

Bimanual Cross-Coupling in Space Telerobotics

by

Victor Wang

**BASc in Mechanical Engineering
University of British Columbia, 2010**

**Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of**

Master of Science in Mechanical Engineering

**at the
Massachusetts Institute of Technology**

February 2012

© Massachusetts Institute of Technology 2012.
All rights reserved

Signature of Author

Department of Mechanical Engineering
January 26, 2012

Certified by

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

Certified by

Neville J. Hogan
Professor, Department of Mechanical Engineering
Faculty Reader

Accepted by

David E. Hardt
Chairman, Committee on Graduate Students

Bimanual Cross-Coupling in Space Telerobotics

by

Victor Wang

Submitted to the Department of Mechanical Engineering on January 26, 2012
in partial fulfillment of the requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

Astronauts spend much time training to control the robotic arm aboard the International Space Station, and must perform a variety of challenging, three-dimensional tasks. They use a unique, bimanual control system to control the velocity of the end-effector; the left hand controls translation in three axes while the right hand simultaneously controls rotation in three axes. Operator inputs to the bimanual controllers can cross-couple through not only intermanual neuromotor pathways, when movement of one hand affects movement of the other hand, but also through intramanual pathways, when movement of one hand affects movement of the same hand in an unintended control axis.

We developed a measurement technique to quantify these directional cross-coupling pathways based on the detection of frequency-coded command signals in a bimanual tracking task. The technique allowed us to characterize the interactions among all six control axes in the form of a cross-coupling matrix of coupling strengths. An experiment using these techniques suggested two principal pathways of intermanual coupling and one of intramanual coupling. By combining information across 18 human subjects to typify the cross-coupling response due to the bimanual control system, we found that the two intermanual pathways exhibited 21% yaw to lateral translation and 15% pitch to vertical translation mean coupling even after significant training. The intramanual pathway exhibited 41% roll to yaw mean coupling.

We found significant differences in bimanual cross-coupling between subjects, and demonstrated that subjects could significantly reduce intermanual cross-coupling with practice, suggesting that these metrics may be useful indicators of control device mastery. We found statistically significant negative correlations between early-stage intramanual coupling and subsequent performance in a simulated space telerobotics track and capture task, suggesting that an intramanual coupling metric may be useful as a predictor of human telerobotic performance. The test technique could ultimately be applied to evaluate cross-coupling during astronaut training and also to reduce undesired cross-coupling through improved hand controller design. Our results supported an ergonomic basis for intermanual cross-coupling that incorporated both biomechanical effects and sensorimotor effects.

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

Acknowledgements

Thanks to Dr. Charles Oman for providing the direction and grounding wisdom for this project. Your mentorship and advice were indispensable.

Thanks to Dr. Alan Natapoff for sharing your experience in statistics, useful research insights (such as the conception of the multiplicative gap filter), interesting anecdotes, granola bars, enthusiasm and friendship.

Thanks to Dr. Andrew Liu for inviting me into the MVL, helping with many software and hardware aspects of the RWS simulator, and always being supportive.

Thanks for Dr. Neville Hogan for your ongoing educational and research support, both during the 2.151 recitations and also as my thesis reader.

Thanks to Dr. Dagmar Sternad for introducing us to your lab at Northeastern University and providing interesting research perspectives.

Thanks to Rachel Forman, Caroline Lowenthal, Raquel Galvan, Dominic Gallegos, Corine Sinnette, Erin Evans, Melanie Rueger, Dr. Steven Lockley and everybody else involved in the fatigue study who helped make those sleepless nights tolerable.

Thanks to Teresa Pontillo for providing information about your previous research, and giving me a head start on my own literature search.

Thanks to Liz Zotos for always being there and for making sure my subjects were happily paid.

Thanks to all of my subjects for your patience and enthusiasm.

Thanks to all the other present and former MVL students and postdocs who provided never-ending opportunities for fun as well as different research perspectives. You helped make my time here both enjoyable and productive.

Thanks to all my friends and other outstanding people I have had the privilege to meet, know, or work with during my time in Cambridge. Your encouragement, companionship, and inspirational excellence at whatever you do have forever enriched my life both personally and professionally.

Finally, thanks to my family. Thanks to my brother for always supporting me, and especially thanks to my parents, whose love and personal sacrifice over the years have made possible all of this and more.

This research was supported by the National Space Biomedical Research Institute through NASA Contract NCC9-58, and by a Canadian National Sciences and Engineering Research Council (NSERC) Postgraduate Scholarship to the author.

Table of Contents

Abstract.....	3
Acknowledgements.....	4
Table of Contents.....	5
List of Equations	8
List of Figures	8
List of Tables	8
1 Introduction.....	9
2 Background	11
2.1 Intermanual Cross-Coupling Phenomena.....	11
2.1.1 Alternatives to a Neuromotor Cross-Coupling Pathway.....	13
2.2 Space Telerobotics.....	13
2.2.1 Overview of Equipment and Training.....	13
2.2.2 Bimanual Cross-Coupling Issues.....	14
2.3 Bimanual Cross-Coupling Measurement Prior Research.....	16
2.3.1 Task Specification.....	17
2.3.2 Input Methodology	19
2.3.3 Output Methodology.....	20
2.3.4 Cross-Coupling Analysis Techniques.....	20
3 Specific Research Goals.....	25
3.1 Motivation	25
3.2 Key Hypotheses	26
3.2.1 Stability.....	26
3.2.2 Validity.....	26
3.2.3 Utility.....	27
4 Bimanual Cross-Coupling Test for Space Telerobotics	28
4.1 Cross-Coupling Test Design.....	28
4.1.1 Overview.....	28
4.1.2 Task Specification.....	28
4.1.3 Input Methodology	30
4.1.4 Output Methodology.....	32
4.1.5 Cross-Coupling Analysis Techniques.....	33
4.1.6 Trial Structuring	35
4.2 Human Experiments	37
4.2.1 Ethics Statement.....	37

4.2.2	Participants.....	37
4.2.3	Apparatus, Stimuli, and Procedure.....	37
4.2.4	Protocol Overview.....	39
4.3	Validation of Metrics	39
4.4	Refinement of Metrics.....	41
5	Applications of Results.....	44
5.1	Applications in Space Telerobotics	44
5.1.1	Understanding Cross-Coupling Behavior	44
5.1.2	Reducing Cross-Coupling and Improving Manual Controller Design.....	45
5.1.3	Measuring and Predicting Space Telerobotics Performance.....	46
5.1.4	Supporting Measures of Spatial Ability.....	47
5.2	Neuromechanics and the Study of Manual Control	48
5.2.1	Causes of Intermanual Cross-Coupling.....	48
5.2.2	Dimensionless Squared Jerk as a Measure of Skill Smoothness	50
6	Conclusions & Suggestions for Future Work.....	51
7	Glossary.....	53
8	References	54
Appendix A: Hand Controller Characteristics		62
A.1	Translational Hand Controller	62
A.2	Rotational Hand Controller	63
Appendix B: TCT Drift and Control Parameters.....		64
Appendix C: Logistical Details of Human Experiment.....		65
Appendix D: Questionnaires.....		67
D.1	Pre-study Questionnaire	67
D.2	Pre-session Questionnaire	69
D.3	Post-session Questionnaire	71
D.4	Post-study Questionnaire	72
Appendix E: Spatial Ability Tests.....		74
E.1	Perspective Taking Ability (PTA) Test.....	74
E.2	Mental Rotations Test (MRT).....	74
Appendix F: Summarized Subject Data.....		81
F.1	Cross-Session Data.....	81
F.2	Per Session Data.....	85
Appendix G: Cross-Correlations of Interest		93
Appendix H: Software Code		95
H.1	Python/Vizard	95

H.2	MATLAB	113
Appendix J:	Multiplicative Gap Filter	125
J.1	Abstract	125
J.2	Introduction to FIR Filters	125
J.3	FIR Filter for Identifying Multiple Peaks of Known Separation.....	128
J.4	Multiplicative Gap Filter	129
J.5	Results	129
Appendix K:	Intermanual Cross-Coupling Cancellation	131
Appendix L:	Adaptive Visual Tracking	155

List of Equations

Equation 2.1.....	20
Equation 2.2.....	21
Equation 2.3.....	21
Equation 2.4.....	22
Equation 2.5.....	23
Equation 4.1.....	33
Equation 4.2.....	34
Equation 4.3.....	41
Equation 4.4.....	42

List of Figures

Figure 1.1 – An astronaut attached to the end effector of the SSRMS.	9
Figure 2.1 – A general model of why bimanual actions are coupled.....	11
Figure 2.2 – A simplified model of bimanual crosstalk.....	12
Figure 2.3 – Illustration of the RWS, showing the THC and RHC.....	15
Figure 4.1 – Early two-dimensional, two-object version of the BCCT.....	30
Figure 4.2 – Screenshot of the updated, single object, 3D BCCT display.....	31
Figure 4.3 – Example of time series data.....	33
Figure 4.4 – Example of the processed spectral data.....	35
Figure 4.5 – A photo of the RWS simulation environment.....	38
Figure 4.6 – Comparison of BCCT with SBCCT.....	40
Figure 4.7 – Histograms of intraday fractional differences in BCCT/PBCCT.....	41
Figure 5.1 – Pitch to vertical translation (M52st) boxplots, showing session effects.....	44
Figure 5.2 – Pitch to vertical translation (M52st) boxplots, showing subject effects.....	45

List of Tables

Table 4.1 – Structuring of commanded pairs of axes.....	36
Table 4.2 – The values of the matrix M_{avg3}	42
Table 4.3 – Correlation coefficients between the two principal intermanual coupling ratios.....	43
Table 5.1 – Correlation coefficients between TCT _{successst} and M46st.....	46
Table 5.2 – Mean \pm standard deviation of the TCT performance metrics for each session.....	47
Table 5.3 – Correlation coefficients between MRT score and important coupling ratios.....	48

1 Introduction

Pat your head and rub your stomach at the same time. If you found that too easy, switch spontaneously to rubbing your head and patting your stomach. This popular game for children is a pertinent example of the topic of this thesis. An important aspect of any bimanual task is the inherent neuromotor coordination between each hand. Often, as in the game described, this coordination manifests as undesired intermanual cross-coupling. Telerobotic control of the Space Station Remote Manipulator System (SSRMS), shown in Figure 1.1, requires simultaneous coordinated motion of both hands using a unique, asymmetric, six degree of freedom hand controller system, and intermanual cross-coupling may impact the performance of these tasks. This research project aimed to develop a methodology to allow quantitative measurement of the extent of such cross-coupling in space telerobotics, and to apply these metrics to the betterment of astronaut training, hand controller design, and our understanding of neuromotor behavior.

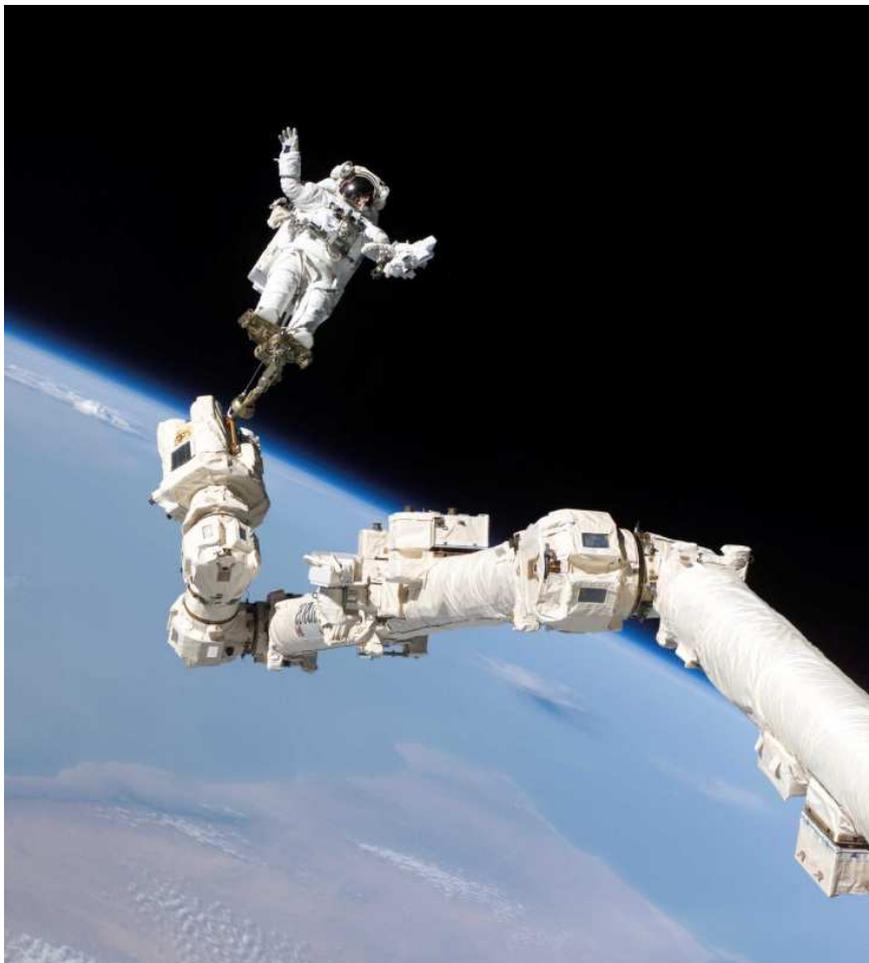


Figure 1.1 – An astronaut attached to the end effector of the SSRMS.

This thesis provides background firstly on the possible causes of intermanual cross-coupling, secondly on why this phenomenon is particularly interesting in the context of space telerobotics,

and lastly on prior research regarding the measurement of the cross-coupling. With this background and conceptual framework, we summarize our motivation to develop a bimanual cross-coupling measurement technique that is relevant to space telerobotics, describing our key hypotheses, both regarding the behavior of such a metric and its utility in practice. To provide this metric, we then detail the design and implementation of our novel Bimanual Cross-Coupling Test (BCCT). Next, we describe a series of human experiments which were conducted to validate and refine the BCCT techniques, and which also provide data to test our practical hypotheses. Using the refined BCCT techniques, we discuss the results of analyzing the experimental data with applications to astronaut training, hand controller design, and our general understanding of neuromotor phenomena. Finally, we conclude the discussion and provide suggestions for future work. In essence, we first provide the motivation and design for the BCCT, and then we analyze the results of human experiments using the BCCT.

2 Background

2.1 Intermanual Cross-Coupling Phenomena

Possible models of bimanual coordination have been studied previously in the literature. The most commonly accepted model appears to be the “intermanual crosstalk model”, in which independent motor plans are created for each limb in its contralateral brain hemisphere, and the overall behavior of each limb is influenced by mutual interaction between each motor plan [1-3]. Different studies attribute this interaction, or cross-coupling, to different factors, such as the parieto-premotor areas of the right hemisphere [3], a large system of intercerebellar relationships [4], and a subcortical locus of temporal coupling for discrete movements, with communication across the corpus callosum helping to synchronize continuous movements [5]. Most of the findings currently published appear to be in agreement with the general model in Figure 2.1, showing communication between hemispheres, the bi-hemispheric thalamocortical radiations which carry feedback signals from both hands back to the cortex, and the low-level crosstalk due to uncrossed corticospinal projections (generally, nerve fibers from the left hemisphere control the right hand, and vice versa, but 8%-15% of fibers are not crossed this way [6]).

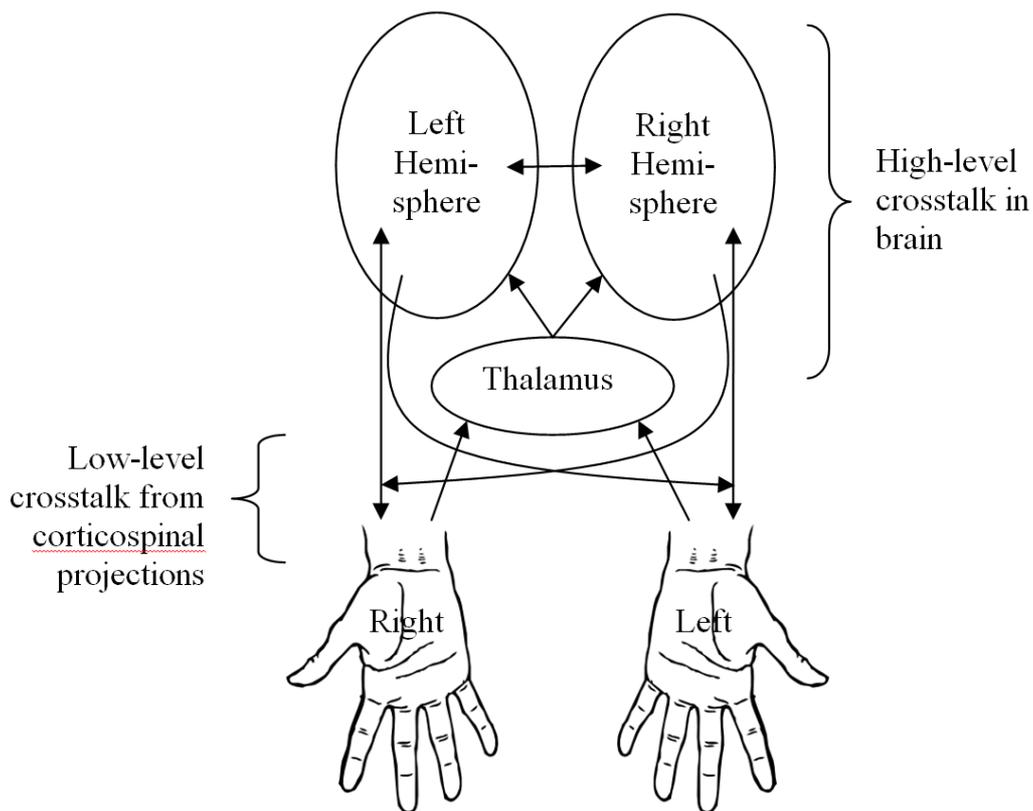


Figure 2.1 – A general model of why bimanual actions are coupled [2].

The general model can be simplified to the equivalent model in Figure 2.2. Although computational time in the brain as well as nerve signal propagation time and muscular response time likely result in variable time delays in each path, these may be essentially ignored if average coupling strength over a period of time much longer than the delay times is deemed to be representative of the magnitude of overall cross-coupling.

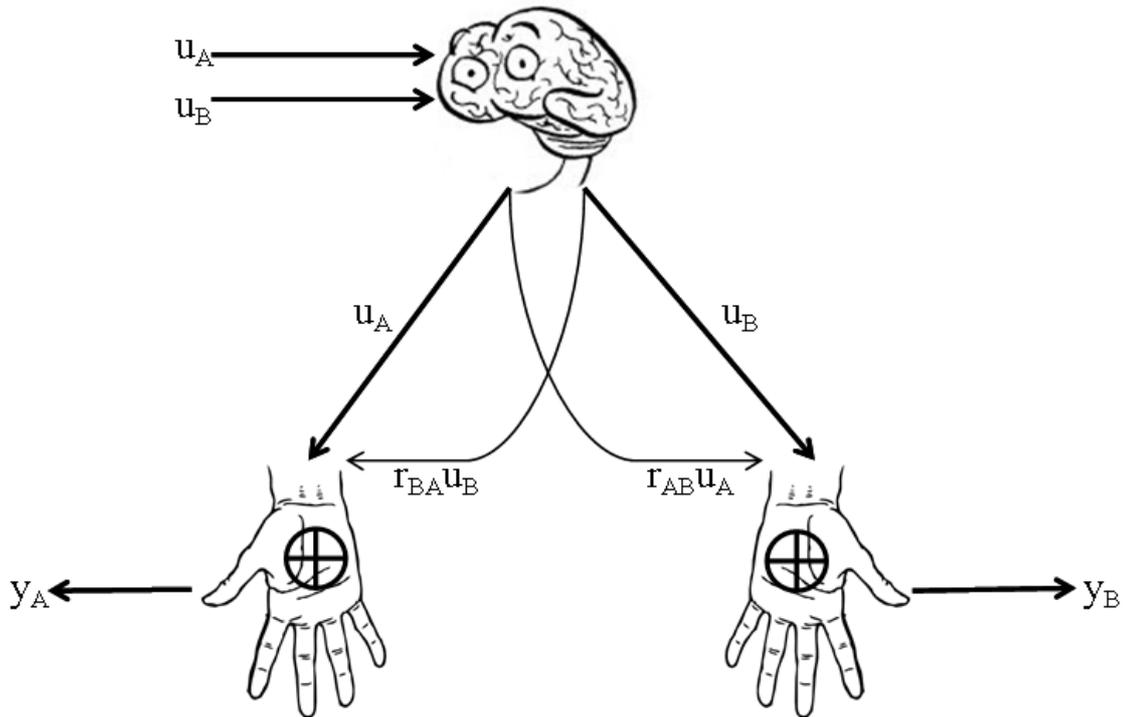


Figure 2.2 – A simplified model of bimanual crosstalk. r_{BA} and r_{AB} are the directional coupling ratios, u_A and u_B are the input commands to the human, and y_A and y_B are the output signals.

The signal weighting and connectivity is shown in terms of the directional coupling ratios, r_{AB} and r_{BA} . For example, assuming a symmetric coupling relationship such that $r = r_{AB} = r_{BA}$, if only peripheral ipsilateral pathways exist, then $r = 0$. If only peripheral pathways exist, and they are split into 80% contralateral fibers and 20% ipsilateral fibers, then $r = 0.8$. If equally weighted thalamic projections were the only connection topology, then $r = 0.5$. If callosal fibers are also included, then r will increase towards 1. [2]

However, because some aspects of bimanual behavior are computed only in the hemisphere contralateral to the dominant hand [7], because the non-dominant motor cortex is in fact suppressed during bimanual finger movements [1], and because in bimanual tasks the drive from the dominant to the non-dominant sensorimotor cortex has actually been shown to be stronger than the other way around [8], the model in Figure 2.2 reflects an allowance for unequal coupling ratios between the hands, r_{AB} and r_{BA} .

Even with this extension, it is not possible to unambiguously identify the underlying connection topology (for example, as represented generally in Figure 2.1) from knowledge of only the coupling ratios, but it will still be possible to obtain a quantitative measure of the degree of time-averaged mutual interaction, which has practical utility for training and hand controller design.

Note that this interaction occurs not only in the form of hand displacements, but also shows similar cross-coupling behavior when measured through isometric experiments where subjects produce force with no net motion [9]. Also, this interaction may be similarly present in bipedal motions, which, although fundamentally different in neurophysiology, may share some similar neural roots [10].

2.1.1 Alternatives to a Neuromotor Cross-Coupling Pathway

Although it has been shown that cross-coupling primarily emerges at the efferent level of movement planning and organization rather than the afferent level of sensing [11], research using highly asymmetric magnitudes of visual feedback from the motion of each hand has created evidence for a visually related bimanual interaction pathway [12]. To prevent this pathway from influencing the simplified model described above, it will be necessary to ensure that the magnitude of visual feedback from each hand is similar, and that visual cues to bimanual movement are as unambiguous as possible.

Another possible cross-coupling pathway that may be related to the visual pathway is through cross-coupling of right and left hand motor control learning response to error. It was recently shown in a bimanual discrete tracking task that gradual manipulation of the visual error feedback of one hand resulted in not only adaptation of the motion of that hand, but also coupled adaptation of the motion of the other hand [13].

Moreover, it is possible, depending on the experimental setup, that any bimanual interactions detected are a result of a phenomenon other than neuromotor or even visual cross-coupling. For example, a test subject's hands may be directly coupled by a mechanical linkage, possibly in the form of the subject's own torso. The distinction between terms bimanual "crosstalk" and bimanual "cross-coupling" is not always clear in the literature, but since "crosstalk" tends to imply a neuromotor interaction at the signal level, the author will from this point onwards refer to the general interaction phenomenon as "bimanual cross-coupling," and use qualifiers such as "neuromotor," "visual," or "mechanical," as appropriate. Please refer to the Glossary for details of terminology.

2.2 Space Telerobotics

2.2.1 Overview of Equipment and Training¹

Teleoperated robotic arms have been used onboard both the Space Shuttle and the International Space Station (ISS) to perform tasks such as satellite deployment, payload maintenance, repair and inspection of the Space Shuttle, and construction of the ISS. Although the Payload Deployment and Retrieval System (PDRS) aboard the Space Shuttle has been retired along with the Space Shuttle program, the Space Station Remote Manipulator System (SSRMS) is still actively used for inspection, maintenance and repair tasks aboard the ISS.

¹ Portions of this section were adapted from a 2007 research proposal titled "Advanced Displays for Efficient Training and Operation of Robotic Systems", NSBRI RFA-07001-52, CM Oman, Principal Investigator, with the permission of the authors.

The robotic arm is controlled by a human operator via a Robotic Workstation (RWS), which consists of a number of displays and two joystick hand controllers. The translational hand controller (THC) controls three axes of translational motion and the rotational hand controller (RHC) controls three axes of rotational motion. Together, these two joysticks allow six degree of freedom control over the velocity of the end effector of the robotic arm, with the corresponding arm joint motions calculated using inverse kinematics.

The time commitment for a NASA astronaut to become qualified as a robotic arm operator is significant. The first course, Generic Robotics Training (GRT), consists of fifteen lesson modules that teach basic manipulation tasks and strategies such as controlling the arm, grappling objects, and choosing appropriate camera views. After GRT, two more training courses provide specific instruction for the ISS arm since the Space Shuttle has stopped flying. After this training and once astronauts are assigned to a mission, candidates must complete flight specific training, with in-flight roles determined by performance during previous training.

Due to the extensive time and resources required to train astronauts in telerobotic tasks, any means of quantitatively validating performance or improving teaching methods would be greatly beneficial to NASA. One useful way to evaluate an astronaut's inherent ability or training progress may be to measure the astronaut's degree of bimanual cross-coupling.

2.2.2 Bimanual Cross-Coupling Issues

The cross-coupling phenomenon is particularly problematic in space telerobotics because astronauts frequently use the robotic arm to perform inspection or track and capture tasks that require simultaneous, coordinated, and smooth movements of both hands in multiple degrees of freedom. These conditions seem to be where bimanual interaction problems are most evident. Astronauts practice making coordinated movements early in their training, for example by flying the arm along a curving pathway using the end-effector camera view, keeping the camera centered on the path and the bottom of the frame aligned with the path. Although some astronauts find bimanual coordination easier than others, all require significant practice to perform well. Quantifying the amount of undesired bimanual movement would allow NASA to quantify individual differences between astronauts in bimanual control ability.

The use of bimanual cross-coupling as a performance metric or inherent ability indicator has not been investigated adequately. To evaluate operator performance, NASA currently uses subjective metrics such as situational awareness, spatial and visual perception, collision avoidance, motion smoothness, and the ability to maneuver along more than one axis at a time. Even specialized neuromotor tests for pilot or crane driver selection, such as the Vienna Test System which consists of a sensorimotor coordination test, a two-hand coordination test, and a time movement anticipation test, do not specifically measure intermanual cross-coupling [14]. In a bi-manual, two axes per hand tracking task, it was found that cross-coupling between hands tends to decrease with practice [15], so the use of an objective metric derived from or representing the aggregate amount of intermanual cross-coupling could provide useful insight into the skill development of an operator, helping in decisions to undertake either additional training or early advancement. In addition, spatial ability tests such as the Vandenberg Mental Rotation Test or the Purdue Spatial Visualization Test have been investigated as means of predicting operator performance and potential [16, 17]; while spatial ability addresses one important characteristic of an effective operator, a standardized test to measure bimanual cross-coupling may be representative of inherent neuromotor coordination ability and may therefore

provide another predictor of performance that can be useful in candidate selection. For example, experiments using transcranial stimulation (TMS) combined with electromyography (EMG) have shown that movement-related facilitation from the right premotor to left primary motor cortex (rPmd-IM1) may be predictive of performance in anti-phase bimanual movements [18].

On a more basic scientific level, it would be helpful to understand why it is difficult for humans to generate the required movements, and there are a number of issues which make space telerobotics a particularly interesting area to study bimanual cross-coupling.

2.2.2.1 Mechanical Asymmetry

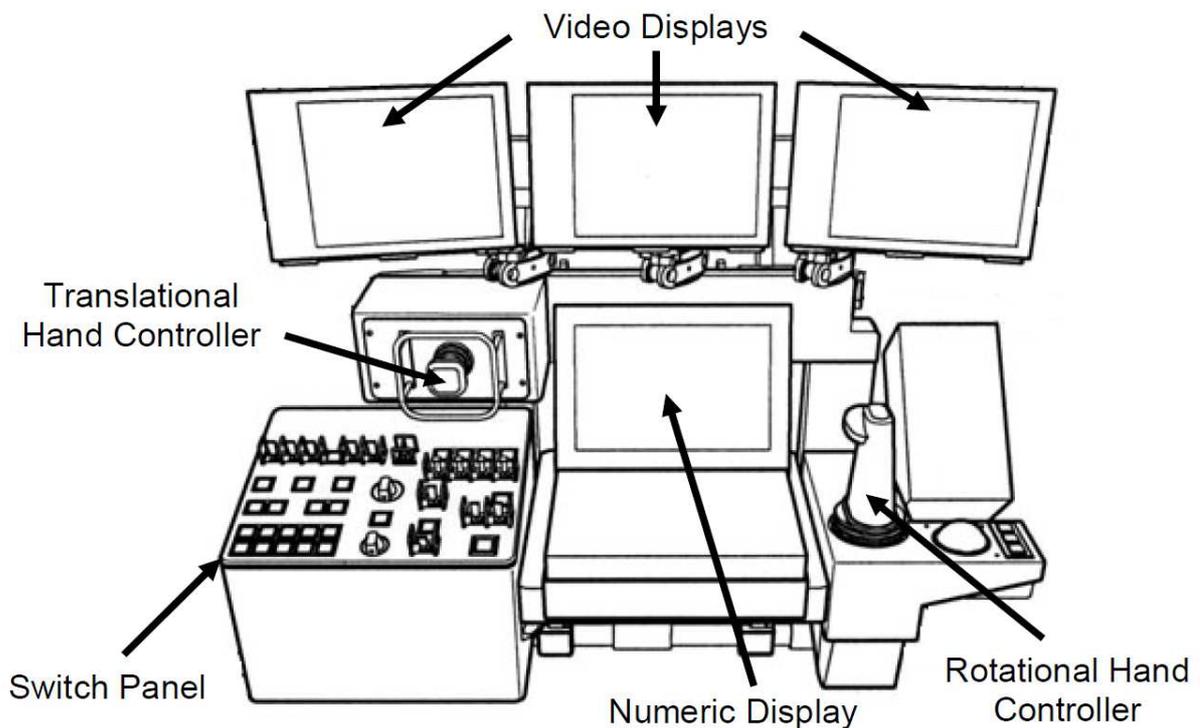


Figure 2.3 – Illustration of the RWS, showing the THC (left) and RHC (right) joysticks

While bimanual cross-coupling has been studied in the literature, the phenomenon has not been adequately studied in the context of the RWS, which has a unique mechanical configuration. Refer to Figure 2.3 for an illustration of the RWS. The RHC side resembles a typical consumer joystick one might use to control a flight simulator or similar computer game. However, its center of rotation is designed to be nominally at the center of the user's grip, such that it can rotate about this point in three degrees of freedom (roll, pitch and yaw). The THC is unique in that it consists of a hand grip on a special mount, allowing its grip point to be displaced in three orthogonal degrees of freedom (x, y, z translation). The THC is also outfitted with a hand rail to permit the negation of reaction forces caused by displacing the controller (generally, a pinky is hooked around the hand rail, "anchoring" the hand in place). Both joysticks are spring-loaded to return to a default position, and they control the robotic arm end effector through first-order (velocity) control. Together, these controllers present an interesting problem when bimanual cross-coupling is considered because it has been suggested that bimanual control should use

physically similar control devices for each hand [19], and previous studies of bimanual cross-coupling have been performed in this way [15, 20-22]. Therefore, the physically asymmetric configuration of the RWS presents a novel research area; it is not clear how movement of one hand couples with movement of the other.

2.2.2.2 Unnatural Control Paradigm

The well-established kinematic chain (KC) theory of bimanual action suggests that our hands naturally operate like a pair of motors in series, with the dominant hand conducting higher frequency, smaller amplitude movements relative to the preparatory action of the nondominant hand [23]. For example, in a right-handed person the left hand may position a block of wood while the right hand whittles, or the left hand may act as the fulcrum of a broom and control the overall position while the right hand generates the periodic sweeping action. The two hands often cooperate to control overlapping degrees of freedom, despite operating at different temporal and spatial scales. The right-hand-rotation, left-hand-translation scheme of the RWS does not fully conform to this natural bimanual behavior, and it has been shown that decomposition of motor commands into orthogonal axes (as the RWS decomposes control into independent axes of translation and rotation) is computationally demanding for the operator [24]. From observing operators during computer simulated operation of the robotic arm, it is clear that simultaneously moving and coordinating both hands is one of the most challenging aspects of the operation, and understanding the role of bimanual cross-coupling in the context of the RWS may allow improved training practices or even an improved design for the control devices.

2.2.2.3 Variable Hand Grip

Another complicating factor in the consideration of cross-coupling on the RWS is the hand grip used by the operator. Since the left and right finger representations in the motor cortex do not receive callosal fibers, they are functionally independent of one another, unlike the rest of the motor cortex, and it has also been suggested that the premotor neurons can control hand movements independently of the primary motor cortex [25]. Manual movements using only the fingers and hand may thus have different cross-coupling characteristics compared to movements involving the entire arm. It is therefore important to establish what kind of movements operators actually perform while using the RWS, and this depends on the grip used. From experience, the typical grip used results primarily in motions of the right wrist with concomitant motion of the arm for controlling the rotational joystick, while motions of the left fingers and wrist with concomitant motion of the arm control the translational joystick. If the wrist is controlled more like the fingers than like the arm, the physiological difference may not be as important because both joysticks will be controlled through the finger/wrist neural pathways. Nevertheless, it would be prudent to ensure that the hand grip and manipulation techniques used in this research are consistent with what is normally used in space telerobotics. Moreover, using a power grip versus a precision grip can also change which areas of the brain are involved in motor control and therefore cross-coupling [25], but this is not a probable complicating factor in the context of the RWS due to the low forces and high precision involved.

2.3 Bimanual Cross-Coupling Measurement Prior Research

Many techniques have been used successfully in the context of bimanual interaction and the study of neuromotor phenomena to quantify the extent and nature of cross-coupling between two signals. This section will describe a conceptual framework for categorizing the methods

according to their implementation of the four general procedural requirements of specifying the task (this affects the nominal transfer function from input to output, which in Figure 2.2 is simply a unitary constant, or pass-through), sending a human subject a set of command inputs (u_A and u_B in Figure 2.2), reading the resulting outputs from the human (y_A and y_B in Figure 2.2), and then performing some kind of analyses on the output data to characterize the coupling strengths (for example, r_{AB} and r_{BA} in Figure 2.2, although there are alternative representations).

2.3.1 Task Specification

Generally, as properties of the overall task, task specification choices will reflect both the properties of the inputs to the human and the outputs from the human.

2.3.1.1 Musculature

Due to the aforementioned evidence that different body parts may have fundamentally different neural control pathways, the physical musculature used in execution of the desired task is important. For example, a button pressing task involving only index finger motion [26] may activate different cross-coupling mechanisms compared to a bimanual reaching task involving gross arm movement. Because it is not clear how physical body parts map to neural control pathways, and in particular, where specific delineations in neural representation occur, it is also important to understand and potentially control to what extent the task is isolated to a nominal body part. For example, it is apparent that the neural pathways responsible for generating finger movements are different compared to those controlling the upper arm [27], and unless restricted in some way, either pathway could be used to position a fingertip.

2.3.1.2 Directionality

There is evidence to suggest that high-level motion planning has less to do with which body parts or muscles are involved and more to do with the perceived motion in real space, or to a higher-level congruency of goals [28]. Tasks used in bimanual cross-coupling experiments generally specify the movement directions of the task, whether it is a linear translation of each hand [20], moving the hand along a curve in a plane [29], or a simple flexion/extension of the index finger [26].

2.3.1.3 Continuity

Discrete tasks consist of individually separate and distinct motions whereas continuous tasks consist of relatively long sequences of uninterrupted motion. Certain tasks may be ambiguous in their continuity. For example, a finger tapping task may include discrete pauses when performed at low tapping frequencies, whether or not the tapping is intermittent or at a fixed frequency, but may become essentially continuous at higher frequencies. There is evidence to suggest that continuous and discrete tasks have different underlying control processes [5, 30-32].

2.3.1.4 Periodicity

To the subject, aperiodic tasks do not appear to repeat themselves and are therefore generally unpredictable by the subject, whereas periodic tasks repeat over perceivably short intervals. Periodic tasks may be continuous or discrete, as in bimanual circle drawing tasks with or without intermittent pauses [30]. Periodic motions are often called “rhythmic” in the literature [32].

2.3.1.5 Magnitude/Amplitude

It has been shown that bimanual cross-coupling can manifest itself in the magnitudes of simple point-to-point hand translations [22], and moreover the magnitude, or distance covered, by a given motion may influence its underlying neuromotor causes. For example, research on submovements shows that upper limb movement is produced by centrally generated discrete building blocks [33]. The number of submovements can vary with the magnitude of the movement, but the literature is inconclusive about the actual neuromotor cause of the phenomenon.

From a periodic point of view, it is customary to use the term “amplitude” to refer to the maximum magnitude of oscillatory motion.

2.3.1.6 Speed/Frequency

Continuous or periodic tasks may be described by their frequency content, with higher frequencies generally corresponding to faster movement. Faster movement may involve different neural control pathways compared to slow movements, which have inherently more submovements [34] and also due to implicitly different accuracy requirements may manifest through a different neuromotor control loop [35]. Note that magnitude and speed/frequency are inherently related to each other according to the well-known “Fitts’s Law,” which describes with an empirical mathematical formula the tradeoff between speed and accuracy [36] (assuming that the required accuracy is implied relative to the magnitude of the movement). A task may not necessarily specify a speed or frequency, as in a circle drawing task in which the only timing requirement is to keep the hands synchronized [37], or in the “pat your head and rub your stomach” game described in the Introduction.

2.3.1.7 Symmetry

Symmetry here refers to the degree of similarity of the tasks assigned to each hand. There are many possible measurands with which to judge symmetry or asymmetry and it is useful to group them into the same three properties of task type that we used above:

- Musculature – if the tasks assigned to each hand nominally activate homologous muscles, then the task may be considered symmetrical with respect to the musculature used. If heterologous muscles are activated however, the task would be considered asymmetric. However, it is not clear that musculature is representative of the neural correlates of motion.
- Directionality – assuming similar translational movements of each hand for example, each hand may be tasked to translate in the same real-world axis, which would be symmetric, or they may be tasked to move in orthogonal axes to each other, which would be asymmetric. Moreover it is possible for one hand to be tasked with essentially a rotation while the other hand performs a translation, which would clearly represent asymmetric directionality, in addition to asymmetric musculature. There is evidence to suggest that both musculature and directionality contribute to neural cross-coupling of bimanual signals [26].
- Continuity and Periodicity – in most experiments, both hands are similarly continuous or periodic, but this need not be the case. For example, the novel pilot experiment in bimanual cross coupling cancellation as described in the appendices (see Appendix I:

- Multiplicative Gap Filter) uses a continuous, periodic task on one hand with a discrete, aperiodic task on the other hand to introduce binary white noise as a coupled disturbance.
- Frequency – frequency content can be similar or dissimilar for each hand. Assuming a single frequency of periodic motion at each hand, it has been shown that asymmetric frequencies are much more difficult to produce, especially for certain non-multiple ratios of frequency such as 1:4.8 [29]. Assuming that each hand moves periodically and shares an identical frequency, another important characteristic of the task becomes the relative phase:
 - Phase – the hands may move either in or out of phase. Of course this depends on how the axes for each hand are defined, and the commonly accepted definition of symmetry is “mirror-symmetry” whereby each hand moves in opposite directions in real space. When this happens, movements are said to be “in-phase”, whereas movements in which the hands move at any instant in the same direction in real world coordinates are considered “anti-phase” [38]. It has been shown that mirror-symmetrical, in-phase motion is easier to perform than anti-phase motions [26], and that either of these conditions are easier to maintain than any situation in which the relative phases are different by a non-multiple of π radians [38]. This can be interpreted as supporting the hypothesis that musculature forms an important aspect of the neural representation of the motion, since it is generally the case that mirror-symmetric motions activate homologous muscles.
 - Magnitude – the movement tasked to each hand may be specified to be of similar or different magnitude [20] such that magnitude cross-coupling effects can be measured. If this is not specified, it has been shown in Down Syndrome subjects that continuity of the task tends to increase naturally chosen movement amplitude [38], and therefore, perhaps nonintuitively, magnitude may be related to the choice between continuous or discrete motion.

2.3.2 Input Methodology

The inputs to the human can be typified by the type of sensory channel or perception used and also whether a feedback input is provided. Note that although we will not consider it as a practical means to achieve our goals here, transcranial magnetic stimulation (TMS) has been used to disable brain functionality and observe the resulting impact on bimanual tasks for the purposes of discovering the neural correlates of bimanual motion [29], and could be considered an alternative, non-command input to the subject.

2.3.2.1 Sensory Channel

Command inputs to the subject can be presented through a multitude of sensory channels including visual, auditory, or tactile. Examples of visual inputs include the movement of a pair of projected dots for target tracking [15], intermittently appearing dots to indicate which finger to tap [37], or a video of drumsticks moving up and down [38]. Examples of auditory inputs include the tick of a metronome used to indicate a desired hand movement frequency, verbal cues by an examiner [38], or the sound of a drum or symbol being hit [38]. An example of tactile inputs is the gentle guiding forces provided by a pair of human assistants, one on each hand of the subject [39]. There is evidence to suggest that, at least in Down Syndrome subjects, the

sensory channel used for input and the implementation of that input can significantly impact performance on continuous tasks [38].

2.3.2.2 Feedback

The inputs to the subject can either be closed-loop, in which the results of the subject's actions are fed back to the subject, or they can be open-loop (as in Figure 2.2), in which case the subject performs so-called "feed forward" control based only on the command inputs, oblivious to the outputs.

2.3.3 Output Methodology

The outputs from the human can be typified by the signal type and also where those signals are read.

2.3.3.1 Signal Type

The output signal can either be measured as bioelectromagnetic signals through functional magnetic resonance imaging (fMRI) of the brain [29], electroencephalography (EEG) through the scalp [40], or electromyography (EMG) of specific muscle groups [40, 41], or it can be measured as a mechanical signal through force and displacement of one or more body parts. In the case of a simple button press [26], the signal may essentially be a binary representation of displacement, which is generally used with an analysis based on reaction time.

2.3.3.2 Signal Locations

While MRI is able to generate complete 3D models of brain activity, EEG and EMG require careful choice on where the electrodes are placed and therefore what is actually measured [40]. Similarly, mechanical force or displacement readings require a careful choice of the locations from which to take readings. The output signal locations generally correspond to the task musculature.

2.3.4 Cross-Coupling Analysis Techniques

Dependent on the task type chosen and the output signals available, a variety of analysis techniques may be used to quantify the extent of cross-coupling of the output signals.

2.3.4.1 Time-Series Methods

Time domain methods may include variations on the following techniques:

- Sample covariance. Calculated between a pair of time-series datasets x and y with N samples each:

$$q_{xy} \equiv \frac{1}{n-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}) \quad \text{Equation 2.1}$$

where \bar{x} and \bar{y} denote the sample mean of x and y , respectively.

A related concept is the correlation, which is normalized to unit variance and is given by

$$R_{xy} \equiv \frac{q_{xy}}{s_x s_y} \quad \text{Equation 2.2}$$

where s_x and s_y denote the sample standard deviation of x and y , respectively.

Either of these techniques will give a single metric of the extent to which the two signals vary in time with each other. However, these simplistic techniques are often poor choices due to strong sensitivity to phase uncertainty. For example, two sinusoids that are perfectly out of phase, mirror images of each other would have a covariance and correlation of zero, despite being highly related.

Multivariate correlation can be analyzed between multiple signals, and the resulting matrix of raw signal-signal correlations is sometimes referred to as “full correlation” to distinguish it from partial correlation, which is the form of that the full correlation takes after each signal-signal correlation value has been adjusted by regressing out the influence of all other signals [42]. However, despite being a superior measure of the direct, real relationship between signals, partial correlation shares a fundamental weakness with full correlation: the causality between each pair of signals remains ambiguous. Besides the fact that one cannot judge based on correlation which signal is physically influencing the other, analyses based on these techniques may be confounded by external factors which causally influence both signals.

- Time-series cross-correlation, also known as sliding dot product. The cross-correlation operator \star is defined between sampled time-series datasets x and y with N samples each as a function of the delay n

$$(x \star y)[n] \equiv \sum_{i=0}^N x^*[i]y[n+i] \quad \text{Equation 2.3}$$

where f^* denotes the complex conjugate of f . This gives a series of measures of correlation in a way that is robust to phase offsets and time delays. Cross-covariance is similarly defined.

However, despite phase robustness and insight as to which signal appears to lead in time, without a detailed model of the underlying time delays in the system, leading in time does not unambiguously determine which signal is causally influencing the other.

- Comparing magnitudes [20] or reaction times [22] at specific times corresponding to the unpredictable, discrete stimuli. Increased reaction times represent increased motor programming time and can be used to make inferences about the neural processing and pathways involved [20, 22, 26, 30]. Directional causality is intuitive to determine given reasonable assumptions regarding the neuromotor model because, for example, the tasks can be specified such that one hand is known to require more task planning time, and therefore increases the reaction time of the other hand

through coupling in the neuromotor system at the level of movement command generation.

- Other causality methods. The causality limitations of the above techniques have been noted in the literature, and although relatively new in the field of neuroscience, Time Series Interference (TSI) techniques have been developed based on the concepts of Wiener-Granger Causality (WGC) and other information-theoretic approaches [43]. Assuming a multivariate autoregressive (MVAR) model of the signals of interest, we can try to predict the next value in the time series of a signal using only known previous values of the signal itself, and if this prediction can be made more accurate by incorporating knowledge of a second signal, then that second signal is said to “G-cause” the first signal, according to WGC [43]. This theory has been implemented in a variety of ways for fMRI, EEG, and EMG data, but due to their MVAR models they are generally very sensitive to stochastic processes which are not wide-sense stationary [43] and may therefore may be difficult to apply to complex physical processes. Similar alternatives such as Dynamic Causal Modeling (DCM) have been discussed, and nonlinear variants that do not depend on an autoregressive model constitute an area of active research [43].

2.3.4.2 Signal Transforms

Raw data is generally sampled as a time-series of real numbers (for example, representing hand displacement in a certain direction), so a mathematical transformation must be used to convert the data to a spectral representation in frequency domain or other spectral analyses are desired. If we let $s(t)$ denote the neurophysiological signal of interest as a function of time t , then the time-dependent spectral representation of this signal can be represented as

$$S(f, t) \equiv |S(f, t)| \cdot e^{i\varphi(f, t)} \quad \text{Equation 2.4}$$

where the complex-valued representation consists of an amplitude value $|S(f, t)|$ and a phase value $\varphi(f, t)$, both of which may now be dependent on the frequency f [44].

Common transforms to obtain some representation of $S(f, t)$ include Hilbert, Fourier, or wavelet transforms. In the literature, it appears that Hilbert transforms are most often used to analyze the time-course of phase at a specific frequency, Fourier transforms are most often used to analyze the “time-averaged” magnitude or spectral power across a range of frequencies, and wavelets are most often used to perform time-frequency analysis.

However, depending on how they are implemented, all of these transforms can be utilized so as to be functionally and even mathematically equivalent [44].

2.3.4.3 Spectral Methods

Once the spectral representations of the signals are obtained, frequency domain and other spectral cross-coupling analysis methods may be used, including variations on the following techniques:

- Coherence, or spectral coherence. This is certainly the most traditional approach to detect cross-coupling in electrophysiological signals [45] such as EEG, and is essentially the “normalized cross-spectrum”, calculated as:

$$\kappa_{xy}(f) \equiv \frac{|S_x(f) \cdot S_y^*(f)|}{S_x(f) \cdot S_y(f)} \quad \text{Equation 2.5}$$

where the asterisk denotes complex conjugate, and the dot operator denotes the statistical expectation value, which can be obtained by averaging across time [44]. In the literature, this function is sometimes defined as the squared coherence [46], in which case the actual coherence would be essentially the square root of this function as presented here. Note that since some degree of time-averaging is implied with the Fourier transform, this would simply be equivalent to the element-by-element multiplication of a pair of frequency-series vectors containing the Fourier transformed data. Thus, coherence at any given frequency is conceptually analogous to time-series cross-correlation at any given time. Analogous to cross-correlation methods, analysis techniques based on coherence suffer from lack of specificity regarding directional causality, and also suffer from sensitivity to external confounding factors.

One example of an attempt to alleviate these external confounding factors is “task-related coherence,” which can be calculated between a pair of neurophysiological signals by subtracting the coherence at a resting state from the coherence in the active state (while performing the specified task) [40]. For example, if a bimanual hand tremor with certain frequency components is always present (if it is unrelated to the task), task-related coherence would help to remove the influence of this factor from the raw coherence measure and therefore provide a better measure of coupling caused by the specified task.

Although there is physical fMRI evidence to support this assumption of a linear superposition of task-evoked neuronal activity and ongoing spontaneous activity [47], even task-related coherence does not provide further insight into the directional causality between the observed signals, nor does it account for external confounding factors that are only present while performing the task. For example, imagine an experiment to measure bimanual movement coupling, in which the subject’s attention during the task is lost and regained at regular intervals, perhaps due to difficulty in concentrating on certain aspects of the visual command inputs, or difficulty in concentrating on both hands at once. Since attentional distribution affects manual movement magnitudes [48], attentional coupling would show up in task-related coherence, despite not strictly being a desired component of a neuromotor coupling measure.

Analogous to time-series correlation analysis, multivariate formulations of coherence have been devised, which can be decomposed into “lower order” measures of multiple and partial coherence, representing multivariate interactions among signals [46]. However these techniques remain weak in causal explanatory value.

