
% end %if
```

```
% insert bogus wrist rotation command
wr=90*ones(size(out5(:,5)));
wr(1:37)=(-90:5:90);
out5(:,5)=wr;

% % for bsa7r only, change zeroed upper body values
to suit neutral position
% nsf=35;
% nsa=40;
% nhr=-85
% nef=42;
%
% out5(:,1)=nsf*ones(size(wr));
% out5(:,2)=nsa*ones(size(wr));
% out5(:,3)=nhr*ones(size(wr));
% out5(:,4)=nef*ones(size(wr));


% reverse sign of humerus rotation
 out5(:,3)=-out5(:,3);

% set speed, based on 5 hz file
% find joint with max range of movement
range=max(out5)-min(out5);
jmax=find(range==max(range));

% find max velocity of that joint
jvel=max(abs(diff(out5(:,jmax))))*5;

disp([ 'max velocity= ',num2str(jvel)])
disp([ 'on joint ',num2str(jmax)])

% save files
outpath='PowerMac8100:Data:robot_trajectories:';
eval([ 'save ', outpath,infile,' out5 -ascii -tabs'])
```

```
% convert to trj file
success=trajconvert(outpath,infile)

% check that trj file is ok
fileok=checktrj(outpath,[ infile,'.trj']);

if fileok
      disp('Trajectory file is formatted correctly')
end %if fileok

% plot joint angles
figure(1)
subplot(411)
plot(out5(:,1),'r')
subplot(412)
plot(out5(:,2),'g')
subplot(413)
plot(out5(:,3),'b')
subplot(414)
plot(out5(:,4),'c')

figure(2)
subplot(411)
plot(out5(:,5),'r')
subplot(412)
plot(out5(:,6),'g')
subplot(413)
plot(out5(:,7),'b')
subplot(414)
plot(out5(:,8),'c')

figure(3)
subplot(411)
plot(out5(:,9),'r')
subplot(412)
```

```
plot(out5(:,10),'g')
subplot(413)
plot(out5(:,11),'b')
subplot(414)
plot(out5(:,12),'c')


function out=limitcheck(in, flag)
% out=limitcheck(in)
% checks command angles against robot's joint angle
limits.
% Modifies commmand file as follows:
% * all commands for joint out of bounds -> command
90% displacement of joint
%   to closer limit
% * command sometimes exceeds range ->  compress out
of bounds by 1 of 2 methods
% selected according to flag input
% flag ==0 => linear to compress out of bounds into
nearest 10 degrees
% flag ==1 => nonlinear to match velocities
%

limits=[ -15 180; ... % shoulder flexion
0 90; ... % shoulder abduction
-90 90; ... % humeral rotation
0 130; ... % elbow flexion
-90 90; ... % wrist rotation
10 20;...% hip abduction  % changed hip abduction 4/
00 to avoid torque limit
0 50;...% hip flexion  % changed hip flexion to 50 5/
10/00 to avoid hf torque limit
-15 15;...% thigh rotation
0 130;...% knee flexion
-20 20;...%ankle rotation
-45 30;... % ankle flexion
-20 20] '; % ankle inversion

% % set slightly tighter limits for eualr,bsa2r
shoulder flexion so sf range <180
% % this is done so that bogus wr data sets speed to
50 deg/sec
% limits=[ -7 173; ... % shoulder flexion
% 0 90; ... % shoulder abduction
% -90 90; ... % humeral rotation
% 0 130; ... % elbow flexion
% -90 90; ... % wrist rotation
% 0 45;...% hip abduction
% 0 100;...% hip flexion
% -22 22;...% thigh rotation
% 0 130;...% knee flexion
% -20 20;...%ankle rotation
% -45 30;... % ankle flexion
% -20 20] '; % ankle inversion


a=[ (limits(1,:)+10);(limits(2,:)-10)] ;
midpoints=[ mean(limits)] ;
ranges=[ limits(1,:)-midpoints;midpoints-lim-
its(1,:)] ;
bwidth=([ .95*ranges(1,:); .95*ranges(2,:)] );

bwidth(find(abs(bwidth)<15))=sign(bwidth(find(abs(bw
idth)<15)))*15;
%b=[ midpoints+.95*ranges(1,:);mid-
points+.95*ranges(2,:)] ;
b=[ midpoints;midpoints] +bwidth;
flag=1;
%keyboard
out=zeros(size(in));

% find extremes of motion
```

```
maxangle=max(in);
minangle=min(in);


jointok=(maxangle<limits(2,:)) & (minangle>lim-
its(1,:)); % in bounds
case1a=(maxangle<limits(1,:)) & (minangle<lim-
its(1,:));  % always low
case1b=(maxangle>limits(2,:)) & (minangle>lim-
its(2,:));  % always high
case2a=(maxangle>limits(2,:)) & (minangle>lim-
its(1,:))  & ~case1b; %sometimes high
case2b=(minangle<limits(1,:)) & (maxangle<lim-
its(2,:)) & ~case1a; % sometimes low
case2c=(minangle<limits(1,:)) & (maxangle>lim-
its(2,:)); % both high and low

% convert to indices
jointok=find(jointok);
case1a=find(case1a)
case1b=find(case1b)
case2a=find(case2a)
case2b=find(case2b)
case2c=find(case2c)

if jointok
      out(:,jointok)=in(:,jointok);
end %if
if case1a
      factor1a=diag(.9*limits(1,case1a));

out(:,case1a)=ones(size(in(:,case1a)))*factor1a;
end % if

if case1b
      factor1b=diag(.9*limits(2,case1b));
out(:,case1b)=ones(size(in(:,case1b)))*factor1b;
end %if



if case2a
      %factor2a=diag(abs(.9*(limits(1,case2a))./
(maxangle(case2a))));
      %val=in(:,case2a);
      %max=ones(size(val))*diag(maxangle(case2a));
      %lim=ones(size(val))*diag(limits(2,case2a));
      %factor2a=1./(1+((max-.9.*lim)./max).*(val./
.9./lim));
      %keyboard
      for index=1:length(case2a)
            col=case2a(index);
            %keyboard
            out(:,col)=fixlim-
its(in(:,col),a(:,col),b(:,col),flag);
      end %for

      %out(:,case2a)=in(:,case2a)*factor2a;
end %if

if case2b
      %val=in(:,case2b);
      %max=ones(size(val))*diag((minangle(case2b)));
      %lim=ones(size(val))*diag((limits(1,case2b)));
      %factor2b=abs(1./(1+((max-.9.*lim)./
max).*(val./.9./lim)));
      %keyboard
      %factor2b=(abs(.9*(limits(1,case2b))./(minan-
gle(case2b))));
      %out(:,case2b)=in(:,case2b);
```

```
%           for index=1:length(case2b)
%           %keyboard
%           lz=find(in(:,case2b(index))<0)
%           mz=find(in(:,case2b(index))>=0);
%                 case2b(index)
%
out(lz,case2b(index))=in(lz,case2b(index))*factor2b(
index);
%
out(mz,case2b(index))=in(mz,case2b(index));
%           keyboard
%           end %for

      for index=1:length(case2b)
            col=case2b(index);
            out(:,col)=fixlim-
its(in(:,col),a(:,col),b(:,col),flag);
      end %for

end %if case2b

if case2c
%           factor2c=diag(.9*(limits(2,case2c)-lim-
its(1,case2c))./(maxangle(case2c)-minan-
gle(case2c)));
%           out(:,case2c)=in(:,case2c)*factor2c;

      for index=1:length(case2c)
            col=case2c(index);
            out(:,col)=fixlim-
its(in(:,col),a(:,col),b(:,col),flag);
            %keyboard
      end %for
end %if


function success=trajconvert(inpath,infile)
% success =trajconvert(inpath,infile);
% This script loads a tab delimited ascii trajectory
file and converts it to the
% file format used by rssta for robot trajectories
% inpath is the path for the input file (sting)
% infile is the filename of the input file (string)

% rssta file format is :
% 3 lines of text, followed by
% n lines of 12 numbers each in %2.2f format
% all lines terminated with \r (line feed, ctrl-v)
% _no_ carriage returns (\n, ctrl-m) at ends of lines
% every 6th character is a decimal point
% 4 characters between line beginning and 1st decimal
point
% 2 characters between last decimal point and \r
% carriage return (ctrl-m,\n) is end of file charac-
ter

% load ascii file
%inpath='PowerMac8100:Data:robot_trajectories:';
%infile='aua8r5';

eval([ 'load ',inpath,infile,' -ascii'] );
eval([ 'data=',infile,';'] );

% prompt for 3 header text lines
text1=input('Enter 1st text line [ Line 1]:  ','s');
disp([ ] )
```

```
text2=input('Enter 2nd text line [Line 2]:  ','s');
disp([ ])

text3=input('Enter 3rd text line [Line 3]:  ','s');
disp([ ])

if isempty(text1)
      text1='Line 1';
end %if

if isempty(text2)
      text2='Line 2';
end %if

if isempty(text3)
      text3='Line 3';
end %if

% open file and write text lines
outfile=[infile,'.trj'];
outpath=inpath;

[fid, message]=fopen([outpath,outfile],'wt');

keyboard
% check that file successfully opened
if fid==-1
      error('Unable to open trajectory file for writ-
ing')
end %if

%fid=1;  % print to screen for debugging


formatstring1='%c';
formatstring2='%c';
formatstring3='%c';

for fs1=1:length(text1)-1
      formatstring1=strcat(formatstring1,'%c ');
end %for

for fs2=1:length(text2)-1
      formatstring2=strcat(formatstring2,'%c');
end %for

for fs3=1:length(text3)-1
      formatstring3=strcat(formatstring3,'%c');
end %for
formatstring1=strcat(formatstring1);

%fprintf(fid,[formatstring1,'\r'],text1)
l1=sprintf([formatstring1,'\r'],text1);
l2=sprintf([formatstring2,'\r'],text2);
l3=sprintf([formatstring3,'\r'],text3);

fprintf(fid,l1);
fprintf(fid,l2);
fprintf(fid,l3);

% loop through data array, saving to output file
% find length of data
nlines=size(data,1);

for index=1:nlines
      for col = 1:12

eval(['c',num2str(col),'=sprintf(''%2.1f'',data(inde
x,col));']);
            % determine how many leading spaces this
element needs
```

```
            eval([ 'ls = 7-
length(c',num2str(col),');'] );
            if col==1
                  ls=ls-1;
            end %if col

            % add leading spaces

eval([ 'c',num2str(col),'=[ blanks(ls),c',num2str(col)
,'] ;'] );
      end %for col
      % assemble line
      line_out=str-
cat(c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12);

      % write line to file
      % CR is EOF character
      formatstring4='%c';
      for j=1:82
            formatstring4=str-
cat(formatstring4,'%c');
      end %for

      if index==nlines
            fprintf(fid,[ formatstring4] ,line_out);
      else
            fprintf(fid,[ formatstring4,'
\r'] ,line_out);
      end %if
      % end of file characters


end % for index
fprintf(fid,'%c\n \n\r',' ');

% close file
```

```
st =fclose(fid);
%st

if st>=0
      success=1;
else
      st
      disp('could not close file')
      success=0;
end %if


function success=checktrj(inpath,infile);
%
% checktrj.m
% success=checktrj(inpath,infile)
% checktrj.m opens .trj files, checks for proper line
termination,
% file termination and spacing between decimal points

%inpath='c:\Patricia\robot\';
%inpath=('PowerMac8100:Data:robot_trajectories:');
%infile='testtraj2.trj'

success =0;
fid=fopen([ inpath,infile] );

if (fid ==-1)
      error([ 'Unable to open file ',inpath,infile,] )
end %if

traj=fscanf(fid,'%c');
trajascii=double(traj);
fclose(fid);
% find all of the special characters
% find ctrl-j's (newlines)
```

```
newlines=find(trajascii==10);
nlspace=diff(newlines);
numlines=length(newlines);


% find ctrl-m's (carriage returns)
crs=find(trajascii==13);

% find decimal points
decs=find(trajascii==46);


% find spaces
spaces=find(trajascii==32);
%keyboard
% check to see that special characters are in the
file
if isempty(newlines)
   error('File contains no line feeds');
end%if

if isempty(crs)
   error('File contains no carriage returns');
end %if

if isempty(decs)
   error('File contains no decimal points');
end %if

if isempty(spaces)
   error('File contains no spaces');
end %if

% if it passes this far, check to see that they are
in the
% right places

% spacing of newlines (line length)--this should
catch corrupted files

if nlspace(4:numlines-2) ~= 85*ones(1,numlines-5)
      error('Line length is incorrect')
end %if

% check decimal point spacing
trajcd=trajascii(newlines(3):length(trajascii));  %
cut out header
trajcd(newlines(4:length(newlines))-new-
lines(3))=[];  % cut out newlines
decsdata=find(trajcd==46);
decspace=diff(decsdata); % now there should be a dec-
imal point every 7th char
nomdecspace=7*ones(size(decspace));  % compare
actual to nominal

if decspace~=nomdecspace
      error('Decimal point spacing is incorrect')
end %if

% check for ctrl-m at end, following last decimal
point
lastdec=decs(length(decs));
correct_termination=trajascii(lastdec+3)==13;

if ~correct_termination
      error('File termination is incorrect')
end %if

success=1;  % file is ok


% maindatareduction.m
```

```
% main script for data reduction
% calls loadrobotdata.m (hdrload.m), register.m

clear
datapath={ 'Macintosh HD:Users:Patricia:Docu-
ments:Data:'};
outrawpath={ datapath 'reduced_data:raw_data:'};
outprocpath={ datapath 'reduced_data:proc_data:'};
regflag=1;  % 1 for register, 0 for no register
eval([ 'load ''', datapath,'rdatastruct.mat''']);

