

# Influence of Spatial Orientation and Spatial Visualization Abilities on Space Teleoperation Performance

by

María Alejandra Menchaca Brandan

B.S. in Mechanical Engineering

Universidad Nacional Autónoma de México, 2005

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

Master of Science in Aeronautics and Astronautics

at the  
Massachusetts Institute of Technology

June 2007

© Massachusetts Institute of Technology 2007.  
All rights reserved

Signature of Author \_\_\_\_\_

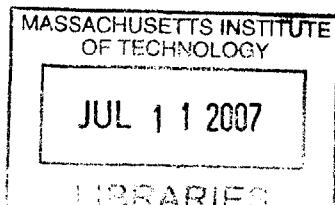
Department of Aeronautics and Astronautics  
May 30<sup>th</sup>, 2007

Certified by \_\_\_\_\_

Charles M. Oman  
Senior Lecturer  
Department of Aeronautics and Astronautics  
Thesis Supervisor

Accepted by \_\_\_\_\_

Jaime Peraire  
Professor of Aeronautics and Astronautics  
Chair, Committee on Graduate Students



AERO

# **Influence of Spatial Orientation and Spatial Visualization on Space Teleoperation Performance**

By

María Alejandra Menchaca Brandan

Submitted to the Department of Aeronautics and Astronautics on May 30<sup>th</sup>, 2007, in partial fulfillment of the requirements for the degree of Master of Science in Aeronautics and Astronautics

## **Abstract**

Astronauts perform space teleoperation tasks with visual feedback from outside cameras. Individuals differ greatly in the ability to integrate camera views, understand the workspace, and ensure clearances between the robot arm and obstacles. We believe that these individual differences correlate with two known subcomponents of spatial intelligence: perspective-taking (PT) and spatial visualization (SV). A preliminary study [1] supports this hypothesis. We believe astronauts use PT (the ability to imagine an object from a different viewpoint) to integrate camera information into an environmentally-referenced frame defined by the arm control axes. In some cases, it may be easier to visualize the manipulation of the payload with respect to the robot arm itself, than to the environment. In that case, SV (i.e., the ability to mentally manipulate an object from an egocentric perspective) may be exploited. We measured the performance of 25 naïve subjects who used hand-controllers to rotate and translate, and 3 environmentally-fixed camera views. These devices controlled a 2-boom, 6 degree-of-freedom virtually-simulated arm to perform pickup and docking subtasks. To challenge the subjects' spatial ability we introduced a wide separation between camera views for some tests, and misalignments between the translation control and the display reference frames. We used the Perspective-Taking Ability test (PTA) and the Purdue Spatial Visualizations Test: Visualization of Views (PSVT:V) to measure PT, and the Cube Comparisons test (CC) to assess SV. We concluded that PTA predicted performance on pickup and docking subtasks, but PSVT:V did not. CC scores correlated with those measures of performance that did not necessarily require PT. High perspective-taking scorers performed the pickup task significantly more efficiently than low, but not faster. In docking, however, they were both significantly faster and more accurate, collided less often, and docked more accurately. In both tasks they moved along only one axis at a time. High CC scorers docked significantly more accurately and rotated about fewer axes at any one time. Whenever we found a significant effect of PSVT:V on a dependent variable, we also found one for PTA; but not the reverse. We had expected higher PT scorers to perform better than others under the challenge of wider camera angles and greater control-display frame misalignments, but we could not demonstrate this. On average females were slower and had lower docking accuracy, an effect related, perhaps, to their lower spatial ability scores. This study of performance during the first two hours of teleoperation training may help define issues for future research.

Thesis Supervisor: Charles M. Oman  
Title: Senior Lecturer

Keywords: teleoperation, remote manipulation, spatial ability, perspective taking, spatial visualization, mental rotation.



# Acknowledgements

Thanks to Dr. Andy Liu, for getting me interested in the project, and developing the virtual interface for the experiment. He dedicated all the time necessary to make sure everything was up and running, from my thinking process to my lousy server connection.

Thanks to Dr. Alan Natapoff, for being my coach, statistics mentor, etymology reference, writing advisor, cultural translator, granola bar dealer and good friend.

Thanks to Dr. Chuck Oman, my thesis supervisor, for always making sure everything was under control, and for his effort to support me through hard times.

Thanks to Dr. Hiro Aoki, for his insight in the experiment's thought process and help running subjects on extra hours.

Thanks to Dr. Dava Newman, for her magical power to fuel-up my desire to do things in a finger snap. I cannot imagine what I would have done without her generous support and personal advice throughout this last year.

Thanks to Barbara Lechner, for being the angel behind me in every step of this Masters. If her words had the power to change the weather, people in Cambridge would not know what clouds look like.

Thanks to Liz Zotos and the students in the Man Vehicle Lab, for making this aerospace stage of my life such an enjoyable one.

Thanks to all of my subjects, for their patience while doing such a long experiment. Believe me, in space the experience is much cooler.

I owe my mental sanity, nutrition, appendices, improved figures, and cross referencing to Cecilia, Liang, Dan J, Mira and Gina. They were there –ready to give up a good night's sleep if necessary – to hold me in one piece during the last stage of this rollercoaster.

Although most of them were not direct contributors to this thesis, some people deserve special credit for making these two years at MIT unforgettable. Thanks to the person at on-campus housing who matched Mira and I; most of the fun, excitement, social consciousness, cooking skills, trips and laughter of my grad life I owe to her. To Ceci and Gina, for all the good vibes, parties and clandestine memories. To Samar and Rossella, for sharing those insignificant daily highs, as well as those terrible low moments of grad school. To Taki, for taking me to the coolest talks and most unforgettable walks, for making me laugh constantly and reminding me that nothing is impossible. To the Maleks, for adopting me that



week, knowing exactly what I needed to regain my energy and confidence. To Claire and Liang, for all that chocolate and good times at the lab. To Danial, for sharing with me his pure and sincere approach to life. To my roommates, Caro and Federico, for listening so patiently to my daily adventures. To Appu et al., for reminding that fun at MIT *does* exist.

At last but certainly not least, thanks to my family. Specially, to my parents, for being my life advisors; for teaching me the tricks of survival and standing next to me all these years. To my sister Nuria, for always providing our home with calm and a touch of normality. To my (hi-tech) grandma Yayi, for cheering me online ever since I got to the US. To grandpa Pepé for his letters full of love, wisdom (and soccer updates), religiously attached to my travel magazines.

This research was supported by the Science and Technology Mexican Council (CONACyT) Fellowship and from NASA Cooperative Agreement NCC9-58 with the National Space Biomedical Research Institute.

# Table of contents

Acknowledgements .....	3
Table of contents.....	5
Table of figures.....	6
Chapter 1. Introduction .....	7
Chapter 2. Background .....	9
2.1. RMS operations, training and performance assessment .....	9
2.2. Human spatial abilities .....	11
2.3. Previous research on teleoperation performance and spatial ability.....	15
2.4. Preliminary study and rationale for experiment design.....	16
Chapter 3. Objective and hypotheses .....	21
3.1. Objective.....	21
3.2. Hypotheses .....	21
Chapter 4. Methods.....	23
4.1. Camera configurations .....	23
4.2. MIT Remote Manipulation System Simulator (MIT RMSS) .....	25
4.3. Environment configuration.....	26
4.4. Task .....	28
4.5. Task performance metrics .....	31
4.6. Spatial ability metrics .....	34
4.7. Subjects .....	34
4.8. Procedure .....	35
Chapter 5. Results.....	37
5.1. Overview of spatial ability test scores.....	39
5.2. Effect of perspective-taking ability .....	41
5.3. Effect of spatial visualizations.....	49
5.4. Effect of gender differences .....	52
5.5. Other effects .....	54
Chapter 6. Discussion .....	56
6.1. Strategies of motion.....	56
6.2. Accuracy .....	57
6.3. Use of multiple displays .....	57
Chapter 7. Conclusions .....	59
Chapter 8. Suggestions for future work.....	61
References .....	63
Appendices.....	66

# Table of figures

Figure 1. The robot arm controlled by astronauts .....	8
Figure 2. Robotic Workstation (RWS) [4] .....	10
Figure 3. Sample problems from the Cube Comparisons test .....	13
Figure 4. Screenshot of the Perspective-Taking Ability test [5] .....	14
Figure 5. Purdue Spatial Visualizations: Visualizations of Views test (PSVT:V) .....	14
Figure 6. Camera configurations $\alpha$ , $\beta$ and $\gamma$ .....	18
Figure 7. Average task performance in preliminary study per configuration ( $\alpha$ , $\beta$ , $\gamma$ ) .....	19
Figure 8. Camera configurations A, B and C .....	23
Figure 9. Location of the three camera views relative to the horizontal axis x .....	24
Figure 10. MIT Remote Manipulator System Simulator .....	25
Figure 11. Translational Hand Controller (left) and Rotational Hand Controller (right) .....	26
Figure 12. Robot arm mounted on two large fixed trusses .....	28
Figure 13. Pickup of the payload dish and docking of the cargo module onto the ISS node .....	29
Figure 14. Trial paths defining the starting and ending point of the arm's end effector .....	30
Figure 15. Two-dimensional path with semi-transparent wall .....	30
Figure 16. Docking offsets .....	34
Figure 17. Correlation between the spatial ability tests .....	40
Figure 18. Effect of PTA and PSVT:V test scores on pickup performance metrics .....	41
Figure 19. Effect of PTA and PSVT:V test scores on pickup performance metrics .....	42
Figure 20. Effect of PTA and PSVT:V test scores on docking performance metrics .....	44
Figure 21. Effect of PTA and PSVT:V test scores on docking performance metrics .....	45
Figure 22. Effect of CC test score on pickup performance metrics .....	49
Figure 23. Effect of CC test score on docking performance metrics .....	50
Figure 24. Significant gender differences on pickup performance metrics .....	52
Figure 25. Gender differences on docking performance metrics .....	53
Figure 26. Number of cameras used simultaneously for each configuration .....	58

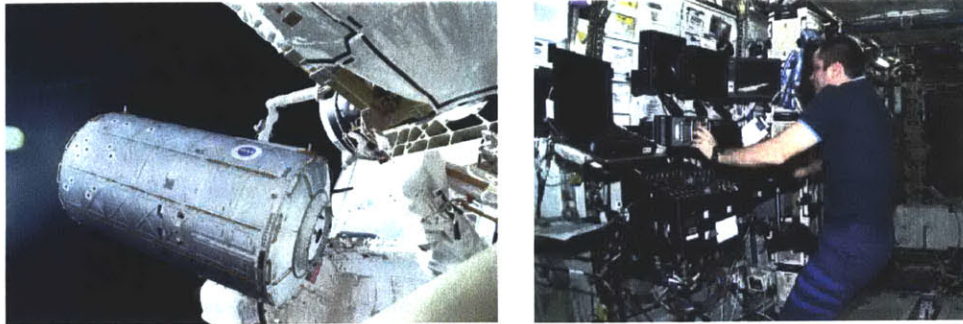
# Chapter 1. Introduction

Think of the robot arm that is mounted on the Space Shuttle or the Space Station. Now think of the astronauts, inside the Shuttle or the Station, trying to manipulate this robot using three visual displays and two hand controllers. The displays provide camera views of the arm and workspace from three different environmentally fixed perspectives, and the controllers govern the translational and rotational motion of the end effector. Manipulation is relatively simple if a camera view is aligned with the arm's motion (i.e., when the astronaut moves the arm to the right, the arm on the screen moves to the right). The task is harder, however, if the camera view and the arm's motion are not aligned (for example, when moving the arm to the right, the arm on the screen moves forward). When all the camera views are misaligned with the control axes, the task becomes even more complicated. NASA astronaut teleoperation trainers say that the ability to mentally integrate these camera views varies from one astronaut to another, and this thesis is an attempt to understand why.

Teleoperation is widely used in areas such as medicine, underwater exploration and space activities to perform tasks that require the intelligence and awareness of a human but the physical capabilities of a robot to perform tasks in environments that are dangerous or inaccessible to people. In the case of space exploration diverse tasks such as deployment of satellites, maintenance of payload, inspection and repair of the Space Shuttle or the construction of the International Space Station (ISS) have been performed through teleoperation, using either the Space Shuttle or the ISS Remote Manipulator Systems (RMS) (Figure 1). Maintaining awareness of the spatial location and the relative motion between all the elements in the workspace, such as payloads, structures, the arm itself and the different cameras is not an easy task, and operating the remote manipulators requires intensive training. Consequently, current space remote manipulation procedures require having two operators to carefully monitor each movement of the arm, select and orient each camera view, and each other's actions, maintaining spatial situation awareness and alertness for potential collisions or spatial misjudgments. These procedures increase the duration of RMS operations, and require additional valuable crew time on each mission.

The robot arms are frequently operated with respect to a control reference frame that is allocentric: The arm consistently moves in the same direction –with respect to the external environment– for the same input of the hand controller, independently of where the camera view points are. This is referred to as “external” control mode. To manipulate the arm in this mode, the operator must observe the robot from multiple camera perspectives which are often not aligned with the control reference frame. As a result, the motions commanded on the hand controllers will not necessarily correspond to a similar motion on the camera display (e.g., a leftward motion on the hand controller may result in an arm movement in depth for Display 1, and motion to the right in Display 2!). Safe operation requires having the ability to integrate the different camera views by perceiving and mentally processing the

misalignment between each image and the control reference frame of the arm. The failure to perform this cognitive task has led to several operational difficulties in space operations; in most cases, astronauts have trouble finding the optimal camera views to define the adequate clearances from the surrounding structure [2].



**Figure 1. The robot arm (left) is controlled by astronauts inside the International Space Station or Space Shuttle (right)**

During the RMS training, individual differences in the astronaut trainees' abilities to select the correct camera view, understand the images provided, maneuver the robot arm and maintain adequate clearances between objects have come to light,<sup>1</sup> suggesting that a specific set of factors within the operators' general intelligence could be influencing their teleoperation performance. We believe that spatial ability is one of those factors, and that it should be possible to predict the strengths and weaknesses of each astronaut in specific aspects of teleoperation, before they are assigned to a mission, or even before they are trained as operators. This would allow each astronaut to follow RMS training tailored to his/her specific needs, and would allow matching RMS operators with complementary skills. Moreover, in the case of an onboard emergency requiring an unrehearsed teleoperation task, it may be possible to predict who would be the best candidate to perform the task. This could lead to shorter, safer and more efficient space operations.

This thesis presents a first attempt to define the aspects of space teleoperation performance that are influenced by the operator's spatial abilities and suggests a research strategy for future experiments. Chapter 2 reviews a) RMS operations, training and performance assessment, b) previous research on human spatial intelligence and ability testing, and c) the results of successive preliminary pilot experiment that lead to, in Chapter 3, the objectives and hypotheses for the experiment described in this thesis. Chapter 4 presents the details of our experimental method. Chapters 5 and 6 present the results and a discussion of their relevance for teleoperation performance.. Conclusions are presented in Chapter 7 and suggestions for further research in Chapter 8.

---

<sup>1</sup> J. Young, NASA Johnson Space Center, personal communication.

# Chapter 2. Background

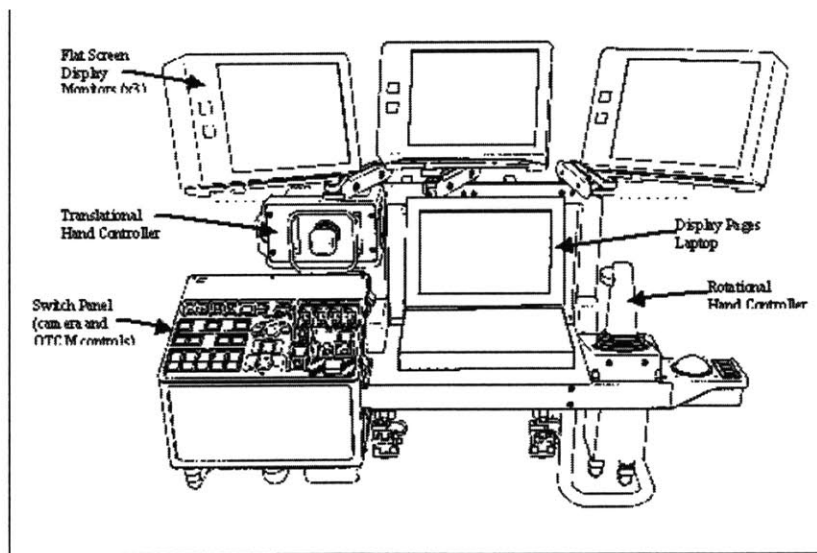
## 2.1. RMS operations, training and performance assessment<sup>2</sup>

The Remote Manipulator Systems (RMS) on the Space Shuttle and the International Space Station (ISS) are used by astronauts to perform orbital deployment, maintenance and repair of satellites (such as the Hubble Space Telescope), build large structures (such as the ISS itself), and more recently, to monitor the state of the Shuttle's thermal shield before reentry. They consist of a robot arm and a Robotic Workstation (RWS). The Shuttle and RMS robot arms are very similar, with the main difference that the Shuttle arm has 6 degrees-of-freedom (DOF), while that in the ISS has 7 DOF. They are manipulated from the RWS, which is composed of three video displays, numeric displays of arm state including position and orientation of the end effector, and two hand controllers, for translation and rotation respectively (Figure 2) [3]. The Space Shuttle has rear cockpit windows looking into the payload bay that permit direct visual monitoring of the operations. Each of the video monitors can show an image from a camera located on the surrounding structure of the Shuttle or Station, and on the end effector itself. Astronauts select an appropriate camera view and can pan, tilt and zoom each camera to their preference, in order to obtain the best possible view. Visual feedback is usually the only dependable information<sup>3</sup> source for arm motion and clearance, while additional information on the position and orientation of the end effector is provided on a separate numeric display. The translation and rotation of the robot arm can be aligned with a control frame fixed to the spacecraft ("external control mode") or fixed to the moving end effector ("internal mode"), as specified by the operator.

---

<sup>2</sup> This section was adapted from a 2007 research proposal "Advanced Displays for Efficient Training and Operation of Robotic Systems", NSBRI RFA-07001-S2, CM Oman, Principal Investigator, with the permission of the authors.

<sup>3</sup> DOUG (Dynamic Onboard Ubiquitous Graphics, see last paragraph of this section) is used as supplementary spatial awareness system, though the information is not deemed accurate enough to depend on for clearance assessment.



**Figure 2. Robotic Workstation (RWS) [4]**

Beginning in 2003, all NASA astronaut candidates must complete a one hour-long “Robotics Aptitude Assessment” (RAA), a computerized test battery. The RAA is designed to test the candidates’ ability to judge clearances, to perform a docking task with two hand controllers, to use both internal and external command modes to reorient objects, to translate the end effector through specific paths and rotate it to different positions, and to use both hand controllers simultaneously and multi-axially. The RAA has not been validated as a selection tool, thus the data from the assessment are not used as a selection criterion. Once selected, the astronauts begin their teleoperation training with Generic Robotics Training (GRT), where they are taught the basic robot arm manipulation tasks (e.g., flying the arm, grappling objects, or choosing the appropriate camera views) in a sequence of 15 consecutive lesson modules. Two training systems are used throughout the GRT sessions: BORIS (Basic Operational Robotics Instructional System) and the Multi-Use Remote Manipulator Development Facility (MRMDF). Both systems closely simulate the real RMSs, virtually (BORIS, 6DOF) or physically (MRMDF, 7DOF), and provide multiple camera views and two hand controllers.

Two astronauts are required to operate the real robot arms. In a space teleoperation mission, the R1 operator is the one in charge of manipulating the arm hand controllers, while R2 is in charge of tracking the moving object with the camera, switching the camera views as instructed by R1, and monitoring obstacle clearance and situational awareness. Those trainees who show the best training performance are classified as primary operators (R1), while those who present lower but acceptable skills are classified as secondary operators (R2). Usually, operators are assigned to a mission as R1 or R2, according to their original post-training classification; however, there are times when the mission R2 is someone who had a R1 evaluation designation and the mission task R1 is someone who had only attained R2 scores. This latter person serves as the R1 for scripted tasks, but should a problem arise,

then the two operators would switch roles with the “better” operator assuming R1 duties. In all nominal cases, the R2 has been fully trained so they are capable of performing the nominal task, whether they received R1 or R2 scores in training<sup>4</sup>.

During under robotics training, astronauts are evaluated after specific lessons by a group of Robotics Instructors and Instructor Astronauts. Performance scores are given based on a weighed sum of nine standardized criteria.<sup>5</sup> The weighting of the scores is based on estimated impact on mission success, therefore higher weighting is given to criteria relating to spatial/visual perception, situational awareness, clearance monitoring and maneuvering, which cover collision and singularity avoidance, correct visualization of end position and attitude, adequate camera selection and real time tracking, motion smoothness and the ability to maneuver in more than an axis at a time. Interestingly, this evaluation method assigns those astronauts with better performance to R1 positions, when in reality it is the R2 operator’s main task to provide clearance and situational awareness support.

The astronauts who do not achieve the minimum requirements (acceptable/strong grades) must undergo remedial training, including methods to help trainees visualize the orientation of the control reference frame. These visualization methods include the use of small scale physical models of the arm and the reference axes, and also DOUG (Dynamic Onboard Ubiquitous Graphics), a laptop computer virtual recreation of the International Space Station that allows the user to navigate freely around the Station and obtain a better understanding of the location of the different structures. DOUG is also used for supplemental awareness during robotic operations, although it cannot be used as a dependable source of clearance information, due to its accuracy limitations. These visualization methods strongly suggest that spatial ability plays an important role in space teleoperation performance, and being able to predict beforehand the spatial weaknesses and strengths might simplify the RMS training process and make it more efficient.

## 2.2. Human spatial abilities

General intelligence is the manifestation of the cognitive capacities of an individual. Those capacities (such as reading, logical thinking, memorizing, multiplying, etc.) can be measured by psychometric tests, which allow the differentiation of cognitive abilities between individuals or groups. One of the components of an individual’s general intelligence is the spatial ability.

Our ability to imagine, transform and remember any kind of information acquired by visual experience such as maps, pictures, or 3D drawings, can be described by our spatial ability, and it is believed that this

---

<sup>4</sup> J. Tinch, Chief Engineer of the Astronaut Office Robotics Branch, NASA Johnson Space Center, personal communication.

<sup>5</sup> The Instructor Astronauts are not involved in all the evaluations, but mostly on the final stage.



ability depends on factors such as gender, age, and personal experience. Different subcomponents of spatial ability have been identified over the past 40 years, each of them defining a specific mental function required for image processing. . The typical approach has used principal component analysis to decompose performance into 2-7 underlying spatial ability subcomponents.

Although the classification of such subcomponents slightly differs among authors, two main classes can be defined: *spatial orientation (SO)* and *spatial visualization (SV)*. *Spatial orientation* refers to the ability to imagine different viewpoints of an object, and has recently been subdivided into *perspective-taking (PT)* and *mental rotation (MR)*. The main difference between PT and MR is the frame of reference which is manipulated to achieve this viewpoint. In PT the observer's frame of reference moves (i.e., the object stays still and the observer moves around it), whereas in MR the display reference frame is fixed (i.e., the observer stays still) and the object is imagined as rotating. For instance, to describe how the school library looks from the tall building in front of it, we would tend to use PT and imagine our own translation while climbing the tall building and peeking out the window. However, to describe how the back side of a quarter appears, most people would use MR and visualize themselves flipping a coin, instead of moving their own viewpoint. Recently, Kozhevnikov and Hegarty [5] and Hegarty and Waller [6] have determined that PT and MT are distinguishable abilities, although highly correlated. *Spatial visualization (SV)* can be described as the ability to visualize the transformation of objects or surfaces in an image into other configurations (unfolding a paper sheet, for instance).

In addition to spatial orientation and visualization, other subcomponents of spatial ability have been proposed. For example, Carroll [7] includes *closure speed* (ability to rapidly access representations from long-term memory), *flexibility of closure* (ability to maintain the representation of an object in working memory while trying to distinguish it in a complex pattern), *perceptual speed* (ability to rapidly compare or find symbols or figures), and *visual memory* (ability to remember the spatial distribution of objects or figures). Pellegrino et al. [8], and Contreras et al. [9] have suggested the existence of *dynamic spatial performance*, as a supplemental set of factors, including the abilities to perceive and extrapolate real motion, predict trajectories and estimate the arrival time of a moving object.

### **2.2.1 Measures of Spatial Orientation and Spatial Visualization**

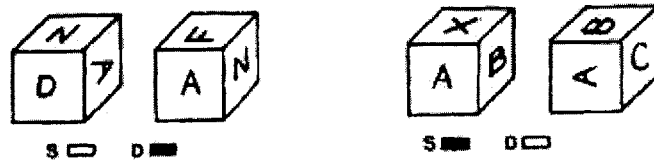
Many different cognitive tests have been developed to measure each subcomponent of spatial ability. The majority of these tests are still paper-and-pencil based, but the trend is to move towards computer-based tests, which enable the measurement of a greater number of variables, including the temporal aspects of performance. The Kit of Factor-Referenced Cognitive Tests [10] contains several paper-and-pencil tests that have been used for several years in the field of spatial cognition research, such as the Cube Comparisons test (SV) and the Paper Folding test (SV). Other widely used paper-and-pencil tests are the Vandenberg and Kuse Mental Rotations Test, developed by Vandenberg and Kuse (MR) [11], and the Purdue Spatial Visualization Test by Guay [12], which has three sections: Rotations (MR), Views (PT)

and Development (SV). Among the computerized tests, there are the Perspective-Taking Ability test or PTA [13], and the Pictures Test [6] which test two- and three-dimensional *PT*, respectively.

The Cube Comparisons test has generally been classified as a means of measuring MR ([14] [15], [16]), however, recent reviews of the test now call for its classification as a measure of Spatial Visualization ([17], [18], [7]).

The Cube Comparisons (CC) test, the Perspective-Taking Ability test (PTA), and the Purdue Spatial Visualization – Visualizations of Views (PSVT:V) test are of particular interest for this thesis, and are described in more detail below.

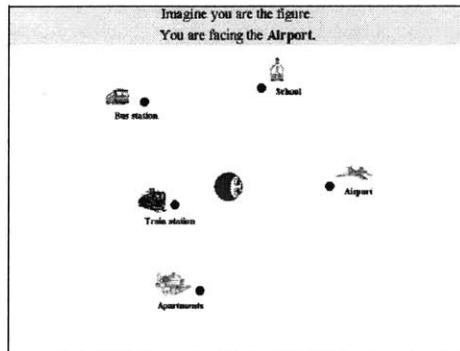
In the CC test, subjects are shown pairs of cubes with a letter or a geometric figure on each face, and are asked to determine whether the cubes are different or could be the same, but rotated. Twenty one pairs of cubes are given (Figure 3) on each of the two 3-minute sessions. This test measures a person's ability to rotate an object –an imaginary wooden cube– in their mind. Just and Carpenter [16] explain that there are four common strategies to solve this test: mental rotation around standard (intrinsic) axes, mental rotation around task-defined axes, use of orientation-free rules, and perspective-taking. Their results suggest that low scorers use mostly mental rotations about standard axes (defined by the faces of the cubes), while high CC scorers use mental rotations and comparisons of object features utilizing task-defined (non-standard) axes. Hence, many reviewers believe the CC test draws on spatial visualization skills in addition to simple mental rotation abilities.



**Figure 3. Sample problems from the Cube Comparisons test showing different cubes (left) and same but rotated cubes (right)**

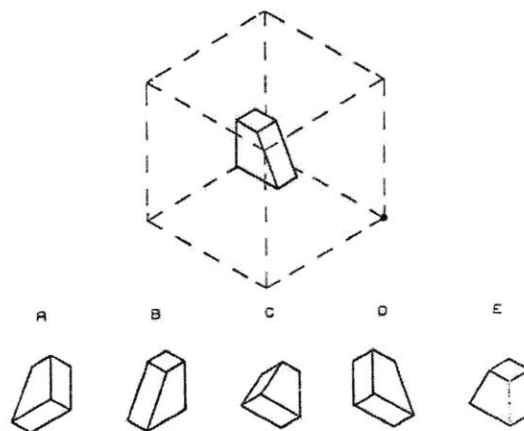
In the computerized version of the Kozhevnikov Perspective-Taking Ability test (PTA) the subject is shown a top-down plan view of a person surrounded by seven items (airport, hotel, school, etc.). He is instructed to imagine himself facing one of the items and is given five seconds to study the relative locations of the other elements. Then, one of the other items flashes and he is instructed to indicate the direction to the indicated target, with respect to his egocentric reference frame. There are 58 trials in the test, and the score is based on angular error as well as response time (Figure 4). Kozhevnikov and Hegarty [5] found that two main strategies can be used to solve this test: perspective-taking and mental rotations. The choice of strategy depends on the angle between the egocentric reference frame and the imagined heading: mental rotation is commonly used for smaller angles (< 100°), and perspective-taking is used for large angles. The PTA test measures perspective-taking ability in a two dimensional azimuthal plane, and presents targets using a map. Kozhevnikov and Hegarty mention that a three-dimensional

perspective view adaptation of PTA might encourage the use of PT strategies to a greater degree, “suppressing” the use of mental rotations.



**Figure 4. Screenshot of the Perspective-Taking Ability test [5]**

In the Purdue Spatial Visualization Test: Visualization of Views (PSVT:V) (Figure 5), a subject is presented with pictures of various solid objects located in the center of a “glass” cubic box, shown in an isometric view. On each trial, a black dot is located on one of the vertices of the box, and the subject must imagine how the object looks from that perspective, and choose one of 5 alternative two-dimensional projections. Although PSVT:V has not been widely validated, the majority of the subjects that have taken it informally as part of a Man Vehicle Laboratory experiment have reported the use of PT over any other strategy to solve the test. This information suggests that the PSVT:V test could be a useful measure of three-dimensional PT, and may be differently correlated to the performance of three-dimensional PT tasks –such as space teleoperation– than a two-dimensional PT test like PTA. More specifically, this test may provide richer information on the ability to take a perspective at wider three-dimensional angles (between the observer’s egocentric reference frame and the imagined heading).



**Figure 5. Purdue Spatial Visualizations: Visualizations of Views test (PSVT:V)**

We hypothesized that the most important spatial ability subcomponents when performing teleoperation tasks are the PT component of *spatial orientation*, as measured by the PTA and PSVT:V tests, and *spatial visualizations*, as measured by the CC test.

### **2.2.2 Gender differences in spatial ability**

Several studies have found an effect of gender on spatial ability tests, but the results do not consistently define a clear trend. Findings suggest that male subjects tend to perform better in such tests [19] – women are only favored only when visuospatial memory is required [20]. Although consistent large gender differences favoring males are only found in mental rotation tasks [21], [22], males also generally outperform females in way-finding tasks [23]. Ecuyer-Dab and Robert [24] speculate these differences originate from natural selection. They argue that the male species requires stronger navigational skills and understanding the physics of moving objects –such as projectiles–, while the females species is entitled to a low-risk and conservative behavior, focusing on nearby spatial cues over large-space navigation.

## **2.3. Previous research on teleoperation performance and spatial ability**

Different aspects of teleoperation that have been studied, such as the adaptation to changes in reference frames or the use of displays, provide indirect evidence that spatial abilities may be correlated with performance. Lamb and Owen [2] found that using an egocentric reference frame for space teleoperation tasks resulted on higher performance than when using an exocentric (world) reference frame. In this experiment, subjects used a head-mounted display and two controllers to fly a robot arm towards a payload target, grasp it, and locate it into the cargo bay of the Space Shuttle, in a virtually-recreated environment. DeJong, Colgate and Peshkin [25] investigated the influence of rotations between the different reference frames (e.g., hand controllers, camera and display, end effector) and concluded that performance improved as the number of rotations between the reference frames decreased. They recommend matching the controllers and displays with their corresponding elements in the remote site, in terms of reference frame as well as physical location, to allow the operator to have a better understanding of the work area. Spain and Holzhausen [26] found that performance is not necessarily improved if the number of camera viewpoints available to perform teleoperation tasks is increased. Although the alternate orthogonal views provide useful depth information which could have contributed to improve the task performance, subjects often did not use the additional camera viewpoints. According to the authors, these additional views increase the subject's mental workload, which may reflect the effort to spatially integrate the view.

A few previous studies have found direct correlations between teleoperation performance and spatial abilities. However none of them included the use of multiple displays to perform the tasks. Laparoscopic

surgery (also called minimally invasive surgery) is an important application of teleoperation in the medical field. Eyal and Tendick [28] found a significant correlation between the ability of novice subjects to learn the proper position of an angled laparoscope and their scores on Card Rotations, Paper Folding and Perspective-taking tests, which measure SV (paper folding), MR (Card Rotations) and PT (paper PTA test) aspects of spatial abilities. Tacey and Lathan [29] measured lower completion times in a single-display teleoperator pick-and-place task for subjects with higher spatial ability, as measured by a composite test of the Paper Folding test and Stumpf's Cube Perspective test. Lathan and Tracey [30] studied 2D navigation performance using a mobile telerobot, with a single camera for visual feedback. They measured the operators' spatial recognition ability with the Complex Figures test and Stumpf's Spatial Memory test, and spatial manipulation ability with the Block Rotation and Stumpf's Cube Perspective test, and found a correlation between high performance and high test scores.

## 2.4. Preliminary study<sup>6</sup> and rationale for experiment design

We performed a preliminary study [1] correlating *spatial orientation* ability and teleoperation performance in a simulated teleoperation task where the operator used three stationary cameras and external (spacecraft, not end-effector) referenced control. Spatial ability was tested with the Perspective-Taking Ability test (PTA), Cube Comparisons (CC) test, and Purdue Spatial Visualization – Visualization of Views test (PSVT:V). The relationship between the camera views and the orientation of the control reference frame was also systematically manipulated. Despite the small number of subjects used in this preliminary experiment, interesting trends were found, indicating that people with better spatial abilities as measured by PTA and PSVT:V performed better. This section describes the methods and results of the study, followed by a discussion of the limitations/problems which led to the development of the thesis experiments.

### 2.4.1 Reference frames

We defined two reference frames related to the teleoperation task: *control reference frame* and *display reference frame*. The *control reference frame* was a world-fixed frame (i.e., independent to the perspective from which the robot is observed) whose axes were defined by the actions of the translational and rotational hand controllers. In our experiments, the *control reference frame* was always aligned with the world reference frame, i.e. it corresponded to the “external” control mode of the Space Shuttle or ISS RMS.

---

<sup>6</sup> A more complete description of this study, including methods and analysis, can be found in the conference paper in APPENDIX K.

The *display reference frame*, on the other hand, corresponded to what would be the observer's egocentric reference frame if he were located at a camera viewpoint. For purposes of this experiment, we assumed all three cameras were in a plane containing the Z (depth) axis, and pointed toward the center of the work area. Although the observer could look at any camera, we arbitrarily defined the primary axis of the *display reference frame* as aligned with the view axis of the central camera, corresponding in some sense to the observer's "average" viewpoint.

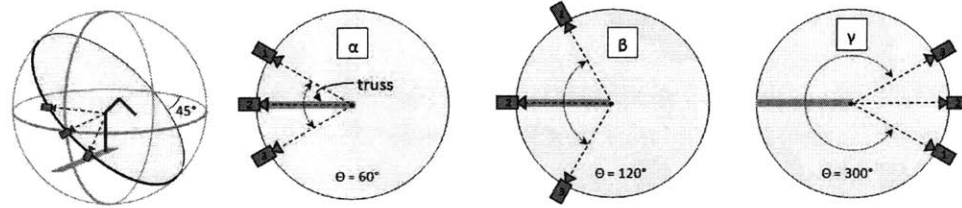
Axes on the *control reference frame* were designated in capital letters ( $\pm X, \pm Y, \pm Z$ ), whereas those in the *display reference frame* were designated in lower case letters ( $\pm x, \pm y, \pm z$ ).

## 2.4.2 Overall design

Given the absence of studies relating spatial abilities and multiple-screen teleoperation, we decided to run an experiment that would correlate the *perspective-taking* (PT) and *spatial visualization* (SV) ability of individuals with their performance on a RWS-simulated environment. We wanted to define an experiment design that would challenge the PT ability of subjects, and provide information on the relative contribution of each spatial ability to the performance of a series of pickup-and-dock tasks. Considering that the ability to imagine how a scene looks from a camera view is strongly related to the angular separation between the observer's egocentric reference frame and the imagined viewing location [5], we designed a within-subjects experiment using three camera configurations, each with a different angular separation between the cameras' lines of view. The hypothesis was that with greater angular separation, poor perspective-takers would present lower performance. Spatial visualization (SV) ability was also measured, expecting that good high scorers would rotate the end effector around multiple standard axes at a time [16], independent of the changes in camera angular separation.

We created a space teleoperation environment, including a virtual 6 degrees-of-freedom (DOF) Unimation PUMA-like robotic arm mounted on a fixed truss, three fixed virtual camera views projected in three computer screens and two hand controllers. Subjects were instructed to use the virtual arm to grapple a simulated cargo module then dock it onto an ISS node as fast and accurately as possible. Three camera configurations ( $\alpha, \beta, \gamma$ ) were tested. The configurations consisted of three cameras equidistant to the base of the arm, and located on a plane tilted  $45^\circ$  from the horizontal axis (Figure 6). In configuration  $\alpha$  and  $\beta$ , the control and the display reference frames were aligned, whereas in configuration  $\gamma$  the reference frames were horizontally misaligned by  $180^\circ$  about the vertical axis. The angular separation between the two external cameras in configurations  $\alpha$  and  $\gamma$  was  $60^\circ$ , and in  $\beta$  it was  $120^\circ$ . We hypothesized that subjects with higher perspective-taking ability would perform better in the configurations with high angular separation, as opposed to those with low ability. We expected CC score to have an effect on docking accuracy (specifically on the angular offset), but not on the other measures of performance, since the task was focused on challenging the subject's perspective-taking ability. Four different trials were presented to the subjects, characterized by different initial positions and orientations of the cargo module. The arm initial position and the ISS node location and orientation

were the same for every trial. Each trial was repeated four times during each camera configuration, reaching a total of 16 trials per configuration. To account for learning effects, subjects were divided into two groups, each with a different order of camera configurations ( $\alpha$ - $\beta$ - $\gamma$ , and  $\alpha$ - $\gamma$ - $\beta$ ). The experiment included a total of 56 trials, including 8 training trials before configuration  $\alpha$ .



**Figure 6. The three cameras are positioned along a circumference (black contour) tilted  $45^\circ$  from the horizontal plane. Camera configurations  $\alpha$ ,  $\beta$  and  $\gamma$  are defined by the angular distance between cameras 1 and 3.**

Three tests were administered to assess spatial ability: Cube Comparisons test (CC), Perspective-Taking Ability test (PTA) and Purdue Spatial Visualization: Visualization of Views test (PSVT:V). We used two different perspective-taking tests because – as discussed earlier - we considered them to be complementary: PTA has been validated with a large number of subjects but involves only two-dimensional perspective-taking, whereas PSVT:V involves three-dimensional perspective-taking, although no major validation has been performed.

The performance metrics measured were: total time to perform the task, observation time before first controller input, percentage of the task time during which there was a controller input (%motion), the input of axial and angular degrees of freedom, and the position and orientation docking offsets of the module with respect to the node. Seven subjects finished the experiment (3 female, 4 male, all Aerospace Engineering students).