- Phase consistency (occasionally called “imaginary coherence” [45]). By eliminating the amplitude term from the spectral representation in Equation 2.4, coherence from Equation 2.5 can be formulated with respect to phasing only [44]. This is a common treatment when using the Hilbert transform, allowing an analysis of phase coupling at any instant in time, given a pair of signals at a single frequency. It is possible to perform such an analysis at multiple frequencies by running a narrow band-pass filter on multispectral data prior to each Hilbert transform. As an alternative to phase

- consistency measures, although less common, it is possible to derive “envelope correlation” by eliminating the phase and performing some consequent manipulations involving mean shifting and normalization [44]. However, these techniques generally all present the aforementioned causality ambiguities.
- Comparing amplitudes or phases at specific frequencies. As mentioned above, the nature of the Hilbert transform has lent itself handily to the measurement of phase coupling at a specific frequency. Similar to the causality implied by time delay analysis, a phase lead can imply causality, although this is not necessarily the case. Comparing spectral phases at a specific frequency could be considered the spectral analog of comparing time series reaction times in response to a simultaneous timing stimulus. When framed this way, the spectral analog to comparing time series displacement magnitudes in response to a pair of simultaneous, different magnitude commands would be comparing spectral amplitudes (or alternatively, spectral power or absolute magnitude) at a pair of different frequencies. However, this latter technique does not appear to have been used in the literature. In fact, as we will discuss later, multiple frequencies per hand can be detected in this way by using multispectral input commands, and the uniqueness of these resulting frequency codes can allow for unambiguous specification of directional causality.
 - Analogous to the recent development of TSI in neuroscience, a formulation of Spectral WGC has been developed, along with similar techniques such as the Directed Transfer Function (DTF) and Partial Directed Coherence (PDC) [43]. In neuroscience, DTF and PDC are seeing growing use in analyzing EEG signals in order to alleviate the confounding (of any measure involving phase) by zero-phase artifacts in the form of the gradual change in electrode potentials known as “volume conduction” [49]. Interestingly, analogous to the distinction between full correlation and partial correlation in time series, DTF combines direct as well as indirect causal connections between signals, whereas PDC reveals exclusively direct connections [45], which may make DTF more useful from an explanatory point of view. However, both of these measures express the interaction between, say one signal with another signal out of a multitude of signals, as a ratio of their directional interaction relative to the first signal’s interaction with all signals [45], such that they would need to incorporate some measure of the total interaction in order to provide the coupling ratios as formulated in Figure 2.2. Moreover, both of these methods are based on an MVAR model, and suffer the associated practical issues [45].

3 Specific Research Goals

3.1 Motivation

We desire a bimanual cross-coupling test (BCCT) and a corresponding metric, or set of metrics, which quantify the nature of an individual's bimanual neuromotor coupling (as modeled in Figure 2.2) over the course of the BCCT. However, to summarize 2.1 Intermanual Cross-Coupling Phenomena, there is presently no commonly accepted neurophysiological model of bimanual cross-coupling that is capable of explaining all the properties of the phenomenon as perceived in the myriad experiments conducted over the last two to three decades. In other words, bimanual cross-coupling as we understand it is very particular to the specific context of the task, such that changing any aspect of the task could potentially change the observed cross-coupling characteristics as viewed by any kind of simplified model such that shown as in Figure 2.2. Thus, for our purposes we must tailor the BCCT to reflect the particular characteristics of space telerobotics. Referring to 2.3 Bimanual Cross-Coupling Measurement Prior Research, we should strive to carefully choose our Task Specification, Input Methodology and Output Methodology to as closely emulate the conditions of space telerobotics as possible.

This begs the question of why we should not just use an actual space telerobotic task to measure the cross-coupling. There are a number of important reasons:

- 1) Actual space telerobotic tasks are often highly ambiguous in multiple degrees of freedom and there is strong visual coupling between the inputs of each hand controller. It is therefore difficult to send a command input to a specific axis without inherently coupling that command to other axes.
- 2) Cross-coupling analysis of real telerobotic tasks would not be able to rely on any useful, known command inputs, since the tasks are free-form with respect to the movement of specific axis at specific times/frequencies. Typical space telerobotics tasks are of course more specific than the "pat your head and rub your stomach" task described in the Introduction, but are still not specific enough to easily allow the type of cross-coupling analysis we desire. It might be possible to derive some amount of information from a correlation-based analysis across axes or a nonlinear causal analysis, but the results would be ambiguous and incomplete at best, because in many tasks, the subject may choose to not even use certain axes.
- 3) If we are going to artificially design a telerobotics-like task that is visually unambiguous and also provides specific, known command inputs, then we are really designing a specialized test. We will need to consider how closely it emulates actual space telerobotic tasks, and we might as well consider the parameters of such a test from the ground up in order to gain insight into the development of effective neuromotor test design.

To decide on the cross-coupling analysis techniques, it is useful to refer to the guidelines of developing SMART (Specific, Measurable, Actionable, Relevant, and Timely) metrics [50], which are widely accepted in fields ranging from product and software development to organizational and human resources management [51-53] in order to ensure the practical utility of the metrics. In the context of our cross-coupling metrics, "Specific" suggests that we measure only undesired neuromotor cross-coupling, with separate directional coupling values between each of the six control axes, and for a task resembling space telerobotics. "Measurable" suggests that these coupling ratios must not only be quantitative measures, but must be

accurate and precise. “Actionable” suggests that the metrics are easy to understand, performance can be charted over time, and it is clear which direction is “good” (i.e. smaller value of coupling) and which direction is “bad” (i.e. larger value of coupling). “Relevant” suggests that we must be careful not to measure phenomena that are not important or are not specified by the metric; for example, we need to ensure that the coupling values do not include other coupling phenomena such as described in 2.1.1 Alternatives to a Neuromotor Cross-Coupling Pathway. “Timely” suggests that the metrics must be able to be collected for a given individual when it is needed, such that the BCCT can be performed in a reasonable amount of time and performance over time can be tracked by conducting multiple instances of the BCCT.

3.2 Key Hypotheses

Based on our knowledge of the phenomenon as described in the Background section, we can formulate a set of hypotheses about the “SMART” bimanual cross-coupling metrics. These hypotheses can be categorized by what they say about the stability (Measurable and Timely), validity (Specific and Relevant), and utility (Actionable) of the measures.

3.2.1 Stability

If the BCCT is designed properly, the coupling metrics will tend to take on repeatable and/or predictable values.

- Individuals exhibit characteristic, consistently higher or lower levels of coupling relative to other individuals, representative of some inherent neuromotor trait.
- Measures of coupling for an individual are generally repeatable within days and across days, although some variability is expected due to variations in, for example, sleep, motivation, or motor learning/retention effects.

3.2.2 Validity

The metrics represent undesired neuromotor cross-coupling phenomena.

- The metrics are greatly reduced when one hand is stationary [41], demonstrating that the measures are of a neuromotor origin.
- Specific metrics are correlated across subjects; certain axes have a higher tendency for coupling, which can be explained through physical reasoning and intuition.
 - Measures can be averaged across subjects to generate a “device coupling matrix” that objectively validates physical intuition about stronger coupling in similar motion axes and represents how humans generally use a manual control interface.
- Coupling strengths between different axes are correlated within subjects; they are aspects of a general neuromotor trait.
- The metrics increase for BCCT cases in which visual coupling is expected to be a strong component of the effect.

3.2.3 Utility

The metrics are related with other performance and ability measures in useful ways.

- Measures of higher bimanual cross-coupling are correlated with poorer performance in space telerobotics metrics, since the ability to rotate and translate independently is useful in telerobotic operations.
 - Early measures of bimanual cross-coupling can be used to predict future telerobotics performance.
- Measures of lower bimanual cross-coupling are (probably weakly) correlated with higher spatial ability scores since both may be components of general intelligence, and depending on the configuration of the BCCT, certain cross-coupling measures may be affected by visual coupling as well, which is represented by spatial ability.

4 Bimanual Cross-Coupling Test for Space Telerobotics

4.1 Cross-Coupling Test Design

4.1.1 Overview

To determine the characteristics of BCCT for space telerobotics, we will use our framework as outlined in 2.3 Bimanual Cross-Coupling Measurement Prior Research to present the rationale for each major design decision.

4.1.2 Task Specification

We seek to emulate the telerobotic task consisting of tracking a target floating in space while approaching it for capture. Called “Track and Capture,” this task is typical in space telerobotics and allows us to focus more easily on the neuromotor aspects of controlling the robot arm. We will use an experimental setup as similar to the actual track and capture operation as possible, including the use of a simulated RWS environment with a THC and RHC as described in Appendix A: Hand Controller Characteristics.

4.1.2.1 Musculature & Directionality

Because of the use of the THC and RHC, we can make sure the subjects use similar musculature as in real telerobotics by observing how users typically grip and use the hand controllers in telerobotic tasks, and then train them to use the hand controllers in the same way during the BCCT. Similarly, proper training in the use of the hand controllers will result in the directionality associated with each translation/rotation pair being similar to those in telerobotic tasks.

4.1.2.2 Continuity

The nature of the track and capture task is nominally continuous, since the floating target drifts with constant velocity in six degrees of freedom. However, due to human error and the need for occasional corrections, there is some element of discrete motion, with complete halts in motion possible. To emulate this, the cross-coupling BCCT should also involve a similar, nominally continuous tracking task.

4.1.2.3 Periodicity

The track and capture task is aperiodic in that the drift velocity of the target does not reverse at regular intervals, and the human errors that must be corrected for, though somewhat oscillatory, are generally unpredictable. We should thus make the BCCT similarly unpredictable, though it may actually oscillate with some imperceptible period.

4.1.2.4 Magnitude/Amplitude

A well-trained telerobotic operator will often have very small magnitudes of varying motion in the middle portion of the approach, after initial displacement of the hand controllers to match the

essentially steady drift of the target. However, corrections for errors that accrue can easily use the full range of displacement of the hand controllers, and these are the instances in which bimanual cross-coupling becomes an important factor. Thus, we will scale the commands during the BCCT to use most of the range of motion of the hand controllers.

4.1.2.5 Frequency

Track and capture consists of a multitude of corrective motions superimposed on a very slow motion which tracks the floating target. We characterized the typical frequency content of the total motion by averaging the frequency spectra of over 500 simulated track and capture trials with a variety of different target drift velocities (up to 90 seconds per trial, with time-series data buffered to the same length and mean zeroed) collected from a variety of sources. The results showed that the corrective oscillations of interest generally occur at low frequencies in the range of zero to about 0.2 Hz. The BCCT should thus incorporate similar frequencies.

As discussed in 2.3.4.3 Spectral Methods, most established techniques in the literature for analyzing the coupling between two signals are ambiguous about causality and thus the direction of the coupling. However, this causality is a key aspect of even our simplified cross-coupling model, and knowledge of its direction would be very helpful in practical training of telerobotics candidates. Since established MVAR-based methods appear very particular to implementation details, we choose here a novel method of representing the command inputs for each hand by a pair of superimposed sinusoids of different frequencies. This forms a special frequency-coded signal that can be unambiguously detected on another hand controller axis. According to the above analysis of typical frequencies in track and capture tasks, we choose 0.03 Hz and 0.19 Hz for the RHC and 0.07 Hz and 0.17 Hz for the THC. These frequencies are chosen such that the frequencies of interest are easy to distinguish, they are not immediately multiples of each other, and the differences between them do not form immediate multiples of any frequency of interest. As we'll see later, this careful choice is beneficial in the frequency domain analysis.

This choice of superimposed frequencies simultaneously satisfies our desire for a generally unpredictable (apparently aperiodic) task specification in order to emulate the error correction aspects of track and capture, since the superposition of the two frequencies creates a net motion that seems empirically random [54].

4.1.2.6 Symmetry

Having the same hand controllers in the BCCT as in the track and capture task ensures that the musculature used and the directionality are similarly asymmetric, with the degree of asymmetry depending on which translation and rotation pair is being considered.

In track and capture, because each hand performs a similar tracking task in each axis, both hands are essentially symmetric in continuity and periodicity. Assuming roughly equal amounts of translation and rotation insofar as corrective actions utilize the full range of motion of each hand controller, the magnitudes are also symmetric. The BCCT should correspond to the track and capture task in all of these respects.

4.1.3 Input Methodology

Our choice of input methodology is governed by our desire to emulate track and capture as closely as possible.

4.1.3.1 Sensory Channel

Track and capture command inputs to the subject are represented by the tracking error as shown on a computer screen, which displays a two-dimensional camera projection of a three-dimensional environment. The BCCT will similarly provide command inputs to the subject in the form of a visual tracking task. However, the exact implementation of the visual display requires careful consideration.

As shown in Figure 4.1, the first iteration of the BCCT used a two-dimensional, two-object visual representation to avoid ambiguities in specifying target position and orientation. All BCCT software was developed using the WorldViz Vizard integrated development environment built on top of the Python programming language.

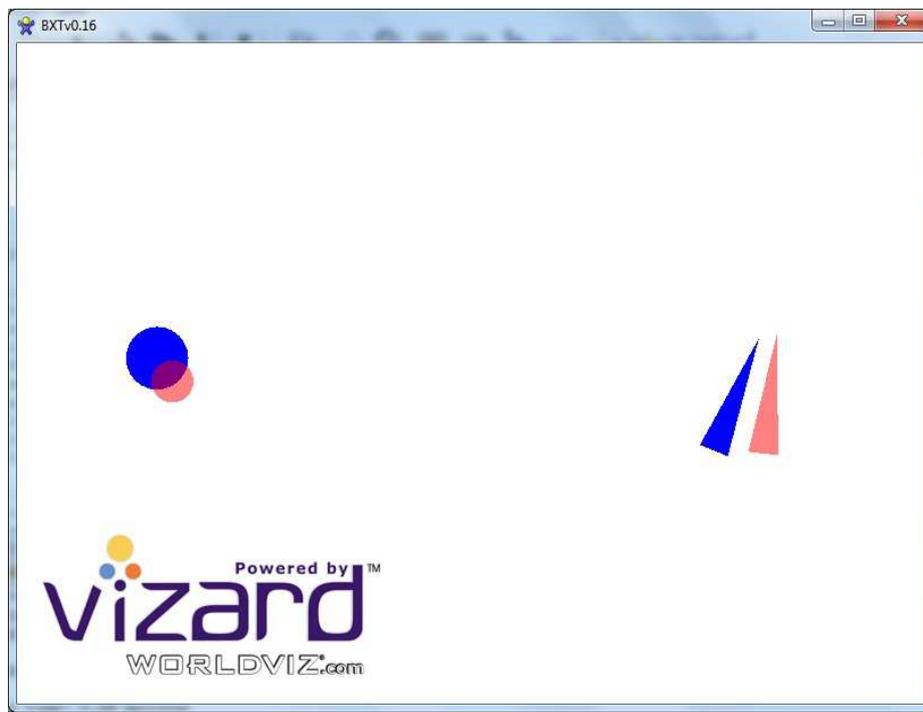


Figure 4.1 – Early two-dimensional, two-object version of the BCCT. The operator is tasked to track the motion of a pair of blue targets on a computer screen using a pair of corresponding red cursors. The circular target on the left side translates in a circular trajectory and scales in size. To track these fluctuations, the operator must move the THC in three corresponding axes. The triangular target on the right side translates in a circular trajectory and rotates about its center. To track these fluctuations, the operator must move the RHC in three corresponding axes.

Unfortunately, although there was no visual ambiguity per se, the separation of the command input into two objects resulted in difficulty focusing attention on both targets simultaneously.

Thus, movements tended to be jerky and non-continuous as the operator's gaze switched between two locations on the computer screen. Since this behavior seemed empirically different compared to what is experienced during track and capture, and because there may be separate neural representations for bimanual control of two objects versus bimanual control of a single object [55], the software was updated to use a single-object, three-dimensional target representation, as shown in Figure 4.2.

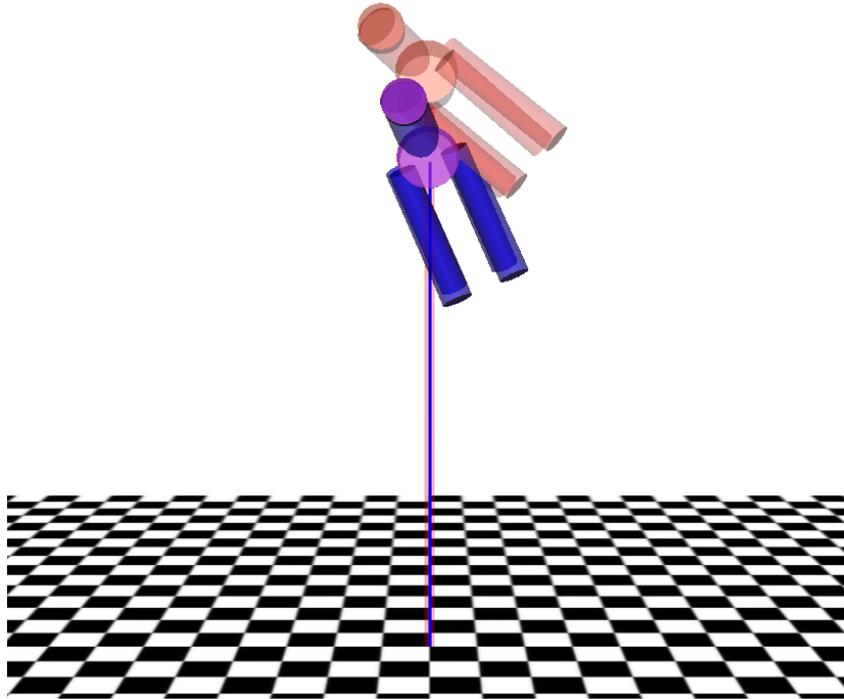


Figure 4.2 – Screenshot of the updated, single object, 3D BCCT display. In the trial shown, the target (more translucent, upper animal figure) was moving with a combination of roll and vertical translation. The subject was required to simultaneously use both the THC and RHC to move the cursor (more opaque, lower animal figure) to track the target.

Although in theory, an integrated two-dimensional target will result in better performance [56], it is very difficult in our case to unambiguously represent six degrees of freedom in a single two-dimensional object. Switching to a single-object, three-dimensional representation causes two main problems. First, tracking in three dimensions results in large errors in the depth axis [57], even if a stereoscopic display is used [58]. Second, depending on the nature of the three-dimensional object and which axes are displaced, it may be difficult to visually discern whether a given tracking error is due to translation or rotation. Although the second problem can largely be avoided by the choice of a spatially unambiguous object, the first problem is one that we chose to tolerate, as the alternative would be to avoid motion in the depth axis by reassigning the THC depth axis to control some other aspect of the cursor. This is not a desirable solution because it would break the intuitive correspondence between the three-dimensional controls paradigm in track and capture and the similar controls in the BCCT. We are able to partially ameliorate the poor depth tracking by adding a drop line extending from the object to a planar grid which is unambiguous in depth.

4.1.3.2 Feedback

Our simplified model of bimanual cross-coupling (Figure 2.2) is feed forward, but clearly some visual feedback is required in a tracking task. Because we are dealing with potentially six degrees of freedom with three axes per hand and any single directional coupling ratio is defined between only two of these axes, we actually have some flexibility in not only how we choose to present visual feedback of THC and RHC deflections, but also in how we present the command inputs.

In fact, preliminary trials showed that simultaneous tracking in six degrees of freedom at the specified frequencies was far too difficult even for a well-trained subject. This was not unexpected, since our simulated track and capture tasks never simultaneously require active motion in all six degrees of freedom, and empirically it appears that subjects only actively correct for errors in two or three degrees of freedom at a time during track and capture.

The ability to track the target effectively is important because the frequency commands present in the target motion must be passed through the subject relatively intact. Although the frequency detection can be made robust to any linear, time-invariant model of the human (i.e. an FIR representation of the human, consisting of time-invariant delays and scaling factors) that does not drastically cut off the specific command frequencies, any nonlinearities should be small enough that they do not cause significant frequency shifts.

Thus, it was decided to present target motion in only two degrees of freedom per trial, with one degree of freedom per hand, and to use multiple varied trials over the course of the entire BCCT to collect all of the coupling relationships.

4.1.4 Output Methodology

The outputs from the human in terms of both signal type and signal locations are clearly defined by our use of the same hand controllers as used in the simulated track and capture tasks. The signals consist of digital joystick readings sampled by a desktop computer at 30 Hz via the built-in Microsoft Windows XP™ game controller USB interface. The readings are normalized in the BCCT software to represent ratios of maximal hand controller displacement, and thus are a combination of mostly wrist and finger movements, depending on the motion, acting in conjunction with the specific mechanical configuration of the THC and RHC. Musculature used is kept approximately consistent with track and capture musculature by keeping the training essentially the same.

We see in Figure 4.3 an example time-series representation of the data that is collected.

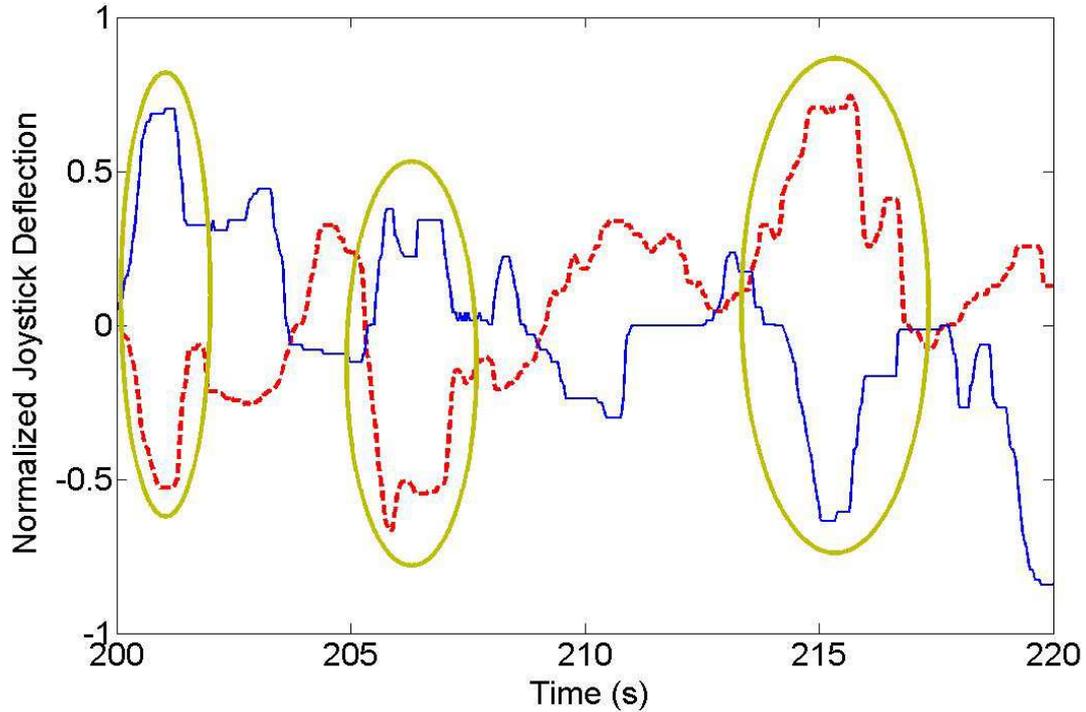


Figure 4.3 – Example of time series data, showing several possible instances of intermanual coupling (circled) from yaw (solid line) to lateral translation (dashed line). This is an excerpt of two of the six axes collected over 6.5% of the duration of a single BCCT trial.

4.1.5 Cross-Coupling Analysis Techniques

Our design choices so far strongly suggest, and actually were made in conjunction with, the design of the overall cross-coupling analysis technique: a spectral, frequency domain method that compares signal magnitudes at the specific, known command frequencies and compiles information from multiple trials to form a complete description of the causal relationship between each of the six control axes. After the time series data is logged by the Python-based BCCT, all analyses are performed offline in the Mathworks MATLAB® numerical computing and programming environment.

For its relative simplicity and computational efficiency in calculating spectral power, the Fourier transform is used to perform the spectral transformation on the time series data. Specifically, the MATLAB implementation of the Discrete-Time Fourier Transform (DTFT), using the Fast Fourier Transform (FFT) algorithm, is performed on each of the six time series $x_n[k]$ where n is the axis number (1 through 6) and k is the sample number. This gives the complex-valued spectral representation, X_n as a function of frequency ω :

$$X_n(\omega) \equiv \sum_{k=-\infty}^{\infty} x_n[k] \cdot e^{-i\omega k} \quad \text{Equation 4.1}$$

In any given trial, only two of these spectra will represent the “commanded axes” which are receiving command inputs due to corresponding motion of the visual target. Let us denote the

spectrum of a commanded axis as $X_c(\omega)$, and the specific command frequencies (0.03 Hz and 0.19 Hz for the RHC; 0.07 Hz and 0.17 Hz for the THC) for the given axis as ω_{c1} and ω_{c2} . Then the directional coupling ratio from this commanded axis to any other “receiving axis” n we can now define as

$$r_{cn} \equiv \left| \frac{X_n(\omega_{c1})X_n(\omega_{c2})}{X_c(\omega_{c1})X_c(\omega_{c2})} \right| \quad \text{Equation 4.2}$$

such that the coupling ratio represents a nonlinear (multiplicative) combination of the spectral information at the command frequencies (see Appendix I: Multiplicative Gap Filter for the rationale behind this choice). The amplitude and phase information are combined to form a scalar representation of magnitude, and the result is normalized such that the ratio becomes unity if the entire command signal, as detected on the commanded axis, appears equally on the coupled axis. We avoided calculating the coupling ratio from a commanded axis to another commanded axis, because a large command signal on the receiving axis was found to result in relatively large tracking error correction signals of unpredictable frequency, and this noise produced poor signal-to-noise ratios when we wanted to identify the relatively weaker coupled signal originating from another axis. To allow for a small amount of frequency shift, we take the maximum values of X_n and X_c over a three-sample window around the specified frequencies.

To illustrate these techniques, the same data from Figure 4.3 is shown in frequency domain in Figure 4.4.

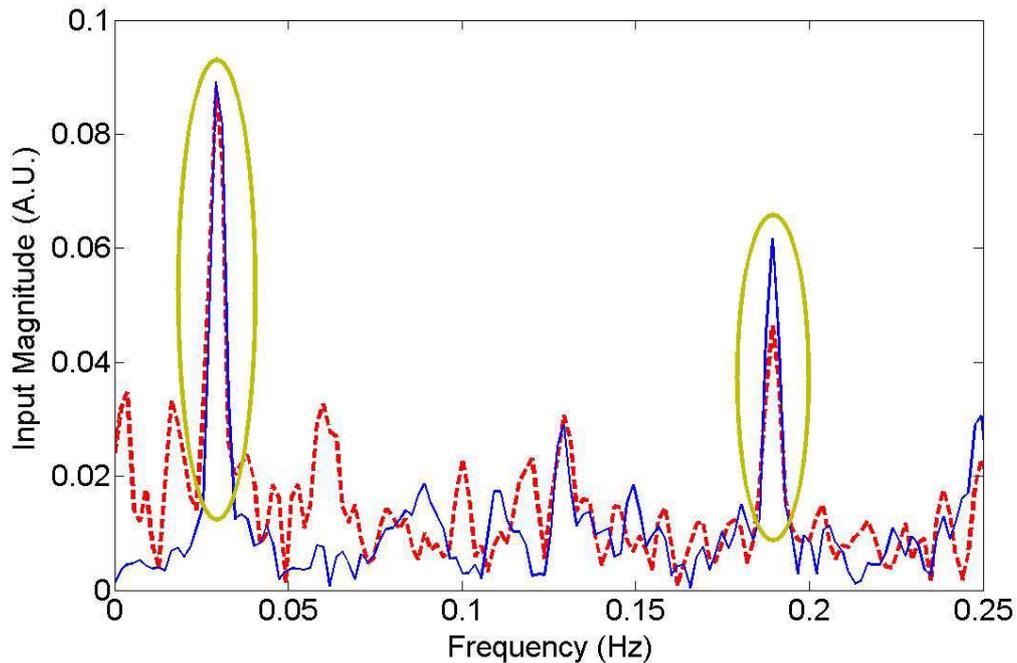


Figure 4.4 – Example of the processed spectral data, showing strong coupling of peaks at 0.03 Hz and 0.19 Hz. This figure is derived from the same BCCT time series data from which Figure 4.3 was excerpted. The strong coupling results in a calculated ratio of 0.86.

4.1.6 Trial Structuring

Each hand required three trials to measure the coupling ratios from each of the three possible commanded axes on the hand, and another three trials to measure coupling ratios to each of the three axes which could not be measured during the first three trials due to the avoidance of measuring coupling to a commanded axis. Thus, we chose to measure the complete bimanual cross-coupling matrix using the minimum of six trials by selecting the commanded pairs of axes so that each trial provided new coupling ratios of interest for both hands.

Despite minimizing the number of trials to complete the matrix, we obtained redundant information. Each trial allowed the measurement of intramanual coupling ratios from each commanded axis to two receiving axes on the same hand, for a total of four intramanual coupling ratios. Likewise, because we did not measure coupling from a commanded axis to another commanded axis, each trial allowed the measurement of intermanual coupling ratios from each commanded axis to two receiving axes on the opposing hand, for a total of four intermanual coupling ratios. Therefore, over the course of the six trials structured as shown in Table 4.1, we could measure up to 48 coupling ratios (eight per trial) for only 30 unique cross-coupling matrix values (a six-by-six matrix without the diagonal comprises 30 values), resulting in 18 redundant measurements calculated a second time, using a second pair of commanded axes. These 18 redundant measurements corresponded to three redundant measurements for each of six commanded axes, also corresponding to three for each of six receiving axes. No coupling ratio was measured redundantly more than once, and all redundant measurements took place in trials 4, 5, and 6.

Table 4.1 – Structuring of commanded pairs of axes. The resulting coupling ratio measurements for each trial are shown by checkmarks, each checkmark representing the measurement of two coupling ratios (one from each commanded axis). Cells with a gray background indicate that the measurement from the commanded axis on the THC was redundant and checkmarks with a white fill indicate that the measurement from the commanded axis on the RHC was redundant.

Trial #	Commanded Axes		Measured Coupling from Commanded Axes to...					
	THC	RHC	Lateral	Vertical	Depth	Roll	Pitch	Yaw
1	Lateral	Pitch	✘	✓	✓	✓	✘	✓
2	Vertical	Roll	✓	✘	✓	✘	✓	✓
3	Depth	Yaw	✓	✓	✘	✓	✓	✘
4	Vertical	Pitch	✓	✘	✓	✓	✘	✓
5	Lateral	Yaw	✘	✓	✓	✓	✓	✘
6	Depth	Roll	✓	✓	✘	✘	✓	✓

For example, in trial 1, commanding lateral translation and pitch allowed the measurement of coupling ratios for lateral translation and pitch to vertical translation, depth translation, roll, and yaw. Then in trial 4, commanding vertical translation and pitch allowed the measurement of coupling ratios from vertical translation and pitch to lateral translation, depth translation, roll, and yaw. Comparing the measurements of trial 1 and trial 4, there were thus the following three redundant measurements in trial 4 which were already measured by trial 1 using the same commanded RHC axis: pitch to depth translation, roll, and yaw. Similarly comparing the measurements of trial 2 and trial 4, one finds the following additional redundant measurements in trial 4 which were already measured by trial 2 using the same commanded THC axis: vertical translation to lateral translation, depth translation, and yaw.

We found that target geometry combined with commanded motion for trials 4 and 5 resulted in some visual ambiguity. For example, based on observation of the target motion and commentary from test subjects, we found that when vertical translation and pitch were commanded together (trial 4) there were intermittent periods of difficulty in discerning translation and rotation. When we had the choice of two commanded pairs of axes available for a coupling ratio calculation, we used the value derived from the visually explicit pair. Because the visual ambiguities appeared to be isolated to trials 4 and 5, to reduce neurovisual cross-coupling effects we simply discarded the redundant measurements from the latter half of the BCCT. Therefore, the only contributions to the cross-coupling matrix from trials in which visual ambiguity was significant were the measurements from trial 4 of pitch to lateral translation and of vertical translation to roll, and the measurements from trial 5 of yaw to depth translation and of lateral translation to pitch. Due to their directional and muscular asymmetry, we expected these four intermanual pairs to be very weakly coupled. Because visual ambiguity would only increase the measured cross-coupling by contributing an additional pathway for the independent command signals to interact, the only likely effect that visual cross-coupling would have on the BCCT results would be to increase the apparent coupling ratios of these intermanual pairs. However, as we verified in preliminary trials and as later shown in Table 4.2, these pairs showed weak coupling and were not important to the overall analysis, indicating that our results were indeed substantially unaffected by visual ambiguity.

Each of the six trials lasted for 5 minutes and 40 seconds. Including a brief rest period before each trial and an abbreviated, warm-up version of each trial at the start of the test made the entire BCCT 40 minutes in length.

4.2 Human Experiments

A human study was conducted to validate the BCCT with respect to our key hypotheses regarding stability and validity of the metrics, while simultaneously collecting the requisite data to potentially support our key hypotheses regarding utility of the metrics.

4.2.1 Ethics Statement

As a study involving humans, this experiment was designed in accordance with the rules of the Committee on the Use of Humans as Experimental Subjects (COUHES) at MIT, and was approved by the MIT Institutional Review Board (IRB). All subjects completed IRB-approved informed consent forms prior to commencement in the study.

4.2.2 Participants

Eighteen subjects (nine male) completed the study. The subjects were between 18 and 32 years of age, right handed (based on the revised Edinburgh Handedness Inventory [60], laterality quotients ranged from 56 to 100 on a ± 100 scale), with corrected vision of at least 20/40 Snellen acuity in each eye, no other significant visual impairments and no self-reported neuromotor diseases or disorders.

Subjects were recruited either through posters placed throughout the MIT campus or through emails distributed through MIT student mailing lists. As a result, a large portion of the subjects were either graduate or undergraduate students, but none of the subjects were from the MVL or had prior experience with a space telerobotics simulator. They were paid an hourly rate of \$10.50 with a bonus of \$20 for completing the entire study.

4.2.3 Apparatus, Stimuli, and Procedure

4.2.3.1 RWS Simulator

We used the physical RWS simulation setup in the Man Vehicle Lab at MIT, including a set of computer monitors and a THC and RHC pair designed to emulate as closely as practical the setup on the ISS.

4.2.3.2 Track and Capture Test (TCT)

We simulated the telerobotic “track and capture” task of tracking a target drifting in space while approaching it for capture. It is typical of tasks in space telerobotics and its neuromotor simplicity allowed us to focus on the neuromotor aspects of controlling the SSRMS. The experimental setup was made as similar to the real track and capture operation as possible. It used a simulated RWS environment, with simulated camera views of the SSRMS and control through a pair of THC and RHC devices that are similar to those used on the RWS. The simulator itself was developed at the Man Vehicle Lab (MVL) at the Massachusetts Institute of Technology (MIT) [17]. A photo of the simulation environment is shown in Figure 4.5.



Figure 4.5 – A photo of the RWS simulation environment, showing the THC, RHC, and camera view screens. The central view screen has been photographically enhanced to show the target alignment markings.

For our study, we designed a Track and Capture Test (TCT) of twelve simulated trials of the task. In each trial, the target drifted in space with a different combination of translational and rotational velocities. This motion required the subject to coordinate the THC and RHC, using the left and right hand respectively, to align the markings fixed on the two-dimensional end-effector camera view with corresponding markings on the drifting target. Control was performed in an internal frame of reference, with a coordinate system relative to the current orientation of the end-effector. We measured the quality of human telerobotic performance by metrics including the number of successful captures achieved in the twelve trials, the average time per successful trial from start to capture, and the dimensionless squared jerk [61] of the control inputs from start to capture.

Each TCT trial lasted one minute. With short breaks before each trial and a warm-up trial at the start, the entire TCT lasted fifteen minutes.

4.2.3.3 Bimanual Cross-Coupling Test (BCCT)

Two variations on the BCCT were introduced in this study in order to demonstrate the intraday stability of the BCCT and to establish the validity of the BCCT as a measure of strictly neural coupling. The Partial Bimanual Cross-Coupling Test (PBCCT) was simply the first two trials of the BCCT, so that intra-day repeatability could be tested without having to redo the entire test.

The Stationary Bimanual Cross-Coupling Test (SBCCT) commanded only one hand at a time, while the other hand was held stationary in a position offset from the mechanical dead band of

the hand controller so that any motion was detected. The SBCCT used the fact that neuromotor cross-coupling is negligible when the non-dominant hand is motionless or passively moving [41]. This fact could establish that the BCCT was measuring neuromotor cross-coupling and not cross-coupling caused by, for example, mechanical coupling through the subject's body or the table on which the RWS is placed. By running two trials, each commanding a different hand, we could verify the absence of this direct mechanical coupling by verifying that the same algorithm used in the BCCT detected minimal coupling in the SBCCT.

4.2.4 Protocol Overview

The overall experimental study took place over the course of thirteen to seventeen days for each subject, in three separate sessions. Session 1 lasted up to three hours on the first day and consisted of standardized training to allow the subject to execute the TCT and BCCT at an acceptable level, followed by the actual BCCT and TCT. Session 2 lasted up to two hours on the third day and consisted of the BCCT and TCT, followed by the MRT and PTA to measure the subject's inherent spatial abilities. Session 3 lasted for up to two hours nominally on day fifteen, with a two-day scheduling window, and consisted of the BCCT and TCT, followed by the PBCCT and SBCCT. The subjects completed pre- and post-study surveys to record such data as age and experience with musical instruments. Subjects also completed pre- and post-session surveys to collect session-specific data, such as the number of hours of sleep the subject had the night before and whether the subject was taking caffeine.

4.3 Validation of Metrics

The SBCCT results confirmed our hypothesis that the intermanual BCCT measurements were of a neural rather than mechanical origin. Each of the two SBCCT trials allowed calculation of the coupling ratio from the commanded axis to three axes on the nominally stationary hand. These six SBCCT coupling ratios are compared to their corresponding BCCT ratios in Figure 4.6. Clearly, the SBCCT values were much smaller, indicating that direct mechanical coupling was negligible.

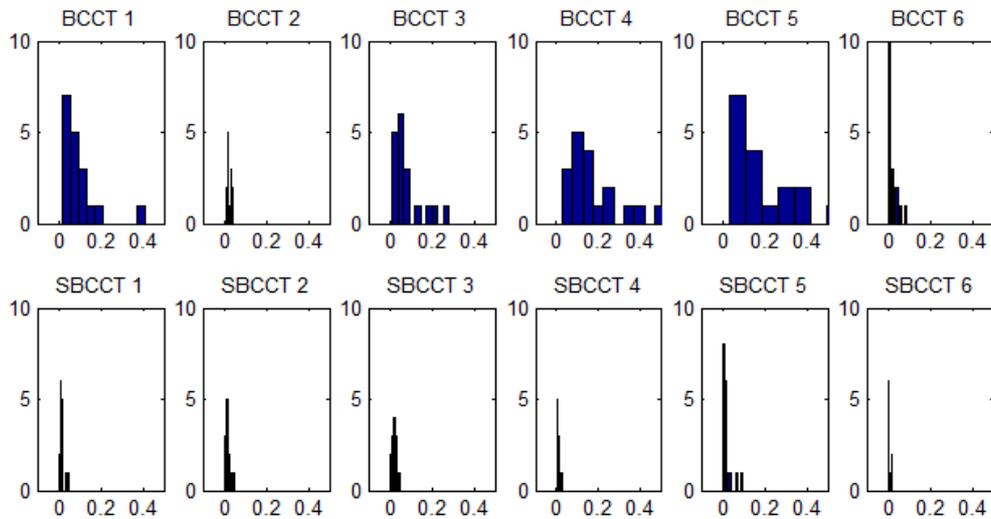


Figure 4.6 – Comparison of BCCT with SBCCT. Each histogram shows the coupling ratios on the horizontal plot axis and the counts (out of 18 subjects) on the vertical plot axis. The SBCCT measurements are shown directly below their corresponding BCCT measurements. Because the significantly coupled measurements were greatly reduced in the SBCCT case, we conclude that the BCCT measured neuromotor cross-coupling and not mechanical cross-coupling. The title numbers 1 through 6 refer to lateral translation to roll, lateral translation to pitch, lateral translation to yaw, pitch to lateral translation, pitch to vertical translation, and pitch to depth translation coupling ratios, respectively.

The PBCCT was designed to assess the intraday variability of the BCCT measurements. The difference between each coupling value obtained through the PBCCT and the corresponding coupling value obtained through the BCCT in Session 3 (the day the PBCCT is conducted) reflects both measurement error and intraday variability in the underlying coupling strength. We divided this difference by the mean of the two measurements to calculate the fractional difference. As shown in Figure 4.7, these values were approximately clustered about zero, showing that most of the variability was random in nature and there appeared to be very little systematic learning effect within Session 3. However, one-sample KS tests rejected the null hypothesis that the distributions were normal.

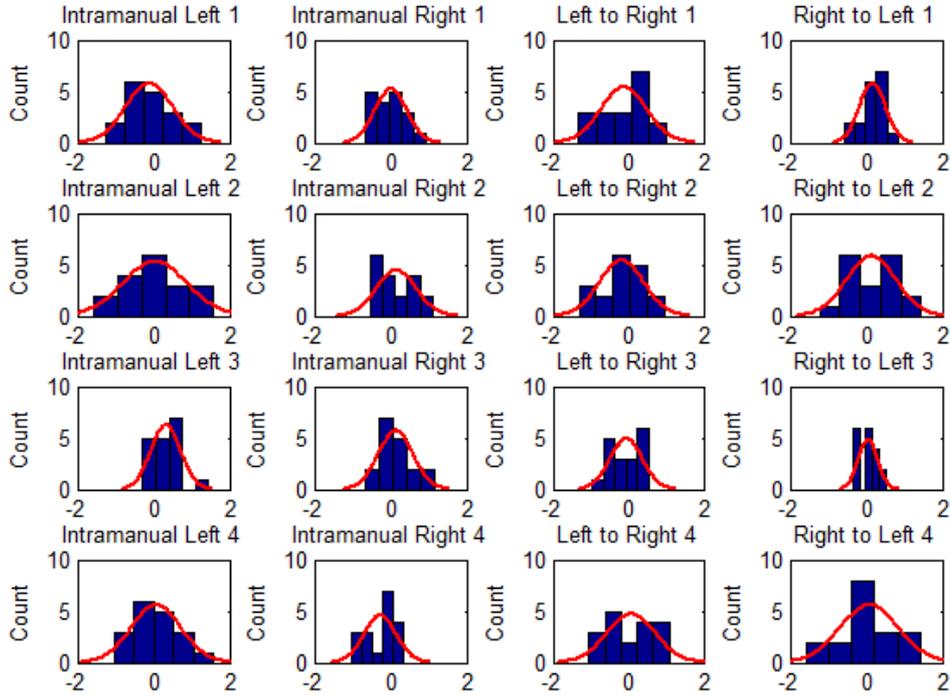


Figure 4.7 – Histograms of intraday fractional differences in BCCT/PBCCT measurements for four coupling directions in each of four categories. Best-fit normal curves are shown.

The measurement of coupling from pitch to vertical translation (the third right-to-left coupling in Figure 4.7) was the most precise. The standard deviation of its fractional difference was 0.26. From the Chebyshev inequality, we can expect a single such intermanual coupling measurement for one subject to lie within two standard deviations, i.e. $\pm 52\%$, of its true value with a probability of at least 0.75. Other coupling directions had poorer measurement precision, but incidentally, it is this most precisely measured coupling that is the most important in characterizing intermanual cross-coupling, as we will see in the following section.

4.4 Refinement of Metrics

Using the techniques described above, we generated a cross-coupling matrix M_{total}^{st} from each subject s and each session t . Its matrix elements, measured between two control axes, included not only the direct coupling but also indirect coupling in all orders through multiple intermediate control axes. As a second-order example, a motion originating from target tracking in yaw, affecting lateral translation due to intermanual coupling, then affecting vertical translation through intramanual coupling would increase the measured lateral to vertical translation coupling ratio. To simplify the analysis and focus on the underlying first-order cross-coupling, we estimated the direct, first-order matrix as follows:

$$M_{direct}^{st} = M_{total}^{st} - (M_{total}^{st} - I)^2 \quad \text{Equation 4.3}$$

where I is the identity matrix. Fluctuations in M_{total}^{st} caused some entries of M_{direct}^{st} to be negative, and some diagonal entries to be unequal to unity. We found the corrected, direct cross-coupling matrix, M^{st} by

$$M_{ij}^{st} = \max(0, M_{direct,ij}^{st})$$

Equation 4.4

$$M_{ii}^{st} = 1$$

for each row i and each column j . Because we found significant changes from M^{s1} to M^{s2} , we averaged only the stabilized M^{s3} over all 18 subjects to characterize the typical cross-coupling behavior of a population of trained subjects using our RWS simulator. The rows designate axes under tracking command and the columns designate axes receiving the corresponding coupled signals. The rows/columns are designated in the following order: lateral translation, vertical translation, depth translation, roll, pitch, and yaw. Thus, the upper-left quadrant represents left-hand (THC) intramanual coupling, the bottom-right quadrant represents right-hand (RHC) intramanual coupling, the bottom-left quadrant represents right-to-left intermanual coupling, and the upper-right quadrant represents left-to-right intermanual coupling. The values of the averaged matrix M^{avg3} are shown using this topology in Table 4.2.

Table 4.2 – The values of the matrix $M^{avg3} = \frac{1}{18} \sum_{a=1}^{18} M^{a3}$. Specific coupling values are designated by row/column subscripts of the form M_{ij}^{avg3} .

		j						
		To Left			To Right			
		Lateral	Vertical	Depth	Roll	Pitch	Yaw	
i	From Left	Lateral	1	.12	.01	.07	.00	.05
		Vertical	.29	1	.07	.01	.05	.02
		Depth	.15	.09	1	.00	.01	.04
	From Right	Roll	.06	.01	.01	1	.30	.41
		Pitch	.05	.15	.00	.18	1	.11
		Yaw	.21	.05	.02	.03	.02	1

The largest intermanual couplings were both right-to-left: M_{61}^{avg3} , representing yaw to lateral translation ($r = 0.21$), and M_{52}^{avg3} , representing pitch to vertical translation ($r = 0.15$). This asymmetry may have been a result of an asymmetry in our experiment (we used only right-handed subjects) and the inherent mechanical and kinematic asymmetries of the hand controllers. The ergonomics of the RHC also resulted in a very large value of M_{46}^{avg3} , representing roll to yaw intramanual coupling ($r = 0.41$).

These findings agreed with the results of a principal component analysis that decomposed M_{total}^{s3} into orthogonal linear combinations of the original couplings called principal components which, in sum, accounted for the full variance. Over 80% of the variance was explained by the two most prominent principal components of M_{total}^{s3} . The first of those components assigned the greatest weight (loading) to $M_{total,46}^{s3}$. This suggests that the first principal component represents the intramanual coupling due to RHC/hand biomechanics. In the second most prominent principal component (the one that explains the second largest fraction of the variance) the

weights of $M_{total,61}^{s3}$ and $M_{total,52}^{s3}$ were dominant. This suggests that the second principal component represents the intermanual cross-coupling represented by these axes.

These coupling ratios were strongly correlated. M_{61}^{st} was significantly correlated with M_{52}^{st} across all sessions ($p \leq 0.01$) as shown in Table 4.3.

Table 4.3 – Correlation coefficients between the two principal intermanual coupling ratios, M_{61}^{st} and M_{52}^{st} , for each session. All entries are significant at $p \leq 0.01$.

	M_{61}^{s1}	M_{61}^{s2}	M_{61}^{s3}
M_{52}^{s1}	0.65	0.90	0.74
M_{52}^{s2}	0.67	0.67	0.74
M_{52}^{s3}	0.62	0.58	0.76

At the end of 4.3 Validation of Metrics, we quantified the measurement precision of $M_{total,52}^{st}$. We did not take repeated measures of $M_{total,61}^{st}$ as part of the two of six repeated trials in the PBCCT, but the above analysis suggests that the neuromotor properties of $M_{total,61}^{st}$ are similar to those of $M_{total,52}^{st}$; they are both principal measures of intermanual cross-coupling. Therefore their measurement precisions may be similar. Note also that, as detailed in Table 4.1 **Error! Reference source not found.**, both of these important cross-coupling ratios were derived from BCCT trials in which there were no significant visual ambiguities between commanded axes.

5 Applications of Results

5.1 Applications in Space Telerobotics

5.1.1 Understanding Cross-Coupling Behavior

Learning effects across sessions appeared to result in lasting reduction of intermanual cross-coupling with practice, as may be expected of a measure of neuromotor skill. These session effects are shown for M_{52}^{st} in Figure 5.1.

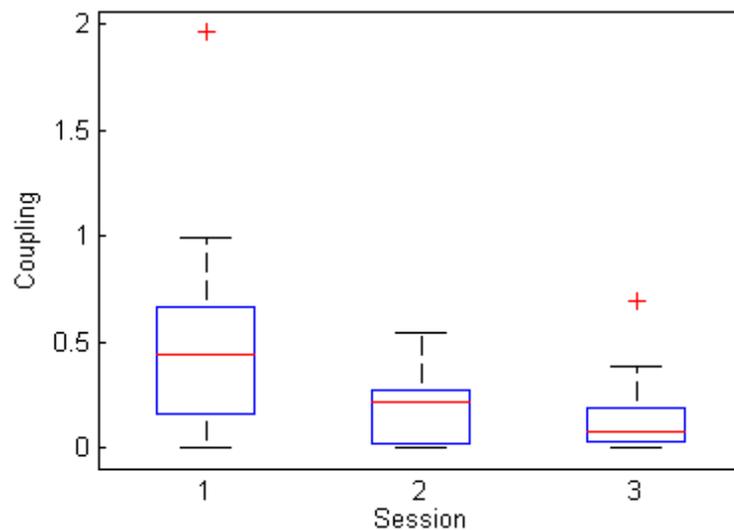


Figure 5.1 – Pitch to vertical translation (M_{52}^{st}) boxplots, showing session effects. A learning effect reduced coupling between the closely-spaced first two sessions. Strong retention between the last two sessions prevented the acquired motor skill from degrading despite the thirteen-day gap in practice. Note the outlier in Session 1; the BCCT algorithm allowed for a coupling ratio greater than unity, which could result from poor motor skills.

Individual subjects appeared to exhibit distinct levels of coupling, as would be expected of a measure of inherent neuromotor ability. These subject effects are shown in Figure 5.2.

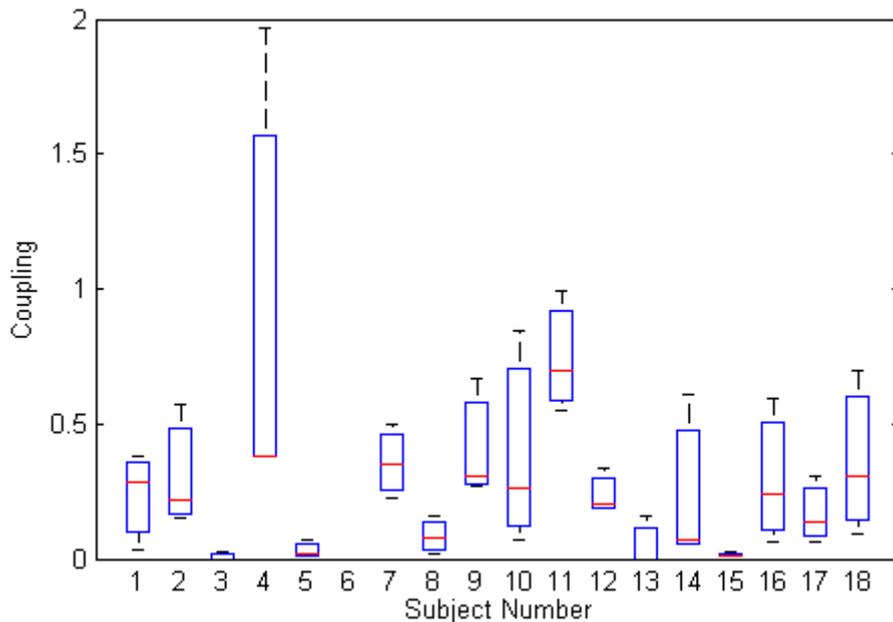


Figure 5.2 – Pitch to vertical translation (M_{52}^{st}) boxplots, showing subject effects. Although each boxplot consists of only three measures across the non-independent sessions, they illustrate the range of coupling strengths exhibited by subjects.