% i=1;  % put loops here
%j=3;  % subject #

for j=4
%for j=4
for i=1
     [ data] =loadrobotdata(Robotdata(i,j));
      if ~isempty(data) & ~isempty({ data.full-
speed.suited} )

           if regflag==1
                % find shifts needed for time sync
and store start, end, offsets in their
                % places in Robotdata
                [ robotdataij] =register(Robot-
data(i,j),data,i);
                Robotdata(i,j)=robotdataij;
                save 'Macintosh HD:Users:Patri-
cia:Documents:Data:rdatastruct.mat' Robotdata
           else
                % use already-saved reg info
                robotdataij=Robotdata(i,j);
           end %if

                % now implement shifting and store sync'd
versions in data
               [ newdata] =adjust(data,Robotdata(i,j));
                data=newdata;  % data files now same
length or [], can be subtracted

                % do processing on data
                outdata=processdata(data,robotdataij,i);

                % save files
                fname=Robotdata(i,j).trialtype;

                eval([ 'save ''',outraw-
path,fname,'raw.mat''',' data']);
                eval([ 'save ''',outproc-
path,fname,'.mat''',' outdata']);


     end %if
end %for
end %for
```

```
% robotdatastruct.m
% sets up big nested struct with data file names,
types, reduced data file names and registration info


% set temp variables with information

trialtypetemp={ 'aa1' 'ba1' 'ca1' 'ea1';'aa2' 'ba2'
'ca2' 'ea2';'aa3' 'ba3' 'ca3' 'ea3';...
'aa4' 'ba4' 'ca4' 'ea4';'aa5' 'ba5' 'ca5' 'ea5';'aa6'
'ba6' 'ca6' 'ea6';...
```

```
'aa7' 'ba7' 'ca7' 'ea7';'aa8' 'ba8' 'ca8' 'ea8';'aa9'
'ba9' 'ca9' 'ea9';...
'aa10' 'ba10' 'ca10' 'ea10';'aa11' 'ba11' 'ca11'
'ea11';'ab1' 'bb1' 'cb1' 'eb1';...
'ab2' 'bb2' 'cb2' 'eb2';'ab3' 'bb3' 'cb3' 'eb3';'ab4'
'bb4' 'cb4' 'eb4';...
'ab5' 'bb5' 'cb5' 'eb5';'ab6' 'bb6' 'cb6' 'eb6';'ab7'
'bb7' 'cb7' 'eb7';...
'ab9_1' 'bb9_1' 'cb9_1' 'eb9_1';'ab9_2' 'bb9_2'
'cb9_2' 'eb9_2';'ab9_3' 'bb9_3' 'cb9_3' 'eb9_3';...
'ab9_4' 'bb9_4' 'cb9_4' 'eb9_4';'ab10' 'bb10' 'cb10'
'eb10'};

trajtemp=cell(size(trialtypetemp));

for i=1:23
      for j=1:4
            typetemp=char(trialtypetemp(i,j));
            trajtemp(i,j)={[typetemp(1),'s',type-
temp(2:length(typetemp)),'r']};
      end %for j
end % for i

% enter 'none' for files we don't have/trials not
done

% no sub a group a trials
[trajtemp(1:11,1)]=deal({'none'});
[trajtemp(13:14,1)]=deal({'none'});
[trajtemp(18:20,1)]=deal({'none'});

% few b's missing
[trajtemp(5,2),trajtemp(16:17,2)]=deal({'none'}); %
get rid of 16 and 17 when unsuited files run

% no c's are missing

% one e is missing
[trajtemp(18,4)]=deal({'none'});

% now set up structs
nest1=struct('nws',struct('name','stuff','rms',0,'of
fset',0,'torque-
file','fname'),'ws',struct('name','stuff','rms',0,'o
ffset',0,'torque-
file','fname'),'suited',struct('name','stuff','start
',0,'finish',0,'offset',0));
nest2=nest1;
Robotdata=struct('trialtype',trialtypetemp,'traj-
file',trajtemp,'fullspeed',nest1,'halfspeed',nest2);

% this has the right structure--now need to fill in
filenames

% long list of filename assignments--68 of them



eval('cd ''Macintosh HD:Users:Patricia:Docu-
ments:Data:suited_data''')

for i=1:23
      for j=1:4
            trajfilename=char(Robotdata(i,j).traj-
file);
            if trajfilename(1:4)~='none';
                  tfn=trajfilename(1:length(trajfile-
name)-1);
                  eval('cd ''Macintosh
HD:Users:Patricia:Documents:Data:robot_data''')
```

```
                    nwsf-
files=[ dir([ tfn,'fs.dat'] ),dir([ tfn,'fs2.dat'] ),dir(
[ tfn,'fs3.dat'] ),dir([ tfn,'fs4.dat'] )] ;
                    wsf-
files=[ dir([ tfn,'w.dat'] ),dir([ tfn,'w2.dat'] ),dir([ t
fn,'w3.dat'] ),dir([ tfn,'w4.dat'] )] ;
                    warning off
                    wsffiles=[ wsf-
files,dir([ tfn,'fsw.dat'] ),dir([ tfn,'fsw2.dat'] ),dir
([ tfn,'fsw3.dat'] ),dir([ tfn,'fsw4.dat'] )] ;
                    warning on
                    nwsh-
files=[ dir([ tfn,'hs.dat'] ),dir([ tfn,'hs2.dat'] ),dir(
[ tfn,'hs3.dat'] ),dir([ tfn,'hs4.dat'] )] ;
                    wsh-
files=[ dir([ tfn,'hsw.dat'] ),dir([ tfn,'hsw2.dat'] ),di
r([ tfn,'hsw3.dat'] ),dir([ tfn,'hsw4.dat'] )] ;

                    eval('cd ''Macintosh
HD:Users:Patricia:Documents:Data:suited_data''')
                    suitfsfiles=dir([ tfn,'sf*.dat'] );
                    suithsfiles=dir([ tfn,'sh*.dat'] );

                    Robotdata(i,j).fullspeed.nws.name=[ nwsf-
files.name] ;
                    Robotdata(i,j).fullspeed.ws.name=[ wsf-
files.name] ;
                    Robotdata(i,j).halfspeed.nws.name=[ nwsh-
files.name] ;
                    Robotdata(i,j).halfspeed.ws.name=[ wsh-
files.name] ;

                    Robotdata(i,j).full-
speed.suited.name=[ suitfsfiles.name] ;
                    Robotdata(i,j).half-
speed.suited.name=[ suithsfiles.name] ;
```

```
            end % if traj
        end %for j
end %fori


% save big struct in Data directory
save 'Macintosh HD:Users:Patricia:Docu-
ments:Data:rdatastruct.mat' Robotdata
```

```
% robotdatastruct.m
% sets up big nested struct with data file names,
types, reduced data file names and registration info


% set temp variables with information

trialtypetemp=[ 'aa1' 'ba1' 'ca1' 'ea1';'aa2' 'ba2'
'ca2' 'ea2';'aa3' 'ba3' 'ca3' 'ea3';...
'aa4' 'ba4' 'ca4' 'ea4';'aa5' 'ba5' 'ca5' 'ea5';'aa6'
'ba6' 'ca6' 'ea6';...
'aa7' 'ba7' 'ca7' 'ea7';'aa8' 'ba8' 'ca8' 'ea8';'aa9'
'ba9' 'ca9' 'ea9';...
```

```
'aa10' 'ba10' 'ca10' 'ea10';'aa11' 'ba11' 'ca11'
'ea11';'ab1' 'bb1' 'cb1' 'eb1';...
'ab2' 'bb2' 'cb2' 'eb2';'ab3' 'bb3' 'cb3' 'eb3';'ab4'
'bb4' 'cb4' 'eb4';...
'ab5' 'bb5' 'cb5' 'eb5';'ab6' 'bb6' 'cb6' 'eb6';'ab7'
'bb7' 'cb7' 'eb7';...
'ab9_1' 'bb9_1' 'cb9_1' 'eb9_1';'ab9_2' 'bb9_2'
'cb9_2' 'eb9_2';'ab9_3' 'bb9_3' 'cb9_3' 'eb9_3';...
'ab9_4' 'bb9_4' 'cb9_4' 'eb9_4';'ab10' 'bb10' 'cb10'
'eb10'};

trajtemp=cell(size(trialtypetemp));

for i=1:23
      for j=1:4
            typetemp=char(trialtypetemp(i,j));
            trajtemp(i,j)={[typetemp(1),'s',type-
temp(2:length(typetemp)),'r']};
      end %for j
end % for i

% enter 'none' for files we don't have/trials not
done

% no sub a group a trials
[trajtemp(1:11,1)]=deal({'none'});
[trajtemp(13:14,1)]=deal({'none'});
[trajtemp(18:20,1)]=deal({'none'});

% few b's missing
[trajtemp(5,2),trajtemp(16:17,2)]=deal({'none'}); %
get rid of 16 and 17 when unsuited files run

% no c's are missing

% one e is missing
[trajtemp(18,4)]=deal({'none'});

% now set up structs
nest1=struct('nws',struct('name','stuff','rms',0,'of
fset',0,'torque-
file','fname'),'ws',struct('name','stuff','rms',0,'o
ffset',0,'torque-
file','fname'),'suited',struct('name','stuff','start
',0,'finish',0,'offset',0));
nest2=nest1;
Robotdata=struct('trialtype',trialtypetemp,'traj-
file',trajtemp,'fullspeed',nest1,'halfspeed',nest2);

% this has the right structure--now need to fill in
filenames

% long list of filename assignments--68 of them



eval('cd ''Macintosh HD:Users:Patricia:Docu-
ments:Data:suited_data''')

for i=1:23
      for j=1:4
            trajfilename=char(Robotdata(i,j).traj-
file);
            if trajfilename(1:4)~='none';
                  tfn=trajfilename(1:length(trajfile-
name)-1);
                  eval('cd ''Macintosh
HD:Users:Patricia:Documents:Data:robot_data''')
                  nwsf-
files=[dir([tfn,'fs.dat']),dir([tfn,'fs2.dat']),dir(
[tfn,'fs3.dat']),dir([tfn,'fs4.dat'])];
```

```
                    wsf-
files=[ dir([ tfn,'w.dat'] ),dir([ tfn,'w2.dat'] ),dir([ t
fn,'w3.dat'] ),dir([ tfn,'w4.dat'] )] ;
                    warning off
                    wsffiles=[ wsf-
files,dir([ tfn,'fsw.dat'] ),dir([ tfn,'fsw2.dat'] ),dir
([ tfn,'fsw3.dat'] ),dir([ tfn,'fsw4.dat'] )] ;
                    warning on
                    nwsh-
files=[ dir([ tfn,'hs.dat'] ),dir([ tfn,'hs2.dat'] ),dir(
[ tfn,'hs3.dat'] ),dir([ tfn,'hs4.dat'] )] ;
                    wsh-
files=[ dir([ tfn,'hsw.dat'] ),dir([ tfn,'hsw2.dat'] ),di
r([ tfn,'hsw3.dat'] ),dir([ tfn,'hsw4.dat'] )] ;

                    eval('cd ''Macintosh
HD:Users:Patricia:Documents:Data:suited_data''')
                    suitfsfiles=dir([ tfn,'sf*.dat'] );
                    suithsfiles=dir([ tfn,'sh*.dat'] );

                    Robotdata(i,j).fullspeed.nws.name=[ nwsf-
files.name] ;
                    Robotdata(i,j).fullspeed.ws.name=[ wsf-
files.name] ;
                    Robotdata(i,j).halfspeed.nws.name=[ nwsh-
files.name] ;
                    Robotdata(i,j).halfspeed.ws.name=[ wsh-
files.name] ;

                    Robotdata(i,j).full-
speed.suited.name=[ suitfsfiles.name] ;
                    Robotdata(i,j).half-
speed.suited.name=[ suithsfiles.name] ;
```

```
            end % if traj
        end %for j
end %fori


% save big struct in Data directory
save 'Macintosh HD:Users:Patricia:Docu-
ments:Data:rdatastruct.mat' Robotdata
```

```
function [ robotdataij] =register(robot-
dataij,data,motion)
% register.m  plots all data files for the same sub-
ject, motion and speed, prompts
% for clicks on same feature of each data trace and
beginning and end, calculates index
% shifts needed to align files and stores start, end
and offset in robotdataij struct
% Robotdata(i,j) can then be set to robotdataij in
main script

clg
% determine active column of data
colindices=2*[ 1 2 4 3 7 8 9 8 11 10 8 2 1 6 7 1 1 9 7
7 7 7 7] ;
active=colindices(motion);
```

```
colors=[ 'k';'r';'g';'b';'c';'m';'y';'x'];
colorindex=1;  % colorindex-1=number of plots done

nwsplots=[ ];
wsplots=nwsplots;
suitedplots=nwsplots;
firstsuited=[ ];
% do full speed files first

for index=1:5
      if ~isempty(data.fullspeed(index).nws)
            plot(data.full-
speed(index).nws(:,active), colors(rem(colorin-
dex,7)+1))
            hold on
            nwsplots=[ nwsplots, colorindex];
            colorindex=colorindex+1;
      end %if

      if ~isempty(data.fullspeed(index).ws)
            plot(data.full-
speed(index).ws(:,active), colors(rem(colorin-
dex,7)+1))
            hold on
            wsplots=[ wsplots,colorindex];
            colorindex=colorindex+1;
      end %if


      if ~isempty(data.fullspeed(index).suited)
            if isempty(firstsuited)
                  firstsuited=index;
            end %if
            plot(data.full-
speed(index).suited(:,active), colors(rem(colorin-
dex,7)+1))
            hold on
            suitedplots=[ suitedplots;colorindex];
            colorindex=colorindex+1;
      end %if

end %for

% take mouse clicks

% offsets
title([ robotdataij.trialtype ' Full speed:  Click on
same feature in r, g, b, c, m, y, k'])
ylabel('angle')
xlabel('index')

clickoffset=ginput(colorindex-1);  % colorindex-1 =
number of plots done
clickoffset=clickoffset(:,1);

cosuited=round(clickoffset(suitedplots));
conws=round(clickoffset(nwsplots));
cows=round(clickoffset(wsplots));