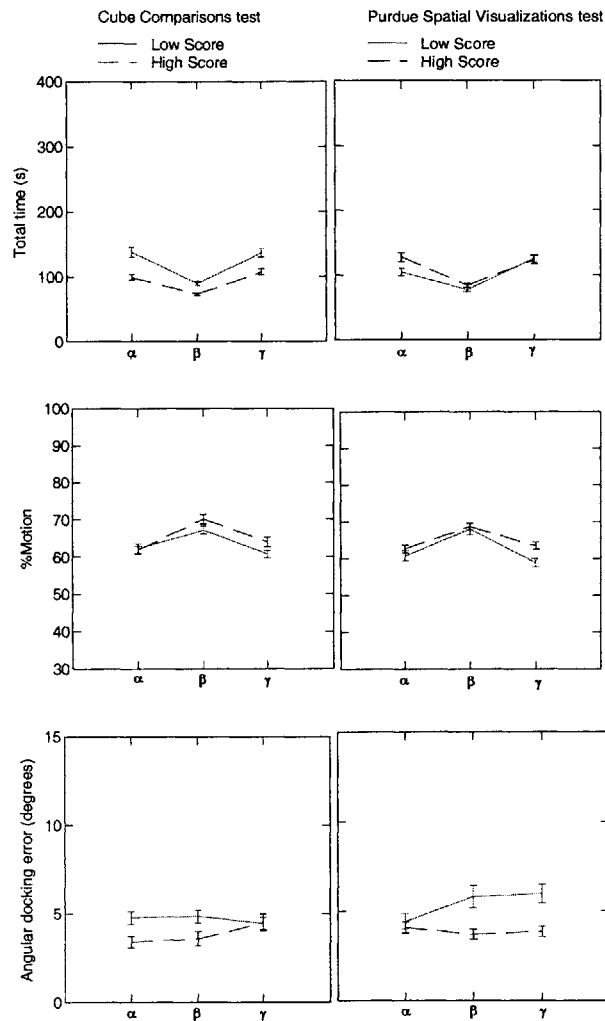
After finishing the experiment, subjects completed a questionnaire in which they described the different level of discomfort they encountered throughout the session, as well as the strategies used to perform the trials.

### 2.4.3 Results

Subjects with better spatial visualization ability performed the task faster, and required less observation time, whereas those with high perspective-taking ability performed the task in longer times, required longer observation times and rotated in multiple axes at a time.

The highest performance by subjects occurred in configuration  $\beta$ , with shorter times, greater %motion, and increased use of multi-axial translation. Configuration  $\alpha$  and  $\gamma$  were characterized by longer total times and lower %motion (Figure 7). The presence of left-right reversal in  $\gamma$  was reflected by longer

observation times, as well as the tendency to translate in one orthogonal axis at a time. No significant correlations were found between docking accuracy and test scores or configuration.



**Figure 7. Average task performance in preliminary study per configuration ( $\alpha$ ,  $\beta$ ,  $\gamma$ ) grouped by subjects scoring low (solid lines) or high (dashed lines) on the Cube Comparison(left) and Purdue Spatial Visualization (right) tests. Total time (top); Center: %motion (center); Angular docking error (bottom).**

Learning trends were characterized by a reduction in total and observation times, and increase in %motion and in diagonal translation. A decrease in docking accuracy with each repetition was also detected, although it may have been caused by fatigue. No change was observed on the number of standard axes around which the end effector was simultaneously rotated, suggesting that if subjects do eventually learn to use multi-axis simultaneous control, the time required exceeds the length of our experiment.



#### **2.4.4 Potential experimental design improvements**

During our analysis of the questionnaire and data, a number of confounding factor resulting from our simulation and experiment became apparent.

Subjects reported using the central camera much more than the other two cameras, since it provided most of the visual information they needed, and was exactly aligned (or misaligned  $180^\circ$  in  $\gamma$ ) with the control reference frame. Additionally, due to the fact that the angular separation between the cameras was not the same, subjects were more likely to integrate a pair of views that included the central camera than the one that did not. The primary use of the central camera may have reduced the correlation of spatial ability (PT) on performance, since this ability was not utilized by their viewing strategy. This strategy choice may also have been a result of the lack of sufficient visual cues to establish an allocentric frame within which they could understand the relationships between the camera views. Thus, they might have relied on mental rotation-based strategies to determine how to move the arm. This strategy would also only require one view most of the time.

We were not able to define a metric to assess the distance traveled in the direction opposite to the final goal (inverse motion) because subjects had the freedom to choose any path to perform the task. For example, some of the subjects reported that they found it easier to move the module, in the sense of avoiding a collision between the robot arm, module, or truss, in front of the arm than behind, even if the distance traveled was larger. The choice of path may have been driven by the desire to avoid collisions. Consequently, paths lengths differed significantly, affecting the total time of the task, as well as complicating the use of a metric of motion in the wrong direction (called inverse motion by Akagi [27]). Including this metric could provide useful information regarding the understanding of the misalignment between the control reference frame and the display reference frame.

Although subjects were instructed to avoid collisions, no feedback was given to them, and no quantitative information was available to assess whether subject had followed the rules, other than the experimenter's observations. If subjects had commanded arm motion regardless of whether a collision would occur, their task completion time would be artificially shorter than if they had to correct the condition before proceeding with their task.

Since the ISS node remained in the same position and orientation for every trial, subjects may have mastered the docking technique without further need the information from the camera views, and reducing the need to use spatial abilities to perform the task. This monotony may even have led to boredom within subjects, and a decrease in the docking accuracy with repetitions. In general, boredom from the length of the experiment (2.5 to 4 hours, including breaks) could have changed the operators' speed-accuracy trade-off and confounded the results. Subjects consistently reported feeling mentally overloaded or bored by the last configuration.

# Chapter 3. Objective and hypotheses

## 3.1. Objective

The objective of this experiment was to answer the following experimental questions:

- i. Is perspective-taking ability (measured by the PSVT:V and PTA tests) correlated to performance on teleoperation subtasks such as pickup and docking?
  - a. How is this correlation affected by wider camera separations?
  - b. How is this correlation affected by misalignments between the control and display reference frames?
- ii. Is spatial visualization ability (measured by the CC test) correlated to performance on teleoperation subtasks such as pickup and docking?
- iii. Is gender correlated to performance on simulated space teleoperation tasks?

## 3.2. Hypotheses

According to the description of space teleoperation activities and human spatial ability addressed above, we predicted the following answers to the experimental questions:

- i. **Effect of perspective-taking ability on pickup and docking task performance:**

We hypothesized performance is affected by the subject's perspective-taking ability. We expected subjects with better perspective-taking ability (higher scores on PTA or PSVT:V test) would perform faster, with more continuous and accurate motion, fewer collisions, and use more axes simultaneously during translation than those with poor perspective-taking ability. We also expected such subjects to have a higher docking accuracy in docking tasks. In particular, we expected that:

  - a. Subjects with high scores on the PSVT:V test would perform better than those with low scores when the cameras' lines of view were more widely separated in angle. Their ability to perform better at larger angular separations would be reflected in faster and more accurate translation, higher docking accuracy in position and orientation, and the simultaneous use of a larger number of axes during translation.
  - b. High PTA scorers would perform better than low scorers when there was a horizontal misalignment between the control reference frame and the display reference frame. Their ability to perform better at larger misalignments angles should be reflected in shorter times to perform a task and smaller translational errors.

ii. **Effect of spatial visualization ability on pickup and docking task performance:**

We hypothesized that performance would be affected by the subject's spatial visualization ability.

We expected subjects with better spatial visualization ability (higher scores on the CC test) would perform better in rotational motion than low scorers. While manipulating the end effector, we expected them to rotate about more standard axes at any one time, as a way to improve their efficiency. We also expected such subjects to have a higher orientation (angular and roll) docking accuracy in docking tasks.

iii. **Effect of gender on pickup and docking task performance:**

If women have, as reported, lower spatial ability than men, we expected they may translate along fewer axes simultaneously, require longer task times and deliver lower docking accuracy.

# Chapter 4. Methods

To test these hypotheses, and based on the recommendations from the preliminary study, we conducted a main experiment according to the following design.

## 4.1. Camera configurations

Table 1 defines the three configurations (A, B and C) were defined, where A and B were aligned to the control reference frame and C was horizontally misaligned by an angle  $\beta = 90^\circ$ . The angular separation,  $\alpha$ , between the cameras in A and C was the same ( $55^\circ$ ), and cameras in B had a greater separation ( $80^\circ$ ). Note that unlike the preliminary study, the camera view axes do not lie one same plane, and so formally speaking there is no “central” camera. As in the preliminary experiment, the setting of three cameras was fixed.

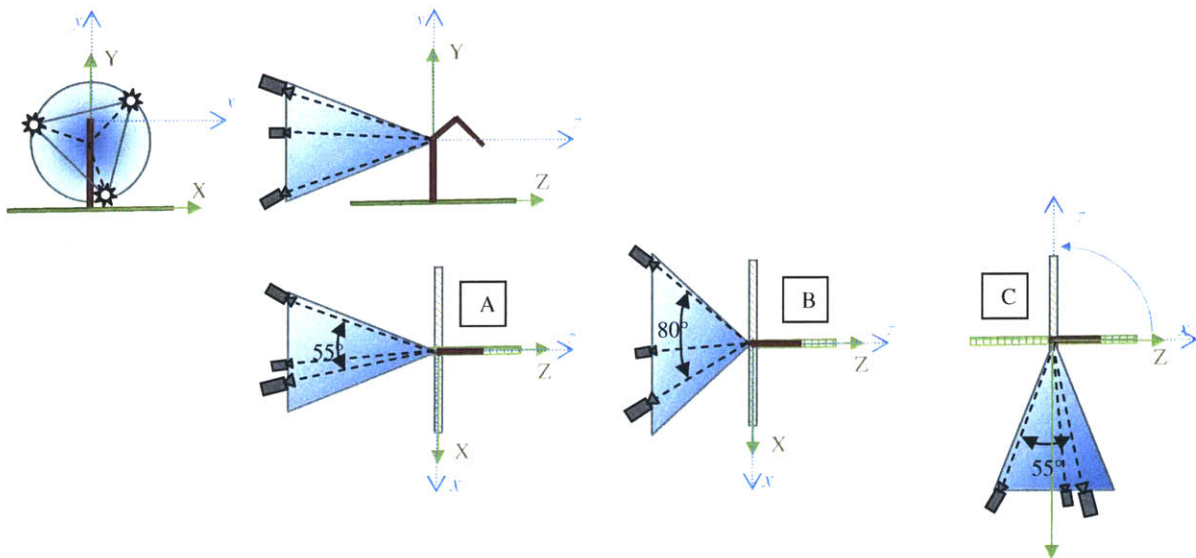


Figure 8. Camera configurations.

Top –

Side (left) and front (right) view of the robot arm mounted on two fixed trusses. Lines of view of three fixed cameras -located on the surface of a cone- point towards the shoulder of the arm. Solid arrows represent the control reference frame (XYZ). Dotted arrows represent the display reference frame (xyz).

Bottom –

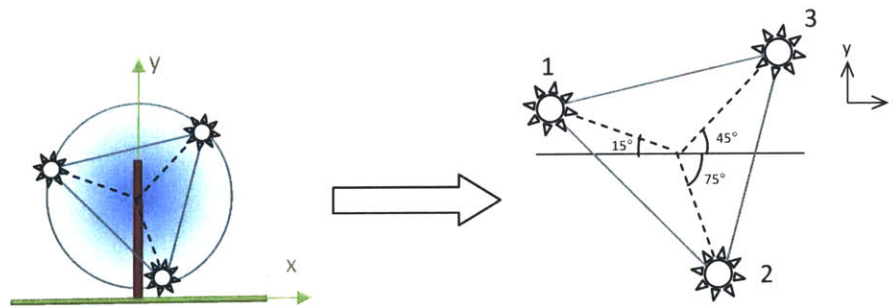
- A (left), with 55 degrees separation between the lines of view of the cameras, reference frames are aligned.
- B (center), with 80 degrees separation between the lines of view of the cameras, reference frames are aligned.
- C (right) with 55 degrees separation between the lines of view of the cameras, reference frames are misaligned by 90 degrees.

**Table 1. Description of the camera configurations in terms of angular separation between the camera views and angular misalignment between the control and display reference frame**

Configuration	Angular separation $\alpha$	Angular misalignment $\beta$
A	55°	0°
B	80°	0°
C	55°	90°

Configuration B was designed –similarly to configuration  $\beta$  in our preliminary study– to challenge the subjects’ three-dimensional perspective-taking ability by increasing the angle between the camera’s lines of view. Configuration C, in contrast, was designed to challenge the subjects’ two-dimensional perspective-taking ability by increasing the angle between the control and the display reference frame.

The main difference between configuration  $\gamma$  in the preliminary study, and configuration C in this study is the magnitude of the horizontal misalignment (180° in  $\gamma$ , and 90° in C). The rationale behind this change was our suspicion that subjects might have used strategies (e.g. simple control reversal rules) different from perspective-taking when being presented to an exact left-right reversed configuration, as in configuration  $\gamma$ . We chose a 90° misalignment so that the control-display reference frame misalignment would be large enough to challenge perspective-taking, but not so large as to motivate subjects to use strategies different from perspective-taking to understand the camera views. Even though greater misalignments (e.g., 135°) would have provided a consistent left-right reversal in all the cameras, we feared that larger separation would only be understood by the best perspective takers, forcing the average subject to use alternative strategies to perform the task. We believe, however, that this assumption should be addressed in future studies.



**Figure 9. Location of the three camera views relative to the horizontal axis x. The three cameras are mounted on the corner of an equilateral triangle parallel to the plane xy of the display reference frame.**

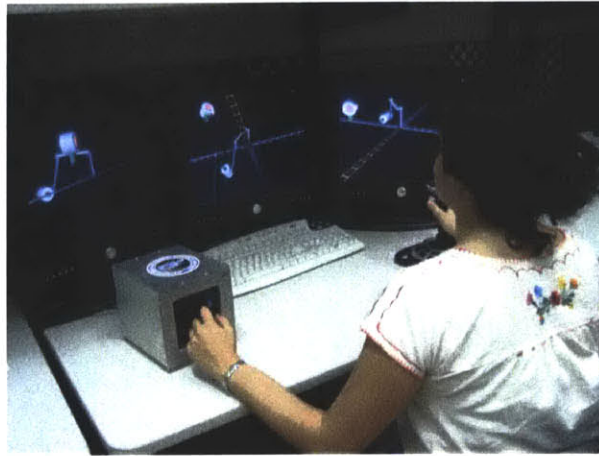
The three cameras were placed on the corners of an imaginary equilateral triangle, with the normal vector to its surface pointing towards the shoulder of the robot arm, as shown in Figure 9. The triangle was rotated around its center point so that all cameras were as separated as possible from the vertical or horizontal planes. The cameras 3, 1, and 2 were always located at 45, 165 and 285 degrees (counterclockwise) respectively, with respect to the horizontal plane (defined by the left/right and



front/back of the control frame). The distance from each of the cameras to the shoulder of the robot arm was the same.

## 4.2. MIT Remote Manipulation System Simulator (MIT RMSS)

A generic Robotic Workstation (RWS)-like simulator was developed at MIT. It included three display screens, a Translational Hand Controller (THC), a Rotational Hand Controller (RHC) and a keyboard, as shown in Figure 10.



**Figure 10. MIT Remote Manipulator System Simulator**

The displays were Dell 17" flat LCD screens located at the same distance (3' approx.) from and pointing towards the subject. The simulation was run on two Windows computers using Vizard's networking capabilities. The main simulation server (1.5GHz Pentium4 PC with dual head nVidia GeForce 6600 graphics) performed the kinematic calculations, graphics processing and hand controller I/O. The second computer (550MHz Pentium3 PC with nVidia GeForce3 graphics) rendered the third camera viewpoint.

The images shown in the left, central and right computer monitors displayed the views from the left, central and right cameras, respectively.

The two hand controllers, modeled after the RWS, controlled the translation and rotation of the robot end effector. For the translational hand controller (THC) a 3DOF linear joystick was custom-built from a 2DOF joystick, a linear potentiometer and a USB controller card. The controllers had a constant force/torque gradient, and central dead zone in all the degrees of freedom used.

The two hand controllers, modeled after the actual RWS joysticks, controlled the translation and rotation of the arm's end effector. For the translational hand controller (THC, Figure 11 left) a 3DOF linear joystick was custom-built from a 2DOF joystick, a linear potentiometer and a USB controller card.

This THC can be moved in three different directions: up/down, right/left and front/back. The rotational hand controller (RHC, Figure 11 right) was a commercial Logitech Extreme3DPro USB game controller that had 3 DOF (right/left, front/back, and twist). The controllers had a constant force/torque gradient, and small central dead zone in all the degrees of freedom used. On the THC, the force gradient was the same for the up/down and left/right motion, and higher for the front/back motion. Similarly, on the RHC it was the same for front/back (pitch) and right/left (roll), but lower for twist yaw).

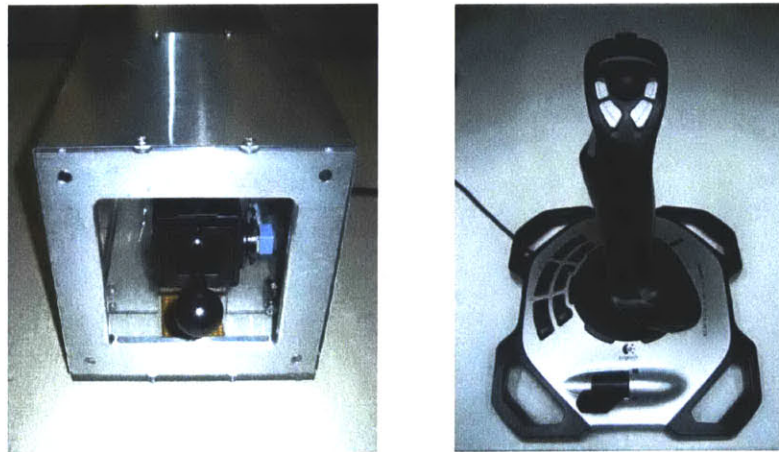


Figure 11. Translational Hand Controller (left) and Rotational Hand Controller (right)

### 4.3. Environment configuration

Using the Vizard 3.0 VR development package (WorldViz, Santa Barbara, CA.), a simulated RMS workspace was created that was similar to the BORIS training software used in General Robotics Training<sup>7</sup>. It included a Unimation PUMA (Programmable Universal machine for Assembly)-class 6-degree-of-freedom (DOF) robotic arm of similar dimension to the Shuttle RMS (Figure 12). Three of the joints on the arm provided the translation of the end effector (the tool at the end of the arm used to grab the payload), and other three joints rotated it. The software library RRG Kinematix v.4 (Robotics Research Group, Univ. of Texas) was incorporated into a plug-in module for Vizard to calculate the forward and inverse kinematics of the robot arm. The dynamics of the arm and other objects were not modeled in this simulation.

Subjects used the THC to control the linear rate of translation (1.8 m/s max) of the end effector along the three orthogonal axes of a fixed external control reference frame. (Table 2) Subjects used the RHC to

---

<sup>7</sup> The main code for this simulation was written by Dr. Andrew M. Liu. Minor adaptations related to the specific design of this experiment were implemented by M. Alejandra Menchaca-Brandan. The complete code is documented in APPENDIX G.

control the angular rates of the Euler angle rotations (180°/s max) that define the end effector orientation. The RHC degrees of freedom were mapped to the Euler angles is shown in Table 3. In this implementation of rotation control, pitch control (e.g. rotation about the X-axis of the control frame) is, in fact, performed using the same external control reference frame as translation control. However, because an Euler angle representation involves rotations about axes of a moving reference frame, the subsequent Y- and Z-axis rotations are usually not aligned with the external control reference frame.<sup>8</sup>

**Table 2. Mapping of the THC joystick input on the end effector translation**

THC Joystick motion	End effector axis of translation
Left/right	-X/+X
Up/down	+Y/-Y
In/out	+Z/-Z

**Table 3. Mapping of the RHC joystick input on the end effector rotation**

RHC Joystick motion	Euler angle rotation axis
forward/backward	X-axis
Left/right	Z-axis
twist	Y-axis

It should be noted that the simulation does not exactly replicate the actual RMS control modes, which is typically in either an external or internal reference frame for both translation and rotation.

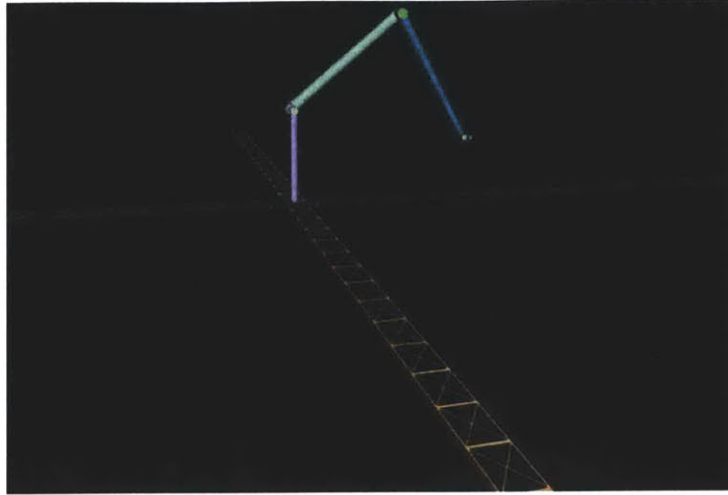
The base of the arm was mounted on the intersection of two perpendicular structural trusses, which replaced the single fixed truss used in the preliminary experiment. These trusses served as obstacles and provided additional environmental cues, promoting the use of perspective-taking to perform the task. One end of one truss was colored differently (orange) than the rest (gray), in order to indicate the forward direction of the control reference frame.<sup>9</sup>

---

<sup>8</sup> If the X-axis rotation is zero, then y-axis rotation will be aligned with the external control frame. Similarly, if both x- and y-axis rotation is zero, then Z-axis rotation is aligned with the external control frame

<sup>9</sup> Before running the final experiment we tested three subjects on an environment where the trusses were all colored differently and subjects were instructed to use the colors of the trusses to identify the orientation of the control reference frame. We discovered, however, that the truss colors provided so much visual information about the orientation of the control reference frame, that all of the subjects used mental rules to perform the task (e.g., “I knew that pushing the hand controller would always move the end effector along the blue truss”) and none used their perspective-taking ability. Moreover, none of them perceived any changes between camera configurations. For the final experiment, we decided to remove most of the colors from the trusses, and leave a more discrete visual indication of the control reference frame orientation. We also stopped instructing the subjects to use to color of the truss as an orientating cue. We expected these final changes would help subjects to orient themselves, without forcing them to use a specific manipulation strategy.





**Figure 12. Robot arm mounted on two large fixed trusses.**  
The base of the arm was 2.5 m<sup>10</sup> long and the forearm and upper arm, 4 m long.

The three virtual camera viewpoints of the environment (described in section 4.1. ) were displayed on the three computer monitors of the MIT Remote Manipulation System Simulator. The viewpoints had a field of view of 40.

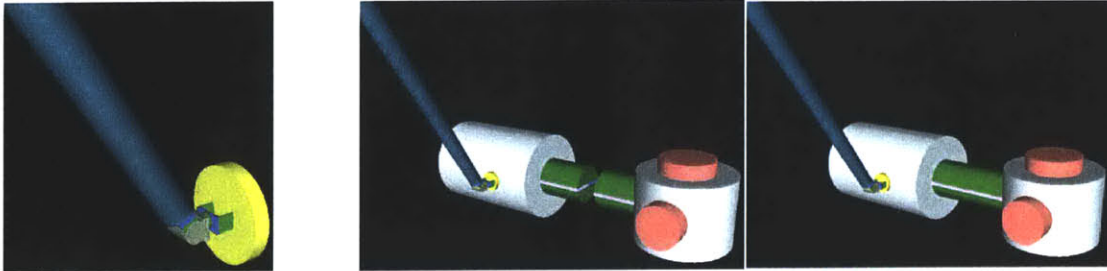
## 4.4. Task

The preliminary study showed that mastering rotational hand controller skills may require considerably more time than originally provided. As a consequence, we divided the experiment that originally involved pickup and docking within the same trial, into two separate phases: pickup and docking. On each pickup trial –which required mostly translation– subjects were instructed to manipulate the arm to capture a payload dish by its front (yellow) face (Figure 13, left). Pickup happened automatically after touching the dish with the end effector. During the docking trials –requiring both translation and rotation of the end effector–, subjects had to translate a cargo module (already attached to the end effector) towards an ISS node and assemble them by aligning both docking ports. Each of the docking ports had a blue and a pink stripe, one on each side of the port, which had to be matched (in color) during docking (Figure 13, center and right). As long as the port was visible, at least one of the stripes was visible too. Subjects pressed the space bar when they considered that the module’s docking port

---

<sup>10</sup> Although the size of the arm was specified in meters, in this thesis we use “unit length” as the measure of distance. 1 unit length = 1 m.

was as accurately aligned with the node's port as possible (that is, coaxial and with the stripes on both docking ports properly aligned, Figure 13). No feedback regarding time or accuracy was provided.



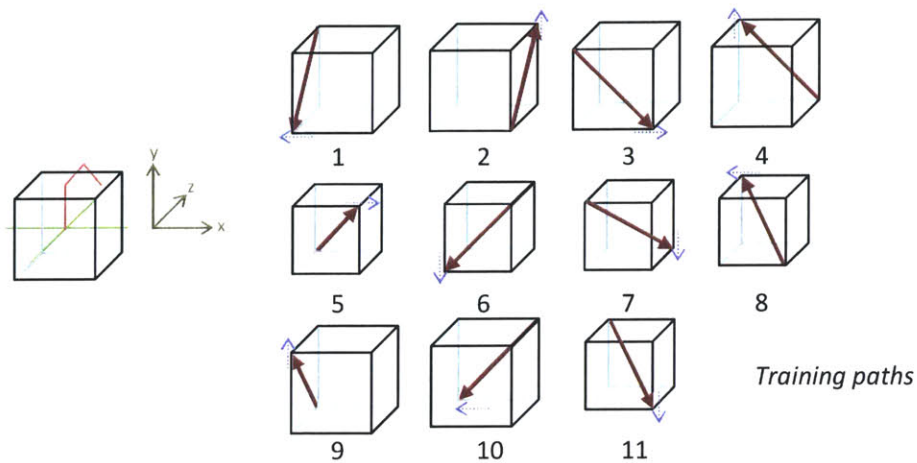
**Figure 13. Pickup of the payload dish (left) and docking of the cargo module onto the ISS node (center and right)**

The initial position of the robot arm, as well as the position and orientation of the payload dish, cargo module and ISS node changed between trials. As shown in Figure 14 the ideal trajectories from the point of departure to the ideal final point were defined by the lines connecting the vertices of a cube centered on the intersection of the structural trusses. Two-dimensional trajectories were selected from those paths on the sidewalls of the cube, and the three-dimensional trajectories matched those paths connecting opposite corners of it.

The payload dish and the ISS node were oriented in such way that the last part of each trial (grasping in pickup phases, and docking in docking phases) had to be performed in a specific direction, as indicated by the short arrows in Figure 14. For instance, if the surface of the payload dish was pointing parallel to the +X direction, the end effector had to be moved in the -X direction in order to grasp the dish. Similarly, if the node's docking port was facing down, the module had to be translated upwards to be docked. Therefore, on each trial the target objects were oriented to allow one of the following four orientations (Figure 14): up (+y, with respect to the observer's reference frame), down (-y), right (-x) or left (+x).

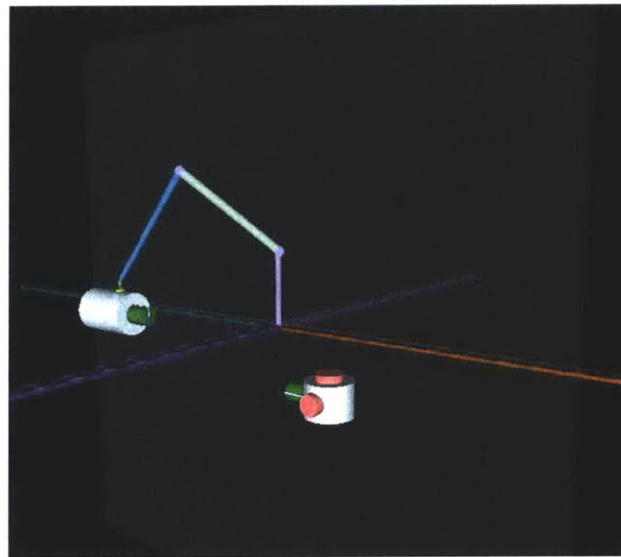
The initial orientation of the cargo module was defined so that only one 90-degree rotation around an external reference frame axis was needed to dock with the node for the right/left docking directions, and an additional 45-degree rotation around another axis was needed for the up/down directions.

A given ideal trajectory was used in both the pickup and docking phases to allow comparison between phases. These paths, as well as the orientation of the objects, were always aligned with the observer's reference frame such that the paths looked the same regardless of the camera configurations. For statistical analysis, this also eliminated the cross effects of path and configuration. All paths had the same length.



**Figure 14. Trial paths (thick arrows) defining the starting and ending point of the arm's end effector. The center of each cube is defined by the intersection of the two trusses (left). Trials can be two-dimensional (1-4) or three-dimensional (5-8). The training paths are also shown (9-11). Short gray arrows indicate the direction of the pickup/docking ( $\pm x, \pm y$ ). Paths are defined with respect to the display reference frame (the arrangement of the objects looks the same in configurations A and C).**

In the docking trials using two-dimensional paths, a semi-transparent vertical wall was placed in alignment with one of the structural trusses to constrain the space within which to move the arm. This was to reduce the large variance in path distances observed in the preliminary study. Subjects were instructed to cross through the wall when manipulating the robot.



**Figure 15. Two-dimensional path with semi-transparent wall**

Subjects were notified of a collision by a red message displayed on the top right corner of each display. Collisions were defined as the intersection of any two objects in the environment, including the end effector with other links of the robot arm. Only the contact between the cargo module and the node was not considered a collision. A blue warning sign appeared on the screens when the robot arm passed through a kinematic singularity, which sometimes resulted in losing control of the arm and having to repeat the trial (this situation only happened twice in the experiment).

## 4.5. Task performance metrics

During each trial, the position and orientation of the end effector, and a collision indicator were recorded at 60 Hz during pickup trials, and 30 Hz during docking trials. From these data, the following metrics were calculated to assess the operator’s performance. Some of the metrics were derived from Akagi et al [27] who analyzed the correlation of various measures of robot arm operator performance, and defined a set of key measures to be used as teleoperation performance metrics. A summary of the metrics is presented in Table 4.

**Table 4. Measures of performance for each phase. The third and fourth columns indicate which measure was used to assess performance on each phase.**

Measures of performance	Description	Pickup Phase	Docking Phase
Observation time (T_obs) <sup>11</sup>	Time between the initiation of the task and the first hand controller input	X	X
Task time (T_task)	Time required to complete the task between first and last hand controller inputs	X	X
Confirmation time (T_conf)	Time between the final hand controller input and the end of the trial		X
%motion	Percent of T_task during which the end effector was moving (active time)	X	X
Axial time (T_ax)	Total time where the end effector was translating	X	X
Angular time (T_ang)	Total time where the end effector was rotating	X	X
Collision time (T_coll)	Total time during which a collision was detected	X	X
# of collisions (CollNum)	Number of collisions per trial	X	X
Degree of inverse motion (DIMx, DIMy, DIMz)	Distance travelled away from the target (payload dish or ISS node docking port) on each axis (world/control reference frame)	X	X
Axial DOF input (DOFax)	Average of simultaneous use of axial degrees of freedom (DOF) during T_task	X	X

<sup>11</sup> Total time to complete a trial = Tobs + Ttask + Tconf

	(DOF <sub>ax</sub> = 1 if the subject never moved on more than one standard axis <sup>12</sup> at a time; DOF <sub>ax</sub> = 3 if the subject moved on the three axes every time throughout the task)		
Angular DOF input (DOF <sub>ang</sub> )	Average of simultaneous use of angular degrees of freedom (DOF) during T <sub>task</sub> (DOF <sub>ang</sub> = 1 if the subject never rotated around more than one standard axis <sup>13</sup> at a time; DOF <sub>ang</sub> = 3 if the subject rotated around the three standard axes every time throughout the task)	X	X
%bimanual	Percentage of active time in which both hand controllers were used simultaneously	X	X
Axial offset (Ax_Offs)	Radial distance between the axes of the two docking ports (cargo module and ISS node) (Figure 16)		X
Angular offset (Ang_Offs)	Angular separation between the axes of the two docking ports (Figure 16)		X
Roll offset (Roll_Offs)	Angular separation between the lateral marks in the two docking ports (Figure 16)		X

**Time to complete the task, observation and confirmation time (T<sub>task</sub>, T<sub>obs</sub>, T<sub>conf</sub>):** For both pickup and docking subtasks, the task time was defined as the time between the moment of the first movement of the arm and its last. The period of time between beginning of the trial and the arm's first movement was called observation time (T<sub>obs</sub>), assuming that the subject used those seconds to analyze (observe) the scene before reacting. Similarly (in docking trials only), the period of time between the arm's last movement and the moment at which the subject decided to terminate the trial –by pressing the space bar– was called confirmation time (T<sub>conf</sub>). We assumed that this time was used to confirm the accuracy of the docking. The total trial time was divided into these smaller components, after the belief that each of these times provided different information on the subject's strategy to complete a trial.

**Axial and angular time (T<sub>ax</sub>, T<sub>ang</sub>):** In each docking trial, two additional times were measured: the time during which the subject was exclusively translating, axial time (T<sub>ax</sub>), and exclusively rotating, angular time (T<sub>ang</sub>). These additional measurements were included in order to analyze the individual parts of the docking task separately. In particular, comparing the time spent during parallel parts of the two phases (pickup and docking) was of particular interest. The parallel times were T<sub>ax</sub>, the time of linear motion during docking, and T<sub>task</sub>, the task time during pickup. This choice was influenced by the fact that pickup requires almost no rotation. Therefore, the difference between T<sub>ax</sub> (docking) and T<sub>task</sub> (pickup) is expected to reflect the intrinsic difference between pickup and docking on otherwise comparable tasks.

---

<sup>12</sup> Standard axes for translation are referenced to the control reference frame.

<sup>13</sup> Standard axes for rotation are referenced to the end effector's local reference frame.

**Collision time and number of collisions (T\_coll, CollNum):** The total number of collisions per trial (CollNum), and the total time spent in a collision condition (T\_coll) were measured. It was unknown whether subjects would spend different times colliding (e.g., one subject could have collided ten times within ten seconds, and another could have one single ten-second long collision), and thus both measures were defined.

**Percent of motion (%motion):** This index of continuity in the motion of the robot arm corresponded to the fraction of the time T\_task spent in motion (whether linear or rotational). A subject's style, related to the continuity of his motion toward the target, was the fraction of the time that he spent moving as opposed to pausing or thinking.

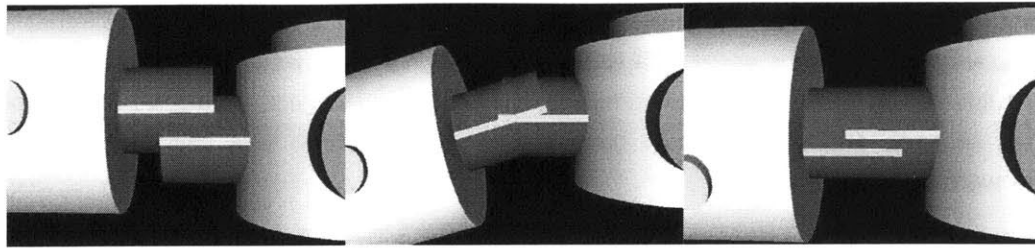
**Axial and angular input of degrees of freedom (DOFax, DOFang):** The number of axes along which the arm was linearly moved at any one time was called axial input of degrees of freedom (DOFax). Similarly, number of (standard) axes about which the arm was rotated at any one time was called angular input of degrees of freedom (DOFang). These measures, adapted from Akagi et al [27], were intended to characterize a subject's style for controlling the robot arm: a conservative subject would tend to move/rotate about only one axis at a time, while a subject who was more confident of their actions or more of a risk taker would move/rotate about multiple axes at a time. A change in control movements in from one axis to two would indicate an increase in the subject's confidence to move within the environment.

**Percent of bimanual control (%bimanual):** Subjects could use both controllers simultaneously when operating the arm, in an effort to decrease the task time. The fraction of the task time (T\_task) spent in bimanual control was called %bimanual. This measure would account for those subjects that might have been moving along/about one single axis at a time, but with both controllers at the same time.

**Degree of inverse motion (DIMx, DIMy, DIMz):** This measure, initially proposed by Akagi et al. [31], characterize the subject's efficiency of translation. Smaller values indicate less extraneous (and hence, more efficient) robot arm movement. The degree of inverse motion in the direction of, say, the X axis (DIMx) is defined as the linear distance traveled away from the target in the X direction during a trial. Correspondingly, inverse motion in the Y and Z directions are DIMy and DIMz.

**Docking offsets (Ax\_Offs, Ang\_Offs, Roll\_Offs):** Accuracy of docking was measured by (1) axial offset (Ax\_Offs), or the radial distance between the cargo module and the ISS node docking ports; (2) angular offset (Ang\_Offs), or the angle between the docking ports' axes; and (3) roll offset (Roll\_Offs), or the angle between the docking stripes .





**Figure 16. Docking offsets:**  
**Axial offset (Ax\_Offs, left): radial distance between the docking ports**  
**Angular offset (Ang\_Offs, center): angle between the docking ports**  
**and Roll offset (Roll\_Offs, right): angular separation between the stripes on the docking ports**

## 4.6. Spatial ability metrics

The metrics used to measure the subjects' spatial ability were the Cube Comparisons test (CC), the Perspective-Taking Ability test (PTA), and the Purdue Spatial Visualization test: Visualization of Views (PSVT:V). CC is a paper-and-pencil test designed to measure individual's mental rotation ability. These tests are described in the Background section.

The CC test score was calculated by counting the number of correct answers minus incorrect answers. The published PSVT:V test was untimed, however, a six-minute constraint was set for this experiment. This constraint encouraged subjects to work quickly, utilizing PT skills rather than other non-PT strategies. None of the subjects had time to solve more than 26 out of the 30 trials. The net corrected score was the number of correct answers (-) one fourth of the number of incorrect answers, to compensate for the estimated number of correct answers arrived at by random guess. (The original scoring method does not account for random guesses.)

## 4.7. Subjects

The experimental protocol was reviewed and approved by MIT's institutional human subject experimental review board. A total of 26 subjects (11 female, 14 male) was tested. One female subject elected not to finish the last 8 trials, because of mild eye strain discomfort –she had forgotten her eye drops. Their ages ranged from 22 to 34, and all except for three were or had been MIT students. The other three subjects were professionals. Five subjects were left handed. The majority of the subjects said they used the computer between 5 and 7 hours a day, and did not currently play videogames. Seven of them had never had any experience with video games or computer games. They received \$10/hour compensation for their participation.

## 4.8. Procedure

The experiment was divided into two sessions. During the first session (45 minutes on average), subjects answered a questionnaire with general information and specific questions regarding their daily use of the computer and their previous experience with video games, virtual reality software and game controllers. Finally, they completed three spatial ability tests in the following order: Cube Comparisons test (CC), Purdue Spatial Visualization Test: Visualization of Views (PSVT:V) test, and Perspective-Taking Ability test (PTA). If the subjects had previously taken any of these tests as part of a previous MVL experiment then their scores from the previous test were kept to avoid any effect of learning or familiarity. They were not required to re-take these tests.

The second session (from 1.5h to 4h, 3h on average) focused on space teleoperation testing. Subjects were introduced to the simulation using a 15-minute Power Point Presentation (theoretical training), included as Appendix J. This presentation described the environment configuration, the kinematics of the robot arm, the use of the hand controllers, the procedures to pickup the payload and dock the cargo module onto the ISS node, the rules to follow (e.g., avoiding collisions and singularities, and remaining within the work area), and the overall structure of the experiment. They were instructed to do the task as fast and accurately as possible. During this training, they were given two 2-minute sessions to practice the use of the translational and rotational hand controller, respectively.

To ensure that all subjects had a basic understanding of the tasks, they were asked to verbally repeat the main rules to be followed throughout the experiment, and were reminded of those they could not remember.

Each phase had 8 testing trials. These trials were balanced according to the dimensionality of the paths (2D, 3D), the direction of the pickup and docking (up/down/right/left), and the vertical direction of motion from initial to final point (up/down). The trial order was the same for both phases<sup>14</sup>.

Two groups of subjects were randomly selected, and the groups were balanced based on gender and CC scores: both groups had the same number of males and females, and the mean CC scores were kept as similar as possible (Group 1 = 20.25, Group 2 = 21.24). We chose CC test scores to balance the groups because in the preliminary study CC provided the strongest effects and the test that has been validated over a greater population of subjects. Each group performed the different camera configurations of the experiment on a different sequence, to account for increase or decrease in performance due to learning or fatigue effects, respectively. Group 1 followed the sequence of configurations A-B-C, whereas Group 2 performed the experiment using the configurations in the order A-C-B (Table 5).