We performed a mixed regression of the cross-couplings for each of the three coupling directions of interest. Subject was treated as a random effect and session number as a fixed effect.

Over the three sessions, there was a significant effect of session on $\log M_{61}^{st}$ (-0.356 ± 0.080 per session, $p < 0.0005$), indicating a learning effect. A learning retention effect was significant because the session coefficient became much smaller in magnitude when only data from last two sessions were used (-0.110 ± 0.027 per session, $p < 0.0005$); improvement was not as pronounced over the 13-day gap in practice, but on the average there was no loss of motor skill. There was also a significant effect of subject (-1.248 ± 0.206 , $p < 0.0005$).

There was no significant learning effect seen for $\log M_{52}^{st}$ but the effect of subject was significant (-1.418 ± 0.388 , $p < 0.0005$).

There was no significant learning effect seen for the intramanual (square root of) M_{46}^{st} , but the effect of subject was again significant (0.570 ± 0.121 , $p < 0.0005$). The consistently significant effects of subject support the promise held out by further analysis of differences between subjects.

5.1.2 Reducing Cross-Coupling and Improving Manual Controller Design

It is ergonomically intuitive that the intramanual coupling M_{46}^{st} (yaw to roll) tends to be large: the RHC joint mechanically couples forearm pronation/supination with lateral translation. This lateral translation may manifest as palmar extension/flexion, actuating the yaw axis. The reason we do not see as much yaw to roll coupling (the opposite direction) is probably because of a

mechanical dead band in the joystick that makes it easier to actuate yaw alone. Note that the real RHC on the ISS is designed to roll about the center of the operator's hand. This design would reduce the undesired intramanual coupling, but would likely not provide significant advantages for the intermanual cross-coupling.

Apart from providing separate training to reduce intermanual cross-coupling between problematic axes, engineers could alter the design of the hand controllers to reduce cross-coupling. One way to do so could exploit the differential strengths of input and output coupling for certain control axes. For example $\sum_{a=1}^6 M_{1a}^{avg3} = 1.25$ while $\sum_{a=1}^6 M_{a1}^{avg3} = 1.76$, and subtracting the 1 contributed by the matrix diagonal from each term indicates that the lateral translation axis was about three times more susceptible to receiving coupled signals than sending them to other axes. Simply by decreasing the computer's input sensitivity on such axes which receive coupled signals more readily, in principle one can decrease the overall intermanual cross-coupling that is detected by the computer and fed to the telerobotic system.

The presence of intermanual cross-coupling in isometric tasks [62] and the observation that cross-coupling is transient and largest during initiation of discrete tasks [20] suggests that cross-coupled neuromotor signals may manifest primarily as force signals through unintended muscle activation, only indirectly affecting the actively controlled displacements. If so, then one can attenuate the undesired cross-coupling by increasing the mechanical stiffness of any control axes identified as particularly susceptible to receive coupling. This decrease in mechanical sensitivity from force to displacement may achieve a similar effect to decreasing the computer's input sensitivity, but without sacrificing the range of the control signal or its sensitivity to the desired displacement commands. A digital low-pass filter may achieve a similar effect by reducing force-based intermittent cross-coupling while retaining the desired displacement commands. More research on the topic of intermanual cross-coupling reduction would be beneficial.

5.1.3 Measuring and Predicting Space Telerobotics Performance

The primary measure of performance on the TCT was the number of successful captures out of the twelve trials, $TCT_{success}^{st}$, for each subject s and each session t . We found negative and predominantly significant correlations between $TCT_{success}^{st}$ and the intramanual coupling M_{46}^{st} , as summarized in Table 5.1.

Table 5.1 – Correlation coefficients between number of TCT successes ($TCT_{success}^{st}$) in each session and intramanual coupling (M_{46}^{st}) in each session. All entries except '-0.43' ($p=0.08$) are significant at $p \leq 0.05$.

	M_{46}^{s1}	M_{46}^{s2}	M_{46}^{s3}
$TCT_{success}^{s1}$	-0.59	-0.50	-0.75
$TCT_{success}^{s2}$	-0.71	-0.57	-0.62
$TCT_{success}^{s3}$	-0.63	-0.43	-0.46

The significant correlations between intramanual coupling measured at the first session and subsequent success at the TCT suggest that intramanual coupling may be a useful predictor of

telerobotics performance, as we expect from a measure of motor skill. More research on this topic would clarify the predictive utility of BCCT-based metrics.

Because subjects were instructed to capture the targets in the least amount of time possible, the mean time required to capture a TCT target, TCT_{time}^{st} (seconds, not including failed trials), was another useful metric of performance which we expected would increase with increased coupling. The positive correlations between TCT_{time}^{st} and M_{46}^{st} reached statistical significance ($p < 0.05$) for three of the nine possible correlations (between sessions, analogous to Table 5.1). Moreover, although subjects were instructed to capture the target when within a certain spatial envelope, without specifically reducing alignment errors beyond what is required, we expected alignment errors to increase with increased coupling. The positive correlations between angular alignment error at the instant of capture, TCT_{error}^{st} (degrees, vector norm of three dimensions), and M_{46}^{st} reached statistical significance for five of the nine possible correlations. These results together suggest that intramanual coupling may be a useful measure of hand controller mastery, which affects both speed and accuracy of telerobotics performance. BCCT-based metrics may thus be useful as diagnostics to suggest remedial training or to verify the effectiveness of training.

Similar correlations with the intermanual couplings M_{52}^{st} and M_{61}^{st} were not statistically significant.

Summary data on the TCT performance metrics are presented in Table 5.2 below.

Table 5.2 – Mean \pm standard deviation of the TCT performance metrics for each session.

Session (t)	$TCT_{success}^{st}$	TCT_{time}^{st}	TCT_{error}^{st}
1	9.6 \pm 2.4	38.4 \pm 3.7	1.70 \pm 0.50
2	10.6 \pm 2.7	36.5 \pm 4.3	1.69 \pm 0.81
3	11.7 \pm 0.75	35.5 \pm 2.9	1.58 \pm 0.65

5.1.4 Supporting Measures of Spatial Ability

Spatial ability, as represented by the MRT score, was significantly correlated with telerobotics performance. Across the 18 subjects, the correlation coefficients between MRT score and $TCT_{success}^{s1}$, $TCT_{success}^{s2}$, and $TCT_{success}^{s3}$ were 0.51 ($p = 0.03$), 0.48 ($p = 0.04$), and insignificant (0.28, $p = 0.25$), respectively. This reproduces, in part, the positive correlation seen elsewhere between spatial ability and telerobotic performance [16, 17]. Correlations between MRT score and TCT_{error}^{st} upon target capture for each session were negative but not significant (-0.32, -0.37, and -0.37).

We might expect spatial ability (as measured by MRT score) and bimanual cross-coupling to be related, because a variety of spatial and perceptuo-motor tasks have been demonstrated to activate similar brain regions, possibly as aspects of a general intelligence factor [63], and spatial and motor skills are often proposed to be part of an even more specific spatial-mechanical intelligence factor [64]. It is also possible that because the BCCT required interpretation of the spatial orientation and position of a target, spatial ability directly contributed to measurement of the bimanual cross-coupling despite our test design that mitigated these effects. However, only two of the nine correlations shown in Table 5.3 are significant.

Table 5.3 – Correlation coefficients between MRT score and each of the important coupling ratios for each session. Entries less than -0.46 are significant at $p < 0.05$.

Session (t)	M_{46}^{st}	M_{61}^{st}	M_{52}^{st}
1	-0.44	-0.41	-0.43
2	-0.60	-0.35	-0.39
3	-0.55	-0.36	-0.26

Although only the MRT correlations with M_{46}^{s2} and M_{46}^{s3} were statistically significant ($p = 0.01$ and $p = 0.02$, respectively), six of the other seven results fell in the insignificant, but nearby interval between -0.36 and -0.46. Another study, perhaps with more subjects, might give more consistently significant correlations between increased MRT and increased ability to independently control multiple degrees of freedom.

The PTA scores exhibited relationships to cross-coupling similar to those seen in MRT – again largely insignificant. The correlation between MRT and PTA scores was 0.53 ($p = 0.02$), for 18 subjects.

5.2 Neuromechanics and the Study of Manual Control

5.2.1 Causes of Intermanual Cross-Coupling

We discarded data obtained from BCCT trials noted to exhibit visual cross-coupling, but could not categorically remove all visual cross-coupling from our measurements – the subtraction of indirect effects to estimate M^{st} (Equation 4.3) caused data obtained from the less visually explicit BCCT trials 4 and 5 to affect all elements of M^{st} to at least a small extent. Although we excluded visual ambiguity as much as practicable, a future experiment is clearly needed to verify the minimization of visual cross-coupling by comparing cross-coupling ratios obtained using several different types of visual objects and drift configurations.

In 4.4 Refinement of Metrics, we identified M_{52}^{st} and M_{61}^{st} as the dominant metrics of intermanual cross-coupling. Given the hand controller grip techniques that the subjects were trained to use (demonstrated in Figure 4.5), the quantity M_{52}^{st} (pitch to vertical translation coupling ratio) represents right-hand radial/ulnar deviation coupling to mostly left-hand palmar extension/flexion, and M_{61}^{st} represents right-hand palmar extension/flexion coupling to mostly left-hand radial/ulnar deviation. This cross-coupling between non-homologous muscles supports the previously held assumption that that neuromotor intermanual cross-coupling is not directly associated with muscular symmetry [28, 65, 66].

Instead of muscular symmetry, these intermanual coupling pairs are alike in task-space response as trained in the TCT, for which the task-space is the two-dimensional end-effector camera view. Taking M_{52}^{st} as an example, in track and capture, when the user performs a pitch motion on the RHC the response that appears on the screen is similar to that when performing a vertical translation on the THC. Since these motions contribute to closely related responses of the camera view, subjects must learn to coordinate these hand motions to achieve the desired alignments. Similarly for M_{61}^{st} , the coupling from yaw to lateral translation, subjects learn a visuomotor association while they are engaged in TCT and carry it over to the BCCT, where the

visual association is much weaker and the association between motor commands is in fact disadvantageous to have. Other common space telerobotics scenarios besides track and capture are performed with an external frame of reference for control, similar to the BCCT, and may suffer from the same learning carry-over effect as from the TCT to the BCCT. Previous findings related to motor learning and consolidation suggest that such implicitly learned visuomotor behaviors can persist even when the task is changed, potentially conflicting with explicit task goals [67]. A portion of the measured BCCT cross-coupling may therefore be caused by learned visuomotor associations in TCT that carry over into involuntary neuromotor cross-coupling in the BCCT despite inherently decoupled visual responses to motor commands in BCCT.

Beyond this learning carry-over effect, these intermanual pairs may be the most strongly coupled because they are the pairs that most strongly reflect “ergonomic symmetry,” the symmetry of the real-space motions that would result if the hand controllers were actuated in an ergonomically natural way, free of artificial mechanical constraints. For example, the Logitech joystick that we used as the RHC rotates in roll, pitch, and yaw about a point located below the right hand. Pitching the RHC is perceived to require primarily radial/ulnar deviation, which if not for the mechanical constraint of the RHC pivot point, would naturally also result in vertical translation of the hand and fingers; this concomitant vertical translation due to radial/ulnar deviation may explain why pitch couples with vertical translation on the THC.

Similarly, yawing the RHC is perceived to require primarily palmar extension/flexion, which if not for the pivot point fixed in translation and a desire to avoid coupled roll motion, would naturally also result in lateral translation of the hand and fingers; this concomitant lateral translation due to palmar extension/flexion may explain why yaw couples with lateral translation on the THC.

Rolling the THC is perceived to require primarily forearm pronation/supination, which if not for the mechanical pivot point that causes concomitant lateral translation, would not naturally involve any translation of the hand; this lack of a concomitant translation due to forearm pronation/supination may explain why roll does not couple strongly to the THC. This explanation based on ergonomic motion similarly accounts for our results parsimoniously and adds to a growing body of literature supporting the perceptual bases of neural control in bimanual tasks [28, 65, 66].

It is also to be expected on an ergonomic basis that the intramanual coupling M_{46}^{st} (roll to yaw) should be large: the RHC pivot point mechanically couples the forearm pronation/supination of roll with lateral translation of the hand. This lateral translation may manifest as palmar extension/flexion, actuating the yaw axis. The reason we do not see as much yaw to roll coupling (the opposite direction) is probably because of a mechanical dead band in the joystick that makes it easier to actuate yaw alone. Note that the real RHC on the Space Station is designed to roll about the center of the operator’s hand. This design would reduce the undesired intramanual coupling, but would likely not provide significant advantages for the intermanual cross-coupling.

Apart from providing separate training to reduce intermanual cross-coupling between problematic axes, engineers could alter the design of the hand controllers to reduce cross-coupling. One way to do so could exploit the differential strengths of input and output coupling for certain control axes. For example $\sum_{a=1}^6 M_{1a}^{avg3} = 1.25$ while $\sum_{a=1}^6 M_{a1}^{avg3} = 1.76$, and subtracting the 1 contributed by the matrix diagonal from each term indicates that the lateral translation axis was about three times more susceptible to receiving coupled signals than

sending them to other axes. Simply by decreasing the computer's input sensitivity on such axes which receive coupled signals more readily, in principle one can decrease the overall intermanual cross-coupling that is detected by the computer and fed to the telerobotic system.

The presence of intermanual cross-coupling in isometric tasks [62] and the observation that cross-coupling is transient and largest during initiation of discrete tasks [20] suggests that cross-coupled neuromotor signals may manifest primarily as force signals through unintended muscle activation, only indirectly affecting the actively controlled displacements. If so, then one can attenuate the undesired cross-coupling by increasing the mechanical stiffness of any control axes identified as particularly susceptible to receive coupling. This decrease in mechanical sensitivity from force to displacement may achieve a similar effect to decreasing the computer's input sensitivity, but without sacrificing the range of the control signal or its sensitivity to the desired displacement commands. A digital low-pass filter may achieve a similar effect by reducing force-based intermittent cross-coupling while retaining the desired displacement commands. More research on the topic of intermanual cross-coupling reduction would be beneficial.

5.2.2 Dimensionless Squared Jerk as a Measure of Skill Smoothness

We also investigated the use of dimensionless squared jerk [61] as a track and capture performance metric, to improve upon the subjective measures of input smoothness that NASA currently uses. For each TCT, the vector norm of the dimensionless squared jerks of the six control inputs was calculated and averaged for all successful trials. Because subjects were instructed to perform smooth movements, this reciprocal measure of smoothness serves as another potential telerobotic performance metric. Based on correlations between the data taken over all three sessions, jerk does not appear to have a consistent correlation with the other performance metrics we considered. However, in Session 2 alone, jerk seemed to reflect a reciprocal of general motor skill, showing sporadic correlations across subjects (correlation coefficients given per session, values greater than 0.46 significant at $p < 0.05$) with increased M_{46}^{st} (0.51, 0.27, 0.23), increased TCT_{time}^{st} (0.50, 0.61, 0.00), increased TCT_{error}^{st} (0.44, 0.49, 0.07), and decreased $TCT_{success}^{st}$ (0.30, 0.40, 0.44). It is possible that jerk becomes representative of an individual's inherent abilities only after significant day-after-day training. These inconsistent suggestions of the data warrant further investigation with more data.

6 Conclusions & Suggestions for Future Work

We designed and implemented a bimanual cross-coupling test (BCCT) for space telerobotics that quantifies both intermanual and intramanual cross-coupling in a matrix that allows the identification of problem axes corresponding to principal cross-coupling pathways. Using our simulated space telerobotics joysticks and combining BCCT data across 18 test subjects (Table 4.2), these axes were found to be yaw to lateral translation (21% coupling) and pitch to vertical translation (15% coupling), in the intermanual case, and roll to yaw (41% coupling) in the intramanual case. The cross-coupling matrix represents the “black box” input-output characteristics of the human controller. We argue that these observed cross-coupling effects are probably ergonomic in nature, caused by a combination of biomechanics and perceptual expectations about the motions required to actuate the hand controllers. We suggest methods for decreasing the influence of undesired cross-coupling in bimanual control systems.

We found significant differences in bimanual cross-coupling between subjects (Figure 5.2), and demonstrated that subjects could significantly reduce intermanual cross-coupling with practice (Figure 5.1), suggesting that these metrics may be useful indicators of control device mastery. Principal intermanual coupling ratios were strongly correlated with each other (Table 4.3), demonstrating that the tendency to have high or low intermanual cross-coupling was an individual subject trait. We found statistically significant correlations in the range of -0.43 to -0.75 between early-stage intramanual coupling and subsequent performance in a simulated space telerobotics track and capture task, with eight of nine correlations across sessions being significant (Table 5.1), suggesting that an intramanual coupling metric may be useful as a predictor of human telerobotic performance. Intramanual cross-coupling was significantly correlated with individual spatial ability (Table 5.3). The BCCT technique could ultimately be applied to evaluate and improve performance during astronaut training and also to reduce undesired cross-coupling through improved hand controller design.

These results should be investigated further using the specialized RWS hardware actually used by NASA, which is probably less prone to intramanual coupling of roll to yaw. Further experiments comparing different BCCT tracking displays are needed to determine the relative role of ambiguities in object motion versus cross-coupling in motor outflow. It may also be possible to simplify the BCCT, focusing on typically highly coupled axes, thereby creating an “operational” version of the BCCT that offers greater predictive or diagnostic benefit in a shorter amount of testing time.

The role of bimanual cross-coupling may be clarified by investigating it under the challenging conditions of microgravity faced by astronauts in space [68]. There have been investigations that seek to quantify spatial disorientation during re-entry into a gravitational field [69], to reduce the cognitive workload of spacecraft operators undergoing such disorientation [70], and to understand the effect of Coriolis cross-coupling stimulus on the vestibular system [71]. The effect of such vestibular disturbances on manual control abilities or the reduction of bimanual control workload through controller designs that minimize undesired cross-coupling between axes has received less attention. There is evidence to suggest an intimate connection between the neuromuscular control of manual tasks and whole-body postural control [72], which depends on perceived vestibular inputs. Because of the time limitations in subsonic parabolic flight, these experiments may need to be conducted on the ISS, depending on the extent to which the duration of the BCCT can be shortened.

Our findings regarding ergonomic symmetry in bimanual control and the use of mixed translation/rotation bimanual motions may even be useful to incorporate into rehabilitative robotics [73]. It has been suggested that robotic rehabilitation results may be improved by incorporating aspects of bimanual symmetry [74] or symmetry breaking [75], yet as of 2012, these aspects have been underexplored in the field of rehabilitation [75].

7 Glossary

- Coupling
 - Mutual interaction between two entities, implies either one-way or symmetric nature
 - “Roll and yaw are coupled, but not symmetrically.”
 - “The roll to yaw coupling ratio was 0.5.”
- Cross-Coupling
 - Interaction between two or more entities; implies potential asymmetry if only two entities are specified.
 - “Roll and yaw cross-coupling can be characterized by the roll to yaw coupling strength and the yaw to roll coupling strength.”
 - “The six control axes are cross-coupled with each other.”
- Intermanual
 - Between the two hands.
 - “Intermanual cross-coupling depends on the nature of the task.”
- Intramanual
 - Within a single hand.
 - “There is also intramanual cross-coupling between control axes.”
- Bimanual
 - Performed with both hands. May be used to refer jointly to intermanual and intramanual effects relevant to a two-handed task.
 - “This bimanual task involves both intermanual and intramanual cross-coupling between control axes, which can be characterized by a six-by-six bimanual cross-coupling matrix.”

Note that no cause is implied when the term “intermanual” is used. That is, the author may specify “neuromotor intermanual cross-coupling” or “mechanical intermanual cross-coupling through the subject’s body.” The author avoids the term “crosstalk” because while it is sometimes used interchangeably with “cross-coupling” in the literature, “crosstalk” may imply some kind of signal-level corruption. If indeed the word is used in reference to cross-coupling that has been isolated to a neural cause, it can be more accurately described as “neuromotor cross-coupling” or alternatively “neurovisual cross-coupling.”

8 References

1. Aramaki, Y., M. Honda, and N. Sadato, *Suppression of the Non-Dominant Motor Cortex During Bimanual Symmetric Finger Movement: A Functional Magnetic Resonance Imaging Study*. Neuroscience, 2006(141 (2147-2153)).
2. Banerjee, A. and V.K. Jirsa, *How do neural connectivity and time delays influence bimanual coordination?* Biological Cybernetics, 2006. **96**(2): p. 265-278.
3. Wenderoth, N., et al., *Parieto-premotor Areas Mediate Directional Interference During Bimanual Movements*. Cerebral Cortex, 2004. **14**(10 (1153-1163)).
4. Pollok, B., et al., *Intercerebellar Coupling Contributes to Bimanual Coordination*. Journal of Cognitive Neuroscience, 2007. **19**(4 (704-719)).
5. Kennerley, S.W., et al., *Callosotomy patients exhibit temporal uncoupling during continuous bimanual movements*. Nature Neuroscience, 2002. **5**(4): p. 376-381.
6. Boniface, S.J. and U. Ziemann, *Plasticity in the human nervous system : investigations with transcranial magnetic stimulation*. 2003, Cambridge ; New York: Cambridge University Press. xii, 316 p., 8 p. of plates.
7. Viviani, P., et al., *Hemispheric Asymmetries and Bimanual Asynchrony*. Experimental Brain Research, 1998(120 (531-536)).
8. Serrien, D.J., M.J. Cassidy, and P. Brown, *The Importance of the Dominant Hemisphere in the Organization of Bimanual Movements*. Human Brain Mapping, 2003(17 (296-305)).
9. Diedrichsen, J., et al., *The role of the corpus callosum in the coupling of bimanual isometric force pulses*. J Neurophysiol, 2003. **90**(4): p. 2409-18.
10. Hogan, N. *Robotic Therapy to Aid Neuro-Recovery: Is Walking Like Reaching?* in *The Goldstein Lecture Series*. 2011. Israel: Israel Institute of Technology.

11. Swinnen, S.P., et al., *Directional interference during bimanual coordination: is interlimb coupling mediated by afferent or efferent processes*. Behavioural Brain Research, 2003. **139**(1-2): p. 177-195.
12. Weigelt, C. and S.C. de Oliveira, *Visuomotor transformations affect bimanual coupling*. Experimental Brain Research, 2002(148 (439-450)).
13. Kasuga, S. and D. Nozaki, *Cross talk in implicit assignment of error information during bimanual visuomotor learning*. J Neurophysiol, 2011. **106**(3): p. 1218-26.
14. Deml, B. and B. Farber, *Development of a Design Guide for Telepresence-Teleactions Systems*. 5th International Workshop on Presence, 2002(pp88-103).
15. Suzuki, Y., et al. *Learning Process of Bimanual Coordination*. in *International Conference on Control, Automation and Systems*. 2008.
16. Pontillo, T.M., *Spatial Ability and Handedness as Potential Predictors of Space Teleoperation Performance*, in *MS Thesis*. 2010, Massachusetts Institute of Technology.
17. Tomlinson, Z.A., et al. *Influence of Spatial Ability on Primary and Secondary Space Telerobotics Operator Performance*. in *Aviation, Space, and Environmental Medicine*. 2009.
18. Liuzzi, G., et al., *Coordination of uncoupled bimanual movements by strictly timed interhemispheric connectivity*. Journal of Neuroscience, 2011. **31**(25): p. 9111-7.
19. Deml, B., *Human Factors Issues on the Design of Telepresence Systems*. Presence, 2007. **16**(5:471-487).
20. Heuer, H., W. Spijkers, and T. Kleinsorge, *The time course of cross-talk during the simultaneous specification of bimanual movement amplitudes*. Exp Brain Res, 1998(118:381-392).
21. Maki, Y., et al., *Asymmetric control mechanisms of bimanual coordination: An application of directed connectivity analysis to kinematic and functional MRI data*. NeuroImage, 2008(42:1295-1304).

22. Marteniuk, R.G., C.L. MacKenzie, and D.M. Baba, *Bimanual movement control: Information processing and interaction effects*. The Quarterly Journal of Experimental Psychology Section A, 1984(36:335-365).
23. Guiard, Y. and T. Ferrand, *Asymmetry in Bimanual Skills*, in *Manual asymmetries in motor performance*. 1996, CRC Press, Inc.
24. Bock, O., C. Worringham, and S. Dawson, *On the Use of Different Spatial Reference Frames for Movement Control*. Motor Control, 2009. **13**(1:43-53).
25. Kandel, E.R., J.H. Schwartz, and T.M. Jessell, *Voluntary Movement*, in *Principles of Neural Science*, 4/e. 2000, McGraw-Hill Companies.
26. Obhi, S.S. and M.A. Goodale, *Bimanual interference in rapid discrete movements is task specific and occurs at multiple levels of processing*. Journal of Neurophysiology, 2005. **94**(3): p. 1861-1868.
27. de Noordhout, A.M., et al., *Corticomotoneuronal synaptic connections in normal man - An electrophysiological study*. Brain, 1999. **122**: p. 1327-1340.
28. Kunde, W., H. Krauss, and M. Weigelt, *Goal congruency without stimulus congruency in bimanual coordination*. Psychological Research-Psychologische Forschung, 2009. **73**(1): p. 34-42.
29. Duque, J., et al., *Monitoring Coordination during Bimanual Movements: Where Is the Mastermind?* Journal of Cognitive Neuroscience, 2010. **22**(3): p. 526-542.
30. Summers, J.J., et al., *Coordination dynamics and attentional costs of continuous and discontinuous bimanual circle drawing movements*. Hum Mov Sci, 2008. **27**(5): p. 823-37.
31. Schaal, S., et al., *Rhythmic arm movement is not discrete*. Nature Neuroscience, 2004. **7**(10): p. 1136-1143.
32. Sternad, D., *Towards a unified theory of rhythmic and discrete movements - Behavioral, modeling and imaging results*. Coordination: Neural, Behavioral and Social Dynamics, 2008: p. 105-133.

33. Dipietro, L., et al., *Submovement changes characterize generalization of motor recovery after stroke*. *Cortex*, 2009. **45**(3): p. 318-324.
34. Fradet, L., G. Lee, and N. Dounskaia, *Origins of submovements in movements of elderly adults*. *Journal of Neuroengineering and Rehabilitation*, 2008. **5**.
35. Fautrelle, L., et al., *Pointing to Double-Step Visual Stimuli from a Standing Position: Motor Corrections When the Speed-Accuracy Trade-Off Is Unexpectedly Modified in-Flight. A Breakdown of the Perception-Action Coupling*. *Neuroscience*, 2011. **194**: p. 124-135.
36. Fitts, P.M., *The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement*. *Journal of Experimental Psychology*, 1954. **47**(6): p. 381-391.
37. Bangert, A.S., et al., *Bimanual coordination and aging: neurobehavioral implications*. *Neuropsychologia*, 2010. **48**(4): p. 1165-70.
38. Ringenbach, S.D., et al., *Specific instructions are important for continuous bimanual drumming in adults with Down syndrome*. *Downs Syndr Res Pract*, 2006. **11**(1): p. 29-36.
39. Rosenbaum, D.A., A.M. Dawson, and J.H. Challis, *Haptic tracking permits bimanual independence*. *Journal of Experimental Psychology: Human Perception and Performance*, 2006. **32**(5): p. 1266-1275.
40. Andres, F.G., et al., *Functional coupling of human cortical sensorimotor areas during bimanual skill acquisition*. *Brain*, 1999. **122**: p. 855-870.
41. Ridderikhoff, A., C.E. Peper, and P.J. Beek, *Unraveling interlimb interactions underlying bimanual coordination*. *Journal of Neurophysiology*, 2005. **94**(5): p. 3112-3125.
42. Smith, S.M., et al., *Network modelling methods for FMRI*. *Neuroimage*, 2011. **54**(2): p. 875-891.
43. Bressler, S.L. and A.K. Seth, *Wiener-Granger Causality: A well established methodology*. *Neuroimage*, 2011. **58**(2): p. 323-329.
44. Bruns, A., *Fourier-, Hilbert- and wavelet-based signal analysis: are they really different approaches?* *Journal of Neuroscience Methods*, 2004. **137**(2): p. 321-332.

45. Schlogl, A. and G. Supp, *Analyzing event-related EEG data with multivariate autoregressive parameters*. Event-Related Dynamics of Brain Oscillations, 2006. **159**: p. 135-147.
46. Yacoub, K., *Relationship between Multiple and Partial Coherence Functions*. IEEE Transactions on Information Theory, 1970. **16**(6): p. 668-+.
47. Fox, M.D. and M.E. Raichle, *Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging*. Nature Reviews Neuroscience, 2007. **8**(9): p. 700-711.
48. Franz, E.A., *Attentional distribution of task parameters to the two hands during bimanual performance of right- and left-handers*. Journal of Motor Behavior, 2004. **36**(1): p. 71-81.
49. Supp, G.G., et al., *Directed Cortical Information Flow during Human Object Recognition: Analyzing Induced EEG Gamma-Band Responses in Brain's Source Space*. Plos One, 2007. **2**(8).
50. Wesner, J.W., J. Hiatt, and D. Trimble, *Winning with quality : applying quality principles in product development*. Engineering process improvement series. 1995, Reading, Mass.: Addison-Wesley Pub. Co. xx, 300 p.
51. Gill, N.S. and P.S. Froyer. *Component-based measurement: a few useful guidelines*. in *ACM SIGSOFT Software Engineering Notes*. 2003.
52. Harbour, J.L., *The basics of performance measurement*. 2nd ed. 2009, Boca Raton: CRC Press. vii, 93 p.
53. Campbell, G.K., *How to Use Metrics*, in *CSO: The Resource for Security Executives*. 2006. p. 5.
54. Barnes, G.R. and C.J. Ruddock, *Factors affecting the predictability of pseudo-random motion stimuli in the pursuit reflex of man*. J Physiol, 1989. **408**: p. 137-65.
55. Howard, I.S., J.N. Ingram, and D.M. Wolpert, *Composition and Decomposition in Bimanual Dynamic Learning*. Journal of Neuroscience, 2008. **28**(42): p. 10531-10540.
56. Balakrishnan, R. and K. Hinckley. *Symmetric Bimanual Interaction*. in *ACM CHI2002*. 2002. New York.

57. Massimino, M.J., T.B. Sheridan, and J.B. Roseborough, *One-Handed Tracking in 6-Degrees of Freedom*. 1989 Ieee International Conference on Systems, Man, and Cybernetics, Vols 1-3, 1989: p. 498-503.
58. Zhai, S., P. Milgram, and A. Rastogi, *Anisotropic human performance in six degree-of-freedom tracking: An evaluation of three-dimensional display and control interfaces*. Ieee Transactions on Systems Man and Cybernetics Part a-Systems and Humans, 1997. **27**(4): p. 518-528.
59. Cardoso de Oliveira, S. and S. Barthelemy, *Visual feedback reduces bimanual coupling of movement amplitudes, but not of directions*. Experimental Brain Research, 2005. **162**(1): p. 78-88.
60. Williams, S., *A major revision of the Edinburgh Handedness Inventory*: Colchester, Essex, UK. p. 2.
61. Hogan, N. and D. Sternad, *Sensitivity of smoothness measures to movement duration, amplitude, and arrests*. J Mot Behav, 2009. **41**(6): p. 529-34.
62. Diedrichsen, J., *The Role of the Corpus Callosum in the Coupling of Bimanual Isometric Force Pulses*. Journal of Neurophysiology, 2003. **90**(4): p. 2409-2418.
63. Duncan, J., et al., *A neural basis for general intelligence*. Science, 2000. **289**(5478): p. 457-460.
64. Mohler, J.L., *A Review of Spatial Ability Research*. Engineering Design Graphics Journal, 2008. **72**(3).
65. Muller, K., et al., *Perceptual influence on bimanual coordination: an fMRI study*. European Journal of Neuroscience, 2009. **30**(1): p. 116-124.
66. Mechsner, F., *Response to commentaries: Actions as perceptual-conceptual "gestalts"*. Journal of Motor Behavior, 2004. **36**(4): p. 408-417.
67. Krakauer, J.W., *Motor Learning and Consolidation: The Case of Visuomotor Rotation*. Progress in Motor Control: A Multidisciplinary Perspective, 2009. **629**: p. 405-421.

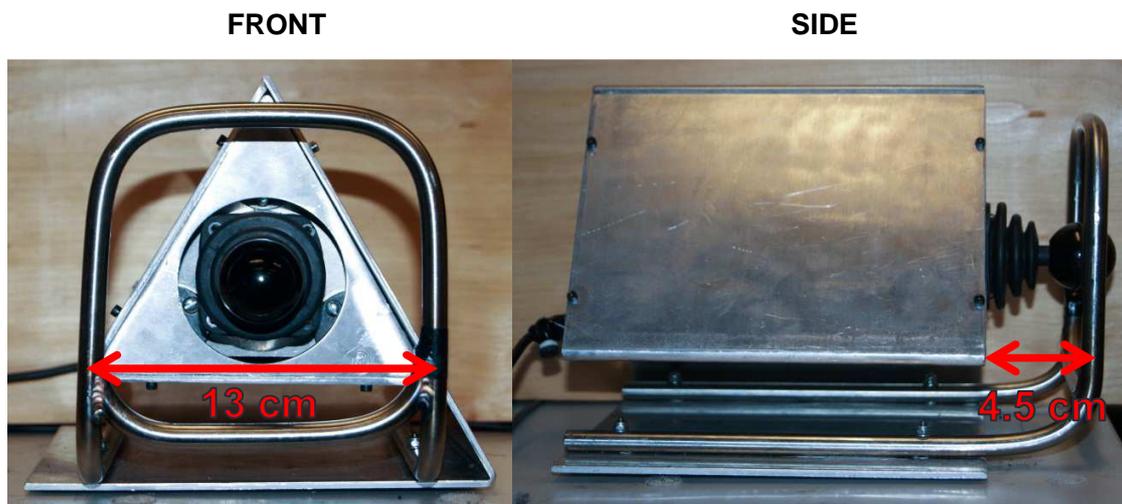
68. Oman, C.M., *Spatial Orientation and Navigation in Microgravity*, in *Spatial processing in navigation, imagery, and perception*, F. Mast and L. Jäncke, Editors. 2007, Springer: New York. p. 210-242.
69. Clark, T.K., et al., *Numerical simulation of human orientation perception during lunar landing*. *Acta Astronautica*, 2011. **69**(7-8): p. 420-428.
70. Young, L.R., et al. *Sensorimotor Controls and Displays for Safe and Precise Lunar Landing*. in *61st International Astronautical Congress*. 2010. Prague, Czech Republic.
71. Mateus, J., et al. *Cross-Coupled Stimulus During Artificial Gravity: Asymmetric Tumbling Sensation Response*. in *Eight Symposium on Vestibular Research*. 2011.
72. Bonnetblanc, F., O. Martin, and N. Teasdale, *Pointing to a target from an upright standing position: anticipatory postural adjustments are modulated by the size of the target in humans*. *Neuroscience Letters*, 2004. **358**(3): p. 181-184.
73. Hogan, N., et al., *Interactive robotic therapist*, U.S.P.a.T. Office, Editor. 1995, Massachusetts Institute of Technology.
74. McAmis, S. and K.B. Reed, *Symmetry Modes and Stiffness for Bimanual Interaction*, in *IEEE International Conference on Rehabilitation Robotics*. 2011, Rehab Week Zurich: ETH Zurich Science City, Switzerland.
75. Sleimen-Malkoun, R., et al., *Bimanual training in stroke: How do coupling and symmetry-breaking matter?* *BMC Neurol*, 2011. **11**: p. 11.
76. Schlogl, A., *A comparison of multivariate autoregressive estimators*. *Signal Processing*, 2006. **86**(9): p. 2426-2429.
77. Smith, S.W., *The Scientist and Engineer's Guide to Digital Signal Processing*. 1997: California Technical Publishing.
78. Banerjee, A. and V.K. Jirsa, *How do neural connectivity and time delays influence bimanual coordination?* 2006(96 (265-278)).

79. Aramaki, Y., M. Honda, and N. Sadato, *Suppression of the Non-Dominant Motor Cortex During Bimanual Symmetric Finger Movement: A Functional Magnetic Resonance Imaging Study*. 2006(141 (2147-2153)).
80. Wenderoth, N., et al., *Parieto-premotor Areas Mediate Directional Interference During Bimanual Movements*. 2004. **14**(10 (1153-1163)).
81. Pollok, B., et al., *Intercerebellar Coupling Contributes to Bimanual Coordination*. 2007. **19**(4 (704-719)).
82. Kennerley, S., et al., *Callosotomy patients exhibit temporal uncoupling during continuous bimanual movements*. 2002. **5**(4).
83. Viviani, P., et al., *Hemispheric Asymmetries and Bimanual Asynchrony*. 1998(120 (531-536)).
84. Aramaki, Y., M. Honda, and N. Sadato, *Suppression of the Non-Dominant Motor Cortex During Bimanual Symmetric Finger Movement: A Functional Magnetic Resonance Imaging Study*. 2006(141 (2147-2153)).
85. Serrien, D.J., M.J. Cassidy, and P. Brown, *The Importance of the Dominant Hemisphere in the Organization of Bimanual Movements*. 2003(17 (296-305)).

Appendix A: Hand Controller Characteristics

Throughout all TCT and BCCT trials, the THC and RHC were spaced 80 cm apart center-to-center and oriented with their front edges aligned with the straight edge of the table. Subjects were allowed to adjust chair height to a comfortable position; generally, the forearms were approximately horizontal. Subjects were instructed to hook the pinky finger of the left hand around the THC rail for support, as astronauts do.

A.1 Translational Hand Controller



The THC, shown above, had a range of translational motion of 3 cm in each direction, with an approximately linear spring constant requiring the following forces at maximum displacement of the grip point:

Lateral (left or right): 4 N

Vertical (up or down): 4 N

Depth (push in): 9 N

Depth (pull out): 13 N

A.2 Rotational Hand Controller



The RHC, shown above, had a range of rotational motion of about 20 degrees in each direction, with an approximately linear spring constant requiring the following forces at maximum displacement of the approximate center of grip (~10 cm from the pivot point, which is visually obscured by the rubber gasket):

Roll (left or right): 5 N

Pitch (forward or backward): 5 N

These torques are equivalent to an approximate torque of 0.5 N·m about the pivot point.

The yaw was measured as an approximate torque about the joystick shaft at maximum displacement:

Yaw (left or right): 0.1 N·m

Appendix B: TCT Drift and Control Parameters

Trial #	Target Drift				
	Lateral (m/s)	Vertical (m/s)	Depth (m/s)	Pitch (degrees/s)	Yaw (degrees/s)
Warm Up	0.01	0.01	0.005	0.5	0.5
1	-0.01	0.01	-0.005	0	0
2	0.01	-0.01	-0.005	-0.7	0
3	0.01	0.01	0.005	0	0.7
4	-0.01	-0.01	0.005	-0.7	0
5	0.01	0.01	0.005	0	0
6	-0.01	0.01	-0.005	0	0.7
7	0.01	-0.01	-0.005	0	0
8	-0.01	-0.01	0.005	0	0.7
9	-0.01	0.01	-0.005	-0.7	0
10	0.01	-0.01	-0.005	0	0.7
11	0.01	0.01	0.005	-0.7	0
12	-0.01	-0.01	0.005	0	0

Appendix C: Logistical Details of Human Experiment

The study duration was nominally 15 days with training and test sessions nominally on Days 1, 3 and 15. The details of each test session are given in the schema below, with the times representing approximate upper limits:

Day 1 (2 hrs 40 mins):

- Consent & pre-study screening questionnaire (15 mins)
- Pre-session questionnaire (5 mins)
- BCCT training (15 mins)
- TCT training (55 mins)
- Break (5 mins)
- BCCT (40 mins)
- Break (5 mins)
- TCT (15 mins)
- Post-session questionnaire (5 mins)

Day 3 (1 hr 40 mins):

- Pre-session questionnaire (5 mins)
- BCCT (40 mins)
- Break (5 mins)
- TCT (15 mins)
- Spatial ability tests (30 mins)
- Post-session questionnaire (5 mins)

Day 15 (1 hr 55 mins):

- Pre-session questionnaire (5 mins)
- BCCT (40 mins)
- Break (5 mins)
- TCT (15 mins)
- Break (5 mins)
- Partial BCCT (15 mins)
- Stationary BCCT (15 mins)
- Post-session questionnaire (5 mins)
- Post-study questionnaire & debrief (10 mins)

For all sessions, subjects were requested to arrive well-rested and free of influence from any drugs that they do not routinely take. All sessions took place in the Man Vehicle Lab, and were supervised by the author.

During Day 1, scheduled for 3 hours by appointment, each subject was informed of the nature of the study and was screened for eligibility. The pre-study screening questionnaire and vision screening collected relevant subject information (i.e. age, gender, history of playing video games, time of day most alert, etc.). If ineligible, the subject was paid for one hour and thanked for their participation; if eligible, the subject continued according to the schema. The pre-session

questionnaire was then administered to determine the subject's current state (i.e. tired or well-rested, use of drugs, etc.). Unless the subject was incapable of proceeding with the session, the pre-session questionnaire was only be used for informational purposes to aid data analyses.

The subjects began with a 15-minute BCCT introductory training block, involving a presentation of instructions and a set of brief training runs in each of the six pairs of commanded axes.

After the BCCT training, the subject was given simulated telerobotics training in a Track and Capture Test (TCT). The training period consisted of instructional slides, personal instruction and hands-on practice to prepare the subject to take the TCT. The TCT consisted of a fifteen minute series of track and capture trials. Typically, a single track and capture trial requires under one minute, and several seconds were given between trials for the subject to rest.

After a short break, the actual BCCT that the subject was earlier trained for was administered. Every BCCT included at the start a brief review and warm-up period. Another short break was taken before the actual TCT that the subject was also trained for. Every TCT also included at the start a brief review and warm-up period.

Finally, the post-session questionnaire determined how the subject felt about his/her performance, and ensured the subject's well-being (asking whether there was any excessive mental exertion, boredom, negative emotions, etc.).

Day 3, scheduled for 2 hours by appointment, consisted of the BCCT and TCT, separated by a short break, in addition to the standard pre- and post-session questionnaires. Then, two standard spatial ability tests were administered. The Perspective Taking Ability (PTA) test was computer-based, recording responses using the computer mouse, and required about 15 minutes. The Mental Rotation Test (MRT) was paper-and-pencil-based and required about 10 minutes.

Day 15, scheduled for 2 hours by appointment, took place nominally after an eleven day break (two weeks after Day 1). Depending on availability and scheduling, Day 15 may have taken place between 12 and 16 days after Day 1, so that the range of possible study durations was actually 13 to 17 days. In Day 15, the subject performed the BCCT and TCT again to determine telerobotic skill retention and stability of BCCT metrics. After a break, the subject performed the Partial BCCT, which consisted of redoing two of the six pairs of commanded axes to verify intra-day repeatability of the measurements. The subject then performed the Stationary BCCT. The fifteen minutes allotted for the Stationary BCCT included a brief explanation and practice period. A post-study debrief and questionnaire was then administered, which measured the subject's own estimate of skill retention and the subject's overall impression of the study.

Appendix D: Questionnaires

The following questionnaires differ in format from the versions given to the subjects. White space and heading fields including time, date and subject identification have been removed for brevity.

D.1 Pre-study Questionnaire

Please answer the following questions honestly, to the best of your ability. Your personal information is confidential and will only be stored in a non-identifiable format.

1. What is your gender?
(Male) (Female)
2. What is your age?
3. What is your height?
4. What is your weight?
5. Are you an MIT student?
(Yes) (No) If "Yes," please write your course of study (including grad/undergrad)
6. Please mark the box that best describes which hand you use for the activity in question:

	Always Left	Usually Left	No Preference	Usually Right	Always Right
Writing					
Throwing					
Scissors					
Toothbrush					
Knife (without fork)					
Spoon					
Match (when striking)					
Computer mouse					

7. Do you have any visual issues such as myopia (nearsightedness), hyperopia (farsightedness), astigmatism, strabismus, amblyopia, diplopia, color blindness, cataracts, etc?
(Yes) (No) If "Yes," please describe
8. Have you been prescribed corrective lenses?
(Yes) (No) If "Yes," please write the prescription/strength in each eye

9. What is the (corrected) Snellen visual acuity of your left eye (e.g. 20/20, 20/40, 20/80)? If you are not sure, please ask the investigator to test you using a standard Snellen chart.

10. What is the (corrected) Snellen visual acuity of your right eye?

11. Do you have any neurological, neuromotor, or musculoskeletal issues such as schizophrenia, autism, split brain syndrome, epilepsy, Parkinson's disease, Huntington's disease, multiple sclerosis, cerebral palsy, spina bifida, muscular dystrophy, arthritis, etc?

(Yes) (No) If "Yes," please describe

12. Do you normally consume caffeine in the form of coffee, caffeinated soft drinks, etc?

(Yes) (No) If "Yes," please describe the form/quantity of caffeine you typically consume per day

13. Do you normally take any other over-the-counter or prescription drugs?

(Yes) (No) If "Yes," please list the drugs and specify daily dosage

14. At what times of day are you most alert (check up to four boxes)?

<input type="checkbox"/>	12 AM to 3 AM
<input type="checkbox"/>	3 AM to 6 AM
<input type="checkbox"/>	6 AM to 9 AM
<input type="checkbox"/>	9 AM to 12 PM
<input type="checkbox"/>	12 PM to 3 PM
<input type="checkbox"/>	3 PM to 6 PM
<input type="checkbox"/>	6 PM to 9 PM
<input type="checkbox"/>	9 PM to 12 AM

15. At what times of day are you most sleepy (check up to four boxes)?

<input type="checkbox"/>	12 AM to 3 AM
<input type="checkbox"/>	3 AM to 6 AM
<input type="checkbox"/>	6 AM to 9 AM

<input type="checkbox"/>	9 AM to 12 PM
<input type="checkbox"/>	12 PM to 3 PM
<input type="checkbox"/>	3 PM to 6 PM
<input type="checkbox"/>	6 PM to 9 PM
<input type="checkbox"/>	9 PM to 12 AM

16. How many hours do you normally sleep per day?

<input type="checkbox"/>	Approximately zero
<input type="checkbox"/>	1 or 2 hours
<input type="checkbox"/>	3 or 4 hours
<input type="checkbox"/>	5 or 6 hours
<input type="checkbox"/>	7 or 8 hours
<input type="checkbox"/>	9 or 10 hours
<input type="checkbox"/>	More than 10 hours

17. Do you have experience with virtual environments (e.g. 3-D games, CAD, graphic design)?

(Yes) (No) If "Yes," please describe this experience

18. Do you have experience with joysticks/game controllers (e.g. computer/video games, robotics)?

(Yes) (No) If "Yes," please describe this experience

19. Do you have experience with two-handed musical instruments (e.g. piano, guitar, flute)?

(Yes) (No) If "Yes," please describe this experience

D.2 Pre-session Questionnaire

Please answer the following questions honestly, to the best of your ability. Your personal information is confidential and will only be stored in a non-identifiable format.

1. Are you currently wearing corrective lenses?

(Yes) (No)

2. Are you currently under the influence of caffeine?

(Yes) (No) If "Yes," describe the form/quantity of caffeine and how long ago it was consumed

3. Are you currently under the influence of alcohol or any other drugs (including over-the-counter and prescription drugs)?

(Yes) (No) If "Yes," list the drugs, with dosage and how long ago you took the last dose

4. How many hours did you sleep in the last 24 hours?

<input type="checkbox"/>	Approximately zero
<input type="checkbox"/>	1 or 2 hours
<input type="checkbox"/>	3 or 4 hours
<input type="checkbox"/>	5 or 6 hours
<input type="checkbox"/>	7 or 8 hours
<input type="checkbox"/>	9 or 10 hours
<input type="checkbox"/>	More than 10 hours

5. Please indicate your sleepiness during the 5 minutes before this rating by checking the appropriate description below. Also use the intermediate steps:

1	<input type="checkbox"/>	very alert
2	<input type="checkbox"/>	
3	<input type="checkbox"/>	alert to normal level
4	<input type="checkbox"/>	
5	<input type="checkbox"/>	neither alert nor sleepy
6	<input type="checkbox"/>	
7	<input type="checkbox"/>	sleepy, but no effort to keep awake
8	<input type="checkbox"/>	

9 very sleepy, great effort to keep awake, fighting sleep

6. Is there currently any other physical, mental, or emotional factors that may affect your ability to consistently coordinate motions of both hands in response to visual feedback from a computer display?

(Yes) (No) If "Yes," please describe

D.3 Post-session Questionnaire

Please answer the following questions honestly, to the best of your ability. Your personal information is confidential and will only be stored in a non-identifiable format.

1. If you experienced any of the following, please circle your level of discomfort:

EFFECT	NONE				SEVERE
Nausea	1	2	3	4	5
Dizziness	1	2	3	4	5
Disorientation	1	2	3	4	5
Eyestrain	1	2	3	4	5
Blurred vision	1	2	3	4	5
Sweating	1	2	3	4	5
Headache	1	2	3	4	5
General discomfort	1	2	3	4	5
Mental fatigue	1	2	3	4	5
Hand fatigue	1	2	3	4	5
Hand pain	1	2	3	4	5
Other	1	2	3	4	5

2. How enjoyable/interesting were the BCCT tasks?

Boring

1	2	3	4	5
---	---	---	---	---

 Captivating
Comments?

3. How enjoyable/interesting were the TCT tasks?

Boring

1	2	3	4	5
---	---	---	---	---

 Captivating
Comments?

4. Rate your ability to perform the following tasks during the session:

		UNABLE				EXPERT
A.	Tracking BCCT target	1	2	3	4	5

B.	Predicting BCCT target motion	1	2	3	4	5
C.	Moving each hand independently during BCCT	1	2	3	4	5
D.	Moving each joystick only in the desired axis during BCCT	1	2	3	4	5
E.	Tracking TCT target	1	2	3	4	5
F.	Predicting TCT target motion	1	2	3	4	5
G.	Moving each hand independently during TCT	1	2	3	4	5
H.	Moving each joystick only in the desired axis during TCT	1	2	3	4	5

5. How difficult were the BCCT tasks?

Very easy

1	2	3	4	5
---	---	---	---	---

 Very difficult
Comments?

6. How difficult were the TCT tasks?

Very easy

1	2	3	4	5
---	---	---	---	---

 Very difficult
Comments?

D.4 Post-study Questionnaire

Please answer the following questions honestly, to the best of your ability. Your personal information is confidential and will only be stored in a non-identifiable format.

1. How and to what extent did the gap in practice between Session 2 and today affect your ability to perform the following tasks?