% start and end-this has to be for suited #1 only

      clg

      plot(data.fullspeed(first-
suited).suited(:,active),'r')

      title([ robotdataij.trialtype ' Full speed:
Click on start then end'])
      startend=ginput(2);
      startend=round(startend(:,1));
```

```
      robotdataij.fullspeed.suited.start(first-
suited)=startend(1);
      robotdataij.fullspeed.suited.finish(first-
suited)=startend(2);


% shift all other files to **** 1st suited file *****
%(this is arbitrary-it should exist for all data we
analyze)
nwsindex=1;
wsindex=1;
suitindex=1;

for index=1:5
      if ~isempty(data.fullspeed(index).nws)
            robotdataij.fullspeed.nws.off-
set(index)=conws(nwsindex)-cosuited(1);
            nwsindex=nwsindex+1;
      end %if

      if ~isempty(data.fullspeed(index).ws)
            robotdataij.fullspeed.ws.off-
set(index)=cows(wsindex)-cosuited(1);
            wsindex=wsindex+1;
      end %if

      if ~isempty(data.fullspeed(index).suited)
            robotdataij.fullspeed.suited.off-
set(index)=cosuited(suitindex)-cosuited(1);
            suitindex=suitindex+1;
      end %if

end %for
```

```
% now do same for ********halfspeed***********
clg
nwsplots=[];
wsplots=nwsplots;
suitedplots=nwsplots;
firstsuited=[];
colorindex=1;
% do full speed files first

for index=1:5
      if ~isempty(data.halfspeed(index).nws)
            plot(data.half-
speed(index).nws(:,active), colors(rem(colorin-
dex,7)+1))
            hold on
            nwsplots=[nwsplots, colorindex];
            colorindex=colorindex+1;
      end %if

      if ~isempty(data.halfspeed(index).ws)
            plot(data.half-
speed(index).ws(:,active), colors(rem(colorin-
dex,7)+1))
            hold on
            wsplots=[wsplots,colorindex];
            colorindex=colorindex+1;
      end %if


      if ~isempty(data.halfspeed(index).suited)
            if isempty(firstsuited)
                  firstsuited=index;
            end %if
            plot(data.half-
speed(index).suited(:,active), colors(rem(colorin-
dex,7)+1))
```

```
            hold on
            suitedplots=[ suitedplots;colorindex] ;
            colorindex=colorindex+1;
      end %if

end %for

% take mouse clicks

% offsets
title([ robotdataij.trialtype ' Half speed:  Click on
same feature in r, g, b, c, m, y, k'] )
ylabel('angle')
xlabel('index')

clickoffset=ginput(colorindex-1);  % colorindex-1 =
number of plots done
clickoffset=clickoffset(:,1);

cosuited=round(clickoffset(suitedplots));
conws=round(clickoffset(nwsplots));
cows=round(clickoffset(wsplots));

% start and end-this has to be for suited #1 only

      clg

      plot(data.halfspeed(first-
suited).suited(:,active),'r')

      title([ robotdataij.trialtype ' Full speed:
Click on start then end'] )
      startend=ginput(2);
      startend=round(startend(:,1));
      robotdataij.halfspeed.suited.start(first-
suited)=startend(1);
      robotdataij.halfspeed.suited.finish(first-
suited)=startend(2);


% shift all other files to **** 1st suited file *****
%(this is arbitrary-it should exist for all data we
analyze)
nwsindex=1;
wsindex=1;
suitindex=1;

for index=1:5
      if ~isempty(data.halfspeed(index).nws)
            robotdataij.halfspeed.nws.off-
set(index)=conws(nwsindex)-cosuited(1);
            nwsindex=nwsindex+1;
      end %if

      if ~isempty(data.halfspeed(index).ws)
            robotdataij.halfspeed.ws.off-
set(index)=cows(wsindex)-cosuited(1);
            wsindex=wsindex+1;
      end %if

      if ~isempty(data.halfspeed(index).suited)
            robotdataij.halfspeed.suited.off-
set(index)=cosuited(suitindex)-cosuited(1);
            suitindex=suitindex+1;
      end %if

end %for
```

```
function [ outdata] =adjust(data,robotdataij)
% adjust.m  takes data files and applies offset,
start, and end from
% register.m outputting a modified version of data

% start with fullspeed
% check to see that start+offset >=1 and finish+off-
set<=length(datafile)
% then set outdata with shifted data file
% if exception, don't set outdata and display info on
file that's bad


% set start and finish based on 1st suited file


% do this first for full speed, then same for half
speed
startset=0;
for i=1:5
      if ~startset
            start=robotdataij.full-
speed.suited.start(i);
            finish=robotdataij.fullspeed.suited.fin-
ish(i);
            startset=1;
      end %if not startset

end %for i

outdata=data;

offtol=10;  % offset tolerance

for i=1:5

      % no wet suit

      % check that file exists
      if ~isempty(data.fullspeed(i).nws)

            offset=robotdataij.fullspeed.nws.off-
set(i);
            len=size(data.fullspeed(i).nws,1);
            shortstart=[];
            shortend=[];

            % if s
            if start+offset<1 & start+offset>=-off-
tol  % fix if start is slightly off
                  shortstart=ones(-1*(start+offset-
1),1)*data.fullspeed(i).nws(1,:);
            end %if

            if finish+offset-len<=offtol & fin-
ish+offset-len>0  % fix if end is slightly off
                  shortend=ones(finish+offset-
len,1)*data.fullspeed(i).nws(len,:);
            end %if

            if finish+offset-len>offtol | start+off-
set<-offtol  % reject if either start or end is more
than 5 samples off
                  outdata.fullspeed(i).nws=[];
                  disp([ robotdataij.full-
speed.nws.name(i)] )
                  start
                  finish
```

```
                offset
                len
            else     % do adjustment here...store in
outdata
                outdata.fullspeed(i).nws=[ short-
start;data.fullspeed(i).nws(max([ start+off-
set,1] ):min([ finish+offset,len] ),:);shortend] ;
            end %if finish

      end %if ~isempty


% wet suit

      % check that file exists
      if ~isempty(data.fullspeed(i).ws)

            offset=robotdataij.fullspeed.ws.off-
set(i);
            len=size(data.fullspeed(i).ws,1);
            shortstart=[ ] ;
            shortend=[ ] ;

            % if s
            if start+offset<1 & start+offset>=-off-
tol  % fix if start is slightly off
                shortstart=ones(-1*(start+offset-
1),1)*data.fullspeed(i).ws(1,:);
            end %if

            if finish+offset-len<=offtol & fin-
ish+offset-len>0  % fix if end is slightly off
                shortend=ones(finish+offset-
len,1)*data.fullspeed(i).ws(len,:);
            end %if
```

```
            if finish+offset-len>offtol | start+off-
set<-offtol  % reject if either start or end is more
than 5 samples off
                outdata.fullspeed(i).ws=[ ] ;
                disp([ robotdataij.full-
speed.ws.name(i)] )
                start
                finish
                offset
                len
            else     % do adjustment here...store in
outdata
                outdata.fullspeed(i).ws=[ short-
start;data.fullspeed(i).ws(max([ start+off-
set,1] ):min([ finish+offset,len] ),:);shortend] ;
            end %if finish

      end %if ~isempty


      % suited

      % check that file exists
      if ~isempty(data.fullspeed(i).suited)

            offset=robotdataij.fullspeed.suited.off-
set(i);
            len=size(data.fullspeed(i).suited,1);
            shortstart=[ ] ;
            shortend=[ ] ;

            % if s
            if start+offset<1 & start+offset>=-off-
tol  % fix if start is slightly off
                shortstart=ones(-1*(start+offset)-
1,1)*data.fullspeed(i).suited(1,:);
```

```
            end %if

            if finish+offset-len<=offtol & fin-
ish+offset-len>0  % fix if end is slightly off
                  shortend=ones(finish+offset-
len,1)*data.fullspeed(i).suited(len,:);
            end %if

            if finish+offset-len>offtol | start+off-
set<-offtol  % reject if either start or end is more
than 5 samples off
                  outdata.fullspeed(i).suited=[];
                  disp([ robotdataij.full-
speed.suited.name(i)] )
                  start
                  finish
                  offset
                  len
            else     % do adjustment here...store in
outdata
                  outdata.fullspeed(i).suited=[ short-
start;data.fullspeed(i).suited(max([ start+off-
set,1] ):min([ finish+offset,len] ),:);shortend] ;
            end %if finish

      end %if ~isempty




end %for



% now for half speed
```

```
startset=0;
for i=1:5
      if ~startset
            start=robotdataij.half-
speed.suited.start(i);
            finish=robotdataij.halfspeed.suited.fin-
ish(i);
            startset=1;
      end %if not startset

end %for i



for i=1:5

      % no wet suit

      % check that file exists
      if ~isempty(data.halfspeed(i).nws)

            offset=robotdataij.halfspeed.nws.off-
set(i);
            len=size(data.halfspeed(i).nws,1);
            shortstart=[];
            shortend=[];

            % if s
            if start+offset<1 & start+offset>=-off-
tol  % fix if start is slightly off
                  shortstart=ones(-1*(start+offset-
1),1)*data.halfspeed(i).nws(1,:);
            end %if

            if finish+offset-len<=offtol & fin-
ish+offset-len>0  % fix if end is slightly off
```

```
                shortend=ones(-1*(finish+offset-
len),1)*data.halfspeed(i).nws(len,:);
            end %if

            if finish+offset-len>offtol | start+off-
set<-offtol  % reject if either start or end is more
than 5 samples off
                outdata.halfspeed(i).nws=[];
                disp([ robotdataij.half-
speed.nws.name(i)])
                start
                finish
                offset
                len
            else     % do adjustment here...store in
outdata
                outdata.halfspeed(i).nws=[ short-
start;data.halfspeed(i).nws(max([ start+off-
set,1]):min([ finish+offset,len]),:);shortend];
            end %if finish
%           disp('hs nws')
%           keyboard

      end %if ~isempty


% wet suit

      % check that file exists
      if ~isempty(data.halfspeed(i).ws)

            offset=robotdataij.halfspeed.ws.off-
set(i);
            len=size(data.halfspeed(i).ws,1);
            shortstart=[];
            shortend=[];
            % if s
            if start+offset<1 & start+offset>=-off-
tol  % fix if start is slightly off
                shortstart=ones(-1*(start+offset-
1),1)*data.halfspeed(i).ws(1,:);
            end %if

            if finish+offset-len<=offtol & fin-
ish+offset-len>0  % fix if end is slightly off
                shortend=ones(finish+offset-
len,1)*data.halfspeed(i).ws(len,:);
            end %if

            if finish+offset-len>offtol | start+off-
set<-offtol  % reject if either start or end is more
than 5 samples off
                outdata.halfspeed(i).ws=[];
                disp([ robotdataij.half-
speed.ws.name(i)])
                start
                finish
                offset
                len
            else     % do adjustment here...store in
outdata
                outdata.halfspeed(i).ws=[ short-
start;data.halfspeed(i).ws(max([ start+off-
set,1]):min([ finish+offset,len]),:);shortend];
            end %if finish
%           disp('hs ws')
%           keyboard
      end %if ~isempty


      % suited
```

```
      % check that file exists
      if ~isempty(data.halfspeed(i).suited)

            offset=robotdataij.halfspeed.suited.off-
set(i);
            len=size(data.halfspeed(i).suited,1);
            shortstart=[];
            shortend=[];

            % if s
            if start+offset<1 & start+offset>=-off-
tol  % fix if start is slightly off
                  shortstart=ones(-1*(start+offset-
1),1)*data.halfspeed(i).suited(1,:);
            end %if

            if finish+offset-len<=offtol & fin-
ish+offset-len>0  % fix if end is slightly off
                  shortend=ones(finish+offset-
len,1)*data.halfspeed(i).suited(len,:);
            end %if

            if finish+offset-len>offtol | start+off-
set<-offtol  % reject if either start or end is more
than 5 samples off
                  outdata.halfspeed(i).suited=[];
                  disp([ robotdataij.half-
speed.suited.name(i)])
                  start
                  finish
                  offset
                  len
            else     % do adjustment here...store in
outdata
                  outdata.halfspeed(i).suited=[ short-
start;data.halfspeed(i).suited(max([ start+off-
set,1]):min([ finish+offset,len]),:);shortend];
            end %if finish
%           disp('hs suited')
%           keyboard
      end %if ~isempty



end %for



function [ outdata]=adjust(data,robotdataij)
% adjust.m  takes data files and applies offset,
start, and end from
% register.m outputting a modified version of data

% start with fullspeed
% check to see that start+offset >=1 and finish+off-
set<=length(datafile)
% then set outdata with shifted data file
% if exception, don't set outdata and display info on
file that's bad


% set start and finish based on 1st suited file


% do this first for full speed, then same for half
speed
startset=0;
for i=1:5
      if ~startset
```

```
            start=robotdataij.full-
speed.suited.start(i);
            finish=robotdataij.fullspeed.suited.fin-
ish(i);
            startset=1;
      end %if not startset

end %for i

outdata=data;
offtol=10;  % offset tolerance

for i=1:5

      % no wet suit

      % check that file exists
      if ~isempty(data.fullspeed(i).nws)

            offset=robotdataij.fullspeed.nws.off-
set(i);
            len=size(data.fullspeed(i).nws,1);
            shortstart=[];
            shortend=[];

            % if s
            if start+offset<1 & start+offset>=-off-
tol  % fix if start is slightly off
                  shortstart=ones(-1*(start+offset-
1),1)*data.fullspeed(i).nws(1,:);
            end %if

            if finish+offset-len<=offtol & fin-
ish+offset-len>0  % fix if end is slightly off
                  shortend=ones(finish+offset-
len,1)*data.fullspeed(i).nws(len,:);
            end %if

            if finish+offset-len>offtol | start+off-
set<-offtol  % reject if either start or end is more
than 5 samples off
                  outdata.fullspeed(i).nws=[];
                  disp([ robotdataij.full-
speed.nws.name(i)])
                  start
                  finish
                  offset
                  len
            else     % do adjustment here...store in
outdata
                  outdata.fullspeed(i).nws=[ short-
start;data.fullspeed(i).nws(max([ start+off-
set,1]):min([ finish+offset,len]),:);shortend];
            end %if finish

      end %if ~isempty


% wet suit

      % check that file exists
      if ~isempty(data.fullspeed(i).ws)

            offset=robotdataij.fullspeed.ws.off-
set(i);
            len=size(data.fullspeed(i).ws,1);
            shortstart=[];
            shortend=[];