---

<sup>14</sup> Path order: 5 – 2 – 3 – 8 – 6 – 1 – 4 – 7. Training path order: 9 – 10 – 11. Sections with only one training trial included path 9. For more detail on the paths, refer to Figure 14.



**Table 5 Experiment procedure. p = pickup, d = docking, tp/td = training pickup/docking**

	Training + Testing											
Phase	1	2	3	4	5	6	7	8	9	10	11	12
Configuration (Group 1)	A				B				C			
Configuration (Group 2)	A				C				B			
Section (phase)	tp	p	td	d	tp	p	td	d	tp	p	td	d
Trials	3	8	1	8	1	8	0	8	1	8	0	8

Configuration A included three pickup and one docking practice trials. Configurations B and C involved only one training trial in the pickup phase, to adapt to the new camera configuration. A total of 54 trials were completed during the experiment.

Subjects took a 5-minute break between each section. During this time they talked to the experimenter about their impressions of the recently finished part. Subjects were questioned verbally (e.g., “tell me about your strategies”, “how did you find the views?”, “how did you feel your translation/rotation movements?”, “did you notice any difference with respect to the last sections?”) The same questions were asked to all subjects, and they were deliberately phrased to avoid providing hints that could affect their performance in the forthcoming sections. These inter-session questions are listed in APPENDIX C.

At the end of the experiment, subjects completed a final written questionnaire to gather information regarding the strategies they used to perform the different tasks, possible difficulties encountered while using the controllers, and assess potential discomforts caused by the task.

# Chapter 5. Results

This chapter provides an overview of the statistical analysis of the performance data and the spatial ability test scores. Then, each hypothesis (as listed in section 3.2. ) is addressed individually, dividing the results by task phase (pickup or docking).

We analyzed the relationship between the tested measures of spatial ability and performance using mixed regression modeling (Systat v.11). The fixed effects in the model were: camera configuration (A, B, or C), trial path (1-8, see section 4.4. ), gender, group (1 for subject who completed the camera configurations in the order A-B-C, and 2 for those who followed A-C-B), standardized age, previous videogame experience, and the cross effect of group \* camera configuration. Since we found no correlation between the score on the CC test and either perspective-taking test score (PTA or PSVT:V), we used two separate models to examine the effect of those test scores (either CC and PTA, or CC and PSVT:V). Those test scores were included as covariates, each standardized (converted into a Z value) over the subject sample. We also examined the cross effect of the corresponding perspective-taking test (PTA or PSVT:V) with configuration. The only random effect was subject.

**Table 6 List of independent (left) and dependent (right) variables used in the statistical analysis. Variables marked with ~ were only used in the analysis of the docking phase.**

<b>Independent variables</b>	<b>Type</b>	<b>Dependent Variables</b>
PT score (PTA or PSVT:V)	<i>covariate</i>	T_task
CC score	<i>covariate</i>	T_ax ~
Configuration * PT score	<i>category</i>	T_ang ~
	A*PT,B*PT,C*PT	T_obs
Configuration	<i>category</i>	T_conf ~
	A, B, C	T_coll
Group	<i>category</i>	%motion
	1, 2	DOFax
Gender	<i>category</i>	DOFang
	Female, Male	%bimanual <sup>15</sup>
Trial path	<i>category</i>	DIMx
	1 – 8	DIMy
Videogame Experience	<i>category</i>	DIMz
	0/1	Ax_Offs ~
SetCam	<i>category</i>	Ang_Offs ~
	0/1	Roll_Offs ~
Configuration * Group	<i>covariate</i>	
	A*1, B*1, C*1	
	A*2, B*2, C*2	
Age	<i>covariate</i>	

<sup>15</sup> The values obtained for this measure of performance were so low that it would not be analyzed.

Table 6 lists the independent and dependent variables used in the analysis. One regression model was obtained for each phase (pickup or docking), combined with each dependent variable (10 for pickup, 16 for docking), for each PT tests, PTA and PSVT:V. A total of 52 regressions were obtained.<sup>16</sup>

During the breaks between configurations we asked subjects to describe the location of the cameras: Some subjects knew with precision where the cameras were located, and were even surprised at the question (“of course I know where the cameras were”); in contrast, the other subjects were unable to describe anything related to the camera location (“I don’t know. Maybe up, maybe down”, “I didn’t really pay attention”, or even “understanding the camera locations would have required too much workload, I didn’t need to know it”). We thought that this added information might correlate with performance, so we defined a category variable (SetCam: 0= no knowledge of camera locations, 1= correct knowledge of camera locations) which we included in the model.

In our preliminary analyses we found no clear effect of the number of hours subjects used a computer or of the number of hours they spend playing video or computer games. We did not include these as independent variables in the final analysis. We had few left-handed subjects, all male, and for that reason did not include handedness in our model.

To conform to the regression model assumptions of comparable variance and normal distribution of the residuals, we analyzed the logarithms of the time and accuracy data, not the original numbers themselves. The %motion and DOF values (i.e. DOFax, DOFang), were treated after an  $\arcsin(\sqrt{\cdot})$  transformation, since the data ranges were intrinsically constrained.

Four performance metrics, collision time (Tcoll), multiaxial linear motion (DOFax), and multiaxial rotation (DOFang) were not normally distributed and had to be analyzed separately. The analysis was done instead with the average values for each phase (pickup or docking) within each configuration (6 data points per subject). The number of collisions (CollNum) was not included as a factor in the model because it was strongly correlated with the collision time ( $R = 0.859$ ).

The analysis included all the measurements taken for 24 subjects –all our subjects except two. Subject 29 was removed because his test scores and performance were very different from and uncharacteristic of the pattern seen in the other subjects<sup>17</sup>. The data corresponding to configuration A of subject 15 was also removed because he did not perform the same experiment as the others: he reported not understanding the instructions and treating those test trials as training trials.

---

<sup>16</sup> A summary of all the effects found in the several mixed regressions can be viewed in APPENDIX H.

<sup>17</sup> The subject’s scores on the three tests were the highest among the subjects (his scores on PTA and PSVT:V were the highest we have recorded in the laboratory). He finished the test in 1 hour, half the time of the next fastest subject, and his accuracy was the best of any.

Since the PTA test has been validated among a larger pool of subjects, we present the full results of the mixed regression model only for PTA, and for brevity, not for PSVT:V. We describe the PSVT:V regression model results only that illustrate the difference between the two tests on predicting task performance. The complete set of results for both tests is provided in 0.

## 5.1. Overview of spatial ability test scores

The descriptive statistics of the spatial ability test scores are presented in Table 7.

**Table 7 Descriptive statistics of spatial ability test scores**

Test	Mean (Median)	SD	Max	Min	Male Mean	Female Mean
PTA	19.97 (21)	7.79	32	7	24.0	19.5
PSVT:V	21.42 (21)	4.52	29	8.5	16.0	11.4
CC	13.60 (13.5)	4.81	21	1.25	20.1	21.7

The PTA scores of the subjects in our sample were distributed roughly normally, but the other two test scores, for PSVT:V and CC, had nearly uniform distributions. There was no significant correlation between CC scores and either perspective-taking test, PSVT:V (Figure 17a) and PTA (Figure 17b), although a previous experiment in our laboratory<sup>18</sup> found such a correlation between PTA and CC ( $R = 0.0.648$ ). We did find, however, that PTA and PSVT:V were significantly correlated ( $R = 0.577$ ) among our subjects (Figure 17). The majority of our subjects reported using the appropriate mental strategy used for each spatial test.<sup>19</sup>

---

<sup>18</sup> Hirofumi Aoki, personal communication.

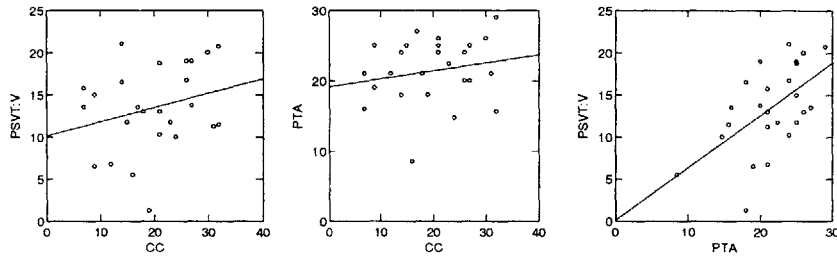
<sup>19</sup> For the CC test, the majority of our subjects visualized rotations of the cubes. They did not use other strategies as imagining taking a particular perspective or using simplifying formal rules that did not require the use of spatial abilities, as mentioned by Just and Carpenter in [16].

For the PTA test, most subjects used perspective-taking tactics. Some reported using mental rotation of arrows as described by Hegarty [13], and others used other strategies that did not require spatial ability by predicting by predicting the region in which the answer would fall in order to reduce the response time.

Finally, most of the subjects approached the PSVT:V test by taking the designated perspective. The few who attempted to use mental rotations (either by rotating the glass cube and comparing it to the multiple choices, or rotating each of the multiple choices into the desired position) shortly realized that their strategy was time-consuming and soon switched to perspective-taking. One subject, who had the lowest PSVT:V score, reported that he could not understand which strategy to use to find the right answer. His PTA score, however, was close to the mean.

These observations on strategies summarize the experimenter's impressions based on discussions with most, but not all subjects.

On both perspective-taking ability tests (PTA and PSVT:V) males subject outperformed females by a small difference. On the CC test, however, females had a slightly (non-significant) higher mean score. While the scores for male subjects repeat those reported by Ekstrom et al. [10], those for females subjects are twice as high. This probably originates due to the fact that most female subjects were MIT engineering students, who arguably were preselected for spatial skills.

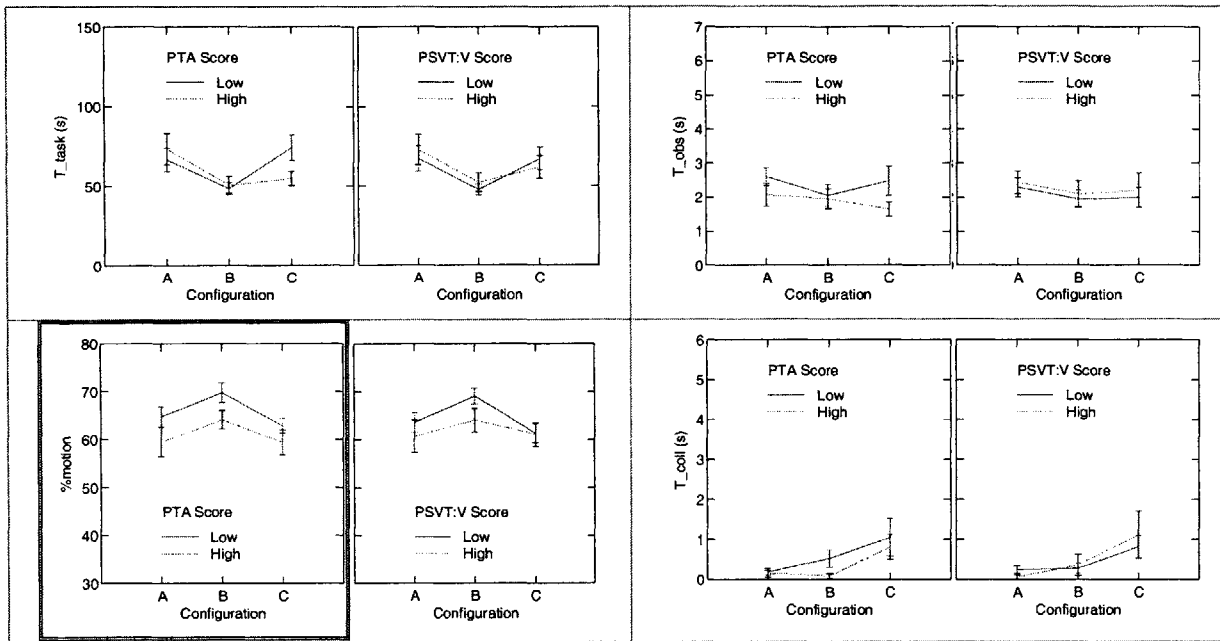


**Figure 17. Correlation between the spatial ability tests:  
a) PSVT:V vs. CC, b) PTA vs. CC, c) PSVT:V vs. PTA**

## 5.2. Effect of perspective-taking ability

### 5.2.1 Pickup phase

Figure 18 and Figure 19 show the trends of the most relevant pickup performance metrics, between configurations. Performance is grouped into high and low PTA and PSVT:V score groups<sup>20</sup> divided by the sample mean. (See Table 7)

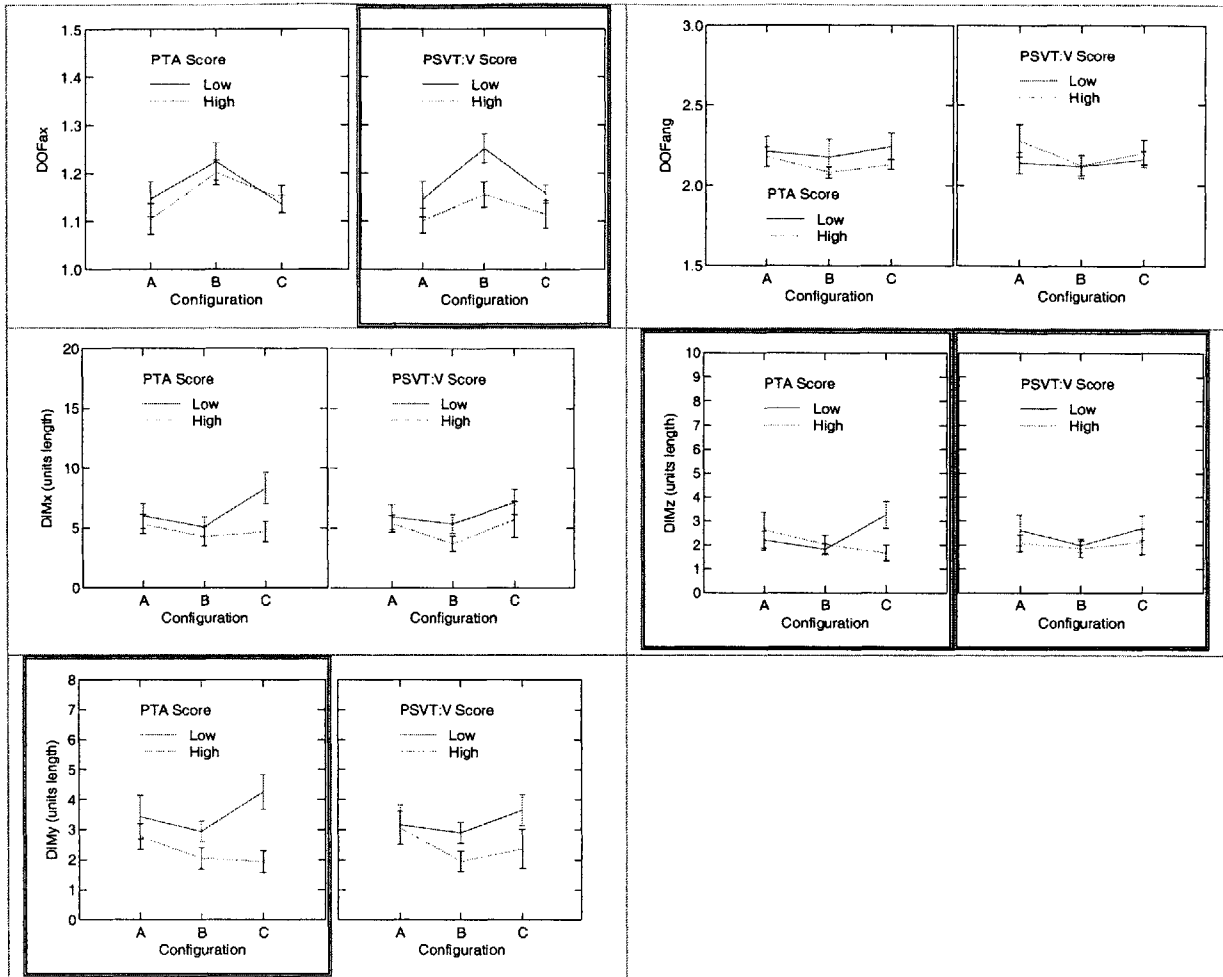


**Figure 18. Effect of PTA and PSVT:V test scores on pickup performance metrics, by configuration (PART I).**

Dotted lines correspond to high scorers and continuous lines to low scorers. The dependent variables for which a significant effect of score was found are framed in double lines. The dependent variables represented in pairs (PTA, PSVT:V) are, from left to right, and top to bottom:

a) task time (T<sub>task</sub>), b) observation time (T<sub>obs</sub>), c) % motion, d) collision time (T<sub>coll</sub>),

<sup>20</sup> The Subject Data table in O lists which subjects were in the high and low groups. The statistical analysis included the test scores as covariates in the model. Because the mixed regression model separates out the effects of other independent variables, while plots show group averages that may include effects from other factors, there may be slight inconsistencies between the plot and the full model.



**Figure 19. Effect of PTA and PSVT:V test scores on pickup performance metrics, by configuration (PART II). Dotted lines correspond to high scorers and continuous lines to low scorers. The dependent variables for which a significant effect of score was found are framed in double lines. The dependent variables represented in pairs (PTA, PSVT:V) are, from left to right, and top to bottom:**  
**a) average number of axes of simultaneous translation (DOF<sub>ax</sub>, value range 1-3), b) average number of axes of simultaneous rotation (DOF<sub>ang</sub>, value range 1-3),**  
**c, d, e) degree of inverse motion along the X, Z and Y axes (DIM<sub>x</sub>, DIM<sub>z</sub>, DIM<sub>y</sub>)**

High and low perspective-takers show consistent differences in performance: high scorers (on either test) had a higher efficiency in horizontal and vertical motion (low DIM<sub>x</sub>, DIM<sub>y</sub>, DIM<sub>z</sub>), and moved less continuously (low %motion) than low scorers. High PSVT:V scorers translated along fewer axes at any one time than lower scorers (low DOF<sub>ax</sub>), and high PTA scorers also rotated about fewer axes at a time than low scorers (low DOF<sub>ang</sub>), but not significantly.

Table 8 shows the size of these effects, and compares them to the mean value of the dependent variable for pickup trials. Some of the differences between high and low scorers are consistent but very small. In %motion high scorers were only a few units below low scorers. Similarly, high scorers used axial degrees of freedom to translate (DOFax) slightly more than low scorers did. The differences in degree of inverse motion between high and low scorers varied from one tenth to one third of the mean value, and were about twice the diameter of the payload dish (0.3 of a unit.)

**Table 8. Magnitude of the effect of the two different levels (high/low) of score (PTA or PSVT:V) and their mean, for the performance measures of pickup.**

Variable	Mean	PTA (p-value)	PSVT:V (p-value)
T_task (s)	61.34		
T_obs (s)	2.15		
T_coll (s)	0.48		
% motion	63.55	-3.22 (0.046)	-3.02
DOFax (axes)	1.16		-0.04 (0.001)
DOFang (axes)	2.17		
DIMx (units of length)	5.65	-0.70	-.031
DIMz (units of length)	2.92	-0.57 (0.005)	-0.67 (0.047)
DIMy (units of length)	2.26	-0.83 (0.001)	-0.72

***Effect of configuration***

***Configuration B – wider angular separation between cameras***

High PTA scorers had shorter collision times than low scorers in configuration B ( $p = 0.005$ ), but this difference is small, because only a few collisions were observed in pickup.

Contrary to what we expected, we did not find significant differences in task performance between high and low PSVT:V scorers in this configuration.

***Configuration C – control-display reference frame misalignment***

High scorers completed the task about 4 seconds faster than low and the corresponding cross-effects are significant (PTA,  $p = 0.037$ ; PSVT:V,  $p = 0.004$ ). Those cross-effects, however, tend to cancel the

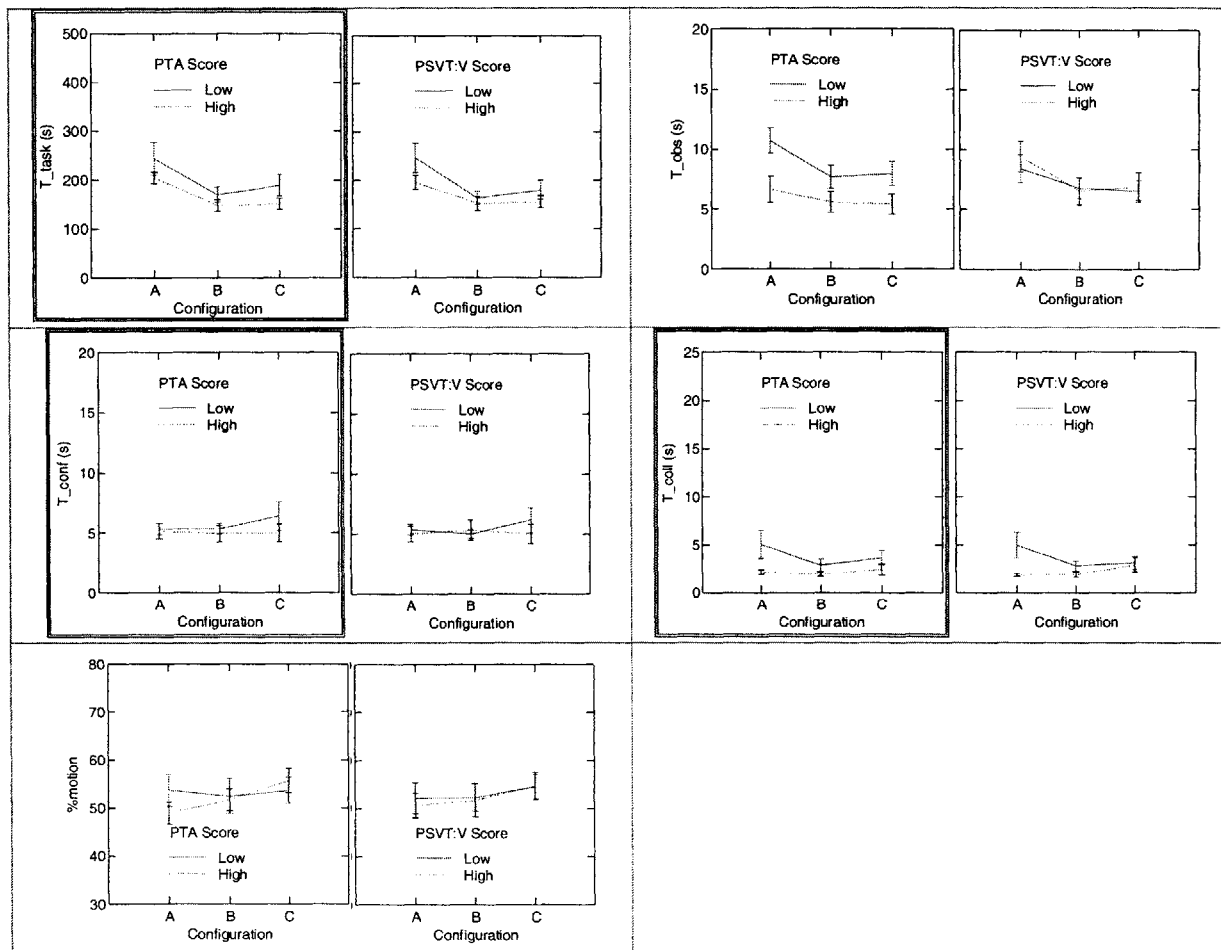


corresponding main effects. We conclude that they are small and have no major impact on our hypotheses.

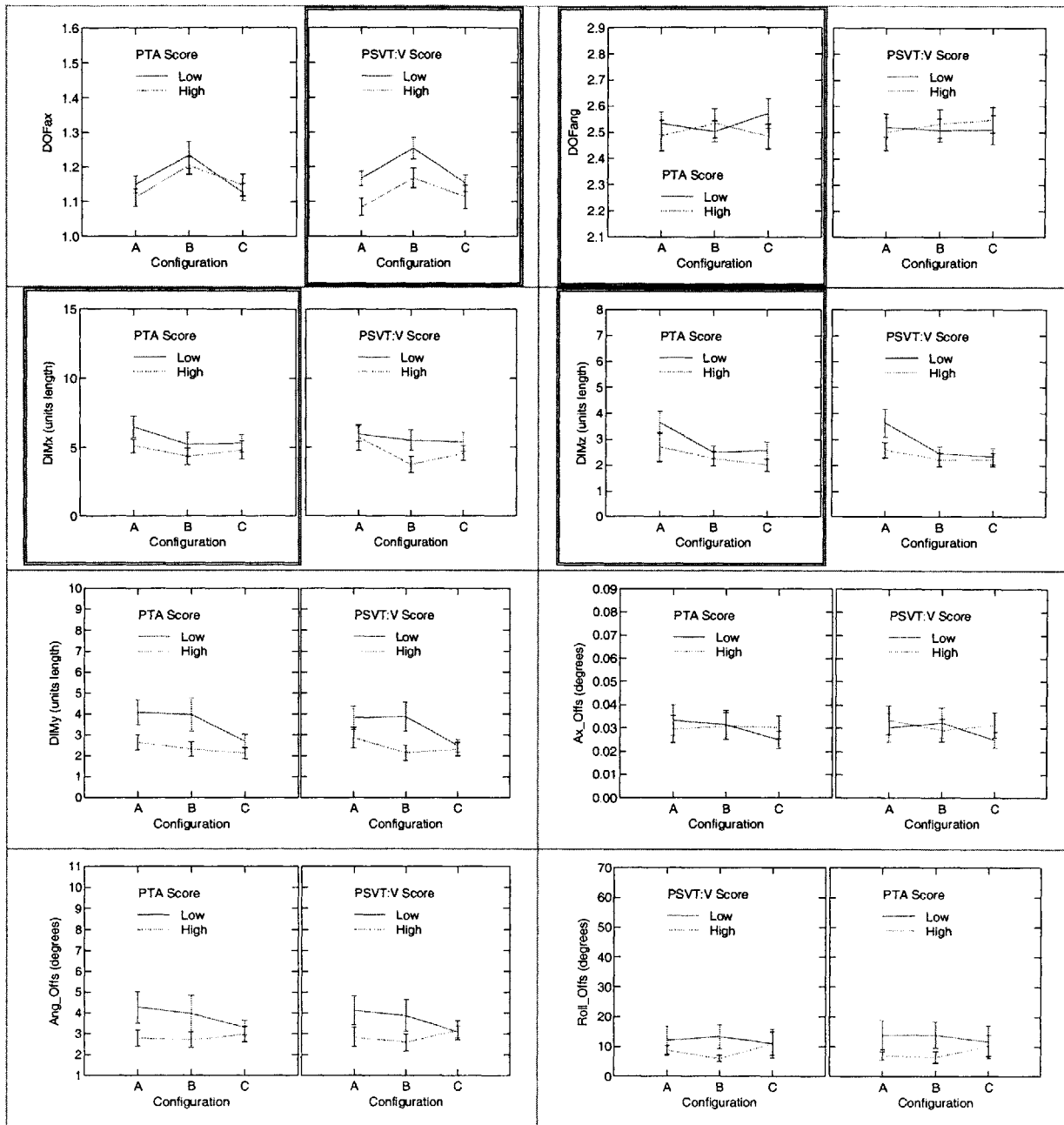
Although Figure 18c suggests that low PTA scorers had greater amounts of inverse motion along all three axes (DIMx, DIMy, DIMz) in configuration C, there was no significant cross effect of test score and configuration, for those variables.

## 5.2.2 Docking phase

Figure 21 and Figure 21 show the results on the performance metrics in docking phase, for high and low PTA scorers, by configuration.



**Figure 20. Effect of PTA and PSVT:V test scores on docking performance metrics, per configuration (PART I).** Dotted lines correspond to high scorers and continuous lines to lower scorers. In a frame are those dependent variables where a significant effect of score was found. The dependent variables represented in pairs (PTA, PSVT:V) are, from left to right, from top to bottom: a) task time ( $T_{task}$ ), b) observation time ( $T_{obs}$ ), c) confirmation time ( $T_{conf}$ ), d) collision time ( $T_{coll}$ ), e) %motion



**Figure 21. Effect of PTA and PSVT:V test scores on docking performance metrics, per configuration (PART II).** Dotted lines correspond to high scorers and continuous lines to lower scorers. In a frame are those dependent variables where a significant effect of score was found. The dependent variables are represented in pairs (PTA, PSVT:V) are, from left to right, from top to bottom:

- a) number of axes of simultaneous translation (DOFax, value range 1-3),
- b) number of axes of simultaneous rotation (DOFang, value range 1-3),
- c, d, e) degree of inverse motion along the X, Z and Y axes (DIMx, DIMz, DIMy),
- f, g, h) axial, angular and roll docking offsets (Ax\_Offs, Ang\_Offs, Roll\_Offs)

As in pickup, in docking we find a significant effect of perspective-taking test scores on performance. Results indicate that high scorers performed the task almost 30 seconds faster (low T\_task), and had about one fourth shorter collision times than low scorers. From this task time, their translation time (T\_ax) was about 3s shorter, but their angular time (T\_ang) was not different from that of low scorers. In addition, high PTA scorers required 1 second (significantly) less to observe the scene (T\_obs).

High scorers showed higher degrees of inverse motion in horizontal and vertical motion (low DIMx, DIMy, DIMz), translating less than low scorers to go from the initial point of the end effector to its last. These differences are of about the same magnitude as that on pickup. They also showed a non-significant trend toward higher angular and roll accuracy in docking (Ang\_Offs, Roll\_Offs).

To a small degree, high PTA scorers rotated about fewer axes at a time (low DOFang) than lower scorers, while high PSVT:V scorers translated along fewer axes at a time (low DOFax).

The size of these effects and their significance are shown in Table 9.

**Table 9. Magnitude of the effect of the two different levels (high/low) of score (PTA or PSVT:V) and their mean, for the performance measures of docking**

Variable	Mean	PTA	PSVT:V
T_task (s)	184.96	-29.82 (0.019)	
T_obs (s)	7.38	-1.27	
T_coll (s)	3.04	-1.04 (0.003)	
% motion	52.72		
DOFax (axes)	1.16		-0.04 (0.025)
DOFang (axes)	2.52	-0.09 (0.002)	
DIMx (units length)	5.24	-0.63 (0.023)	-0.38
DIMz (units length)	3.00	-0.52 (0.045)	-0.57
DIMy (units length)	2.62	-0.54	-0.70
T_ax (s)	51.20	-3.32 (0.000)	
T_ang (s)	37.30		
Ax_Offs (units length)	0.03		
Ang_Offs (degrees)	3.36	-0.08	-0.56
Roll_Offs (degrees)	10.68	-0.70 (0.045)	-0.56
T_conf (s)	5.36	-0.84 (0.019)	

## ***Effect of configuration***

### ***Configuration B – wider angular separation between cameras***

When the cameras had wider angular separations, high PSVT:V scorers needed significantly more time to confirm the accuracy of the docking ( $T_{\text{conf}}$ ,  $p = 0.047$ ) than lower scorers –a small effect. The results are consistent with the (non-significant) trend that high PSVT:V scorers had higher accuracy when moving along X and Y (low DIMx, DIMy) than low scorers did.

There were no significant effects of PTA score on any dependent variable for configuration B.

### ***Configuration C – control-display reference frame misalignment***

High perspective-taking scorers (both tests) required shorter (but not significantly shorter) confirmation times than low scorers, when there was a misalignment between the control and display reference frames.

## **5.2.3 Interpretation of perspective taking results**

Our hypothesis predicted that subjects with high perspective-taking test scores would perform better in teleoperation subtasks such as pickup and docking. In both tasks we expected high scorers to move more accurately and continuously, with higher multi-axial translation and fewer collisions, than low scorers. We also expected them to complete the task faster.

As predicted, high scorers (in both tests) moved more accurately than low scorers in pickup and in docking. Their degree of inverse motion –the total integrated distance-traveled away from the final target along each axis– was smaller in all three configurations. We also found that high scorers (PTA) collided significantly less and performed the task faster than low scorers in the docking phase. (The number of collisions in the pickup phase was too small to indicate any large effect of score.)

We were surprised by the results found for %motion and multi-axial translation (DOF<sub>ax</sub>). Contrary to what we expected, we found that high scorers consistently moved with less continuity than low scorers in the pickup phase, even if this difference was small. This means that they paused more often throughout the task. We also found that high PSVT:V scorers translated along fewer axes at a time than low scorers in both tasks, while high PTA scorers rotated about fewer axes at a time than low (in docking only). If it is true that better performers tend to use fewer degrees of freedom to translate and rotate, it would suggest that NASA may wish to review its performance evaluation metric of multi-axial motion. NASA encourages astronauts to move along and rotate about multiple axes, but if the present results are correct, that policy may reduce performance, at least during the initial phase of training studied in this experiment.

The results above suggest that good perspective-takers do, as our hypothesis suggested, perform better on pickup and docking tasks than poor perspective-takers, independent of the location of the cameras. Their ability to integrate the camera views more efficiently may allow them to plan their desired trajectory, and foresee potential collisions. Their criterion for selecting a trajectory of motion, however, apparently was not based on the shortest distance to travel, but on the easiest motion to perform: along one axis at a time. It is likely that this strategy made it easier to understand and predict the motion of the arm, and this enabled them to estimate adequate clearances in a timely manner, and avoid collisions.

As stated in the hypothesis, we expected the effect of high PSVT:V score to be particularly strong when the separation of the camera views was large (i.e., when the task was especially challenging in the sense of 3D perspective-taking). We did not find evidence to support this prediction.

We also predicted that subjects with higher PTA scores would perform better when there was a misalignment between the control and display reference frames. In configuration C, high PTA scorers performed the pickup task significantly faster and had non-significantly lower inverse motion than low scorers. In the absence of stronger evidence, we cannot conclude that the misalignment between the reference frames had different effects on the performance of good and poor perspective-takers, though such effects were seen in the preliminary study.

There are two possible explanations for the lack of significant cross effects between PT scores (PSVT:V and PTA) and configuration (B and C, respectively): 1) Test scores do not reliably predict an effect on the subjects' performance in pickup and docking tasks, or 2) Our experiment did not challenge the subjects' perspective-taking ability enough to show an effect of score on performance. We found significant main effects of PTA score on several dependent variables which suggests that it may be a good predictor of performance on our tasks, and that those tasks challenge the abilities measured by PTA. A more challenging experiment that includes a larger misalignment (say, of 135° rather than 90°) might help resolve this uncertainty. PTA measures PT ability only in azimuth. However, we found few significant main effects of PSVT:V. This suggests that it may not be the predictive measure of 3D perspective-taking performance we had hoped for.

## 5.3. Effect of spatial visualizations

### 5.3.1 Pickup phase

The CC test provided an estimate of a subject's SV ability. Figure 22 shows the variation in selected pickup performance metrics as between camera configurations. The performance of high and low CC scorers is shown separately.

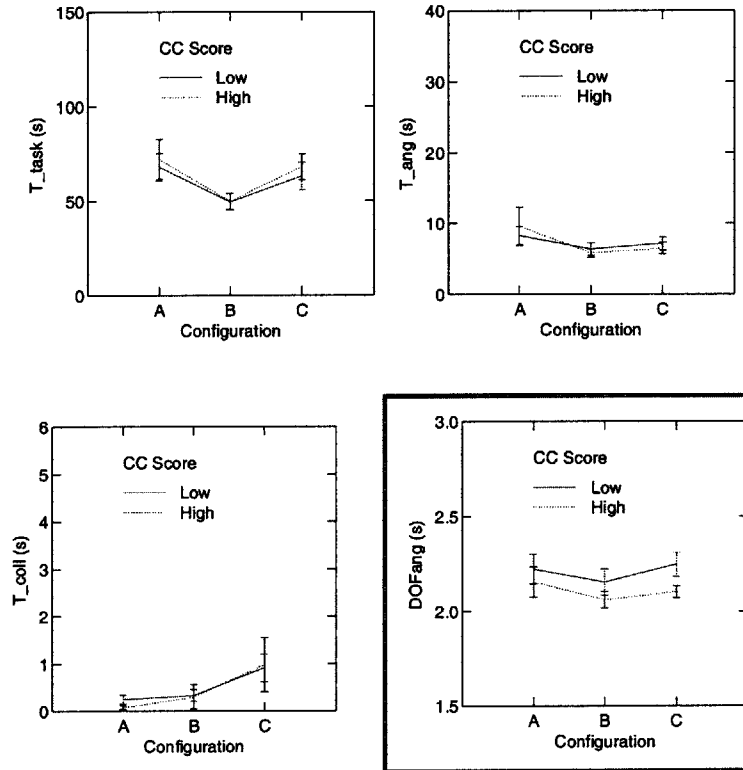
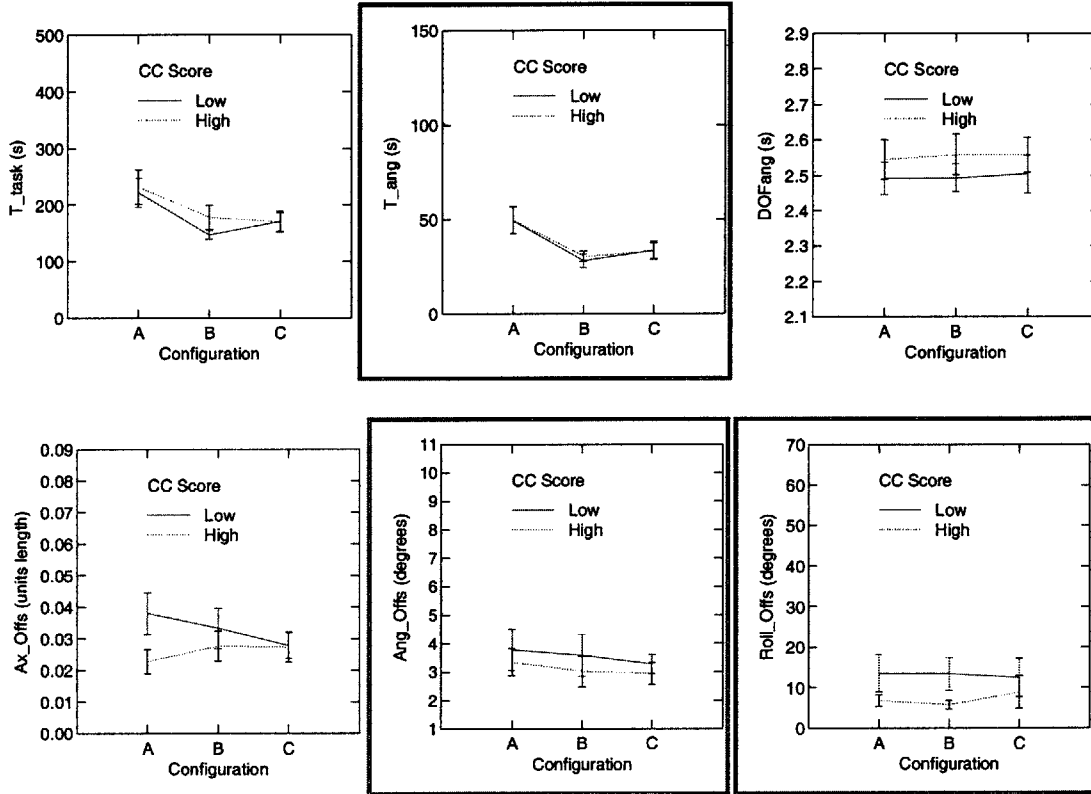


Figure 22. Effect of CC test score on pickup performance metrics, by configuration. Dotted lines correspond to high scorers and continuous lines to lower scorers. The dependent variables for which a significant effect of score was found are framed in double lines. The dependent variables represented are, from left to right, from top to bottom: a) task time ( $T_{task}$ ), b) angular time ( $T_{ang}$ ), c) collision time ( $T_{coll}$ ), d) number of axes of simultaneous rotation ( $DOF_{ang}$ , value range 1-3)

The only significant effect of CC score on any pickup performance metric was on  $DOF_{ang}$ , and its magnitude was small: High CC scorers rotated simultaneously about fewer axes than low scorers ( $DOF_{ang}$ , mean = 2.17, effect (between high and low scorers) = 0.06,  $p = 0.016$ ).

### 5.3.2 Docking phase

Figure 21 shows the variation in selected docking performance metrics as between camera configurations. The performance of high and low CC scorers is shown separately.