	MUCH WORSE		SAME		MUCH BETTER
Tracking BCCT target	1	2	3	4	5
Predicting BCCT target motion	1	2	3	4	5
Moving each hand	1	2	3	4	5

independently during BCCT					
Moving each joystick only in the desired axis during BCCT	1	2	3	4	5
Tracking TCT target	1	2	3	4	5
Predicting TCT target motion	1	2	3	4	5
Moving each hand independently during TCT	1	2	3	4	5
Moving each joystick only in the desired axis during TCT	1	2	3	4	5

2. Any other suggestions or comments about this study?

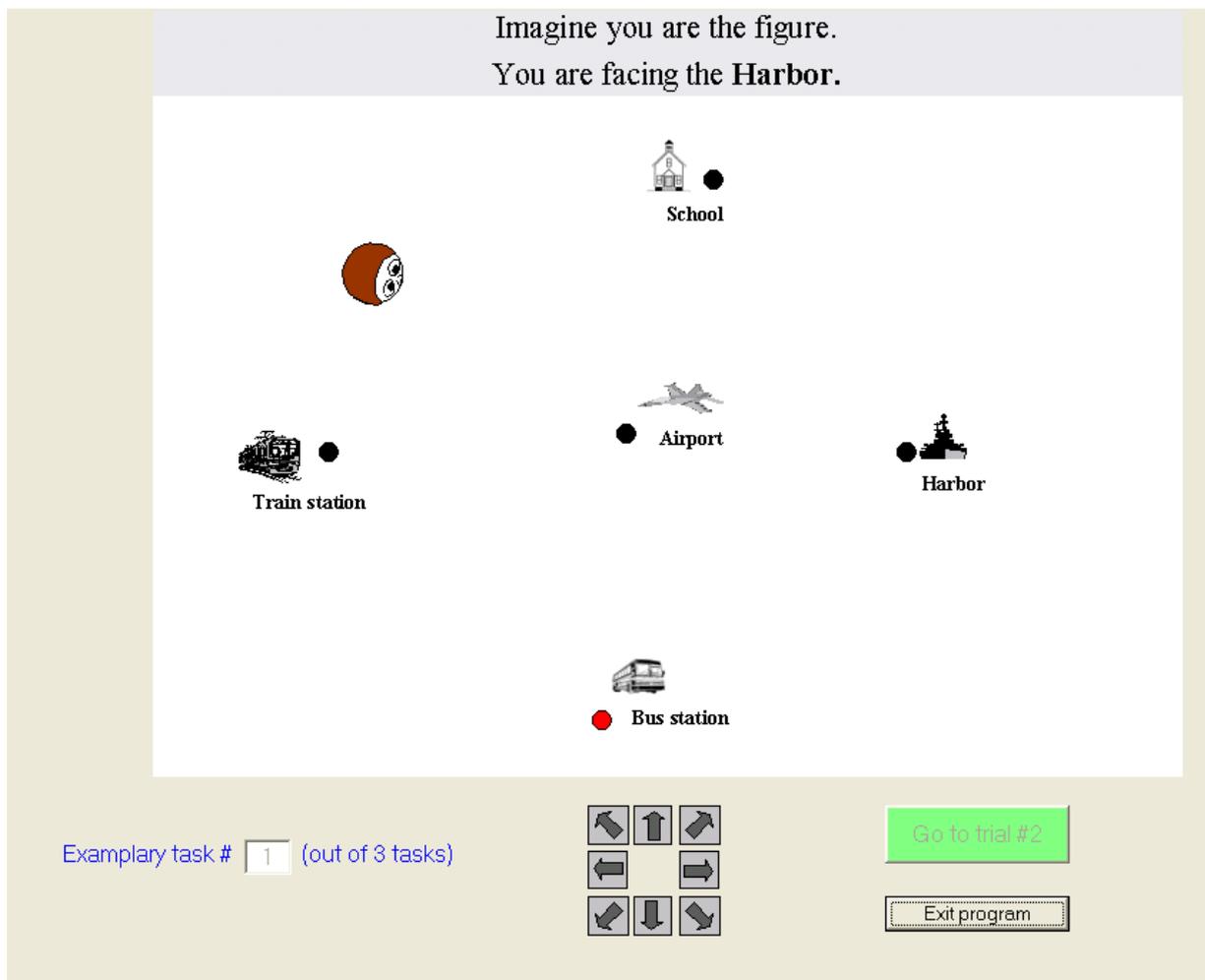
Appendix E: Spatial Ability Tests

E.1 Perspective Taking Ability (PTA) Test

The computerized PTA test that we used was created by MM Virtual Design, LLC, and is available from the following web address:

http://www.mmvirtualdesign.com/html/pta_test.html

The test requires subjects to imagine the visual perspective of a red circular figure within a plane. A screenshot illustrating the test shown below:



E.2 Mental Rotations Test (MRT)

The Mental Rotations Test (MRT) that we used is presented in the following six pages.

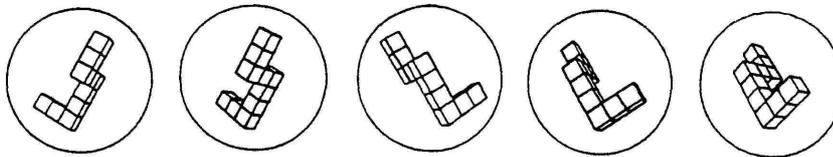
M.R.T. Test

Experiment:

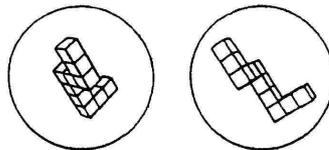
Subject #:

Date:

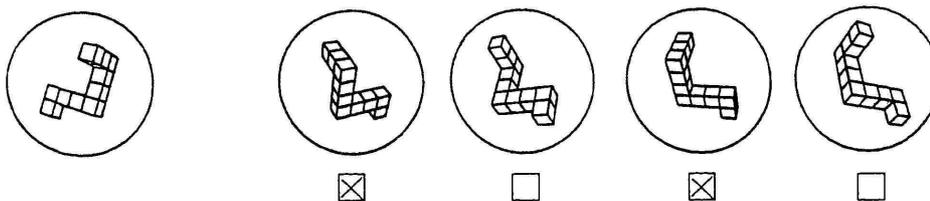
This is a test of your ability to look at a drawing of a given object and find the same object within a set of dissimilar objects. The only difference between the original object and the chosen object will be that they are presented at different angles. An illustration of this principle is given below, where the same single object is given in five different positions. Look at each of them to satisfy yourself that they are only presented at different angles from one another.



Below are two drawings of new objects. They cannot be made to match the above five drawings. Please note that you may not turn over the objects. Satisfy yourself that they are different from the above.



Now let's do some sample problems. For each problem there is a primary object on the far left. You are to determine which two of four objects to the right are the same object given on the far left. In each problem always two of the four drawings are the same object as the one on the left. You are to put Xs in the boxes below the correct ones, and leave the incorrect ones blank. The first sample problem is done for you.

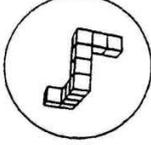
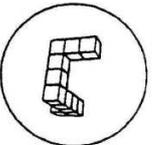
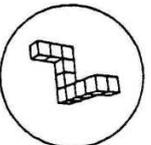
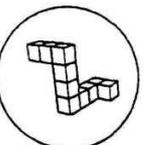
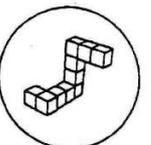


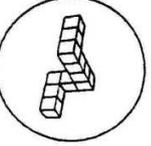
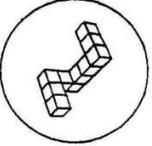
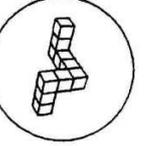
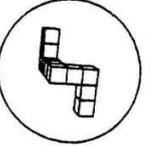
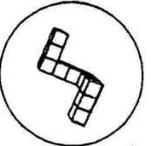
Go to the next page

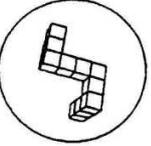
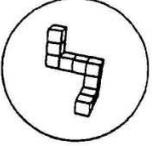
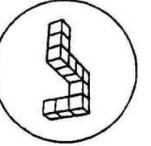
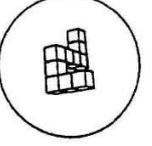
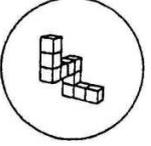
Adapted by S. G. Vandenberg, University of Colorado, July 15, 1971
Revised instructions by H. Crawford, U. of Wyoming, September, 1979
Images digitized and reprinted by Susanna Douglas, University of Texas, March, 1996

This is a public domain document and does not require copyright permission.

Do the rest of the sample problems yourself. Which two drawings of the four on the right show the same object as the one on the left? There are always two and only two correct answers for each problem. Put an X under the two correct drawings.

1.     

2.     

3.     

- Answers: 1. first and second drawings are correct
 2. first and third drawings are correct
 3. second and third drawings are correct

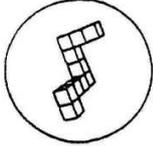
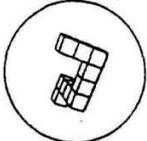
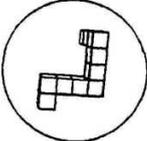
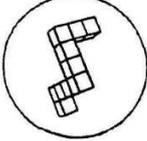
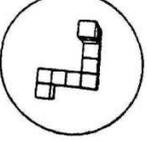
This test has two parts. You will have 3 minutes for each of the two parts. Each part has two pages. When you have finished Part I, STOP. Please do not go on to Part 2 until you are asked to do so. Remember: There are always two and only two correct answers for each item.

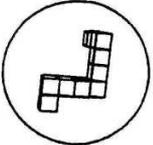
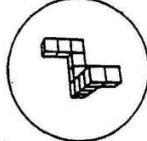
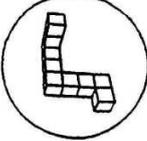
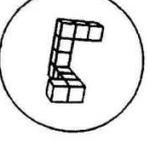
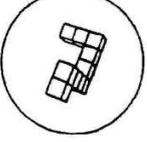
Work as quickly as you can without sacrificing accuracy. Your score on this test will reflect both the correct and incorrect responses. Therefore, it will not be to your advantage to guess unless you have some idea which choice is correct.

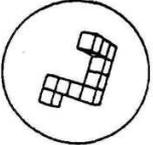
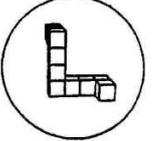
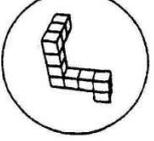
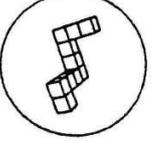
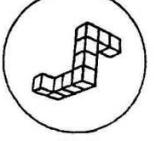
DO NOT TURN THIS PAGE UNTIL ASKED TO DO SO.

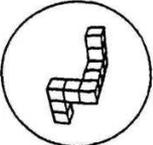
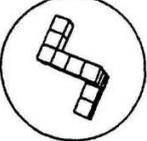
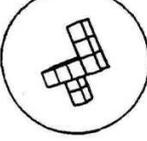
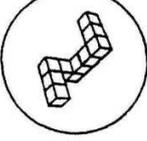
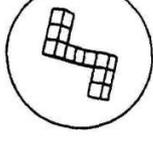
1.						<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.						<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.						<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.						<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.						<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

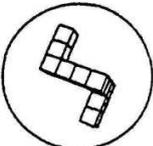
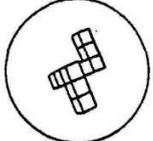
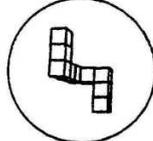
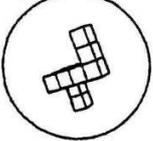
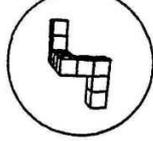
GO ON TO THE NEXT PAGE

6.     

7.     

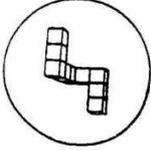
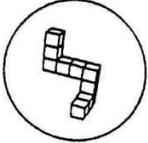
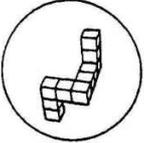
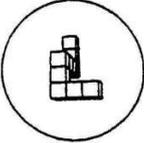
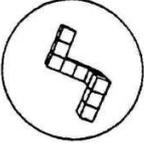
8.     

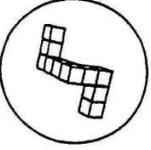
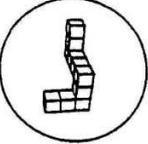
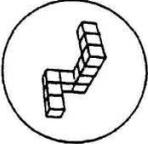
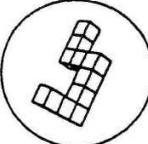
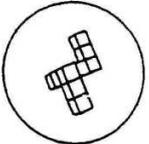
9.     

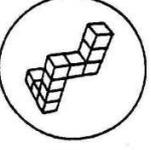
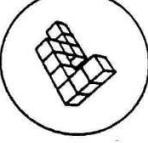
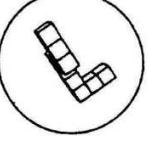
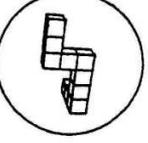
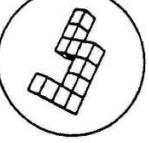
10.     

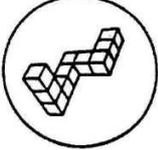
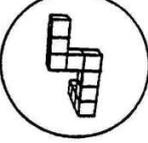
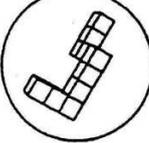
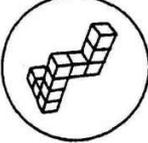
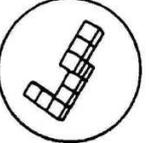
DO NOT TURN THIS PAGE UNTIL ASKED TO DO SO.

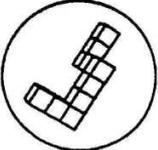
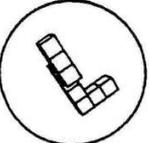
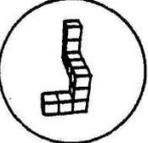
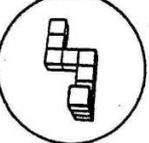
STOP

11.     

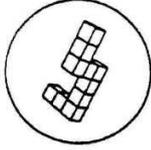
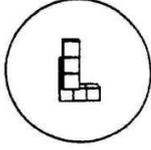
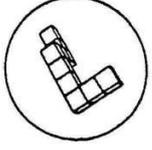
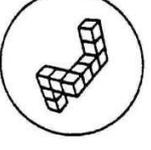
12.     

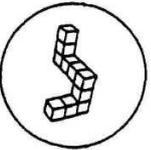
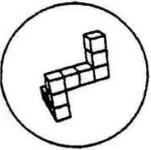
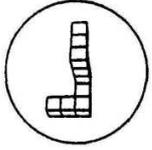
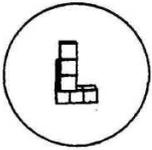
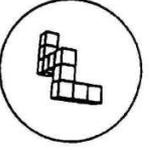
13.     

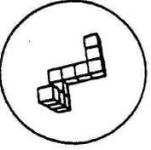
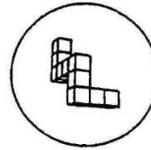
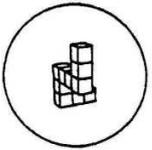
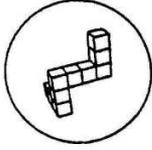
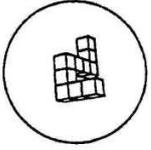
14.     

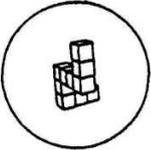
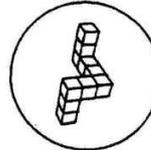
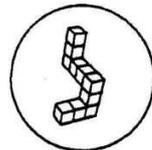
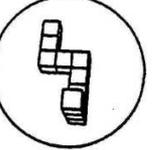
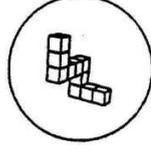
15.     

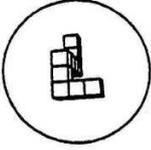
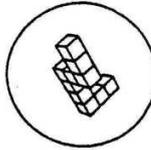
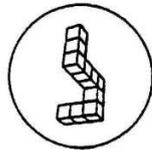
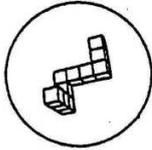
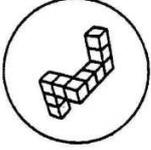
GO TO THE NEXT PAGE

16.     

17.     

18.     

19.     

20.     

DO NOT TURN THIS PAGE UNTIL ASKED TO DO SO. STOP

Appendix F: Summarized Subject Data

F.1 Cross-Session Data

Data Numbering Legend

Data #	Description
1	Gender (Female=0, Male=1)
2	REDACTED Age (Years)
3	REDACTED Height (m)
4	REDACTED Weight (kg)
5	REDACTED MIT Student (No=0, Yes=1)
6	REDACTED MIT Course Number (0=N/A)
7	REDACTED Laterality Quotient, Edinburgh (-100=Left, 100=Right)
8	REDACTED Visual Issues (No=0, Yes=1)
9	REDACTED Prescribed Lenses (No=0, Yes=1)
10	REDACTED Snellen Acuity Left (20/xx)
11	REDACTED Snellen Acuity Right (20/xx)
12	REDACTED Neuromotor Disease (No=0, Yes=1)
13	REDACTED Caffeine Consumer (No=0, Yes=1)
14	REDACTED Drug User (No=0, Yes=1)
15	Most Alert 0 h - 3 h (No=0, Yes=1)
16	Most Alert 3 h - 6 h (No=0, Yes=1)
17	Most Alert 6 h - 9 h (No=0, Yes=1)
18	Most Alert 9 h - 12 h (No=0, Yes=1)
19	Most Alert 12 h - 15 h (No=0, Yes=1)
20	Most Alert 15 h - 18 h (No=0, Yes=1)
21	Most Alert 18 h - 21 h (No=0, Yes=1)
22	Most Alert 21 h - 24 h (No=0, Yes=1)
23	Most Sleepy 0 h - 3 h (No=0, Yes=1)
24	Most Sleepy 3 h - 6 h (No=0, Yes=1)
25	Most Sleepy 6 h - 9 h (No=0, Yes=1)
26	Most Sleepy 9 h - 12 h (No=0, Yes=1)
27	Most Sleepy 12 h - 15 h (No=0, Yes=1)
28	Most Sleepy 15 h - 18 h (No=0, Yes=1)
29	Most Sleepy 18 h - 21 h (No=0, Yes=1)
30	Most Sleepy 21 h - 24 h (No=0, Yes=1)
31	Normal Sleep (hrs)
32	Virtual Environments (No=0, Yes=1)
33	Joysticks/Controller (No=0, Yes=1)
34	Bimanual Instruments (No=0, Yes=1)
35	Perspective Taking Ability Score
36	Mental Rotations Test Score
37	Gap Change BCCT tracking (1=Worse, 5=Better)
38	Gap Change BCCT predicting (1=Worse, 5=Better)

39	Gap Change BCCT bimanual indep. (1=Worse,5=Better)
40	Gap Change BCCT intrajoystick indep. (1=Worse,5=Better)
41	Gap Change TCT tracking (1=Worse,5=Better)
42	Gap Change TCT predicting (1=Worse,5=Better)
43	Gap Change TCT bimanual independ. (1=Worse,5=Better)
44	Gap Change TCT intrajoystick indep. (1=Worse,5=Better)
45	Gap Length (Days, [Session3Date-Session2Date])
46	PBCCT Average X Error (1=Doing Nothing)
47	PBCCT Average Y Error (1=Doing Nothing)
48	PBCCT Average Z Error (1=Doing Nothing)
49	PBCCT Average Roll Error (1=Doing Nothing)
50	PBCCT Average Pitch Error (1=Doing Nothing)
51	PBCCT Average Yaw Error (1=Doing Nothing)
52	PBCCT X->Y
53	PBCCT X->Z
54	PBCCT X->Roll
55	PBCCT X->Yaw
56	PBCCT Y->X
57	PBCCT Y->Z
58	PBCCT Y->Pitch
59	PBCCT Y->Yaw
60	PBCCT Roll->X
61	PBCCT Roll->Z
62	PBCCT Roll->Pitch
63	PBCCT Roll->Yaw
64	PBCCT Pitch->Y
65	PBCCT Pitch->Z
66	PBCCT Pitch->Roll
67	PBCCT Pitch->Yaw
68	SBCCT Average X Error (1=Doing Nothing)
69	SBCCT Average Y Error (1=Doing Nothing)
70	SBCCT Average Z Error (1=Doing Nothing)
71	SBCCT Average Roll Error (1=Doing Nothing)
72	SBCCT Average Pitch Error (1=Doing Nothing)
73	SBCCT Average Yaw Error (1=Doing Nothing)
74	SBCCT X->Y
75	SBCCT X->Z
76	SBCCT X->Roll
77	SBCCT X->Pitch
78	SBCCT X->Yaw
79	SBCCT Pitch->X
80	SBCCT Pitch->Y
81	SBCCT Pitch->Z
82	SBCCT Pitch->Roll
83	SBCCT Pitch->Yaw

Cross-Session Data

Data #	Subject Number																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	0	0	1	1	1	0	0	1	0	1	0	0	0	1	0	1	1
2																		
3																		
4																		
5																		
6																		
7																		
8																		
9																		
10																		
11																		
12																		
13																		
14																		
15	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0
18	1	1	1	1	0	1	1	0	0	0	0	0	1	0	1	1	0	1
19	0	1	1	0	1	0	1	1	1	1	0	1	0	1	0	0	0	1
20	0	1	1	0	1	1	1	1	1	1	0	1	1	1	0	0	0	1
21	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	1	1	0
22	0	0	0	0	1	1	0	1	0	0	1	0	1	1	1	0	1	0
23	0	1	1	0	0	1	1	0	1	0	0	0	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
25	1	0	0	0	1	0	1	1	1	1	0	1	1	1	0	0	0	1
26	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
27	0	0	0	1	0	1	0	0	0	0	1	0	0	0	1	0	1	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
29	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
30	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
31	5.5	7.5	7.5	5.5	7.5	5.5	7.5	7.5	7.5	5.5	7.5	5.5	5.5	7.5	7.5	7.5	7.5	7.5
32	1	1	1	1	1	1	1	0	1	0	0	1	1	1	0	0	1	1
33	1	0	1	1	1	1	1	0	1	0	1	1	0	1	0	1	0	1
34	0	1	1	1	1	0	0	1	0	1	0	1	1	0	1	1	1	1
35	11	20	12	20	21	17	27	25	13	12	19	13	18	1.4	28	21	21	26
36	1	17	5	0	29	28	30	13	14	17	12	8	24	17	33	21	26	32
37	4	4	4	3	3	4	3	3	3	2	4	3	2	4	4	3	2	3
38	4	3	4	3	3	3	3	4	4	2	3	3	3	3	4	4	3	3
39	4	3	4	3	3	3	3	3	4	2	4	4	2	3	3	3	3	3
40	4	3	4	3	3	3	3	3	4	2	4	3	2	3	3	3	3	2
41	5	3	4	5	3	3	4	3	5	4	4	5	3	5	3	3	3	3
42	5	3	5	5	3	3	4	4	5	4	4	4	3	5	4	3	3	3
43	5	3	5	5	3	3	4	3	5	4	4	5	3	5	2	3	3	3

44	5	3	5	5	3	3	4	3	5	4	4	4	3	5	3	3	3	3
45	11	13	12	11	12	10	12	12	12	13	12	13	14	11	12	12	12	12
46	0.12	0.16	0.11	0.09	0.08	0.07	0.07	0.11	0.18	0.19	0.14	0.17	0.38	0.22	0.11	0.08	0.11	0.08
47	0.12	0.13	0.09	0.13	0.10	0.07	0.10	0.42	0.28	0.16	0.15	0.18	0.23	0.24	0.14	0.10	0.17	0.10
48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
49	0.07	0.07	0.05	0.06	0.05	0.07	0.06	0.21	0.11	0.09	0.10	0.09	0.12	0.14	0.09	0.06	0.08	0.06
50	0.19	0.08	0.09	0.16	0.09	0.10	0.08	0.12	0.22	0.16	0.13	0.14	0.20	0.15	0.09	0.06	0.09	0.08
51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
52	0.16	0.07	0.24	0.32	0.02	0.13	0.11	0.12	0.08	0.08	0.07	0.22	0.21	0.15	0.07	0.04	0.08	0.26
53	0.01	0.10	0.08	0.04	0.00	0.02	0.10	0.02	0.04	0.00	0.02	0.12	0.00	0.01	0.02	0.02	0.01	0.00
54	0.20	0.03	0.16	0.09	0.06	0.08	0.03	0.03	0.03	0.05	0.03	0.05	0.12	0.04	0.14	0.04	0.01	0.25
55	0.02	0.25	0.20	0.02	0.10	0.10	0.02	0.05	0.05	0.03	0.04	0.04	0.03	0.03	0.03	0.01	0.06	0.19
56	0.88	0.25	0.82	0.38	0.46	0.91	0.50	0.33	0.77	0.44	0.89	0.17	0.33	0.22	0.69	0.22	0.12	0.19
57	0.13	0.06	0.09	0.10	0.05	0.11	0.15	0.14	0.28	0.01	0.14	0.04	0.06	0.03	0.13	0.00	0.04	0.01
58	0.02	0.06	0.13	0.07	0.07	0.02	0.04	0.05	0.22	0.04	0.09	0.08	0.09	0.14	0.04	0.02	0.12	0.08
59	0.03	0.07	0.14	0.07	0.06	0.05	0.12	0.04	0.12	0.04	0.05	0.08	0.05	0.09	0.02	0.09	0.01	0.11
60	0.28	0.28	0.12	0.30	0.06	0.36	0.39	0.05	0.24	0.20	0.47	0.18	0.11	0.17	0.12	0.07	0.15	0.26
61	0.02	0.02	0.03	0.09	0.02	0.05	0.13	0.06	0.08	0.01	0.05	0.03	0.03	0.03	0.04	0.00	0.01	0.01
62	0.12	0.36	0.47	0.51	0.16	0.08	0.31	0.32	0.62	0.43	0.11	0.38	0.25	0.39	0.16	0.40	0.42	0.29
63	0.63	0.06	1.01	0.65	0.19	0.51	0.80	0.09	1.38	0.72	0.46	0.66	0.32	0.52	0.06	0.34	0.15	0.62
64	0.08	0.33	0.10	0.58	0.05	0.04	0.46	0.12	0.30	0.14	0.66	0.34	0.09	0.22	0.04	0.12	0.19	0.21
65	0.02	0.06	0.03	0.03	0.00	0.01	0.05	0.04	0.07	0.00	0.11	0.06	0.00	0.01	0.01	0.01	0.00	0.00
66	0.11	0.16	0.13	0.22	0.17	0.35	0.12	0.16	0.12	0.40	0.24	0.24	0.65	0.44	0.14	0.19	0.21	0.13
67	0.11	0.54	0.39	0.17	0.14	0.06	0.12	0.20	0.24	0.18	0.08	0.07	0.14	0.06	0.06	0.06	0.13	0.26
68	0.22	0.26	0.37	0.33	0.19	0.24	0.18	0.22	0.37	0.69	0.25	0.40	0.57	0.41	0.23	0.19	0.25	0.16
69	0.57	0.28	0.50	0.42	0.13	0.28	0.13	0.24	0.22	0.22	0.33	1.00	0.70	0.39	0.23	0.18	0.27	0.10
70	0.45	0.58	0.30	0.17	0.07	0.09	0.27	0.39	0.52	0.18	0.26	0.32	0.10	0.13	0.12	0.41	0.48	0.33
71	0.15	0.09	0.04	0.07	0.05	0.06	0.26	0.09	0.17	0.13	0.08	0.10	0.08	0.09	0.09	0.16	0.09	0.06
72	0.35	0.23	0.23	0.29	0.17	0.30	0.16	0.27	0.47	0.29	0.34	0.40	0.22	0.34	0.21	0.22	0.20	0.14
73	0.10	0.09	0.04	0.15	0.06	0.13	0.36	0.14	0.16	0.17	0.08	0.16	0.11	0.12	0.08	0.24	0.12	0.09
74	0.06	0.04	0.27	0.08	0.02	0.09	0.02	0.03	0.07	0.06	0.09	0.08	0.12	0.10	0.09	0.05	0.03	0.04
75	0.01	0.01	0.01	0.01	0.00	0.01	0.00	0.01	0.04	0.00	0.02	0.01	0.01	0.00	0.00	0.02	0.00	0.01
76	0.01	0.01	0.01	0.02	0.05	0.01	0.02	0.02	0.02	0.01	0.03	0.01	0.02	0.04	0.00	0.02	0.01	0.01
77	0.00	0.01	0.01	0.02	0.05	0.02	0.02	0.03	0.02	0.01	0.02	0.02	0.01	0.01	0.01	0.04	0.02	0.01
78	0.01	0.02	0.02	0.03	0.05	0.01	0.02	0.02	0.02	0.02	0.02	0.05	0.03	0.03	0.01	0.01	0.03	0.01
79	0.00	0.00	0.01	0.01	0.00	0.00	0.01	0.01	0.02	0.01	0.01	0.02	0.02	0.03	0.01	0.00	0.00	0.00
80	0.01	0.00	0.03	0.01	0.02	0.01	0.01	0.01	0.07	0.02	0.01	0.03	0.02	0.10	0.01	0.02	0.01	0.01
81	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.02	0.01	0.01	0.00	0.00	0.00	0.00
82	0.06	0.17	0.06	0.12	0.17	0.24	0.26	0.05	0.14	0.30	0.17	0.11	0.52	0.42	0.17	0.16	0.24	0.16
83	0.10	0.17	0.34	0.15	0.11	0.04	0.05	0.12	0.22	0.17	0.04	0.20	0.12	0.11	0.07	0.05	0.19	0.10

F.2 Per Session Data

Data Numbering Legend

Data #	Description
1	Time (start + 1 hour, rounded to closest hour)
2	Wearing Lenses (0=No,1=Yes)
3	Caffeine Influence (0=No,1=Yes)
4	Drug Influence (0=No,1=Yes)
5	Last Sleep (hours)
6	Karolinska Sleepiness Scale (1=Alert,9=Sleepy)
7	Other Factors (0=No,1=Yes)
8	Discomfort: Nausea (1=None,5=Severe)
9	Discomfort: Dizziness (1=None,5=Severe)
10	Discomfort: Disorientation (1=None,5=Severe)
11	Discomfort: Eyestrain (1=None,5=Severe)
12	Discomfort: Blurred Vision (1=None,5=Severe)
13	Discomfort: Sweating (1=None,5=Severe)
14	Discomfort: Headache (1=None,5=Severe)
15	Discomfort: General (1=None,5=Severe)
16	Discomfort: Mental Fatigue (1=None,5=Severe)
17	Discomfort: Hand Fatigue (1=None,5=Severe)
18	Discomfort: Hand Pain (1=None,5=Severe)
19	Discomfort: Other (1=None,5=Severe)
20	Enjoyability of BCCT (1=Boring,5=Captivating)
21	Enjoyability of TCT (1=Boring,5=Captivating)
22	Ability in BCCT tracking (1=Unable,5=Expert)
23	Ability in BCCT predicting (1=Unable,5=Expert)
24	Ability in BCCT bimanual indep. (1=Unable,5=Expert)
25	Ability in BCCT intrajoystick indep. (1=Unable,5=Expert)
26	Ability in TCT tracking (1=Unable,5=Expert)
27	Ability in TCT predicting (1=Unable,5=Expert)
28	Ability in TCT bimanual independ. (1=Unable,5=Expert)
29	Ability in TCT intrajoystick indep. (1=Unable,5=Expert)
30	Difficulty of BCCT (1=Easy,5=Difficult)
31	Difficulty of TCT (1=Easy,5=Difficult)
32	BCCT Average Tracking Error, X
33	BCCT Average Tracking Error, Y
34	BCCT Average Tracking Error, Z
35	BCCT Average Tracking Error, Roll
36	BCCT Average Tracking Error, Pitch
37	BCCT Average Tracking Error, Yaw
38	BCCT Coupling X->Y (Drivers: X & Pitch)
39	BCCT Coupling X->Z (Drivers: X & Pitch)
40	BCCT Coupling X->Roll (Drivers: X & Pitch)
41	BCCT Coupling X->Pitch (Drivers: X & Yaw)
42	BCCT Coupling X->Yaw (Drivers: X & Pitch)
43	BCCT Coupling Y->X (Drivers: Y & Roll)
44	BCCT Coupling Y->Z (Drivers: Y & Roll)
45	BCCT Coupling Y->Roll (Drivers: Y & Pitch)
46	BCCT Coupling Y->Pitch (Drivers: Y & Roll)

47	BCCT Coupling Y->Yaw (Drivers: Y & Roll)
48	BCCT Coupling Z->X (Drivers: Z & Yaw)
49	BCCT Coupling Z->Y (Drivers: Z & Yaw)
50	BCCT Coupling Z->Roll (Drivers: Z & Yaw)
51	BCCT Coupling Z->Pitch (Drivers: Z & Yaw)
52	BCCT Coupling Z->Yaw (Drivers: Z & Roll)
53	BCCT Coupling Roll->X (Drivers: Y & Roll)
54	BCCT Coupling Roll->Y (Drivers: Z & Roll)
55	BCCT Coupling Roll->Z (Drivers: Y & Roll)
56	BCCT Coupling Roll->Pitch (Drivers: Y & Roll)
57	BCCT Coupling Roll->Yaw (Drivers: Y & Roll)
58	BCCT Coupling Pitch->X (Drivers: Y & Pitch)
59	BCCT Coupling Pitch->Y (Drivers: X & Pitch)
60	BCCT Coupling Pitch->Z (Drivers: X & Pitch)
61	BCCT Coupling Pitch->Roll (Drivers: X & Pitch)
62	BCCT Coupling Pitch->Yaw (Drivers: X & Pitch)
63	BCCT Coupling Yaw->X (Drivers: Z & Yaw)
64	BCCT Coupling Yaw->Y (Drivers: Z & Yaw)
65	BCCT Coupling Yaw->Z (Drivers: X & Yaw)
66	BCCT Coupling Yaw->Roll (Drivers: Z & Yaw)
67	BCCT Coupling Yaw->Pitch (Drivers: Z & Yaw)
68	BCCT Coupling Redundant X->Y (Drivers: X & Yaw)
69	BCCT Coupling Redundant X->Z (Drivers: X & Yaw)
70	BCCT Coupling Redundant X->Roll (Drivers: X & Yaw)
71	BCCT Coupling Redundant Y->X (Drivers: Y & Pitch)
72	BCCT Coupling Redundant Y->Z (Drivers: Y & Pitch)
73	BCCT Coupling Redundant Y->Yaw (Drivers: Y & Pitch)
74	BCCT Coupling Redundant Z->X (Drivers: Z & Roll)
75	BCCT Coupling Redundant Z->Y (Drivers: Z & Roll)
76	BCCT Coupling Redundant Z->Pitch (Drivers: Z & Roll)
77	BCCT Coupling Redundant Roll->X (Drivers: Z & Roll)
78	BCCT Coupling Redundant Roll->Pitch (Drivers: Z & Roll)
79	BCCT Coupling Redundant Roll->Yaw (Drivers: Z & Roll)
80	BCCT Coupling Redundant Pitch->Z (Drivers: Y & Pitch)
81	BCCT Coupling Redundant Pitch->Roll (Drivers: Y & Pitch)
82	BCCT Coupling Redundant Pitch->Yaw (Drivers: Y & Pitch)
83	BCCT Coupling Redundant Yaw->Y (Drivers: X & Yaw)
84	BCCT Coupling Redundant Yaw->Roll (Drivers: X & Yaw)
85	BCCT Coupling Redundant Yaw->Pitch (Drivers: X & Yaw)
86	TCT Total Captures (out of 12)
87	TCT Average Time to Capture (out of total captures)
88	TCT Average Positional Error on Capture
89	TCT Average Angular Error on Capture
90	TCT Total Collisions (in all 12 trials)
91	TCT Total Failed Grapples (in all 12 trials)

Note: All BCCT coupling values listed here are total measured values, prior to estimation of first-order effects only.

Session 1 Data

Data	Subject Number																	
#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	19	18	17	20	14	8	15	17	11	19	21	14	11	20	22	11	10	15
2	0	1	1	1	0	1	1	0	0	1	1	0	1	1	1	0	1	1
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1
4	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0
5	5.5	7.5	7.5	7.5	5.5	7.5	7.5	7.5	7.5	5.5	7.5	7.5	5.5	7.5	9.5	7.5	7.5	7.5
6	3	3	1	1	3	1	3	3	5	3	3	3	4	3	3	2	3	3
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	2	1	1	2	1	1	1
10	1	2	1	1	1	1	1	1	1	1	1	2	1	1	1	1	2	1
11	1	3	1	4	2	4	1	2	3	3	1	1	1	3	2	1	2	3
12	1	1	1	1	1	1	1	1	1	2	1	3	1	1	3	1	1	2
13	1	1	1	1	1	2	1	2	1	1	1	1	1	1	1	1	1	1
14	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	2	2	1	1	1	2	1	1	1	2	1	3	2	1	3	1	1	1
16	1	2	1	2	1	1	1	2	2	2	1	2	2	1	2	1	2	2
17	1	2	1	3	2	5	2	3	2	3	2	3	3	4	3	2	3	3
18	2	1	1	1	2	4	1	2	1	1	2	2	2	4	2	2	3	1
19	1	1	1	1	1	1	1	1	1	1	1	0	1	2	1	1	1	1
20	4	3	3	4	1	3	4	1	3	3	5	2	3	3	2	3	2	2
21	4	4	3	5	4	4	5	3	4	5	5	3	4	5	4	5	4	4
22	3	3	3	4	3	4	4	2	3	2	3	2	3	3	3	3	3	4
23	3	2	4	3	4	2	3	2	3	3	4	2	2	1	2	2	2	4
24	3	3	3	4	5	3	4	3	3	3	3	3	3	3	3	3	3	4
25	3	1	4	3	5	2	4	2	2	2	2	2	2	2	4	2	4	3
26	3	4	4	3	5	4	4	4	3	4	3	3	3	4	4	3	4	3
27	4	3	3	4	5	4	4	4	3	3	4	2	3	4	3	3	4	3
28	3	4	3	4	5	4	4	3	4	5	3	4	3	2	4	3	4	4
29	3	3	4	2	5	4	4	2	3	3	3	2	3	1	5	3	4	2
30	3	4	3	2	4	5	4	4	4	4	4	4	3	2	4	3	4	2
31	3	3	3	3	1	3	2	2	3	2	4	3	3	4	2	2	3	4
32	0.49	0.40	0.37	0.48	0.23	0.29	0.31	0.35	0.71	0.60	0.38	0.60	0.84	0.57	0.46	0.34	0.44	0.32
33	0.82	0.60	0.57	0.99	0.45	0.43	0.51	0.56	1.30	0.75	0.71	0.91	0.84	0.77	0.64	0.45	0.64	0.48
34	1.68	1.67	1.36	1.63	1.15	1.49	1.25	1.46	2.27	1.43	1.54	1.74	1.88	1.66	1.10	1.19	1.90	0.88
35	0.21	0.18	0.16	0.18	0.13	0.16	0.16	0.20	0.30	0.22	0.31	0.50	0.39	0.39	0.19	0.13	0.26	0.17
36	0.75	0.42	0.63	0.95	0.30	0.67	0.43	0.73	1.01	0.65	0.71	0.63	0.79	0.67	0.54	0.31	0.44	0.57
37	0.60	0.43	0.44	0.59	0.47	0.64	0.58	0.56	0.83	0.56	0.53	0.75	0.87	0.82	0.46	0.37	0.68	0.32
38	0.20	0.29	0.42	0.25	0.12	0.12	0.14	0.09	0.12	0.34	0.23	0.37	0.23	0.15	0.09	0.15	0.07	0.36
39	0.10	0.05	0.11	0.03	0.02	0.08	0.09	0.03	0.15	0.01	0.23	0.09	0.04	0.00	0.01	0.06	0.01	0.04
40	0.12	0.46	0.23	0.05	0.03	0.24	0.04	0.04	0.03	0.15	0.04	0.05	0.04	0.05	0.21	0.10	0.03	0.24
41	0.03	0.03	0.05	0.03	0.01	0.02	0.01	0.02	0.01	0.02	0.04	0.02	0.07	0.04	0.02	0.01	0.02	0.01
42	0.05	0.23	0.26	0.05	0.03	0.34	0.10	0.24	0.05	0.05	0.10	0.05	0.03	0.03	0.03	0.03	0.03	0.06
43	0.34	0.08	0.45	0.24	0.90	0.60	0.37	0.38	0.48	0.33	0.66	0.17	0.19	0.05	0.33	0.11	0.08	0.22
44	0.63	0.00	0.12	0.10	0.09	0.22	0.17	0.27	0.20	0.03	0.29	0.07	0.05	0.01	0.05	0.01	0.03	0.15

45	0.03	0.04	0.11	0.04	0.02	0.09	0.02	0.03	0.04	0.04	0.05	0.11	0.15	0.11	0.04	0.05	0.07	0.07
46	0.09	0.12	0.11	0.07	0.04	0.21	0.03	0.10	0.06	0.04	0.05	0.26	0.07	0.13	0.03	0.05	0.14	0.11
47	0.07	0.12	0.17	0.11	0.05	0.11	0.10	0.06	0.12	0.11	0.08	0.35	0.02	0.19	0.03	0.12	0.05	0.07
48	0.09	0.30	0.43	0.36	0.14	0.31	0.07	0.11	0.30	0.68	0.18	0.06	0.14	0.38	0.18	0.12	0.13	0.09
49	0.12	0.18	0.25	0.24	0.19	0.19	0.14	0.09	0.26	0.19	0.04	0.17	0.21	0.20	0.04	0.07	0.11	0.11
50	0.04	0.04	0.07	0.01	0.01	0.01	0.01	0.04	0.03	0.03	0.14	0.03	0.02	0.01	0.01	0.01	0.04	0.03
51	0.05	0.02	0.04	0.03	0.02	0.06	0.01	0.03	0.05	0.12	0.05	0.02	0.06	0.06	0.02	0.03	0.03	0.01
52	0.15	0.09	0.13	0.13	0.05	0.13	0.09	0.05	0.39	0.13	0.11	1.15	0.05	0.19	0.04	0.11	0.28	0.10
53	0.27	0.19	0.11	0.52	0.11	0.39	0.59	0.07	0.44	0.26	0.34	0.27	0.15	0.04	0.10	0.17	0.08	0.16
54	0.10	0.35	0.06	0.63	0.07	0.16	0.20	0.03	0.18	0.10	0.17	0.11	0.14	0.11	0.05	0.03	0.02	0.05
55	0.18	0.01	0.03	0.21	0.05	0.11	0.15	0.04	0.09	0.02	0.12	0.02	0.02	0.01	0.01	0.01	0.02	0.03
56	0.06	0.51	0.82	0.49	0.13	0.54	0.27	0.40	0.55	0.31	0.12	0.46	0.22	0.53	0.18	0.48	0.43	0.38
57	0.69	0.31	0.92	1.19	0.11	0.83	0.66	0.18	0.86	0.64	0.97	2.32	0.09	0.49	0.10	0.49	0.36	0.21
58	0.39	0.23	0.20	0.21	0.29	0.24	0.27	0.21	0.46	0.07	0.31	0.27	0.14	0.14	0.22	0.08	0.07	0.08
59	0.54	0.83	0.26	2.14	0.12	0.10	0.57	0.20	0.94	0.94	1.12	0.51	0.27	0.75	0.04	0.62	0.32	0.75
60	0.18	0.06	0.05	0.17	0.03	0.03	0.05	0.03	0.23	0.02	0.40	0.03	0.02	0.01	0.01	0.08	0.01	0.04
61	0.25	0.09	0.38	0.09	0.03	0.28	0.02	0.22	0.09	0.15	0.20	0.25	0.39	0.26	0.11	0.15	0.16	0.12
62	0.16	0.36	1.36	0.14	0.09	0.28	0.10	0.23	0.15	0.13	0.14	0.12	0.06	0.09	0.02	0.08	0.09	0.07
63	0.60	0.85	0.38	0.94	0.24	0.16	0.20	0.11	1.02	0.48	1.44	0.30	0.30	0.93	0.33	0.64	0.17	0.25
64	0.23	0.43	0.30	0.15	0.12	0.16	0.18	0.06	0.93	0.36	0.03	0.39	0.30	1.04	0.07	0.11	0.08	0.12
65	0.10	0.07	0.06	0.19	0.01	0.03	0.08	0.02	0.26	0.05	0.23	0.11	0.07	0.03	0.01	0.04	0.03	0.04
66	0.08	0.02	0.14	0.00	0.01	0.05	0.03	0.05	0.11	0.04	0.15	0.03	0.05	0.04	0.03	0.01	0.10	0.05
67	0.09	0.03	0.04	0.04	0.05	0.07	0.03	0.05	0.15	0.02	0.09	0.08	0.14	0.13	0.04	0.02	0.04	0.04
68	0.08	0.27	0.42	0.08	0.08	0.08	0.10	0.01	0.03	0.30	0.04	0.34	0.18	0.08	0.04	0.04	0.12	0.15
69	0.06	0.07	0.09	0.12	0.01	0.12	0.08	0.01	0.04	0.12	0.15	0.14	0.01	0.01	0.00	0.06	0.01	0.15
70	0.04	0.02	0.03	0.01	0.01	0.02	0.01	0.03	0.02	0.04	0.03	0.01	0.05	0.01	0.01	0.01	0.04	0.01
71	0.45	0.24	0.68	0.29	0.41	0.68	0.56	0.40	0.40	0.21	0.58	0.28	0.41	0.33	0.74	0.14	0.08	0.25
72	0.16	0.08	0.12	0.12	0.23	0.12	0.22	0.19	0.11	0.00	0.49	0.08	0.12	0.03	0.11	0.04	0.05	0.17
73	0.05	0.04	0.34	0.03	0.04	0.04	0.03	0.04	0.05	0.04	0.03	0.09	0.05	0.05	0.02	0.02	0.05	0.10
74	0.13	0.65	0.63	0.19	0.31	0.14	0.08	0.08	0.27	0.52	0.25	0.23	0.49	0.46	0.17	0.21	0.17	0.17
75	0.30	0.12	0.18	0.18	0.21	0.11	0.48	0.03	0.16	0.32	0.11	0.18	0.22	0.15	0.08	0.04	0.20	0.06
76	0.07	0.11	0.11	0.07	0.04	0.04	0.04	0.05	0.10	0.20	0.10	0.42	0.11	0.14	0.09	0.03	0.34	0.04
77	0.13	0.80	0.20	0.49	0.11	0.47	0.09	0.03	0.38	0.20	0.43	0.07	0.24	0.27	0.19	0.26	0.07	0.12
78	0.15	0.28	0.43	0.46	0.08	0.19	0.36	0.32	0.41	0.35	0.07	0.52	0.12	0.43	0.23	0.24	0.36	0.39
79	0.61	0.29	1.23	0.96	0.15	0.61	0.60	0.13	0.96	0.31	0.69	1.28	0.14	0.83	0.09	1.08	0.29	0.85
80	0.23	0.14	0.07	0.21	0.12	0.05	0.16	0.09	0.25	0.01	0.50	0.08	0.05	0.06	0.05	0.02	0.04	0.07
81	0.08	0.22	0.17	0.09	0.06	0.23	0.03	0.09	0.09	0.14	0.07	0.22	0.25	0.45	0.02	0.15	0.16	0.10
82	0.15	0.09	1.06	0.15	0.07	0.12	0.16	0.11	0.19	0.13	0.08	0.12	0.08	0.29	0.08	0.04	0.09	0.24
83	0.11	0.23	0.21	0.08	0.03	0.04	0.05	0.05	0.31	0.08	0.09	0.20	0.20	0.11	0.04	0.04	0.12	0.05
84	0.13	0.08	0.09	0.03	0.03	0.02	0.03	0.05	0.13	0.04	0.11	0.04	0.16	0.06	0.02	0.02	0.15	0.03
85	0.05	0.05	0.07	0.04	0.02	0.04	0.02	0.03	0.13	0.04	0.09	0.04	0.20	0.08	0.09	0.01	0.07	0.03
86	9	11	5	8	12	12	8	12	9	10	9	4	12	9	11	12	12	8
87	36.1	40.8	42.7	36.9	36.0	39.7	35.3	35.8	39.7	45.6	40.8	40.3	34.8	36.6	37.5	36.9	31.3	44.9
88	0.15	0.23	0.26	0.18	0.16	0.17	0.19	0.23	0.24	0.23	0.16	0.19	0.21	0.15	0.17	0.20	0.23	0.22
89	1.96	1.54	1.42	1.35	0.92	1.37	1.14	1.48	1.85	1.77	1.56	3.20	1.74	2.11	1.54	2.30	1.74	1.60
90	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Session 2 Data

Data #	Subject Number																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	21	18	17	20	14	9	15	17	12	12	23	20	11	18	22	11	11	13
2	0	1	1	1	0	1	1	0	0	1	1	0	1	1	1	0	1	1
3	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1
4	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0
5	5.5	7.5	5.5	7.5	5.5	7.5	7.5	7.5	5.5	5.5	9.5	7.5	5.5	5.5	5.5	7.5	5.5	7.5
6	3	3	1	1	4	1	3	3	5	4	3	2	2	4	2	2	7	3
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	1	1	1	1	2	1	2	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1
10	1	2	1	1	1	1	1	1	1	1	1	1	1	4	1	1	1	1
11	1	4	1	3	1	2	1	3	3	2	1	1	1	4	2	1	1	2
12	1	2	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	2	1	1	1	1	1	1	1	3	1	1	1	3	2	1	1	1
16	1	3	1	2	1	2	1	2	2	2	1	1	2	3	2	1	1	2
17	1	2	1	1	1	3	3	2	1	3	1	2	2	4	3	2	2	2
18	1	1	1	1	1	3	1	2	1	2	1	1	1	4	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	1
20	4	3	3	3	2	3	4	1	3	2	4	3	3	1	2	3	1	2
21	4	4	3	4	3	2	5	2	4	4	5	4	4	5	4	5	3	3
22	3	3	3	4	3	4	4	2	3	2	4	4	3	3	3	3	3	4
23	4	2	4	3	4	2	3	2	3	2	4	3	3	4	2	3	3	4
24	4	3	5	4	5	4	3	3	3	2	3	3	3	1	3	4	4	4
25	3	1	4	4	5	4	4	2	3	2	3	4	3	4	4	3	2	4
26	4	4	4	3	5	4	5	4	4	4	4	3	3	4	4	4	4	4
27	3	3	5	4	5	2	4	3	4	3	4	3	3	4	4	4	3	4
28	4	4	4	4	5	4	5	3	4	5	4	4	3	4	4	4	4	4
29	3	2	3	3	5	4	4	2	4	4	4	2	3	2	4	4	3	4
30	3	4	3	2	3	4	3	4	3	5	4	2	3	4	4	4	3	2
31	2	3	3	3	1	3	2	2	2	2	3	3	3	3	3	3	2	3
32	0.35	0.39	0.35	0.32	0.24	0.29	0.23	0.33	0.46	0.52	0.37	0.64	0.62	0.62	0.31	0.30	0.40	0.23
33	0.71	0.52	0.44	0.72	0.43	0.41	0.40	0.59	0.88	0.66	0.60	0.67	0.72	0.79	0.55	0.36	0.63	0.36
34	1.32	1.40	1.16	1.18	1.39	1.13	1.20	1.43	1.65	1.57	1.39	1.57	1.55	2.13	0.96	1.07	1.81	0.75
35	0.17	0.15	0.16	0.18	0.13	0.13	0.14	0.20	0.34	0.24	0.22	0.30	0.32	0.37	0.16	0.11	0.23	0.15
36	0.71	0.43	0.40	0.70	0.41	0.45	0.35	0.58	0.83	0.47	0.42	0.51	0.49	0.73	0.38	0.21	0.44	0.29
37	0.45	0.51	0.60	0.49	0.36	0.47	0.50	0.52	0.60	0.47	0.48	0.49	0.64	0.93	0.34	0.46	0.60	0.27
38	0.27	0.30	0.16	0.10	0.08	0.08	0.07	0.07	0.06	0.20	0.10	0.12	0.26	0.10	0.09	0.17	0.12	0.17
39	0.03	0.02	0.07	0.02	0.02	0.01	0.04	0.01	0.01	0.01	0.04	0.07	0.04	0.00	0.01	0.06	0.01	0.04
40	0.11	0.20	0.19	0.09	0.02	0.23	0.02	0.04	0.02	0.13	0.09	0.04	0.05	0.06	0.11	0.07	0.01	0.25
41	0.04	0.01	0.06	0.04	0.03	0.01	0.02	0.01	0.01	0.01	0.01	0.03	0.02	0.04	0.02	0.01	0.02	0.02
42	0.05	0.17	0.42	0.05	0.04	0.59	0.09	0.16	0.01	0.03	0.14	0.03	0.04	0.05	0.07	0.02	0.05	0.20
43	0.52	0.14	0.32	0.49	0.49	0.64	0.39	0.39	0.41	0.28	0.71	0.07	0.27	0.14	0.34	0.14	0.22	0.12
44	0.45	0.02	0.07	0.17	0.08	0.07	0.06	0.22	0.01	0.01	0.40	0.07	0.02	0.00	0.02	0.02	0.14	0.10