            % if s
            if start+offset<1 & start+offset>=-off-
tol  % fix if start is slightly off
```

```
                shortstart=ones(-1*(start+offset-
1),1)*data.fullspeed(i).ws(1,:);
            end %if

            if finish+offset-len<=offtol & fin-
ish+offset-len>0  % fix if end is slightly off
                shortend=ones(finish+offset-
len,1)*data.fullspeed(i).ws(len,:);
            end %if

            if finish+offset-len>offtol | start+off-
set<-offtol  % reject if either start or end is more
than 5 samples off
                outdata.fullspeed(i).ws=[];
                disp([ robotdataij.full-
speed.ws.name(i)] )
                start
                finish
                offset
                len
            else     % do adjustment here...store in
outdata
                outdata.fullspeed(i).ws=[ short-
start;data.fullspeed(i).ws(max([ start+off-
set,1] ):min([ finish+offset,len] ),:);shortend] ;
            end %if finish

      end %if ~isempty


      % suited

      % check that file exists
      if ~isempty(data.fullspeed(i).suited)
            offset=robotdataij.fullspeed.suited.off-
set(i);
            len=size(data.fullspeed(i).suited,1);
            shortstart=[];
            shortend=[];

            % if s
            if start+offset<1 & start+offset>=-off-
tol  % fix if start is slightly off
                  shortstart=ones(-1*(start+offset)-
1,1)*data.fullspeed(i).suited(1,:);
            end %if

            if finish+offset-len<=offtol & fin-
ish+offset-len>0  % fix if end is slightly off
                  shortend=ones(finish+offset-
len,1)*data.fullspeed(i).suited(len,:);
            end %if

            if finish+offset-len>offtol | start+off-
set<-offtol  % reject if either start or end is more
than 5 samples off
                  outdata.fullspeed(i).suited=[];
                  disp([ robotdataij.full-
speed.suited.name(i)] )
                  start
                  finish
                  offset
                  len
            else     % do adjustment here...store in
outdata
                  outdata.fullspeed(i).suited=[ short-
start;data.fullspeed(i).suited(max([ start+off-
set,1] ):min([ finish+offset,len] ),:);shortend] ;
            end %if finish
```

```
      end %if ~isempty



end %for



% now for half speed
startset=0;
for i=1:5
      if ~startset
            start=robotdataij.half-
speed.suited.start(i);
            finish=robotdataij.halfspeed.suited.fin-
ish(i);
            startset=1;
      end %if not startset

end %for i



for i=1:5

      % no wet suit

      % check that file exists
      if ~isempty(data.halfspeed(i).nws)

            offset=robotdataij.halfspeed.nws.off-
set(i);
            len=size(data.halfspeed(i).nws,1);
            shortstart=[];
            shortend=[];

            % if s
            if start+offset<1 & start+offset>=-off-
tol  % fix if start is slightly off
                  shortstart=ones(-1*(start+offset-
1),1)*data.halfspeed(i).nws(1,:);
            end %if

            if finish+offset-len<=offtol & fin-
ish+offset-len>0  % fix if end is slightly off
                  shortend=ones(-1*(finish+offset-
len),1)*data.halfspeed(i).nws(len,:);
            end %if

            if finish+offset-len>offtol | start+off-
set<-offtol  % reject if either start or end is more
than 5 samples off
                  outdata.halfspeed(i).nws=[];
                  disp([ robotdataij.half-
speed.nws.name(i)] )
                  start
                  finish
                  offset
                  len
            else     % do adjustment here...store in
outdata
                  outdata.halfspeed(i).nws=[ short-
start;data.halfspeed(i).nws(max([ start+off-
set,1] ):min([ finish+offset,len] ),:);shortend] ;
            end %if finish
%           disp('hs nws')
%           keyboard

      end %if ~isempty
```

```
% wet suit

      % check that file exists
      if ~isempty(data.halfspeed(i).ws)

            offset=robotdataij.halfspeed.ws.off-
set(i);
            len=size(data.halfspeed(i).ws,1);
            shortstart=[];
            shortend=[];

            % if s
            if start+offset<1 & start+offset>=-off-
tol  % fix if start is slightly off
                  shortstart=ones(-1*(start+offset-
1),1)*data.halfspeed(i).ws(1,:);
            end %if

            if finish+offset-len<=offtol & fin-
ish+offset-len>0  % fix if end is slightly off
                  shortend=ones(finish+offset-
len,1)*data.halfspeed(i).ws(len,:);
            end %if

            if finish+offset-len>offtol | start+off-
set<-offtol  % reject if either start or end is more
than 5 samples off
                  outdata.halfspeed(i).ws=[];
                  disp([ robotdataij.half-
speed.ws.name(i)])
                  start
                  finish
                  offset
                  len
            else     % do adjustment here...store in
outdata
                  outdata.halfspeed(i).ws=[ short-
start;data.halfspeed(i).ws(max([ start+off-
set,1]):min([ finish+offset,len]),:);shortend];
            end %if finish
%           disp('hs ws')
%           keyboard
      end %if ~isempty


      % suited

      % check that file exists
      if ~isempty(data.halfspeed(i).suited)

            offset=robotdataij.halfspeed.suited.off-
set(i);
            len=size(data.halfspeed(i).suited,1);
            shortstart=[];
            shortend=[];

            % if s
            if start+offset<1 & start+offset>=-off-
tol  % fix if start is slightly off
                  shortstart=ones(-1*(start+offset-
1),1)*data.halfspeed(i).suited(1,:);
            end %if

            if finish+offset-len<=offtol & fin-
ish+offset-len>0  % fix if end is slightly off
                  shortend=ones(finish+offset-
len,1)*data.halfspeed(i).suited(len,:);
            end %if
```

```
            if finish+offset-len>offtol | start+off-
set<-offtol  % reject if either start or end is more
than 5 samples off
                  outdata.halfspeed(i).suited=[];
                  disp([ robotdataij.half-
speed.suited.name(i)])
                  start
                  finish
                  offset
                  len
            else     % do adjustment here...store in
outdata
                  outdata.halfspeed(i).suited=[ short-
start;data.halfspeed(i).suited(max([ start+off-
set,1]):min([ finish+offset,len]),:);shortend];
            end %if finish
%           disp('hs suited')
%           keyboard
      end %if ~isempty



end %for
```

Appendix B

Modeling Matlab Scripts

```
function [ alpha, beta, x] =idx(tor, ang)
% [ alpha, beta, x] =idx(tor, ang)
% does hys model id according to Ge and Jouaneh
manualflag=1;
% force inputs to be columns
tor=tor(:);
ang=ang(:);

% input is torque, output is angle

if manualflag
            clg
            plot(tor)
            title('click on the maxes and mins')
            peakinds=round(ginput(14));  % this must
be an even number
            peakinds=peakinds(:,1);
else
            % find peaks of input--these are alpha
values
            sd=sign(diff(ang));
            % peak is where deriv changes sign
            peakinds=find(sd.*[ 0;0;0;0;
sd(1:length(sd)-4)] <0)-4;
            peakinds=[ setdiff(peakinds,peakinds+2);
length(ang)] ;
end %if

% prevent running over array ends at beginning and
end
peakinds(find(peakinds<10))=10;
peakinds(find(peakinds>length(tor)-10))=length(tor)-
10;

% assuming positive initial slope and even number of
peakinds , maxes are odd values of peakinds
%keyboard
maxes=peakinds(2*(0:round(length(peakinds)/2-1))+1);
mins=peakinds(2:2:length(peakinds));

%mins=peakinds(2*(0:round(length(peakinds)/2-1))+1);
%maxes=peakinds(2:2:length(peakinds));

% search nearest 10 points for max
for i=1:length(maxes)
            maxadd=find(tor(maxes(i)+(-
5:1:5))==max((tor(maxes(i)+(-5:1:5)))));
            maxadd=maxadd(1);
            maxes(i)=maxes(i)+maxadd-6;
end %for i

% search nearest 10 points for min
for i=1:length(mins)
            minadd=find(tor(mins(i)+(-
5:1:5))==min((tor(mins(i)+(-5:1:5)))));
            minadd=minadd(1);
            mins(i)=mins(i)+minadd-6;
end %for i


alpha=tor(maxes);

for i=1:length(maxes)
            peakinds=[ peakinds;maxes(i);mins(i)] ;
end %for
peakinds(1:2*length(maxes))=[ ] ;
```

```
d=diff(peakinds);
ncols=max(d(1:2:length(d)))+1;
% allocate x, beta
x=-999*ones(length(maxes),ncols);
beta=x;

% % do id
if maxes(1)<mins(1)
            for i=1:length(maxes)
                        inds=peakinds(2*i-
1):peakinds(2*i);

x(i,1:length(inds))=ang(maxes(i))-ang(inds)';

beta(i,1:length(inds))=tor(inds)';
            end % for

else
            for i=1:length(maxes)-1

inds=peakinds(2*i):peakinds(2*i+1);

x(i,1:length(inds))=ang(maxes(i))-ang(inds)';

beta(i,1:length(inds))=tor(inds)';
            end % for
end

x=[ ang(maxes) x];
beta=[ min(tor)*ones(size(alpha)) beta];

%keyboard
```

```
function [ alpha, beta, x]=idx(tor, ang)
% [ alpha, beta, x]=idx(tor, ang)
% does hys model id according to Ge and Jouaneh
manualflag=1;
% force inputs to be columns
tor=tor(:);
ang=ang(:);

% input is torque, output is angle

if manualflag
            clg
            plot(tor)
            title('click on the maxes and mins')
            peakinds=round(ginput(14));  % this must
be an even number
            peakinds=peakinds(:,1);
else
            % find peaks of input--these are alpha
values
            sd=sign(diff(ang));
            % peak is where deriv changes sign
            peakinds=find(sd.*[ 0;0;0;0;
sd(1:length(sd)-4)]<0)-4;
            peakinds=[ setdiff(peakinds,peakinds+2);
length(ang)];
end %if

% prevent running over array ends at beginning and
end
peakinds(find(peakinds<10))=10;
peakinds(find(peakinds>length(tor)-10))=length(tor)-
10;
```

```
% assuming positive initial slope and even number of
peakinds , maxes are odd values of peakinds
%keyboard
maxes=peakinds(2*(0:round(length(peakinds)/2-1))+1);
mins=peakinds(2:2:length(peakinds));

%mins=peakinds(2*(0:round(length(peakinds)/2-1))+1);
%maxes=peakinds(2:2:length(peakinds));

% search nearest 10 points for max
for i=1:length(maxes)
            maxadd=find(tor(maxes(i)+(-
5:1:5))==max((tor(maxes(i)+(-5:1:5)))));
            maxadd=maxadd(1);
            maxes(i)=maxes(i)+maxadd-6;
end %for i

% search nearest 10 points for min
for i=1:length(mins)
            minadd=find(tor(mins(i)+(-
5:1:5))==min((tor(mins(i)+(-5:1:5)))));
            minadd=minadd(1);
            mins(i)=mins(i)+minadd-6;
end %for i


alpha=tor(maxes);

for i=1:length(maxes)
            peakinds=[peakinds;maxes(i);mins(i)];
end %for
peakinds(1:2*length(maxes))=[];


d=diff(peakinds);
ncols=max(d(1:2:length(d)))+1;

% allocate x, beta
x=-999*ones(length(maxes),ncols);
beta=x;

% % do id
if maxes(1)<mins(1)
            for i=1:length(maxes)
                        inds=peakinds(2*i-
1):peakinds(2*i);

x(i,1:length(inds))=ang(maxes(i))-ang(inds)';

beta(i,1:length(inds))=tor(inds)';
            end % for

else
            for i=1:length(maxes)-1

inds=peakinds(2*i):peakinds(2*i+1);

x(i,1:length(inds))=ang(maxes(i))-ang(inds)';

beta(i,1:length(inds))=tor(inds)';
            end % for
end

x=[ang(maxes) x];
beta=[min(tor)*ones(size(alpha)) beta];

%keyboard



function [m]=fsvolume(coeffs,phiall);
% fsvolume.m
```

```
% integrates volume up and diffs wrt bend angle to
get m(phi)
% according to fay and steele model

%r=coeffs(1);
deadband=coeffs(1);
setpoint=coeffs(2);
r=.139/2; % elbow
if setpoint<0
            setpoint=0;
end %if

if deadband<0
            deadband=0;
end %if
%r=.10;
p=30000;
L=1;
phi0=.001*pi/180;

phiall=phiall-setpoint;
poshalf=phiall(find(phiall>0));
neghalf=phiall(find(phiall<0));
posshift=max([ 1 find(abs(poshalf-
deadband)==min(abs(poshalf-deadband)))] );
negshift=max([ 1 length(neghalf)-
find(abs(neghalf+deadband)==min(abs(neghalf+deadband
)))] );
phishift=[ -
neghalf';.001*ones(posshift+negshift,1);poshalf'];
phishift=phishift(negshift:length(phishift)-
posshift);

phiall=phishift;

i=0;
%for phi=phi0:.01:phi0+.87/2%pi/4
for i=1:length(phiall)
            phi=phiall(i);
            thetastart=0;
            thetaend=phi;

[ theta,vol]=ode45('fsvolumeintbent',[ thetastart
thetaend],0,[ ],phi,r,phi0);
            vol=vol+(L-2*r*(phi-phi0))*pi*r^2;
            volphi(i,:)=[ phi vol(length(vol))];
end %for
%volphi=[ 0 pi*r^2*L;volphi];
warning off
m=-p*[ diff(volphi(:,2))./diff(volphi(:,1))];
warning on
m=[ m(1); m];
m=m-min(m);