**Figure 23. Effect of CC test score on docking performance metrics, by configuration. Dotted lines correspond to high scorers and continuous lines to lower scorers. The dependent variables for which a significant effect of score was found are framed in double lines. The dependent variables represented are, from left to right, from top to bottom: a) task time ( $T_{task}$ ), b) angular time ( $T_{ang}$ ), c) number of axes of simultaneous rotation ( $DOF_{ang}$ , value range 1-3), d, e, f) axial, angular and roll docking offsets ( $Ax\_Offs$ ,  $Ang\_Offs$ ,  $Roll\_Offs$ )**

High CC scorers had a significantly higher docking accuracy than low scorers. The differences between their angular and roll offsets ( $Ang\_Offs$ ,  $Roll\_Offs$ ) were of about one fifth of the mean values. There is a parallel but non-significant effect of CC score on axial docking accuracy ( $Ax\_Offs$ ).

We found a small but significant effect of CC score on angular time ( $T_{ang}$ )—high scorers used about 8% more time to rotate than low.

**Table 10. Magnitude of the effect of the two different levels (high/low) of CC score and their mean, for the performance measures of docking**

Variable	Mean	CC
T_task (s)	184.96	14.81
DOFang (axes)	2.52	0.01
T_ang (s)	37.30	+3.44 (0.016)
Ax_Offs (units length)	0.03	-0.005
Ang_Offs (degrees)	3.36	-0.66 (0.002)
Roll_Offs (degrees)	10.68	-2.46 (<0.0005)

### 5.3.3 Interpretation of CC test results

We expected CC test score to have a significant effect on the measurements of rotation performance of the end effector. We hypothesized that subjects with higher CC scores would rotate the end effector about more axes simultaneously (DOFang, pickup and docking), that they would have a higher orientation docking accuracy (i.e., angular and roll offsets), and that they would spend less time in rotation, throughout the docking task. Our results do not support that hypothesis completely.

As predicted, subjects with high CC scores had significantly higher accuracy in docking. Their angular and roll offsets were smaller than those of low scorers.

Contrary to our hypothesis, high CC scorers rotated about fewer axes at any time (DOFang) during the pickup trials. This tendency of high CC scorers to be slightly more conservative in their motion resembles that found for PTA and PSVT:V scores. It suggests that a better understanding of the controls does not necessarily lead to multi-axial motion, but to more conservative movements. This strategy probably allows better scorers to do the task more efficiently, and with a lower mental workload.

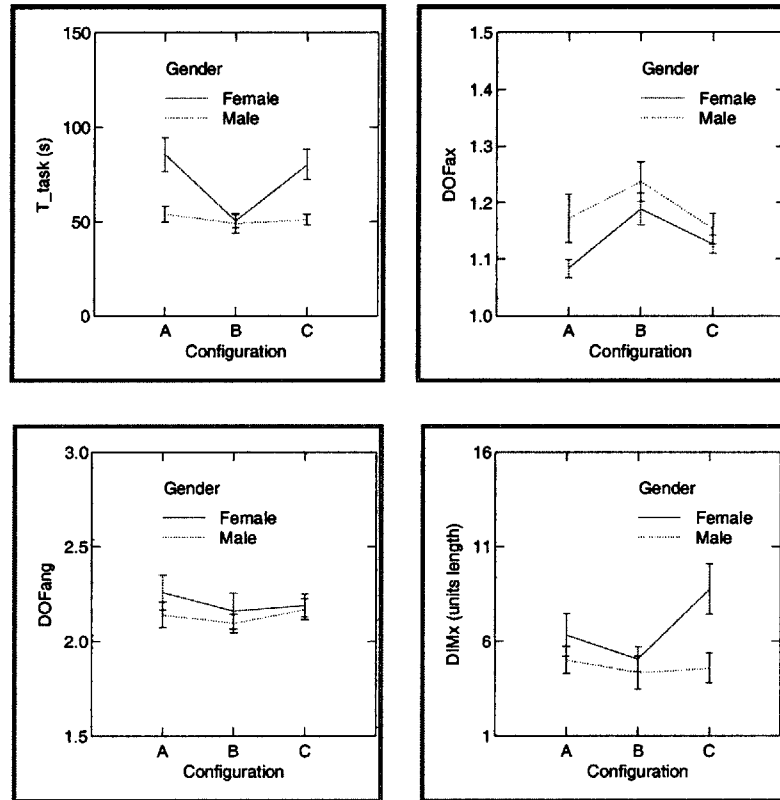
That failure of CC score to predict performance suggests that skill at spatial visualizations and mental rotation may not be a good predictor of teleoperation performance. Even if operators were good visualizers or mental rotators, they would still need the perspective-taking ability that would allow them to integrate the camera views efficiently. The ability may be necessary, but it may not be sufficient to support an outstanding operator.



## 5.4. Effect of gender differences

### 5.4.1 Pickup Phase

Figure 24 shows (only) the significant effects of gender on performance in pickup trials, by configuration.



**Figure 24. Significant gender differences on pickup performance metrics, by configuration. Dotted lines are for male, and continuous lines for female subjects. The dependent variables represented are, from left to right: a) task time ( $T_{task}$ ), b, c) number of axes for simultaneous translation and rotation (DOFax, DOFang), d) degree of inverse motion along the X axis (DIMx)**

Women required significantly longer times (almost 30%) than men to complete the pickup trials ( $T_{task}$ ). They translated along fewer axes at any one time (DOFax), but rotated about more axes at any given time (DOFang). As shown in Table 11, both differences are small.

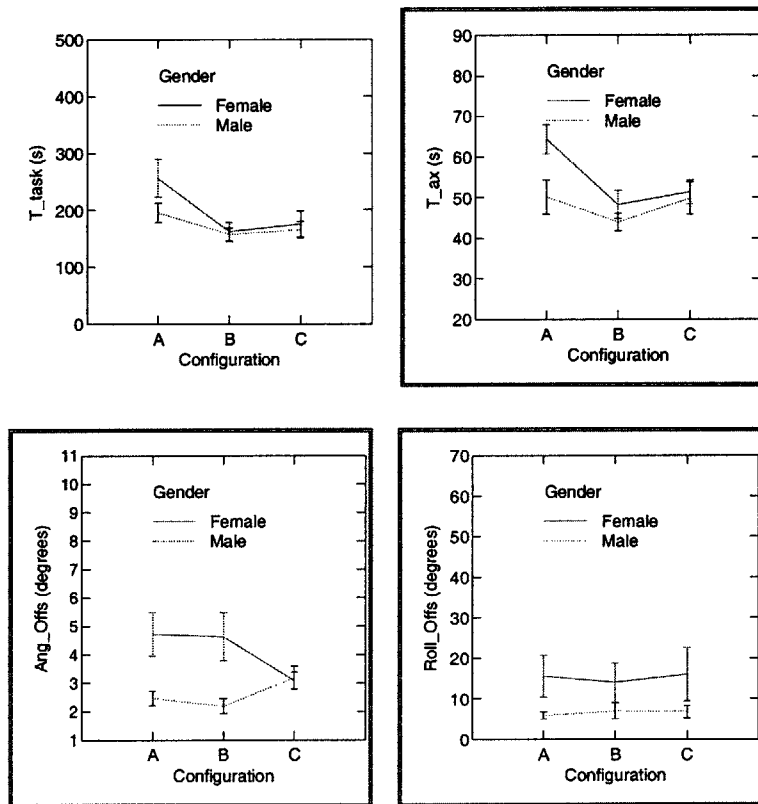
Women moved less efficiently along the X axis, as measured by X inverse motion, DIMx. Although this difference is not significant, it is greater than the any of the significant differences between high and low tests scorers.

**Table 11. Magnitude of the effect of gender and their mean, for the performance measures of pickup**

Variable	Mean	Gender
T_task (s)	61.34	17.12 (0.001)
DOFax (axes)	1.16	0.03 (0.052)
DOFang (axes)	2.17	-0.11 (0.012)
DIMx (units length)	5.65	1.00 (0.050)

### 5.4.2 Docking phase

Figure 25 shows the effects of gender on some of the performance measures in docking trials, by configuration.



**Figure 25. Gender differences on docking performance metrics, by configuration. The dependent variables for which a significant effect of gender was found are framed in double lines. Dotted lines are for male, and continuous lines for female subjects. The dependent variables represented are, from left to right: a) task time (T\_task), b) axial time (T\_ax), c, d) angular and roll docking offsets (Ang\_Offs, Roll\_Offs)**

As in pickup, in docking women needed about 11% more time than male subjects to move axially (T\_ax). They also had lower orientation docking accuracy (Ang\_Offs; Roll\_Offs), as shown in Table 12.

**Table 12. Magnitude of the significant effect of gender and their mean, for the performance measures of pickup**

Variable	Mean	Gender
T_ax (s)	51.20	6.03 (0.050)
Ang_Offs (degrees)	3.36	1.25 (0.001)
Roll_Offs (degrees)	10.68	3.20 (0.008)

### 5.4.3 Interpretation of gender effects

According to our hypothesis, we expected to find an effect of gender on performance in pickup and in docking. There was, indeed, a significant effect of gender in some performance measures. Except for T\_task (pickup) and T\_ax (docking), which can be considered as similar variables, no other dependent variable showed significant effects of gender on both pickup and docking.

The performance of female subjects is consistent with their lower scores on the spatial tests and with several studies (see section 2.2.2) in which women have shown poorer spatial performance – they were slower in translation (T\_ax), and had lower accuracy in docking (Ang\_Offs, Roll\_Offs). There is one minor exception to this trend. The number of axes along which female subjects translated (DOFax) in pickup was significantly higher than that of male subjects, when we would have expected it to be lower (as it is for other low scorers). The size of the effect, however, is small, and given the many dependent variables we have studied, we would expect to encounter an occasional false positive.

## 5.5. Other effects

### 5.5.1 Age

We saw few significant effects of age on the measures of performance. In pickup, younger subjects had significantly higher inverse motion along the Z axis (DIMz,  $p=0.011$ ). In docking, they needed less observation time (T\_obs,  $p = 0.003$ ), and had lower roll offset (Roll\_Offs,  $p = 0.036$ ). These observations, apart from the hypotheses for which the experiment was designed, suggest that younger subjects may have traded accuracy and attention for speed. An experiment designed to better control speed/accuracy tradeoffs and to test for effects of age could contribute to further study of these differences.

### **5.5.2 Understanding of camera location (SetCam)**

As mentioned in the first part of this chapter, we found a clear division between those subjects who were able to describe the location of the cameras (SetCam = 1) and those who were not (SetCam = 0). We did not find a significant correlation between subjects' values of SetCam and their spatial ability scores.

We did find some significant effects of SetCam on the performance metrics. In pickup, subjects (SetCam=1) who were able to describe the positions of the cameras showed higher inverse motion along the X axis (DIMx,  $p = 0.001$ ), compared to those who could not. In docking, they required significantly longer times (Tconf,  $p = 0.043$ ) to confirm the accuracy of their docking.

These effects are small, sparse, and inconsistent across phases. It is therefore difficult to draw any conclusions about how this knowledge relates to task performance. The metric may reflect whether a subject was able to integrate the different sources of information into their representation of the environment, but whether this has any impact on task performance is unknown.

### **5.5.3 Past videogame experience**

We found significant effects of past videogame experience on the performance measures of the docking phase. Those subjects who reported that they had habitually played videogames at some time needed significantly less time (4.5 seconds) to rotate to the desired position (T\_ang,  $p = 0.001$ ) than those who had never that habit. They also rotated about fewer axes at a time (DOFang,  $p = 0.042$ ).

These results indicate that subjects with previous videogame experience might have felt more comfortable in rotation than other subjects. It could also be that they had more experience with joysticks and were, therefore, more fluent in the use of the rotational hand controller. Again, subjects with more experience tend to adopt a control strategy of using fewer degrees of freedom for rotation.

# Chapter 6. Discussion

## 6.1. Strategies of motion

After each configuration, we asked our subjects about their strategies to translate and rotate the robot arm. All reported understanding translation during the first set of practice trials. Three did not immediately understand that the motion of the arm was under end-effector control, and kept trying to operate the arm by moving each of the joints with the translational hand controller. We clarified that issue before the testing phase started. The translation of the arm in configurations A and B was generally described as “easy” and “intuitive”. In configuration C, some subjects noticed –but could not explain– the misalignment between the display and reference frames: They reported that “the arm was doing something weird”. Some resorted to trial-and-error strategies to move right/left and front/back in response to their not being able to understand the translation motion, but did not consider it a challenge.

In contrast, subjects found rotation control very difficult to understand. The majority (as opposed to “some” in translation) used trial-and-error strategies to operate the rotational hand controller. They explained that in order to rotate the cargo module into the desired position they had to activate the hand controller randomly until they achieved orientation by accident. There are two explanations for this confusion. First, the model of joystick we used was hard to rotate about only one axis at a time – subjects were likely, for example, when trying to roll the stick, to yaw it involuntarily at the same time. Second, the end effector was rotated about a moving reference frame (see section 4.3. ), which was not clearly explained to the subjects (see APPENDIX J). Consequently, the number of axes about which subjects rotated at the same time (DOFang) was consistent and very high throughout the experiment (close to three, its highest possible value).

### *Inverse motion*

The degree of inverse motion was considerably higher in the X direction, DIM<sub>x</sub>, than in the Y or Z directions, DIM<sub>y</sub> or DIM<sub>z</sub>. Since the paths were designed and balanced to have similar motion along all three axes on each phase, the higher value of DIM<sub>x</sub> suggests that we measured something beyond inverse motion. Subjects may have felt more comfortable in moving, exploring, in the xy plane (*display frame of reference*), and more readily in the horizontal (X) than in the vertical (Y) direction. This would explain why, in configuration C (where the X and Z axes are visually exchanged), the inverse motion along Z decreased significantly while along X it increased. Although DIM<sub>x</sub> likely measured something beyond inverse motion, we find that these variables (DIM<sub>x</sub>, DIM<sub>y</sub>, DIM<sub>z</sub>) do measure the efficiency of the subject’s motion.

## 6.2. Accuracy

We used three measures of docking accuracy: the radial distance between the centers of the docking ports (Ax\_Offs), the angular distance between their axes (Ang\_Offs), and their roll misalignment (Roll\_Offs) –the angular distance between the stripes on the docking ports. Although we expected these quantities to measure the subject’s ability to integrate the different camera views to position the cargo module, we found that most of these offsets were very small for most subjects –the mean axial offset was only about 10% of the diameter of the docking port, and the mean angular offset was only about 3 degrees. These errors might even have been at the very limit of what subjects could perceive as an offset. This was likely due to our insistence during the training that subjects dock as accurately as possible. Consequently, subjects spent much of their time ensuring a “perfect” docking. On a scale from 1 to 5 (“not important” to “very important”), most of our subjects treated each of the aspects of docking as “very important”. As small as they were, for all subjects, there were significant differences among subjects and they correlated with CC scores. In the future, by limiting the time a subject may spend on a task, we may be able to measure the trade-off between time and docking accuracy; Defining a “accurate docking” range would be a good way to keep subjects from investing excessive time on a perfect docking when a near-perfect docking would be fully acceptable.

## 6.3. Use of multiple displays

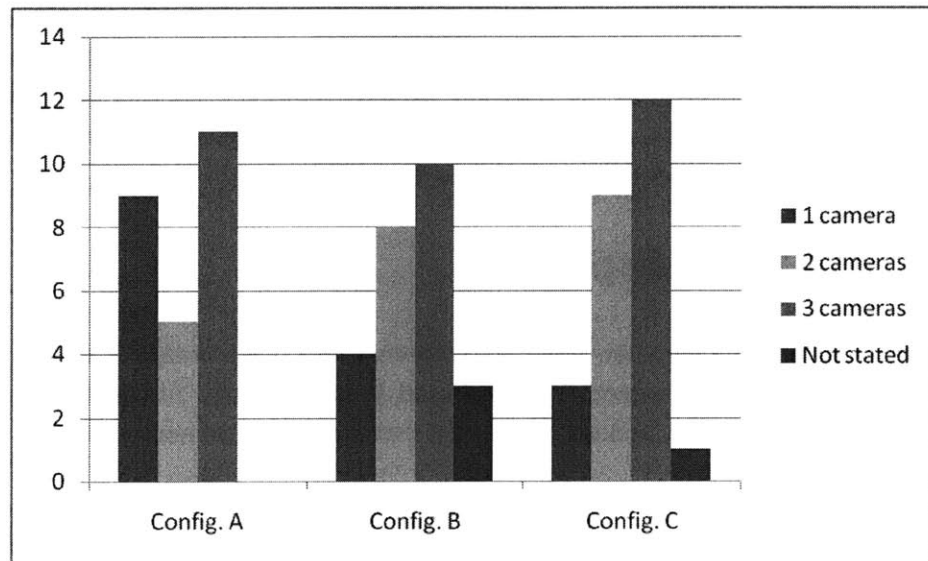
After completing each configuration, we asked our subjects if they had a preferred camera view (during that configuration) and how many cameras they had used, simultaneously. Although their answers may not have been consistent with their actions, there were some trends worth noting:

**Table 13. The combinations of camera(s) subjects preferred for each configuration. Cameras 1, 2 and 3 correspond to the left, center and right screens, respectively. Subjects preferred the combination of cameras 1 and 3 (underlined).**

Camera(s)	Config A	Config B	Config C
1 only	2	2	3
2 only			1
3 only	3	1	1
1 & 2	1	2	
<u>1 &amp; 3</u>	<u>9</u>	<u>5</u>	<u>7</u>
2 & 3		1	1
1, 2 and 3	2	5	7
Not stated	8	9	5
<i>Total</i>	25	25	25

As Table 13 shows, the most commonly used combination of cameras, over all configurations, was 1 and 3 together. Almost no subjects used camera 2, either individually or in combination with other cameras.

This may be because camera 2's position was (Figure 8) at the bottom, beneath the XY control plane, and because understanding a scene may be harder when it was viewed from below. In contrast, as shown in Fig. 8, cameras 1 and 3 had approximately the same elevation. One explanation for the preference for cameras 1 and 3 may be found in recent reports [32] of tests on imagined rotations. They are faster to perform about the body's principal axis than about the roll axis. The rotation that would carry camera 1 to camera 2 (which is below it) is a roll; the rotation that would carry it to camera 3 (which is approximately at the same altitude) is a yaw--about the body's principal axis. Therefore, transitions between the use of cameras 1 and 3 are, from that result, more convenient than those between cameras 1 and 2 or cameras 3 and 2 (which is also a roll). This may contribute to the evident preference observed in our subjects.



**Figure 26. Number of cameras used simultaneously for each configuration, as reported by the 25 subjects after completing each configuration.**

As shown in Figure 26, the number of subjects who reported using only one camera at any one time decreased considerably from configuration A to B and C, whereas the number using two cameras simultaneously increased, correspondingly. We can interpret this result in two ways: 1) Subjects learned how to use the cameras efficiently only after practicing on configuration A (which was presented first), or 2) Subjects used more cameras simultaneously to navigate the more difficult configurations.

# Chapter 7. Conclusions

Preliminary findings suggest that spatial ability strongly influences performance on some of the significant subtasks of space teleoperation. The relevant components of spatial ability are perspective-taking and spatial visualization. To confirm preliminary findings, we measured the performance of 25 naïve subjects who used translation and rotation hand controllers and 3 environmentally fixed camera views to control a 2 boom, 6 degree-of-freedom simulated arm while performing pickup and docking subtasks. To challenge the subjects' spatial ability we tested subjects on narrow- and wide- separations between camera views. We also introduced misalignments between the translation control and the display reference frames to explore the robustness of their performance.

We used the Perspective-Taking Ability test (PTA) and the Purdue Spatial Visualizations Test: Visualization of Views (PSVT:V) to measure perspective-taking ability, and the Cube Comparisons test (CC) to assess spatial visualization ability.

Our study found that scores on the PTA, Perspective-Taking Ability test, predict the quality of performance in subtasks of teleoperation. The PSVT:V scores are less reliable predictors.

High perspective-taking scorers performed the pickup task with significantly higher efficiency (lower inverse motion) than low scorers, but not necessarily faster. In the docking task, however, they performed significantly faster ( $T_{\text{Task}}$  was about 30s less out of 180s,  $p=0.019$ ) and more accurately (horizontal inverse motion in  $x$ ,  $z$ : DIM $x$ , DIM $z$ ), respectively, they collided with obstacles less (i.e., spent fewer seconds in collisions), 1 second less out of 3 seconds of average collision time ( $p=0.003$ ), and had higher docking accuracy (Roll\_Offset, 0.7 degrees less out of 10.7 degrees,  $p=0.045$ ).

In both tasks high perspective-taking (PSVT:V) scorers moved along only one axis at a time. This is an interesting result since during robotics training, NASA encourages astronauts to move along several axes at once, to shorten their time to target. Our best-performing subjects, however, moved along single axes, successively, for both pickup and docking tasks.

The experimental results do *not* allow us to conclude that subjects with higher PTA scores will perform better when the misalignment between the control and display reference frames is as large as 90 degrees. (Testing at larger misalignment angles may be necessary to demonstrate this effect).

In the docking task, high spatial visualization (CC) scorers showed significantly higher accuracy (Roll\_Offset=-2.46 out of 10.7 degrees,  $z=3.49$ ,  $p<0.0005$ ; AngOffset=-0.66 out of 3.3 degrees total,  $z=3.09$ ,  $p=0.002$ )—as we expected— but rotated about fewer axes at any one time, on the average—which we did not expect. Presumably, the reasons for this and for the corresponding result on translation cited above are the same.



The study found effects of gender on performance. On average, females required longer times to complete the tasks, and had lower accuracy in the attachment, an effect partially explained by spatial ability scores. There were no significant effects beyond those explained by differences in spatial ability as measured by PTA and CC.

We conclude that PTA scores are useful for predicting the efficiency with which a trainee can, after two hours of training in virtual reality, and avoid collisions of the robot with obstacles. Our findings suggest that strategies used to solve the PTA test are similar to those used to understand the visual feedback provided by fixed displays, in teleoperation tasks.

These findings can also be considered when assigning astronauts to R1 and R2 roles on a teleoperations team: The abilities measured correspond to performances that weigh differently on operators R1 and R2. PTA score can, therefore, help us prepare personnel to make best use of their tested abilities, through customized, targeted training. The results of this study can help predict which areas of teleoperation will benefit most from the investment of training time with a particular operator.

Although this study focused on the performance of subjects during their first two hours of teleoperation experience, it defines a path for future research and relevant measures of teleoperation performance.

## Chapter 8. Suggestions for future work

This experiment, along with the preliminary study, is a first attempt to understand the underlying effects of individual spatial ability on space teleoperation performance. Many questions, however, remain unsolved, and future experiments should provide a broader understanding of these effects.

– To answer the question of whether perspective-taking ability has a greater effect on performance for large misalignments between the control and display reference frame, misalignments greater than 90° should be studied.

– This experiment did not address the effect of spatial ability on operator selection of optimal sets of camera views. This ability is heavily weighted in astronaut teleoperation performance evaluations, and should be looked at in more detail. The feature of selecting camera views is already implemented in our virtual reality simulation.

– Our study was limited to fixed camera views; however, a camera mounted on the end effector is constantly used by astronauts. The use of this camera requires to integrate both fix and moving images, which might be a different challenge to perspective-taking ability of the subjects, and might even need greater mental rotation ability – a question that should be answered. Combining the use of this camera with fixed views would allow to define the effects of spatial ability on the performance of tasks that are closer to real teleoperations.

– Although we measured the number of collisions, we did not specifically assess the subjects' ability to foresee a proper clearance/collision. This aspect of performance is also heavily weighted by NASA trainers. It would be interesting to design an experiment to test the correlation between spatial ability and the subjects' ability to use different camera views to estimate the clearance of an object and an obstacle.

The following modifications to our procedure could also provide valuable improvements:

- Define end effector rotation with respect to the “external” reference frame: In this and the preliminary experiment, rotation was referenced to a coordinate frame mounted on the end effector, a movable frame. Actual RMS operations in “external” mode are referenced to the external frame, not the end effector frame.
- Limit the task time: Since our experiment was not timed, the effect of fatigue was stronger on those who required longer times to finish, probably affecting their performance. Limiting task time would reduce fatigue.

We also found that some subjects traded time for docking accuracy. This made it more difficult to define whether a higher performer was who had shorter times, or greater accuracy. In this type of trials time would no longer be a measure of performance.

- Define an “acceptable docking” range: Another way of controlling the speed/accuracy tradeoff would be to define an “acceptable docking” range, and to make the docking automatic (as pickup). This range would allow subjects to finish a trial once they have pre-docked accurately enough, and force everyone to aim for the same level of accuracy. In this type of trial docking accuracy would no longer be a measure of performance.
- Provide real-time scores: Subjects would make a greater effort to perform better if they were given real-time feedback on their performance. This would allow them to define which aspects of performance to trade, based on known evaluating criteria.
- Improve the rotational hand controller mechanical characteristics: Subjects could accidentally rotate about more axes than those intended, resulting in cross-coupled control. Stiffer springs might contribute to eliminate this cross-coupling, by requiring greater forces to activate the controller on each direction. A rotational hand controller with stiffer springs might allow subjects to have a better understanding (feeling) of rotation.
- Assessment of the understanding of camera location before (or during) each test: We found that some of our subjects did not understand (nor were interested in) the location of the cameras with respect to the arm. Although this may have been because their strategies of motion were not dependent on camera location, we know that in real life astronauts *do* know where each camera is, even if they did not use that information to mentally integrate the views. It would be useful then, to ensure that all subjects are aware of the location of the cameras, by asking them (before the testing trials) with the images in front of them, to describe the relative position of each view. After their description (which could be graded on a checklist), subjects should be informed of the real location of the cameras, to ensure that they all have the same basic understanding of the environment before the testing session.
- More practice trials: Our experiment allowed subjects relatively little time to practice before being tested. This time was enough for subjects to understand the essence of the task, but not to gain experience on it and experiment with different strategies. Longer training periods with shorter testing trials (or multiple testing sessions) would ensure subjects’ performance reached a plateau on their learning curve.

# References

1. Menchaca-Brandan, M.A., et al., *Influence of Perspective-Taking and Mental Rotation Abilities in Space Teleoperation*, in *2007 ACM/IEEE International Conference on Human-Robot Interaction*. 2007: Washington, DC.
2. Lamb, P. and D. Owen. *Human Performance in Space Telerobotic Manipulation*. in *ACM Symposium on Virtual Reality Software and Technology*. 2005. Monterey, CA: ACM.
3. Currie, N.J. and B. Peacock. *International Space Station Robotic Systems Operations - A Human Factors Perspective*. in *Human Factors and Ergonomics Society Annual Meeting*. 2001. Baltimore, MD: HFES.
4. Visinsky, M. and A. Spinler. *ISS Operations For The Special Purpose Dexterous Manipulator (SPDM): Experiences From The Robotic Systems Evaluation Laboratory (RSEL)*. in *AIAA Annual Technical Symposium*. 1999.
5. Kozhevnikov, M. and M. Hegarty, *A dissociation between object manipulation, spatial ability and spatial orientation ability*. *Memory and Cognition*, 2001. **29**(5): p. 745-756.
6. Hegarty, M. and D. Waller, *A Dissociation between Mental Rotation and Perspective-Taking Spatial Abilities*. 2003.
7. Carroll, J., *Human Cognitive Abilities: A survey of factor analytical studies*. 1993, New York: Cambridge University Press.
8. Pellegrino, J.W., et al., *A computer-based test battery for the assessment of static and dynamic spatial reasoning abilities*. *Behavior Research Methods, Instrumentation, & Computers*, 1987. **19**(2): p. 231-236.
9. Contreras, M.J., et al., *Is Static Spatial Performance Distinguishable From Dynamic Spatial Performance? A Latent-Variable Analysis*. *The Journal of General Psychology*, 2003. **130**(3): p. 277-288.
10. Ekstrom, R.B., et al., *Manual for Kit of Factor-Referenced Cognitive Tests*. 1976, Educational Testing Service.: Princeton, NJ.
11. Vandenberg, S.G. and A.R. Kuse, *Mental rotations, a group test of three-dimensional spatial visualization*. *Perceptual and Motor Skills*, 1978. **47**: p. 599-604.
12. Guay, R.B., *Purdue Spatial Visualization Tests*. 1977, Purdue Research Foundation: West Lafayette, IN.

13. Kozhevnikov, M., et al., *Perspective-Taking vs. Mental Rotation Transformations and How They Predict Spatial Navigation Performance*. Appl. Cognit. Psychol., 2006. **20**: p. 397-417.
14. Michael, W.B., et al., *The description of spatial-visualization abilities*. Education and Psychological Measurement, 1957. **17**: p. 185-199.
15. McGee, M.G., *Human spatial abilities: psychometric studies and environmental, genetic, hormonal, and neurological influences*. Psychological Bulletin, 1979. **86**: p. 889-918.
16. Just, M.A. and P.A. Carpenter, *Cognitive coordinate systems: accounts of mental rotation and individual differences in spatial ability*. Psychological Review, 1985. **92**(2): p. 137-172.
17. Lohman, D.F., *Spatial abilities as traits, processes, and knowledge.*, in *Advances in the psychology of human intelligence*, J. Sternberg, Editor. 1988, Earlbaum: Hillsdale, NY. p. 181-248.
18. Hegarty, M. and D.A. Waller, *Individual Differences in Spatial Abilities*, in *The Cambridge Handbook of Visuospatial Thinking*, P. Shah and A. Miyake, Editors. 2005, Cambridge University Press: New York. p. 121-169.
19. Halpern, D.F. and M.L. Collaer, *Sex differences in visuospatial abilities: More than meets the eye*. The Cambridge Handbook of Visuospatial Thinking. Cambridge University Press, Cambridge, 2005.
20. Silverman, I. and M. Eals, *Sex difference in spatial abilities*. The Adapted Mind: Evolutionary Psychology and the Generation of Culture, 1992.
21. Linn, M.C. and A.C. Petersen, *Emergence and Characterization of Sex Differences in Spatial Ability: A Meta-Analysis*. Child Development, 1985. **56**(6): p. 1479-1498.
22. Masters, M.S. and B. Sanders, *Is the gender difference in mental rotation disappearing?* Behavior Genetics, 1993. **23**(4): p. 337-341.
23. Brown, L.N., C.J. Lahar, and J.L. Mosley, *Age and Gender-Related Differences in Strategy Use for Route Information: A "Map-Present" Direction-Giving Paradigm*. Environment and Behavior, 1998. **30**(2): p. 123.
24. Ecuyer-Dab, I. and M. Robert, *Have sex differences in spatial ability evolved from male competition for mating and female concern for survival?* Cognition, 2004. **91**(3): p. 221-57.
25. DeJong, B.P., J.E. Colgate, and M.A. Preskin. *Improving Teleoperation: Reducing Mental Rotations and Translations*. in *IEEE Intl. Conf. on Robotics and Automation*. 2004. New Orleans, LA.
26. Spain, E.H. and K.-P. Holzhausen. *Stereo Versus Orthogonal View Displays for Performance of a Remote Manipulator Task*. in *Stereoscopic Displays and Applications II*. 1991: SPIE.

27. Akagi, T.M., et al. *Toward the construction of an efficient set of robot arm operator performance metrics*. in *48th Annual Meeting Human Factors and Ergonomics Society*. 2004: Human Factors and Ergonomics Society.
28. Eyal, R. and F. Tendick, *Spatial Ability and Learning the Use of an Angled Laparoscope in a Virtual Environment*, in *Medicine Meets Virtual Reality 2001*, J.D. Westwood and e. al., Editors. 2001, IOS Press: Amsterdam. p. 146-152.
29. Tracey, M.R. and C.E. Lathan, *The Interaction of Spatial Ability and Motor Learning in the Transfer of Training From a Simulator to a Real Task*, in *Medicine Meets Virtual Reality 2001*, J.D. Westwood and e. al., Editors. 2001, IOS Press: Amsterdam. p. 521-527.
30. Lathan, C.E. and M. Tracey, *The Effects of Operator Spatial Perception and Sensory Feedback on Human-Robot Teleoperation Performance*. Presence: Teleoperators and virtual environments, 2002. **11**(4): p. 368-377.
31. SpaceRef.com (2007) *Space Station User's Guide - ISS Elements: Mobile Servicing System. Volume,*
32. Creem, S.H., M. Wraga, and D.R. Proffitt, *Imagining physically impossible self-rotations: geometry is more important than gravity*. Cognition, 2001. **81**: p. 41-64.

# Appendices

## APPENDIX A Subject Data

CODE	Group	F0/M1 Gender	CC	H/L CC	PTA	H/L PTA	PSVT:V	H/L PSVT:V
1	1	0	19	1	18	1	1.25	1
2	1	0	15	1	25	1	11.75	1
3	1	0	32	1	15.6	1	11.5	1
4	1	0	26	1	20	1	19	1
5	1	0	23	1	22.4	1	11.75	1
6	1	0	16	1	8.5	1	5.5	1
8	2	0	24	1	14.7	1	10	1
9	2	0	9	1	19	1	6.5	1
10	2	0	31	1	21	1	11.25	1
11	2	0	26	1	24	1	16.75	1
12	2	0	21	1	25	1	18.75	1
13	2	0	18	1	21	1	13	1
15	1	1	17	1	27	1	13.5	1
16	1	1	13	1	21	1	15.75	1
17	1	1	9	1	25	1	15	1
18	1	1	32	1	29	1	20.75	1
19	1	1	27	1	20	1	13.75	1
20	1	1	21	1	26	1	13	1
21	1	1	12	1	21	1	6.75	1
23	2	1	14	1	24	1	21	1
24	2	1	7	1	16	1	13.5	1
25	2	1	30	1	26	1	20	1
26	2	1	27	1	25	1	19	1
27	2	1	21	1	24	1	10.25	1
28	2	1	14	1	18	1	16.5	1
29	2	1	38	1	34	1	25	1
		<b>Median</b>	21		21.7		13.5	
		<b>SD</b>	8.2		5.2		5.4	
		<b>Mean</b>	20.2		21.9		13.9	

H/L : High / Low scorer group

## APPENDIX B Pre-Test Questionnaire

Gender: F M

Age: \_\_\_\_\_

Right/Left handed: R L

Course #: \_\_\_\_\_

Colorblind? Y N

**1. Do you have any experience with Virtual 3-D environments (e.g. 3-D games, CAD, 3-D graphic design, etc.)?**

(Yes No) (If "Yes," can you please describe these experiences?)

**2. Do you have any experience with joysticks or game controllers? (e.g. computer games, TV games, robotic manipulation)**

(Yes No) (If "Yes," can you please describe these experiences?)

**3. How many hours per day do you use the computer?**

0    1-3    3-5    5-7    More than 7

**4. What do you typically use the computer for? (Please check all that apply)**

Email/word processing/web browsing    Design (Graphical/Mechanical)

Programming    Gaming

Other \_\_\_\_\_

**5. Do you have / have you had the habit of playing video games?**

(Yes No) (If "No," you are done with this questionnaire)

**6. What was your age when you started playing videogames?**

< 5    5-12    12-18    18-25    > 25

**7. On average, how often (hours/week) did you play videogames when you played the most**



**frequently?**  
 1-3       3-7       7-14       14-28       > 28

**How many years ago was that?**

0       3-5       5-10       10-15       > 15

**8. On average, how often (hours/week) have you played videogames over the past 3 years?**

0     1-3       3-7       7-14       14-28       > 28

**9. What kind of video games do you play the most? (check as many as apply)**

- First person
- Role-playing/Strategy
- Arcade/Fighting (please specify: 2D 3D)
- Simulation (driving, flying)
- Sports (which? \_\_\_\_\_ )
- Other \_\_\_\_\_

Thank you. You may hand this questionnaire back to the experimenter.

## APPENDIX C Inter-Session Questionnaire

<p><b>1. Can you repeat the rules to follow? (checklist)</b></p> <p>No collision: _____</p> <p>No singularities: _____</p> <p>Remain in work area: _____</p>	<p><b>2. Can you repeat the instructions to follow? (checklist)</b></p> <p>Pickup (automatic): _____</p> <p>Docking (space bar): _____</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------

**3. Session 1** (start time: \_\_\_\_\_ end time: \_\_\_\_\_)

<b>Did you use any special strategy to pick up?</b>	<b>Did you use any special strategy to dock?</b>

1. How many cameras did you use simultaneously?
2. The camera most used (1, 2 and/or 3)?
3. Did you understand the position of camera correctly (Yes/No)? If yes, can you please describe briefly?

**4. Session 2** (start time: \_\_\_\_\_ end time: \_\_\_\_\_)

<b>Did you use any special strategy to pick up?</b>	<b>Did you use any special strategy to dock?</b>

1. How many cameras did you use simultaneously?
2. The camera most used (1, 2 and/or 3)?
3. Did you understand the position of camera correctly (Yes/No)? If yes, can you please describe briefly?
4. Did you notice any change from the last configuration?

**5. Session 3** (start time: \_\_\_\_\_ end time: \_\_\_\_\_)

<b>Did you use any special strategy to pick up?</b>	<b>Did you use any special strategy to dock?</b>

1. How many cameras did you use simultaneously?
2. The camera most used (1, 2 and/or 3)?
3. Did you understand the position of camera correctly (Yes/No)? If yes, can you please describe briefly?
4. Did you notice any change from the last configuration?

## APPENDIX D Post-Test Questionnaire

Congratulations! You have completed the robotic arm manipulation experiment. Thank you very much for your time and effort. The following questions refer specially to your experience with the desktop virtual reality system. Please answer each question and, if you wish, add any comments.

1. If you experienced any of the following while using the virtual environment, please circle the level of discomfort.

EFFECT	NONE					SEVERE				
B. Nausea	1	2	3	4	5					
C. Dizziness	1	2	3	4	5					
D. Disorientation	1	2	3	4	5					
E. Eyestrain	1	2	3	4	5					
F. Blurred vision	1	2	3	4	5					
G. Sweating	1	2	3	4	5					
H. Headache	1	2	3	4	5					
I. General discomfort		1	2	3	4	5				
J. Mental fatigue	1	2	3	4	5					
K. Other _____	1	2	3	4	5					

1. How enjoyable/interesting was your interaction with the virtual environment?  
 Boring      1      2      3      4      5      Captivating

*Comments?*

2. Rate your proficiency on the following items, after going through the Power Point training:

(1 = very low, 5 = expert)

- Understanding the object properties	1	2	3	4	5
- Understanding the task	1	2	3	4	5
- Using the hand controllers	1	2	3	4	5
- Understanding camera viewpoints	1	2	3	4	5
- Understanding the frames of reference	1	2	3	4	5

3. How did you find the training trials to be?

Too many    1    2    3    4    5    Too few

Very useful    1    2    3    4    5    Useless

*Comments?*

4. How difficult was it for you to translate the arm with the translational controller?

Very difficult    1    2    3    4    5    Very easy

*What made it difficult?*

5. How difficult was it for you to rotate the hand with the rotational controller?

Very difficult    1    2    3    4    5    Very easy

*What made it difficult?*

6. Please describe the three camera configurations that were given to you (in terms of difficulties found translating/rotating the arm).

① Section 1:

② Section 2:

③ Section 3:

7. Indicate where you think the cameras were located IN THE LAST SECTION OF THE EXPERIMENT with respect to the truss intersection? (cameras are numbered from left to right screens)

	Right/Center/Left	Up/Center/Down
Camera 1		
Camera 2		
Camera 3		

8. To perform the task, you mostly monitored:
- 3 displays at a time
  - 2 displays at a time
  - 1 display at a time
9. To perform the task, you mostly translated on:
- 1 axis at a time
  - 2 axes at a time
  - 3 axes at a time
10. To perform the task, you mostly rotated along:
- 1 axis at a time
  - 2 axes at a time
  - 3 axes at a time
11. To translate the module towards the node, you:
- first rotated the module to its desired position and then translated it
  - first translated the module to the final position and then rotated it
  - rotated and translated the module simultaneously
12. To avoid collisions, you: (mark as many as apply)
- stayed as far from possible obstacles as possible
  - foresaw possible collisions and rotated the arm in advance to increase clearance
  - avoided rotating the module as much as possible
  - moved slowly in risky regions
  - moved out of the work area by crossing the semi transparent wall, if necessary
  - didn't care about collisions
  - other: \_\_\_\_\_
13. How important were the following criteria for you to consider a "good docking"?  
(1 = not important, 5 = very important)
- |                          |   |   |   |   |   |
|--------------------------|---|---|---|---|---|
| - Parallel docking marks | 1 | 2 | 3 | 4 | 5 |
| - Coaxial docking marks  | 1 | 2 | 3 | 4 | 5 |
| - Aligned docking marks  | 1 | 2 | 3 | 4 | 5 |

14. Do you have additional suggestions/comments regarding this experiment?

**Thank you! Please return this questionnaire to the experimenter.**

## APPENDIX E Pre-Test Questionnaire Results

Note: The columns Q3 through Q9 correspond to the questions (and answers) of the pre-test questionnaire (APPENDIX B).