45	0.03	0.03	0.03	0.02	0.04	0.08	0.02	0.06	0.02	0.07	0.06	0.03	0.08	0.14	0.03	0.03	0.06	0.05
46	0.04	0.07	0.10	0.06	0.05	0.12	0.05	0.08	0.06	0.06	0.10	0.17	0.02	0.13	0.04	0.03	0.15	0.05
47	0.09	0.08	0.19	0.14	0.03	0.10	0.07	0.03	0.12	0.07	0.05	0.23	0.01	0.10	0.02	0.07	0.03	0.07
48	0.06	0.41	0.12	0.35	0.12	0.25	0.25	0.20	0.36	0.66	0.24	0.20	0.59	0.51	0.19	0.31	0.16	0.29
49	0.25	0.31	0.23	0.23	0.16	0.05	0.09	0.12	0.14	0.12	0.10	0.09	0.23	0.23	0.10	0.08	0.06	0.24
50	0.05	0.02	0.09	0.03	0.02	0.06	0.00	0.07	0.07	0.02	0.06	0.01	0.03	0.03	0.01	0.02	0.02	0.02
51	0.02	0.02	0.04	0.06	0.07	0.03	0.03	0.02	0.08	0.05	0.07	0.03	0.06	0.07	0.03	0.02	0.03	0.01
52	0.11	0.04	0.09	0.15	0.04	0.07	0.09	0.06	0.40	0.12	0.16	0.22	0.07	0.11	0.01	0.17	0.09	0.05
53	0.16	0.25	0.07	0.37	0.10	0.33	1.02	0.07	0.33	0.21	0.21	0.11	0.12	0.07	0.08	0.08	0.07	0.14
54	0.05	0.09	0.16	0.21	0.06	0.08	0.29	0.06	0.15	0.06	0.26	0.05	0.14	0.11	0.05	0.06	0.08	0.09
55	0.18	0.01	0.04	0.09	0.03	0.03	0.06	0.03	0.01	0.01	0.08	0.03	0.02	0.00	0.02	0.01	0.03	0.04
56	0.09	0.39	0.89	0.58	0.14	0.26	0.30	0.29	0.57	0.39	0.22	0.26	0.13	0.40	0.13	0.67	0.46	0.37
57	1.18	0.20	0.94	1.32	0.07	0.90	0.61	0.10	0.98	0.35	0.64	1.62	0.03	0.37	0.07	0.87	0.10	0.19
58	0.00	0.15	0.12	0.60	0.12	0.12	0.30	0.16	0.58	0.12	0.20	0.20	0.12	0.14	0.11	0.03	0.22	0.12
59	0.33	0.38	0.17	0.50	0.05	0.03	0.27	0.05	0.34	0.33	0.65	0.25	0.10	0.14	0.03	0.27	0.13	0.42
60	0.11	0.02	0.01	0.03	0.02	0.01	0.02	0.01	0.03	0.01	0.17	0.04	0.02	0.01	0.01	0.04	0.01	0.03
61	0.05	0.18	0.37	0.12	0.14	0.28	0.05	0.07	0.10	0.29	0.22	0.11	0.37	0.32	0.10	0.17	0.16	0.16
62	0.21	0.60	0.82	0.15	0.21	0.26	0.24	0.15	0.16	0.14	0.14	0.11	0.07	0.13	0.06	0.09	0.15	0.26
63	0.27	0.25	0.17	1.18	0.17	0.13	0.20	0.18	0.32	0.42	0.84	0.22	0.41	0.39	0.28	0.36	0.30	0.48
64	0.08	0.15	0.08	0.21	0.07	0.04	0.06	0.11	0.09	0.12	0.11	0.09	0.18	0.26	0.08	0.06	0.17	0.24
65	0.04	0.01	0.03	0.10	0.01	0.01	0.03	0.03	0.03	0.00	0.09	0.01	0.01	0.01	0.01	0.06	0.03	0.05
66	0.11	0.02	0.23	0.07	0.06	0.03	0.02	0.05	0.12	0.02	0.21	0.02	0.08	0.03	0.03	0.01	0.07	0.03
67	0.03	0.03	0.07	0.06	0.14	0.03	0.04	0.02	0.06	0.03	0.12	0.04	0.07	0.10	0.03	0.02	0.07	0.06
68	0.10	0.25	0.34	0.10	0.07	0.14	0.06	0.19	0.07	0.24	0.03	0.11	0.16	0.07	0.02	0.08	0.13	0.03
69	0.06	0.01	0.03	0.05	0.01	0.00	0.05	0.02	0.00	0.00	0.05	0.01	0.01	0.00	0.00	0.07	0.02	0.08
70	0.05	0.01	0.06	0.00	0.01	0.01	0.01	0.04	0.01	0.01	0.05	0.01	0.01	0.02	0.00	0.01	0.01	0.00
71	0.00	0.13	0.35	0.27	0.47	0.61	0.38	0.38	0.49	0.19	0.52	0.26	0.21	0.18	0.47	0.20	0.61	0.42
72	0.08	0.01	0.09	0.10	0.17	0.01	0.09	0.15	0.06	0.00	0.51	0.05	0.03	0.01	0.02	0.02	0.15	0.07
73	0.09	0.10	0.14	0.05	0.02	0.03	0.15	0.05	0.04	0.06	0.05	0.03	0.05	0.08	0.04	0.04	0.05	0.06
74	0.17	0.14	0.10	0.17	0.17	0.27	0.28	0.07	0.22	0.44	0.10	0.41	0.34	0.34	0.20	0.08	0.11	0.19
75	0.21	0.21	0.09	0.15	0.10	0.04	0.23	0.04	0.09	0.17	0.05	0.06	0.33	0.11	0.09	0.05	0.05	0.10
76	0.05	0.14	0.06	0.06	0.05	0.04	0.03	0.07	0.30	0.14	0.01	0.07	0.03	0.29	0.05	0.03	0.08	0.04
77	0.13	0.18	0.29	0.43	0.10	0.43	0.66	0.10	0.25	0.19	0.65	0.12	0.18	0.28	0.24	0.13	0.11	0.22
78	0.33	0.42	0.37	0.50	0.24	0.06	0.30	0.31	0.49	0.30	0.03	0.18	0.25	0.46	0.12	0.40	0.37	0.35
79	1.04	0.14	1.07	1.13	0.11	0.72	0.65	0.21	0.56	0.38	1.12	0.68	0.12	0.20	0.06	0.57	0.22	0.33
80	0.07	0.02	0.04	0.27	0.07	0.01	0.12	0.10	0.17	0.01	0.28	0.05	0.02	0.04	0.01	0.01	0.07	0.07
81	0.06	0.10	0.11	0.04	0.10	0.34	0.04	0.09	0.13	0.20	0.27	0.05	0.39	0.30	0.13	0.16	0.17	0.13
82	0.26	0.38	0.99	0.21	0.05	0.11	0.45	0.11	0.14	0.19	0.27	0.10	0.21	0.31	0.10	0.08	0.12	0.40
83	0.05	0.11	0.06	0.08	0.04	0.03	0.04	0.12	0.08	0.07	0.06	0.05	0.04	0.42	0.01	0.03	0.06	0.03
84	0.20	0.05	0.11	0.02	0.04	0.02	0.02	0.07	0.06	0.03	0.13	0.02	0.02	0.15	0.01	0.01	0.05	0.02
85	0.04	0.04	0.07	0.03	0.05	0.08	0.01	0.03	0.03	0.03	0.03	0.03	0.07	0.23	0.07	0.03	0.09	0.02
86	10	12	7	10	12	12	11	12	12	12	11	1	10	11	12	12	12	12
87	38.7	39.3	35.1	34.4	33.0	40.5	36.4	32.4	35.1	38.9	35.4	51.0	36.3	34.8	33.9	32.4	33.3	35.2
88	0.17	0.23	0.26	0.17	0.15	0.19	0.20	0.23	0.25	0.23	0.18	0.27	0.18	0.16	0.18	0.19	0.24	0.26
89	1.79	1.23	1.97	1.45	0.64	1.37	1.58	1.67	1.69	1.55	1.19	4.65	1.24	2.05	1.39	1.70	1.76	1.43
90	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0
91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Session 3 Data

Data	Subject Number																	
#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	21	21	17	20	14	9	15	15	12	22	22	23	13	19	22	11	10	15
2	0	1	1	1	0	1	1	0	0	1	1	0	1	1	1	0	1	1
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1
4	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0
5	5.5	7.5	7.5	7.5	7.5	7.5	7.5	7.5	5.5	5.5	7.5	5.5	5.5	5.5	5.5	7.5	9.5	7.5
6	3	3	2	1	3	1	3	4	7	6	3	3	2	5	5	3	3	3
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	1	1	1	1	3	1	3	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1
10	1	2	1	1	1	1	1	1	1	1	1	2	1	2	1	1	2	1
11	1	2	1	4	1	2	1	3	3	3	1	1	1	1	1	1	2	2
12	1	2	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1
13	1	1	1	1	1	1	1	2	1	1	1	2	1	1	1	1	1	1
14	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1	1
15	2	2	1	2	1	2	1	2	1	1	1	3	1	2	2	1	1	1
16	1	1	1	1	1	2	1	1	2	2	1	2	1	3	2	1	1	2
17	1	2	1	3	2	3	3	3	2	2	2	2	2	4	2	2	3	3
18	1	1	1	3	1	2	3	1	1	1	1	3	1	3	1	1	2	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	4	2	3	2	2	4	4	1	3	2	4	2	3	1	3	3	2	2
21	4	3	3	4	3	3	5	3	4	4	5	5	4	5	4	5	4	3
22	4	2	4	4	3	4	4	3	3	3	4	3	3	2	4	3	3	4
23	4	2	4	4	4	4	4	2	3	2	4	3	3	2	3	3	1	4
24	4	2	4	5	5	4	4	2	3	3	3	4	3	1	4	4	4	4
25	4	2	4	4	5	4	4	2	4	3	4	3	3	4	4	4	4	3
26	4	3	4	4	5	4	5	3	5	4	5	4	4	4	5	4	4	4
27	4	3	4	4	5	4	5	4	5	4	5	3	3	2	5	4	3	4
28	4	3	5	5	5	4	5	3	5	4	5	4	3	5	4	4	4	4
29	4	2	5	5	5	4	5	2	5	4	5	3	3	2	5	4	3	4
30	2	3	2	3	2	4	3	4	3	4	3	3	3	4	4	4	4	2
31	2	3	1	3	1	3	2	3	2	2	2	2	3	2	2	2	2	3
32	0.35	0.37	0.29	0.27	0.19	0.22	0.22	0.29	0.44	0.42	0.33	0.39	0.80	0.45	0.29	0.27	0.26	0.23
33	0.61	0.50	0.48	0.57	0.39	0.36	0.40	0.47	0.70	0.61	0.49	0.62	0.80	0.61	0.52	0.39	0.47	0.35
34	1.34	1.36	1.29	1.07	1.02	1.19	1.40	1.29	1.45	1.32	1.18	1.47	1.58	1.49	1.03	1.06	1.27	0.98
35	0.15	0.14	0.11	0.14	0.10	0.13	0.13	0.17	0.21	0.21	0.17	0.23	0.29	0.32	0.15	0.10	0.15	0.13
36	0.59	0.37	0.43	0.52	0.31	0.35	0.30	0.40	0.59	0.38	0.34	0.37	0.48	0.43	0.33	0.19	0.24	0.27
37	0.38	0.43	0.42	0.32	0.25	0.37	0.33	0.46	0.50	0.39	0.36	0.52	0.52	0.63	0.37	0.49	0.40	0.26
38	0.14	0.27	0.36	0.27	0.04	0.03	0.08	0.16	0.10	0.14	0.08	0.24	0.16	0.10	0.08	0.19	0.12	0.11
39	0.03	0.01	0.09	0.02	0.00	0.02	0.06	0.00	0.03	0.00	0.03	0.12	0.01	0.00	0.00	0.02	0.00	0.00
40	0.15	0.06	0.41	0.06	0.02	0.09	0.02	0.05	0.09	0.12	0.02	0.03	0.11	0.04	0.12	0.04	0.07	0.18
41	0.03	0.01	0.02	0.04	0.02	0.03	0.01	0.02	0.02	0.02	0.04	0.04	0.03	0.03	0.02	0.01	0.02	0.02
42	0.02	0.18	0.21	0.03	0.07	0.28	0.05	0.09	0.04	0.03	0.05	0.01	0.15	0.05	0.04	0.04	0.07	0.09
43	0.42	0.13	0.46	0.50	0.34	0.58	0.43	0.26	0.47	0.47	0.70	0.17	0.32	0.04	0.41	0.25	0.07	0.15
44	0.18	0.04	0.06	0.11	0.02	0.09	0.22	0.14	0.16	0.00	0.13	0.14	0.07	0.01	0.08	0.01	0.04	0.03

45	0.04	0.04	0.06	0.02	0.04	0.06	0.03	0.05	0.03	0.09	0.07	0.05	0.10	0.06	0.04	0.03	0.04	0.07
46	0.03	0.09	0.23	0.04	0.11	0.06	0.03	0.08	0.13	0.05	0.06	0.09	0.06	0.09	0.04	0.02	0.11	0.06
47	0.06	0.04	0.21	0.09	0.03	0.06	0.07	0.03	0.16	0.11	0.03	0.19	0.02	0.08	0.01	0.06	0.04	0.05
48	0.08	0.33	0.38	0.16	0.14	0.19	0.16	0.06	0.33	0.70	0.38	0.21	0.26	0.07	0.15	0.28	0.29	0.09
49	0.14	0.33	0.24	0.11	0.11	0.03	0.18	0.03	0.27	0.27	0.11	0.05	0.24	0.14	0.06	0.11	0.13	0.05
50	0.01	0.02	0.02	0.02	0.01	0.02	0.02	0.04	0.05	0.01	0.03	0.03	0.03	0.03	0.03	0.01	0.02	0.02
51	0.02	0.01	0.02	0.10	0.03	0.03	0.01	0.05	0.08	0.03	0.05	0.02	0.09	0.03	0.04	0.02	0.04	0.01
52	0.07	0.04	0.16	0.06	0.03	0.06	0.19	0.06	0.11	0.06	0.10	0.22	0.04	0.15	0.04	0.05	0.04	0.07
53	0.17	0.19	0.05	0.24	0.08	0.39	0.71	0.08	0.15	0.14	0.40	0.12	0.10	0.11	0.09	0.04	0.14	0.25
54	0.05	0.13	0.10	0.27	0.06	0.10	0.15	0.04	0.09	0.23	0.18	0.10	0.09	0.16	0.03	0.10	0.05	0.10
55	0.08	0.01	0.01	0.05	0.01	0.05	0.12	0.03	0.04	0.00	0.09	0.04	0.03	0.01	0.03	0.01	0.02	0.01
56	0.04	0.36	0.78	0.39	0.12	0.16	0.43	0.46	0.52	0.28	0.17	0.20	0.27	0.63	0.12	0.28	0.48	0.34
57	1.00	0.06	1.30	0.48	0.06	0.56	0.84	0.05	1.14	0.42	0.57	1.16	0.09	0.24	0.08	0.39	0.24	0.29
58	0.36	0.11	0.14	0.52	0.15	0.10	0.18	0.08	0.23	0.07	0.42	0.16	0.14	0.10	0.26	0.06	0.03	0.08
59	0.10	0.28	0.07	0.56	0.05	0.03	0.40	0.10	0.39	0.19	0.81	0.27	0.11	0.17	0.05	0.11	0.16	0.12
60	0.03	0.01	0.02	0.04	0.00	0.00	0.05	0.01	0.06	0.01	0.09	0.05	0.01	0.01	0.00	0.02	0.01	0.01
61	0.09	0.15	0.25	0.09	0.22	0.36	0.03	0.08	0.20	0.31	0.26	0.24	0.51	0.45	0.18	0.10	0.21	0.12
62	0.13	0.64	0.32	0.17	0.15	0.09	0.26	0.15	0.28	0.20	0.15	0.16	0.44	0.11	0.05	0.17	0.14	0.18
63	0.19	0.30	0.16	0.86	0.11	0.05	0.13	0.06	0.16	0.22	1.10	0.20	0.21	0.11	0.10	0.38	0.32	0.23
64	0.08	0.12	0.12	0.08	0.07	0.02	0.10	0.06	0.10	0.20	0.18	0.12	0.12	0.18	0.02	0.13	0.08	0.04
65	0.12	0.03	0.02	0.07	0.00	0.02	0.05	0.02	0.06	0.00	0.05	0.03	0.01	0.02	0.00	0.02	0.02	0.03
66	0.01	0.02	0.17	0.08	0.02	0.03	0.04	0.04	0.06	0.04	0.05	0.04	0.04	0.07	0.16	0.04	0.12	0.05
67	0.05	0.02	0.07	0.09	0.04	0.06	0.03	0.04	0.03	0.02	0.09	0.04	0.10	0.08	0.08	0.05	0.04	0.06
68	0.16	0.12	0.34	0.05	0.07	0.12	0.07	0.08	0.18	0.15	0.07	0.10	0.07	0.06	0.04	0.05	0.04	0.06
69	0.06	0.02	0.04	0.07	0.00	0.04	0.06	0.01	0.03	0.00	0.03	0.05	0.00	0.01	0.00	0.04	0.01	0.03
70	0.03	0.01	0.02	0.01	0.01	0.01	0.02	0.03	0.02	0.01	0.02	0.02	0.05	0.02	0.01	0.01	0.01	0.01
71	0.46	0.21	0.58	0.39	0.34	0.60	0.40	0.28	0.35	0.29	0.98	0.39	0.41	0.14	0.82	0.25	0.03	0.26
72	0.03	0.04	0.03	0.07	0.07	0.01	0.23	0.13	0.11	0.00	0.24	0.13	0.06	0.01	0.06	0.02	0.02	0.01
73	0.06	0.07	0.18	0.04	0.06	0.02	0.03	0.03	0.05	0.03	0.03	0.03	0.08	0.04	0.01	0.02	0.05	0.07
74	0.08	0.25	0.14	0.48	0.12	0.18	0.12	0.04	0.10	0.53	0.23	0.32	0.15	0.18	0.09	0.22	0.27	0.15
75	0.18	0.56	0.24	0.09	0.18	0.06	0.28	0.06	0.37	0.45	0.08	0.13	0.26	0.21	0.04	0.12	0.13	0.20
76	0.07	0.07	0.05	0.09	0.04	0.04	0.04	0.08	0.21	0.07	0.04	0.18	0.07	0.12	0.05	0.05	0.09	0.04
77	0.15	0.17	0.10	0.52	0.12	0.37	0.57	0.05	0.12	0.19	0.50	0.19	0.09	0.08	0.09	0.27	0.07	0.21
78	0.21	0.36	0.52	0.40	0.07	0.08	0.43	0.46	0.55	0.24	0.12	0.21	0.29	0.50	0.09	0.29	0.65	0.32
79	0.62	0.20	0.97	0.72	0.05	0.58	0.78	0.09	0.52	0.19	0.83	0.97	0.08	0.36	0.06	0.35	0.16	0.49
80	0.07	0.04	0.02	0.13	0.02	0.02	0.11	0.07	0.20	0.01	0.07	0.02	0.04	0.03	0.03	0.01	0.01	0.03
81	0.11	0.13	0.20	0.08	0.12	0.33	0.04	0.13	0.08	0.34	0.31	0.27	0.70	0.33	0.09	0.17	0.28	0.24
82	0.15	0.49	0.62	0.18	0.08	0.07	0.22	0.15	0.20	0.19	0.15	0.13	0.21	0.11	0.12	0.07	0.12	0.40
83	0.05	0.05	0.07	0.05	0.02	0.04	0.02	0.04	0.13	0.09	0.05	0.05	0.05	0.10	0.01	0.04	0.05	0.03
84	0.02	0.02	0.12	0.04	0.03	0.01	0.05	0.05	0.07	0.04	0.05	0.07	0.11	0.06	0.01	0.02	0.07	0.03
85	0.06	0.05	0.05	0.12	0.04	0.05	0.02	0.02	0.04	0.03	0.12	0.06	0.19	0.07	0.12	0.03	0.06	0.02
86	12	12	11	12	12	12	12	12	12	12	12	9	11	12	12	12	12	12
87	33.4	35.6	35.4	37.1	33.8	39.2	34.0	32.7	40.0	32.2	34.4	40.2	34.3	36.1	39.7	32.2	31.1	37.9
88	0.19	0.21	0.25	0.16	0.16	0.18	0.18	0.22	0.23	0.24	0.14	0.19	0.19	0.16	0.16	0.20	0.23	0.25
89	1.38	1.61	2.11	1.52	0.74	1.36	1.31	1.49	1.22	0.96	1.66	2.82	2.69	2.92	1.17	1.22	1.39	0.77
90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Appendix G: Cross-Correlations of Interest

Variable #	Description	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	Session 1 roll->yaw	1.00																		
2	Session 1 yaw->x	0.18	1.00																	
3	Session 1 pitch->y	0.36	0.65	1.00																
4	Session 1 mean(yaw->x, pitch->y)	0.31	0.88	0.94	1.00															
5	Session 2 roll->yaw	0.87	0.22	0.40	0.36	1.00														
6	Session 2 yaw->x	0.27	0.68	0.90	0.89	0.28	1.00													
7	Session 2 pitch->y	0.46	0.67	0.77	0.80	0.48	0.67	1.00												
8	Session 2 mean(yaw->x, pitch->y)	0.38	0.74	0.92	0.93	0.39	0.94	0.88	1.00											
9	Session 3 roll->yaw	0.60	-0.03	0.04	0.01	0.71	-0.11	0.33	0.08	1.00										
10	Session 3 yaw->x	0.30	0.78	0.74	0.83	0.27	0.85	0.74	0.88	-0.03	1.00									
11	Session 3 pitch->y	0.44	0.62	0.64	0.69	0.32	0.58	0.78	0.73	0.24	0.76	1.00								
12	Session 3 mean(yaw->x, pitch->y)	0.39	0.76	0.74	0.82	0.31	0.78	0.81	0.87	0.09	0.95	0.92	1.00							
13	Session 3 mean(pitch->y, PBCCI same)	0.48	0.64	0.70	0.74	0.35	0.63	0.78	0.76	0.23	0.77	0.98	0.92	1.00						
14	Session 1 TCT Captures Out of 12	-0.59	-0.09	-0.22	-0.18	-0.50	-0.14	-0.34	-0.24	-0.75	-0.09	-0.28	-0.18	-0.35	1.00					
15	Session 1 TCT Avg Time to Capture	0.12	0.12	0.17	0.16	0.04	0.12	0.36	0.24	0.25	0.03	0.06	0.04	0.07	-0.44	1.00				
16	Session 1 TCT Avg Position Error	-0.24	-0.32	-0.08	-0.20	-0.29	-0.19	-0.10	-0.16	0.08	-0.20	-0.15	-0.19	-0.13	-0.12	0.32	1.00			
17	Session 1 TCT Avg Angular Error	0.67	0.11	0.01	0.06	0.59	-0.02	0.14	0.05	0.40	-0.01	-0.06	-0.03	-0.02	-0.38	0.09	0.01	1.00		
18	Session 1 TCT Mean Squared Jerk	-0.09	0.34	0.10	0.22	-0.14	0.11	-0.05	0.04	-0.15	-0.05	-0.05	-0.02	-0.11	-0.11	-0.26	0.14	1.00		
19	Session 2 TCT Captures Out of 12	-0.71	0.09	0.07	0.09	-0.57	0.04	-0.02	0.02	-0.62	0.00	-0.04	-0.02	-0.11	0.78	-0.14	-0.03	-0.64	1.00	
20	Session 2 TCT Avg Time to Capture	0.73	-0.12	-0.08	-0.10	0.56	-0.15	0.11	-0.05	0.49	-0.12	0.01	-0.07	0.07	-0.52	0.33	-0.07	0.61	-0.18	-0.74
21	Session 2 TCT Avg Position Error	0.19	-0.34	-0.11	-0.22	0.02	-0.25	0.05	-0.14	0.37	-0.21	-0.03	-0.14	0.00	-0.47	0.47	0.83	0.35	-0.21	-0.39
22	Session 2 TCT Avg Angular Error	0.78	-0.13	-0.06	-0.10	0.61	-0.14	0.03	-0.07	0.57	-0.13	0.00	-0.08	0.06	-0.66	0.10	0.08	0.85	0.01	-0.87
23	Session 2 TCT Mean Squared Jerk	0.51	-0.14	0.11	0.00	0.27	0.01	0.14	0.07	0.23	-0.12	-0.06	-0.10	-0.03	-0.30	0.50	0.17	0.44	-0.11	-0.40
24	Session 3 TCT Captures Out of 12	-0.63	0.22	0.21	0.24	-0.43	0.15	0.11	0.15	-0.46	0.11	0.07	0.10	0.02	0.61	-0.13	-0.12	-0.67	0.16	0.95
25	Session 3 TCT Avg Time to Capture	0.36	-0.05	0.00	-0.02	0.34	0.01	0.00	0.01	0.30	-0.13	0.06	-0.05	0.08	-0.41	0.35	-0.11	0.21	0.07	-0.34
26	Session 3 TCT Avg Position Error	-0.23	-0.45	-0.18	-0.32	-0.24	-0.32	-0.11	-0.25	0.15	-0.38	-0.35	-0.39	-0.34	-0.14	0.40	0.89	0.12	-0.12	0.00
27	Session 3 TCT Avg Angular Error	0.34	0.19	-0.09	0.03	0.21	0.02	-0.21	-0.08	0.18	0.00	-0.02	-0.01	0.06	-0.38	-0.15	-0.07	0.54	0.35	-0.64
28	Session 3 TCT Mean Squared Jerk	-0.04	-0.15	-0.23	-0.21	-0.09	-0.18	-0.29	-0.25	-0.21	-0.12	-0.17	-0.15	-0.20	0.18	-0.17	-0.44	-0.23	0.08	-0.01
29	Gender	0.00	0.07	0.16	0.13	0.11	0.26	0.23	0.27	0.02	0.29	0.23	0.28	0.15	0.16	-0.09	-0.37	-0.32	-0.15	0.31
30	Age	-0.10	0.18	0.08	0.14	0.06	0.02	0.25	0.13	0.31	0.02	0.21	0.11	0.13	-0.02	0.15	0.04	-0.16	0.03	0.28
31	Height	-0.30	-0.10	-0.13	-0.13	-0.27	-0.02	-0.12	-0.07	-0.11	0.15	0.18	0.17	0.11	0.26	-0.51	-0.09	-0.48	-0.16	0.31
32	Weight	-0.41	-0.26	-0.40	-0.37	-0.32	-0.34	-0.19	-0.30	0.13	-0.16	-0.15	-0.17	-0.22	-0.01	-0.06	0.22	-0.37	-0.27	0.05
33	MIT Student?	0.23	-0.04	0.22	0.12	0.29	0.11	0.08	0.10	0.03	-0.14	-0.15	-0.15	-0.09	0.03	-0.06	-0.05	0.39	0.04	0.02
34	Laterality Quotient	-0.16	0.07	0.08	0.08	-0.08	0.06	0.26	0.15	0.10	0.23	0.29	0.27	0.22	0.09	-0.22	0.20	-0.24	-0.53	0.20
35	Controller Experience?	0.41	0.19	0.23	0.24	0.57	0.13	0.35	0.25	0.59	0.17	0.28	0.23	0.29	-0.52	0.16	-0.36	0.09	0.33	-0.28
36	Instrument Experience?	-0.18	-0.25	-0.05	-0.15	-0.30	0.07	-0.26	-0.07	-0.40	0.00	-0.38	-0.17	-0.33	0.08	0.08	0.45	0.05	-0.34	-0.15
37	PTA Score	-0.29	-0.26	-0.06	-0.16	-0.33	0.02	0.01	0.02	-0.40	0.10	0.11	0.11	0.09	0.32	-0.17	0.09	-0.45	-0.53	0.33
38	MRT Score	-0.44	-0.41	-0.43	-0.47	-0.60	-0.35	-0.39	-0.40	-0.55	-0.36	-0.26	-0.34	-0.30	0.51	-0.12	-0.06	-0.32	0.09	0.48

Variable #	Description	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
20	Session 2 TCT Avg Time to Capture	1.00																		
21	Session 2 TCT Avg Position Error	0.36	1.00																	
22	Session 2 TCT Avg Angular Error	0.75	0.51	1.00																
23	Session 2 TCT Mean Squared Jerk	0.61	0.36	0.49	1.00															
24	Session 3 TCT Captures Out of 12	-0.76	-0.45	-0.84	-0.44	1.00														
25	Session 3 TCT Avg Time to Capture	0.43	0.19	0.34	0.00	-0.34	1.00													
26	Session 3 TCT Avg Position Error	0.02	0.84	0.14	0.24	-0.09	-0.10	1.00												
27	Session 3 TCT Avg Angular Error	0.42	0.05	0.57	0.07	-0.65	0.17	-0.18	1.00											
28	Session 3 TCT Mean Squared Jerk	-0.07	-0.42	-0.17	-0.05	-0.05	0.00	-0.41	-0.15	1.00										
29	Gender	-0.22	-0.28	-0.35	-0.35	0.38	0.27	-0.25	-0.52	0.21	1.00									
30	Age	-0.17	0.01	-0.19	-0.09	0.41	-0.05	0.14	-0.15	-0.49	0.29	1.00								
31	Height	-0.46	-0.17	-0.39	-0.66	0.33	-0.07	-0.11	-0.24	0.15	0.69	0.25	1.00							
32	Weight	-0.30	0.16	-0.26	-0.39	0.08	-0.11	0.28	-0.24	0.24	0.36	0.29	0.64	1.00						
33	MIT Student?	0.31	0.04	0.22	0.19	-0.02	0.27	0.07	0.06	-0.40	0.00	-0.13	-0.28	-0.57	1.00					
34	Laterality Quotient	-0.40	0.03	-0.22	-0.06	0.28	-0.51	0.06	-0.41	-0.23	0.07	0.33	0.24	0.33	-0.29	1.00				
35	Controller Experience?	0.13	-0.13	0.19	-0.18	-0.11	0.31	-0.24	0.02	0.26	0.24	0.00	0.00	-0.02	-0.09	-0.28	1.00			
36	Instrument Experience?	-0.06	0.32	0.07	0.23	-0.27	-0.16	0.35	-0.07	0.13	-0.24	-0.58	-0.17	0.09	-0.09	0.12	-0.50	1.00		
37	PTA Score	-0.31	0.00	-0.35	-0.30	0.25	-0.05	-0.01	-0.57	0.03	0.24	-0.29	0.29	0.10	-0.09	0.39	-0.26	0.36	1.00	
38	MRT Score	-0.25	-0.13	-0.37	-0.16	0.28	0.02	0.00	-0.37	0.25	0.13	-0.38	0.16	-0.10	0.18	-0.17	-0.24	0.08	0.53	1.00

Correlation coefficients highlighted in green (positive) and red (negative) are significant at $p < 0.05$.

Appendix H: Software Code

H.1 Python/Vizard

```
#####  
#  
# Title:          Bimanual Cross-Coupling Test (BCCT)  
# Author:         Victor Wang  
# Description:    The purpose of this program is to quantify the amount of  
#                cross-talk between hands while a user performs bimanual  
#                control in multiple degrees of freedom, using the NASA  
#                RWS dual translational/rotational controller setup.  
#  
# REVISION HISTORY  
#  
# v1.1 (6Jul2011)  
# To improve ability to discern Z movement, made target dropline thicker  
# and transparent.  
# Added scripting of whole BCCT and variable movement restrictions.  
# Disabled pause again because everything is automated.  
# Slightly transparent cursor to ease discerning Z movement.  
# Split into familiarization & stationary version.  
#  
# v1.0 (5Jul2011)  
# Changed name to BCCT from BXT, and skip to v1.0.  
# Removed all real-time FFT code and dependency to NumPy & matplotlib, and  
# therefore can now run in Vizard 3.0.  
# Fullscreen mode.  
# Disabled all keyboard commands except pause/unpause unless displayStatus is True.  
# Default displayStatus = False, so basically this is a debug mode.  
# Added timestamps to main output file and changed freq to 30 Hz.  
#  
# v0.33 (18Apr2011)  
# Updated frequency configs.  
#  
# v0.32 (5Apr2011)  
# Now can press F6 to toggle through DOF configurations.  
# Reorder 1-6 in a DOFList entry to create an arbitrary DOF configuration.  
#  
# v0.31 (29Mar2011)  
# Now using 10x max freq & 30 cycles of min freq for better quality.  
# Arbitrary multi-frequencies in freqList now.  
#  
# v0.30 (24Mar2011)  
# Auto-tracking is quantized to 128 steps now to simulate read data.  
# Test with non-linear deadband in auto-track mode (code commented out  
# in quantize()).  
# Saving data with .dat and FFT data with .fft extensions now to ease  
# file handling with MATLAB.  
#  
# v0.29 (15Mar2011)  
# Fixed FFT figure saving multithreading bug with lock on canvas
```

```

# Fixed typo in rotScale1 calculation.
# Fixed gains calculation.
# Running FFT on first buffer fill (no longer dropping first batch of data)
# since at low freq dropping the first batch could take several minutes.
#
# v0.28 (14Mar2011)
# Saving FFT analysis results in separate FFT data files, with figure .png files
#
# v0.27 (14Mar2011)
# Reset both cursor and target position & target phase on DOF change or freq change.
# Also can trigger the reset using 'r'.
# Changed to quaternion rotation representations to prevent matrix drift.
# Added FFT configuration data to freq configuration list because they are dependent.
# Updated to auto data file naming, using local time & including config settings at
top:
#     freqMode, controlOrder, DOF, automate, showGround, displayStatus
#
# v0.26 (14Mar2011)
# Saving inputs, cursor position & target position now. Filename label 4.
# Changed back to relative target to easily maintain gain ratio. E.g.
# target moves half max speed of cursor. Now simply reset position of
# target to center whenever frequency changes. Slight rotational drift from
# numerical rounding is apparent at high frequencies of target (eg >10 Hz)
# but shouldn't be important, esp. at low frequencies.
# As a result of going back to relative, automate works again (was broken).
# FFT plot hides properly now with 'y' (offset offscreen).
#
# v0.25 (10Mar2011)
# Toggle FFT peak averaging with F12.
# Changed frequencies configuration to a scrollable list. Scroll with 'F5'.
# Changed target motion to absolute to avoid target going everywhere when
# changing frequencies and some apparent drift from numerical rounding.
# Can set frequency and amplitude of target independently.
#
# v0.24 (9Mar2011)
# Added automate cursor feature to test sine wave FFT. Toggle with 't'.
# Note: automate only tracks the target properly in 1 DOF mode.
# Converted background to ground plane and drop line. Toggle with 'g'.
# Increased screen size & adjusted data display. Toggle with 'y'.
#
# v0.23 (7Mar2011)
# Added online FFT plot with matplotlib & vizmatplot.py.
# Note that this requires Python 2.7 installed into the Vizard folder
# (even the Vizard 4.0 Beta version of Python 2.7 is missing some files)
# followed by install of NumPy & matplotlib.
#
# v0.22 (7Mar2011)
# Converted to 3D object.
# Data file format 3.
# Reorganized startup parameters.
# Added background texture checker pattern. Toggle with 'b'.
#
# v0.21 (2Mar2011)
# Display FFT status/time/frame/framerate.

```

```

#
# v0.20 (28Feb2011)
# Prototype of on-line FFT.
# Fixed naming error between yaw/roll and consequent order of inputs.
#
# v0.19 (14Feb2011)
# Changed outfile filename to indicate different format.
# Modified to a strange joystick data read order for RWSS desk setup.
# Fixed joystick button pause/unpause.
# Allowed refresh of experiment parameter display before unpause.
#
# v0.18 (11Feb2011)
# Fixed data file output bug.
# Added ability to select DOF with number keys 1-6.
# Added display of control order, frequencies used, and DOF.
# Toggle display with F9.
# Added ability to toggle pause status with spacebar or joystick trigger
#   or joystick button 2.
# Starts paused.
#
# v0.17 (7Feb2011)
# Combined two targets into a single object with cross and circle.
#
# v0.16 (3Dec2010)
# Set deadband to 0 to increase sensitivity to crosstalk, and also since the
# targets never really stop moving so the user should never want to stay
# still anyways. Disabled mouse control.
#
# v0.15 (2Dec2010)
# Write cursor data instead of inputs so it's easier to compare with targets.
# Enabled selection of target frequency modes using F5-F7 keys.
# Write data file to "data/" folder.
#
# v0.14 (1Dec2010)
# Allowed selection of control order ('0' and '1' keys, default 1st order).
# Stopped asking to use joysticks if they are available, just use them.
# Implemented target & input data logging by frame, automatically saved
# based on date & time upon quitting using Esc key.
#
# v0.13 (30Nov2010)
# Due to difficulty of control, switched to 0th order (position) control.
# It's still pretty difficult.
#
# v0.12 (30Nov2010)
# First usable demo of 6-axis tracking using the two joysticks. This
# header info added, and will be maintained from this version on.
#
# v0.11 (24Nov2010)
# Polishing up interface, minor changes etc..
#
# v0.10 (24Nov2010)
# First interface prototype with two targets and a keyboard-controlled
# horizontal cursor.
#

```

```

#####
from __future__ import division
from time import localtime, strftime
from collections import deque
import time
import viz
import math
import vizact
import vizinfo
import vizjoy

#####
#
# Setup variables
#
#####

# timing parameters
loopFreq = 30
loopRate = 1.0/loopFreq          # time between main loop executions
frUpdate = loopFreq              # update frameRate every this many frames

# 39 mins 30 secs total (6 axis-pairs)
practiceTime = 35                # seconds per practice run (33s for .03 Hz cycle)
trialTime = 5*60+40              # mins & secs per trial (5mins 33s for 10 cycles)
breakTime = 20                   # secs of break before each trial

# aesthetics
bgColor = [1,1,1]
groundTex = 'checker_bw.gif'
groundDepth = 3
groundAlpha = 0.5

# cursor/target parameters

cursorFile = 'horse.wrl'
targetFile = 'horse.wrl'

cursorScale = 1.2
targetScale = 1.2

cursorCenter = [0,1.15,0]
targetCenter = [0,1.15,0]

cursorInitialRot = 180
targetInitialRot = 180

cursorColor = [0,0,1]
targetColor = [1,0,0]

cursorAlpha = 0.7
targetAlpha = 0.3                # transparency to allow overlapping
cursor/target

```

```

# how far away to position the cursor/target
# combined with cursor/target scaling, this affects the strength of perspective
viewDepth = 10

# control parameters
deadband = 0.0                                # joystick deadband

# target movement frequency configurations:
# [[leftFreqs], [rightFreqs]]
freqList = [ [.07,.17],[.03,.19]], [.03,.19],[.07,.17]] ]

# DOF configurations:
# [1,2,3,4,5,6] corresponding to [x,y,z,roll,pitch,yaw], change to change order
# enabled by toggling DOF #
DOFList = [ [1,5,2,4,3,6], [2,4,3,6,1,5], [3,6,2,5,1,4], [2,5,1,6,3,4], [1,6,3,4,2,5],
[3,4,1,5,2,6]]

# INITIALIZE VARIABLES

paused = False                                # start out paused
automate = False                              # default to manual control

controlOrder = 1                             # default to 1st order control
DOF = 2                                       # default to 2 DOF mode

ground = viz.addTexQuad()
showGround = True                            # default display of ground & droplines

displayStatus = False                        # default display of debug data & status

freqMode = 0                                 # actual frequency configuration to use right now
DOFMode = 0                                  # actual DOF configuration to use right now

dataLog = []                                 # data log to be recorded to file at end of script
frame = 0                                    # set up frame counter
frameRate = 0.0                              # initialize frame rate

trialMode = 0                                # 0 means practice, 1 means test

#####
#
# Initialize the frequency related parameters
#
#####

def setFreq(mode):
    global leftFreqs, rightFreqs

    [leftFreqs, rightFreqs] = freqList[mode]

#####
#
# Initialize cursor and target gains. Dependent on frequency.

```

```

#
#####

def setGains():
    global transScale0, transScale1, rotScale0, rotScale1, targetTransScale,
    targetRotScale

    # input scaling factors
    transScale0 = 2.0 # translation, 0th order
    transScale1 = 10.0/loopFreq*sum(leftFreqs)/len(leftFreqs) # translation,
1st order
    rotScale0 = 90.0 # rotation, 0th order
    rotScale1 = 2*math.pi/loopFreq*sum(rightFreqs)/len(rightFreqs) #
rotation, 1st order

    # target movement amplitudes, absolute
    targetTransScale = 0.5*transScale1
    targetRotScale = 0.5*rotScale1

#####
#
# Reset
# Reset for a new trial.
#
#####

def reset():
    global frameTime, startTime, dataLog, frame, frameRate

    cursor.setPosition(0,-cursorCenter[1],viewDepth)
    cursor.setEuler(cursorInitialRot, 0, 0)

    target.setPosition(0,-targetCenter[1],viewDepth)
    target.setEuler(targetInitialRot, 0 ,0)

    dataLog = [] # data log to be recorded to file at end of script
    frame = 0 # set up frame counter
    frameRate = 0.0 # initialize frame rate

    setFreq(freqMode)
    startTime = time.clock()

    updateDroplines()

    frameTime = time.clock()

#####
#
# Update Droplines
#
#####

```

```

def updateDropLines():
    [cursorX, cursorY, cursorZ] = cursor.getPosition()
    dropLineC.setVertex(0, [cursorX, cursorY+cursorCenter[1], cursorZ])
    dropLineC.setVertex(1, [cursorX, -groundDepth, cursorZ])

    [targetX, targetY, targetZ] = target.getPosition()
    dropLineT.setVertex(0, [targetX, targetY+cursorCenter[1], targetZ])
    dropLineT.setVertex(1, [targetX, -groundDepth, targetZ])

#####
#
# Check joystick inputs
#
#####
# Collect the data from the two hand controllers and return the delta pitch, yaw,
roll, x,y,z values
# The output of this method should be similar to the output from using the keyboard
# NB: The THC data values are developed using the Logitech Extreme 3D joystick. Other
joysticks may require a sign change for various axes.

def getJoystick():
    global thc, rhc

    roll = yaw = pitch = x = y = z = 0
    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
            else:
                if data[x][y] < 0.0:
                    data[x][y] += deadband
                else:
                    data[x][y] -= deadband

    # Joystick output is +-1.0
    # RHC data is roll (L<->R), pitch (F<->B), yaw (twist)
    rhc = [-data[0][0], -data[0][1], data[0][3]]

```

```

    thc = [data[1][0], -data[1][1], data[1][2]]

    return thc[0], thc[1], thc[2], rhc[0], rhc[1], rhc[2]

#####
#
# Check keyboard inputs. More than one key can be pressed simultaneously.
#
#####

def getKeyboard():
    pitch = yaw = roll = x = y = z = 0

    # Check the right hand keys
    if viz.key.isDown('i'):
        pitch += 1.0
    if viz.key.isDown('k'):
        pitch += -1.0

    if viz.key.isDown('j'):
        roll += 1.0
    if viz.key.isDown('l'):
        roll += -1.0

    if viz.key.isDown('u'):
        yaw += -1.0
    if viz.key.isDown('o'):
        yaw += 1.0

    # Check the left hand keys

    if viz.key.isDown('w'):
        y += 1.0
    if viz.key.isDown('s'):
        y += -1.0

    if viz.key.isDown('d'):
        x += 1.0
    if viz.key.isDown('a'):
        x += -1.0

    if viz.key.isDown('q'):
        z += -1.0
    if viz.key.isDown('e'):
        z += 1.0

    return x, y, z, roll, pitch, yaw

#####
#
# Callback routine for handling instantaneous keyboard inputs
#

```

```
#####
```

```
def keyPressed(key):

    global controlOrder, freqMode, DOF, DOFMode, paused
    global showGround, automate, displayStatus, averagePeak

    if key == viz.KEY_ESCAPE: #end program
        saveFile()
        viz.quit()

    if displayStatus:
        if key == '0': # switch between 0th and 1st order control
            controlOrder = not controlOrder
            reset()
        elif key == '1': # switch to 1 DOF mode
            DOF = 1
            reset()
        elif key == '2': # switch to 2 DOF mode
            DOF = 2
            reset()
        elif key == '3': # switch to 3 DOF mode
            DOF = 3
            reset()
        elif key == '4': # switch to 4 DOF mode
            DOF = 4
            reset()
        elif key == '5': # switch to 5 DOF mode
            DOF = 5
            reset()
        elif key == '6': # switch to 6 DOF mode
            DOF = 6
            reset()
        elif key == viz.KEY_F5: # next target frequency configuration
            freqMode += 1
            freqMode %= len(freqList)
            setFreq(freqMode)
            setGains()
            reset()
        elif key == viz.KEY_F6: # next DOF configuration
            DOFMode += 1
            DOFMode %= len(DOFList)
            setFreq(freqMode)
            reset()
        elif key == 'g': # toggle use of background texture
            showGround = not showGround
            ground.visible(showGround)
            dropLineT.visible(showGround)
            dropLineC.visible(showGround)
        elif key == 't': # toggle use of automation (target tracking)
            automate = not automate
            autoText.visible(automate)
        elif key == 'r': # reset
            reset()
```

```

        elif key == ' ':          # toggle paused status
            paused = not paused
            pausedText.visible(paused)
    if key == 'y':              # toggle status display
        displayStatus = not displayStatus
        frequenciesText.visible(displayStatus)
        controlOrderText.visible(displayStatus)
        DOFText.visible(displayStatus)
        frameText.visible(displayStatus)
        timeText.visible(displayStatus)
        frameRateText.visible(displayStatus)

#####
#
# Callback routine for handling instantaneous joystick button inputs
#
#####

def joystickButtonDown(b):
    global paused
    #     if b.button == 1 or b.button == 2:
    #         paused = not paused
    #         pausedText.visible(paused)

#####
#
# Save all the logged frame-by-frame data to a file on the hard drive
# Includes trial length and config settings at the top of the file.
# Note that configuration settings are only logged at end of program, even though
# settings can change in real time for testing purposes. If recording actual data,
# make sure that config settings have not changed during the trial.
#
#####

def saveFile():
    if trialMode == 0:
        trialString = '-Practice' + str(DOFMode+1) + '-'
    else:
        trialString = '-Test' + str(DOFMode+1) + '-'
    timeString = strftime('%d%b%Y-%Hhr%Mmin%Ss',localtime())
    filename = 'data/' + __file__[__file__.find('.py')-8:__file__.find('.py')] +
trialString + timeString + '.dat'
    print 'Saving data in file: ', filename
    file = open(filename,'w')

    file.write('Duration: ' + currentTime() + '\n')
    file.write('freqMode: %d\t\tcontrolOrder: %d\t\tDOF: %d\t\tautomate: %d\t\tsho
wGround: %d\t\tdisplayStatus: %d\t\t\n' % (freqMode, controlOrder, DOF, automate,
showGround, displayStatus))
    file.write('leftFreq: %s\t\ttrightFreq: %s\t\ttransScale1: %f\t\trotScale1: %f\
n' % (leftFreqs, rightFreqs, transScale1, rotScale1))

```



```

#####

def quantize(x):
    if abs(x) < 0.3: # deadband to test non-linearity
        x = 0;
    return int(x*63)/63.0

#####
#
# Return the next power of 2 greater than or equal to n
#
#####

def nextPow2(n):
    return int(2 ** math.ceil(math.log(n, 2)))

#####
#
# Return the current time in a string
#
#####

def currentTime():
    elapsedMins = (time.clock()-startTime)/60
    elapsedSecs = (time.clock()-startTime)%60
    return '%dmin %ds' % (elapsedMins, elapsedSecs)

#####
#
# Return sinusoidal motion paths
#
#####

def generateMotion():
    motions = [0,0,0,0,0,0]

    for i in range(0,len(leftFreqs)):
        motions[(DOFList[DOFMode])[0]-1] +=
math.cos(leftFreqs[i]*2.0*math.pi/loopFreq*(frame))
        if DOF > 2:
            motions[(DOFList[DOFMode])[2]-1] +=
math.cos(leftFreqs[i]*2.0*math.pi/loopFreq*(frame))
        if DOF > 4:
            motions[(DOFList[DOFMode])[4]-1] +=
math.cos(leftFreqs[i]*2.0*math.pi/loopFreq*(frame))

    for i in range(0,len(rightFreqs)):
        if DOF > 1:
            motions[(DOFList[DOFMode])[1]-1] +=
math.cos(rightFreqs[i]*2.0*math.pi/loopFreq*(frame))
        if DOF > 3:
            motions[(DOFList[DOFMode])[3]-1] +=
math.cos(rightFreqs[i]*2.0*math.pi/loopFreq*(frame))