%vol=volphi(:,2);
%phi=volphi(:,1);
%m(find(abs(phiall)<(setpoint-deadband)))=0;
%keyboard
%phi=[ flipud(-volphi(:,1))-deadband-
setpoint;volphi(:,1)+deadband-setpoint]

%m=[ flipud(-m);m];

% patch up Nan where diff(allphi)==0
for index=1:length(m)
            % trap first or last cases
            % first is nan
            if index==1 & isnan(m(index))

topind=min(find(~isnan(m(index+1:length(m)))))+index
;
```

```
                        slope=m(topind+1)-m(topind);

m(index)=m(topind)+slope*(topind-index);
            end %if

            if isnan(m(index))
                        if index~=length(m)
                                    bottom=m(index-
1);

topind=min(find(~isnan(m(index+1:length(m)))))+index
;
                                    top=m(topind);

m(index)=bottom+(top-bottom)/(topind-index);

                        else % last is nan

botind=max(find(~isnan(m)));
                                    slope=m(botind)-
m(botind-1);

m(index)=m(botind)+(index-botind)*slope;
                        end %if
            end %if
end %for


oldm=m;


m(1:length(neghalf))=-m(1:length(neghalf));
posm=m(length(neghalf)+1:length(m));
m(length(neghalf)+1:length(neghalf)+posshift+negshif
t)=0;
m=[m(1:length(neghalf)+posshift+negshift);posm];
```

```
m=m(negshift+1:length(m)-posshift-1)';



function
vdiff=fsvolumeintbent(theta,y,flags,phi,r,phi0)

xp=0;
yp=2*r*(1-theta);
yp(find(theta>(phi-phi0)))=2*r*(1-(phi-phi0));
ys=2*r*(1-cos(phi-theta));
xs=2*r*sin(phi-theta);

H=sqrt((xp-xs)^2+(yp-ys)^2);
A=pi*(r^2-(r-H/2)^2);

%vdiff=A*(-H/2+yp);
vdiff=A*r;

%H=(2*r*(1-cos(phi-theta)).^2+(2*r.*(1-phi)-
2*r.*sin(phi-theta)).^2).^.5;
%vdiff=pi*(r^2-(r-H/2)^2)*r;



% mpsmodel.m
% calculates moment curvature relation for
pressurized beam, as in
% main, peterson, strauss paper

clear k b theta0

% set parameters
```

```
p=30000;  % pa
r=.14; % meters
nu=.5;  % poisson ratio
E=13500; % newton/meter
i=0;
a1=.20; % meters
a2=.15;
a3=.20;

for M=1:.5:160
            i=i+1;
            % find theta0
            thetatest=(0:.5:179)*pi/180;
            mpr3num=(pi/2*((pi-
thetatest)+sin(thetatest).*cos(thetatest))-nu.*((pi-
thetatest).^2-(pi-
thetatest).*sin(thetatest).*cos(thetatest)-
(2.*sin(thetatest)).^2));
            mpr3den=sin(thetatest)+(pi-
thetatest).*cos(thetatest);
            mpr3=mpr3num./mpr3den;
            thetaind=find(abs(mpr3-(M/p/
r^3))==min(abs(mpr3-(M/p/r^3))));
            theta0(i)=thetatest(thetaind);

            % find k
            k(i)=(M-2*nu*p*r^3.*sin(theta0(i)))./
(E*r^3.*((pi-
theta0(i))+sin(theta0(i)).*cos(theta0(i))));

end %for


% find bending angle b as f(k)  in degrees
            b1=k*a1*180/pi;
            b2=k*a2*180/pi;
            b3=k*a3*180/pi;

% moment for plotting
M=1:.5:160;

clg
plot(b1,M,'b-')
hold on
%plot(eangmean-60,(etormean-1.28),'o') % this
variable comes from elbowbeamcomp625.mat
%plot(kfang-mean(kfang),kftor-14.4,'o'); % variables
from data:reduced_data:kneetorques.mat
plot(haang-8,hator,'o'); % variables from
data:reduced_data:hatorques.mat
% plot(b2,M/4,'r--')
% plot(b3,M/4,'r-.')
%
plot(-b1,-M,'b-')
% plot(-b2,-M/4,'r--')
% plot(-b3,-M/4,'r-.')
%
 %legend('a=10cm','a=15cm','a=20cm')
set(gca,'fontsize',18)
grid on
hleg=legend('Beam model','Hip abduction data');
set(hleg,'position',[.1286 .8199 .3954 .1041])
xlabel('Bending angle')
ylabel('Bending moment (Nm)')
 axis([0 10 0,160])
% set(gca,'xtick',[-50 -25 0 25 50])
%axis([0 10 min(M) max(M)])
```

Appendix C

Work envelope Matlab scripts

```
%function arminvkin(void)
% arminvkin.m
% inverse kinematics of arm, using Korein's method
% Korein, JU A Geometric Investigation of Reach p
57-75


outpath='''Macintosh
HD:Users:Patricia:Documents:scripts:''';
cl=clock;
cl=num2str(fix(cl(2:5)));
cl(isspace(cl))=[];

outfile=[ 'canreach',cl,'.mat']

clear n bestsfa bestsaa bestefa besthra bestw
bestmetric metric beste e
e=[ 0;0;0] ';
canreach=zeros(25,25,25);
canreachind=0;
changecount1=0;
changecount2=0;
outcount=0;
xind=0;
yind=0;
zind=0;
tau=zeros(10,1);

%viewpoint offsets
viewx=-.205;
viewy=-.1;
viewz=.20;

% angle limits
hrmin=0;
hrmax=180;
elbowmax=180;
elbowmin=60;
sfmax=85;
sfmin=20;
samax=70;
samin=30;


tic
% link parameters

% % 50th percentile arm lengths
% lu=.366;
% ll=.3825;

% 95th percentile arm lengths
lu=.394;
ll=.416;

% big slow loop

for wx=-lu-ll:.0624:ll+lu+.0624
            xind=xind+1
            for wy=-lu-ll:.0624:ll+lu+.0624
                        yind=yind+1;
                        for wz=-lu-
ll:.0624:ll+lu+.0624
                                    zind=zind+1;
```

```
w=[ wx,wy,wz] ;
for index=1:10
taustart=hrmin*pi/180;
taustop=hrmax*pi/180;

% check that target position is in fov--doing this
without changing 23x23x23 indexing
% first convert w to eyepoint coords
if (wy+viewy)<0  |
~infov(wx+viewx,(wz+viewz),wy+viewy)
            canreach(xind,yind,zind)=0;
            outcount=outcount+1;
            break
end %if


if norm(w)>lu+ll

            canreach(xind,yind,zind)=0;
            outcount=outcount+1;
            break
else


tau(index)=taustart+(taustop-taustart)/10*index;  %
twist angle

% find elbow angle epsilon and shoulder angle sigma
based on radius to target
% law of cosines
coseps=(lu^2+ll^2-norm(w)^2)/(2*lu*ll);
epsilon=acos(coseps);

% % force epsilon to be in 1st or 2nd quadrants
(0<epsilon<pi)
if epsilon>pi
            epsilon=2*pi-epsilon;
end %if

if epsilon > 2*pi
            epsilon=epsilon-2*pi;
end %if

% if epsilon<elbowmin*pi/180 | epsilon >
elbowmax*pi/180
%           break
% end %if

% really want angle between upper arm _extension_
and lower arm
% epsilon=0 -> arm is straight
%epsilon=pi-epsilon;

% shoulder angle sigma-never used again
cossig=(norm(w)^2+lu^2-ll^2)/(2*norm(w)*lu);
sigma=acos(cossig);

% wrist position after elbow and shoulder bent <--
no, only elbow
% sin(epsilon) always >0, so this always moves hand
towards body
w1=lu*[ 0;0;1] +ll*[ sin(epsilon);0;-cos(epsilon)] ;

% rotate upper arm by tau and get new wrist position
w2=rot([ 0;0;1] ,tau(index))*w1;

% define shoulder rotation axis needed
a=[ (w(2)-w2(2));w2(1)-w(1);0] ;  % are subscripts
right here? YES
a=a/norm(a);

% find rotation angle eta needed
```

```
projw=cross(cross(a,w),a);
projw2=cross(cross(a,w2),a);

etasin=norm(cross(projw,projw2)/norm(projw)/
norm(projw2));
etacos=dot(projw,projw2)/norm(projw)/norm(projw2);
eta=atan2(etasin,etacos);

% determine sign for eta
if dot(cross(w,w2),a)>0
            eta=-eta;
end %if


% now get elbow position for this value of tau
etwist=rot([ 0;0;1],tau(index))*[ 0;0;lu];
e(index,:)=(rot(a,eta)*etwist)';   % I'm not sure
this is right


% finally, pick off the phi, sa and sf angles for
this configuration
% old
% phi(index)=atan2(e(2),e(1));
% sf(index)=atan2(e(3),e(2));
% sa(index)=atan2(e(3),e(1));

% robot's shoulders are rotated inward 10 degrees,
so rotate e out before
% picking off sf
%%%%%%%% rotated back afterwards, so comment out **
both ** if necessary
e(index,:)=e(index,:)*rot([ 0;0;1],-15*pi/180);


% new 5/19/01
sf(index)=atan2(e(index,2),e(index,3));
%sa(index)=atan2(-e(index,1),e(index,3));
if (sf(index)~=0)
            sa(index)=atan2(-
sin(sf(index))*e(index,1),e(index,2));
else
            sa(index)=atan2(-sin(sf(index)+pi/
4)*e(index,1),e(index,2));
end %if
%keyboard

% now rotate elbow position back
e(index,:)=e(index,:)*rot([ 0;0;1],15*pi/180);


% wrist position
wact=rot(a,eta)*w2;

% check for compatibility
if abs(norm(wact'-e(index,:))-ll)>2*eps
            %error('Lower arm length ~=ll')
            break
end %if

if abs(norm(e(index,:))-lu)>2*eps
            %error('Upper arm length ~=lu')
            break
end %if

% test if hand on target
wtest=rot(a,eta)*w2;

if abs(sum(wtest-w'))>50*eps
            error('hand not on target')
end %if
```

```
epsilon=epsilon*180/pi;
goodangles=(find((epsilon< elbowmax)&(sf>sfmin*pi/
180) & (sf<sfmax*pi/180)& (sa>samin*pi/180) &
(sa<samax*pi/180)));
% goodangles never gets used again

end %if radius is ok

end % for twist loop


if exist('epsilon')
            [bestindex metric]=choosebest(sf, sa,
tau,epsilon*pi/180*ones(size(sf)));
                        if bestindex

canreachind=canreachind+1;

bestsfa(canreachind)=sf(bestindex);

bestsaa(canreachind)=sa(bestindex);

besthra(canreachind)=tau(bestindex);

bestefa(canreachind)=epsilon;

bestw(canreachind,:)=w;

beste(canreachind,:)=e(bestindex,:);

bestmetric(canreachind)=metric;


n(canreachind)=(size(canreach,1))^2*(zind-
1)+(size(canreach,1))*(yind-1)+xind;

canreach(xind,yind,zind)=1;

%                                   if
bestsaa(canreachind)>80*pi/180
%
keyboard
%                                   end %if
                        else

canreach(xind,yind,zind)=0;
                        end %if bestindex
            end %if sf

sa=zeros(10,1);
tau=zeros(10,1);
sf=sa;
clear epsilon
end %for z
zind=0;

end %for y
yind=0;

end % for x
metric=canreach;
metric(n)=bestmetric;

eval([ 'save ',outpath,outfile,' bestsfa bestsaa
besthra bestefa bestw beste n metric canreach']);

toc
```

```
function out=infov(x0,z0,front)
% determines whether point is in EMU fov, based on
nsts07700
% all dimensions in **meters**

if front<=0
            error('front must be >0')
end %if

up=70*pi/180;
down=70*pi/180;
left=85*pi/180;
right=85*pi/180;
leftup=62*pi/180;
rightup=62*pi/180;
leftdown=85*pi/180;
rightdown=85*pi/180;

% define vertices
x1=0;
z1=-front*tan(up);

x2=1/sqrt(2)*front*tan(rightup);
z2=-x2;

x3=front*tan(right);
z3=0;

x4=1/sqrt(2)*front*tan(rightdown);
z4=x4;

x5=0;
z5=front*tan(down);

x6=-1/sqrt(2)*front*tan(leftdown);
z6=-x6;

x7=-front*tan(left);
z7=0;

x8=-1/sqrt(2)*front*tan(leftup);
z8=x8;

% divide big polygon into 3 convex quadrilaterals
% polygon #1:  7 8 5 6
% polygon #2:  8 1 2 5
% polygon #3:  2 3 4 5
% call in4poly to determine whether x0,z0 is in any
of the 3 quadrilaterals

out1=in4poly(x0,z0,x7,z7,x8,z8,x5,z5,x6,z6);
out2=in4poly(x0,z0,x8,z8,x1,z1,x2,z2,x5,z5);
out3=in4poly(x0,z0,x2,z2,x3,z3,x4,z4,x5,z5);

%keyboard
out=(out1+out2+out3);  % returns 1 if sum>=0
%keyboard

%h=plot([ x1 x2 x3 x4 x5 x6 x7 x8 x1],-[ z1 z2 z3 z4
z5 z6 z7 z8 z1] )
%axis([ -1 1 -1 1] )
%hold on
%keyboard
%plot(x0,-z0,'o')
%pause
%clg



function out=in4poly(x0,y0,x1,y1,x2,y2,x3,y3,x4,y4)
```

```
% out=in4poly(x0, y0, x1,y1,x2,y2,x3,y3,x4,y4)
% determines if point x0,y0 is in polygon bounded by
x1,y1,x2,y2,x3,y3,x4,y4
% returns 1 if point is in polygon, 0 if not
% vertices specified should be in consecutive order