Code	F0/M1 Gender	L0/R1 handed	Age	Course	V3D exp	JS exp	Q3	VG	Q4	Q6	Q7	Q7b	Q8	Q9
11	0	1	23	15	0	0	5	0	1,3					
12	0	1	24	16	1	0	5	0	1,3					
13	0	1	22	2	1	1	4	0	1					
14	0	1	25	9	1	1	5	1	1,3	4	2	2	1	1,5
15	0	1	29	22	1	1	5	1	1,3	3	2	4	1	1,4,6
16	0	1	25	9	0	1	5	1	1,3	3	3	3	1	6
17	0	1	39	4	1	0	4	0	1,2					
18	0	1	26	6	1	0	5	1	1,3	3	2	3	1	1
19	0	1	24	3	0	1	3	1	1	3	1	3	1	3,4
20	0	1	22	16	1	1	3	1	1,5	2	3	3	2	1,2,4
21	0	1	23	4	1	1	5	1	1,2,3,4	1	3	1	3	1,2,4
22	0	1	25	22	1	1	4	1	1,3,5	3	2	2	1	1,2
51	1	1	26	16	1	1	5	1	1,3	3	5	3	1	1,2
52	1	0	24	1	0	1	3	1	1	2	2	3	2	4
54	1	1	22	2	1	1	5	1	1,3	5	2	3	0	1,5,6
55	1	1	35	6	1	1	5	1	1,5	2	3	3	2	1,4
57	1	1	27	3	0	1	4	1	1,3	3	1	4	1	5
59	1	1	34	4	1	1	4	1	1,2,3	5	2	2	1	5
60	1	0	24	1	0	0	5	1	1,5	3	2	3	1	1,2
61	1	0	31	12	0	0	4	0	1,3,5					
62	1	0	26	6	1	1	4	1	1,3,5	2	3	3	2	2,4,5,6
63	1	1	24	16	1	1	5	1	1,3,4	1	4	2	3	1,2,4
64	1	1	24	2	1	1	4	1	1,3	1	4	4	2	1,2
65	1	1	23	2	1	0	2	0	1,2					
66	1	0	24	4	1	0	3	0	2					
68	1	1	30	16	1	1	4	1	1,3,5	2	3	4	3	1,2,3,4

## APPENDIX F Post Test Questionnaire Results

Note: The description of questions Q3 through Q9 are given in the questionnaire in APPENDIX D.

Code	Q8	3Cams	B cams	C cams	Q2	Q9	Q10	Q11	Q12	Q13					
										1	2	3	4	5	6
11	1	0	0	1	2	3	2	1	2					1	
12	0	0	0	0	3	2	2	2	1	1	1		1		
13	0	1	1	1	4	2	2	1	3		1	1			
14	1	0	0	1	4	2	1	1	2		1		1		
15	0	0	0	0	4	1	1	1	2	1			1		
16	0	0	0	0	5	1.5	1.5	1	2.5		1			1	
17	0	1	0	1	4	1	1	1	2		1		1		
18	1	1	0	0	3	1	2	1	2	1	1		1		
19	0	1	0	1	3	2	1	1	2	1	1				
20	1	1	0	1	4	1	1	1.5	1.5	1	1	1	1		
21	1	1	0	0	4	1	1	3	1	1	1				
22	1	1	1	1	4	3	2	1	1		1		1		
51	1	0	0	0	4	2	2	1	2				1		
52	0	1	0	0	3	1	1	3	2	1	1				1
54	0	0	1	1	4	2	1	2	2			1			
55	0	1	1	0	5	2	1	1	2	1		1	1		
57	1	1	1	0	3	1	2	2	1	1	1	1			
59	1	1	0	1	4	1	1	1	1.5	1	1				
60	1	0	1	1	4	1	2	1	1		1				
61	1	1	0	1	2	2	1	2	3	1	1				
62	1	0	0	1	4	3	2	1	2				1		
63	1	0	1	1	4	2	1	1	2			1			
64	0	0	0	0	3	3	2	1	1		1				
65	0	0	0	0	4	2	2	1	1	1	1				
66	1	0	1	0	2	3	2	2	3	1	1				1
68	1	1	1	1	4	1	1	1	2						

## APPENDIX G Source Code

```
#
# RWS-Exp1.py
#
# This script creates a PUMA arm simulation with dual joystick control and multiple camera views of the workspace. It opens a network
# connection with another computer running a client version of the program to show a 3rd view of the workspace. This example uses the
# RRG.dll Python extension that encapsulates the RRG Kinematix robot kinematics library. This same script is used for either the
server or
# client - a different callback routine is called depending on the use.
#
# version 1.0 - January 2006
# This version is derived from RWS-sim-xl.py v1.2 (12/05). It has been modified for the preliminary experiments on spatial ability and
# SSRMS operation. It includes new features to read experiment input files defining camera views and initial locations for MPLM target
# objects. Initial experiments will use the arm with an external command frame of reference and fixed camera views (not user
switchable)
# There will be 3 sets of camera views and 4 initial starting locations for the MPLM. Changes are made to the following methods:
GetView(),
# ParseObjectsFile(), keybd()
#
# version 2.0 - Feb 2006
# Fixed some problems with the oType (see comments in code). Using Euler_ZYX mode for EEF vector. Also fixed the MyNetwork()
# callback so the client now works properly with the new object file formats. Had to create some new messages for sending info about
drops
# and new trials. Changed some of the default file names for the experiment control file. Can now handle multiple 'goal objects' in the
object
# file, and scene should be updated in both server and client.
#
# version 3.0 - March 2006
# Look into using the cluster method of networking. This should simplify code since only graphics related stuff is displayed on the
remote
# computers. Only drawback is that some server and client software needs to be started before running the script. On the plus side, the
script
# only has to on the main server machine.
#
# version 4.0 - July 2006
# Changed the DoIK() function to also perform the EEFvector Update. Check if singularity is reached using the latest joystick inputs. If
# singularity is reached, then do not update EEFvector with hand controller inputs, but return original values. Changed the format of the
# return value from RRG method GetHandPosition - now it returns the status (either MOM [measure of manipulability] or the error
code)
# Added a new column to the JA_history list for each trial - now has column whether arm is at a singularity point. 0 = "no" Required a
change
# in WriteJADataToFile() to accommodate extra column
#
# version 5.0 - Sept 2006
# Update the various functions to be compatible with Vizard v3. These updates include changes to the node3d collision methods (disable
# dynamics and use physics engine to check nodes), joysticks/gamepads
#
# version 6.0 - Feb 2007
# Revisions by Alejandra to allow new initial arm positions by trial - in ParseExperimentFile() function. All changes are commented by
# MAMB or ALE notation. New truss configuration is used - set up with 2 trusses along the x and z axes. Made changes to the .arm file
to
# only have 7 columns (eliminate the baserotation)
#
# version 7.0 - Feb 2007
# Added revisions by Alejandra to specify ArmPosition. AML added code to keep the position and orientation history of the target
(position
# in world coordinates, orientation in axis/angle in global coordinates. Still need to write some analysis code to recreate target positions
from
# these values for collision analysis. Can't get joint limits working - GetHandPosition returns funny values...
#
# version 8.0 - Feb 2007
# The program is now compatible with pickup and docking modalities.
#
# version 9.0 - Mar 2007
```



```

#      Included the new Vizard module vizinfo for displaying messages on the window (e.g., for singularities, collisions, etc) Added a new
section
#      to mainloop() for collision checking on the mplm/target after it has been docked. It uses a copy of the target object updated to be at the
#      proper location and checks for collisions with the truss parts and node. Added an output line for keeping track of collisions
throughout the
#      trial.

#      version 10.0 - Mar 2007
#      Some changes from Ale (Progress bar on the dialogs between trials). Added code so forearm is added to objects checked for
collisions. For
#      pickup tasks, will check arm collisions, but for docking, it will just see if upper arm is colliding with mplm.

import RRG
import viz
import string
import vizmat
import time
import os
import vizinfo

#if viz.version()[0] == '3':
#    viz.quit()

viz.go(viz.NO_DEFAULT_KEY | viz.FULLSCREEN)
#viz.framerate()

### Define functions first #####

def MatrixVectorMult(mat,vec):
    outvec = []
    for x in range(0,len(vec)):
        outvec.append(0)
        for y in range(0,len(vec)):
            #          print '%f + (%f * %f)' % (outvec[x],mat[y*4+x],vec[y])
            outvec[x] = outvec[x] + mat[y*4+x]*vec[y]
    return outvec

#creates a new viewpoint based on the parameters given in list
def GetView(cameraview):
    paramlist = string.split(cameraview)
    newView = viz.add(viz.VIEWPOINT)
    newView.translate(string.atof(paramlist[0]),string.atof(paramlist[1]),string.atof(paramlist[2]))    # PosX, PosY, PosZ
    newView.rotate(string.atof(paramlist[3]),string.atof(paramlist[4]),string.atof(paramlist[5]))    # theta, phi, rho
    return newView

# Parses the file with the list of camera viewpoints used in the experiment/simulation
def ParseCameraViewFile(CVFname = 'RWS_Exp1_default.cam'):
    # Still need to figure out what to do for having a cameraview on arm....
    print "Reading Camera file <",CVFname,">..."
    CVlist = []
    fp = open(CVFname, 'r')
    all = fp.readlines()
    if len(all) < 3:
        print "*** ERROR: Must have at least three camera views ***"
        fp.close()
        viz.quit()
    else:
        for line in all:
            CVlist.append(GetView(line))
    fp.close()
    return CVlist

# Updates the viewpoint in a given window
def UpdateWindowViewpoint(winum,viewnum):
    if viewnum == 5:

```

```

#             print "update with view 6"
CameraViewArray[viewnum].update(por.get(viz.MATRIX,viz.ABSOLUTE_WORLD))
winnum.viewpoint(CameraViewArray[viewnum])
#
#     else:
#             print "update with view",viewnum+1
winnum.viewpoint(CameraViewArray[viewnum])

# Writes data to the output file after each trial. Summary data includes the information about input files (camera
# and object files) in a header. Each trial's data is listed on a separate line listing
# trial number, object number, trial starting time, time to grapple object, PointOfResolution matrix at grappling,
# time to completion, target matrix at end of trial.

def WriteSummaryDataToFile(basename, targetdata):
    fp = open('%s-summary.dat' % (basename), 'a')
    outstr = '%d\t%d\t%d\t%6.2f' % (targetdata[0],targetdata[1],targetdata[2],targetdata[3])
    for item in targetdata[4]:
        outstr = outstr + '\t' + str(item)
    outstr = outstr + '\t%6.2f' % (targetdata[5])
    for item in targetdata[6]:
        outstr = outstr + '\t' + str(item)
    outstr = outstr + '\n'
    fp.write(outstr)
    fp.close()

# This function is used to write to current joint angle state of the arm to a file. This will allow reconstruction
# of the path of the EEF and post-experiment checking for collisions or singularities

def WriteJADataToFile(basename,num,data):
    filename = '%s_t%d.ja' % (basename,num)
    fp = open(filename, 'w')
    for line in data:
        outstr = '%s\t%s\t%s\t%s\t%s' % (line[0],line[1],line[2],line[3])
        for item in line[4]:
            outstr = outstr + '\t' + str(item)
        outstr = outstr + '\n'
        fp.write(outstr)
    fp.close()

def WriteTargetDataToFile(basename,num,data):
    filename = '%s_t%d.tgt' % (basename,num)
    fp = open(filename, 'w')
    for line in data:
        outstr = '%s\t%s\t%s\t%s\t%s' % (line[0],line[1],line[2],line[3])
        for item in line[4]:
            outstr = outstr + '\t' + str(item)
        for item in line[5]:
            outstr = outstr + '\t' + str(item)
        outstr = outstr + '\n'
        fp.write(outstr)
    fp.close()

# Read the file that lists all targets and boxes in simulation and add to scene.
# The input file has columns for object type, filename for model, init position, init orientation

def ParseObjectsFile(filename = 'RWS_Exp1_default.obf'):
    print "Reading Object File <",filename,">..."
    targets = []
    goals = []
    obstacles = []

    fp = open(filename, 'r')
    all = fp.readlines()
    for line in all:
        s = string.split(line)
        if s[0] == 'target':
            targets.append([s[1],string.atof(s[2]),string.atof(s[3]),string.atof(s[4]),string.atof(s[5]),string.atof(s[6]),string.atof(s[7])])

```

```

#             thing.collidemesh()
            elif s[0] == 'goal':
goals.append([s[1],string.atof(s[2]),string.atof(s[3]),string.atof(s[4]),string.atof(s[5]),string.atof(s[6]),string.atof(s[7])])
            elif s[0] == 'obstacle':
obstacles.append([s[1],string.atof(s[2]),string.atof(s[3]),string.atof(s[4]),string.atof(s[5]),string.atof(s[6]),string.atof(s[7])])

fp.close()
return [targets, goals, obstacles]

# This function reads the Experiment File which lists the camera views and objects used in the experiment, as
# well as the configuration of each trial (targets, goals, obstacles)

def ParseExperimentFile(datafile, expfile = 'RWS_Exp1_default.exp'):           #MAMB: added arm initial position
    global CurrentTrialNum, NumofTrials
    print "Reading Experiment File <",expfile,">..."
    viewarray = []
    TargetList = []
    Goallist = []
    ObstacleList = []
    HC_gain = [1.0,1.0]
    arminitialposition = []
    fp = open(expfile, 'r')
    trial_list = fp.readlines()
    fp.close()
    NumofTrials = len(trial_list)-6
    # Write out a header for the summary data output file
    if (datafile+'-summary.dat') in os.listdir('.'):
        # Find out which was the last trial completed
        print "Experiment has been previously started"
        ans = viz.ask("Experiment has been started previously. Click YES if you want to continue with the next trial. Click NO to
start anew.")
        if ans:
            newstrial = 0
            PrevTrialFlag = 0
            fp = open(datafile+'-summary.dat','a+')
            prevtrials = fp.readlines()
            for entry in prevtrials:
                prevtrialnum = string.split(entry)[0]
                if prevtrialnum.isdigit() and prevtrialnum > CurrentTrialNum:
                    PrevTrialFlag = 1
                    print "trial num",prevtrialnum,"completed"
                    newstrial = int(prevtrialnum)
            if PrevTrialFlag:
                CurrentTrialNum = newstrial+1
            else:
                CurrentTrialNum = 0
        else:
            fp = open(datafile+'-summary.dat','w')
            CurrentTrialNum = 0
    else:
        fp = open(datafile+'-summary.dat','w')

    fp.write('ExperimentFile\t%s\n' % (expfile))

    for headerline in trial_list[0:6]:           #was [0:5] --MAMB.
        token = string.split(headerline)
        if token[0] == 'version':
            version = token[1]
            fp.write('ExptFileVersion\t%s\n' % (version))
        if token[0] == 'hc_gain':
            HC_gain[0] = float(token[1])
        if token[0] == 'hc_gain':
            HC_gain[1] = float(token[1])
        if token[0] == 'camerafile':
            viewarray = ParseCameraViewFile(token[1])
            fp.write('CameraViewFile\t%s\n' % (token[1]))

```

```

*****ALE*****
    if token[0] == 'armfile':
        [arminitialposition] = ParseInitJAngFile(token[1])
        fp.write('InitJAngFile\t%s\n' % (token[1]))
        print 'InitJAng file read'
*****ALE*****

    if token[0] == 'objectfile':
        if token[1] == "":
            print "ERROR: No object file specified"
            viz.quit()
        else:
            [TargetList, GoalList, ObstacleList] = ParseObjectsFile(token[1])
            fp.write('ObjectDescriptionFile\t%s\n' % (token[1]))

    fp.close()

    return viewarray, arminitialposition, HC_gain, [TargetList, GoalList, ObstacleList], trial_list[6:len(trial_list)] # Skip first 5 lines of
    expt file

*****ALE*****
def ParseInitJAngFile(filename = 'RWS_Exp1_default.arm'):
    global InitJAng
    print "Reading Object File <",filename,">..."
    InitJAng = []
    fp = open(filename,'r')
    all = fp.readlines()
    for line in all:
        s = string.split(line)
        InitJAng.append([string.atof(s[1]),string.atof(s[2]),string.atof(s[3]),string.atof(s[4]),string.atof(s[5]),string.atof(s[6]),string.atof(s[7])])
    fp.close()
    return [InitJAng]

# This function reads the Arm File which lists the initial joint angles of the arm in the trials of the experiment.
*****ALE*****

# Collect information about current experiment
def GetExptInfo():
    global IsDocking, Stage, Group
    subjnum = viz.input('Enter the subject number')
    ##
    if Stage > 1:
    ##
        if (Group == 'g1' and Stage == '2') or (Group == 'g2' and Stage == '3'):
            config = 'B'
    ##
        else:
            config = 'C'
    ##
    else:
        config = 'A'
    ##
    if IsDocking:
        default = 'Exp2_Config'+config+'_d'
    ##
    else:
        default = 'Exp2_Config'+config+'_p'
    ##
    rootname = default
    default = 'Exp2_ConfigA_d'
    rootname = viz.input('Enter the root name of the experiment:',default)
    currentdate = time.strftime("%Y%m%d",time.localtime())
    outputname = 'Subj%s_%s_%s' % (subjnum,rootname,currentdate)
    return rootname+'.exp',outputname

# Update the end-effector position and orientation based on joystick inputs
def UpdateEEFvector(eef, thc, rhc, EEflag):
    global VernierFlag

    # To move the position of the manipulator, simply increment the position entries of the EEF vector
    # using the sid output. NOTE: The y- and z-axes are switched because RRG uses RH coordinates, but
    # Vizard uses LH coordinates.

    # Xform sid data vector by appropriate coordinate frame. If in R0 mode, use identity matrix. If in EE mode

```

```

# then need to xform by inverse of current EEf attitude.
if VernierFlag:
    vernier = vGAIN
else:
    vernier = 1.0

if EEflag:      # If using the internal frame, figure out sid position change in world coordinates
    oldthc = thc
    X = vizmat.Transform()
    X.makeldent()
    X.makeTrans([1,2,3])
    X.set(por.get(viz.MATRIX,viz.ABSOLUTE_WORLD))
    print por.get(viz.EULER,viz.ABSOLUTE_WORLD)
    print por.get(viz.MATRIX,viz.ABSOLUTE_WORLD)
    X.setTrans([0,0,0])
    newvec = MatrixVectorMult(X.get(),[thc[0],thc[1],thc[2],0])
    thc[0] = newvec[0]
    thc[1] = newvec[1]
    thc[2] = newvec[2]
    print "[%4.2f,%4.2f,%4.2f],[%4.2f,%4.2f,%4.2f]" % (oldthc[0],oldthc[1],oldthc[2],thc[0],thc[1],thc[2])#
por.get(viz.MATRIX,viz.ABSOLUTE_WORLD)
#
    pass;
    eef[0] += (HCgains[0] * vernier*thc[0])
    eef[1] += (HCgains[0] * vernier*thc[2])
    eef[2] += (HCgains[0] * vernier*thc[1])

# You can move the end effector orientation similarly by modifying the
# orientation entries of the EEf vector. The end effector angles are in radians!!
# For now we just switch between pos or orient

if oType == 1:
    eef[3] += (-rhc[2]*vernier*HCgains[1])    # Sign flip for the Logitech 3D Extreme joystick
    eef[4] += (rhc[1]*vernier*HCgains[1])
    eef[5] += (rhc[0]*vernier*HCgains[1])
else:
    eef[3] += (rhc[0]*vernier*HCgains[1])
    eef[4] += (rhc[1]*vernier*HCgains[1])
    eef[5] += (rhc[2]*vernier*HCgains[1])

#
# poreuler = j3.get(viz.EULER,viz.ABSOLUTE_WORLD)
# print 'EEf angle mode %d: %6.4f, %6.4f, %6.4f %6.4f, %6.4f, %6.4f %'
(oType,eef[3],eef[4],eef[5],poreuler[0],poreuler[1],poreuler[2])
return eef

# Convert to degrees
def ToDeg(num):
    return num*57.296
# Perform the inverse kinematic calculations to find joint angles from EEf vector
# Returns joint angles in radians
# Changed in version 4 to also call UpdateEEfvector() and to check if arm reaches singularity.
def DoIK(JointAngles, EEfvector, oType, thc, rhc, EEflag):
    global SingFlag
    # Calculate the new EEfvector based on hand controller inputs.
    newEEfvector = UpdateEEfvector( EEfvector, thc, rhc, EEflag ) #Note that y- and z-axes are switched for position in called fcn
    # Calculate the 4x4 homogeneous matrix from the EEf vector
    handvector = RRG.FromVectorToXForm( newEEfvector, oType )
    #After motion, recalculate the inverse kinematics to recover joint angles
    [status,newJointAngles,newLimitStatus] = RRG.GetJointPosition( robotnum, handvector )

# Test out limit status
#
# print newLimitStatus, newEEfvector
# print map(ToDeg,newJointAngles)[0], newJointAngles[0], map(ToDeg,newJointAngles)[1], newJointAngles[1], newLimitStatus
# if (1 in newLimitStatus):
#     print "Joint violation", map(ToDeg,newJointAngles)
#
# else:
#     print map(ToDeg,newJointAngles)
#
# if status < 0.0:

```

```

        # Singularity has been reached - don't update anything
        if SingFlag == 0:
            SingText.visible(viz.ON)
            SingFlag = 1
            return [JointAngles, EEVector]
    else:
# Convert angles back to degrees for Vizard
# for x in range(0,6):
#     JointAngles[x] = JointAngles[x] * RAD2DEG
#     print status, newJointAngles
#     if SingFlag == 1:
#         SingText.visible(viz.OFF)
#         SingFlag = 0

    return [newJointAngles, newEEVector]

# Resets the joint angles and EEF matrix to the starting position. Used to reset arm position between trials.
def ResetArmPose():
    global EEVector, ArmPosition
    # Reset joint angles of the arm
    c2.rotate(0,1,0, -InitJAng[0])
    c3.rotate(0,0,1, -InitJAng[1])
    c4.rotate(0,0,1, -InitJAng[2])
    j1.rotate(0,1,0, InitJAng[3])
    j2.rotate(0,0,1, -InitJAng[4])
    j3.rotate(0,1,0, InitJAng[5])

***ALE***
    sa = string.split(TrialList[CurrentTrialNum])
    ArmPosition = int(string.atof(sa [len(sa)-1]))
    print "ArmPosition: () ", ArmPosition
    c2.rotate(0,1,0, -InitJAng[ArmPosition][0])
    c3.rotate(0,0,1, -InitJAng[ArmPosition][1])
    c4.rotate(0,0,1, -InitJAng[ArmPosition][2])
    j1.rotate(0,1,0, InitJAng[ArmPosition][3])
    j2.rotate(0,0,1, -InitJAng[ArmPosition][4])
    j3.rotate(0,1,0, InitJAng[ArmPosition][5])
#     j3.rotate(1,0,0, InitJAng[ArmPosition][5]) #Good to rotate the hand toward the module on docking trials
***ALE***

    for x in range (0,6):
#         JAnglesRad[x] = InitJAng[x] * DEG2RAD
#         JAnglesRad[x] = InitJAng[ArmPosition][x] * DEG2RAD #MAMB

    # Calculate the initial EEf position and orientation (in robot base coordinates, not Vizard)
    InitMatrix = RRG.GetHandPosition(robotnum,JAnglesRad)
    EEVector = RRG.FromXFormToVector( InitMatrix, oType )

    # Reset the handpos position so collision detection works ok
    handpos.translate(EEVector[0]/1000.0,(EEVector[2]/1000.0), (EEVector[1]/1000.0))

#####
#     End of script functions
#####
# Define the callback routines used in the main part of the simulation. mainloop() is used as the main RWS simulation in Server or Debug mode.
keybd() is also active on the main RWS simulation server.
# This callback removes objects from the last trial and draws the new ones. Also resets the arm position back to the starting position and resets
the counter for grappling each target.

def PrepareNextTrial(num):
    global CurrentTrialNum, dropflag, pickupflag
    global target, target_cd, GoalObject, Obstacle
    global trial_start_time, trial_end_time
    global JA_history, Tgt_history, NumFrames
    global Section, NumofTrials

    linespace = 200*' '

```

```

if CurrentTrialNum < len(TrialList):
    print "Preparing trial", CurrentTrialNum, "..."
    # Remove goal object from the last trial. New one is drawn later
    if len(GoalObject):
        for thing in GoalObject:
            thing.remove()
            print "Removing goal objects from previous trial..."
        GoalObject = []
****ALE***
    # Remove obstacle object from the last trial. New one is drawn later
    if len(Obstacle):
        for thing in Obstacle:
            thing.remove()
            print "Removing obstacles from previous trial..."
        Obstacle = []
****ALE***
    # Remove objects from the last trial
    for thing in target:
        thing.remove()
        print "Removing target from previous trial..."
    # Remove copies from the last trial
    for thing in target_cd:
        thing.remove()
        print "Removing target from previous trial..."

    target = []          # Reset this list for each trial
    target_cd = []
    ResetArmPose()      # Reset the arm to starting position
    dropflag = 0        # Reset list of picked up objects
    pickupflag = 0      # Restore state to not grappling anything

    # Now put in the new ones
    tokentype = 0
    s = string.split(TrialList[CurrentTrialNum])
    for token in s:
        if token.isdigit():
            if tokentype < len(ObjectList):          #NOTE by MAMB: it doesn't like it that there are no
obstacles --that's why the -1
                thing_params = ObjectList[tokentype][int(string.atof(token))-1]
                print "Object", thing_params[0]
                thing = viz.add(thing_params[0])
                thing.translate(thing_params[1], thing_params[2], thing_params[3])
                thing.rotate(thing_params[4], thing_params[5], thing_params[6])
                if tokentype == 0:
                    target.append(thing)
                    print 'target', target
                    thing.collideMesh()
                    thing.disable(viz.DYNAMICS)
                    thing_cd = thing.copy()
                    target_cd.append(thing_cd)          # Make a copy of the target for collision
detection purposes
                    thing_cd.collideMesh()
                    thing_cd.disable(viz.DYNAMICS)
                    thing_cd.visible(viz.OFF)
                elif tokentype == 1:
                    GoalObject.append(thing)
                    thing.collideMesh()
                    thing.disable(viz.DYNAMICS)
                elif tokentype == 2:
                    Obstacle.append(thing)
                    thing.color(1,1,1)
                    thing.alpha(0.2)
                elif tokentype == 3:
                    ArmPosition = int(string.atof(token))
                    print "arm position: ", ArmPosition
                ****ALE*****
                ****ALE*****

```

```

else:
    if token == 'target':
        print "Adding Target objects"
        tokentype = 0
    elif token == 'goal':
        print "Adding Goal Objects"
        tokentype = 1
#####ALE#####
    elif token == 'obstacle':
        print "Obstacle..."
        tokentype = 2
#####ALE#####
    else:
        tokentype = 3
print 'Added %d Goal objects:' % len(GoalObject)
print 'Added %d Targets' % len(target)
print 'Added %d Obstacles' % len(Obstacle)
if CurrentTrialNum == 0:
    line1 = "
else:
    line1 = 'Good job! \n'
line2 = 'Progress bar: '+CurrentTrialNum**' '+ (NumofTrials-CurrentTrialNum)*'_' '+'\n\n'
line3 = '%d trials to go!' \n\n' % (NumofTrials - CurrentTrialNum)
line4 = 'When you are ready to begin the next trial, press the RETURN key.

viz.ask('\n\n\n\n'+line1+line2+line3+line4+linespace+'\n\n\n\n\n')
trial_start_time = time.time()
print "Trial timer started at",trial_start_time
JA_history = []
Tgt_history = []
TrialData = []
NumFrames = 0
print "Resetting joint angle history"
viz.callback(viz.TIMER_EVENT, mainloop)
viz.starttimer(1)

else:
    print "Current Trial Num",CurrentTrialNum
    if Section == '3':
        q = viz.ask(' ***** \n\n\nMISSION ACCOMPLISHED!\n You have finished the
experiment.\n\n\nClick YES to quit.\n *****')
        #
        else:
            q = viz.ask(' ***** \n\n\nCONGRATULATIONS!\n You have finished this part of the
experiment!\n\n\nClick YES to quit.\n *****')
            #
            line10 = 20**'+linespace
            line11 = '\n\n\nCONGRATULATIONS!\n You have finished this part of the experiment!\n\n\nClick YES to quit.\n'
            q = viz.ask(line10+line11+line10)
            if q:
                RRG.DestroyRobot( robotnum )
                viz.quit()

collision_info = vizinfo.add('COLLISION')
collision_info.bgcolor(1.0,0.0,0.0,0.7)
collision_info.visible(viz.OFF)

def mainloop(num):
    global EEFvector, oType
    global robotnum, dropflag, pickupflag, collisionflag
    global JAnglesRad, controlMode
    global trial_grab_time, trial_grab_state
    global JA_history, Tgt_history, SingFlag, TrialData, NumFrames
    global CurrentTrialNum

    ## Need to make changes here for running an actual experiment - keep track of trials and measurements.
    ## Measurements include arm joint angles (complete record), time to pickup and time to place MPLM on node,
    ## final position of MPLM

```



```

# print 'Elapsed Time:',viz.elapsed(),'seconds'
if NumFrames == 0:
    print "Frame counter is set to",NumFrames
    NumFrames += 1

if controlMode == 1:
    data = []
    # Get data from joystick and implement a deadband to prevent drift of manipulator
    # Signs may change depending on the SID that is used!!
    # New method to get joystick data in Vizard3
    siddata = sid.getPosition()
    data.append([siddata[0],siddata[1],sid.getSlider(),sid.getTwist(),siddata[2]]) # Data from SID #1
    siddata = sid2.getPosition()
    data.append([siddata[0],siddata[1],sid2.getSlider(),sid.getTwist(),siddata[2]]) # Data from SID #2
    # Filter data from joysticks and sliders with a central deadband (no inputs)
    for x in range(0,2):
        for y in range(0,4):
            if abs(data[x][y]) < DEADBAND: # Check if joystick values are within deadband
                data[x][y] = 0.0
    if not THCflag: # This means that SID #1 is the Translational Hand Controller
        thc = [data[0][0], -data[0][1], data[0][2]] # The up/down axis sign is flipped because
        rhc = [data[1][0], -data[1][1], data[1][3]] # of the USB gamepad output (MVL). Use twist for rhc
    else:
        thc = [data[1][0], -data[1][1], data[1][2]]
        rhc = [data[0][0], -data[0][1], data[0][3]]
    print data[0]

# # This line has been moved into the DoLK function, so that we can test if singularity is reached but hand
# controller input. IF it is, then just return the undated EEfvector.
# EEfvector = UpdateEEfvector( EEfvector, thc, rhc, EEcontrolModeFlag ) #Note that y- and z-axes are switched for
position in called fcn

# Calculate the new joint angles from inverse kinematic equations
[JAnglesRad, EEfvector] = DoLK( JAnglesRad, EEfvector, oType, thc, rhc, EEcontrolModeFlag )

# Transform the individual links with the calculated joint angles. Note the sign changes since
# Vizard uses LH coordinates and RRG is using RH coordinates.

c2.rotate(0,1,0, -JAnglesRad[0] * RAD2DEG)
c3.rotate(0,0,1, -JAnglesRad[1] * RAD2DEG)
c4.rotate(0,0,1, -JAnglesRad[2] * RAD2DEG)
j1.rotate(0,1,0, JAnglesRad[3] * RAD2DEG)
j2.rotate(0,0,1, -JAnglesRad[4] * RAD2DEG)
# j3.rotate(1,0,0, JAnglesRad[5] * RAD2DEG) #Good to rotate the hand toward the module on docking trials (MAMB)
j3.rotate(0,1,0, JAnglesRad[5] * RAD2DEG)

c4mat = c4.get(viz.MATRIX,viz.ABSOLUTE_WORLD)
c4_cd.update(c4mat, viz.ABSOLUTE_WORLD)
current_time = time.time()
JA_history.append([current_time,pickupflag,dropflag, SingFlag, JAnglesRad]) # Should I include another column with
singularity flag? 7/3/06

# Display a dot at point of resolution. This also requires a shuffle of the axes of translation.
# I need this object to perform collision detection which does not work with hierarchical objects. :(
handpos.translate(EEfvector[0]/1000.0,(EEfvector[2]/1000.0), (EEfvector[1]/1000.0))
# print EEfvector[0]/1000.0,(EEfvector[2]/1000.0), EEfvector[1]/1000.0

# 1/11/06 AML - need to fix this because of new format for establishing viewpoints
# Update the camera view on display 1 if it is the camera view
# if 6 in DisplayStatus:
#     if DisplayStatus.index(6) == 0:
#         UpdateWindowViewpoint(win1,5)
#     elif DisplayStatus.index(6) == 1:
#         UpdateWindowViewpoint(win2,5)

# Keep track of the position of the target during the trial.

```

```

#         Tgt_history.append([current_time, pickupflag, dropflag, collisionflag, target[dropflag].getPosition(viz.ABS_GLOBAL),
target[dropflag].getAxisAngle(viz.ABS_GLOBAL)])
        Tgt_history.append([current_time, pickupflag, dropflag, collisionflag, target[dropflag].getPosition(viz.ABS_GLOBAL),
target[dropflag].getEuler(viz.ABS_GLOBAL)])
        #print target[dropflag].getAxisAngle(viz.ABS_GLOBAL)
        # Use the undocumented collision detection routines to check if grabbing targets
#         print viz.phys.intersectNode(target[dropflag])
        if pickupflag:
            # When module is picked up, start checking collisions with the copy of the object
            target_cd[dropflag].update(target[dropflag].get(viz.MATRIX, viz.ABSOLUTE_WORLD), viz.ABSOLUTE_WORLD)
            target_cd[dropflag].visible(viz.ON)
            #clunk = viz.phys.intersectionNode(target_cd[dropflag])
            if len(set([c4_cd, gx1, gx2, gy1, gy2, GoalObject[dropflag]].intersection(viz.phys.intersectNode(target_cd[dropflag]))) != 0
or len(set([gx1, gx2, gy1, gy2, GoalObject[dropflag], target[dropflag]].intersection(viz.phys.intersectNode(c4_cd))) != 0:
                collision_info.visible(viz.ON)
                collisionflag = 1
            else:
                collision_info.visible(viz.OFF)
                collisionflag = 0

            # Don't have to do anything since target will be part of the end effector scene graph.
            # It's position and orientation will automatically be updated.
            #pass
        else:
            if len(set([gx1, gx2, gy1, gy2, GoalObject[dropflag], target[dropflag]].intersection(viz.phys.intersectNode(c4_cd))) != 0:
                collision_info.visible(viz.ON)
                collisionflag = 1
            else:
                collision_info.visible(viz.OFF)
                collisionflag = 0

        if (handpos in viz.phys.intersectNode(target[dropflag])) and
vizmat.Distance(handpos.get(viz.POSITION), target[dropflag].get(viz.POSITION)) < GRAB_DISTANCE:
            trial_grab_time = time.time() - trial_start_time
            trial_grab_state = por.get(viz.MATRIX, viz.ABSOLUTE_WORLD)
            print 'Grabbed target after %6.2f seconds' % (trial_grab_time)
            pickupflag = 1
            por.visible(viz.ON)

            # Attach the target to the end effector. Update the target's matrix to remain at the current
            # location in world coordinates. When attaching to the end effector, the target's matrix in
            # local object coordinates is used, which will shift and rotate the target.
            mat = target[dropflag].get(viz.MATRIX, viz.ABSOLUTE_WORLD)
            target[dropflag].parent(por)
            target[dropflag].update(mat, viz.ABSOLUTE_WORLD)
            if not IsDocking:
                trial_end_time = time.time() - trial_start_time
                trial_end_state = target[dropflag].get(viz.MATRIX, viz.ABSOLUTE_WORLD)

                print 'Framerate for trial', CurrentTrialNum, ':', NumFrames/trial_end_time, 'fps', NumFrames

            target[dropflag].alpha(0.5) # Lighten color to

indicate dropped

        WriteSummaryDataToFile(basename, [CurrentTrialNum, dropflag, trial_start_time, trial_grab_time, trial_grab_state, trial_end_time, trial_
end_state])

        dropflag = dropflag + 1
        if dropflag < len(target):
            pickupflag = 0
        else:
            print 'Trial # %d took %6.2f seconds' % (CurrentTrialNum, trial_end_time)
            WriteJADataToFile(basename, CurrentTrialNum, JA_history)
            WriteTargetDataToFile(basename, CurrentTrialNum, Tgt_history)
            CurrentTrialNum = CurrentTrialNum + 1
            viz.callback(viz.TIMER_EVENT, PrepareNextTrial)

```

```

viz.starttimer(1)

def kybd(key):
    global dropflag, pickupflag, ClientFlag, EEcontrolModeFlag
    global controlMode, CmodeText, EEFvector, JAnglesRad, oType
    global DisplayStatus, CurrentTrialNum
    global trial_start_time, trial_end_time, trial_end_state
    global basename

    if key == 'q':
        RRG.DestroyRobot(robotnum)
        print "Done"
        viz.quit()

    bounds = len(CameraViewArray)

    if key == viz.KEY_F1:
        window #F1 corresponds to the left-most
            q = input("Select view 1-3 for window 1")
            DisplayStatus[0] = q
            if q >= 1 and q <= bounds:
                CameraViewArray #if input number is in the bounds of
                    viz.cluster.setMask(viz.MASTER)
                    UpdateWindowViewpoint(mainWin,q-1) #change viewpoint on screen 1
                    viz.cluster.setMask(viz.ALLCLIENTS)

                elif key == viz.KEY_F2:
                    window #F2 corresponds to the middle
                        q = input("Select view 1-3 for window 2")
                        DisplayStatus[1] = q
                        if q >= 1 and q <= bounds:
                            viz.cluster.setMask(viz.MASTER)
                            UpdateWindowViewpoint(win2,q-1) #change viewpoint on screen 2
                            viz.cluster.setMask(viz.ALLCLIENTS)

                        elif key == viz.KEY_F3:
                            most window #F3 corresponds to the right-
                                q = input("Select view 1-3 for window 3")
                                DisplayStatus[2] = q
                                if q >= 1 and q <= bounds:
                                    viz.cluster.setMask(viz.CLIENT1)
                                    UpdateWindowViewpoint(mainWin,q-1)
                                    viz.cluster.setMask(viz.ALLCLIENTS)

                                elif key == viz.KEY_ESCAPE:
                                    #if escape hit, quit program
                                        RRG.DestroyRobot(robotnum)
                                        print "Done"
                                        viz.quit()

                                elif key == '' and pickupflag:
                                    #signal for arm to
                                        drop object
                                            trial_end_time = time.time()-trial_start_time
                                            trial_end_state = target[dropflag].get(viz.MATRIX, viz.ABSOLUTE_WORLD)

                                            print 'Framerate for trial',CurrentTrialNum,',',NumFrames/trial_end_time,'fps ',NumFrames

                                            mat = target[dropflag].get(viz.MATRIX,viz.ABSOLUTE_WORLD)
                                            target[dropflag].parent(viz.WORLD)
                                            target[dropflag].update(mat, viz.ABSOLUTE_WORLD)

                                            target[dropflag].alpha(0.5) # Lighten color to indicate dropped

                                            por.visible(viz.OFF)

                                WriteSummaryDataToFile(basename,[CurrentTrialNum,dropflag,trial_start_time,trial_grab_time,trial_grab_state,trial_end_time,trial_
                                end_state])

                                dropflag = dropflag + 1
                                if dropflag < len(target):