```

```

        if DOF > 5:
            motions[(DOFList[DOFMode])[5]-1] +=
math.cos(rightFreqs[i]*2.0*math.pi/loopFreq*(frame))

    [x, y, z, roll, pitch, yaw] = motions

    x /= len(leftFreqs)
    y /= len(leftFreqs)
    z /= len(leftFreqs)
    roll /= len(leftFreqs)
    pitch /= len(leftFreqs)
    yaw /= len(leftFreqs)

    return [x, y, z, roll, pitch, yaw]

#####
#
# Main loop, running on a timer cycle
#
#####

def mainLoop():

    global frame # frame counter
    global frameRate, frameTime # to calculate framerate
    global dataLog # data log
    global trialMode, DOFMode, paused

    # experiment scripting
    if trialMode == 0: # practice
        if time.clock()-startTime > practiceTime: # next practice trial
            saveFile()
            if DOFMode == len(DOFList)-1: # on to test
                DOFMode = 0
                trialMode = 2
            else: # not done
                DOFMode += 1
            reset()
    elif trialMode == 1 and time.clock()-startTime > trialTime: # next trial
        saveFile()
        if DOFMode == len(DOFList)-1: # done
            viz.quit()
        else: # not done
            DOFMode += 1
            trialMode = 2
            reset()
    if trialMode == 2: # break
        paused = 1
        breakText.visible(1)
        breakText.message('Break: ' + str(int(breakTime - (time.clock()-
startTime))) + ' s')
        if time.clock()-startTime > breakTime: # time to start test
            trialMode = 1

```

```

        reset()
        paused = 0
        breakText.visible(0)

    # Update status displays
    controlOrderText.message('Control Order: ' + str(int(controlOrder)))
    frequenciesText.message('Freqs: [%.4f, %.4f] [%.4f, %.4f]' % (leftFreqs[0],
leftFreqs[1], rightFreqs[0], rightFreqs[1]))
    DOFText.message('DOF: ' + str(DOF) + ' /w mode ' + str(DOFMode+1))
    DOFText.message('DOF: ' + str(DOF) + ' /w mode ' + str(DOFMode+1))
    timeText.message('Time: ' + currentTime())
    frameText.message('Frame: ' + str(int(frame)))
    frameRateText.message('Frame Rate: %.2f' % frameRate)
    if not paused:
        if frame%frUpdate == 0:
            frameRate = frUpdate/(time.clock()-frameTime)
            frameTime = time.clock()

    logEntry = []# temporary storage of this frame's data log entry
    frame += 1          # increment frame counter
    logEntry.append((time.clock()-startTime, frame))          # log the
frame number

    # get inputs
    if automate:
        [x, y, z, roll, pitch, yaw] = generateMotion()
        x = quantize(x)
        y = quantize(y)
        z = quantize(z)
        roll = quantize(roll)
        pitch = quantize(pitch)
        yaw = quantize(yaw)
    elif useJoysticks:
        [x, y, z, roll, pitch, yaw] = getJoystick()
    else:
        [x, y, z, roll, pitch, yaw] = getKeyboard()

    # log the control inputs
    logEntry.append([x, y, z])
    logEntry.append([roll, pitch, yaw])

    if controlOrder == 0:
        # scale inputs
        x *= transScale0
        y *= transScale0
        z *= transScale0
        roll *= rotScale0
        pitch *= rotScale0
        yaw *= rotScale0

        # translate cursor
        cursor.setPosition([x, y-cursorCenter[1], z+viewDepth],
viz.ABS_PARENT)

```

```

        # rotate cursor
        cursor.setEuler([yaw-cursorInitialRot, -pitch, -roll],
viz.ABS_PARENT)

    else: # control order == 1
        # scale inputs
        if automate:
            pitch *= targetRotScale
            roll *= targetRotScale
            yaw *= targetRotScale
            x *= targetTransScale
            y *= targetTransScale
            z *= targetTransScale
        else:
            pitch *= rotScale1
            roll *= rotScale1
            yaw *= rotScale1
            x *= transScale1
            y *= transScale1
            z *= transScale1

        # disable axes as appropriate
        motions = [x, y, z, roll, pitch, yaw]
        if DOF < 6:
            motions[(DOFList[DOFMode])[5]-1] = 0
        if DOF < 5:
            motions[(DOFList[DOFMode])[4]-1] = 0
        if DOF < 4:
            motions[(DOFList[DOFMode])[3]-1] = 0
        if DOF < 3:
            motions[(DOFList[DOFMode])[2]-1] = 0
        if DOF < 2:
            motions[(DOFList[DOFMode])[1]-1] = 0
        [x, y, z, roll, pitch, yaw] = motions

        # move cursor
        cursor.setPosition([x,y,z], viz.REL_PARENT)
        cursor.setQuat([pitch, yaw, roll, 1], viz.REL_PARENT)

    # log the cursor position
    [cursorX, cursorY, cursorZ] = cursor.getPosition()
    logEntry.append([cursorX, cursorY, cursorZ])

    # log the cursor rotation
    [cursorYaw, cursorPitch, cursorRoll] = cursor.getEuler()
    logEntry.append([cursorRoll, cursorPitch, cursorYaw])

    # target translation/rotation
    [x,y,z,roll,pitch,yaw] = generateMotion()
    x *= targetTransScale
    y *= targetTransScale
    z *= targetTransScale
    roll *= targetRotScale

```

```

    pitch *= targetRotScale
    yaw *= targetRotScale
    target.setPosition([x,y,z], viz.REL_PARENT)
    target.setQuat(pitch, yaw, roll, 1, viz.REL_PARENT)

    # log the target position
    [targetX, targetY, targetZ] = target.getPosition()
    logEntry.append([targetX, targetY, targetZ])

    if showGround:
        updateDropLines()

    # log the target rotation
    [targetYaw, targetPitch, targetRoll] = target.getEuler()
    logEntry.append([targetRoll, targetPitch, targetYaw])

    # write the log entry to the data log
    dataLog.append(logEntry)

#####
#
# Initialization
#
#####

#####
# Set up inputs
#####

rhc = [0.0,0.0,0.0]
thc = [0.0,0.0,0.0]

useJoysticks = 0; # default to keyboard input, changed by dialog if available

# Attempt to install two joysticks for the simulation
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()

#set booleans for condition of joystick buttons
trigger_down = False
button3_down = False

# Check that both joysticks added are valid.
# If so, use them.
if sid.valid() and sid2.valid():
    useJoysticks = 1; # just use the joysticks if they're available

viz.callback(viz.KEYBOARD_EVENT, keyPressed)
viz.callback(vizjoy.BUTTONDOWN_EVENT, joystickButtonDown)

#####

```

```

# Set up graphics
#####

#viz.setOption('viz.window.width','1024')
#viz.setOption('viz.window.height','768')
viz.setOption('viz.fullscreen.monitor','2')
viz.setOption('viz.fullscreen','1')

viz.clearcolor(bgColor)
viz.mouse(viz.OFF)           # disable mouse navigation
viz.eyehight(0)

# create cursor
cursor = viz.add(cursorFile)
cursor.alpha(cursorAlpha)
cursor.color(cursorColor)
cursor.center(cursorCenter)
cursor.setScale(cursorScale,cursorScale,cursorScale)

# create target
target = viz.add(targetFile)
target.alpha(targetAlpha)
target.color(targetColor)
target.center(targetCenter)
target.setScale(targetScale,targetScale,targetScale)

# create ground
matrix = vizmat.Transform()
matrix.setScale(2,2,1)
ground.texmat(matrix)
tex = viz.addTexture(groundTex)
tex.wrap(viz.WRAP_S, viz.REPEAT)
tex.wrap(viz.WRAP_T, viz.REPEAT)
ground.texture(tex)
ground.alpha(groundAlpha)
ground.setScale(10,10,0)
ground.setPosition([0, -groundDepth, viewDepth]) #put quad in view
ground.setEuler(0,90,0)
ground.visible(showGround)

# create droplines
viz.startLayer(viz.LINES)
[x, y, z] = cursor.getPosition()
viz.vertex(x, y+cursorCenter[1], z)
viz.vertex(x, -groundDepth, z)
viz.lineWidth(3)
viz.pointSize(10)
dropLineC = viz.endLayer()
dropLineC.color(cursorColor)
dropLineC.visible(showGround)
dropLineC.dynamic()
dropLineT = dropLineC.copy()
dropLineT.color(targetColor)
dropLineT.visible(showGround)

```

```

dropLineT.dynamic()
dropLineT.alpha(targetAlpha)
dropLineT.lineWidth(10)

# create status displays

breakText = vizinfo.add('Break: 0 s')
breakText.translate(.3,.91)
breakText.visible(0)

frequenciesText = vizinfo.add('Freqs: [0.000] [0.000]')
frequenciesText.translate(0.98,0.98)
frequenciesText.visible(displayStatus)

controlOrderText = vizinfo.add('Control Order: 0')
controlOrderText.translate(0.98,0.91)
controlOrderText.visible(displayStatus)

DOFText = vizinfo.add('DOF: 0')
DOFText.translate(0.98,0.84)
DOFText.visible(displayStatus)

frameText = vizinfo.add('Frame: 0000')
frameText.translate(0.98,0.46)
frameText.visible(displayStatus)

timeText = vizinfo.add('Time: 0mon 00s')
timeText.translate(0.98,0.39)
timeText.visible(displayStatus)

frameRateText = vizinfo.add('Frame Rate: 00.00')
frameRateText.translate(0.98,0.32)
frameRateText.visible(displayStatus)

autoText = vizinfo.add('AUTO-TRACKING')
autoText.translate(0.98,0.60)
autoText.visible(automate)

pausedText = vizinfo.add('PAUSED')
pausedText.translate(0.575,0.54)
pausedText.visible(paused)

setFreq(freqMode)

#####
# Go!
#####

setGains()
reset()

# Start the main simulation loop to run
loopTimer = vizact.ontimer(loopRate,mainLoop)
viz.go()

```

H.2 MATLAB

```
%% compileAll.m
% Compiles all Bimanual Cross-Coupling in Space Telerobotics study data
% into a SYSTAT-compatible, tab-delimited data file. Put this with the
% master subject data folder.
% Requires compileSubject.m, compileBCCT.m, and processBCCT.m

%% clean up workspace
close all
clear all
clc

%% set execution parameters
dataFolder = 'SubjectData'; % name of the data folder

%% set up data storage structs
BCCSTdata = struct('name', '', 'PBCCT', [76], 'SBCCT', {}, ...
    'subjectData', [], 'session', []);
dirs = dir(dataFolder); % get directory structure, including files/folders
isDirs = [dirs.isdir]; % get vector of folder indexes

%% process each subject and add data struct to BCCSTdata
if find(isDirs(3:length(dirs))) % there are subfolders (for subject data)
    for i = 3:length(dirs) % skip '.' and '..' which are 1 and 2 index
        if isDirs(i) == 1 % it's a folder
            % call compileSubject on it
            subjectRecord = compileSubject(...
                strcat(dataFolder, '/', dirs(i).name))
            % add the result
            l = length(BCCSTdata)+1;
            BCCSTdata(l:l+length(subjectRecord)-1) = subjectRecord;
        else
            % it's a file, do nothing
        end
    end
else
    error = 'No subject data found!'
end
save('BCCSTdata.mat', 'BCCSTdata');

%% convert to SYSTAT format
% left to right:
% subject# [subjectData] [PBCCTdata] [SBCCTdata] ...
% session# [sessionData] index value

% generate matrix in file compatible form
systat = [];
% for each subject
for i = 1:length(BCCSTdata)
    % setup subject data common to all sessions
    PBCCTdata = compileBCCT(BCCSTdata(i).PBCCT.CCmatrix, [], ...
```

```

        BCCSTdata(i).PBCCT.errors, 0);
SBCCTdata = compileBCCT(BCCSTdata(i).SBCCT.CCmatrix,[],...
        BCCSTdata(i).SBCCT.errors, 1); % 1 to specify stationary
subjectRow = [i BCCSTdata(i).subjectData' PBCCTdata SBCCTdata];

% for each session
for j = 1:3
    % setup session data common to all indices
    sessionRow = [BCCSTdata(i).session(j).data']; % session # first
    % setup values array
    BCCTline = compileBCCT(BCCSTdata(i).session(j).CCmatrix,...
        BCCSTdata(i).session(j).CCmatrix2,...
        BCCSTdata(i).session(j).errors, 0);
    TCTline = [BCCSTdata(i).session(j).TCT.captures...
        BCCSTdata(i).session(j).TCT.avgTime...
        BCCSTdata(i).session(j).TCT.avgPerror...
        BCCSTdata(i).session(j).TCT.avgAerror...
        BCCSTdata(i).session(j).TCT.totalCollisions...
        BCCSTdata(i).session(j).TCT.totalFailedGrapples];
    values = [BCCTline TCTline];

    % for each index/value pair
    for k = 1:length(values)
        % get the index value
        newRow = [subjectRow sessionRow k values(k)];
        systat = [systat; newRow];
    end
end
end

% write matrix to file
header = ['Subject\tGender (Female=0,Male=1)\tAge (Years)\tHeight (m)\t'...
'Weight (kg)\tMIT Student (No=0,Yes=1)\tMIT Course Number (0=N/A)\t'...
'Laterality Quotient, Edinburgh (-100=Left,100=Right)\t'...
'Visual Issues (No=0,Yes=1)\tPrescribed Lenses (No=0,Yes=1)\t'...
'Snellen Acuity Left (20/xx)\tSnellen Acuity Right (20/xx)\t'...
'Neuromotor Disease (No=0,Yes=1)\tCaffeine Consumer (No=0,Yes=1)\t'...
'Drug User (No=0,Yes=1)\tMost Alert 0 h - 3 h (No=0,Yes=1)\t'...
'Most Alert 3 h - 6 h (No=0,Yes=1)\t'...
'Most Alert 6 h - 9 h (No=0,Yes=1)\t'...
'Most Alert 9 h - 12 h (No=0,Yes=1)\t'...
'Most Alert 12 h - 15 h (No=0,Yes=1)\t'...
'Most Alert 15 h - 18 h (No=0,Yes=1)\t'...
'Most Alert 18 h - 21 h (No=0,Yes=1)\t'...
'Most Alert 21 h - 24 h (No=0,Yes=1)\t'...
'Most Sleepy 0 h - 3 h (No=0,Yes=1)\t'...
'Most Sleepy 3 h - 6 h (No=0,Yes=1)\t'...
'Most Sleepy 6 h - 9 h (No=0,Yes=1)\t'...
'Most Sleepy 9 h - 12 h (No=0,Yes=1)\t'...
'Most Sleepy 12 h - 15 h (No=0,Yes=1)\t'...
'Most Sleepy 15 h - 18 h (No=0,Yes=1)\t'...
'Most Sleepy 18 h - 21 h (No=0,Yes=1)\t'...
'Most Sleepy 21 h - 24 h (No=0,Yes=1)\t'...
'Normal Sleep (hrs)\tVirtual Environments (No=0,Yes=1)\t'...

```

'Joysticks/Controller (No=0,Yes=1)\t'...

'Bimanual Instruments (No=0,Yes=1)\t'...

'Perspective Taking Ability Score\tMental Rotations Test Score\t'...

'Gap Change BCCT tracking (1=Worse,5=Better)\t'...

'Gap Change BCCT predicting (1=Worse,5=Better)\t'...

'Gap Change BCCT bimanual indep. (1=Worse,5=Better)\t'...

'Gap Change BCCT intrajoystick indep. (1=Worse,5=Better)\t'...

'Gap Change TCT tracking (1=Worse,5=Better)\t'...

'Gap Change TCT predicting (1=Worse,5=Better)\t'...

'Gap Change TCT bimanual independ. (1=Worse,5=Better)\t'...

'Gap Change TCT intrajoystick indep. (1=Worse,5=Better)\t'...

'Gap Length (Days, [Session3Date-Session2Date])\t'...

'PBCCT Average X Error (1=Doing Nothing)\t'...

'PBCCT Average Y Error (1=Doing Nothing)\t'...

'PBCCT Average Z Error (1=Doing Nothing)\t'...

'PBCCT Average Roll Error (1=Doing Nothing)\t'...

'PBCCT Average Pitch Error (1=Doing Nothing)\t'...

'PBCCT Average Yaw Error (1=Doing Nothing)\t'...

'PBCCT X->Y\tPBCCT X->Z\tPBCCT X->Roll\tPBCCT X->Yaw\t'...

'PBCCT Y->X\tPBCCT Y->Z\tPBCCT Y->Pitch\tPBCCT Y->Yaw\t'...

'PBCCT Roll->X\tPBCCT Roll->Z\tPBCCT Roll->Pitch\tPBCCT Roll->Yaw\t'...

'PBCCT Pitch->Y\tPBCCT Pitch->Z\tPBCCT Pitch->Roll\t'...

'PBCCT Pitch->Yaw\t'...

'SBCCT Average X Error (1=Doing Nothing)\t'...

'SBCCT Average Y Error (1=Doing Nothing)\t'...

'SBCCT Average Z Error (1=Doing Nothing)\t'...

'SBCCT Average Roll Error (1=Doing Nothing)\t'...

'SBCCT Average Pitch Error (1=Doing Nothing)\t'...

'SBCCT Average Yaw Error (1=Doing Nothing)\t'...

'SBCCT X->Y\tSBCCT X->Z\tSBCCT X->Roll\tSBCCT X->Pitch\t'...

'SBCCT X->Yaw\tSBCCT Pitch->X\tSBCCT Pitch->Y\tSBCCT Pitch->Z\t'...

'SBCCT Pitch->Roll\tSBCCT Pitch->Yaw\t'...

'Session Number\tTime (start + 1 hour, rounded to closest hour)\t'...

'Wearing Lenses (0=No,1=Yes)\tCaffeine Influence (0=No,1=Yes)\t'...

'Drug Influence (0=No,1=Yes)\tLast Sleep (hours)\t'...

'Karolinska Sleepiness Scale (1=Alert,9=Sleepy)\t'...

'Other Factors (0=No,1=Yes)\tDiscomfort: Nausea (1=None,5=Severe)\t'...

'Discomfort: Dizziness (1=None,5=Severe)\t'...

'Discomfort: Disorientation (1=None,5=Severe)\t'...

'Discomfort: Eyestrain (1=None,5=Severe)\t'...

'Discomfort: Blurred Vision (1=None,5=Severe)\t'...

'Discomfort: Sweating (1=None,5=Severe)\t'...

'Discomfort: Headache (1=None,5=Severe)\t'...

'Discomfort: General (1=None,5=Severe)\t'...

'Discomfort: Mental Fatigue (1=None,5=Severe)\t'...

'Discomfort: Hand Fatigue (1=None,5=Severe)\t'...

'Discomfort: Hand Pain (1=None,5=Severe)\t'...

'Discomfort: Other (1=None,5=Severe)\t'...

'Enjoyability of BCCT (1=Boring,5=Captivating)\t'...

'Enjoyability of TCT (1=Boring,5=Captivating)\t'...

'Ability in BCCT tracking (1=Unable,5=Expert)\t'...

'Ability in BCCT predicting (1=Unable,5=Expert)\t'...

'Ability in BCCT bimanual indep. (1=Unable,5=Expert)\t'...

'Ability in BCCT intrajoystick indep. (1=Unable,5=Expert)\t'...

```

'Ability in TCT tracking (1=Unable,5=Expert)\t'...
'Ability in TCT predicting (1=Unable,5=Expert)\t'...
'Ability in TCT bimanual independ. (1=Unable,5=Expert)\t'...
'Ability in TCT intrajoystick indep. (1=Unable,5=Expert)\t'...
'Difficulty of BCCT (1=Easy,5=Difficult)\t'...
'Difficulty of TCT (1=Easy,5=Difficult)\t'...
'Index\tValue\n'];
file = fopen('BCCST-SYSTAT.txt','w');
fprintf(file,header);
fclose(file);
dlmwrite('BCCST-SYSTAT.txt',systat,'-append','delimiter','\t');

```

```
% Reference for index meanings:
```

- % 1 BCCT Average Tracking Error, X
- % 2 BCCT Average Tracking Error, Y
- % 3 BCCT Average Tracking Error, Z
- % 4 BCCT Average Tracking Error, Roll
- % 5 BCCT Average Tracking Error, Pitch
- % 6 BCCT Average Tracking Error, Yaw
- % 7 BCCT Coupling X->Y (Drivers: X & Pitch)
- % 8 BCCT Coupling X->Z (Drivers: X & Pitch)
- % 9 BCCT Coupling X->Roll (Drivers: X & Pitch)
- % 10 BCCT Coupling X->Pitch (Drivers: X & Yaw) **** *
- % 11 BCCT Coupling X->Yaw (Drivers: X & Pitch)
- % 12 BCCT Coupling Y->X (Drivers: Y & Roll)
- % 13 BCCT Coupling Y->Z (Drivers: Y & Roll)
- % 14 BCCT Coupling Y->Roll (Drivers: Y & Pitch) **** *
- % 15 BCCT Coupling Y->Pitch (Drivers: Y & Roll)
- % 16 BCCT Coupling Y->Yaw (Drivers: Y & Roll)
- % 17 BCCT Coupling Z->X (Drivers: Z & Yaw)
- % 18 BCCT Coupling Z->Y (Drivers: Z & Yaw)
- % 19 BCCT Coupling Z->Roll (Drivers: Z & Yaw)
- % 20 BCCT Coupling Z->Pitch (Drivers: Z & Yaw)
- % 21 BCCT Coupling Z->Yaw (Drivers: Z & Roll)
- % 22 BCCT Coupling Roll->X (Drivers: Y & Roll)
- % 23 BCCT Coupling Roll->Y (Drivers: Z & Roll)
- % 24 BCCT Coupling Roll->Z (Drivers: Y & Roll)
- % 25 BCCT Coupling Roll->Pitch (Drivers: Y & Roll)
- % 26 BCCT Coupling Roll->Yaw (Drivers: Y & Roll)
- % 27 BCCT Coupling Pitch->X (Drivers: Y & Pitch) **** *
- % 28 BCCT Coupling Pitch->Y (Drivers: X & Pitch)
- % 29 BCCT Coupling Pitch->Z (Drivers: X & Pitch)
- % 30 BCCT Coupling Pitch->Roll (Drivers: X & Pitch)
- % 31 BCCT Coupling Pitch->Yaw (Drivers: X & Pitch)
- % 32 BCCT Coupling Yaw->X (Drivers: Z & Yaw)
- % 33 BCCT Coupling Yaw->Y (Drivers: Z & Yaw)
- % 34 BCCT Coupling Yaw->Z (Drivers: X & Yaw) **** *
- % 35 BCCT Coupling Yaw->Roll (Drivers: Z & Yaw)
- % 36 BCCT Coupling Yaw->Pitch (Drivers: Z & Yaw)
- % 37 BCCT Coupling Redundant X->Y (Drivers: X & Yaw)
- % 38 BCCT Coupling Redundant X->Z (Drivers: X & Yaw)
- % 39 BCCT Coupling Redundant X->Roll (Drivers: X & Yaw)
- % 40 BCCT Coupling Redundant Y->X (Drivers: Y & Pitch)
- % 41 BCCT Coupling Redundant Y->Z (Drivers: Y & Pitch)

```

% 42 BCCT Coupling Redundant Y->Yaw (Drivers: Y & Pitch)
% 43 BCCT Coupling Redundant Z->X (Drivers: Z & Roll)
% 44 BCCT Coupling Redundant Z->Y (Drivers: Z & Roll)
% 45 BCCT Coupling Redundant Z->Pitch (Drivers: Z & Roll)
% 46 BCCT Coupling Redundant Roll->X (Drivers: Z & Roll)
% 47 BCCT Coupling Redundant Roll->Pitch (Drivers: Z & Roll)
% 48 BCCT Coupling Redundant Roll->Yaw (Drivers: Z & Roll)
% 49 BCCT Coupling Redundant Pitch->Z (Drivers: Y & Pitch)
% 50 BCCT Coupling Redundant Pitch->Roll (Drivers: Y & Pitch)
% 51 BCCT Coupling Redundant Pitch->Yaw (Drivers: Y & Pitch)
% 52 BCCT Coupling Redundant Yaw->Y (Drivers: X & Yaw)
% 53 BCCT Coupling Redundant Yaw->Roll (Drivers: X & Yaw)
% 54 BCCT Coupling Redundant Yaw->Pitch (Drivers: X & Yaw)
% 55 TCT Total Captures (out of 12)
% 56 TCT Average Time to Capture (out of total captures)
% 57 TCT Average Positional Error on Capture
% 58 TCT Average Angular Error on Capture
% 59 TCT Total Collisions (in all 12 trials)
% 60 TCT Total Failed Grapples (in all 12 trials)

%% compileSubject.m
% Generates the compiled data struct for a given subject
% subjectRecord.name is the three-letter subject code/identifier
% subjectRecord.subjectData is the array of manually entered data common
%   to all sessions with the subject
% subjectRecord.PBCCT is the struct result of the PBCCT
% subjectRecord.SBCCT is the struct result of the SBCCT
% subjectRecord.session is the array of data structs for each session

function subjectRecord = compileSubject(folder, plotFig)

if nargin < 2
    plotFig = 0;
end

% initialize subjectRecord
name = folder(end-2:end);
subjectRecord = struct('name', '', 'PBCCT', {}, 'SBCCT', {}, ...
    'subjectData', [], 'session', []);
subjectRecord(1).name = name;

% get file data into subjectData
fileData = tdfread(strcat(folder, '/', name, 'SubjectData.txt'));
subjectRecord(1).subjectData = fileData.(name); % array of doubles

% generate data for PBCCT
dataFiles = dir(strcat(folder, '/Session 3/PBCCT*Test*.dat'));
subjectRecord(1).PBCCT = processBCCT(...
    strcat(folder, '/Session 3/'), dataFiles, plotFig);

```

```

% generate data for SBCCT
dataFiles = dir(strcat(folder, '/Session 3/SBCCT*Test*.dat'));
subjectRecord(1).SBCCT = processBCCT(...
    strcat(folder, '/Session 3/'), dataFiles, plotFig);

% generate data structs for each session
for i = 1:3
    sessionFolder = [folder ' /Session ' num2str(i) '/'];

    % generate BCCT results
    dataFiles = dir(strcat(sessionFolder, 'BCCT*Test*.dat'));
    BCCT = processBCCT(...
        sessionFolder, dataFiles, plotFig);

    subjectRecord(1).session(i).CCmatrix = BCCT.CCmatrix;
    subjectRecord(1).session(i).CCmatrix2 = BCCT.CCmatrix2;
    subjectRecord(1).session(i).errors = BCCT.errors;

    % compile TCT results
    % TCT.captures is the number of successful TCT captures (out of 12)
    % TCT.avgTime is the average time to capture for successful trials
    % TCT.avgPerror is the average position error at successful capture
    % TCT.avgAerror is the average angular error at successful capture
    % TCT.totalCollisions is the total number of collisions over 12 trials
    % TCT.totalFailedGrapples is the total number of failed grapples
    % TCT.MSJ is the mean squared jerk across all successful grapples

    % There should only be one valid dataFile per folder
    dataFiles = dir(strcat(sessionFolder, 'SubjEXP_*_TCT_Exp_summary.dat'));
    % There will be 13 raw input data files
    inputFiles = dir(strcat(sessionFolder, 'SubjEXP_*_TCT_Exp_trial*_input.dat'));
    % Note that first 3 columns are ignored by MATLAB during import
    % The 12 header lines below includes skipping the warmup trial
    fileData = importdata(strcat(sessionFolder, dataFiles(1).name), '\t', 12);
    fileData = fileData.data;
    TCT = struct('captures', 0, 'avgTime', 0, 'avgPerror', 0, 'avgAerror', 0, ...
        'totalCollisions', 0, 'totalFailedGrapples', 0, 'MSJ', 0); % initialize
    for j = 1:12
        % do if successful grapple:
        if fileData(j,4) > 0 % non-zero grapple time means success
            TCT.captures = TCT.captures + 1;
            TCT.avgTime = TCT.avgTime + fileData(j,4);
            TCT.avgPerror = TCT.avgPerror + fileData(j,5);
            TCT.avgAerror = TCT.avgAerror + fileData(j,6);
            % note the +1 below to skip warm-up trial
            inputs = importdata(strcat(sessionFolder, inputFiles(i+1).name), '\t', 1);
            framerate = length(inputs.data)/(inputs.data(end,1)-inputs.data(1,1));
            jerk = diff(inputs.data(:,3:end),3)/framerate^3;
            % multiply sum of squared jerks by duration^5 to get dimensionless
            % also should divide by amplitude^2 but all tasks have similar
            % amplitude.

```

```

        % note that 6 DOF inputs are normalized to a scalar
        totalsqjerk = norm(sum(jerk.^2))*fileData(j,4)^5;
        TCT.MSJ = TCT.MSJ + totalsqjerk;
    end
end
% simple averaging & totaling calculations
TCT.avgTime = TCT.avgTime/TCT.captures;
TCT.avgPerror = TCT.avgPerror/TCT.captures;
TCT.avgAerror = TCT.avgAerror/TCT.captures;
TCT.totalCollisions = sum(fileData(:,14));
TCT.totalFailedGrapples = sum(fileData(:,13));
TCT.MSJ = TCT.MSJ/TCT.captures;

% add to record
subjectRecord(1).session(i).TCT = TCT;

% extract manually entered session data
% get file data into subjectData
fileData = tdfread(...
    strcat(sessionFolder,name,'Session',num2str(i),'Data.txt'));
subjectRecord(1).session(i).data = fileData.(name); % array of doubles
end

```

```

%% compileBCCST.m
% Condenses (P/S)BCCT data into a single line representation

function line = compileBCCT(CCmatrix,CCmatrix2,errors,stationary)

```

```

% determine BCCT type
dims = size(errors); % first element is number of rows,
                    % 6 for BCCT, 2 for P/SBCCT
if dims(1) == 2 && ~stationary
    partial = 1;
else
    partial = 0;
end

```

```

% Calculate average errors
if stationary || partial
    % just two trials, so normal average
    avgError = mean(errors);
else
    % average two non-zero values per 6-row column
    avgError = mean(errors)*3;
end

```

```

% Compile CCmatrix appropriately into one line
if stationary
    Ccline = [CCmatrix(1,2:6) CCmatrix(5,1:4) CCmatrix(5,6)];

```

```

elseif partial
    CCline = [CCmatrix(1,2:4) CCmatrix(1,6) CCmatrix(2,1) CCmatrix(2,3)...
              CCmatrix(2,5:6) CCmatrix(4,1) CCmatrix(4,3) CCmatrix(4,5:6)...
              CCmatrix(5,2:4) CCmatrix(5,6)];
else % normal BCCT
    CCline = [CCmatrix(1,2:6) CCmatrix(2,1) CCmatrix(2,3:6)...
              CCmatrix(3,1:2) CCmatrix(3,4:6) CCmatrix(4,1:3)...
              CCmatrix(4,5:6) CCmatrix(5,1:4) CCmatrix(5,6)...
              CCmatrix(6,1:5) CCmatrix2(1,2:4)...
              CCmatrix2(2,1) CCmatrix2(2,3) CCmatrix2(2,6)...
              CCmatrix2(3,1:2) CCmatrix2(3,5) CCmatrix2(4,1)...
              CCmatrix2(4,5:6) CCmatrix2(5,3:4) CCmatrix2(5,6)...
              CCmatrix2(6,2) CCmatrix2(6,4:5)];
end

line = [avgError CCline];

end

```

```

%% processBCCT.m
% Generates the CC matrix for a given data folder
% BCCT.CCmatrix is the 6x6 coupling matrix (row # affects column #)
% BCCT.CCmatrix2 is the redundant matrix of coupling factors,
% not applicable for BCCT & SBCCT
% BCCT.errors is RMS tracking error in each axis (column #),
% for each trial (row #)

```

```
function BCCT = processBCCT(folder, dataFiles, plotFig)
```

```
numFiles = length(dataFiles);
```

```
% initialize matrix with knowledge that diagonals are 1
BCCT = struct('CCmatrix',eye(6),'CCmatrix2',eye(6),...,
             'errors',zeros(numFiles,6));
```

```
if numFiles == 0 % no valid data files
    CCmatrix = []; % so return null (don't add to CCmatrices)
    CCmatrix2 = [];
    error = 'Cannot find BCCT files!'
end
```

```
for k = 1:numFiles
```

```

    %% Setup
    CC = zeros(6);
    CC2 = zeros(6);
    stationary = dataFiles(k).name(1) == 'S'; % is this a SBCCT?
    partial = dataFiles(k).name(1) == 'P'; % is this a PBCCT?
    name = strcat(folder,dataFiles(k).name); % file name/location
    data = importdata(name,'\t',4); % ignore header (4 lines)
    data = data.data; % overloading variable for convenience
    L = (60*5+40)*30; % max input length (5 min 40 s @ 30 Hz)

```

```

L = 2^nextpow2(L); % pad to a good FFT length

%% Process errors

% unwrap and de-reverse angles
for j = 1:length(data)
    % TARGET
    % unwrap yaw
    if data(j,20) < -1 % 1 degree tolerance
        data(j,20) = data(j,20) + 360;
    end
    if data(j,20) < 1 % 1 degree tolerance
        data(j,20) = data(j,20) + 180;
        % use yaw info to de-reverse pitch
        if data(j,19) > 1 % 1 degree tolerance
            data(j,19) = 180-data(j,19);
        end
        if data(j,19) < -1 % 1 degree tolerance
            data(j,19) = -180-data(j,19);
        end
    end
    % unwrap roll
    if data(j,18) < -179 % 1 degree tolerance
        data(j,18) = data(j,18)+180;
    elseif data(j,18) > 179 % 1 degree tolerance
        data(j,18) = data(j,18)-180;
    end

    % CURSOR
    % unwrap yaw
    if data(j,14) < -1 % 1 degree tolerance
        data(j,14) = data(j,14) + 360;
    end
    if data(j,14) < 1 % 1 degree tolerance
        data(j,14) = data(j,14) + 180;
        % use yaw yaw info to de-reverse pitch
        if data(j,13) > 1 % 1 degree tolerance
            data(j,13) = 180-data(j,13);
        end
        if data(j,13) < -1 % 1 degree tolerance
            data(j,13) = -180-data(j,13);
        end
    end
    % unwrap roll
    if data(j,12) < -179 % 1 degree tolerance
        data(j,12) = data(j,12)+180;
    elseif data(j,12) > 179 % 1 degree tolerance
        data(j,12) = data(j,12)-180;
    end
end

% take care of any remaining excessive wraps
% data(:,9:14)=unwrap(data(:,9:14));

```

```

%     if plotFig
%         figure
%         plot(data(:,9:14))
%         legend('x','y','z','roll','pitch','yaw');
%         title('cursor')
%         figure
%         plot(data(:,15:20))
%         legend('x','y','z','roll','pitch','yaw');
%         title('target')
%     end

errors = data(:,9:14)-data(:,15:20); % tracking error

% get RMS, normalized to 1 being no action
errors(:,1:3) = errors(:,1:3)/0.5200; % translations
errors(:,4:6) = errors(:,4:6)/90/0.8325; % rotations
rmserror = sqrt(sum(errors.^2)/length(errors)); % calculate RMS

%     p=figure;
%     plot(data(:,1),errors)
%     title(name)
%     legend('X','Y','Z','Roll','Pitch','Yaw')
%     xlabel('Time (s)')
%     ylabel('Normalized Error')
%     set(p, 'Name', name)
%     saveas(p,['Figures/' stprep(stprep(name,'/','-'),'.dat','')...
%             'Errors.jpg'], 'jpg');
%     close

if plotFig
    p=figure;
    plot(data(:,1),data(:,3:8))
    title(name)
    legend('X','Y','Z','Roll','Pitch','Yaw')
    xlabel('Time (s)')
    ylabel('Inputs')
    set(p, 'Name', name)
end

BCCT.errors(k,:) = rmserror; % save with BCCT record

%% Process cross-coupling
inputs = data(:,3:8); % joystick inputs
% process inputs, remove mean
inputs = inputs-ones(length(inputs),1)*mean(inputs);
% pad to standard length
inputs = padarray(inputs,[L-length(inputs),0],'post');

Fs = 29.84; % emperically determined actual sample rate
% this is pretty consistent across samples
NFFT = L; % already padded to a power of 2 for fft
f = Fs/2*linspace(0,1,NFFT/2+1); % fft frequency scale

% array indices of excitation. 1 & 4 right hand. 2 & 3 left hand.

```

```

% next smallest numbers used based on where the peaks generally
% are in the actual data. Should be .03, .07, .17, .19
excitation = [find(f>0.0291,1), find(f>0.0692,1), ...
              find(f>0.169,1), find(f>0.189,1)];

%% Plot single-sided amplitude spectrum for left cursor/target

% generate raw ffts
y = fft(inputs.*repmat(hanning(L),1,6),NFFT)/L;
y = 2*y(1:NFFT/2+1,:); % prior to abs()

if plotFig
    p=figure;
    plot(f,abs(y))
    title(name)
    legend('X','Y','Z','Roll','Pitch','Yaw')
    xlabel('Frequency (Hz)')
    ylabel('Magnitude of Joystick Deflections (A.U.)')
    xlim([0 0.25])
    ylim([0 0.1])
    set(p, 'Name', name)
end
% saveas(p,['Figures/' strrep(strrep(name,'/','-'),'dat','')...
%         'FFT.jpg'], 'jpg');
% close

% % ideal calculation of peaks
% peaks = [y(excitation(1),:); y(excitation(2),:); ...
%         y(excitation(3),:); y(excitation(4),:)]; % peak y values

% calculate peak y values as max within one frequency unit of nominal
% (allows for a barely measurable amount of frequency shift)
peaks = zeros(4,6);
for i = 1:4
    for j = 1:6
        peaks(i,j) = max(y((excitation(i)-1):(excitation(i)+1),j));
    end
end

% get DOFList from Python code. Just put in semicolons.
if stationary
    DOFList = [ [1,5,2,4,3,6]; [5,1,4,2,6,3] ];
elseif partial
    DOFList = [ [1,5,2,4,3,6]; [2,4,3,6,1,5] ];
else
    DOFList = [ [1,5,2,4,3,6]; [2,4,3,6,1,5]; [3,6,2,5,1,4]; ...
              [2,5,1,6,3,4]; [1,6,3,4,2,5]; [3,4,1,5,2,6] ];
end
% parse out the test number
testnum = str2double(name(strfind(name,'Test')+4));

% how strong are the driving signals? multiply values of complex fft
% at the excitation frequencies.
% this helps to reduce white noise influence compared to just adding

```

```

% or taking a single peak
signalL = peaks(2,DOFList(testnum,1))*peaks(3, DOFList(testnum,1));
% need to then take abs value. sqrt is just to "undo" the earlier
% multiplication of two peaks & prevent number from being very small
signalL = sqrt(abs(signalL));
% same thing for right hand signal strength
signalR = peaks(1,DOFList(testnum,2))*peaks(4, DOFList(testnum,2));
signalR = sqrt(abs(signalR));

% create CCmatrix (influence of row #, on column #)
for i = 1:6
    % don't count excited axes (low signal to noise)
    if (i ~= DOFList(testnum,1)) && (i ~= DOFList(testnum,2)) ...
        || stationary % run through all for stationary since there
            % is only one excited axis

        % first do influences caused by left side (or right for the
        % second trial of a SBCCT)
        CCLmagnitude = sqrt(abs(peaks(2,i)*peaks(3,i)));
        % check if already a CC value here
        if BCCT.CCmatrix(DOFList(testnum,1),i) ~= 0
            % yes, so put new value in CCmatrix2
            CC2(DOFList(testnum,1),i) = CCLmagnitude/signalL;
        else
            % not (yet) redundant
            CC(DOFList(testnum,1),i) = CCLmagnitude/signalL;
        end

        if ~stationary
            % same for influences caused by right side
            CCRmagnitude = sqrt(abs(peaks(1,i)*peaks(4,i)));
            if BCCT.CCmatrix(DOFList(testnum,2),i) ~= 0
                CC2(DOFList(testnum,2),i) = CCRmagnitude/signalR;
            else
                CC(DOFList(testnum,2),i) = CCRmagnitude/signalR;
            end
        end
    end
end

% add current axis-pair result to overall matrix
BCCT.CCmatrix = BCCT.CCmatrix+CC;
BCCT.CCmatrix2 = BCCT.CCmatrix2+CC2;
end

```

Appendix I: Multiplicative Gap Filter

I.1 Abstract

This report introduces some central concepts related to signal processing filters based on convolution. A filtering technique for reducing noise and isolating multiple signal peaks of known separation is then described, with application to identifying a noise-corrupted Fourier-transformed signal of known frequency content.

I.2 Introduction to FIR Filters

The convolution of f and g is written $f * g$ and is defined as

$$(f * g)(t) \\ \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau = \int_{-\infty}^{\infty} f(t - \tau) g(\tau) d\tau.$$

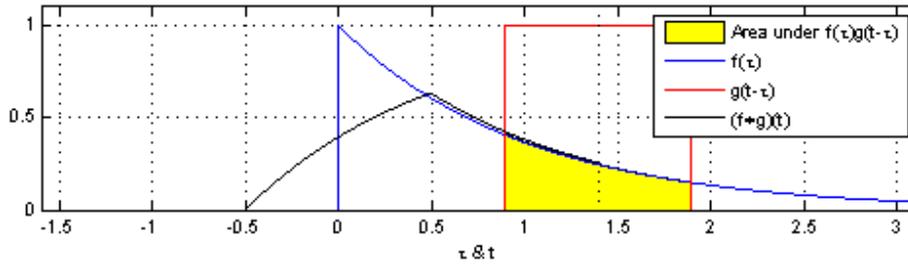
The second line is a consequence of the commutative properties of convolution. Note that the cross-correlation of f and g is defined similarly, except that for cross-correlation, the complex conjugate of f is used.

For discrete signals, convolution is defined as

$$(f * g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f[m] g[n - m] \\ = \sum_{m=-\infty}^{\infty} f[n - m] g[m].$$

If f is defined to be the input signal to a convolution filter producing $f * g$, then g is referred to as the filter kernel, and represents the impulse response of the filter. Such convolution-based filters are known as Finite Impulse Response (FIR) filters, since the filter output represents a finite sum of impulse responses caused by each input data point.

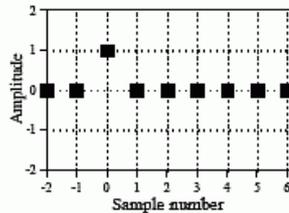
Convolution is a common technique in signal processing for noise reduction. Perhaps the most common application is the moving average filter, which averages the last m data points to achieve noise reduction proportional to the square root of m , assuming the noise is normally distributed and has a constant standard deviation at all data points [77]. The kernel for this filter is essentially a square wave pulse, which is multiplied through all time by the signal to be filtered. An illustration depicting this moving average calculation can be seen below, with the input in blue, the “sliding” kernel in red, and the filter output in black:



Incidentally, the moving average filter is optimal in time domain noise reduction, but has very poor frequency domain properties. By changing the shape of the kernel (for example, by using a sinc function, which has ideal frequency domain properties), a variety of time domain and frequency domain filters can be achieved. To provide a sense of the variety of mathematical operations possible simply by changing the kernel, several examples are shown below:

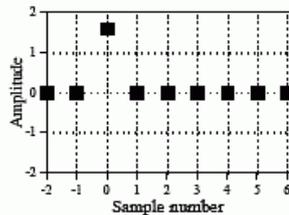
a. Identity

The delta function is the *identity* for convolution. Convoluting a signal with the delta function leaves the signal unchanged. This is the goal of systems that transmit or store signals.



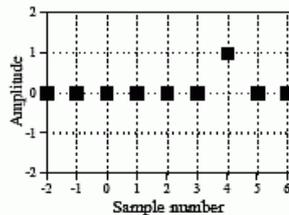
b. Amplification & Attenuation

Increasing or decreasing the amplitude of the delta function forms an impulse response that *amplifies* or *attenuates*, respectively. This impulse response will amplify the signal by 1.6.



c. Shift

Shifting the delta function produces a corresponding shift between the input and output signals. Depending on the direction, this can be called a *delay* or an *advance*. This impulse response delays the signal by four samples.



d. Echo

A delta function plus a shifted and scaled delta function results in an *echo* being added to the original signal. In this example, the echo is delayed by four samples and has an amplitude of 60% of the original signal.

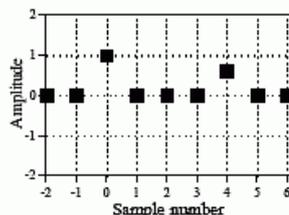
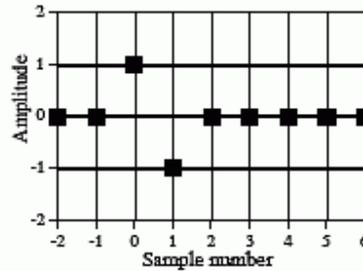


FIGURE 7-1 Simple impulse responses using shifted and scaled delta functions.

a. First Difference

This is the discrete version of the *first derivative*. Each sample in the output signal is equal to the *difference* between adjacent samples in the input signal. In other words, the output signal is the *slope* of the input signal.



b. Running Sum

The running sum is the discrete version of the *integral*. Each sample in the output signal is equal to the sum of all samples in the input signal to the *left*. Note that the impulse response extends to infinity, a rather nasty feature.

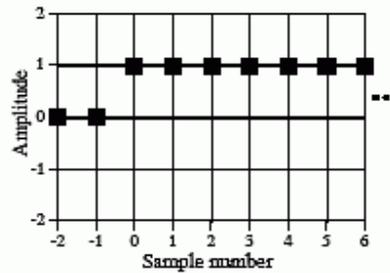


FIGURE 7-2 Impulse responses that mimic calculus operations.

However, regardless of the specific kernel used, commonly used filters in modern digital signal processing (DSP) practice, even so-called “custom” FIR filters and recursion-based infinite impulse response (IIR) filter, appear to be based on a linear sum of these impulse responses. The different types of filters used in modern practice can be summarized in the table below:

		FILTER IMPLEMENTED BY:	
		Convolution <i>Finite Impulse Response (FIR)</i>	Recursion <i>Infinite Impulse Response (IIR)</i>
FILTER USED FOR:	Time Domain <i>(smoothing, DC removal)</i>	Moving average (Ch. 15)	Single pole (Ch. 19)
	Frequency Domain <i>(separating frequencies)</i>	Windowed-sinc (Ch. 16)	Chebyshev (Ch. 20)
	Custom <i>(Deconvolution)</i>	FIR custom (Ch. 17)	Iterative design (Ch. 26)

TABLE 14-1 Filter classification. Filters can be divided by their *use*, and how they are *implemented*.

I.3 FIR Filter for Identifying Multiple Peaks of Known Separation

Norbert Wiener introduced in the 1940's the idea of comparing a noisy signal to the known properties of a desired signal for noise reduction purposes. The FIR Wiener filter for a discrete series can be written as

$$x[n] = \sum_{i=0}^N a_i w[n - i].$$

where $x[n]$ is the output of the filter, $w[n]$ is the input to be filtered, and a_i are the coefficients of the filter, which can be solved as

$$\mathbf{T}\mathbf{a} = \mathbf{v}$$

$$\Rightarrow \begin{bmatrix} R_w[0] & R_w[1] & \dots & R_w[N] \\ R_w[1] & R_w[0] & \dots & R_w[N-1] \\ \vdots & \vdots & \ddots & \vdots \\ R_w[N] & R_w[N-1] & \dots & R_w[0] \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} R_{sw}[0] \\ R_{sw}[1] \\ \vdots \\ R_{sw}[N] \end{bmatrix}$$

where R_w is the autocorrelation of the input signal and R_{sw} is the crosscorrelation of the desired signal with the input signal. This can be shown to give the minimum squared error between $x[n]$ and the desired signal $s[n]$, thus reducing noise, which is assumed to be normally distributed and uncorrelated with the input (Wiener filter).

For the case where the desired signal is known to be a sum of Dirac impulse functions, R_{sw} will only be non-zero at the location of these impulses, giving non-zero coefficients of a_i at these locations. This can be applied to the frequency domain measurement of a set of signals of known frequency, since in the frequency domain, the desired signal becomes a sum of delta functions located at the known frequencies. Although in the case of a bimanual cross-coupling measurement, Wiener's assumption of uncorrelated, normally distributed noise is not necessarily valid, we can still achieve empirically useful results by applying this noise-reducing principle of using non-zero FIR filter coefficients only at the known locations of the desired signal.

Consider the discrete Fourier transform of a sum-of-sinusoids time domain signal with discretized frequency domain peaks located at $i_p = i_1, i_2, \dots, i_n$. Processing this finite signal using a kernel similar to the Echo kernel of Figure 7-1d, but with non-zero coefficients located at $i_k = 0, i_2 - i_1, \dots, i_n - i_1$ and zero coefficients elsewhere results in a desirable reduction of noise emphasizing the first desired peak above all else. The variance of any uncorrelated, zero-mean noise would be reduced by a factor of $1/n$ due to averaging over n locations. In the case of $n=2$, this can be shown as

$$E[(e(t)+e(t+\tau))/2]^2 = E[(e(t)/2)^2 + e(t)e(t+\tau) + (e(t)/2)^2] = \sigma^2/4 + 0 + \sigma^2/4 = \sigma^2/2$$

where σ^2 is the variance of uncorrelated, zero mean noise $e(t)$, and $E[X]$ is the expectation of X .

I.4 Multiplicative Gap Filter

The noise reduction capabilities of the FIR filter discussed above can be improved drastically by considering a non-linear implementation. Whereas common DSP practice involves the use of linear filters, either FIR or IIR, which involve a summation of impulse responses, a multiplication of impulse responses would define a non-linear filter.

The Volterra series exemplifies such a multiplicative impulse response model. A continuous time-invariant system with $x(t)$ as input and $y(t)$ as output can be expanded in Volterra series as:

$$y(t) = k_0 + \sum_{n=1}^{\infty} \frac{1}{n!} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} k_n(t_1, t_2, \dots, t_n) x(t-t_1) x(t-t_2) \cdots x(t-t_n) dt_1 dt_2 \cdots dt_n.$$

Where k_n is called the n -th order Volterra kernel, which is essentially a higher-order impulse response (Volterra Series). Note the similarity to convolution, but with multiple multiplicative input terms.

In the case of noise reduction in the frequency domain bimanual cross-coupling signal, we use a filter of a form similar to the n -th order Volterra series. For example, the 2nd order Volterra series can be written as

$$\begin{aligned} y(t) &= H_2[x(t)] \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) x(t - \tau_1) x(t - \tau_2) d\tau_1 d\tau_2 \end{aligned}$$

Using a Volterra kernel h_2 consisting of a sum of Dirac delta functions located at $i_k = 0, i_2 - i_1$ as before in the linear filter case, where i_1 and i_2 are the locations of the peaks in the desired signal, we have for the two-frequency case

$$y(t) = x(t)x(t+\tau)$$

However, to preserve the magnitude of the peak response, we take the n -th root of this function to arrive at our Multiplicative Gap Filter (MGF):

$$y(k) = [x(k)x(k+\tau_1)\dots x(k+\tau_n)]^{1/n}$$

Because of the multiplicative effect, the variance of any uncorrelated, zero mean noise is theoretically reduced to zero, since

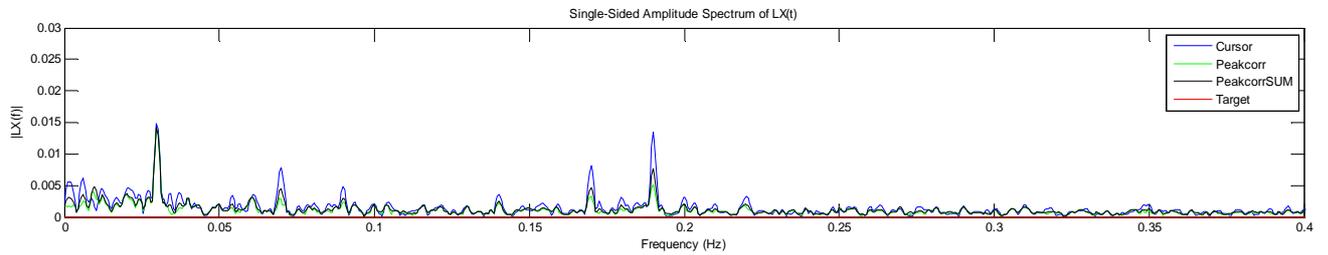
$$E[e(t)e(t+\tau)] = 0$$

by definition of being uncorrelated with itself (white noise).