% see notebook 6 pp.110-114, 118-119

warning off  % suppress division by zero since it's
dealt with otherwise
out=1;
bdry=0;
% find equations for polygon side segments
ma=(y2-y1)/(x2-x1);
ba=y2-ma*x2;

mb=(y3-y2)/(x3-x2);
bb=y3-mb*x3;

mc=(y4-y3)/(x4-x3);
bc=y4-mc*x4;

md=(y1-y4)/(x1-x4);
bd=y1-md*x1;

% check that polygon vertices are valid--not
collinear
if ma==mb & mb==mc & mc==md
            error('vertices are collinear')
end %if

% don't allow horizontal or vertical sides
if sum(isnan([ ma mb mc md] )) | sum(isinf([ ma mb mc
md] ))
            error('vertical or horizontal side')
end %if

% define lines between x0,y0 and each vertex
mv1=(y0-y1)/(x0-x1);
bv1=y0-mv1*x0;

mv2=(y0-y2)/(x0-x2);
bv2=y0-mv2*x0;

mv3=(y0-y3)/(x0-x3);
bv3=y0-mv3*x0;

mv4=(y0-y4)/(x0-x4);
bv4=y0-mv4*x0;

% determine whether x0,y0 is on boundary

if abs(y0-ma*x0-ba)<2*eps
            bdry=1;
            if y0>min([ y1 y2] ) & y0<max([ y1 y2] )
                        out=1;
            else
                        out=0;
            end %if
end %if

if abs(y0-mb*x0-bb)<2*eps
            bdry=1;
            if y0>min([ y2 y3] ) & y0<max([ y2 y3] )
                        out=1;
            else
                        out=0;
            end %if
end %if

if abs(y0-mc*x0-bc)<2*eps
            bdry=1;
```

```
            if y0>min([ y3 y4] ) & y0<max([ y3 y4] )
                        out=1;
            else
                        out=0;
            end %if
end %if

if abs(y0-md*x0-bd)<2*eps
            bdry=1;
            if y0>min([ y1 y4] ) & y0<max([ y1 y4] )
                        out=1;
            else
                        out=0;
            end %if
end %if

% find intersections

if ~bdry
            % don't need to check for intersections
with sides that are touching
            % the current vertex because you know
the point isn't on a side

            % vertex 1 -- check b and c
            xint1b=(bv1-bb)/(mb-mv1);
            xint1c=(bv1-bc)/(mc-mv1);

            % vertex 2 -- check d and c
            xint2d=(bv2-bd)/(md-mv2);
            xint2c=(bv2-bc)/(mc-mv2);

            % vertex 3 -- check a and d
            xint3a=(bv3-ba)/(ma-mv3);
            xint3d=(bv3-bd)/(md-mv3);

            % vertex 4 -- check a and b
            xint4a=(bv4-ba)/(ma-mv4);    % changed
from mv1 to mv4 5/13/01
            xint4b=(bv4-bb)/(mb-mv4);   % changed
from bc to bb in denom 5/13/01

            % check if intersections fall within
segment
            if xint1b>min([ x0 x1] ) & xint1b<max([ x0
x1] )
                        out=0;
            end %if


            if xint1c>min([ x0 x1] ) & xint1c<max([ x0
x1] )
                        out=0;
            end %if

            if xint2d>min([ x0 x2] ) & xint2d<max([ x0
x2] )
                        out=0;
            end %if

            if xint2c>min([ x0 x2] ) & xint2c<max([ x0
x2] )
                        out=0;
            end %if

            if xint3a>min([ x0 x3] ) & xint3a<max([ x0
x3] )
                        out=0;
            end %if

            if xint3d>min([ x0 x3] ) & xint3d<max([ x0
x3] )
```

```
                    out=0;
            end %if

            if xint4a>min([ x0 x4] ) & xint4a<max([ x0
x4] )
                    out=0;
            end %if

            if xint4b>min([ x0 x4] ) & xint4b<max([ x0
x4] )
                    out=0;
            end %if

end %if
%keyboard


warning backtrace  % un-suppress warnings




function [ index, metric] =choosebest(sfa,saa,hra,efa)
% index=choosebest(sfa,saa,hra,efa)
% Inputs are nx1 vectors of possible joint angles.
Returns index of
% angle set that minimizes torque metric and doesn't
exceed any torque limits.

metric=zeros(size(sfa));

% 10% of max iso-velocity torques (N*m)
% sftlim=2*7.7;
% satlim=2*5.4;
% hrtlim=2*3.5;
% eftlim=2*4;

% these are 'transformed' isokinetic #'s
% sftplim=12.3;
% sftnlim=19.3;
%
% satplim=10.2;
% satnlim=14.3;
%
% hrtplim=5.1;
% hrtnlim=9.5;
%
% eftplim=10.2;
% eftnlim=10.2;

%
% this is chaffin's data for 50%ile male, no,
actually 95th!
% % base values are 15%
% sftplim=100/15*15.2;
% sftnlim=100/15*17.3;
%
% satplim=100/15*15.5;
% satnlim=100/15*17.9;
%
% hrtplim=100/15*7.7;
% hrtnlim=100/15*12.5;
%
% eftplim=100/15*16.7;
% eftnlim=100/15*10.1;


% % % this is chaffin's data for 50%ile male, no,
actually 95th!
% % % base values are 15%
```

```
sftplim=100/15*10.7;
sftnlim=100/15*10.1;

satplim=100/15*10.1;
satnlim=100/15*13.8;

hrtplim=100/15*5.0;
hrtnlim=100/15*7.8;

eftplim=100/15*11.6;
eftnlim=100/15*6.9;
```

```
% % fake torque limits to get plot of field of view
% sftplim=100;
% sftnlim=-100;
%
% satplim=100;
% satnlim=-100;
%
% hrtplim=100;
% hrtnlim=-100;
%
% eftplim=100;
% eftnlim=-100;

% find torques
sfa=sfa*180/pi;
saa=saa*180/pi;
hra=hra*180/pi;
efa=efa*180/pi;
```

```
[sft,sat,hrt,eft]=suittorques(sfa,saa,hra,efa);

sfpos=find(sft>0);
sfneg=find(sft<=0);
msf(sfpos)=sft(sfpos)/sftplim;
msf(sfneg)=-sft(sfneg)/sftnlim;

sapos=find(sat>0);
saneg=find(sat<=0);
msa(sapos)=sat(sapos)/satplim;
msa(saneg)=-sat(saneg)/satnlim;

hrpos=find(hrt>0);
hrneg=find(hrt<=0);
mhr(hrpos)=hrt(hrpos)/hrtplim;
mhr(hrneg)=-hrt(hrneg)/hrtnlim;

efpos=find(eft>0);
efneg=find(eft<=0);
mef(efpos)=eft(efpos)/eftplim;
mef(efneg)=-eft(efneg)/eftnlim;
```

```
%keyboard
indices=find(msf<1 & msa<1 & mhr<1 & mef<1);
if ~isempty(indices)

notindices=setdiff((1:length(sfa)),indices);
%                          keyboard

metric(indices)=msf(indices)+msa(indices)+mhr(indice
s)+mef(indices);
```

```
            metric(notindices)=4;

            index=(find(metric==min(metric)));
            metric=min(metric);
            % force a choice, even if 2 or more
configs at min
            % choose the middle one
            if length(index)>1

index=index(round(length(index)/2));

metric=metric(round(length(metric)/2));
            end %if
else
            index=0;
            metric=4;
end %if


function
[ sft,sat,hrt,eft]=suittorques(sfa,saa,hra,efa)
%  [ sft,sat,hrt,eft]=suittorques(sfa,saa,hra,efa)
%  Calculates suit torque predictions for shoulder
flex, shoulder abduction,
%  humerus rotation, elbow flexion
%  Can be called with nx1 or 1xn vector inputs

% regression coeffs from 12/19/00
% sft=.25582*sfa-14.873;
% sat=.8268*saa-49.965;
% %sat=.5768*saa-26.06;
% hrt=-.065117*(hra-90);
% p5=[ 2.9155e-8 -8.8818e-6 1.0533e-3 -5.9161e-2
1.5992 -1.5016e1];
% eft=polyval(p5,180-efa);

% coeffs from 5/19/01  also need to redo elbow
sft=-23.43+.2220*sfa;
%sat=-48.8+.832*saa;

% piecewise linear sa fit
sat(find(saa>50))=-76.5595+1.1764*saa(find(saa>50));
% 5/20/01
sat(find(saa<=50))=-27.2+.3027*saa(find(saa<=50));

hrt=-4.5838-.189*(hra-90);
eft=-9.04+.1551*(180-efa);
% p5=[ 2.9155e-8 -8.8818e-6 1.0533e-3 -5.9161e-2
1.5992 -1.5016e1];
% eft=polyval(p5,180-efa);


% smoothwe.m
% test method for smoothing out work envelope
% look at one zslice
% thicken the 15% area by less than 25%
% subtract stuff that's not in 30% area
% use contour to see how many verts needed to bound
it

clear


% load 15% and 30% files
% use size 50, strength 95
eval([ 'load canreach621317.mat'] ) % 15%
canreach15=canreach;

eval([ 'load canreach621320.mat'] ) %30%
canreach30=canreach;

nvertsbefz=zeros(size(canreach15,1),1);
```

```
nvertsaftezr=nvertsbefz;
percentupallz=nvertsbefz;


for slice=1:size(canreach15,1);

            slice15=zslice(canreach15,slice);
            slice30=zslice(canreach30,slice);

            if sum(sum(slice15))~=0
                        disp([ 'slice
',num2str(slice)] )



nvertsbefz(slice)=nvertcontour(slice15);
                        disp([ 'starting number of
vertices ',num2str(nvertsbefz(slice))] )

                        percentupz=0;
                        count=0;
                        percentupold=1;

%                       disp('before image')
%                       keyboard

                        % first fill holes
                        while
percentupold~=percentupz

percentupold=percentupz;

filledslice=bwmorph(slice15,'bridge',1);

filledslice=bwmorph(filledslice,'fill',1);

percentupz=(bwarea(filledslice)-bwarea(slice15))/
bwarea(slice15);
                        end %while

                        percentupz=0;
                        percentupold=1;

%                       disp('filled holes')
%                       keyboard


                        % use low pass filter to
smooth image
                        h=fspecial('gaussian',3,2);

filteredslice=conv2(filledslice,h);
%
%                       disp('filtered image')
%                       keyboard

                        % convert back to bw with
variable threshold
                        for index=1:30

thickerslice=im2bw(filteredslice,.03*index);

thickerslice=thickerslice(2:length(thickerslice)-
1,2:length(thickerslice)-1);

                                    % then clean up

thickerslice=bwmorph(thickerslice,'spur',1);

thickerslice=bwmorph(thickerslice,'clean',1);
```

```
thickerslice=bwmorph(thickerslice,'fill',10);


thickerslice=double(thickerslice);

slice30=double(bwmorph(slice30,'fill',1));

thickerslice=thickerslice & slice30;


percentupz(index)=(bwarea(thickerslice)-
bwarea(slice15))/bwarea(slice15);

threshold(index)=.03*index;

nverts(index)=nvertcontour(thickerslice);

                        end %for

                        % choose threshold that
                        % 1. adds 30% or less points
and
                        % 2. gives fewest vertices
                        % if cond's 1 and 2 can't be
met, don't change anything


percentright=find(percentupz>0 & percentupz<=.25);

                        if ~isempty(percentright)

goodnverts=find(nverts(percentright)==min(nverts(per
centright)));

goodnverts=goodnverts(1);

thresh=threshold(percentright(goodnverts));

thickerslice=im2bw(filteredslice,thresh);

thickerslice=bwmorph(thickerslice,'spur',1);

thickerslice=bwmorph(thickerslice,'clean',1);

thickerslice=bwmorph(thickerslice,'fill',10);

thickerslice=thickerslice(2:length(thickerslice)-
1,2:length(thickerslice)-1);
                        else

thickerslice=filledslice;
                        end %if



                        slice15thick=thickerslice;

                        % subtract points not in 30%
slice

slice15thick=double(slice15thick);

slice30=double(bwmorph(slice30,'fill',1));
                        slice15thicksub=slice15thick
& slice30;


                        % look at contour again

nvertsafterz(slice)=nvertcontour(slice15thicksub);
```

```
                        disp([ 'ending number of
vertices ',num2str(nvertsafterz(slice))] )

%                       % plot both
%                       clg
%                       subplot(121)
%
imshow(imresize(slice15,10));
%                       title('before')
%
%                       subplot(122)
%
imshow(imresize(slice15thicksub,10));
%                       title('after')
%                       pause
%
                        if ~isempty(percentright)
                                    disp([ 'percent
increase
=',num2str(100*percentupz(percentright(goodnverts)))
] )

percentupallz(slice)=percentupz(percentright(goodnve
rts));
                        else
                                    disp([ 'percent
increase =0'] )

percentupallz(slice)=0;
                        end % if

                        % for z slice

canreachcomposite(:,:,slice)=fliplr(flipud(double(sl
ice15thicksub')));
                                    % for y slice

%canreachcomposite(:,slice,:)=fliplr(double(slice15t
hicksub'));
            else

canreachcomposite(:,:,slice)=slice15;
            end %if sum(sum(slicez))~=0

end %for



%%%%%%%%% make 2nd pass in y direction
% load 15% and 30% files
% use size 50, strength 95
%eval([ 'load canreach621317.mat'] ) % 15%
%canreach15=zslice(canreachcomposite,slice);
canreach15=canreachcomposite;

eval([ 'load canreach621320.mat'] ) %30%
canreach30=canreach;

nvertsbefy=zeros(size(canreach15,1),1);
nvertsaftery=nvertsbefy;
percentupally=nvertsbefy;


for slice=1:size(canreach15,1);

            slice15=yslice(canreach15,slice);
            slice30=yslice(canreach30,slice);

            if sum(sum(slice15))~=0
```

```
                        disp([ 'slice
',num2str(slice)])


nvertsbefy(slice)=nvertcontour(slice15);
                        disp([ 'starting number of
vertices ',num2str(nvertsbefy(slice))])