```

```

        pickupflag = 0
    else:
        print 'Trial # %d took %.2f seconds' % (CurrentTrialNum, trial_end_time)
        WriteJADataToFile(basename, CurrentTrialNum, JA_history)
        WriteTargetDataToFile(basename, CurrentTrialNum, Tgt_history)
        CurrentTrialNum = CurrentTrialNum + 1
        viz.callback(viz.TIMER_EVENT, PrepareNextTrial)
elif key == 'c':
    # Flip control modes between JA
    if controlMode == 1:
        controlMode = 0
        CmodeText[0].visible(viz.ON)
        print "JA on"
    else:
        controlMode = 1
        # Update the EEf vector with the current joint angles
        NewMatrix = RRG.GetHandPosition( robotnum, JAnglesRad )
        EEfvector = RRG.FromXFormToVector( NewMatrix, oType )
        CmodeText[0].visible(viz.OFF)
        print "JA off"
elif key == 't':
    if controlMode and EEcontrolModeFlag == 0:
        EEcontrolModeFlag = 1
        print "Using EE internal control frame"
        CmodeText[1].visible(viz.ON)
    elif controlMode and EEcontrolModeFlag:
        EEcontrolModeFlag = 0
        print "Using default EE external control frame"
        CmodeText[1].visible(viz.OFF)
elif key == 'r':
    ResetArmPose()

elif not controlMode:
    # Use these keys to control arm in JA mode
    if key == '1':
        JAnglesRad[0] = JAnglesRad[0] + JAstep*DEG2RAD
    elif key == '!':
        JAnglesRad[0] = JAnglesRad[0] - JAstep*DEG2RAD
    elif key == '2':
        JAnglesRad[1] = JAnglesRad[1] + JAstep*DEG2RAD
    elif key == '@':
        JAnglesRad[1] = JAnglesRad[1] - JAstep*DEG2RAD
    elif key == '3':
        JAnglesRad[2] = JAnglesRad[2] + JAstep*DEG2RAD
    elif key == '#':
        JAnglesRad[2] = JAnglesRad[2] - JAstep*DEG2RAD
    elif key == '4':
        JAnglesRad[3] = JAnglesRad[3] + JAstep*DEG2RAD
    elif key == '$':
        JAnglesRad[3] = JAnglesRad[3] - JAstep*DEG2RAD
    elif key == '5':
        JAnglesRad[4] = JAnglesRad[4] + JAstep*DEG2RAD
    elif key == '%':
        JAnglesRad[4] = JAnglesRad[4] - JAstep*DEG2RAD
    elif key == '6':
        JAnglesRad[5] = JAnglesRad[5] + JAstep*DEG2RAD
    elif key == '^':
        JAnglesRad[5] = JAnglesRad[5] - JAstep*DEG2RAD

def hatevent(hatval):
    global VernierFlag

    # Check hatswitch for coarse/vernier mode
    if hatval >=0:
        VernierFlag = not VernierFlag
        print "Hat switch pressed", hatval

```

```
#####
```

```

#      End of Callback routines
#####
# Begin main portion of program
#GAIN = 15.0
vGAIN = 0.75          # Sets reduction gain for vernier mode
DEADBAND = 0.25      # Joystick deadband
JASstep = 1.0        # Step size when using joint angle control (these are degrees!)
GRAB_DISTANCE = 0.3  # Max Distance from grapple point to consider MPLM as grabbed. This is the R or the grapple fixture

DEG2RAD = 0.0174532925199
RAD2DEG = 57.2957795131

DebugFlag = 1
ClientFlag = 0
VernierFlag = 0
EEcontrolModeFlag = 0      # If set to 1, then EE control using an internal frame (moves with EE)
SingFlag = 0               # Indicates if arm is presently at a singularity
HCgains = [1.0,1.0]
# OpModes are as follows:
#      0 = Debug - small windows
#      1 = Experiment mode - full windows
OpMode = viz.choose('Select the operating mode',['Debug mode',"Experiment mode"])
##Group = viz.input('Subject Group (g1 or g2)','g1')
##Stage = viz.input('Stage (1,2 or 3)','1')
IsDocking = viz.choose('Select the mission',['Pick up',"Docking"])

# Read the files listing all targets, goals, obstacles for the experiment. See comments with function definition
# for input file format

CurrentTrialNum = 0
[exptfile,basename] = GetExptInfo()
#####ALE#####
#[CameraViewArray,HCgains,ObjectList,TrialList] = ParseExperimentFile(basename, exptfile)      # Default file is 'RWS_Exp1_default.exp'
[CameraViewArray,InitJAng,HCgains,ObjectList,TrialList] = ParseExperimentFile(basename, exptfile)
sa = string.split(TrialList[0])
ArmPosition = int(string.atof(sa [len(sa)-1]))
#####ALE#####

target = []             # Holds list of targets for a single trial
target_cd = []         # Holds copies of targets for collisiondetection during trial (after targets are picked up)
#viz.framerate()
if OpMode == 0:        # Zero means script runs in debug mode
    viz.cluster.setMask(viz.CLIENT1)
    viz.window.size(640,480)
    viz.cluster.setMask(viz.MASTER)
    viz.window.size(1280,480)
    viz.cluster.setMask(viz.ALLCLIENTS)

# Add joysticks or gamepads to control robot. Vizjoy is the new Vizard3 dll
import vizjoy

sid = vizjoy.add()     # This will be the first gamepad/joystick installed
print sid.getName()
sid2 = vizjoy.add()   # This will be the second gamepad/joystick installed
print sid2.getName()

#import sid            # Make sure the sid.py and gamepad.dll are latest ones that allow more than one USB SID!!
#sid2 = sid.add()     # This is the second hand controller - SID #2 Figure out RHC and THC later

THCflag = viz.ask('Is the Translation Hand Controller **NOT** SID #1?')

#sets initial viewpoints
mainView = viz.get(viz.MAIN_VIEWPOINT)
mainWin = viz.get(viz.MAIN_WINDOW)
viz.cluster.setMask(viz.CLIENT1)
mainWin.size(1,1)
mainWin.viewpoint(CameraViewArray[0])          #changed from 2 to 0

```

```

viz.cluster.setMask(viz.MASTER)
# Scrunch main window to left of screen
mainWin.size(.5,1)
mainWin.viewpoint(CameraViewArray[1])          #changed from 0 to 1
#add window on left of screen
win2=viz.add(viz.WINDOW)
win2.position(.5,1)
win2.size(.5,1)
win2.viewpoint(CameraViewArray[2])            #changed from 1 to 2
viz.cluster.setMask(viz.ALLCLIENTS)
DisplayStatus = [0,1,2]                      # Shows the current viewpoint for each of the three displays. By default starts with 1st three entries

#
# These are defined in J. Craig, Intro to Robotics, starting on page 45...
# Assume the fixed frame of reference is the base reference frame. This will correspond to the Joint 0 frame
# which is also the same as Joint 1. x to the right, y into page, z up

oType = 1 # RRG has header file listing 2 = Fixed_XYZ, 0 = Euler_XYZ, 1 = Euler_ZYZ  AML corrected these on 3/31/05
# BUT really seems to work as such 0=Euler_ZYZ, 1=Euler_ZYX and 2=Fixed_XYZ - has implications for
UpdateEEFvector()

# Set initial control mode for robot: 0 = joint angle; 1 = end-effector rate
controlMode = 1
CmodeText = []
CmodeText.append(vizinfo.add('Joint Angle Mode'))
CmodeText[0].translate(0.05,0.95)
CmodeText[0].alignment(vizinfo.UPPER_LEFT)
CmodeText[0].visible(viz.OFF)
CmodeText.append(vizinfo.add('EE Internal Frame Mode'))
CmodeText[1].translate(0.05,0.90)
CmodeText[1].alignment(vizinfo.UPPER_LEFT)
CmodeText[1].visible(viz.OFF)
SingText = vizinfo.add('Singularity')
SingText.translate(0.95,0.85)
SingText.visible(viz.OFF)

# Enable Physics Engine to check collisions between objects
viz.phys.enable()
# Set the initial pose of the robot
#InitJAng = [-0.0, -40.0, 10.0, 0.0, -60.0, 0.0, 0.0]  # These are specified in degrees but RRG uses radians
JAnglesRad = [0.0, 0.0, 0.0, 0.0, 0.1, 0.1, 0.0]      # Holds joint angles in radians!
EEFvector = []
robotnum = RRG.CreateRobot( 'mvpuma-xl.dh',InitJAng[ArmPosition] )

# Add all the robot parts into the scene graph
c1 = viz.add('mvpuma-waist.wrl')
c2 = c1.addchild('mvpuma-shoulder.wrl')
c3 = c2.addchild('mvpuma-upperarm-xl.wrl')
c4 = c3.addchild('mvpuma-forearm-xl.wrl')
j1 = c4.addchild('mvpuma-j4.wrl')
j2 = j1.addchild('mvpuma-j5.wrl')
j3 = j2.addchild('mvpuma-j6.wrl')
por = j3.addchild('joint3.wrl')                 # This object is just to show the point of resolution

# This copy of the c4 link will get checked for collisions.
c4_cd = c4.copy()
c4_cd.collidemesh()
c4_cd.disable(viz.DYNAMICS)
c4_cd.visible(viz.OFF)

# Move the pieces to the correct places in the zero-angle configuration
c2.translate(0, 2.5, 0)
c3.translate(0, 0, .50)
c4.translate(4.0, 0.0, -.30)
j1.translate(0, -4.0, 0)
por.translate(0,-0.22,0)
por.rotate(90,90,0)                            # Rotate to get z-axis pointing in right direction - hope this doesn't mess up pickup

```

```

por.visible(viz.OFF)                # NOTE: Any child objects will also become invisible!!!

##### ALE #####
#g = viz.add('truss2.wrl')
#g.translate(0,0,1)
#g.collideBox()
#g.disable(viz.DYNAMICS)
gx1 = viz.add('truss-orange.wrl')    #initial truss
gx2 = viz.add('truss2.wrl')          #extending initial truss
gx2.translate(0,0,15)
gx1.collideBox()
gx1.disable(viz.DYNAMICS)
gx2.collideBox()
gx2.disable(viz.DYNAMICS)
gx1.color(1,.4,0)    #1 0 0 = red, 1 1 0 = yellow, 1 1 1 = white,
gy1 = viz.add('truss2.wrl')
gy1.translate(15,0,0)
gy1.rotate(90,0,0)
gy2 = viz.add('truss2.wrl')          #perpendicular trusses
gy2.rotate(90,0,0)
gy1.collideBox()
gy1.disable(viz.DYNAMICS)
gy2.collideBox()
gy2.disable(viz.DYNAMICS)

##gx1 = viz.add('truss-orange.wrl')    #initial truss
##gx2 = viz.add('truss-turquoise.wrl') #extending initial truss
##gx2.translate(0,0,15)
##gx1.collideBox()
##gx1.disable(viz.DYNAMICS)
##gx2.collideBox()
##gx2.disable(viz.DYNAMICS)
##gx1.color(1,.4,0)    #1 0 0 = red, 1 1 0 = yellow, 1 1 1 = white,
##gx2.color(0,.5,.5)  #0 1 0 = green, 0 0 1 = blue, 1 0 1 = purple, 0 1 1 = turquoise
##gy1 = viz.add('truss-purple.wrl')
##gy1.translate(15,0,0)
##gy1.rotate(90,0,0)
##gy2 = viz.add('truss-purple.wrl')    #perpendicular trusses
##gy2.rotate(90,0,0)
##gy1.collideBox()
##gy1.disable(viz.DYNAMICS)
##gy2.collideBox()
##gy2.disable(viz.DYNAMICS)
##gy1.color(.5,0,0.5) #1 0 0 = red, 1 1 0 = yellow, 1 1 1 = white,
##gy2.color(.5,0,0.5) #0 1 0 = green, 0 0 1 = blue, 1 0 1 = purple, 0 1 1 = turquoise
#ObstacleCube = viz.add('obstaclecube.wrl')
#ObstacleCube.color(1,1,1)
#ObstacleCube.alpha(.3)
#ObstacleCube.translate(-.05,0,0)

##### ALE #####

# Set limits on joint angle rotation - try to prevent hitting itself
# Still not working in an understandable manner!! AML
#RRG.SetJointLimits( robotnum, 'mvlpuma.limits' )
#print "Set Joint Limits"

#Note that the coordinate frames for each of the pieces is different in Vizard and RRG. In this case, we
# only need a simple sign change to get the pieces to rotate in the correct direction as Vizard uses LH coordinates
# and the RRG convection is to use RH coordinates.

##### ALE #####
#c1.rotate(1,0,0, InitJAng[7])
#c2.rotate(0,1,0, -InitJAng[0])
#c3.rotate(0,0,1, -InitJAng[1])
#c4.rotate(0,0,1, -InitJAng[2])
#j1.rotate(0,1,0, InitJAng[3])

```

```

#j2.rotate(0,0,1, -InitJAng[4])
#j3.rotate(0,1,0, InitJAng[5])
##### ALE #####

# Set our point of resolution between the gripper fingers is about 20cm down the
# z-axis of the last frame. This is where the por object lies.
# This functionality was added to the RRG.dll v1.3 by AML 6-8-05
toolPoint = (0,0,220)
RRG.SetToolPoint( robotnum, toolPoint )

# Set the base pose to be at the same origin as Vizard coordinates. Since one cannot rotate a RH coordinate frame into
# a LH frame, we must still manually re-map the axes to draw correctly....

RobotBaseXform = [1,0,0,0,0,1,0,0,0,0,1,2500,0,0,0,1]
RRG.SetBasePose( robotnum, RobotBaseXform )
# Set some other values for the inverse calculations.
RRG.SetInverseErrorTolerance( robotnum, 0.01, 1000.0 )
RRG.SetMinimumMOT( robotnum, 100 )

# Just put an object at the tool point (see a few lines above). This object is used for the collision detection
# so we can tell when to pick up an object. We don't need to render it to do collision detection.
handpos = viz.add('joint2.wrl')
#handpos.visible(viz.OFF)
handpos.collideMesh()
handpos.disable(viz.DYNAMICS)
# The GetHandPosition method requires an array of angles in radians

for x in range (0,6):
#####ALE#####
        #JAnglesRad[x] = InitJAng[CurrentTrialNum][x] * DEG2RAD
        JAnglesRad[x] = InitJAng[ArmPosition][x] * DEG2RAD
#####ALE#####
#        JAnglesRad[x] = InitJAng[x] * DEG2RAD
#        JAnglesFromIK[x] = InitJAng[x] * DEG2RAD

# Calculate the initial EEf position and orientation (in robot base coordinates, not Vizard)
InitMatrix = RRG.GetHandPosition(robotnum,JAnglesRad)
EEFvector = RRG.FromXFormToVector( InitMatrix, oType )
pickupflag = dropflag = 0          # Set initial state for pick up and drop flags
collisionflag = 0
NumFrames = 0
trial_start_time = 0
trial_grab_time = 0
trial_end_time = 0
trial_grab_state = []
trial_end_state = []
JA_history = []
Tgt_history = []
TrialData = []
GoalObject = []
Obstacle = []
viz.gravity(0)
#sid.callback(sid.HAT_EVENT,hatevent)
viz.callback(viz.KEYBOARD_EVENT,kybd)
viz.callback(viz.TIMER_EVENT, PrepareNextTrial)
viz.starttimer(1)

```



# APPENDIX H Summary of Mixed Regressions

## PICK UP PHASE

### PTA

#### 1. L\_TASK

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	1.711	0.019	89.155	0.000
GROUP	-0.000	0.019	-0.011	0.992
GENDER	0.072	0.021	3.462	0.001
FXD3 (1)	0.043	0.013	3.352	0.001
FXD3 (2)	-0.083	0.013	-6.605	0.000
FXD4 (1)	-0.059	0.024	-2.509	0.012
FXD4 (2)	0.101	0.024	4.288	0.000
FXD4 (3)	-0.164	0.024	-6.940	0.000
FXD4 (4)	-0.067	0.024	-2.848	0.004
FXD4 (5)	0.031	0.024	1.313	0.189
FXD4 (6)	0.086	0.024	3.637	0.000
FXD4 (7)	0.075	0.024	3.159	0.002
VGEXP	-0.003	0.020	-0.138	0.890
SETCAMS	0.013	0.019	0.691	0.490
STDAGE	0.013	0.021	0.596	0.551
STDCC	-0.032	0.020	-1.551	0.121
STDPTA	0.018	0.021	0.857	0.391
FXD10 (1)	-0.005	0.013	-0.405	0.686
FXD10 (2)	0.027	0.013	2.176	0.030
FXD11 (1)	-0.009	0.013	-0.689	0.491
FXD11 (2)	0.035	0.013	2.779	0.005

#### 2. L\_TOBS

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.247	0.035	7.031	0.000
GROUP	-0.026	0.035	-0.731	0.465
GENDER	0.032	0.038	0.840	0.401
FXD3 (1)	0.048	0.012	4.108	0.000
FXD3 (2)	-0.027	0.012	-2.386	0.017
FXD4 (1)	-0.038	0.022	-1.751	0.080
FXD4 (2)	-0.011	0.022	-0.526	0.599
FXD4 (3)	0.047	0.022	2.173	0.030
FXD4 (4)	-0.035	0.022	-1.647	0.100
FXD4 (5)	-0.003	0.022	-0.156	0.876
FXD4 (6)	0.062	0.022	2.883	0.004
FXD4 (7)	0.053	0.022	2.463	0.014
VGEXP	-0.000	0.037	-0.001	0.999
SETCAMS	-0.001	0.035	-0.014	0.988
STDAGE	0.066	0.039	1.691	0.091
STDCC	-0.046	0.037	-1.241	0.214

STDPTA	-0.018	0.038	-0.487	0.626
FXD10 (1)	-0.016	0.012	-1.374	0.169
FXD10 (2)	0.013	0.012	1.115	0.265
FXD11 (1)	-0.031	0.012	-2.594	0.009
FXD11 (2)	0.038	0.012	3.305	0.001

### 3. L\_TCOLL

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.247	0.062	4.002	0.000
GROUP	0.022	0.064	0.339	0.734
GENDER	0.081	0.070	1.159	0.246
FXD3 (1)	0.011	0.070	0.151	0.880
FXD3 (2)	-0.152	0.078	-1.951	0.051
STDAGE	0.105	0.062	1.703	0.089
STDCC	-0.114	0.064	-1.787	0.074
STDPTA	0.072	0.073	0.986	0.324
FXD7 (1)	-0.065	0.072	-0.905	0.366
FXD7 (2)	-0.015	0.078	-0.198	0.843
FXD8 (1)	0.083	0.064	1.288	0.198
FXD8 (2)	-0.185	0.066	-2.806	0.005

### 5. T\_PCMOTION

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.929	0.010	94.121	0.000
GROUP	0.015	0.010	1.502	0.133
GENDER	-0.011	0.011	-0.980	0.327
FXD3 (1)	-0.012	0.006	-1.976	0.048
FXD3 (2)	0.039	0.006	6.462	0.000
FXD4 (1)	-0.013	0.011	-1.191	0.234
FXD4 (2)	-0.026	0.011	-2.357	0.018
FXD4 (3)	0.002	0.011	0.141	0.888
FXD4 (4)	0.019	0.011	1.707	0.088
FXD4 (5)	0.001	0.011	0.129	0.897
FXD4 (6)	0.035	0.011	3.106	0.002
FXD4 (7)	-0.010	0.011	-0.889	0.374
VGEXP	0.005	0.010	0.480	0.631
SETCAMS	0.016	0.010	1.651	0.099
STDAGE	0.015	0.011	1.387	0.166
STDCC	-0.011	0.010	-1.072	0.284
STDPTA	-0.021	0.011	-1.994	0.046
FXD10 (1)	0.004	0.006	0.680	0.497
FXD10 (2)	-0.023	0.006	-3.764	0.000
FXD11 (1)	0.004	0.006	0.590	0.555
FXD11 (2)	-0.010	0.006	-1.722	0.085

### 6. T\_DOFAX

Variable	Estimate	Standardized Error	Z	p-value
----------	----------	--------------------	---	---------

INTERCEPT	0.254	0.016	15.897	0.000
GROUP	0.009	0.016	0.554	0.580
GENDER	-0.032	0.017	-1.942	0.052
FXD3 (1)	-0.035	0.012	-3.001	0.003
FXD3 (2)	0.051	0.011	4.453	0.000
STDAGE	-0.028	0.017	-1.712	0.087
STDPTA	-0.022	0.018	-1.205	0.228
STDCC	0.014	0.017	0.799	0.424
VGEXP	0.012	0.017	0.711	0.477
FXD8 (1)	-0.016	0.012	-1.376	0.169
FXD8 (2)	0.001	0.011	0.072	0.942
FXD9 (1)	0.006	0.012	0.500	0.617
FXD9 (2)	0.001	0.011	0.103	0.918

### 7. T\_DOFANG

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.907	0.022	41.651	0.000
GROUP	-0.018	0.020	-0.900	0.368
GENDER	0.055	0.022	2.510	0.012
FXD3 (1)	0.021	0.012	1.820	0.069
FXD3 (2)	-0.020	0.012	-1.686	0.092
STDAGE	0.064	0.027	2.373	0.018
STDPTA	0.029	0.023	1.281	0.200
STDCC	-0.053	0.022	-2.410	0.016
VGEXP	0.025	0.023	1.117	0.264
FXD8 (1)	0.009	0.012	0.767	0.443
FXD8 (2)	-0.002	0.012	-0.127	0.899
FXD9 (1)	-0.000	0.012	-0.032	0.974
FXD9 (2)	-0.012	0.012	-0.964	0.335

### 8. P\_DOFANXANG

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.586	0.017	33.527	0.000
GROUP	-0.004	0.017	-0.256	0.798
GENDER	-0.005	0.018	-0.276	0.782
FXD3 (1)	-0.015	0.012	-1.323	0.186
FXD3 (2)	0.016	0.011	1.405	0.160
STDAGE	0.005	0.018	0.277	0.782
STDPTA	-0.020	0.020	-1.024	0.306
STDCC	0.008	0.019	0.431	0.666
VGEXP	0.017	0.018	0.911	0.362
FXD8 (1)	0.009	0.012	0.773	0.439
FXD8 (2)	-0.014	0.012	-1.247	0.213
FXD9 (1)	-0.004	0.012	-0.358	0.720
FXD9 (2)	-0.013	0.012	-1.125	0.260

### 9. L\_DIMX

Variable	Estimate	Standardized Error	Z	p-value
----------	----------	-----------------------	---	---------

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.545	0.031	17.732	0.000
GROUP	0.030	0.031	0.980	0.327
GENDER	0.062	0.032	1.960	0.050
FXD3 (1)	-0.061	0.027	-2.289	0.022
FXD3 (2)	-0.064	0.026	-2.434	0.015
FXD4 (1)	0.111	0.047	2.351	0.019
FXD4 (2)	0.125	0.048	2.617	0.009
FXD4 (3)	-0.341	0.054	-6.349	0.000
FXD4 (4)	-0.179	0.053	-3.390	0.001
FXD4 (5)	0.071	0.048	1.470	0.142
FXD4 (6)	0.106	0.047	2.243	0.025
FXD4 (7)	0.062	0.047	1.320	0.187
VGEXP (1)	0.020	0.032	0.619	0.536
SETCAMS (1)	0.037	0.011	3.277	0.001
STDAGE	0.002	0.032	0.054	0.957
STDCC	-0.010	0.031	-0.334	0.739
STDPTA	-0.054	0.032	-1.675	0.094
FXD10 (1)	-0.008	0.027	-0.313	0.755
FXD10 (2)	0.010	0.026	0.365	0.715
FXD11 (1)	-0.022	0.027	-0.829	0.407
FXD11 (2)	0.045	0.026	1.688	0.091

## 10. L\_DIMZ

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.135	0.035	3.920	0.000
GROUP	0.049	0.038	1.306	0.191
GENDER	-0.012	0.042	-0.286	0.775
FXD3 (1)	0.034	0.033	1.046	0.296
FXD3 (2)	-0.177	0.018	-9.827	0.000
FXD4 (1)	0.072	0.015	4.674	0.000
FXD4 (2)	0.107	0.068	1.576	0.115
FXD4 (3)	-0.084	0.078	-1.066	0.287
FXD4 (4)	-0.664	0.065	-10.236	0.000
FXD4 (5)	0.087	0.071	1.223	0.221
FXD4 (6)	0.291	0.068	4.272	0.000
FXD4 (7)	0.088	0.068	1.302	0.193
VGEXP (1)	0.013	0.011	1.112	0.266
SETCAMS (1)	-0.039	0.039	-0.991	0.322
STDAGE	-0.111	0.043	-2.550	0.011
STDCC	0.069	0.041	1.682	0.092
STDPTA	-0.112	0.040	-2.806	0.005
FXD10 (1)	0.027	0.037	0.715	0.475
FXD10 (2)	0.009	0.037	0.248	0.804
FXD11 (1)	0.031	0.037	0.852	0.394
FXD11 (2)	0.009	0.038	0.235	0.814

## 11. L\_DIMY

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.246	0.030	8.107	0.000

GROUP	0.025	0.031	0.832	0.406
GENDER	0.028	0.033	0.862	0.389
FXD3 (1)	0.021	0.027	0.767	0.443
FXD3 (2)	-0.032	0.027	-1.191	0.234
FXD4 (1)	-0.093	0.051	-1.833	0.067
FXD4 (2)	0.174	0.049	3.541	0.000
FXD4 (3)	-0.059	0.053	-1.107	0.268
FXD4 (4)	-0.037	0.049	-0.743	0.458
FXD4 (5)	0.029	0.051	0.573	0.566
FXD4 (6)	-0.021	0.050	-0.413	0.680
FXD4 (7)	0.027	0.050	0.545	0.586
VGEXP (1)	-0.012	0.032	-0.373	0.709
SETCAMS (1)	0.042	0.031	1.353	0.176
STDAGE	0.003	0.033	0.098	0.922
STDCC	-0.038	0.032	-1.191	0.234
STDPTA	-0.127	0.033	-3.849	0.000

## PSVV

### 1. L\_TASK

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	1.713	0.019	88.094	0.000
GROUP	0.014	0.024	0.589	0.556
GENDER	0.079	0.025	3.196	0.001
FXD3 (1)	0.043	0.013	3.389	0.001
FXD3 (2)	-0.083	0.013	-6.568	0.000
VGEXP	-0.002	0.020	-0.093	0.926
SETCAMS	0.026	0.025	1.017	0.309
STDAGE	0.002	0.023	0.107	0.915
STDCC	-0.037	0.023	-1.631	0.103
FXD8 (1)	-0.059	0.024	-2.509	0.012
FXD8 (2)	0.101	0.024	4.289	0.000
FXD8 (3)	-0.164	0.024	-6.941	0.000
FXD8 (4)	-0.067	0.024	-2.848	0.004
FXD8 (5)	0.031	0.024	1.313	0.189
FXD8 (6)	0.086	0.024	3.637	0.000
FXD8 (7)	0.075	0.024	3.159	0.002
STDPSVV	0.031	0.033	0.938	0.348
FXD10 (1)	-0.002	0.013	-0.121	0.904
FXD10 (2)	0.033	0.013	2.547	0.011
FXD11 (1)	0.012	0.013	0.936	0.349
FXD11 (2)	0.025	0.013	1.918	0.055

### 2. L\_TOBS

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.252	0.036	7.017	0.000
GROUP	-0.018	0.044	-0.399	0.690
GENDER	0.049	0.046	1.080	0.280

FXD3 (1)	0.049	0.012	4.170	0.000
FXD3 (2)	-0.028	0.012	-2.373	0.018
VGEXP	0.007	0.038	0.173	0.863
SETCAMS	0.013	0.047	0.280	0.780
STDAGE	0.063	0.043	1.472	0.141
STDCC	-0.059	0.042	-1.424	0.155
FXD8 (1)	-0.038	0.022	-1.736	0.083
FXD8 (2)	-0.011	0.022	-0.522	0.602
FXD8 (3)	0.047	0.022	2.155	0.031
FXD8 (4)	-0.035	0.022	-1.633	0.103
FXD8 (5)	-0.003	0.022	-0.155	0.877
FXD8 (6)	0.062	0.022	2.858	0.004
FXD8 (7)	0.053	0.022	2.442	0.015
STDPSVV	0.017	0.061	0.282	0.778
FXD10 (1)	-0.014	0.012	-1.138	0.255
FXD10 (2)	0.015	0.012	1.252	0.211
FXD11 (1)	0.002	0.012	0.143	0.886
FXD11 (2)	0.013	0.012	1.080	0.280

### 3. L\_TCOLL

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.270	0.070	3.852	0.000
GROUP	0.004	0.076	0.052	0.959
GENDER	0.035	0.072	0.489	0.625
FXD3 (1)	-0.023	0.078	-0.299	0.765
FXD3 (2)	-0.106	0.082	-1.290	0.197
STDAGE	0.095	0.085	1.115	0.265
STDCC	-0.085	0.065	-1.303	0.193
STDPSVV	0.006	0.106	0.053	0.958
VGEXP	0.049	0.064	0.765	0.444
FXD8 (1)	-0.089	0.082	-1.096	0.273
FXD8 (2)	-0.055	0.088	-0.630	0.529
FXD9 (1)	0.012	0.073	0.170	0.865
FXD9 (2)	-0.141	0.071	-1.969	0.049

### 5. T\_PCMOTION

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.929	0.010	88.634	0.000
GROUP	0.005	0.013	0.372	0.710
GENDER	-0.011	0.013	-0.800	0.424
FXD3 (1)	-0.012	0.006	-2.039	0.041
FXD3 (2)	0.039	0.006	6.507	0.000
VGEXP	0.006	0.011	0.589	0.556
SETCAMS	0.010	0.014	0.726	0.468
STDAGE	0.023	0.012	1.845	0.065
STDCC	-0.011	0.012	-0.907	0.364
FXD8 (1)	-0.013	0.011	-1.204	0.229
FXD8 (2)	-0.026	0.011	-2.382	0.017
FXD8 (3)	0.002	0.011	0.142	0.887
FXD8 (4)	0.019	0.011	1.724	0.085

FXD8 (5)	0.001	0.011	0.130	0.896
FXD8 (6)	0.035	0.011	3.138	0.002
FXD8 (7)	-0.010	0.011	-0.898	0.369
STDPSVV	-0.021	0.018	-1.186	0.236
FXD10 (1)	0.001	0.006	0.148	0.883
FXD10 (2)	-0.025	0.006	-4.126	0.000
FXD11 (1)	-0.012	0.006	-1.954	0.051
FXD11 (2)	-0.012	0.006	-1.905	0.057

## 6. T\_DOFAX

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.249	0.014	18.165	0.000
GROUP	-0.017	0.016	-1.072	0.284
GENDER	-0.047	0.014	-3.247	0.001
FXD3 (1)	-0.035	0.012	-3.020	0.003
FXD3 (2)	0.051	0.011	4.458	0.000
STDAGE	0.001	0.017	0.037	0.970
STDCC	0.021	0.014	1.511	0.131
VGEXP	0.006	0.014	0.437	0.662
STDPSVV	-0.064	0.019	-3.379	0.001
FXD8 (1)	-0.018	0.012	-1.471	0.141
FXD8 (2)	0.002	0.012	0.168	0.867
FXD9 (1)	-0.003	0.012	-0.262	0.793
FXD9 (2)	0.004	0.012	0.318	0.751

## 7. T\_DOFANG

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.898	0.022	40.360	0.000
GROUP	-0.032	0.023	-1.409	0.159
GENDER	0.028	0.023	1.245	0.213
FXD3 (1)	0.020	0.011	1.769	0.077
FXD3 (2)	-0.020	0.012	-1.735	0.083
STDAGE	0.075	0.028	2.627	0.009
STDCC	-0.033	0.021	-1.539	0.124
VGEXP	0.010	0.023	0.417	0.677
STDPSVV	-0.034	0.028	-1.212	0.225
FXD8 (1)	0.011	0.012	0.936	0.350
FXD8 (2)	-0.010	0.013	-0.778	0.436
FXD9 (1)	0.004	0.011	0.366	0.714
FXD9 (2)	-0.020	0.012	-1.639	0.101

## 8. P\_DOFANXANG

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.590	0.018	33.217	0.000
GROUP	0.001	0.020	0.054	0.957
GENDER	0.009	0.019	0.476	0.634

FXD3 (1)	-0.015	0.012	-1.272	0.203
FXD3 (2)	0.016	0.012	1.345	0.179
STDAGE	-0.001	0.021	-0.057	0.955
STDCC	-0.003	0.018	-0.159	0.874
VGEXP	0.023	0.019	1.243	0.214
STDPSVV	0.013	0.025	0.512	0.609
FXD8 (1)	0.010	0.012	0.786	0.432
FXD8 (2)	-0.015	0.012	-1.223	0.221
FXD9 (1)	-0.000	0.012	-0.005	0.996
FXD9 (2)	-0.006	0.012	-0.482	0.630

## 9. L\_DIMX

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.550	0.032	17.094	0.000
GROUP	0.019	0.037	0.517	0.605
GENDER	0.076	0.034	2.230	0.026
FXD3 (1)	-0.059	0.027	-2.218	0.027
FXD3 (2)	-0.064	0.026	-2.442	0.015
FXD4 (1)	0.111	0.047	2.338	0.019
FXD4 (2)	0.124	0.048	2.589	0.010
FXD4 (3)	-0.341	0.054	-6.335	0.000
FXD4 (4)	-0.176	0.053	-3.342	0.001
FXD4 (5)	0.071	0.048	1.463	0.144
FXD4 (6)	0.105	0.047	2.230	0.026
FXD4 (7)	0.062	0.047	1.308	0.191
VGEXP (1)	0.028	0.034	0.833	0.405
SETCAMS (1)	0.037	0.012	3.192	0.001
STDAGE	0.014	0.039	0.369	0.712
STDCC	-0.020	0.033	-0.619	0.536
STDPSVV	-0.024	0.042	-0.573	0.566
FXD10 (1)	-0.005	0.027	-0.176	0.860
FXD10 (2)	0.016	0.027	0.587	0.557
FXD11 (1)	0.006	0.027	0.227	0.820
FXD11 (2)	0.029	0.027	1.064	0.287

## 10. L\_DIMZ

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.133	0.036	3.661	0.000
GROUP	-0.012	0.051	-0.229	0.819
GENDER	-0.025	0.052	-0.470	0.638
FXD3 (1)	0.035	0.033	1.065	0.287
FXD3 (2)	-0.176	0.018	-9.698	0.000
FXD4 (1)	0.074	0.015	4.789	0.000
FXD4 (2)	0.105	0.068	1.541	0.123
FXD4 (3)	-0.087	0.078	-1.111	0.266
FXD4 (4)	-0.668	0.065	-10.287	0.000
FXD4 (5)	0.090	0.071	1.267	0.205
FXD4 (6)	0.297	0.068	4.354	0.000
FXD4 (7)	0.086	0.068	1.266	0.206
VGEXP (1)	0.015	0.012	1.336	0.182
SETCAMS (1)	-0.085	0.053	-1.599	0.110



STDAGE	-0.064	0.049	-1.300	0.194
STDCC	0.080	0.048	1.676	0.094
STDPSVV	-0.133	0.067	-1.982	0.047
FXD10(1)	0.023	0.039	0.590	0.555
FXD10(2)	0.018	0.039	0.466	0.641
FXD11(1)	-0.003	0.039	-0.071	0.943
FXD11(2)	0.030	0.039	0.766	0.444

## 11. L\_DIMY

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.248	0.037	6.716	0.000
GROUP	-0.025	0.046	-0.554	0.579
GENDER	0.036	0.047	0.760	0.448
FXD3(1)	0.023	0.027	0.828	0.407
FXD3(2)	-0.033	0.027	-1.217	0.224
FXD4(1)	-0.096	0.051	-1.881	0.060
FXD4(2)	0.175	0.049	3.571	0.000
FXD4(3)	-0.060	0.053	-1.130	0.259
FXD4(4)	-0.035	0.049	-0.709	0.478
FXD4(5)	0.029	0.051	0.565	0.572
FXD4(6)	-0.019	0.050	-0.387	0.698
FXD4(7)	0.028	0.050	0.568	0.570
VGEXP(1)	-0.003	0.039	-0.076	0.939
SETCAMS(1)	0.014	0.048	0.281	0.779
STDAGE	0.047	0.044	1.077	0.281
STDCC	-0.044	0.043	-1.038	0.299
STDPSVV	-0.110	0.063	-1.757	0.079
FXD10(1)	-0.012	0.028	-0.413	0.680
FXD10(2)	0.009	0.028	0.334	0.739
FXD11(1)	-0.051	0.027	-1.841	0.066
FXD11(2)	0.034	0.027	1.240	0.215

## DOCKING PHASE

### PTA

#### 1. L\_TTASK

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	2.167	0.022	99.282	0.000
GROUP	0.004	0.022	0.177	0.860
GENDER	-0.007	0.024	-0.313	0.754
FXD3(1)	0.078	0.012	6.646	0.000
FXD3(2)	-0.048	0.012	-4.155	0.000
VGEXP	-0.041	0.023	-1.772	0.076
SETCAMS	-0.016	0.022	-0.717	0.474
STDAGE	-0.010	0.024	-0.427	0.670
STDCC	0.035	0.023	1.519	0.129

STDPTA	-0.055	0.023	-2.352	0.019
FXD9 (1)	-0.054	0.022	-2.514	0.012
FXD9 (2)	0.022	0.022	1.024	0.306
FXD9 (3)	-0.134	0.022	-6.203	0.000
FXD9 (4)	0.033	0.022	1.505	0.132
FXD9 (5)	0.080	0.022	3.699	0.000
FXD9 (6)	0.215	0.022	9.968	0.000
FXD9 (7)	0.041	0.022	1.887	0.059
FXD10 (1)	-0.025	0.012	-2.125	0.034
FXD10 (2)	0.023	0.012	2.019	0.044
FXD11 (1)	-0.010	0.012	-0.841	0.400
FXD11 (2)	0.008	0.012	0.705	0.481

## 2. L\_TOBS

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.828	0.040	20.818	0.000
GROUP	-0.052	0.040	-1.315	0.188
GENDER	0.032	0.043	0.748	0.454
FXD3 (1)	0.082	0.012	6.814	0.000
FXD3 (2)	-0.042	0.012	-3.546	0.000
FXD4 (1)	0.077	0.021	3.735	0.000
FXD4 (2)	0.006	0.021	0.290	0.772
FXD4 (3)	0.037	0.021	1.800	0.072
FXD4 (4)	0.048	0.021	2.360	0.018
FXD4 (5)	-0.078	0.021	-3.791	0.000
FXD4 (6)	-0.004	0.021	-0.190	0.850
VGEXP (1)	-0.011	0.042	-0.256	0.798
SETCAMS (1)	0.017	0.040	0.428	0.669
STDAGE	0.129	0.044	2.929	0.003
STDCC	-0.064	0.042	-1.507	0.132
STDPTA	-0.051	0.043	-1.205	0.228
FXD10 (1)	-0.007	0.012	-0.569	0.569
FXD10 (2)	0.027	0.012	2.324	0.020
FXD11 (1)	-0.020	0.012	-1.686	0.092
FXD11 (2)	0.018	0.012	1.502	0.133

## 3. L\_TCOLL

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.297	0.037	8.009	0.000
GROUP	0.067	0.041	1.639	0.101
GENDER	-0.039	0.046	-0.849	0.396
FXD3 (1)	0.025	0.044	0.563	0.574
FXD3 (2)	-0.104	0.043	-2.412	0.016
STDAGE	-0.037	0.047	-0.796	0.426
STDCC	0.036	0.046	0.789	0.430
STDPTA	-0.141	0.048	-2.933	0.003
SETCAMS	0.041	0.042	0.974	0.330
FXD8 (1)	0.030	0.044	0.691	0.490
FXD8 (2)	-0.047	0.043	-1.075	0.283
FXD9 (1)	-0.031	0.044	-0.700	0.484
FXD9 (2)	0.017	0.043	0.384	0.701