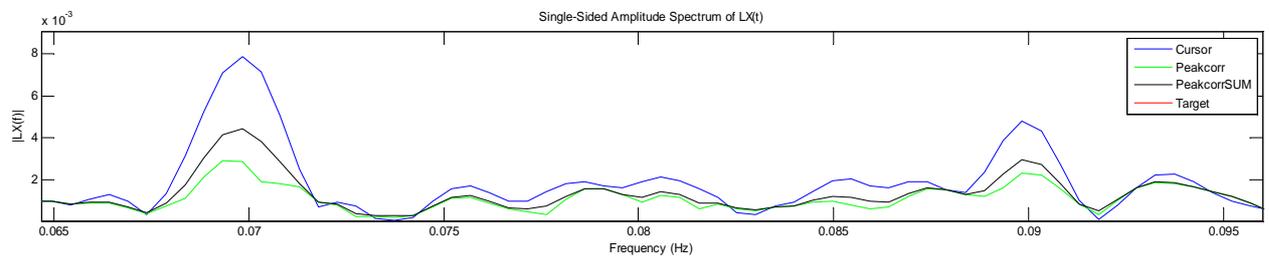
I.5 Results

The figure below shows the generally applicable case when the desired bimanual cross-coupling signal is the dominant non-noise signal in the input. The input signals are in blue, MGF filter results are in green, and FIR filter results are in black. Note that both the MGF filter and

the FIR filter are designed to produce an unchanged magnitude of the base frequency, such that the magnitude of cross-coupling can be measured.

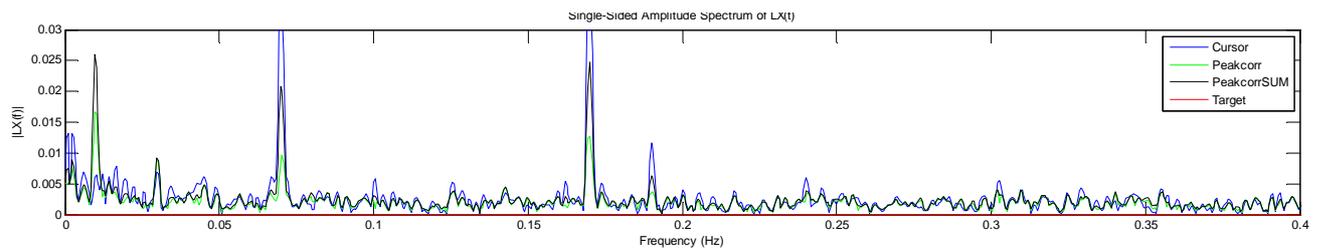


As can be seen in this close-up of some undesired noise components, the MGF is superior to the linear “echo” filter in reducing these peaks:



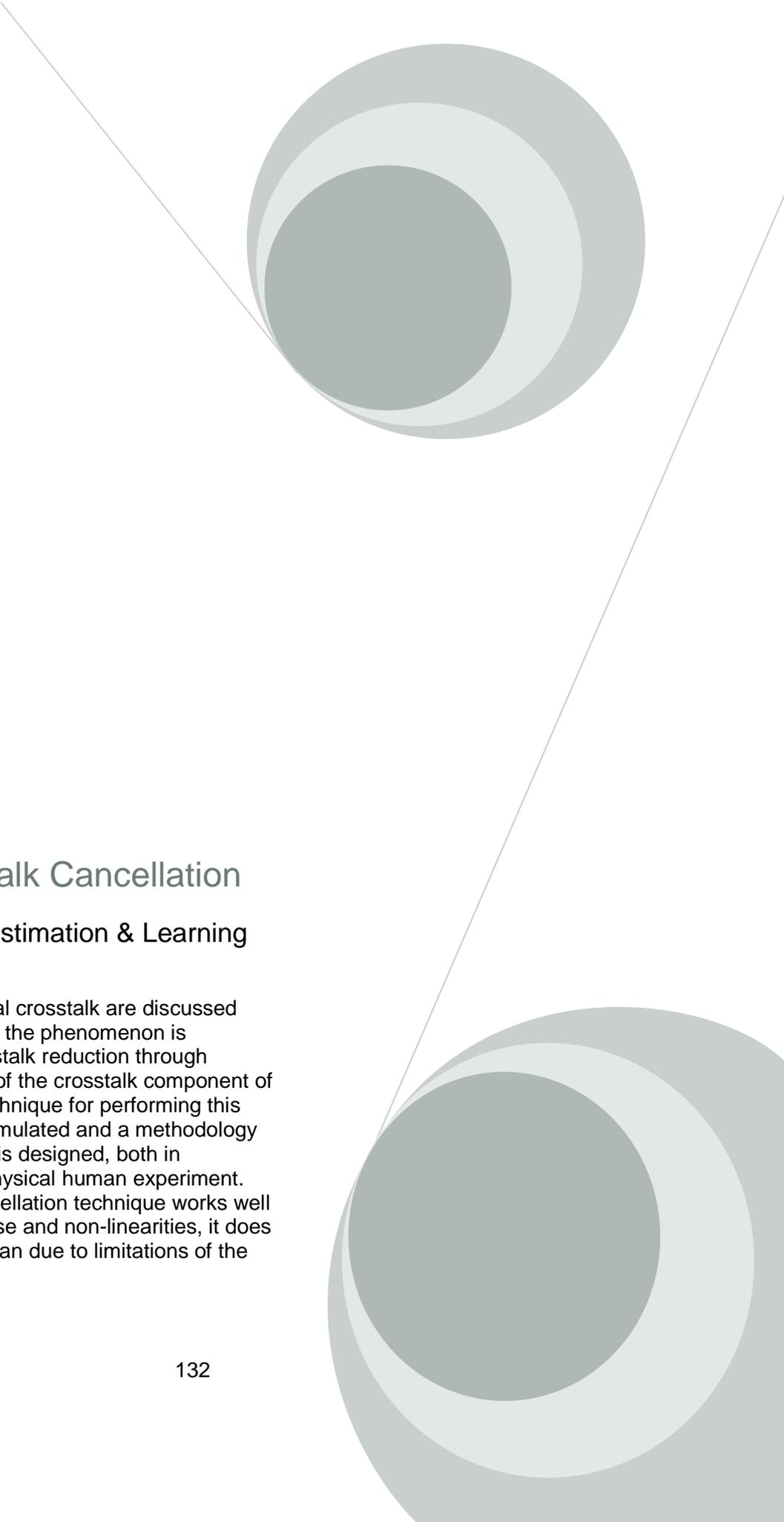
This helps greatly in increasing the robustness of signal detection and identifying the expected frequency components above the background noise.

As seen below, in a case where the desired signal is much smaller than undesired components, these filters are not able to show the desired base frequency as a dominant peak, which may be desirable from the point of view of quantitatively demonstrating the significance of the coupling effect. Of course in the case of white noise, the filters would simply produce white noise.



Appendix J: Intermanual Cross-Coupling Cancellation

The next 23 pages were written for a class project for 2.160 Identification, Estimation and Learning, taught by Prof. Asada in Spring 2011.



Bimanual Crosstalk Cancellation

2.160 Identification, Estimation & Learning

Possible causes of bimanual crosstalk are discussed and a rudimentary model of the phenomenon is developed to facilitate crosstalk reduction through estimation and subtraction of the crosstalk component of a manual movement. A technique for performing this crosstalk cancellation is formulated and a methodology for testing its effectiveness is designed, both in simulation and through a physical human experiment. Although the crosstalk cancellation technique works well in simulation, even with noise and non-linearities, it does not apply well to a real human due to limitations of the static model used.

Victor Wang
5/13/2011

Introduction

Bimanual crosstalk refers to the effect of the movement of one hand on the movement of the other. For example, the familiar “pat your head and rub your stomach” task shows how it can be difficult to coordinate two hands moving independently of each other, since intentional movement of one hand in one axis can result in unintended movement of the other hand in the same axis. The neurophysiological causes of this crosstalk have been discussed in the literature, but are not conclusively established. A general model that shows various contributing factors and is compatible with most other prevailing theories is shown in Figure 1.

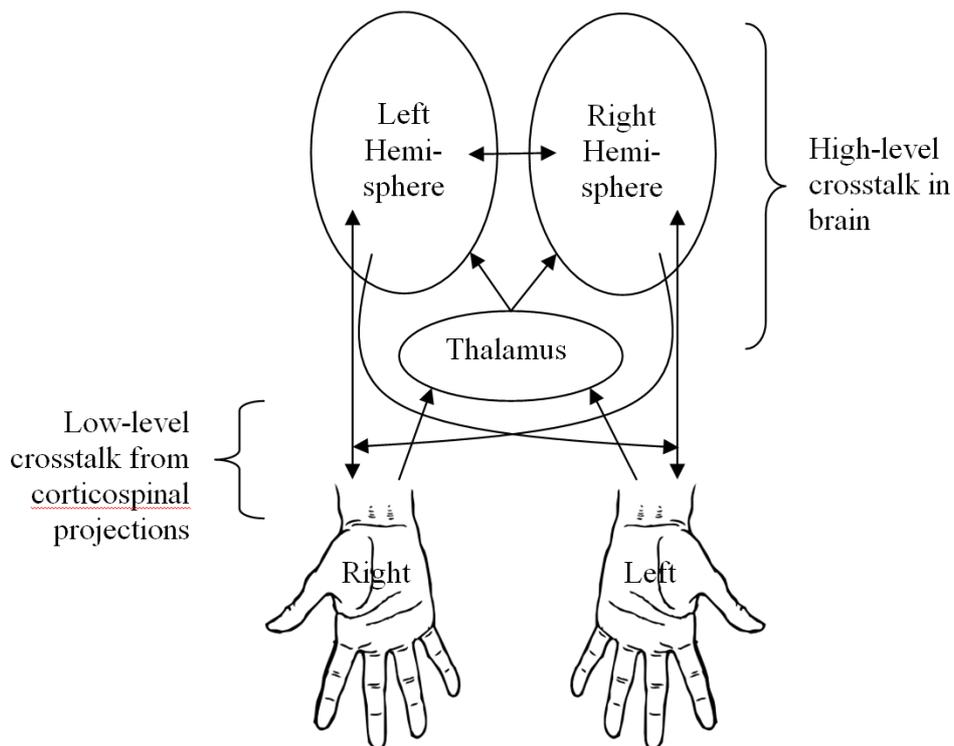


Figure 1 – A general model of why bimanual actions are coupled. Based on (Banerjee 2006)[78]. Communication across brain hemispheres is facilitated by the corpus callosum, while crossed corticospinal projections permanently “hardwire” a certain amount of crosstalk at a lower level, and interactions in the thalamus result in crosstalk in the feedback path.

The resulting crosstalk is undesired in a variety of tasks in which independent operation of each hand is required, such playing musical instruments or bimanual teleoperation of robotic systems such as the robotic arm on the International Space Station, which currently uses separate controllers for end-effector translation (left hand) and rotation (right hand) (Pontillo 2010). It would thus be desirable to be able to cancel out any undesired crosstalk such that performance is improved. To facilitate this crosstalk cancellation, a block diagram model of bimanual crosstalk is proposed in Figure 2.

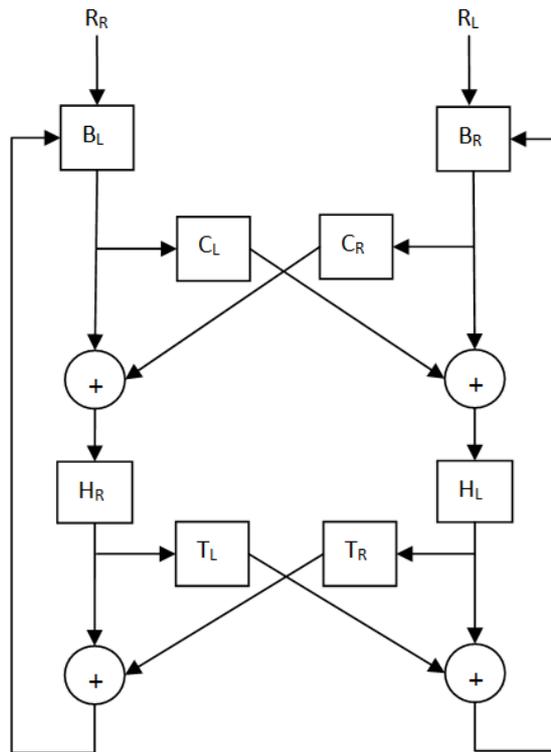


Figure 2 – Proposed block diagram model of bimanual crosstalk. R_R and R_L represent the desired motion of the right and left hand, respectively. B_L and B_R represent the transformations on this reference signal due to each brain hemisphere, incorporating proprioceptive and visual feedback. C_L and C_R represent the combined crosstalk due to callosal connections and also uncrossed corticospinal projections. H_R and H_L represent the signal transformation caused by propagation through the neuromuscular system. T_L and T_R represent the crosstalk due to thalamic interactions in the feedback path.

It should be noted that none of the blocks in Figure 2 are expected to be simple linear gains, and each will incorporate some amount of time delay. However, for certain types of bimanual tasks it may be feasible to simplify the model such that a straightforward adaptive cancellation system can be designed, as shown in Figure 3.

The primary goal of this project is to develop a rudimentary model of bimanual interaction and demonstrate in simulation that the principles covered in 2.160 Identification, Estimation and Learning can be used to identify any undesired interaction and at least partially cancel it out to achieve performance closer to ideal bimanual independence.

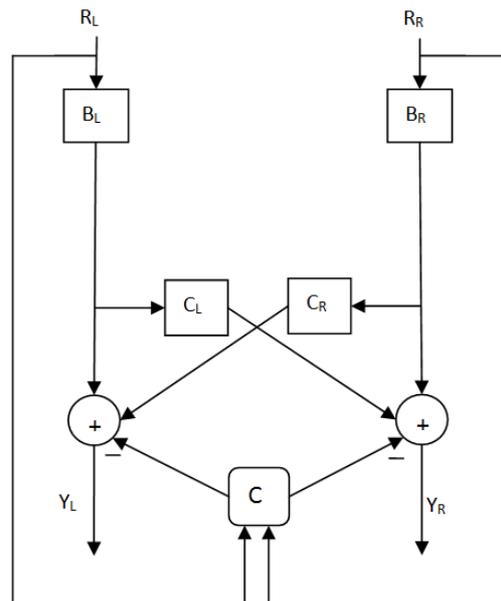


Figure 3 – Proposed adaptive crosstalk cancellation architecture, with compensator C. Feedback is removed as a simplification applicable for the transient initiation of a motion, before the time delayed feedback has had a chance to propagate through the model. B_L and B_R are determined in advance in a set of unimanual parameter estimation trials, and may take the form of finite impulse responses (FIR). In the second, bimanual, phase of training the compensator, inputs R_R and R_L are known by nature of being inputs to the human from the computer, and Y_R and Y_L are known by measuring joystick deflections. C_L and C_R FIR dynamics are determined by a parameter estimator that minimizes the difference in Y_R and Y_L between the actual bimanual case and the estimated equivalent unimanual case using knowledge of B_L and B_R . Once the system is trained, assuming the FIR model is still valid, the model-based compensator C should be able to use the established parameters to reduce crosstalk in a specific type of task without using knowledge of R_R and R_L .

Literature Survey

Possible models of bimanual coordination have been studied previously in the literature. The most commonly accepted model is the intermanual crosstalk model, in which independent motor plans are created for each limb in its contralateral brain hemisphere, and the overall behavior of each limb is influenced by mutual interaction between each motor plan (Banerjee 2006, Aramaki 2006, Wenderoth 2004). Different studies attribute this interaction, or crosstalk, to different factors, such as the parieto-premotor areas of the right hemisphere (Wenderoth 2004), a large system of intercerebellar relationships (Pollok 2007), and a subcortical locus of temporal coupling for discrete movements, with communication across the corpus callosum helping to synchronize continuous movements (Kennerley 2002). Most of the findings currently published appear to be in agreement with the general model in Figure 1, showing communication between hemispheres, the bi-hemispheric thalamocortical radiations which carry feedback signals from both hands, and the low-level crosstalk due to uncrossed corticospinal projections (generally, nerve fibers from the left hemisphere control the right hand, and vice versa, but not all fibers are crossed this way).

However, because some aspects of bimanual behavior are computed only in the hemisphere contralateral to the dominant hand (Viviani 1998), because the non-dominant motor cortex is in fact suppressed during bimanual finger movements (Aramaki 2006), and because in bimanual tasks the drive from the dominant to the non-dominant sensorimotor cortex has actually been shown to be stronger than the other way around (Serrien 2003), the model in Figure 2 extends the model proposed by Banerjee to allow for unequal coupling relationships between the hands.

There does not appear to be any literature on artificial cancellation of bimanual crosstalk.

Methodology

By keeping one hand steady while moving the other, it is easy to demonstrate that for crosstalk to occur, there must be simultaneous motion of both hands. Also, in order to achieve a strong coupling effect to study and to simplify the model by removing feedback, it is desirable for the driver of the crosstalk to be a discrete, quick motion. Therefore, the bimanual task selected is a continuous sinusoidal motion of the left hand from side-to-side, with discrete side-to-side motions of the right hand, which should affect the motion of the left hand through crosstalk. For the purposes of evaluating identification and cancellation techniques, this one-way analysis is adequate.

The sinusoidal motion of the left hand is commanded at a single frequency of 0.5 Hz based on what is humanly achievable while simultaneously following the right hand commands. The main goal here is just to keep the left hand moving as smoothly as possible such that crosstalk from the right hand can be easily identified. The right hand commands are selected as binary white noise, because this gives the highest crest factor and thus produces better signal-to-noise ratios for identification purposes. The noise sampling frequency is selected to be 2 Hz so that a human can actually respond to the commands adequately.

Ideally, referring to Figure 3, the B_L FIR would be identified through least squares estimation in a unimanual task, the B_R FIR would be identified in a separate unimanual task, and then the coupling C_R FIR would be identified by observing the difference in Y_L during a bimanual task compared to the estimated Y_L in a unimanual case. That is, collecting three sets of data,

$$\text{Input } R_L, \text{ Output } Y_L: \hat{B}_L = \operatorname{argmin}_{B_L} E \left[(Y_L(t) - \phi_{R_L}^T(t) B_L)^2 \right]$$

$$\text{Input } R_R, \text{ Output } Y_R: \hat{B}_R = \operatorname{argmin}_{B_R} E \left[(Y_R(t) - \phi_{R_R}^T(t) B_R)^2 \right]$$

$$\text{Input } R_L \& R_R, \text{ Output } Y_L: \hat{C}_R = \operatorname{argmin}_{C_R} E \left[\left(Y_L(t) - \phi_{R_L}^T(t) \hat{B}_L - \phi_{\phi_{R_R}^T(t) \hat{B}_R}^T(t) C_R \right)^2 \right]$$

However, human response to stimulus changes depending on workload, such that the transfer functions B_R and B_L are expected to exhibit larger delays when the human is attempting to follow two sets of commands versus a single unimanual command. Therefore, even though the outputs would be corrupted by crosstalk, it may be more appropriate to perform the unimanual identification of B_L and B_R using data from the bimanual case. That is, using a single set of data,

Input R_L & R_R , Output Y_L & Y_R :

$$\hat{B}_L = \operatorname{argmin}_{B_L} E \left[(Y_L(t) - \phi_{R_L}^T(t) B_L)^2 \right]$$

$$\hat{B}_R = \operatorname{argmin}_{B_R} E \left[(Y_R(t) - \phi_{R_R}^T(t) B_R)^2 \right]$$

$$\hat{C}_R = \operatorname{argmin}_{C_R} E \left[\left(Y_L(t) - \phi_{R_L}^T(t) \hat{B}_L - \phi_{\phi_{R_R}^T(t) \hat{B}_R}^T(t) C_R \right)^2 \right]$$

This is facilitated by the use of binary white noise as an input for the right hand, because the zero-mean, uncorrelated nature of the resulting cross-coupling term maintains consistency of the model. The variance of Y_L simply increases in a way that can be compensated for by taking enough data, as per the principles of asymptotic distribution of parameter estimates. Once B_L , B_R and C_R are identified, the efficacy of the technique can be tested by using a new set of data and subtracting the estimated coupling effect from Y_L . The error in the bimanual Y_L signal is considered to be the difference in the bimanual Y_L signal compared to the estimated Y_L signal using only the left-hand input. That is, using a single set of data,

Input R_L & R_R , Output Y_L :

$$Y_{L,cancelled}(t) = Y_L(t) - \phi_{\phi_{R_R}^T(t) \hat{B}_R}^T(t) \hat{C}_R$$

And the effectiveness of this cancellation is measured by comparing to

$$Y_{L,reference}(t) = \phi_{R_L}^T(t) \hat{B}_L$$

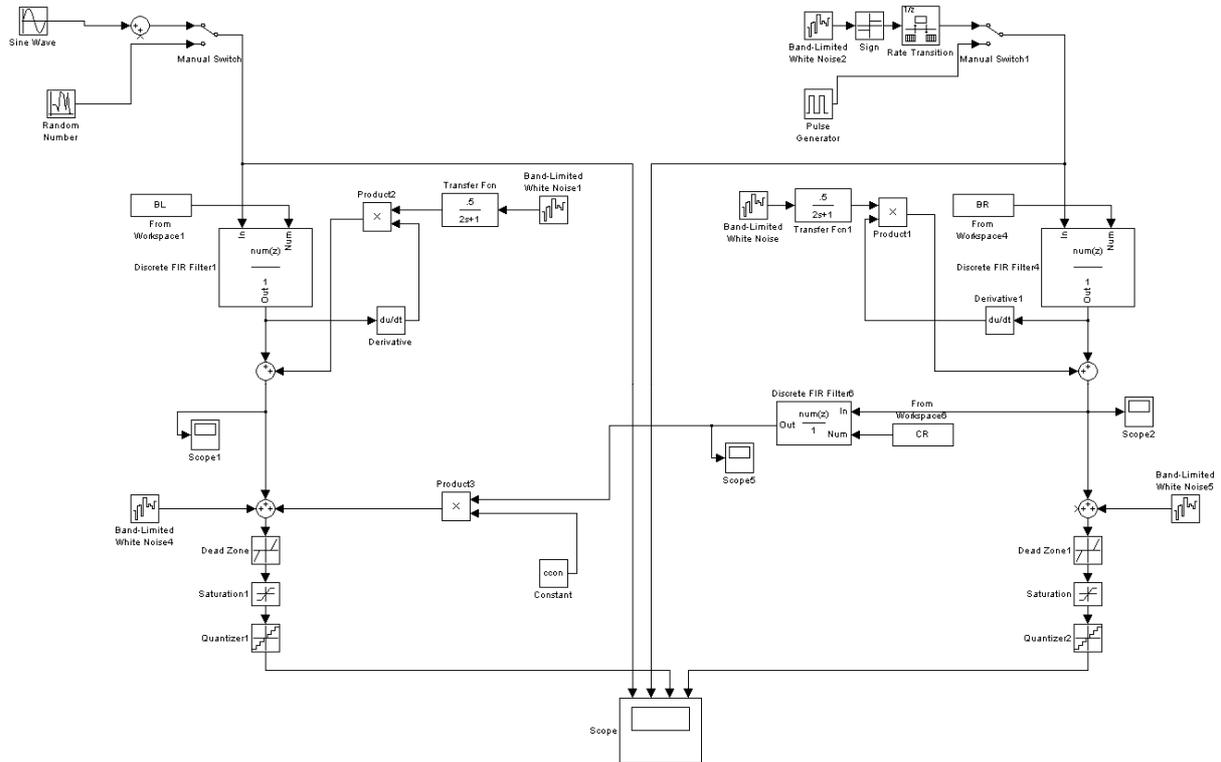
The techniques described are first tested in simulation and then tested using real data from a human.

Simulation Results

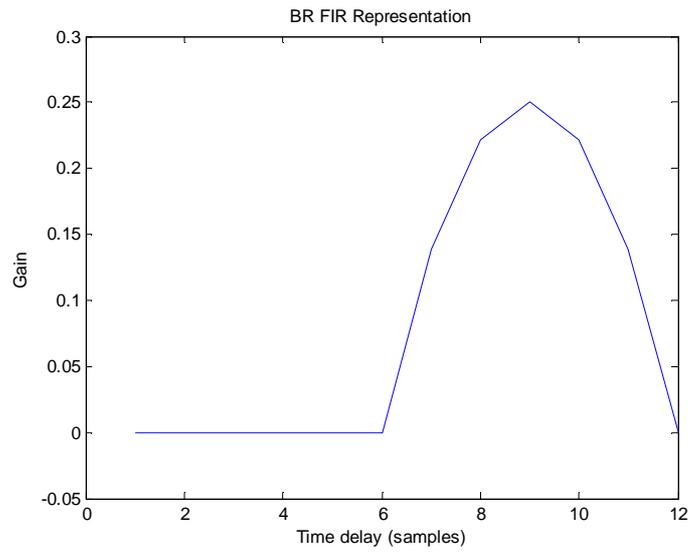
As discussed in the Term Project Interim report, a MATLAB/Simulink model was created to test the methodology of identifying the feedforward transfer function of each hand unimanually, then using these transfer functions to determine the coupling transfer functions using a bimanual trial, and finally using knowledge of all transfer functions to cancel out crosstalk effects in a separate bimanual trial. The inputs used in this preliminary trial were two superimposed sinusoids of different frequencies on each hand. The technique appeared to be feasible in the linear, noise-free case, and was able to reduce the mean squared error in the bimanual case when compared to the expected unimanual output.

To further verify this technique, the simulation model has been augmented with noise and other non-linearities expected from a real system collecting data from a human via a pair of joysticks. Specifically, a colored noise term that is scaled according to the rate of change of the base signal is added after each B_L and B_R block, to give a rough representation of neuromotor noise, which increases with increasing force applied. A white noise term is added to represent measurement/electrical noise, a dead zone represents the mechanical dead zone at the center

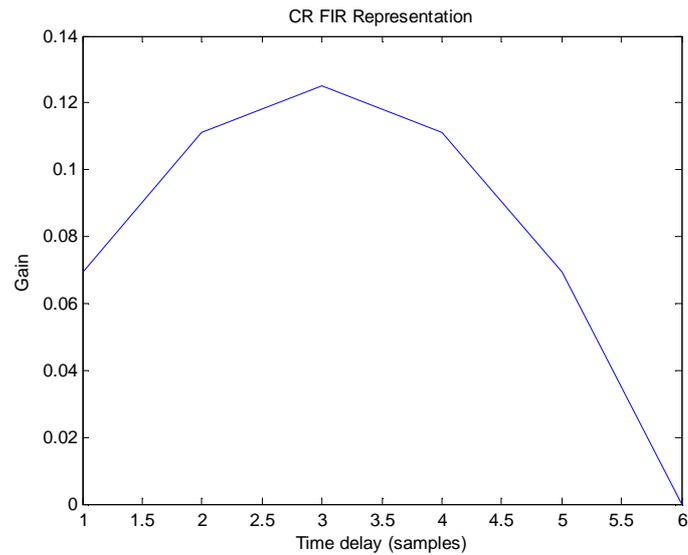
of each joystick, saturation limits represent the output limits of the joysticks, and finally the signals are quantized to represent analog-to-digital conversions. The revised Simulink model also uses the sinusoidal left-hand input and binary noise right-hand input devised in the Methodology section, and can be seen below:



MATLAB code (see Appendix) is used to perform the estimation and cancellation, and to generate reasonable FIR models of the original B_L , B_R and C_R blocks to be identified. For example, the underlying FIR for B_R can be seen below, essentially representing a low-pass filter with time delay:

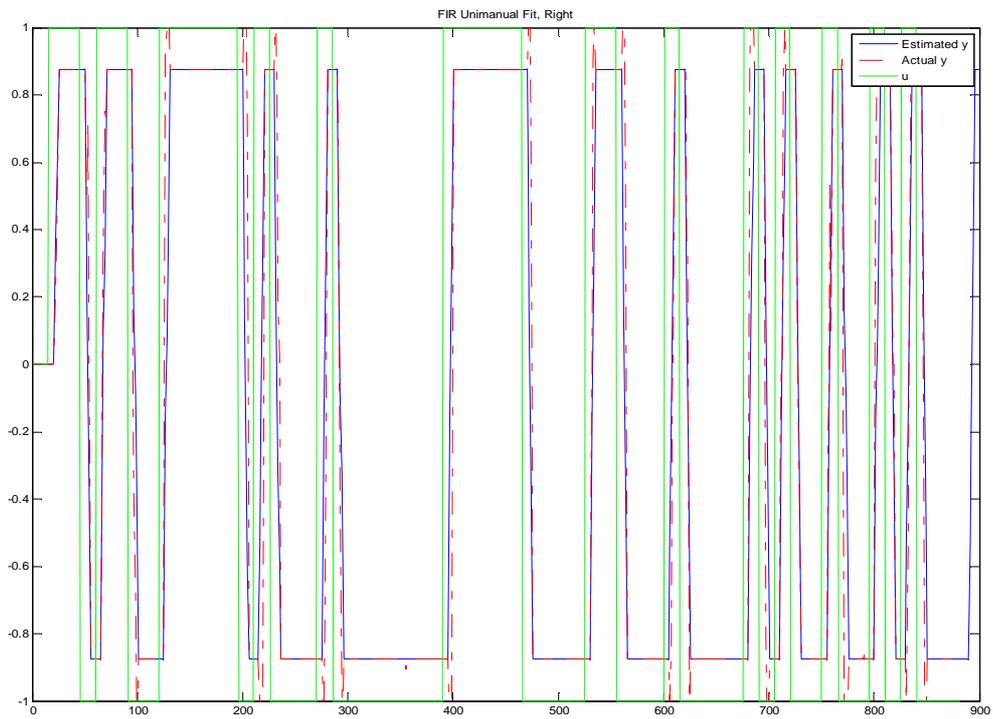
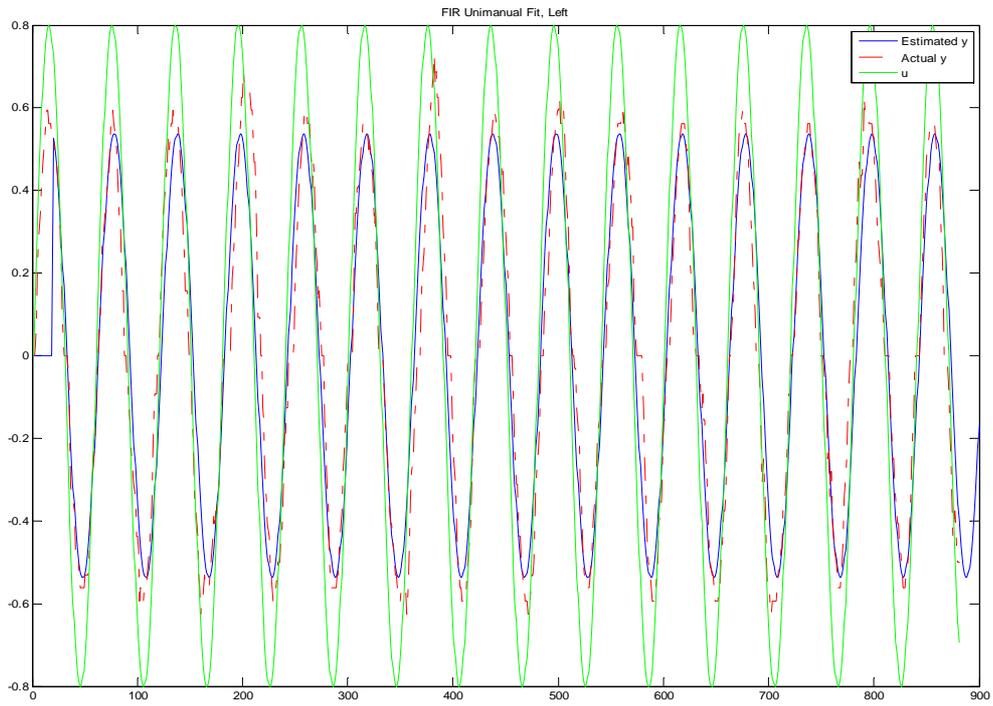


And the CR used is as follows:

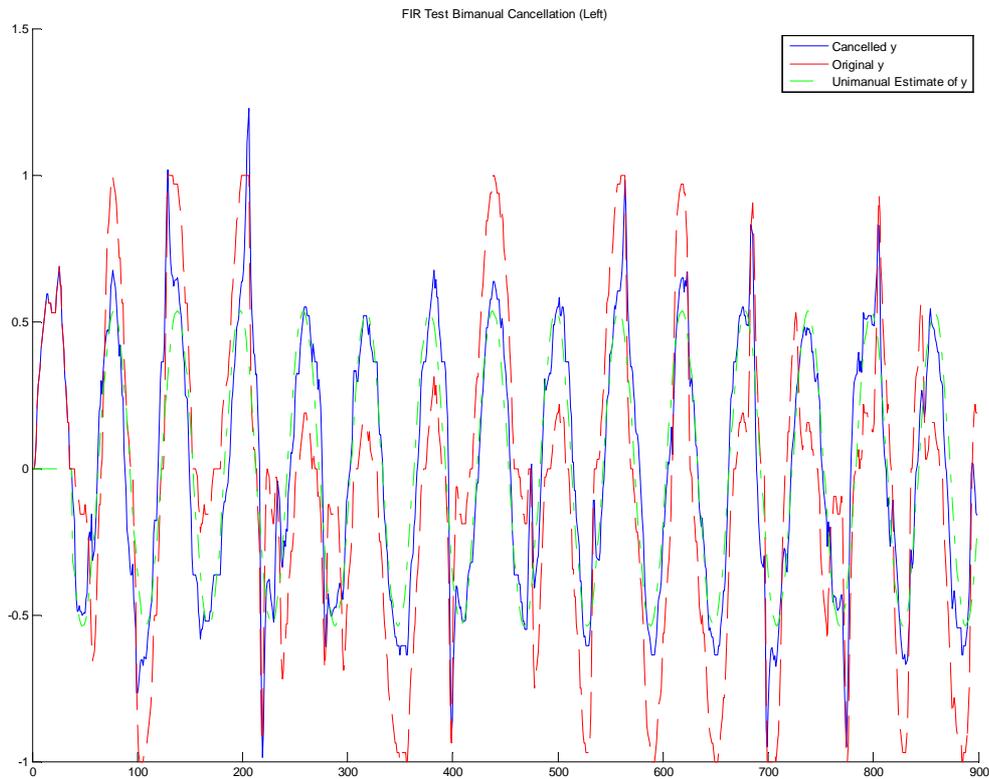


The simulation frequency is 30 Hz, so assuming that a human's response times are not excessive, 20th order FIRs are used throughout. Because this is not a large number and all computations are performed offline, data compression is not required. Thirty seconds of data are generated.

The simulation results are presented below, with each x-axis representing time in samples:

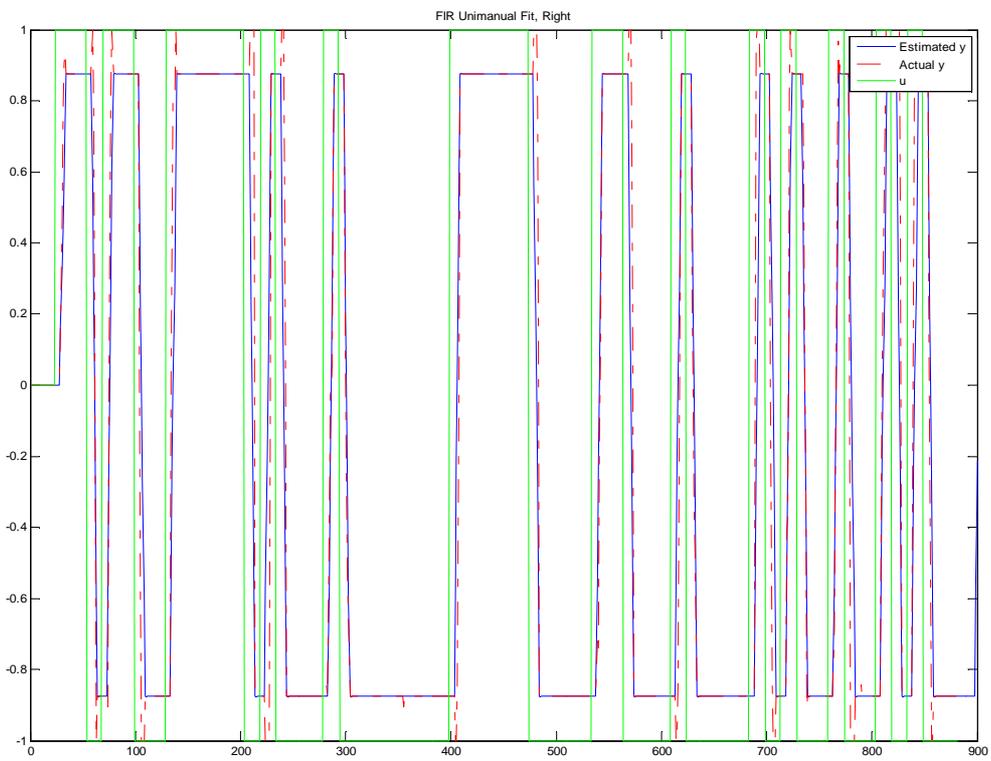
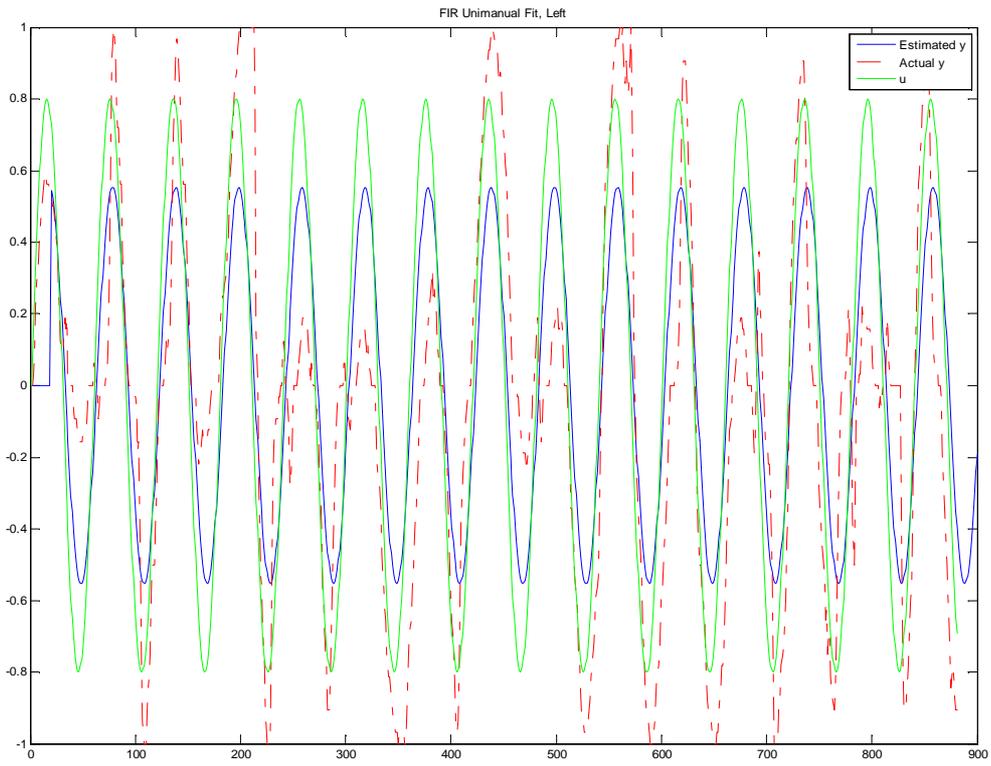


The top plot shows the input, output, and the fitted estimate of output for the left hand. The bottom plot shows the same for the right hand. It is important that these fits are accurate because the error in the bimanual case will be judged relative to these estimates.

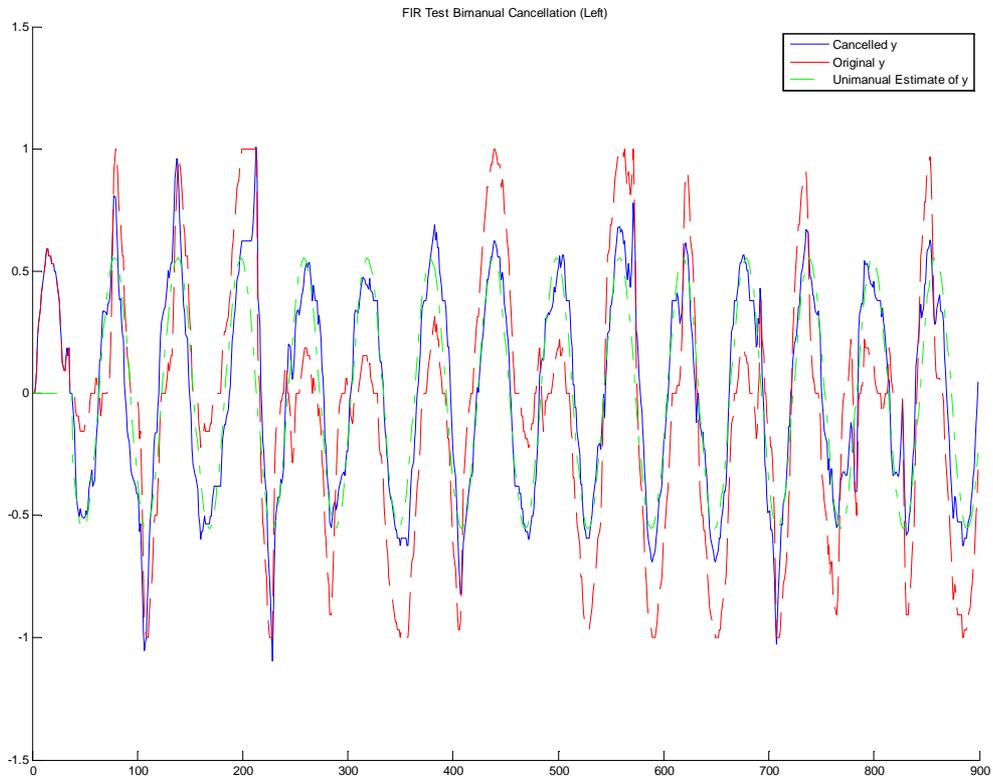


This plot shows the left hand output, with the original, crosstalk-corrupted output in red, the post-cancellation output in blue, and the unimanual estimate in green. Comparing to the unimanual estimate as a reference, the mean squared error of the post-cancellation output is 0.0368, compared to 0.1388 for the original crosstalk-corrupted signal, demonstrating that this technique can create a good fit to the data.

However, as discussed in the Methodology section, it is desirable to be able to identify B_L and B_R using bimanual data, so to verify robustness to the crosstalk, the simulation is tested by estimating B_L and B_R using only data from the bimanual case:

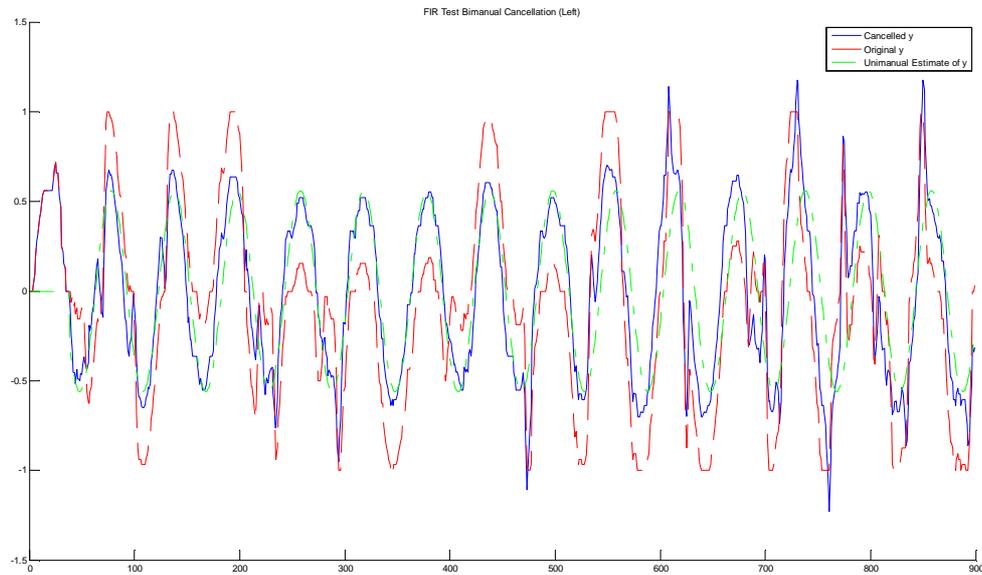


Note that crosstalk is apparent in the actual outputs for the left hand (top plot), but the fit is still essentially identical to the previous fit in the unimanual case.



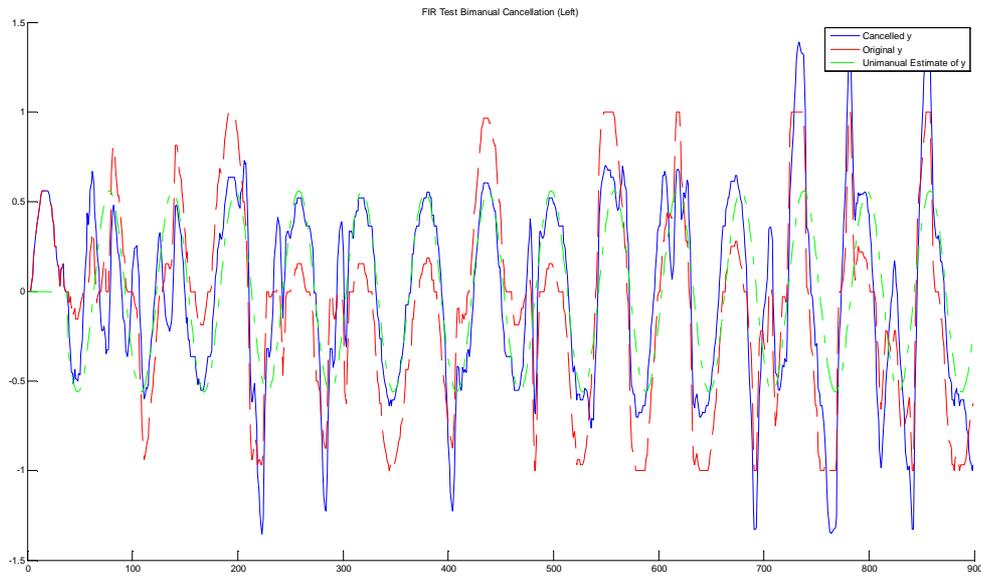
As can be seen, use of uncorrelated noise for the right hand input allows the technique to remain successful while only using bimanual data, with a post-cancellation mean squared error of 0.0315, compared to an un-cancelled mean squared error of 0.1360.

To verify robustness to noise, the estimated FIR parameters are retained but the simulation is re-run using different seeds for the white noise, simulating a new set of data:



The mean squared error here is higher than before, as expected given a new set of data, but the technique is still very effective, reducing mean squared error to 0.0756 from an original 0.1870.

However, if a delay is added to the right hand input, which effectively simulates a changed reaction speed, the ability of this technique to reduce mean squared error is greatly compromised.



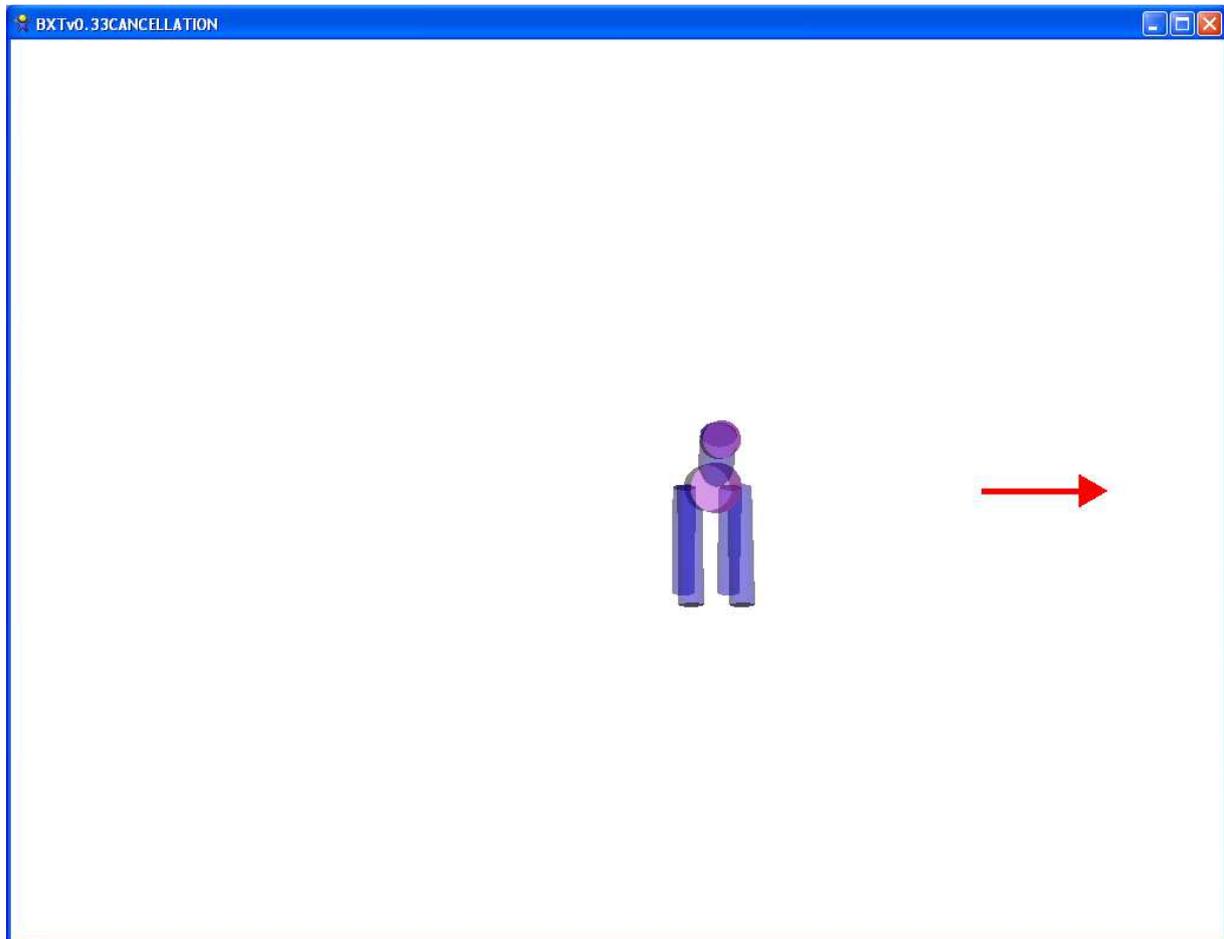
The post-cancellation mean squared error is still a rather high 0.1542, compared to the original 0.1754. The technique is not robust to changing model parameters because if the estimated crosstalk no longer represents the real crosstalk, the compensator is just adding error.