                        percentupy=0;
                        count=0;
                        percentupold=1;

%                       disp('before image')
%                       keyboard

                        % first fill holes
                        while
percentupold~=percentupy

percentupold=percentupy;

filledslice=bwmorph(slice15,'bridge',1);

filledslice=bwmorph(filledslice,'fill',1);

percentupy=(bwarea(filledslice)-bwarea(slice15))/
bwarea(slice15);
                        end %while

                        percentupy=0;
                        percentupold=1;

%                       disp('filled holes')
%                       keyboard
```

```
                        % use low pass filter to
smooth image
                        h=fspecial('gaussian',3,2);

filteredslice=conv2(filledslice,h);
%
%                       disp('filtered image')
%                       keyboard

                        % convert back to bw with
variable threshold
                        for index=1:30

thickerslice=im2bw(filteredslice,.03*index);

thickerslice=thickerslice(2:length(thickerslice)-
1,2:length(thickerslice)-1);

                                        % then clean up

thickerslice=bwmorph(thickerslice,'spur',1);

thickerslice=bwmorph(thickerslice,'clean',1);

thickerslice=bwmorph(thickerslice,'fill',10);


thickerslice=double(thickerslice);

slice30=double(bwmorph(slice30,'fill',1));

thickerslice=thickerslice & slice30;
```

```
percentupy(index)=(bwarea(thickerslice)-
bwarea(slice15))/bwarea(slice15);

threshold(index)=.03*index;

nverts(index)=nvertcontour(thickerslice);

                        end %for

                        % choose threshold that
                        % 1. adds 30% or less points
and
                        % 2. gives fewest vertices
                        % if cond's 1 and 2 can't be
met, don't change anything


percentright=find(percentupy>0 & percentupy<=.25);

                        if ~isempty(percentright)

goodnverts=find(nverts(percentright)==min(nverts(per
centright)));

goodnverts=goodnverts(1);

thresh=threshold(percentright(goodnverts));

thickerslice=im2bw(filteredslice,thresh);

thickerslice=bwmorph(thickerslice,'spur',1);

thickerslice=bwmorph(thickerslice,'clean',1);

thickerslice=bwmorph(thickerslice,'fill',10);

thickerslice=thickerslice(2:length(thickerslice)-
1,2:length(thickerslice)-1);
                        else

thickerslice=filledslice;
                        end %if



                        slice15thick=thickerslice;

                        % subtract points not in 30%
slice

slice15thick=double(slice15thick);

slice30=double(bwmorph(slice30,'fill',1));
                        slice15thicksub=slice15thick
& slice30;


                        % look at contour again

nvertsaftery(slice)=nvertcontour(slice15thicksub);
                        disp([ 'ending number of
vertices ',num2str(nvertsaftery(slice))] )

%                       % plot both
%                       clg
%                       subplot(121)
%
imshow(imresize(slice15,10));
%                       title('before')
%
%                       subplot(122)
```

```
%
imshow(imresize(slice15thicksub,10));
%                       title('after')
%                       pause

                        if ~isempty(percentright)
                                    disp([ 'percent
increase
=',num2str(100*percentupy(percentright(goodnverts)))
])

percentupally(slice)=percentupy(percentright(goodnve
rts));
                        else
                                    disp([ 'percent
increase =0'])

percentupally(slice)=0;
                        end % if

                        % for z slice

%canreachcomposite(:,:,slice)=fliplr(flipud(double(s
lice15thicksub')));

                        % for y slice

canreachcomposite(:,slice,:)=fliplr(double(slice15th
icksub))';
            else

canreachcomposite(:,:,slice)=slice15;
            end %if sum(sum(slicez))~=0

end %for
```

```
function nverts=nvertcontour(z)
% finds number of vertices in contour needed to
bound foreground area
% in binary image z

if sum(sum(z))~=0

            c=contourc(z,1);

            % deal with case when there are more
than 1 contour lines
            clen=size(c,2);
            lenfirst=c(2,1);

            if lenfirst==clen-1  % true if one
contour
                        c=c(:,2:clen);
            else
            %error('more than one contour')
            % don't know if commented out stuff is
necessary or works
            %           indend=0;
            %           while indend<clen
            %
indstart=indend+2;
            %
indend=indend+c(2,indend+1)+1;
            %                       c=[ c
c(:,indstart:indend)];
            %           end %while

            % choose longest contour to look at

            %index=lenfirst+2;
            index=1;
```

```
            indices=index;
            while index<clen

nextindex=index+c(2,index)+1;
                        indices=[ indices nextindex];
                        index=nextindex;
            end %while
            lengths=c(2,indices(1:length(indices)-
1));

indstart=indices(find(lengths==max(lengths)))+1;
            indstart=indstart(1);
            indend=indstart+c(2,indstart-1)-1;

            c=c(:,indstart:indend);

            end %if

            %keyboard
            warning off
            dydx=diff(c(2,:))./diff(c(1,:));
            warning on
            dydx(find(isinf(dydx)))=1000;
            d2ydx2=diff(dydx);
            nverts=sum(d2ydx2~=0);

else
            nverts=0;
end %if



% plotreachenv.m
```

```
% makes presentable 3d plots of the reach envelope
from reachenv.m
% loads file reachenv829.mat and plot comes from
variable canreach
clg
hold on
cols=[ 'rgbcmyk'];
%cols=[ 'rbrbrbr'];
%width=55; %old, based on suit clearances?
%width=44.3;% m95
%width=41.1;%m50
%width=39; %f95
width=35.7; %f50


%colormap lines
ind2cm=6.24;  % scale and bias for converting index
to distance in cm
indoff=-11.5;
canreachsize=size(canreach,1);
indstart=floor((canreachsize-1)/2);
indend=canreachsize-indstart-1;

%x=-12*6.24:3/2*6.24:2*11*6.24+3/2*6.24;
x=-indstart*6.24:6.24:indend*6.24;
y=x;
%z=x;


[x,y]=meshgrid(x,y);
%[x,z]=meshgrid(x,z);
%eval('load reachenv829.mat')

for ind=1:1:canreachsize  % make slices in z
direction
            index=num2str(ind);
```

```
eval([ 'slicez',index,'=reshape(canreach(:,:,',index,
'),canreachsize,canreachsize);']); % one slice in z

%eval([ 'slicey',index,'=reshape(canreach(:,',index,'
,:),24,24);']); % slice in y


            % check to see if any points in slice
are in rs

eval([ 'currentslicer=slicez',index,';']);
            currentslicel=flipud(currentslicer);

            points(ind)=sum(sum(currentslicer));
            if sum(sum(currentslicer))~=0;
                        [ c,v] =contour3(-y,x,-
5*(ind+indoff)*ind2cm*currentslicer,-
(ind+indoff)*ind2cm*[ 1 1],cols(rem(ind,7)+1));
                        set(v,'linewidth',2)
                        [ c2,v2] =contour3(-y-
width,x,-5*(ind+indoff)*ind2cm*currentslicel,-
(ind+indoff)*ind2cm*[ 1 1],cols(rem(ind,7)+1));
                        set(v2,'linewidth',2)
                        view(140,20)

%                       [ c,v] =contour3(-
5*(ind+indoff)*ind2cm*currentslicer,x,z,-
(ind+indoff)*ind2cm* [ 1 1],cols(rem(ind,7)+1));
%                       set(v,'linewidth',2)
%                       [ c2,v2] =contour3(-y-55,x,-
5*(ind+indoff)*ind2cm*currentslicel,-
(ind+indoff)*ind2cm*[ 1 1],cols(rem(ind,7)+1));
%                       set(v2,'linewidth',2)
%                       view(140,20)
                        %pause
                        %(ind+indoff)*ind2cm*[ 1 1]
                        %keyboard

            end %if

end %for
grid on
%plot3(0,0,0,'ro')
axis square
set(gca,'fontsize',18)
% set(gca,'xtick',[ -50 -25 0 25 50] )
% set(gca,'ytick',[ -50 -25 0 25 50] )
% set(gca,'ztick',[ -50 -25 0 25 50] )


drawastrocyl(width);
% colormap lines
%axis([ -150,125,-100,175,-150,125] )
axis([ -150 125 -75 150 -150 75] )
set(gca,'xtick',[ -100 -50 0 50 100] )
set(gca,'ytick',[ -100 -50 0 50 100] )
set(gca,'ztick',[ -150 -100 -50 0 50] )
spin(20)
pause(1)
% view(40,20)
xlabel('Left-right (cm)')
ylabel('Front-back (cm)')
zlabel('Vertical (cm)')



function drawastrocyl(torsowidth)
```

```
% uses drawbox, drawsphere,drawcyl to draw astro and
reach envelope
% uses cylinders for arms and legs
% set sizes for everything


% set geometry

% body segment angles
sf=36;
sa=39;
ef=58;
hf=52;
kf=47;
af=21;

% torso
torsoheight=50;
%torsowidth=55;
torsodepth=40;%torsowidth*.8;
torsox=90;
torsoz=0;

% torsocenter=[ -.5*torsowidth -.5*torsodepth -
.5*torsoheight];
torsocenter=[ -.5*torsowidth -.45*torsodepth -
.5*torsoheight];



% head
headradius=17;
headcenter=[ -.5*torsowidth -.5*torsodepth
.6*headradius];
```

```
% plss
plssheight=80;
plsswidth=60;
plssdepth=20;
plssx=torsox;
plssz=torsoz;
plsscenter=[ torsocenter(1) torsocenter(2)-
torsodepth/2-plssdepth/2 torsocenter(3)+10];


% arm

armwidth=15;
lu=36.6;
ll=38.25;%+armwidth/2;

ua1x=sf-90;
ua1z=-sa;
ua2x=ua1x;
ua2z=sa;

la1x=ua1x+ef;
la1z=0;
la2x=la1x;
la2z=la1z;

uaend=lu*rot([ 1;0;0],(ua1x)*pi/
180)*rot([ 0;0;1],ua1z*pi/180)*[ 0;1;0];
uacenter1=uaend'/2-[ 0 0 armwidth/2];
uacenter2=uacenter1+[ -torsowidth-uacenter1(1)-
armwidth*.75 0 0];

lacenter1=((uaend-[ 0 0 armwidth/
2] ')+.5*ll*rot([ 1;0;0],la1x*pi/
180)*rot([ 0;0;1],la1z*pi/180)*[ 0;1;0] )';
```

```
lacenter2=lacenter1+[ -torsowidth-2*lacenter1(1) 0
0] ;

%keyboard

%uacenter1=[ .5*armwidth -.5*armwidth -.5*lu] ;
% uacenter1=[ 0 0 -.5*lu] ;
% lacenter1=[ 0 .5*ll -lu-.5*armwidth] ;
%
% uacenter2=[ uacenter1(1)-torsowidth uacenter1(2)
uacenter1(3)] ;
% lacenter2=[ lacenter1(1)-torsowidth lacenter1(2)
lacenter1(3)] ;


% leg
leglength=90;
legwidth=20;


ul1x=-90+hf;
ul1z=0;
ul2x=ul1x;
ul2z=ul1z;

ll1x=ul1x+kf-90;
ll1z=0;
ll2x=ll1x;
ll2z=ll1z;

%legcenter=[ torsocenter(1) torsocenter(2)+leglength/
4*sin(hf*pi/180) torsocenter(3)-.5*torsoheight-
.5*leglength+0*legwidth*sin(pi/
180*hf)+.25*leglength*cos(hf*pi/180)] ;
ulegcenter=[ torsocenter(1)-legwidth/2
torsocenter(2)+leglength/4*sin(hf*pi/180)
torsocenter(3)-.5*torsoheight-
.5*leglength+0*legwidth*sin(pi/
180*hf)+.25*leglength*cos(hf*pi/180)+leglength/4] ;
llegcenter=ulegcenter+[ 0 +leglength/4*cos(hf*pi/
180)-leglength/4*cos(ll1x*pi/180)+legwidth/
2*sin(hf*pi/180) -leglength/4*sin(hf*pi/180)-
leglength/4*sin(ll1x*pi/180)+legwidth/2*cos(hf*pi/
180)-leglength/2] ;
% foot
footlength=25+legwidth*.5;
footwidth=15;

f1x=ll1x+90-af;
f1z=0;
f2x=f1x;
f2z=f1z;

footcenter=[ llegcenter(1)+legwidth/2
llegcenter(2)+.5*footlength llegcenter(3)-
.25*leglength-1/3*footwidth] ;

% nasa work envelope
reradius=26/39*100/2;
redepth=18/39*100;

recenter=[ -torsowidth/2 torsodepth/2+5/39*100 0] ;

% now draw everything

% head
hhead=drawsphere(headradius,headcenter);

% plss
%hplss=drawbox([ plsswidth plssdepth
plssheight],plsscenter);
```

```
hplss=drawellcylinder(plsswidth/2, plssdepth/2,
plssheight,plsscenter,1);

% torso
%drawbox([ torsowidth torsodepth
torsoheight],torsocenter)
htorso=drawellcylinder(torsowidth/2,torsodepth/
2,torsoheight,torsocenter,1);


% legs
%drawbox([ 2*legwidth legwidth leglength],legcenter)
huleg1=drawcylinderrot(legwidth/2,leglength/
2,ulegcenter,ul1x*pi/180,ul1z*pi/180);
huleg2=drawcylinderrot(legwidth/2,leglength/
2,ulegcenter+[ legwidth 0 0],ul2x*pi/180,ul2z*pi/
180);

hlleg1=drawcylinderrot(legwidth/2,leglength/
2,llegcenter,ll1x*pi/180,ll1z*pi/180);
hlleg2=drawcylinderrot(legwidth/2,leglength/
2,llegcenter+[ legwidth 0 0],ll2x*pi/180,ll2z*pi/
180);

drawsphere(legwidth*.55,ulegcenter+[ 0 leglength/
4*cos(hf*pi/180)+.25*legwidth*sin(hf*pi/180) -
leglength/4*sin(hf*pi/180)] );
drawsphere(legwidth*.55,ulegcenter+[ legwidth
leglength/4*cos(hf*pi/180)+.25*legwidth*sin(hf*pi/
180) -leglength/4*sin(hf*pi/180)] );