## 5. T\_PCMOTION

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.819	0.018	45.629	0.000
GROUP	0.028	0.018	1.555	0.120
GENDER	0.035	0.020	1.803	0.071
FXD3 (1)	-0.012	0.006	-1.983	0.047
FXD3 (2)	-0.010	0.006	-1.734	0.083
VGEXP	-0.000	0.019	-0.000	1.000
SETCAMS	0.024	0.018	1.324	0.185
STDAGE	0.024	0.020	1.229	0.219
STDCC	-0.031	0.019	-1.608	0.108
STDPTA	0.001	0.019	0.056	0.956
FXD9 (1)	-0.032	0.011	-2.882	0.004
FXD9 (2)	-0.014	0.011	-1.233	0.217
FXD9 (3)	-0.034	0.011	-3.100	0.002
FXD9 (4)	-0.017	0.011	-1.589	0.112
FXD9 (5)	-0.014	0.011	-1.305	0.192
FXD9 (6)	-0.000	0.011	-0.011	0.991
FXD9 (7)	0.033	0.011	2.971	0.003
FXD10 (1)	-0.005	0.006	-0.777	0.437
FXD10 (2)	0.003	0.006	0.565	0.572
FXD11 (1)	-0.008	0.006	-1.255	0.209
FXD11 (2)	0.001	0.006	0.151	0.880

## 6. T\_DOFAX

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.263	0.018	14.345	0.000
GROUP	0.018	0.018	0.970	0.332
GENDER	-0.005	0.019	-0.251	0.802
FXD3 (1)	-0.028	0.010	-2.745	0.006
FXD3 (2)	0.048	0.010	4.803	0.000
STDAGE	-0.026	0.019	-1.376	0.169
STDPTA	-0.010	0.021	-0.498	0.618
STDCC	0.015	0.019	0.796	0.426
VGEXP	0.007	0.019	0.378	0.705
FXD8 (1)	-0.012	0.010	-1.212	0.226
FXD8 (2)	-0.003	0.010	-0.260	0.795
FXD9 (1)	-0.006	0.010	-0.635	0.525
FXD9 (2)	0.001	0.010	0.100	0.920

## 7. T\_DOFANG

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	2.484	0.025	97.743	0.000
GROUP	0.017	0.025	0.676	0.499
GENDER	-0.027	0.027	-1.013	0.311
FXD3 (1)	-0.015	0.018	-0.811	0.417

FXD3 (2)	-0.002	0.018	-0.123	0.902
STDAGE	-0.011	0.026	-0.420	0.675
STDPTA	-0.089	0.028	-3.116	0.002
STDCC	0.017	0.027	0.623	0.533
VGEXP	-0.054	0.027	-2.031	0.042
FXD8 (1)	-0.012	0.018	-0.663	0.507
FXD8 (2)	-0.003	0.018	-0.160	0.873
FXD9 (1)	-0.023	0.018	-1.279	0.201
FXD9 (2)	0.014	0.018	0.797	0.425

## 9. L\_DIMX

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.455	0.025	18.039	0.000
GROUP	0.021	0.025	0.846	0.397
GENDER	0.038	0.027	1.397	0.162
FXD3 (1)	0.089	0.022	4.059	0.000
FXD3 (2)	0.157	0.022	7.151	0.000
FXD4 (1)	-0.089	0.040	-2.225	0.026
FXD4 (2)	-0.112	0.054	-2.092	0.036
FXD4 (3)	-0.509	0.041	-12.450	0.000
FXD4 (4)	0.029	0.040	0.740	0.459
FXD4 (5)	0.025	0.043	0.588	0.557
FXD4 (6)	0.356	0.040	8.987	0.000
FXD4 (7)	0.247	0.040	6.249	0.000
VGEXP (1)	-0.052	0.026	-1.992	0.046
SETCAMS (1)	-0.007	0.025	-0.260	0.795
STDAGE	-0.013	0.027	-0.468	0.640
STDCC	0.012	0.026	0.465	0.642
STDPTA	-0.060	0.026	-2.281	0.023
FXD10 (1)	-0.051	0.022	-2.354	0.019
FXD10 (2)	0.035	0.022	1.610	0.107
FXD11 (1)	-0.030	0.022	-1.393	0.164
FXD11 (2)	0.044	0.022	2.019	0.043

## 10. L\_DIMZ

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.455	0.024	18.815	0.000
GROUP	0.044	0.023	1.888	0.059
GENDER	0.027	0.025	1.064	0.288
FXD3 (1)	-0.069	0.024	-2.845	0.004
FXD3 (2)	-0.168	0.024	-7.111	0.000
FXD4 (1)	-0.139	0.042	-3.338	0.001
FXD4 (2)	0.095	0.042	2.261	0.024
FXD4 (3)	-0.546	0.068	-8.071	0.000
FXD4 (4)	-0.110	0.048	-2.267	0.023
FXD4 (5)	-0.092	0.042	-2.161	0.031
FXD4 (6)	0.357	0.042	8.565	0.000
FXD4 (7)	0.347	0.042	8.335	0.000
VGEXP (1)	-0.033	0.024	-1.340	0.180
SETCAMS (1)	-0.008	0.023	-0.348	0.728
STDAGE	-0.003	0.026	-0.136	0.892

STDCC	0.021	0.025	0.846	0.398
STDPTA	-0.049	0.025	-2.002	0.045
FXD10(1)	-0.004	0.024	-0.147	0.883
FXD10(2)	0.012	0.023	0.520	0.603
FXD11(1)	-0.045	0.024	-1.919	0.055
FXD11(2)	0.021	0.023	0.913	0.361

### 11. L\_DIMY

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.341	0.036	9.566	0.000
GROUP	0.042	0.036	1.169	0.243
GENDER	0.022	0.039	0.565	0.572
FXD3(1)	0.104	0.023	4.464	0.000
FXD3(2)	-0.014	0.014	-1.043	0.297
FXD4(1)	-0.274	0.051	-5.432	0.000
FXD4(2)	0.123	0.044	2.762	0.006
FXD4(3)	0.033	0.044	0.746	0.456
FXD4(4)	0.115	0.045	2.581	0.010
FXD4(5)	-0.037	0.046	-0.801	0.423
FXD4(6)	0.178	0.047	3.823	0.000
FXD4(7)	0.062	0.046	1.360	0.174
VGEXP(1)	-0.052	0.037	-1.382	0.167
SETCAMS(1)	0.035	0.036	0.966	0.334
STDAGE	0.018	0.039	0.457	0.648
STDCC	0.003	0.038	0.072	0.942
STDPTA	-0.067	0.038	-1.770	0.077
FXD10(1)	-0.025	0.025	-0.994	0.320
FXD10(2)	0.035	0.024	1.437	0.151
FXD11(1)	-0.026	0.025	-1.036	0.300
FXD11(2)	-0.002	0.024	-0.091	0.927

### 13. L\_TAX

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	1.654	0.014	120.983	0.000
GROUP	0.010	0.014	0.751	0.453
GENDER	0.029	0.015	1.958	0.050
FXD3(1)	0.051	0.010	5.320	0.000
FXD3(2)	-0.047	0.009	-5.038	0.000
VGEXP	-0.027	0.014	-1.880	0.060
SETCAMS	0.010	0.014	0.719	0.472
STDAGE	0.013	0.015	0.886	0.375
STDCC	-0.005	0.014	-0.356	0.722
FXD8(1)	-0.025	0.018	-1.424	0.154
FXD8(2)	0.034	0.018	1.914	0.056
FXD8(3)	-0.163	0.018	-9.268	0.000
FXD8(4)	-0.011	0.018	-0.619	0.536
FXD8(5)	0.029	0.018	1.663	0.096
FXD8(6)	0.073	0.018	4.149	0.000
FXD8(7)	0.110	0.018	6.214	0.000
STDPTA	-0.020	0.015	-1.401	0.161
FXD10(1)	-0.014	0.010	-1.501	0.133
FXD10(2)	0.019	0.009	2.047	0.041

FXD11 (1)	-0.022	0.010	-2.268	0.023
FXD11 (2)	0.018	0.009	1.885	0.059

#### 14. L\_TANG

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	1.339	0.022	60.012	0.000
GROUP	0.043	0.022	1.928	0.054
GENDER	-0.031	0.024	-1.298	0.194
FXD3 (1)	0.091	0.020	4.573	0.000
FXD3 (2)	-0.081	0.020	-4.120	0.000
VGEXP	-0.081	0.023	-3.468	0.001
SETCAMS	-0.033	0.023	-1.442	0.149
STDAGE	-0.009	0.025	-0.385	0.700
STDCC	0.057	0.024	2.408	0.016
FXD8 (1)	-0.173	0.037	-4.691	0.000
FXD8 (2)	-0.008	0.037	-0.231	0.817
FXD8 (3)	-0.158	0.037	-4.296	0.000
FXD8 (4)	0.099	0.037	2.697	0.007
FXD8 (5)	0.199	0.037	5.385	0.000
FXD8 (6)	0.324	0.037	8.809	0.000
FXD8 (7)	0.043	0.037	1.165	0.244
STDPTA	-0.116	0.024	-4.900	0.000
FXD10 (1)	-0.041	0.020	-2.066	0.039
FXD10 (2)	0.020	0.020	1.016	0.309
FXD11 (1)	-0.008	0.020	-0.421	0.673
FXD11 (2)	0.011	0.020	0.566	0.571

#### 15. L\_AX\_OFFS

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	-1.665	0.046	-36.319	0.000
GROUP	0.021	0.046	0.468	0.640
GENDER	0.087	0.050	1.753	0.080
FXD3 (1)	0.009	0.019	0.490	0.624
FXD3 (2)	0.008	0.019	0.421	0.674
VGEXP	0.039	0.048	0.815	0.415
SETCAMS	-0.016	0.046	-0.336	0.737
STDAGE	0.017	0.051	0.335	0.738
STDCC	-0.081	0.049	-1.654	0.098
FXD8 (1)	-0.080	0.035	-2.308	0.021
FXD8 (2)	0.159	0.035	4.550	0.000
FXD8 (3)	-0.049	0.035	-1.418	0.156
FXD8 (4)	0.107	0.035	3.068	0.002
FXD8 (5)	-0.150	0.035	-4.292	0.000
FXD8 (6)	-0.035	0.035	-1.002	0.316
FXD8 (7)	0.112	0.035	3.206	0.001
STDPTA	0.022	0.049	0.448	0.654
FXD10 (1)	0.022	0.019	1.166	0.244
FXD10 (2)	0.014	0.019	0.772	0.440
FXD11 (1)	-0.007	0.019	-0.372	0.710
FXD11 (2)	-0.021	0.019	-1.121	0.262

## 16. L\_ANG\_OFFS

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.401	0.029	13.972	0.000
GROUP	0.030	0.029	1.044	0.297
GENDER	0.107	0.031	3.450	0.001
FXD3 (1)	0.020	0.020	1.001	0.317
FXD3 (2)	-0.007	0.020	-0.368	0.713
VGEXP	0.028	0.030	0.918	0.359
SETCAMS	0.047	0.029	1.618	0.106
STDAGE	0.046	0.032	1.456	0.145
STDCC	-0.094	0.030	-3.095	0.002
FXD8 (1)	-0.045	0.037	-1.205	0.228
FXD8 (2)	0.001	0.037	0.015	0.988
FXD8 (3)	-0.050	0.037	-1.330	0.184
FXD8 (4)	0.013	0.037	0.349	0.727
FXD8 (5)	0.059	0.037	1.582	0.114
FXD8 (6)	-0.024	0.037	-0.635	0.525
FXD8 (7)	0.012	0.037	0.325	0.745
STDPTA	-0.009	0.031	-0.298	0.766
FXD10 (1)	0.009	0.020	0.428	0.669
FXD10 (2)	0.007	0.020	0.364	0.716
FXD11 (1)	-0.023	0.020	-1.152	0.249
FXD11 (2)	-0.033	0.020	-1.673	0.094

## 17. L\_ROLL\_OFFS

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.608	0.059	10.364	0.000
GROUP	-0.043	0.059	-0.728	0.466
GENDER	0.167	0.063	2.642	0.008
FXD3 (1)	0.092	0.043	2.145	0.032
FXD3 (2)	-0.038	0.042	-0.900	0.368
VGEXP	-0.005	0.061	-0.082	0.935
SETCAMS	0.108	0.059	1.833	0.067
STDAGE	0.135	0.064	2.095	0.036
STDCC	-0.216	0.062	-3.490	0.000
FXD8 (1)	-0.032	0.079	-0.407	0.684
FXD8 (2)	0.203	0.079	2.573	0.010
FXD8 (3)	-0.065	0.079	-0.819	0.413
FXD8 (4)	0.117	0.079	1.484	0.138
FXD8 (5)	0.098	0.079	1.240	0.215
FXD8 (6)	0.112	0.079	1.415	0.157
FXD8 (7)	0.140	0.079	1.775	0.076
STDPTA	-0.047	0.062	-0.745	0.456
FXD10 (1)	-0.001	0.043	-0.028	0.978
FXD10 (2)	0.027	0.042	0.643	0.520
FXD11 (1)	0.026	0.043	0.610	0.542
FXD11 (2)	-0.049	0.042	-1.153	0.249

## 19. L\_TCONF

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.651	0.024	27.020	0.000
GROUP	-0.011	0.024	-0.474	0.636
GENDER	-0.039	0.026	-1.502	0.133
FXD3 (1)	0.013	0.014	0.917	0.359
FXD3 (2)	-0.008	0.014	-0.576	0.565
VGEXP	-0.050	0.025	-2.018	0.044
SETCAMS	0.049	0.024	2.026	0.043
STDAGE	0.011	0.026	0.436	0.663
STDCC	0.030	0.025	1.199	0.230
FXD8 (1)	-0.086	0.026	-3.342	0.001
FXD8 (2)	-0.014	0.026	-0.550	0.582
FXD8 (3)	0.029	0.026	1.120	0.263
FXD8 (4)	-0.045	0.026	-1.741	0.082
FXD8 (5)	0.140	0.041	3.378	0.001
FXD8 (6)	-0.015	0.026	-0.594	0.553
FXD8 (7)	-0.003	0.026	-0.114	0.909
STDPTA	-0.051	0.026	-2.007	0.045
FXD10 (1)	-0.017	0.014	-1.243	0.214
FXD10 (2)	0.036	0.014	2.521	0.012
FXD11 (1)	-0.009	0.014	-0.648	0.517
FXD11 (2)	0.008	0.014	0.557	0.578

## PSVV

### 1. L\_TTASK

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	2.172	0.024	89.404	0.000
GROUP	-0.004	0.030	-0.141	0.888
GENDER	0.012	0.031	0.387	0.699
FXD3 (1)	0.078	0.012	6.710	0.000
FXD3 (2)	-0.048	0.012	-4.130	0.000
VGEXP	-0.031	0.025	-1.214	0.225
SETCAMS	-0.012	0.032	-0.372	0.710
STDAGE	-0.001	0.029	-0.034	0.973
STDCC	0.021	0.028	0.762	0.446
FXD8 (1)	-0.054	0.022	-2.515	0.012
FXD8 (2)	0.022	0.022	1.024	0.306
FXD8 (3)	-0.134	0.022	-6.205	0.000
FXD8 (4)	0.033	0.022	1.505	0.132
FXD8 (5)	0.080	0.022	3.700	0.000
FXD8 (6)	0.215	0.022	9.970	0.000
FXD8 (7)	0.041	0.022	1.888	0.059
STDPSVV	-0.014	0.041	-0.341	0.733
FXD10 (1)	-0.027	0.012	-2.218	0.027
FXD10 (2)	0.023	0.012	1.984	0.047
FXD11 (1)	-0.011	0.012	-0.951	0.342
FXD11 (2)	0.000	0.012	0.040	0.968



## 2. L\_TOBS

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.608	0.035	17.469	0.000
GROUP	-0.040	0.043	-0.926	0.354
GENDER	0.046	0.044	1.047	0.295
FXD3 (1)	0.066	0.013	5.068	0.000
FXD3 (2)	-0.027	0.013	-2.074	0.038
VGEXP	0.002	0.037	0.059	0.953
SETCAMS	0.020	0.045	0.430	0.667
STDAGE	0.108	0.041	2.612	0.009
STDCC	-0.064	0.040	-1.601	0.109
FXD8 (1)	0.303	0.024	12.601	0.000
FXD8 (2)	0.232	0.024	9.663	0.000
FXD8 (3)	0.263	0.024	10.950	0.000
FXD8 (4)	0.275	0.024	11.428	0.000
FXD8 (5)	0.149	0.024	6.183	0.000
FXD8 (6)	-1.585	0.024	-65.908	0.000
FXD8 (7)	0.223	0.024	9.254	0.000
STDPSVV	-0.006	0.059	-0.101	0.919
FXD10 (1)	-0.011	0.013	-0.820	0.412
FXD10 (2)	0.022	0.013	1.663	0.096
FXD11 (1)	0.013	0.013	0.937	0.349
FXD11 (2)	-0.001	0.013	-0.091	0.928

## 3. L\_TCOLL

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.314	0.048	6.500	0.000
GROUP	0.031	0.054	0.563	0.574
GENDER	-0.016	0.050	-0.314	0.754
FXD3 (1)	0.023	0.042	0.552	0.581
FXD3 (2)	-0.103	0.042	-2.477	0.013
STDAGE	-0.021	0.058	-0.354	0.723
STDCC	0.012	0.049	0.239	0.811
VGEXP	0.046	0.051	0.904	0.366
STDPSVV	-0.082	0.067	-1.215	0.224
FXD8 (1)	0.009	0.044	0.198	0.843
FXD8 (2)	-0.043	0.044	-0.995	0.320
FXD9 (1)	-0.082	0.044	-1.853	0.064
FXD9 (2)	0.015	0.044	0.347	0.728

## 5. T\_PCMOTION

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.818	0.018	45.242	0.000
GROUP	0.023	0.022	1.035	0.300
GENDER	0.029	0.023	1.252	0.211

FXD3 (1)	-0.012	0.006	-1.941	0.052
FXD3 (2)	-0.010	0.006	-1.726	0.084
VGEXP	-0.002	0.019	-0.132	0.895
SETCAMS	0.018	0.024	0.764	0.445
STDAGE	0.028	0.022	1.282	0.200
STDCC	-0.026	0.021	-1.263	0.207
FXD8 (1)	-0.032	0.011	-2.883	0.004
FXD8 (2)	-0.014	0.011	-1.233	0.218
FXD8 (3)	-0.034	0.011	-3.101	0.002
FXD8 (4)	-0.017	0.011	-1.588	0.112
FXD8 (5)	-0.014	0.011	-1.304	0.192
FXD8 (6)	-0.000	0.011	-0.010	0.992
FXD8 (7)	0.033	0.011	2.974	0.003
STDPSVV	-0.011	0.031	-0.366	0.714
FXD10 (1)	-0.007	0.006	-1.080	0.280
FXD10 (2)	0.004	0.006	0.746	0.456
FXD11 (1)	-0.010	0.006	-1.576	0.115
FXD11 (2)	0.005	0.006	0.771	0.441

## 6. T\_DOFAX

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.257	0.017	15.363	0.000
GROUP	-0.003	0.019	-0.184	0.854
GENDER	-0.020	0.017	-1.156	0.248
FXD3 (1)	-0.028	0.010	-2.813	0.005
FXD3 (2)	0.048	0.010	4.983	0.000
STDAGE	-0.002	0.020	-0.097	0.923
STDCC	0.024	0.017	1.422	0.155
VGEXP	0.000	0.018	0.016	0.987
STDPSVV	-0.052	0.023	-2.246	0.025
FXD8 (1)	-0.017	0.010	-1.658	0.097
FXD8 (2)	-0.001	0.010	-0.129	0.897
FXD9 (1)	-0.019	0.010	-1.835	0.066
FXD9 (2)	0.003	0.010	0.324	0.746

## 7. T\_DOFANG

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	1.072	0.018	59.812	0.000
GROUP	0.001	0.020	0.053	0.957
GENDER	-0.003	0.019	-0.173	0.862
FXD3 (1)	-0.006	0.011	-0.541	0.588
FXD3 (2)	-0.000	0.011	-0.043	0.966
STDAGE	0.004	0.022	0.187	0.852
STDCC	-0.004	0.018	-0.240	0.810
VGEXP	-0.025	0.019	-1.326	0.185
STDPSVV	-0.025	0.025	-0.992	0.321
FXD8 (1)	-0.012	0.011	-1.089	0.276
FXD8 (2)	-0.003	0.011	-0.293	0.769
FXD9 (1)	-0.017	0.012	-1.503	0.133
FXD9 (2)	-0.002	0.012	-0.187	0.852

## 9. L\_DIMX

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.455	0.027	16.660	0.000
GROUP	0.002	0.034	0.055	0.956
GENDER	0.045	0.034	1.311	0.190
FXD3 (1)	0.091	0.022	4.169	0.000
FXD3 (2)	0.158	0.022	7.190	0.000
FXD4 (1)	-0.090	0.040	-2.243	0.025
FXD4 (2)	-0.110	0.053	-2.060	0.039
FXD4 (3)	-0.511	0.041	-12.524	0.000
FXD4 (4)	0.029	0.040	0.724	0.469
FXD4 (5)	0.028	0.043	0.664	0.507
FXD4 (6)	0.355	0.040	8.991	0.000
FXD4 (7)	0.247	0.040	6.246	0.000
VGEXP (1)	-0.048	0.028	-1.703	0.089
SETCAMS (1)	-0.016	0.035	-0.451	0.652
STDAGE	0.005	0.032	0.157	0.875
STDCC	0.006	0.031	0.179	0.858
STDPSVV	-0.036	0.046	-0.781	0.435
FXD10 (1)	-0.061	0.022	-2.733	0.006
FXD10 (2)	0.035	0.023	1.539	0.124
FXD11 (1)	-0.053	0.022	-2.367	0.018
FXD11 (2)	-0.003	0.023	-0.112	0.911

## 10. L\_DIMZ

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.453	0.025	18.170	0.000
GROUP	0.021	0.030	0.696	0.487
GENDER	0.022	0.031	0.731	0.465
FXD3 (1)	-0.068	0.024	-2.801	0.005
FXD3 (2)	-0.168	0.024	-7.103	0.000
FXD4 (1)	-0.139	0.042	-3.342	0.001
FXD4 (2)	0.095	0.042	2.259	0.024
FXD4 (3)	-0.546	0.067	-8.101	0.000
FXD4 (4)	-0.110	0.048	-2.278	0.023
FXD4 (5)	-0.092	0.042	-2.166	0.030
FXD4 (6)	0.357	0.042	8.594	0.000
FXD4 (7)	0.347	0.042	8.363	0.000
VGEXP (1)	-0.033	0.025	-1.330	0.184
SETCAMS (1)	-0.025	0.031	-0.802	0.423
STDAGE	0.014	0.029	0.479	0.632
STDCC	0.023	0.028	0.829	0.407
STDPSVV	-0.054	0.041	-1.315	0.189
FXD10 (1)	-0.017	0.025	-0.662	0.508
FXD10 (2)	0.017	0.024	0.688	0.492
FXD11 (1)	-0.063	0.024	-2.588	0.010
FXD11 (2)	0.018	0.024	0.755	0.450

### 11. L\_DIMY

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.340	0.037	9.231	0.000
GROUP	0.003	0.046	0.060	0.952
GENDER	0.011	0.046	0.244	0.807
FXD3 (1)	0.105	0.023	4.532	0.000
FXD3 (2)	-0.014	0.014	-1.014	0.311
FXD4 (1)	-0.272	0.050	-5.391	0.000
FXD4 (2)	0.122	0.044	2.760	0.006
FXD4 (3)	0.033	0.044	0.743	0.457
FXD4 (4)	0.115	0.045	2.579	0.010
FXD4 (5)	-0.038	0.046	-0.834	0.404
FXD4 (6)	0.177	0.047	3.804	0.000
FXD4 (7)	0.063	0.046	1.371	0.170
VGEXP (1)	-0.052	0.038	-1.341	0.180
SETCAMS (1)	0.005	0.048	0.100	0.920
STDAGE	0.048	0.044	1.105	0.269
STDCC	0.010	0.042	0.237	0.813
STDPSVV	-0.086	0.062	-1.375	0.169
FXD10 (1)	-0.030	0.025	-1.191	0.234
FXD10 (2)	0.036	0.025	1.447	0.148
FXD11 (1)	-0.032	0.025	-1.257	0.209
FXD11 (2)	0.006	0.025	0.216	0.829

### 13. L\_TAX

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	1.658	0.014	116.805	0.000
GROUP	0.018	0.018	1.022	0.307
GENDER	0.048	0.018	2.660	0.008
FXD3 (1)	0.052	0.010	5.463	0.000
FXD3 (2)	-0.047	0.009	-5.006	0.000
VGEXP	-0.020	0.015	-1.363	0.173
SETCAMS	0.024	0.018	1.288	0.198
STDAGE	0.010	0.017	0.618	0.537
STDCC	-0.019	0.016	-1.161	0.246
FXD8 (1)	-0.025	0.018	-1.428	0.153
FXD8 (2)	0.034	0.018	1.919	0.055
FXD8 (3)	-0.163	0.018	-9.293	0.000
FXD8 (4)	-0.011	0.018	-0.621	0.535
FXD8 (5)	0.029	0.018	1.668	0.095
FXD8 (6)	0.073	0.018	4.160	0.000
FXD8 (7)	0.110	0.018	6.230	0.000
STDPSVV	0.019	0.024	0.799	0.424
FXD10 (1)	-0.020	0.010	-2.014	0.044
FXD10 (2)	0.023	0.010	2.342	0.019
FXD11 (1)	-0.029	0.010	-2.943	0.003
FXD11 (2)	0.013	0.010	1.358	0.175

### 14. L\_TANG

Standardized

Variable	Estimate	Error	Z	p-value
INTERCEPT	1.346	0.030	44.463	0.000
GROUP	0.008	0.037	0.222	0.824
GENDER	-0.010	0.038	-0.265	0.791
FXD3 (1)	0.091	0.020	4.552	0.000
FXD3 (2)	-0.081	0.020	-4.126	0.000
VGEXP	-0.065	0.032	-2.065	0.039
SETCAMS	-0.045	0.039	-1.135	0.256
STDAGE	0.021	0.036	0.589	0.556
STDCC	0.042	0.035	1.217	0.224
FXD8 (1)	-0.172	0.037	-4.689	0.000
FXD8 (2)	-0.008	0.037	-0.230	0.818
FXD8 (3)	-0.158	0.037	-4.293	0.000
FXD8 (4)	0.099	0.037	2.697	0.007
FXD8 (5)	0.199	0.037	5.376	0.000
FXD8 (6)	0.324	0.037	8.806	0.000
FXD8 (7)	0.043	0.037	1.166	0.244
STDPSVV	-0.072	0.051	-1.408	0.159
FXD10 (1)	-0.042	0.020	-2.049	0.040
FXD10 (2)	0.019	0.020	0.948	0.343
FXD11 (1)	-0.004	0.021	-0.175	0.861
FXD11 (2)	-0.001	0.021	-0.054	0.957

### 15. L\_AX\_OFFS

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	-1.668	0.046	-36.163	0.000
GROUP	0.027	0.057	0.483	0.629
GENDER	0.083	0.059	1.412	0.158
FXD3 (1)	0.011	0.019	0.566	0.572
FXD3 (2)	0.009	0.019	0.460	0.646
VGEXP	0.035	0.048	0.714	0.475
SETCAMS	-0.014	0.060	-0.229	0.819
STDAGE	0.012	0.055	0.212	0.832
STDCC	-0.078	0.053	-1.468	0.142
FXD8 (1)	-0.080	0.035	-2.309	0.021
FXD8 (2)	0.159	0.035	4.552	0.000
FXD8 (3)	-0.049	0.035	-1.419	0.156
FXD8 (4)	0.107	0.035	3.070	0.002
FXD8 (5)	-0.150	0.035	-4.295	0.000
FXD8 (6)	-0.035	0.035	-1.003	0.316
FXD8 (7)	0.112	0.035	3.207	0.001
STDPSVV	0.016	0.078	0.198	0.843
FXD10 (1)	0.021	0.019	1.091	0.275
FXD10 (2)	0.009	0.019	0.455	0.649
FXD11 (1)	-0.010	0.019	-0.505	0.613
FXD11 (2)	-0.028	0.019	-1.421	0.155

### 16. L\_ANG\_OFFS

Variable	Estimate	Standardized Error	Z	p-value
----------	----------	--------------------	---	---------

INTERCEPT	0.392	0.028	14.244	0.000
GROUP	0.001	0.034	0.034	0.972
GENDER	0.079	0.035	2.264	0.024
FXD3 (1)	0.022	0.020	1.073	0.283
FXD3 (2)	-0.006	0.020	-0.320	0.749
VGEXP	0.016	0.029	0.573	0.567
SETCAMS	0.015	0.036	0.429	0.668
STDAGE	0.065	0.033	1.997	0.046
STDCC	-0.075	0.032	-2.368	0.018
FXD8 (1)	-0.045	0.038	-1.203	0.229
FXD8 (2)	0.001	0.038	0.015	0.988
FXD8 (3)	-0.050	0.038	-1.328	0.184
FXD8 (4)	0.013	0.038	0.349	0.727
FXD8 (5)	0.059	0.038	1.580	0.114
FXD8 (6)	-0.024	0.038	-0.634	0.526
FXD8 (7)	0.012	0.038	0.325	0.745
STDPSVV	-0.061	0.047	-1.309	0.190
FXD10 (1)	0.002	0.021	0.083	0.934
FXD10 (2)	0.002	0.021	0.119	0.905
FXD11 (1)	-0.036	0.021	-1.735	0.083
FXD11 (2)	-0.026	0.021	-1.253	0.210

### 17. L\_ROLL\_OFFS

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.610	0.060	10.222	0.000
GROUP	-0.060	0.074	-0.818	0.413
GENDER	0.172	0.075	2.291	0.022
FXD3 (1)	0.091	0.043	2.127	0.033
FXD3 (2)	-0.038	0.042	-0.910	0.363
VGEXP	0.000	0.062	0.007	0.995
SETCAMS	0.099	0.078	1.281	0.200
STDAGE	0.149	0.070	2.121	0.034
STDCC	-0.219	0.069	-3.193	0.001
FXD8 (1)	-0.032	0.079	-0.408	0.683
FXD8 (2)	0.203	0.079	2.569	0.010
FXD8 (3)	-0.065	0.079	-0.820	0.412
FXD8 (4)	0.117	0.079	1.481	0.139
FXD8 (5)	0.098	0.079	1.237	0.216
FXD8 (6)	0.112	0.079	1.412	0.158
FXD8 (7)	0.140	0.079	1.771	0.077
STDPSVV	-0.037	0.101	-0.363	0.717
FXD10 (1)	0.001	0.044	0.034	0.973
FXD10 (2)	0.022	0.043	0.516	0.606
FXD11 (1)	0.016	0.044	0.356	0.722
FXD11 (2)	-0.021	0.044	-0.477	0.633

### 19. L\_TCONF

Variable	Estimate	Standardized Error	Z	p-value
INTERCEPT	0.660	0.026	25.289	0.000
GROUP	-0.002	0.032	-0.063	0.950
GENDER	-0.002	0.033	-0.046	0.963
FXD3 (1)	0.013	0.014	0.926	0.354

FXD3 (2)	-0.007	0.014	-0.521	0.602
VGEXP	-0.035	0.027	-1.275	0.202
SETCAMS	0.072	0.034	2.135	0.033
STDAGE	0.010	0.031	0.317	0.751
STDCC	0.004	0.030	0.136	0.892
FXD8 (1)	-0.086	0.025	-3.384	0.001
FXD8 (2)	-0.015	0.025	-0.573	0.566
FXD8 (3)	0.028	0.025	1.107	0.268
FXD8 (4)	-0.045	0.025	-1.772	0.076
FXD8 (5)	0.143	0.041	3.486	0.000
FXD8 (6)	-0.016	0.025	-0.618	0.537
FXD8 (7)	-0.003	0.025	-0.135	0.893
STDPSVV	0.026	0.044	0.586	0.558
FXD10 (1)	-0.025	0.014	-1.755	0.079
FXD10 (2)	0.043	0.014	2.987	0.003
FXD11 (1)	-0.034	0.014	-2.383	0.017
FXD11 (2)	0.029	0.015	1.990	0.047

# Appendix J: Theoretical Training

## Welcome to the Space Robotic Arm Training

### Training outline:

#### How to operate a robotic arm in 10 steps

During this training, the following topics will be presented:

1. Objective of the experiment
  2. Environment configuration
  3. Introduction to the robotic arm
  4. Translation movements
  5. Rotation movements
  6. Camera views
  7. Introduction to the experiment
  8. Mission 1: Picking up a space dish
  9. Mission 2: Building a Space Station
  10. Rules to follow
- Summary

1. What will I do in this experiment?

### Experiment objectives

In this experiment you will learn how to use a virtual robotic arm to pick-up and manipulate objects.

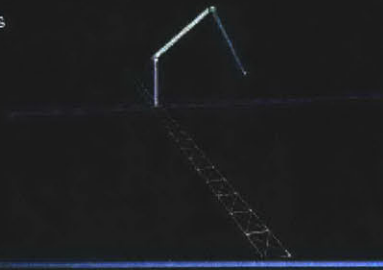
Our objective is to understand how the use of different points of view (cameras in different positions) affects your ability to operate the robotic arm.

2. What will I see in this experiment?

### Environment configuration

The space environment in this experiment is composed by the five elements:

- Structural trusses
- Robotic arm

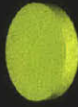




### Environment configuration

The space environment in this experiment is composed by the five elements:

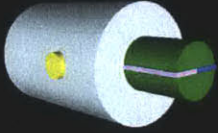
- Space dish



### Environment configuration

The space environment in this experiment is composed by the five elements:

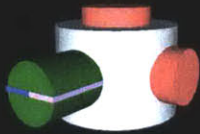
- Space Station module



### Environment configuration

The space environment in this experiment is composed by the five elements:

- Space Station node

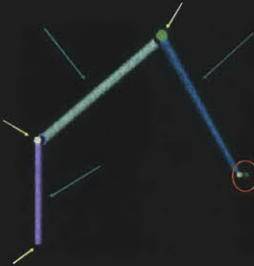


### 3. How does the robotic arm move?

### Parts of the robotic arm

The robotic arm is composed by several links and joints:

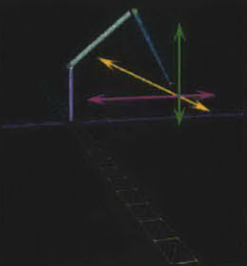
- Links are rigid "bars" that constitute the main structure of the arm.
- Joints generate a relative rotation between a link and the other.
- The **hand** is a combination of small joints and links that allows grasping and manipulating objects.



### Motion

The robotic arm has six degrees of freedom:

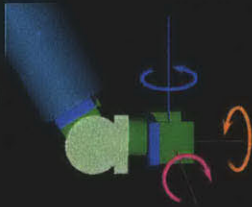
- It can translate in along the three coordinate axes
- Its hand can rotate around the three coordinate axes



## Motion

The robotic arm has six degrees of freedom:

- It can translate in along the three coordinate axes
- Its hand can rotate around the three coordinate axes



## Manipulation

Two controllers are used to manipulate the arm:

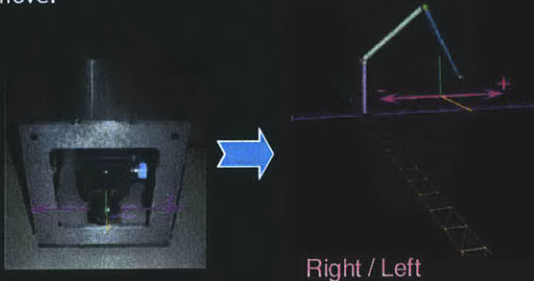
- Translational Controller
- Rotational Controller



4. How do I translate the arm?

## Translation (global reference frame)

The translational controller will cause the hand to move:



## Translation (global reference frame)

The translational controller will cause the hand to move:



### Translation (global reference frame)

The translational controller will cause the hand to move:

Forward / Backwards

### Let's practice!

- Use the **translational controller** to manipulate the robot arm that is shown in the screen next to you.
- You may come back to this presentation when you're done. (maximum 2 minutes)

### 5. How do I rotate the arm's hand?

### Rotation (local reference frame)

The rotational controller will cause the hand to:

Yaw

### Rotation (local reference frame)

The rotational controller will cause the hand to:

Roll

### Rotation (local reference frame)

The rotational controller will cause the hand to to:

### Let's practice!

- Use the **rotational controller** to manipulate the robot arm that is shown in the screen next to you.
- You may come back to this presentation when you're done. (maximum 2 minutes)

6. How will I see the arm's movements?

### Displays

Three monitors will provide you with **three** camera views, as shown below:

Camera 1

### Displays

Three monitors will provide you with **three** camera views, as shown below:

Camera 1

### Displays

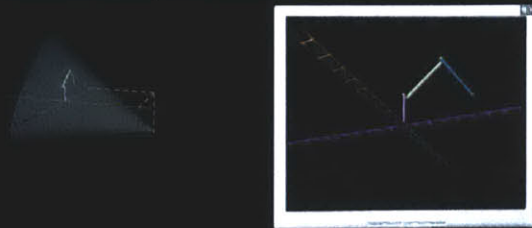
Three monitors will provide you with **three** camera views, as shown below:

Camera 2



## Displays

Three monitors will provide you with three camera views, as shown below:



Camera 3

## Camera position

Throughout the experiment, three different camera configurations will be given to you.



7. What are my missions?

## Missions

You will have two different missions in this experiment:

Mission 1: Pick up **space dishes**

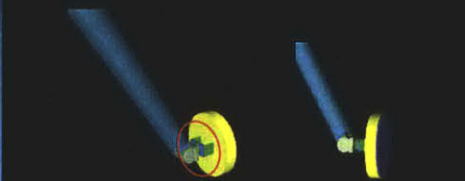
Mission 2: Build the first stage of a Space Station by attaching construction modules to the node

***You should always try to do your task as fast and accurately as possible.***

8. Tell me more about picking up space dishes...

## Mission 1: Picking up a space dish

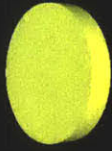
To pick up a dish you will have to bring the hand towards the pickup area, until it touches it.



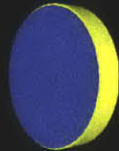
**Once contact has occurred, the target will automatically be attached to the hand.**

### Mission 1: Picking up a space dish

Space dishes must **ONLY** be picked up by their front surface (yellow).



Front face:  
Pickup area

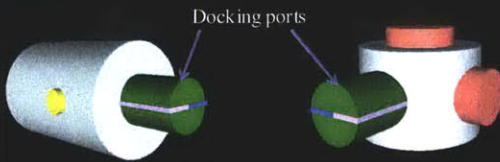


Back face:  
not to be used for pickup

9. Tell me more about building a Space Station...

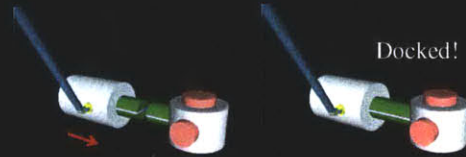
### Mission 2: Building a Space Station

Both the module and the node have **docking ports**.



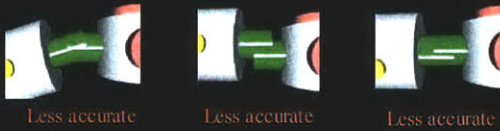
The pink and blue marks will help you to dock more accurately.

To dock the module onto the node, you will need to approach the module's port to the node's port until they **intersect**.



When you consider the space module is properly docked to the node, press the **space** bar.

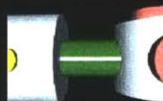
The docking ports should be as: parallel, coaxial and aligned as possible.