% feet
%hfeet=drawbox([ 2*footwidth footlength footwidth],
footcenter);

%hfeet1=drawellcylinderrot(footwidth/2,footwidth/
3,footlength,footcenter-[ legwidth/2 0 0],f1x*pi/
180,f1z*pi/180);
hfeet1=drawcylinderrot(footwidth/
2,footlength,footcenter+[ legwidth/2 0 0],f1x*pi/
180,f1z*pi/180);
hfeet2=drawcylinderrot(footwidth/
2,footlength,footcenter-[ legwidth/2 0 0],f2x*pi/
180,f2z*pi/180);

% arms
%drawbox([ armwidth armwidth lu],uacenter1)
hua1=drawcylinderrot(armwidth/
2,lu,uacenter1,ua1x*pi/180,ua1z*pi/180);
%drawbox([ armwidth ll armwidth],lacenter1)
hla1=drawcylinderrot(armwidth/
2,ll,lacenter1,la1x*pi/180,la1z*pi/180);
drawsphere(armwidth/2,uaend*1.05-[ 0 0 armwidth/2] ');
drawsphere(armwidth/2,1.02*[ -uaend(1)-torsowidth
uaend(2) uaend(3)-armwidth/2] );
drawsphere(armwidth/2,[ 0 0 -armwidth/2] );
drawsphere(armwidth/2,[ -torsowidth 0 -armwidth/2] );

%drawbox([ armwidth armwidth lu],uacenter2)
%drawbox([ armwidth ll armwidth],lacenter2)

hua2=drawcylinderrot(armwidth/
2,lu,uacenter2,ua2x*pi/180,ua2z*pi/180);
hla2=drawcylinderrot(armwidth/
2,ll,lacenter2,la2x*pi/180,la2z*pi/180);

 allh=[ hhead hplss htorso huleg1 huleg2 hlleg1
hlleg2 hfeet1 hfeet2 hua1 hua2 hla1 hla2];
%allh=[ hhead hplss htorso huleg1 huleg2 hlleg1
hlleg2 hfeet1 hfeet2 hua1 hua2];
```

```
l=light('Position',[ 0,2,2] );
set(allh,'facecolor',.75*[ 1 1 1],'edgecolor',.75*[ 1
1
1],'facelighting','gouraud','edgelighting','gouraud'
)
set(allh,'ambientstrength',.3)
material shiny

% reach envelope
%drawcylinder(reradius,redepth,recenter)
%axis([ -150 150 -150 150 -150 150] )
set(gca,'color',[ 1 1 1] )
axis([ -100,100,-100,100,-150,50] )
view(140,20)
%keyboard
```

```
function h=drawsphere(r,c)
% drawsphere(r,c) draws a sphere of radius r at
center c


k=5;
n=2^k-1;
theta=pi*(-n:2:n)/n;
phi=(pi/2)*(-n:2:n)'/n;
X=r*cos(phi)*cos(theta)+c(1);
Y=r*cos(phi)*sin(theta)+c(2);
Z=r*sin(phi)*ones(size(theta))+c(3);
colormap([ 0 0 0;.8 .8 .8] )
h=surf(X,Y,Z,ones(32,32));
shading interp
%surf(X,Y,Z,[ 1 1 1] )
```

```
axis('square')



function hpatch=drawbox(size,center)
% drawbox(size,center) draws a rectangular prism of
size size with center
% at center

x=[ -size(1)/2 size(1)/2] +center(1);
y=[ -size(2)/2 size(2)/2] +center(2);
z=[ -size(3)/2 size(3)/2] +center(3);

vertices=[ x(1) y(1) z(1);x(2) y(1) z(1);x(2) y(2)
z(1);x(1) y(2) z(1);...
x(1) y(1) z(2);x(2) y(1) z(2);x(2) y(2) z(2); x(1)
y(2) z(2)] ;

order=[ 1 2 3 4;2 3 7 6;1 4 8 5;5 6 7 8;1 2 6 5;4 3 7
8] ;
%patch('Vertices',vertices,'Faces',order,'FaceVertex
CData',hsv(6),'FaceColor','flat')

hpatch=patch('Vertices',vertices,'Faces',order,'Face
VertexCData',[ 1 1 1],'FaceColor','flat');
% set(hpatch,'facecolor',.5*[ 1 1 1],'edgecolor',[ .5
.5 .5] )
% l=light('Position',[ 0,2,2] );
%
set(hpatch,'facelighting','gouraud','edgelighting','
gouraud','ambientstrength',.6);

function h=drawellcylinder(r1,r2,t,center,vertical)
% drawcylinder(r,t,center) draws a cylinder of
radius r, thickness t at
```

```
% center coords center

tstep=10;
nstep=360/tstep;
theta=(0:tstep:360-tstep)'*pi/180;
lth=length(theta);
if vertical
            vertices=[ center(1)+r1*sin(theta)
(center(3)-t/2)*ones(size(theta))
center(2)+r2*cos(theta);center(1)+r1*sin(theta)
(center(3)+t/2)*ones(size(theta))
center(2)+r2*cos(theta)];
            % draw ends

hend1=patch(vertices(1:lth,1),vertices(1:lth,3),vert
ices(1:lth,2),'w');

hend2=patch(vertices(lth+1:2*lth,1),vertices(lth+1:2
*lth,3),vertices(lth+1:2*lth,2),'w');
else
            vertices=[ center(1)+r1*sin(theta)
(center(2)-t/2)*ones(size(theta))
center(3)+r2*cos(theta);center(1)+r1*sin(theta)
(center(2)+t/2)*ones(size(theta))
center(3)+r2*cos(theta)];
            % draw ends

hend1=patch(vertices(1:lth,1),vertices(1:lth,2),vert
ices(1:lth,3),'w');

hend2=patch(vertices(lth+1:2*lth,1),vertices(lth+1:2
*lth,2),vertices(lth+1:2*lth,3),'w');
end %if
%order=[ (1:lth);(lth:2*lth)+1;(lth+1:2*lth);(lth+1:2
*lth)-1];
% draw ends
%
patch(vertices(1:lth,1),vertices(1:lth,2),vertices(1
:lth,3),'w')
%
patch(vertices(lth+1:2*lth,1),vertices(lth+1:2*lth,2
),vertices(lth+1:2*lth,3),'w')
% % draw sides
order=[ (1:nstep);(1:nstep)+1;(1:nstep)+nstep+1;(1:ns
tep)+nstep] ;
col1=find(order(1,:)==0);
col2=find(order(2,:)==0);
col3=find(order(3,:)==0);
col4=find(order(4,:)==0);

order(1,col1)=nstep;
order(2,col2)=nstep;
order(3,col3)=2*nstep;
order(4,col4)=2*nstep;

col1=find(order(1,:)>nstep);
col2=find(order(2,:)>nstep);
col3=find(order(3,:)>2*nstep);
col4=find(order(4,:)>2*nstep);

order(1,col1)=order(1,col1)-nstep;
order(2,col2)=order(2,col2)-nstep;
order(3,col3)=order(3,col3)-nstep;
order(4,col4)=order(4,col4)-nstep;

order=order';

if vertical
            hpatch=patch('Vertices',[ vertices(:,1)
vertices(:,3)
vertices(:,2)],'Faces',order,'FaceVertexCData',[ 1 1
```

```
1],'Facecolor','flat');  % using patch gives opaque
cyl
else

hpatch=patch('Vertices',vertices,'Faces',order,'Face
VertexCData',[ 1 1 1],'Facecolor','flat');  % using
patch gives opaque cyl
end %if
% set([ hpatch hend1 hend2],'facecolor',.5*[ 1 1
1],'edgecolor',[ .5 .5 .5] )
% l=light('Position',[ 0,2,2] );
% set([ hpatch hend1
hend2],'facelighting','gouraud','edgelighting','gour
aud','ambientstrength',.6)


% set(hend1,'facecolor',.5*[ 1 1 1],'edgecolor',[ .5
.5 .5] )
%
set(hend1,'facelighting','gouraud','edgelighting','g
ouraud','ambientstrength',.6)
%
% set(hend2,'facecolor',.5*[ 1 1 1],'edgecolor',[ .5
.5 .5] )
%
set(hend2,'facelighting','gouraud','edgelighting','g
ouraud','ambientstrength',.6)

order(:,5)=order(:,1);

hold on
%
% % use plot3 for transparent, patch for opaque
% if vertical
%           plot3(vertices(1:nstep/
2,1),vertices(1:nstep/2,3),vertices(1:nstep/
2,2),'k');
%           plot3(vertices(nstep/
2+1:nstep,1),vertices(nstep/
2+1:nstep,3),vertices(nstep/2+1:nstep,2),'k');
%
%           for i=1:nstep
%
plot3(vertices(order(i,:),1),vertices(order(i,:),3),
vertices(order(i,:),2),'k')
%           end %for
% else
%           plot3(vertices(1:nstep/
2,1),vertices(1:nstep/2,2),vertices(1:nstep/
2,3),'k');
%           plot3(vertices(nstep/
2+1:nstep,1),vertices(nstep/
2+1:nstep,2),vertices(nstep/2+1:nstep,3),'k');
%
%           for i=1:nstep
%
plot3(vertices(order(i,:),1),vertices(order(i,:),2),
vertices(order(i,:),3),'k')
%           end %for
%
% end %if

            %keyboard

%view(140,20)
h=[ hpatch hend1 hend2];
```

```
function h=drawcylinderrot(r,t,center,xangle,zangle)
% drawcylinder(r,t,center) draws a cylinder of
radius r, thickness t at
% center coords center, rotated by xangle about x
axis and zangle about z axis
% x rotation in performed first

tstep=10;
nstep=360/tstep;
theta=(0:tstep:360-tstep)'*pi/180;
lth=length(theta);

% vertices=[ center(1)+r*sin(theta) (center(2)-t/
2)*ones(size(theta))
center(3)+r*cos(theta);center(1)+r*sin(theta)
(center(2)+t/2)*ones(size(theta))
center(3)+r*cos(theta)];

unrotverts=[ r*sin(theta) (-t/2)*ones(size(theta))
r*cos(theta);r*sin(theta) (t/2)*ones(size(theta))
r*cos(theta)];

% rotate cylinder
rotx=rot([ 1;0;0],xangle);
rotz=rot([ 0;0;1],zangle);

rotxz=rotx*rotz;

for index=1:2*lth

vertices(index,:)=(rotxz*unrotverts(index,:)')';
end %for

%  translate to center
transl=[ center(1)*ones(2*lth,1)
center(2)*ones(2*lth,1) center(3)*ones(2*lth,1)];
vertices=vertices+transl;


hend1=patch(vertices(1:lth,1),vertices(1:lth,2),vert
ices(1:lth,3),'w');
hend2=patch(vertices(lth+1:2*lth,1),vertices(lth+1:2
*lth,2),vertices(lth+1:2*lth,3),'w');


%order=[ (1:lth);(lth:2*lth)+1;(lth+1:2*lth);(lth+1:2
*lth)-1];
% draw ends
%patch(vertices(1:lth,1),vertices(1:lth,2),vertices(
1:lth,3),'w')
%patch(vertices(lth+1:2*lth,1),vertices(lth+1:2*lth,
2),vertices(lth+1:2*lth,3),'w')

% draw sides
order=[ (1:nstep);(1:nstep)+1;(1:nstep)+nstep+1;(1:ns
tep)+nstep];
col1=find(order(1,:)==0);
col2=find(order(2,:)==0);
col3=find(order(3,:)==0);
col4=find(order(4,:)==0);

order(1,col1)=nstep;
order(2,col2)=nstep;
order(3,col3)=2*nstep;
order(4,col4)=2*nstep;

col1=find(order(1,:)>nstep);
col2=find(order(2,:)>nstep);
col3=find(order(3,:)>2*nstep);
```

```
col4=find(order(4,:)>2*nstep);

order(1,col1)=order(1,col1)-nstep;
order(2,col2)=order(2,col2)-nstep;
order(3,col3)=order(3,col3)-nstep;
order(4,col4)=order(4,col4)-nstep;

order=order';


hpatch=patch('Vertices',vertices,'Faces',order,'Face
VertexCData',[ 1 1 1],'Facecolor','flat');  % using
patch gives opaque cyl

%set([ hpatch hend1 hend2],'facecolor',.5*[ 1 1
1],'edgecolor',[ .5 .5 .5] )
%l=light('Position',[ 0,2,2] );
%set([ hpatch hend1
hend2],'facelighting','gouraud','edgelighting','gour
aud','ambientstrength',.3)


order(:,5)=order(:,1);

hold on
%
% % use plot3 for transparent, patch for opaque
% if vertical
%           plot3(vertices(1:nstep/
2,1),vertices(1:nstep/2,3),vertices(1:nstep/
2,2),'k');
%           plot3(vertices(nstep/
2+1:nstep,1),vertices(nstep/
2+1:nstep,3),vertices(nstep/2+1:nstep,2),'k');
%
%           for i=1:nstep
%
plot3(vertices(order(i,:),1),vertices(order(i,:),3),
vertices(order(i,:),2),'k')
%           end %for
% else
%           plot3(vertices(1:nstep/
2,1),vertices(1:nstep/2,2),vertices(1:nstep/
2,3),'k');
%           plot3(vertices(nstep/
2+1:nstep,1),vertices(nstep/
2+1:nstep,2),vertices(nstep/2+1:nstep,3),'k');
%
%           for i=1:nstep
%
plot3(vertices(order(i,:),1),vertices(order(i,:),2),
vertices(order(i,:),3),'k')
%           end %for
%
% end %if

           %keyboard

%view(140,20)
h=[ hpatch hend1 hend2];
```


3231-42