Less accurate

Less accurate

Less accurate



Most accurate

10. What are the task rules?

## Rule 1: Avoid collisions

You must **avoid** any kind of **undesired collisions**, such as:

- Hand-arm collision
- Dish/module -- arm/truss collision
- Module-node collision

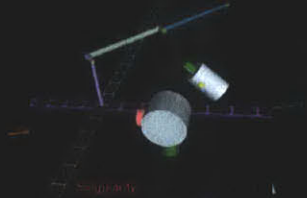


*There is no feedback regarding collisions.*

## Rule 2: Avoid arm singularities

In real life, the arm encounters singularities in the following cases (which you should avoid):

- Full arm extension
- Hand passing over the vertical axis



*If you run into a singularity, you can recover from it by reversing your latest movement, or moving away from the singular point.*

- The rules:
  1. Avoid collisions
  2. Avoid singularities
  3. Remain in the work area (when requested)
- Remember:
  - Pickup: is **automatic** (when the arm touches the space dish).
  - Docking: press the **space bar** when you consider the target and the node are acceptably (accurately) attached.
- The experiment will have the following number of trials:

Camera config.	Space dish pickup		Space Station building	
	Training	Test	Training	Test
I	→ 3	8	1	8 →
II	→ 1	8	→	8 →
III	→ 1	8	→	8 (end)

# Appendix K

## Influence of Perspective-Taking and Mental Rotation Abilities in Space Teleoperation

M. Alejandra Menchaca-Brandan, Andrew M. Liu, Charles M. Oman, and Alan Natapoff

Man-Vehicle Laboratory, Massachusetts Institute of Technology  
77 Massachusetts Avenue, Room 37-219. Cambridge, MA 02139  
+1 (617) 253-7805

menchaca@mit.edu, amliu@mit.edu, coman@mit.edu, natapoff@mit.edu

### ABSTRACT

Operator performance during Space Shuttle and International Space Station robotic arm training can differ dramatically among astronauts. The difficulty making appropriate camera selections and accurate use of hand controllers, two of the more important aspects for performance, may be rooted in a problem mentally relating the various reference frames used by the displays, hand controllers and robot arm. In this paper, we examine whether the origin of such individual differences can be found in certain components of spatial ability. We have developed a virtual reality simulation of the Space Station Robotic Workstation to investigate whether performance differences can be correlated with subjects' perspective-taking and mental rotation abilities. Spatial test scores were measured and correlated to their performance in a docking robotic task. The preliminary results show that both mental rotation strategies and perspective-taking strategies are used by the operator to move the robot arm around the workspace. Further studies must be performed to confirm such findings. If important correlations between performance and spatial abilities are found, astronaut training could be designed in order to fulfill each operator's needs, reducing both training time and cost.

### Categories and Subject Descriptors

H.1.2 [User/Machine Systems]: Human factors.

### General Terms

Performance, Experimentation, Human Factors.

### Keywords

Space teleoperation, robotic arm, perspective-taking, mental rotations, spatial ability.

## 1. INTRODUCTION

### 1.1 Space Teleoperation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HRI '07, March 9-11, 2007, Washington, DC, USA.

Copyright 2007 ACM 978-1-59593-617-2/07/0003...\$5.00.

Teleoperation is used in a wide variety of areas such as medicine, underwater exploration and space activities to perform tasks in environments that are hazardous or inaccessible to human beings. In the case of space activities, the Shuttle and International Space Station (ISS) Remote Manipulator Systems (RMS) are the main teleoperation systems currently being used. Both systems provide a set of camera viewpoints that are used by the operators to complete the task while avoiding any collisions with the surrounding structure of the Shuttle or Space Station. On the Shuttle, operators also have a direct view of the arm through a flight deck window facing onto the payload bay.



**Figure 1. ISS Robotic Workstation. The operator has only three camera viewpoints that provide visual feedback when controlling the robot arm.**

Many operational difficulties arising from the current space manipulator systems have been reported. Typically, operators can have problems determining clearance from structure because the fixed camera locations generally do not provide "optimal" views of the work space [12]. Only three camera viewpoints are seen at any moment, although more viewpoints are available for display, and the location of these cameras must be memorized by the operator. These additional views also likely increase the mental workload during operations. Thus, current procedures require a second operator to provide additional monitoring of the scene to avoid collisions during the task. Because of the danger of a collision with structure or a payload, arm movements are made at



slow velocities and after the operators have established spatial awareness of the situation. These spatial difficulties are also one reason that multi-arm operations have not been performed during a spaceflight.

## 1.2 RMS Training and Performance Assessment

NASA astronauts begin their initial teleoperation training using a generic robot arm simulation called BORIS, or the Basic Operational Robotics Instructional System. This simulation consists of a 6 degrees-of-freedom robot arm located in a cubic room and with different camera views available. As in the Shuttle and Space Station systems, the manipulation of the arm is performed using three camera views and two hand controllers. Candidates are taught to choose the ideal camera views based on clearance visualization, as well as to correctly use the hand controllers while avoiding collisions and singularities.

Teleoperation performance is evaluated by a group of instructor astronauts based on standard criteria covering all aspects of operations. Each aspect of operation is given a different weight depending on its importance to achieving mission success. The criteria given the most importance during evaluation include: spatial/visual perception (i.e., proper camera selection and real time tracking, end position and attitude correctly visualized), situational awareness (i.e., collision and singularity avoidance), and appropriate input of the controls (i.e., ability to control multi-axis movements, motion smoothness). Astronauts that do not reach the proscribed level of skill must go through additional training and practice. Often, different trainers are assigned to help, each providing their own "personal" strategies for visualizing the workspace and accomplishing the training exercises, with the hope that one of these suggestions will enable the astronaut to succeed. (J. Young, personal communication) These strategies are likely based on the spatial skills of the trainer, and perhaps when a trainer and astronaut of similar spatial skills are matched the strategies are more readily learned.

Astronauts in robotics training exhibit significant differences in their final level of performance after initial training as well as the rate at which they acquire the necessary skills. The initial level of skill during training is not a reliable predictor of a trainee's final level of performance [S. Robinson, personal communication].

## 1.3 Spatial Abilities

Spatial ability can be defined as our ability to generate, visualize, memorize, remember and transform any kind of visual information such as pictures, maps, 3D images, etc. This ability is divided into several subcomponents, which relate specifically to each of the different mental functions for image processing.

The subcomponents of spatial ability that are most relevant to teleoperation are *perspective-taking* and *mental rotations*. Perspective-taking (also known as *spatial orientation*) is the ability to imagine how an object or scene looks from perspectives different to the observer's. Mental rotations (also known as *spatial relations*), refers to the ability to mentally manipulate an array of objects. While these two abilities are logically equivalent, the critical difference lies in the coordinate frame which is manipulated to obtain the final view. Perspective-taking

requires a change in the egocentric reference frame within a fixed world coordinate frame, whereas mental rotations and spatial visualization of objects occur within a fixed egocentric reference frame. Recent work by Kozhenikov and Hegarty [10] and Hegarty and Waller [9] have shown a measurable distinction between mental rotation and perspective-taking, although performance is also highly correlated.

*Spatial visualization* is among other subcomponents of spatial ability, and is defined by Ekstrom [5] as "the ability to manipulate or transform the image of spatial patterns into other visual arrangements." Carroll [2] includes other factors such as *closure speed* (ability to rapidly access representations from long-term memory), *flexibility of closure* (ability to maintain the representation of an object in working memory while trying to distinguish it in a complex pattern), *perceptual speed* (ability to rapidly compare or find symbols or figures), and *visual memory* (ability to remember the spatial distribution of objects or figures). Recently, a separate set of factors, dynamic spatial performance, has been suggested [3, 15], where the abilities to perceive and extrapolate real motion, predict trajectories and estimate the arrival time of moving objects are assessed. These factors are not considered in this paper.

Many tests have been developed in order to measure these various factors of spatial ability. Many widely used tests such as the Cube Comparisons test (mental rotation), Paper Folding test (spatial visualization), and Spatial Orientation test can be found in the Kit of Factor-Referenced Cognitive Tests (ETS, Princeton, NJ) [6]. Among the perspective-taking tests are the Perspective-Taking Ability (PTA) Test, a computer-based test developed from the work described in [10], and the Purdue Spatial Visualizations test: Visualization of Views (PSVV), a paper-and pencil test found in [8]. A three-dimensional version of the PTA test, called the Pictures test, was used in [9] but has not been validated with a large population of subjects.

## 1.4 Spatial Abilities in Teleoperation

Indirect evidence that spatial abilities contribute to teleoperation performance comes from experiments that have manipulated the display and control reference frames during teleoperation. Lamb and Owen [11] evaluated the differences in space teleoperation performance when using egocentric (end effector) and exocentric (world) frames of reference, concluding that higher performances were obtained when an egocentric frame of reference was used. DeJong, Colgate, and Peshkin [4] also showed that reducing the number of rotations between the different reference frames can lead to improved performance. Some of the manipulations to eliminate frame rotations, such as physically moving displays, are not practical in RMS operations, because camera views can be changed during the course of the task.

The relationship between spatial abilities and teleoperation has been studied in a few previous studies. Eyal and Tendick [7] studied the effect of spatial ability when novice subjects learned to use an angled laparoscope, a medical form of teleoperation. They measured spatial ability with the Card Rotation, Paper Folding, and Perspective-Taking tests and found significant correlations of laparoscopic performance with all three measures of spatial ability. Lathan and Tracey [13] found a correlation between spatial abilities and mobile robot teleoperation

performance when navigating a maze using a single camera for visual feedback. Spatial ability was measured with four tests: the Complex Figures and Stumpf Spatial Memory Tests, which gauge spatial recognition, and the Block Rotation and Stumpf Cube Perspective Tests, which measure spatial manipulation ability. Tracey and Lathan [17] examined the effect of spatial ability on the transfer of training from simulation to a real teleoperation task and found subjects with lower spatial scores showed increased transfer of training. Spatial ability was measured by the Paper Folding tests and Stumpf's Cube Perspectives Test. All three of these tasks differ from RMS operations, in that they only provide a single view of the workspace.

We believe that both perspective-taking and mental rotation abilities play a major role in the astronaut's performance of RMS tasks. Mental rotation ability is used to understand the individual movements of the arm/payload that is seen in a single view. Perspective-taking ability is probably used when mentally imagining the different camera perspectives displayed when selecting an appropriate view of the space; it is also likely engaged when integrating the multiple viewpoints into a single representation.

This paper presents a first effort to investigate the correlation perspective-taking and mental rotation spatial abilities with teleoperation performance. By manipulating the spatial distance between camera views shown to the operators, we hope to show that operators with higher perspective-taking test scores produce better performance on our simulated task. If our hypothesis can be supported and space teleoperation performance can be predicted to some extent, astronaut training could perhaps be tailored to each astronaut's spatial needs, making the learning process more efficient.

## 2. METHODS

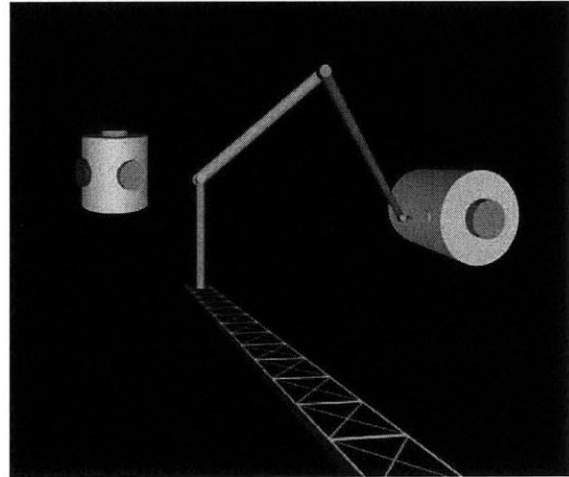
### 2.1 Environment

We created a simulated RMS workspace similar in nature to the BORIS training software using the Vizard VR development package (WorldViz, Santa Barbara, CA). It consisted of a 6 degrees-of-freedom (DOF) PUMA-like robotic arm of similar dimension to the Shuttle RMS mounted on one end of a fixed truss. The kinematics were computed using the RRG Kinematix v.4 software library (Robotics Research Group, Univ. of Texas) as a plug-in module for Vizard. A simulated cargo module and ISS node were created as the objects for grasping and docking. (Figure 2) The dynamics of the arm and other objects were not modeled in this simulation.

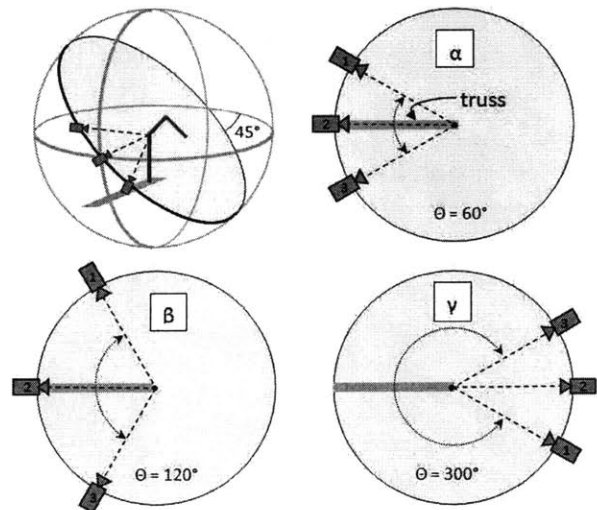
The arm and working environment could be viewed from up to three camera viewpoints at a time displayed on three separate monitors. The simulation was run on two Windows computers using Vizard's networking capabilities. The main simulation server (1.5GHz Pentium4 PC with dual head nVidia GeForce 6600 graphics) performed the kinematic calculations, graphics processing and hand controller I/O. The second computer (550MHz Pentium3 PC with nVidia GeForce3 graphics) rendered the third camera viewpoint.

The subjects controlled the robotic arm using two 3DOF joysticks in a manner similar to that used in spaceflight. A

translational hand controller was custom built in the lab from a 2DOF joystick, a linear potentiometer and USB controller card to provide three DOF control of translation. A Logitech Sidewinder 3DOF USB joystick was used as the rotational hand controller. Throughout the experiment, operators moved the end effector position using velocity control in a fixed world coordinate frame, similar to the "external frame mode" of actual RMS operations. End effector rotations were made with respect to a reference frame fixed to the end effector.



**Figure 2. A view of the simulated teleoperation environment showing the robotic arm, truss, a module near the free end of the arm and the node.**



**Figure 3. The three cameras are positioned along a circumference (black contour) tilted 45° from the horizontal plane. Camera configurations  $\alpha$ ,  $\beta$  and  $\gamma$  are defined by the angular distance between cameras 1 and 3.**

For this experiment, three configurations ( $\alpha$ ,  $\beta$ ,  $\gamma$ ) were created, each consisting of three camera views. In all three configurations, the three cameras were located at the same distance from and

pointed toward the shoulder joint of the robot arm. The central camera (#2) was placed just above the truss and was pointed in a direction along the truss. The other two cameras (#1 and #3) were placed along a line tilted 45° from the XZ plane (Figure 3). These two cameras were separated by 60° (measured from the shoulder joint) in camera configuration  $\alpha$  and 120° in camera configuration  $\beta$ . The third camera configuration,  $\gamma$ , was the same as configuration  $\alpha$  but rotated 180° about the base of the robot. This rotation produced similar camera views as configuration  $\alpha$  but with a left-right reversal of the scene.

## 2.2 Task

On each trial, subjects had to manipulate the arm to capture the module then dock it onto the node. The module was automatically captured when the end effector touched any point of the capture port (Figure 4, left). Docking the module to the node required the subject to align and overlap both docking ports and press the space bar when the final position of the module was considered to be properly aligned. The ideal docking position was defined as when both ports were coaxial and in contact. The axial separation between the docking ports was not considered to be an accuracy factor.

The initial pose of the arm as well as the position and orientation of the node was the same in each trial. The cargo module was initially located in one of four possible positions (location and orientation). Each starting position was used four times, resulting in 16 trials for one camera configuration. The four initial positions were distributed in the workspace to balance their relative locations with respect to the base of the arm (right/left, front/back, up/down).

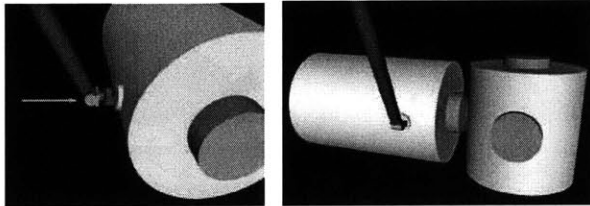


Figure 4. Capture of the module (left) and docking with the node (right)

Before starting the trials, subjects were reminded to avoid collisions between the arm and any of the structural elements in the space (e.g. node, truss) and singularities. No feedback was given to the subjects in the case of a collision and the number and type of collision was not collected at the time of the experiment. Subjects were also reminded to avoid moving the arm beyond its full extension of the arm, which would cause the simulation to crash. This condition required a restart of the simulation and the trial was repeated.

## 2.3 Spatial Ability Metrics

Spatial ability of the subjects was measured using the Cube Comparisons (CC) test, the Perspective-Taking Ability (PTA) test, and the Purdue Spatial Visualization of Views test (PSVV). Subjects were shown pairs of labeled cubes and asked if the

cubes could be identical. The test required mentally rotating one of the cubes to make the comparison. It was completed in 2 three-minute sessions, each with 21 pairs of cubes. Test scores were calculated as the number of correct answers minus incorrect answers. The PTA test was administered on a Windows PC. Subjects were shown a top-down plan view of seven objects distributed in a circular space; they were instructed to imagine they were visualizing one of those objects, and to imagine the object array from that perspective. They then had to indicate the direction to another target object in the array in their local reference frame. The test consisted of 58 trials and scores were based on direction accuracy and response time. In the PSVV test, subjects were shown a three dimensional object at the center of a “glass” cube. The task was to determine which one of five alternative views corresponded to the designated viewpoint, shown by a black dot at a specific corner of the cube. The original test is self-paced, however, a five-minute constraint was set for this experiment, in order to assign some weight to the response time, and avoid the development of strategies different to perspective-taking. The test had 30 trials and the scores were calculated as the number of correct answers minus one fifth of the number of incorrect answers.

## 2.4 Task Performance Metrics

The data acquired from each trial included instantaneous location and orientation of the end effector, time to capture ( $t_0$  coincided with the beginning of the trial), and time to dock. From these data, we calculated the following performance metrics listed in Table 1. These metrics were selected from a larger set described in [1] that characterized operator performance in the BORIS training system.

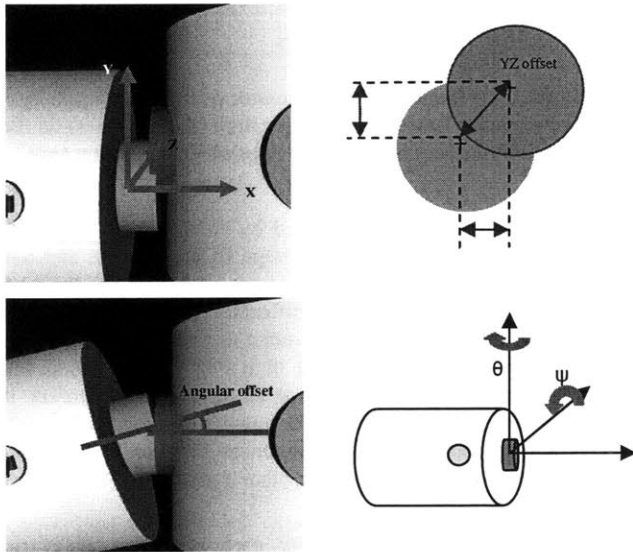
Table 1. A description of the task performance metrics used in the experiment.

Metric	Description
Observation time ( $t_{obs}$ )	Time between $t_0$ and the first hand controller input
Total time ( $t_{total}$ )	Time required to complete the task
%Motion	Percent of $t_{task}$ during which the end effector was moving
Axial DOF input ( $DOF_{Ax}$ )	Average of simultaneous use of axial degrees of freedom (DOF) during $t_{task}$ ( $DOF_{Ax}=1$ if the subject never moved on more than one axis at a time; $DOF_{Ax}=3$ if the subject moved on the three axes every time throughout the task)
Angular DOF input ( $DOF_{Ang}$ )	Average of simultaneous use of angular DOF during $t_{task}$
Docking-position offset (YZ offset)	Distance between the axes of the two docking ports (Figure 5, top)
Docking-attitude	Angle between the axes of the two docking

offset (angular offset)	ports (Figure 5, bottom)
-------------------------	--------------------------

## 2.5 Subjects

Seven subjects, three female and four male, participated in the experiment. Their ages ranged from 23 to 33, and all had an Aerospace Engineering background. On average, they spent the same daily amount of time in front of the computer (7.6 hrs) and on videogames (0.5 hrs), however their past gaming experience varied from 2 to 10 hours a week. They did not receive any monetary compensation.



**Figure 5. Performance metrics: (Top) docking position offset (YZ offset) and (Bottom) docking angular offset ( $\Theta+\Psi$ )**

## 2.6 Procedure

The experiment involved three sessions completed on separate days. During the first session (one hour maximum), subjects were given a questionnaire to obtain data such as gender, age, background, previous gaming experience, and current use of computer. Then subjects completed the Cube Comparisons test followed by the Perspective-Taking Ability test. The second session (from two and a half hours on average, four hours in one case) started with a Powerpoint presentation providing theoretical training about the main elements of the simulation (e.g. the hand controllers) and the instructions for the experiment task. After the theoretical training, the subjects completed a practical training session where they performed similar tasks to the ones in the experiment, but with a different set of objects than the module and the node. The eight training trials were designed to help the participants learn how to manipulate the arm, to capture objects, and to dock them. No feedback was provided during training. Subjects were instructed to do the task as fast and accurately as possible, avoiding any kind of collisions or singularities. Astronauts in actual robotics training are

similarly evaluated on their ability to avoid arm singularities [J. Young, personal communication].

After the training, subjects began the main experiment. A total of 48 trials was divided into three blocks of 16 trials. Within each block, only a single camera configuration was used and four repetitions of each starting location of the module were completed. The order of starting locations of the module was balanced to reduce any effects of order. The subjects were randomly divided into two groups to examine any effects of the order in which the camera configurations were seen. Group A ( $n=4$ , 2m, 2f) performed the experiment using the configurations in the order  $\alpha$ - $\beta$ - $\gamma$ , whereas Group B ( $n=3$ , 2m, 1f) followed the sequence  $\alpha$ - $\gamma$ - $\beta$ . Subjects were allowed a short break between blocks. Finally, a post-experiment questionnaire was given to assess the possible discomforts caused by the test, and to get feedback on the subjects' strategies to perform the tasks.

During a third session, subjects completed the Purdue Spatial Visualization test. This test was given in a separate session because it was added after the main experiments had been completed.

## 3. RESULTS

### 3.1 Spatial Ability Test Scores

The Cube Comparisons test (CC) scores ranged from 23 to 42 (mean: 34.43, SD: 7.18). However, these scores were clearly separable into two groups. Three subjects' scores were above the mean (40 to 42) and four subjects' scores were below (23 to 33). Interestingly, all the female subjects were in the low-scoring group while three of the four male subjects were in the high-scoring category. This result is consistent with previous findings that women's spatial visualization abilities are weaker than men's [14].

The scores for the Purdue Spatial Visualizations test ranged from 7.2 to 23.4 (mean: 17.29, SD: 6.82). These scores were bimodally distributed with two subjects scoring very low (7.2 and 7.8) and five subjects scoring above the mean.

The scores for the Perspective-Taking Ability test ranged from 20.6 to 27.4 (mean: 23.11, SD: 2.59). The scores were unimodally distributed about the mean. No obvious difference between genders was apparent for PSVV or PTA.

Two male subjects had high scores for all three tests, whereas one female scored below the mean on all three tests. No significant correlation was found between CC and PTA scores ( $R^2 = 0.022$ ), or between CC and PSVV scores ( $R^2=0.0831$ ). PTA and PSVV scores were found to be correlated with  $R^2=0.5940$ .

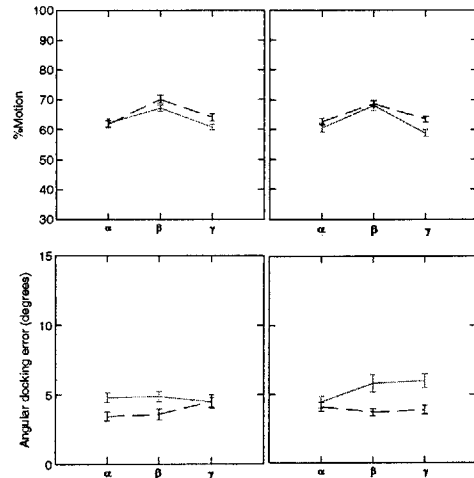
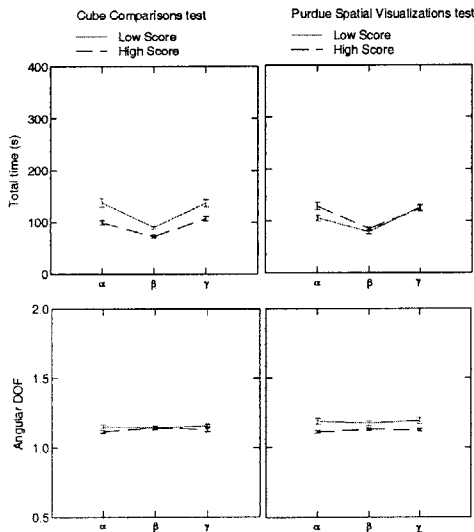
### 3.2 Task Performance Results

We used a mixed regression model (Systat v.11) to statistically analyze the relationship between spatial scores and task performance. The fixed effects considered were: test score, camera configuration ( $\alpha$ ,  $\beta$ ,  $\gamma$ ), group (A, B), the cross effect of group and camera configuration. The only random effect was subject. No effects were found to be significant due to age, daily hours of computer use nor daily hours of gaming. For the regression analysis, in order to fit the model assumptions of comparable variance and normal distribution of the residuals, the

time and accuracy data were transformed to their logarithms, the %Motion and DOF data were transformed by arcsin(sqrt()), transformation often used for percentages [18].

The data show an effect of the camera configurations on the total time for a trial ( $t_{total}$ ), observation time ( $t_{obs}$ ), %Motion and axial degrees of freedom ( $DOF_{ax}$ ). On average,  $t_{total}$  was lower while using configuration  $\beta$  (cameras 1 and 3 separated by  $120^\circ$ ) than with either configuration  $\alpha$  ( $60^\circ$  separation) or  $\gamma$  ( $60^\circ$  “left-right reversed”). This pattern is consistent for both the time to capture the module and the time to dock it to the node. The average  $t_{obs}$  needed by the subjects in both groups remained approximately constant between configurations  $\alpha$  and  $\beta$ , and increased on configuration  $\gamma$ . The high  $t_{total}$  and  $t_{obs}$  obtained for configuration  $\gamma$  was an expected result, given the left-right reversal in the camera views. %Motion and  $DOF_{ax}$  were higher for configuration  $\beta$ . The order of the camera configurations in which the subjects completed the task did not significantly affect their performance.

PTA scores did not show any effect on performance different from PSVV; this is not surprising given the narrow distribution of test scores with our limited set of subjects. The data were consequently grouped by the subjects’ CC and PSVV test scores into high scoring and low scoring groups, relative to the means. When grouped by CC scores (Figure 6, left), the average  $t_{total}$  and  $t_{obs}$  (not shown) for the high scoring group are statistically shorter than for the low scoring group across all three camera configurations ( $t_{total}$ :  $p=0.007$ ;  $t_{obs}$ :  $p=0.003$ ). Even if not significant, differences between the CC score groups are also evident for the %Motion and angular docking offset measures, although not consistently across the camera configurations. The high scoring group generally kept the arm in motion during a greater portion of the trial (configurations  $\beta$  and  $\gamma$ ) and showed better angular docking performance except in configuration  $\gamma$ . No differences between the CC score groups were evident for the remaining performance metrics.



**Figure 6. Average task performance per trial grouped by subjects scoring low (solid lines) or high (dashed lines) on the Cube Comparison and Purdue Spatial Visualization tests.**

When task performance is grouped by PSVV scores (Figure 6, right), a clear difference in the average  $DOF_{ang}$  score is apparent across the three camera configurations. Surprisingly, the high PSVV score group had a significantly lower  $DOF_{ang}$  score ( $p=0.002$ ), indicating that they tended to rotate the end effector around one axis at a time. The high PSVV score group also showed longer  $t_{total}$  in configuration  $\alpha$ , higher %Motion in configuration  $\gamma$ , and showed better angular docking performance for configurations  $\beta$  and  $\gamma$ . Differences between PSVV score groups were not evident for the remaining performance measures.

Comparing task performance for each repetition of a trial provided some insight into the learning curve for the subjects. There was a significant learning trend across repetitions, with decreasing  $t_{total}$  and  $t_{obs}$ , and increasing %Motion and  $DOF_{ax}$  ( $p=0.0005$ , all), the effect being similar for both high and low CC score groups. Low scoring subjects on the PSVV test showed a big decrease in angular accuracy, a %Motion decrease and a slight YZ offset increase in the two last trial repetitions for the last two repetitions. Performance for other metrics show very small changes over repetitions.

### 3.3 Post-experiment Questionnaire Summary

Subjects reported low levels of mental fatigue (1.8 out of 5), disorientation (1.5), and eye strain (2.5) despite the length of the experiment. They reported that translating the end effector was generally intuitive, but they often resorted to trial and error to determine how to control the rotation. A general task strategy (five subjects) was to move in only one axis at a time and mainly use the central camera (#2) for general movement control and the other two cameras for depth information and final alignment. Subjects moved the arm slowly and tried to watch all three views to avoid collisions, although two subjects reported ignoring between the module and arm.

Finally, subjects were requested to describe and sketch the location of the cameras in the three configurations. One subject could not recall any of the camera configurations. All other



subjects correctly recalled configuration  $\gamma$ , but only one subject correctly identified a difference between configurations  $\alpha$  and  $\beta$ . They also properly described the central position of camera 2. Three subjects correctly matched the right/left location of cameras #1 and #3 in configuration  $\gamma$ , and two of them in  $\alpha$ . Finally, only one subject mentioned the up/down location of cameras 1 and 3, but incorrectly recalled that both cameras were above camera 2.

#### 4. DISCUSSION AND FUTURE WORK

Our primary goal was to determine if a correlation between human spatial abilities and space teleoperation performance could be established. Our data does suggest a link between spatial ability and performance, and also provides some important insights into the separate aspects of the tasks and spatial tests. However, with only seven subjects, we must be cautious of the conclusions drawn from the study. For example, the distribution of subjects' PTA scores was markedly different from the bimodal distributions of the CC and PSVV scores. The difference could simply be due to the small number of subjects and nature of our subject population, although it could be reasonably argued that the aerospace background of the subjects is comparable to many current astronauts. Another possible explanation is based on the nature of the spatial ability being tested. The PTA test is performed in a two dimensional space, which is typical for Earth-bound navigation, but less representative of the 3D environments in space teleoperation. Since we all have lifelong experience with 2D navigation, differences in this type of perspective-taking are much smaller across the population. Subjects may utilize other learned strategies from their experience to perform the task – a common problem with spatial tests. The PSVV test, in contrast, tests perspective-taking in three dimensions, which is probably a less frequently utilized skill so scores may reflect an individual's innate spatial ability. Further experiments with a larger subject population are clearly needed.

Our data indicate that CC, PTA and PSVV test scores may be predictive of performance measured by the total and observation times, but this was not true for the docking accuracy or DOF metrics. The Perspective-Taking Ability test scores were found to be significantly correlated to the use of angular degrees-of-freedom. A trend for higher PSVV scorers to require longer total and observation times, as well as to use less angular DOF and higher %Motion was also identified. The fact that the two spatial abilities do not present significant effects on the same performance metrics is somewhat surprising since Hegarty and Waller [9] found perspective-taking and mental rotation abilities to be highly correlated, while also dissociable.

The fact that high CC test scorers required shorter total and observation times to perform the task whereas high perspective-taking ability scorers required longer times suggests that subjects with better perspective-taking ability dedicate more time to analyze the workspace from the multiple viewpoints before manipulating the robotic arm. Subjects with weaker perspective-taking ability would have more difficulty integrating the viewpoints, so like the subjects in [16], they might use only one display and simply begin moving the arm.

Subjects did not change their use of angular degrees of freedom throughout the experiment, which could be explained by their

minimal training with the rotational hand controller and resulting lack of mastery. This is supported by the fact that many subjects reported using trial and error as their main strategy to rotate the end-effector. Astronauts spend over 30 hours for basic robotic arm training but this is also not practical. Spending a preceding day solely on training, such as in [1] could be sufficient to train subjects performance to a sufficient level. It is interesting to note, however, that high perspective-taking scorers consistently used single axis rotation movements more often than the low scorers. It could reflect a trial-and-error strategy for the low PSVV score subjects, or their greater lack of proficiency controlling the arm. The measurement of arm-object collisions or direction reversals in future experiments could help distinguish these possibilities.

The different experiment camera configurations were chosen on the assumption that subjects with better perspective-taking ability would be better able to integrate highly disparate views and thus complete the task faster and more accurately, but this effect was not found. Quite possibly the camera views in configuration  $\beta$  ( $120^\circ$  separation) were not sufficiently different to force the operator to use their perspective-taking ability, so no effect was seen. Instead, operators might have simply relied on mental rotations of the objects to understand the relationship between the three displays. The suggestion is supported by the significant effect of CC test score on total time, and by the fact that subjects could not report the particular camera locations of configuration  $\beta$ . Thus the improvement in total time or %Motion performance for this camera setting was simply a learning effect. The similarity of the effect between the two CC score groups suggests that the effect of mental rotation ability was the same for all three camera configurations. For our next series of experiments, we plan to investigate camera configurations that are more widely dispersed around the working environment. An experiment examining operator performance when selecting appropriate camera views might be more likely to show a correlation of performance with perspective-taking ability tests. As mentioned in the Introduction, this is one of the criteria used to evaluate operator performance during RMS training.

Angular docking accuracy was higher for subjects with superior spatial abilities, and it decreased from configuration  $\alpha$  to configurations  $\beta$  and  $\gamma$  for poor perspective-takers. This could be explained by a speed-accuracy trade-off that also led to a decrease in %Motion score and a slight YZ offset increase in the two last trial repetitions. Results of angular docking error between configurations suggest that with camera configurations, such as  $\alpha$ , which do not provide much depth information and are relatively simple to understand, only mental rotation ability is used to perform the docking. With a configuration that provides more depth cues ( $\beta$ ), either mental rotation or perspective-taking can be utilized; finally, with a hard camera setting, such as  $\gamma$ , subjects only rely on their perspective-taking ability. Although this result need to be confirmed with a greater pool of subjects, it may imply that operators would need strength in both spatial abilities in order to guarantee high accuracy in teleoperation involving a wide set of cameras.

Further research to understand the spatial skills that underlie teleoperation task performance could be helpful in improving current training procedures for astronauts. Training programs could be individualized according to their spatial skill set, and

overall training time might even be shortened. Knowledge of the mechanisms that support spatial reasoning could also have a direct impact on improving the design of interfaces for human-robotic interaction. For example, Trafton, Cassamatis, Bugajska, Brock, Mintz and Schulz [18] applied the concept of perspective-taking to improve the interaction with autonomous robots, such as NASA's Robonaut. Understanding individual differences in spatial ability may suggest guidelines or new methods for displays that can be customized to support the spatial abilities of users, or to lead to insights that improve the training methods for robotic systems.

## 5. CONCLUSIONS

This study investigated the influence of spatial abilities on space teleoperation performance. We have identified a number of task performance metrics that seem to be correlated with our chosen measures of both mental rotation and perspective-taking ability. More specifically, the total task time and observation time were clearly correlated with mental rotation ability, but were inversely correlated with perspective-taking ability. Rotational control behavior of the end-effector also seems to change according to perspective-taking ability, with high scoring subjects exhibiting more single axis control movements. Finally, subjects with higher spatial ability seemed to keep the arm in motion for a higher percentage of time during the task, perhaps reflecting better awareness of the robotic arm in the workspace. Further studies involving a larger subject population are needed to confirm these results.

## 6. ACKNOWLEDGEMENTS

The authors thank Hiro Aoki, Jennifer Young (NASA Johnson Space Center), and NASA astronauts Jeff Hoffman, Koichi Wakata, Steve Robinson and the HRI reviewers for helpful comments. This work was supported by the National Space Biomedical Research Institute through NASA Cooperative Agreement NCC 9-58, and the National Council for Science and Technology of Mexico, CONACyT.

## 7. REFERENCES

- [1] Akagi, T.M., Schlegel, R.E., Shehab, R.L., Gilliland, K., Fry, T.L. and Hughes, Q., Towards the Construction of an Efficient Set of Robot Arm Operator Performance Metrics. In *Human Factors and Ergonomics Society 48th Annual Meeting*, (New Orleans, LA2004), 1194-1198.
- [2] Carroll, J.P. *Human cognitive abilities: a survey of factor-analytical studies*. Cambridge University Press, New York, 1993.
- [3] Contreras, M.J., Colom, R., Hernandez, J.M. and Santacreu, J. Is Static Spatial Performance Distinguishable From Dynamic Spatial Performance? A Latent-Variable Analysis. *The Journal of General Psychology*, 130, 3 (2003), 277-288.
- [4] DeJong, B.P., Colgate, J.E. and Preskin, M.A., Improving Teleoperation: Reducing Mental Rotations and Translations. In *IEEE Intl. Conf. on Robotics and Automation*, (New Orleans, LA, April 26 - May 1, 2004), 3708-3714.
- [5] Ekstrom, R.B., French, J.W. and Hartman, H.H. Cognitive Factors: Their Identification and Replication. In *MBR Monograph*, Society of Multivariate Experimental Psychology, 1979.
- [6] Ekstrom, R.B., French, J.W. and Hartman, H.H. *Manual for kit of factor referenced cognitive tests*. Educational Testing Service, Princeton, NJ, 1976.
- [7] Eyal, R. and Tendick, F. Spatial Ability and Learning the Use of an Angled Laparoscope in a Virtual Environment. In Westwood, J.D. and al., e. eds. *Medicine Meets Virtual Reality 2001*, IOS Press, Amsterdam, 2001, 146-152.
- [8] Guay, R. *Purdue Spatial Visualization Test - Visualization of Views*, Purdue Research Foundation, West Lafayette, IN, 1977.
- [9] Hegarty, M. and Waller, D. A dissociation between mental rotation and perspective-taking spatial abilities. *Intelligence*, 32, 2004 (2004), 175-191.
- [10] Kozhevnikov, M. and Hegarty, M. A dissociation between object manipulation spatial ability and spatial orientation ability. *Memory & Cognition*, 29, 5 (2001), 745-756.
- [11] Lamb, P. and Owen, D., Human Performance in Space Telerobotic Manipulation. In *ACM Symposium on Virtual Reality Software and Technology*, (Monterey, CA, November 7-9, 2005), ACM, 31-37.
- [12] Lapointe, J.F., Dupuis, E., Hartman, L. and Gillett, R., An Analysis of Low-Earth Orbit Space Operations. In *Proceedings of the Joint Association of Canadian Ergonomists/Applied Ergonomics (ACE-AE) Conference*, (Banff, Alberta, Canada, 21-23 Oct, 2002).
- [13] Lathan, C.E. and Tracey, M. The Effects of Operator Spatial Perception and Sensory Feedback on Human-Robot Teleoperation Performance. *Presence: Teleoperators and Virtual Environments*, 11, 4 (2002), 368-377.
- [14] Linn, M.C. and Petersen, A.C. Emergence and characterization of sex differences in spatial ability: a meta-analysis. *Child Development*, 56, 6 (1985), 1479-1498.
- [15] Pellegrino, J.W., Hunt, E.B., Abate, R. and Farr, S. A computer-based test battery for the assessment of static and dynamic spatial reasoning abilities. *Behavior Research Methods, Instrumentation, & Computers*, 19, 2 (1987), 231-236.
- [16] Spain, E.H. and Holzhausen, K.-P., Stereo Versus Orthogonal View Displays for Performance of a Remote Manipulator Task. In *Stereoscopic Displays and Applications II*, (1991), SPIE, 103-110.

- [17] Tracey, M.R. and Lathan, C.E. The Interaction of Spatial Ability and Motor Learning in the Transfer of Training From a Simulator to a Real Task. In Westwood, J.D. ed. *Medicine Meets Virtual Reality 2001*, IOS Press, Amsterdam, 2001, 521-527.
- [18] Trafton, J.G., Cassamatis, N.L., Bugajska, M.D., Brock, D.P., Mintz, F.E. and Schulz, A.C. Enabling Effective Human-Robot Interaction Using Perspective-Taking in Robots. *IEEE Transactions on Systems, Man, and Cybernetics, Pt. A*, 35, 4 (2005), 460-470.
- [19] Wilkinson, L., Blank, G. and Gruber, C. *Desktop Data Analysis with SYSTAT*. Prentice-Hall, Inc., Upper Saddle River, NJ, 1996.