Experimental Results

Data is collected from a human via a computer and dual joystick setup present in the Man Vehicle Lab (Aero/Astro, building 37). A Python application was written to present a human with commands and to collect inputs from the joysticks. A video of the software in action can be seen at the following URL:

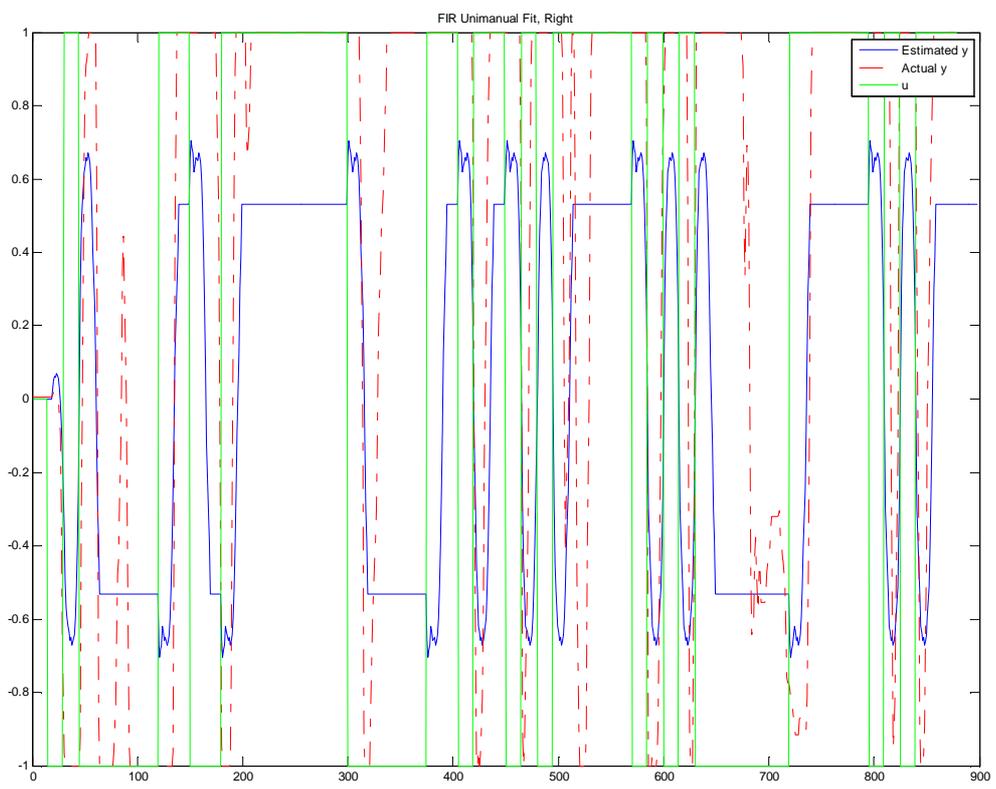
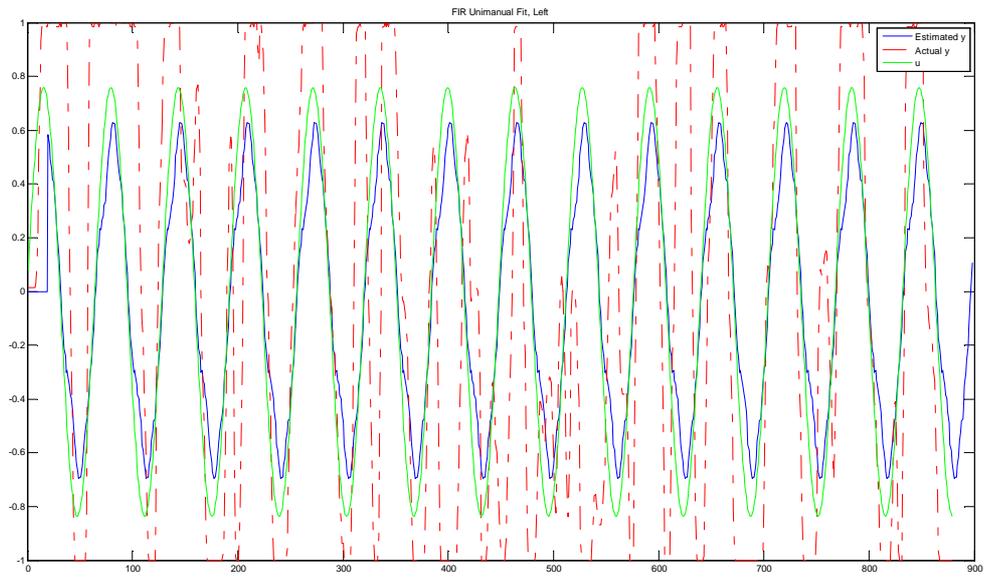
<http://www.youtube.com/watch?v=iw46LiZJ8-o>

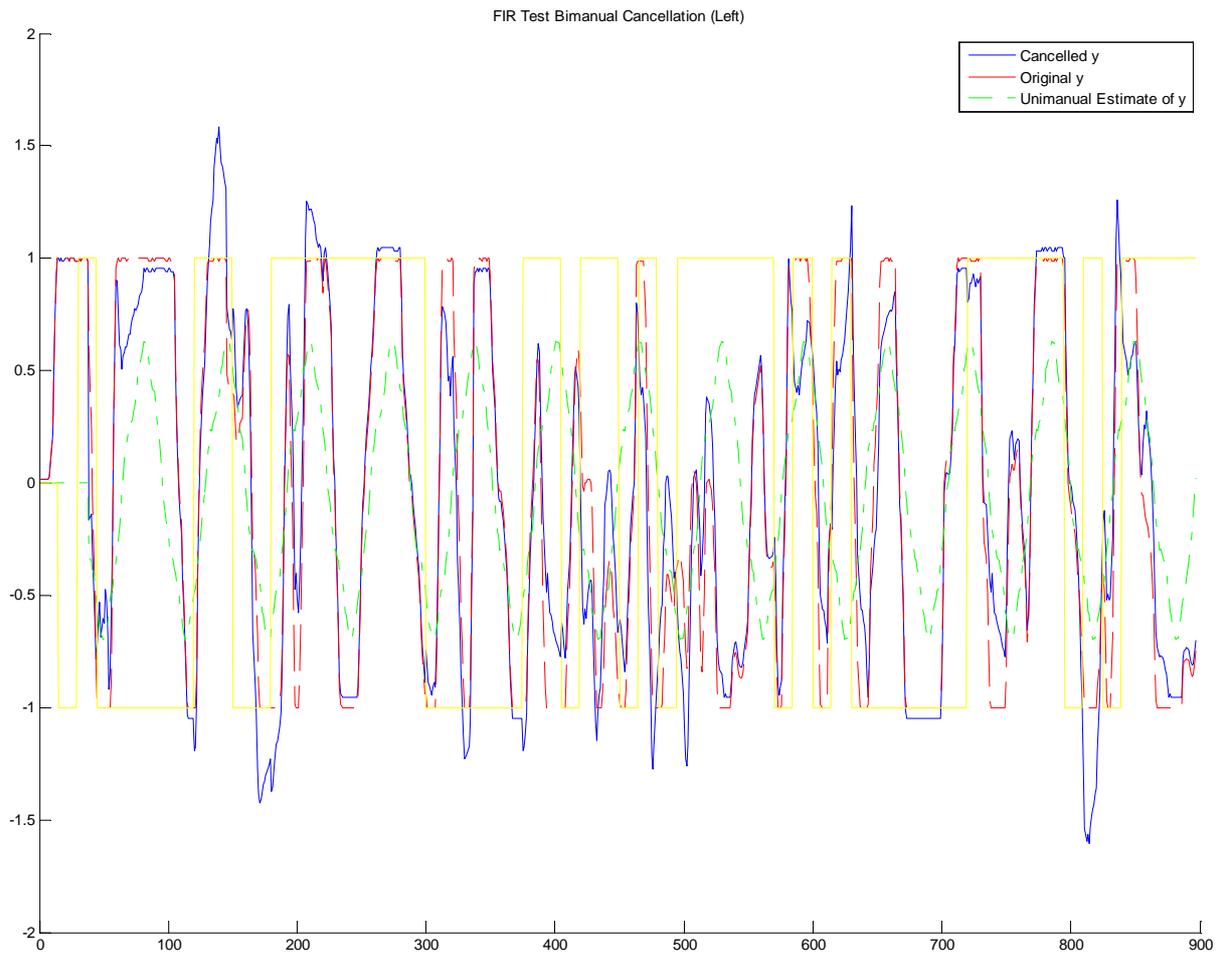
A screenshot can be seen below:



The human's left hand moves a joystick left and right based on the 0.5 Hz sinusoidal position of the purple/blue object on screen. The human's right hand pushes another joystick to the far left or far right in reaction to the arrow signs on the far left and right of the window (only one is visible at a time, depending on a pseudorandom number generator operating at 2 Hz). In this case, the human should move the left joystick slightly to the right and the right joystick all the way to the right.

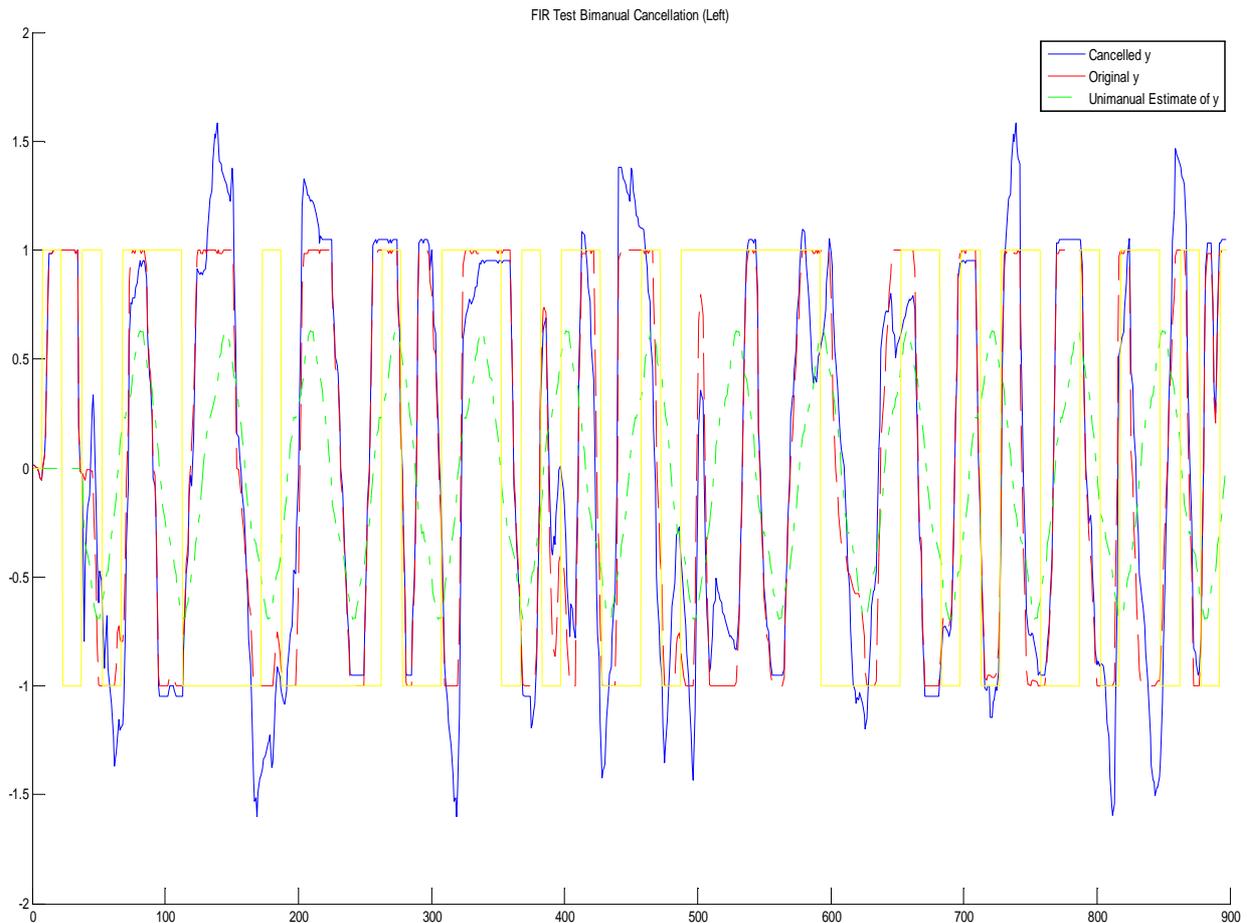
The results using only bimanual information are presented below, using the same parameters and technique as in simulation:





Post-cancellation mean squared error is 0.3971, compared to an original 0.4606, indicating that the technique does not fit to real human data as well. Nevertheless, if the resulting data is looked at carefully, one can distinctly notice periods in time where the cancellation partially compensates for a large crosstalk induced error (note that the right-hand inputs are overlaid on these plots in yellow, to help visually suggest where crosstalk may be expected).

However, using the same estimated FIR parameters with a new data set does not yield the desired reduction in crosstalk:



The post-cancellation mean squared error of 0.6518 is actually greater than the original mean squared error of 0.5989, indicating that despite attempting to stay consistent, the human is not similar enough in its effective FIR parameters, and the previously established model is not fitting enough to benefit in cancellation.

Conclusion

The single-step, bimanual only FIR identification method works well in simulation thanks to the use of binary white noise for the right hand input, but the paradigm of using a static model structure probably is not very practical in designing a reliable bimanual crosstalk cancellation system. It may be possible to improve bimanual crosstalk cancellation performance in a real human using a stochastic or non-linear model structure, or by allowing more time for the human to practice and become consistent, but the improvements may be limited due to the complexity of the underlying neuromotor and visual-cognitive phenomenon. Nevertheless, the techniques developed in this project are generally applicable to a variety of more deterministic multi-input, multi-output systems.

Appendix: MATLAB Code

The code below is modified heavily as required, but exemplifies the general algorithms used throughout this project.

```
%% Bimanual Cross Coupling Cancellation Simulation with Non-Linearities

%% SETUP

close all; clear; clc

Ts = 1/30;

BLdelay = 0.1;
BLspread = 0.1;
BLgain = 1/(4*BLspread/Ts/3);
for i = 1:round((BLdelay+BLspread)/Ts)
    if i > (BLdelay-BLspread)/Ts
        BL(i) = BLgain-BLgain*((i*Ts-BLdelay)/BLspread)^2;
    end
end

BRdelay = 0.3;
BRspread = 0.1;
BRgain = 1/(4*BRspread/Ts/3);
for i = 1:round((BRdelay+BRspread)/Ts)
    if i > (BRdelay-BRspread)/Ts
        BR(i) = BRgain-BRgain*((i*Ts-BRdelay)/BRspread)^2;
    end
end

CRdelay = 0.1;
CRspread = 0.1;
CRgain = .5/(4*CRspread/Ts/3);
for i = 1:round((CRdelay+CRspread)/Ts)
    if i > (CRdelay-CRspread)/Ts
        CR(i) = CRgain-CRgain*((i*Ts-CRdelay)/CRspread)^2;
    end
end

% UNIMANUAL IDENT

ccon = 0; % with CC off (unimanual)

sim('CancellationSim.mdl');

% LEFT HAND
```

```

% load unimanual data
u = ScopeData.signals(1).values;
y = ScopeData.signals(3).values;

% set up parameters
m = 20;
N = length(u)-m;
phi = zeros(m,N);
for i = 1:N
    phi(:,i) = u(i+m-1:-1:i);
end

% least squares fit
P = pinv(phi*phi');
B = zeros(m,1);
for i = 1:N
    B = B + y(i+m-1)*phi(:,i);
end
BLh = P*B;

% test output
yh = zeros(m,1);
for i = 1:N
    yh(i+m-1) = BLh'*phi(:,i);
end
figure
plot(yh, '-')
hold on
plot(y(1:N), 'r-.')
plot(u(1:N), 'g')
title('FIR Unimanual Fit, Left')
legend('Estimated y', 'Actual y', 'u');

% RIGHT HAND

% load unimanual data
u = ScopeData.signals(2).values;
y = ScopeData.signals(4).values;

% set up parameters
N = length(u)-m;
phi = zeros(m,N);
for i = 1:N
    phi(:,i) = u(i+m-1:-1:i);
end

% least squares fit
P = pinv(phi*phi');
B = zeros(m,1);
for i = 1:N

```

```

        B = B + y(i+m-1)*phi(:,i);
    end
    BRh = P*B;

% test output
yh = zeros(m,1);
for i = 1:N
    yh(i+m-1) = BRh'*phi(:,i);
end
figure
plot(yh, '-')
hold on
plot(y(1:N), 'r-.')
plot(u(1:N), 'g')
title('FIR Unimanual Fit, Right')
legend('Estimated y', 'Actual y', 'u')

%% BIMANUAL COUPLING IDENT

ccon = 1; % with CC on (bimanual)

sim('CancellationSim.mdl');

uL = ScopeData.signals(1).values;
uR = ScopeData.signals(2).values;
yL = ScopeData.signals(3).values;
yR = ScopeData.signals(4).values;

% set up parameters
N = length(uL)-m;
phiL = zeros(m,N);
phiR = zeros(m,N);
for i = m:N
    phiL(:,i) = uL(i+m-1:-1:i);
    phiR(:,i) = uR(i+m-1:-1:i);
end

% estimated unimanual output
yLh = zeros(m,1);
yRh = zeros(m,1);
for i = 1:N
    yLh(i+m-1) = BLh'*phiL(:,i);
    yRh(i+m-1) = BRh'*phiR(:,i);
end

% subtract from actual output to get coupling error
yle = yL(1:length(yLh))-yLh;

% remake phi to right size, and using estimated unimanual output
N = length(yLh)-m;
phiL = zeros(m,N);
phiR = zeros(m,N);

```

```

for i = 1:N
    phiL(:,i) = yLh(m+i-1:-1:i);
    phiR(:,i) = yRh(m+i-1:-1:i);
end

% least squares fit to coupling error

P = pinv(phiR*phiR');
B = zeros(m,1);
for i = 1:N
    B = B + yLe(i+m-1)*phiR(:,i);
end
CRh = P*B;

%% TEST CANCELLATION FIT

yReh = zeros(m,1);
for i = 1:N
    yReh(i+m-1) = CRh'*phiR(:,i);
end
figure
hold on
yLc = yL(1:length(yReh))-yReh;
yL = yL(1:length(yReh))';
yLh = yLh(1:length(yReh));
plot(yLc, 'b')
plot(yL, 'r--')
plot(yLh, 'g-')
title('FIR Test Bimanual Cancellation (Left)')
legend('Cancelled y', 'Original y', 'Unimanual Estimate of y');
MSERcancelled = mean((yLc-yLh).^2)
MSERoriginal = mean((yL(1:length(yReh))'-yLh).^2)

```

Works Cited

Aramaki Y., Honda M. and Sadato N. Suppression of the Non-Dominant Motor Cortex During Bimanual Symmetric Finger Movement: A Functional Magnetic Resonance Imaging Study [Journal]. - [s.l.] : Neuroscience, 2006. - 141 (2147-2153).

Banerjee Arpan and Jirsa Victor K. How do neural connectivity and time delays influence bimanual coordination? [Journal]. - [s.l.] : Biological Cybernetics, 2006. - 96 (265-278).

Kennerley Steven [et al.] Callosotomy patients exhibit temporal uncoupling during continuous bimanual movements [Journal]. - [s.l.] : Nature Neuroscience, 2002. - 4 : Vol. 5.

Pollok Bettina [et al.] Intercerebellar Coupling Contributes to Bimanual Coordination [Journal]. - [s.l.] : Journal of Cognitive Neuroscience, 2007. - 4 (704-719) : Vol. 19.

Pontillo Teresa M. Spatial Ability and Handedness as Potential Predictors of Space Teleoperation Performance // MS Thesis. - [s.l.] : Massachusetts Institute of Technology, June 2010.

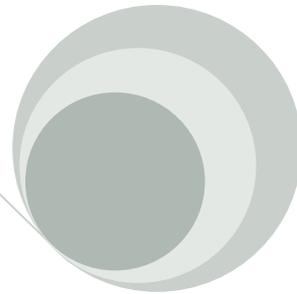
Serrien Deborah J., Cassidy Michael J. and Brown Peter The Importance of the Dominant Hemisphere in the Organization of Bimanual Movements [Journal]. - [s.l.] : Human Brain Mapping, 2003. - 17 (296-305).

Viviani Paolo [et al.] Hemispheric Asymmetries and Bimanual Asynchrony [Journal]. - [s.l.] : Experimental Brain Research, 1998. - 120 (531-536).

Wenderoth Nicole [et al.] Parieto-premotor Areas Mediate Directional Interference During Bimanual Movements [Journal]. - [s.l.] : Cerebral Cortex, 2004. - 10 (1153-1163) : Vol. 14.

Appendix K: Adaptive Visual Tracking

The next 18 pages were written for a class project for 2.165 Robotics, taught by Prof. Slotine in Spring 2011.



Adaptive Visual Tracking of a Floating Target

2.165 – Robotics

This term project report discusses the development of a vision-based controller to replace a human operator during simulated track and capture of a floating target in space using a robotic manipulator arm. Simulated data equivalent to that generated by a machine vision system is fed into an image-space proportional controller. An additional adaptive component is also designed that allows tracking of the target's estimated real-space drift velocities based on only image-space information.

Victor Wang
5/12/2011

Introduction

Currently, teleoperation is used on the International Space Station to control the robotic arm for tracking and capturing of drifting targets. Six degree of freedom velocity control of the end effector is achieved through a three degree of freedom translational joystick and a three degree of freedom rotational joystick, with joint motions calculated automatically through inverse kinematics. Alignment with the target is primarily through a single two-dimensional endpoint camera view, with the goal being to match the endpoint velocity with the target velocity and to approach within a certain spatial envelope in the correct orientation. A Python-based simulator of the robot arm has been developed by the Man Vehicle Lab and is available for research purposes. Three excerpted slides from the telerobotics training program help to describe the task visually and are shown in Figure 1.

- Training Overview
- Flight Rules
- Schedule

➤ Objective:

You will learn how to manipulate a robotic arm in order to perform simulated space teleoperation tasks.

- You will have 90 seconds in which to grapple the HTV.
- Using the techniques that you learned for fly-to and grapple, coordinate translation and rotation such that you maintain proper alignment over the grapple fixture as you approach.




- When in the grapple envelope, pull the trigger to latch onto the HTV.

Alignment Techniques

- Internal Command Frame
- Vernier Rate
- Correcting Alignment Errors:

Error	Technique 1	Technique 2
 Center of crosshairs above center of black circle	Translate down	Pitch down
 Center of crosshairs right of center of black circle	Translate left	Yaw left
 Dot high in white ring	Pitch down	Translate up
 Dot left in white ring	Yaw right	Translate left
 Roll error	Crosshair lines centered in white bar (make sure vertical line on white bar is pointing down)	

Remember: If you input a translation, you may need a rotation to align the target, as the translational and rotational cues are coupled (e.g. pitch down and translate up)

8



Version 1.0 Fall 2010
MVL Robotics Experiment

Figure 1 – Excerpts from the telerobotics training slides showing the controller and screen setup (top), the desired alignment (center), and possible misalignments (bottom) of the target, which consists of a specially marked base and a protruding pin with a small spherical tip. The green lines are visual aids fixed to the camera view. Clearly, this is a complicated task for a human to perform, involving high levels of spatial visualization and manual coordination.

It is proposed that a vision-based adaptive controller be developed that can interface with the simulator software in real time to provide artificial joystick inputs to successfully complete the “track and capture” task while the target drifts arbitrarily. Because the lower-level inverse kinematics controller in the simulator effectively allows the dynamics and kinematics of the arm itself to be neglected, the main components of the artificial control system involve simulated machine vision to feed into the second step of control of input velocities, which requires knowledge of the relationship between real movement and movement of the target on-screen.

Simulated Machine Vision

By extracting six different displacement variables from the end-effector camera view that are each affected differently by the six possible components of relative drift between the target and end effector in real-space, it should be possible to combine these measurements with the inverse image Jacobian to determine target drift velocity in six degrees of freedom. Moreover, if these image-space variables can be manipulated to correspond to image-space and real-space error, it may then be straightforward to design an image-space controller that acts to bring each of these errors to zero.

To this end, the image-space displacements chosen are as illustrated in Figure 2.

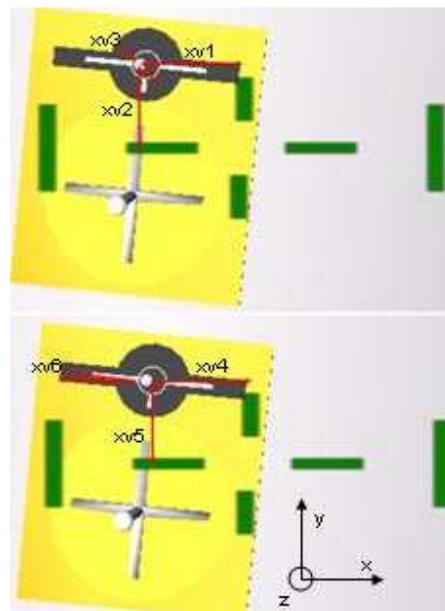


Figure 1 – Image-space displacements, overlaid on two separate screenshots for clarity, with positive coordinate axes indicated at bottom-right. Rotation about the x-axis is referred to as pitch, rotation about the y-axis is referred to as yaw, and rotation about the z axis is referred to as roll. xv1 and xv2 are measured from the target pin tip to the center of the camera view and represent translational error of the target pin tip in the x and y directions in real-space. xv3, the size of the spherical pin tip or any other reliable depth cue, represents the z displacement. With an arbitrary offset depending on the size of this depth cue, the camera properties, and the desired final distance from the target, xv3 can represent z error as well. xv4 and xv5 are measured from the center of the target pin base to the center of the camera view and represent a combined influence of x and y error as well as pitch and yaw error. xv6 is the angle of the long white targeting line in image-space, and represents a direct linear approximation to the real-space roll error.

All of these image-space displacements could be determined by a machine vision system, but that is outside the scope of this project. For the purposes of this project, the image-space displacements are determined by embedding points into the target simulation and projecting their known three-dimensional coordinates to image-space using a function built into the application programming interface. These points are shown in Figure 2, and from the image-space x and y coordinates of these points it is straightforward to determine all the desired image-space displacements.

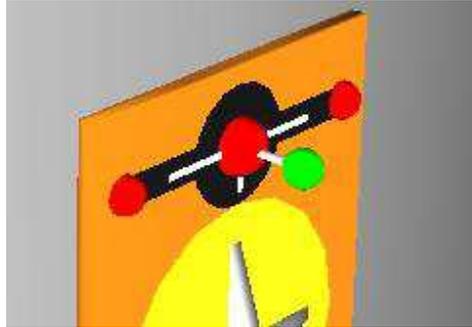


Figure 2 – Points chosen for simulated machine vision, shown as red and green spheres. The image-space projection of the green point at the target pin tip gives x_{v1} and x_{v2} directly. The image-space projection of the large red point at the target pin base gives x_{v4} and x_{v5} directly. The image-space distance between the left and right red points gives x_{v3} , while the angle of the image-space line between these two points gives x_{v6} .

Inverse Image Jacobian

To gain further insight on the design of a suitable controller, and ultimately to derive the image Jacobian, the geometry of the target relative to the camera is analyzed. To simplify the nonlinear relationships between axes, this analysis is performed assuming small errors in alignment, so that the geometry can be considered in each spatial plane independently.

With this assumption, the easiest relationship to determine is that the roll in image-space is a direct representation of roll in real-space:

$$x_{v6} = \varphi \quad (1)$$

Also with the assumption that the spherical target pin tip or other depth cue of diameter d is centered and aligned with the camera with field of view FOV , z and x_{v3} (x_{v3} as a proportion of the total camera image size) can be expressed as

$$z = \frac{d}{x_{v3} \tan(FOV)}$$

which can be written in terms of a lumped unknown parameter K_z :

$$z = \frac{K_z}{x_{v3}} \quad (2)$$

To help determine the other relationships, an illustration of the x - z plane is shown in Figure 4. A similar illustration can be drawn for the y - z plane.

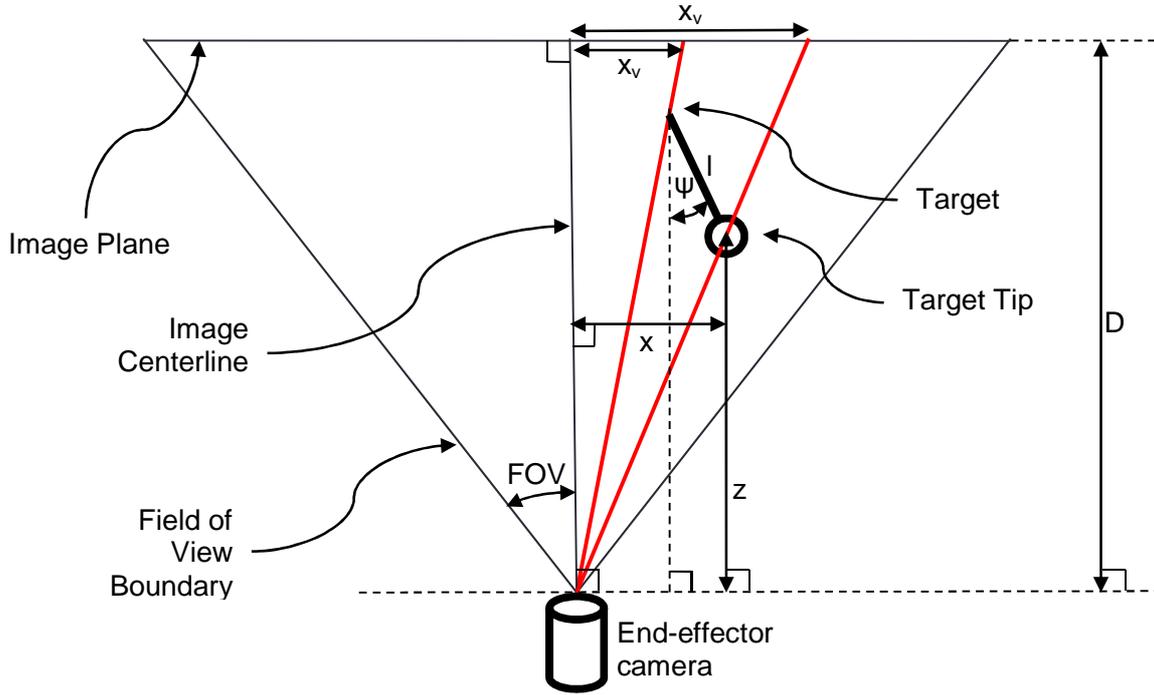


Figure 3 – Illustration of planarized geometry in the x-z plane. x , z and ψ are real-space displacement errors of the target relative to the end effector, x_{v2} and x_{v5} are the image-space displacement errors as described before, measured relative to the size of the image plane. The relationship between these displacements depends on length of the pin (l), the field of view of the camera (FOV), and the arbitrary depth of the image plane (D), which will later be treated as unknowns from the point of view of an image-space controller.

From this representation, it can be seen that when the image-space displacements are measured relative to the size of the image plane (i.e. as a proportion of the camera image width),

$$\frac{x}{z \tan(\text{FOV})} = x_{v2}$$

which can be combined with Equation 2 and written in terms of an unknown parameter K_x :

$$x = K_x \frac{x_{v1}}{x_{v3}} \quad (3)$$

Similarly in the y-z plane,

$$y = K_y \frac{x_{v2}}{x_{v3}} \quad (4)$$

The relationship between x , z , ψ and x_{v5} is as follows:

$$\frac{x - l \sin \psi}{z + l \cos \psi} = \frac{x_{v5}}{D}$$

which for small ψ can be linearized to

$$\frac{x - l\psi}{z + l} = \frac{x_{v5}}{D}$$

and then solved for ψ , combined with Equations 2 and 3, and written in terms of unknowns:

$$\psi = K_{\psi 1} \frac{x_{v5}}{x_{v3}} + K_{\psi 2} x_{v5} + K_{\psi 3} \frac{x_{v1}}{x_{v3}} \quad (5)$$

Similarly in the y-z plane,

$$\theta = K_{\theta 1} \frac{x_{v4}}{x_{v3}} + K_{\theta 2} x_{v4} + K_{\theta 3} \frac{x_{v2}}{x_{v3}} \quad (6)$$

Taking derivatives of Equations 1 through 6 gives the inverse image Jacobian, relating the image-space velocities into real-space velocities:

$$J_I^{-1} = \begin{bmatrix} \frac{K_x}{x_{v3}} & 0 & -K_x \frac{x_{v1}}{x_{v3}^2} & 0 & 0 & 0 \\ 0 & \frac{K_y}{x_{v3}} & -K_y \frac{x_{v2}}{x_{v3}^2} & 0 & 0 & 0 \\ 0 & 0 & -K_z/x_{v3}^2 & 0 & 0 & 0 \\ 0 & \frac{K_{\theta 3}}{x_{v3}} & -K_{\theta 1} \frac{x_{v4}}{x_{v3}^2} - K_{\theta 3} \frac{x_{v2}}{x_{v3}^2} & \frac{K_{\theta 1}}{x_{v3}} + K_{\theta 2} & 0 & 0 \\ \frac{K_{\psi 3}}{x_{v3}} & 0 & -K_{\psi 1} \frac{x_{v5}}{x_{v3}^2} - K_{\psi 3} \frac{x_{v1}}{x_{v3}^2} & 0 & \frac{K_{\psi 1}}{x_{v3}} + K_{\psi 2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Defining

$$x_v = (x_{v1} \ x_{v2} \ x_{v3} \ x_{v4} \ x_{v5} \ x_{v6})^T$$

and multiplying $J_I x_v$, then factoring into measurable (Y) and unknown (a) parts, a real-space target velocity estimate can be written as

$$\hat{v}_t = Y \hat{a} = \begin{bmatrix} A & & & & & \\ & B & & & & \\ & & -\frac{\dot{x}_{v3}}{x_{v3}^2} & & & \\ & & & C & \dot{x}_{v4} & B \\ & & & & D & \dot{x}_{v5} & A \\ & & & & & & \dot{x}_{v6} \end{bmatrix} \begin{pmatrix} K_x \\ K_y \\ K_z \\ K_{\theta 1} \\ K_{\theta 2} \\ K_{\theta 3} \\ K_{\psi 1} \\ K_{\psi 2} \\ K_{\psi 3} \\ 1 \end{pmatrix}$$

where

$$A = \frac{\dot{x}_{v1}}{x_{v3}} - \frac{x_{v1}\dot{x}_{v3}}{x_{v3}^2}$$

$$B = \frac{\dot{x}_{v2}}{x_{v3}} - \frac{x_{v2}\dot{x}_{v3}}{x_{v3}^2}$$

$$C = \frac{\dot{x}_{v4}}{x_{v3}} - \frac{x_{v4}\dot{x}_{v3}}{x_{v3}^2}$$

$$D = \frac{\dot{x}_{v5}}{x_{v3}} - \frac{x_{v5}\dot{x}_{v3}}{x_{v3}^2}$$

Proportional Image-Space Controller

An effective heuristic is used in training humans to operate the arm. This heuristic essentially involves correcting for x and y using the values of x_{v1} - x_{v4} and x_{v2} - x_{v5} , correcting for pitch and yaw using x_{v1} and x_{v2} , correcting for roll using x_{v6} , and constantly approaching along z until the correct distance is reached. This suggests that despite the coupled interactions, it should be possible to pull all the errors to zero by choosing one main real-space influencing axis for each image-space error, and apply a “spring force” to each of these real-space axes based on the corresponding image-space error.

Although the image Jacobian was more straightforward to derive with the definitions of x_v as described so far, due to the practical use of the difference between x_{v1} - x_{v4} and x_{v2} - x_{v5} , and also the inverse nature of x_{v3} related to z, these relationships were substituted in such that in the implementation phase, x_{v4} and x_{v5} are defined relative to x_{v1} and x_{v2} , and x_{v3} is defined as the reciprocal of the size-based depth cue. Updated versions of J_i^{-1} , Y , and a are shown in Appendix A.

A block diagram of the proposed control system is shown in Figure 4.

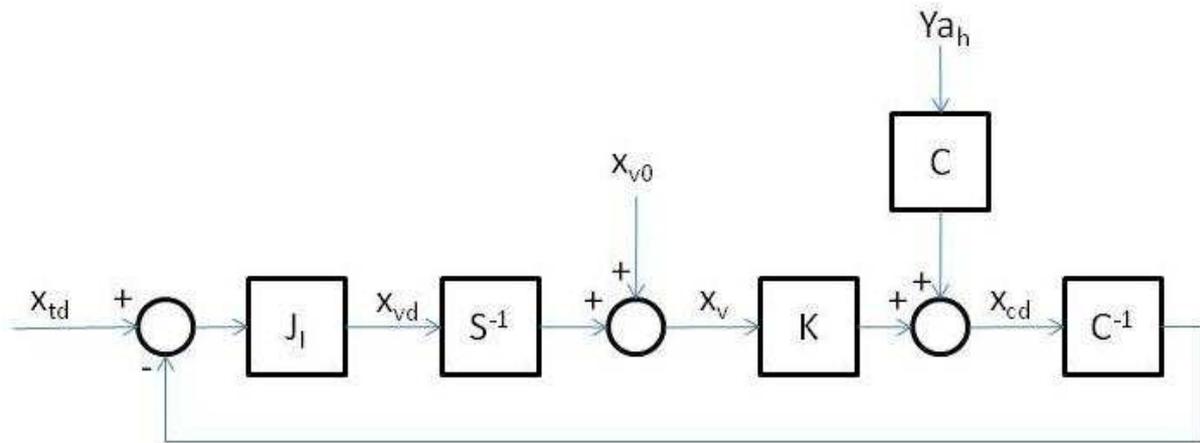


Figure 4 – A block diagram of the proposed control system. x_{td} is the vector of the rate of change of the real-space errors, essentially the drift velocity of the target relative to the end-effector. J_1 is the image Jacobian relating real-space velocity to image-space velocity. s^{-1} represents an integration from velocity to displacement. x_{v0} is the initial value of the image space displacements, x_v . K is the “spring constant” gain matrix converting x_v to a set of real-space end-effector velocities. C^{-1} is the inverse coupling matrix which transforms motion at the target to rigidly coupled motion at the end-effector. Y_{a_h} will be discussed later as part of the adaptive controller.

To a linear approximation, the coupling matrix C that would rigidly couple the target with the end-effector is

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & z & 0 \\ 0 & 1 & 0 & -z & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

with the z terms reflecting translation at the end effector due to rotation of the target. These terms can either be taken from x_{v3} scaled by an appropriate calibration constant, or simply the real value of z , supposing that information from a laser rangefinder is available.

The proposed control law

$$\dot{x}_c = Kx_v$$

transforms image-space errors into end-effector motion through a gain matrix K . This control action affects the real-space target motion relative to the end-effector through C^{-1} , and is fed back to the original target drift velocity.

An energy-like positive scalar term can be expressed as

$$V = \frac{1}{2} x_v^T K x_v$$

which has derivative

$$\dot{V} = x_v^T K \dot{x}_v = x_v^T K J_I (\dot{x}_t - C^{-1} \dot{x}_c) = x_v^T K J_I \dot{x}_t - x_v^T K J_I C^{-1} K x_v$$

which is negative if $x_v^T K J_I \dot{x}_t < 0$ and $J_I C^{-1} > 0$. In other words, Barbalat's Lemma suggests stability is possible if K is chosen such the positive image-space errors tend to result in a negative rate of change of those errors (first condition), and that the image-space error cues are chosen such that increasing image-space error reflects increasing real-space error, and vice versa (second condition).

Referring to Figure 4, the image-space error can be expressed as follows:

$$x_v = J_I (\dot{x}_t - C^{-1} K x_v) s^{-1} + x_{v0}$$

Solving for x_v ,

$$x_v = \frac{J_I \dot{x}_t s^{-1} + x_{v0}}{1 + J_I C^{-1} K s^{-1}}$$

To find steady-state image-space error, we note the Final Value Theorem:

$$e(\infty) = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} s E(s)$$

where $E(s)$ is the Laplace transform of $e(t)$.

Since the input \dot{x}_t is constant (the target drift velocity in real-space does not change), and x_{v0} is of course constant, they can therefore be represented as a step ($1/s$) in the Laplace domain, such that the steady state value of x_v can be written as

$$x_v(\infty) = \lim_{s \rightarrow 0} s \frac{J_I \dot{x}_t s^{-1} + x_{v0}}{1 + J_I C^{-1} K s^{-1}} \frac{1}{s}$$

which can be simplified as follows

$$x_v(\infty) = \lim_{s \rightarrow 0} \frac{J_I \dot{x}_t + x_{v0} s}{s + J_I C^{-1} K} = \frac{J_I \dot{x}_t}{J_I C^{-1} K}$$

to

$$x_v(\infty) = K^{-1} C \dot{x}_t$$

The steady state value of the image-space error vector is therefore a function of the target drift velocity in real-space, scaled by the coupling matrix C (such that being far away will tend to increase certain steady state errors) and the inverse of the gain matrix K (such that large gains can reduce steady state error). Interestingly, the image Jacobian J_I cancels out as it increases the response of the image-space error to real-space drift velocity, but in turn also increases the response of the control action.

Adaptive Augmented Controller

The new control law

$$\dot{x}_c = Kx_v + CY\hat{a}$$

contains an additional term to allow real-space target drift velocity compensation. Note that when using the terms in \dot{x}_v to calculate Y, it is important to remove the influence of the controller action from what is being measured, such that

$$\dot{x}_v = \dot{x}_{v,measured} - J_I C^{-1} \dot{x}_c$$

where $\dot{x}_{v,measured}$ is the numerically differentiated version of the measured x_v .

Referring to the block diagram in Figure X, the image-space error can be expressed as follows:

$$x_v = J_I(\dot{x}_t - C^{-1}(Kx_v + CY\hat{a}))s^{-1} + x_{v0}$$

Solving for x_v ,

$$x_v = \frac{J_I \dot{x}_t s^{-1} - J_I Y \hat{a} s^{-1} + x_{v0}}{1 + J_I C^{-1} K s^{-1}}$$

Expressing steady-state image-space error as before,

$$x_v(\infty) = \lim_{s \rightarrow 0} \frac{J_I \dot{x}_t - J_I Y \hat{a} + x_{v0} s}{s + J_I C^{-1} K}$$

which reduces to

$$x_v(\infty) = K^{-1} C (\dot{x}_t - Y \hat{a})$$

giving zero steady-state image-space error if the following condition is met:

$$Y \hat{a} = \dot{x}_t$$

To help meet this criterion, the learning law can be designed by first expressing a new energy-like Barbalat variable incorporating the estimation error $\tilde{a} = \hat{a} - a$ and a learning parameter P:

$$V = \frac{1}{2} x_v^T K x_v + \frac{1}{2} \tilde{a}^T P^{-1} \tilde{a}$$

which has derivative

$$\dot{V} = x_v^T K \dot{x}_v + \tilde{a}^T P^{-1} \dot{\tilde{a}} = -x_v^T K J_I Y \tilde{a} - x_v^T K J_I C^{-1} K x_v + \tilde{a}^T P^{-1} \dot{\tilde{a}}$$

Selecting the learning law

$$\dot{\hat{a}} = P Y^T J_I^T K^T x_v$$

and substituting back into \dot{V} to cancel the term containing estimation error \tilde{a} gives

$$\dot{V} = -x_v^T K J_I C^{-1} K x_v$$

which is negative and therefore stable under one of the same conditions as the proportional image-space controller, namely that K is chosen such the positive image-space errors tend to result in a negative rate of change of those errors. Interestingly, adaptation removes the condition that the image-space error cues must be chosen such that increasing image-space error reflects increasing real-space error, and vice versa.

Implementation and Performance

The above control strategies were implemented and tested in Python using the telerobotics simulator described in the Introduction. An excerpt of the relevant control code is listed in Appendix B. Internet links to demonstrative videos can be found in Appendix C, along with text descriptions of those videos.

Suitable gains for the proportional controller were chosen as

$$K = \begin{bmatrix} 100G & 0 & 0 & 0 & 0 & 0 \\ 0 & 100G & 0 & 0 & 0 & 0 \\ 0 & 0 & 100G & 0 & 0 & 0 \\ 0 & 0 & 0 & 10G & 0 & 0 \\ 0 & 0 & 0 & 0 & 10G & 0 \\ 0 & 0 & 0 & 0 & 0 & 10G \end{bmatrix}$$

where G is the gain used to multiply the real joystick inputs. During the simulation, this can be toggled between two standard modes: a high gain for gross movements and a smaller, “vernier” gain for fine, slow movements such as those used in a track and capture task. Saturation limits, also dependent on the gain mode, are used to simulate the same limits imposed using physical joysticks. Initial values for \hat{a} are arbitrarily set to 1, and P is tuned to be 2 times the identity matrix.

As seen in Videos 1 and 2, the proportional image-space controller works very well in both vernier and fast gain settings, although there is a noticeable gain-dependent steady state jitter for the high gain case. For the vernier case, steady state image-space error x_v has some noise, but is approximately:

$$[0.014 \ 0.023 \ -0.019 \ -0.007 \ 0.007 \ 0.007]^T$$

Performance is much better than possible with a human. Interestingly, adding the adaptive $Y\hat{a}$ component to the controller does not measurably decrease steady state error, even in vernier rate mode, for which there is a small but visually noticeable amount of steady state error due to the lower gain. This could be because of estimation errors in the adaptive component as seen in Video 3.

The adaptive $Y\hat{a}$ component is isolated in Video 3 to show its ability to track the velocity of the target without regard for eliminating positional errors. As can be seen, the system is able to seemingly couple with the target in image-space, significantly reducing the apparent rate of drift. However, it is not perfect, likely due to linearization errors and also due to the visual difficulty of

separating rotation and translation cues. In fact, it can be seen in the other camera views that the end effector is not translating as much as it should in real-space, relying on rotation more, since the two real-space movements produce very similar effects in image-space.

Conclusion

A block diagram analysis combined with use of Barbalat's Lemma suggests that a stable image-space proportional controller can be designed given an appropriate set of image-space displacement variables and also an appropriate gain matrix. Such a controller was implemented, and its empirically observed effectiveness demonstrates that these conditions are achievable. An additional control component is designed using Barbalat and a geometric analysis to adaptively form an internal representation of the target's real-space velocities and match the end-effector motion to the target motion, which by the Final Value Theorem, should allow zero steady-state error. However it appears that the addition of the adaptive component does not appreciably improve performance. This is likely due to linearization errors in the adaptive controller combined with difficulty in separating real-space pitch and yaw from real-space x and y translation using only image-space information. Additionally, the performance of the proportional image-space controller is already very good when the gain matrix is well-tuned.

Appendix B: Excerpted Python Code Listing

```
#####
#
# Automated tracking controllers
#
#####

# function to saturate an array to min/max values and return saturated array
def limit(numbers, min, max):
    y = []
    for x in numbers:
        if x < min:
            x = min
        elif x > max:
            x = max
        y.append(x)

    return y

# initialize adaptive controller matrices whenever
# the adaptive controller type is toggled on
def InitAdaptiveController():
    global xv, xvd, xcd, Y, ah, K, C, P, xvbuff, xvbuffsize, xvbuffi, Jinv, J

    # these will be updated, just making a spot in memory for them
    xv = numpy.mat(numpy.zeros([6,1]))
    xvd = numpy.mat(numpy.zeros([6,1]))
    xcd = numpy.mat(numpy.zeros([6,1]))
    Y = numpy.mat(numpy.zeros([6,10]))
    ah = numpy.mat(numpy.ones([10,1]))
    K = numpy.mat(numpy.zeros([6,6]))
    C = numpy.mat(numpy.zeros([6,6]))
    Jinv = numpy.mat(numpy.zeros([6,6]))
    J = numpy.mat(numpy.zeros([6,6]))

    # initialize xvd smoothing buffer
    xvbuffsize = 10
    xvbuff = numpy.mat(numpy.zeros([6,xvbuffsize]))
    xvbuffi = 0

    # this is a constant learning gain matrix
    P = numpy.mat(numpy.eye(10))*2

# when any adaptive controller is selected, this function is
# called every main loop cycle to provide inputs instead of
# getting them from keyboard or joystick. because of this,
# note that translational inputs should have 10x gain
def GetAutoInputs():
    global xv, xvd, xcd, Y, ah, K, C, xvbuff, xvbuffi, Jinv, J

    # actual vt, for testing purposes !!! this is not in the right frame of reference
    vt = (numpy.mat(targetTip.getVelocity()+targetTip.getAngularVelocity())).T

    # reset commands
    pitch = yaw = roll = x = y = z = 0
    zstop = 1 # arbitrary representation of z distance to stop at

    screen = RWSS_WindowList[1] # use the middle screen

    # get measures of x,y,pitch,yaw
    xv1, xv2, z = screen.worldToScreen(targetTip.getPosition(viz.ABS_GLOBAL))
    xv4, xv5, z = screen.worldToScreen(targetBase.getPosition(viz.ABS_GLOBAL))
    # normalize xv1,2,4,5 to +/- 1, nominal 0, so now they represent error as well
    xv1 = 2*(xv1-0.5)
    xv2 = 2*(xv2-0.5)
    xv4 = xv1-2*(xv4-0.5) # tip relative to base, gives correct sign of compensatory pitch
    xv5 = xv2-2*(xv5-0.5) # tip relative to base, gives correct sign of compensatory yaw
```

```

# get measures of z & roll
lx, ly, lz = screen.worldToScreen(targetLeft.getPosition(viz.ABS_GLOBAL))
rx, ry, rz = screen.worldToScreen(targetRight.getPosition(viz.ABS_GLOBAL))
# convert to xv3 and xv6
xv3 = rx - lx # 0 to 1, measures proportion of screen taken up
xv3 = -1/xv3 # reciprocal so xv3d is consistent with direction of change of error, with
              negative for correct direction
xv3 += zstop # apply arbitrary xv3 offset to stop at some distance
xv6 = math.atan((ry-ly)/(rx-lx))/math.pi # basically the linearized roll error

print autoControlMode, xv1, xv2, xv3, xv4, xv5, xv6

if autoControlMode == 3: # bang-bang image space controller
    if xv1 > 0:
        x = 10*JointRateGain
    elif xv1 < 0:
        x = -10*JointRateGain
    if xv2 > 0:
        y = 10*JointRateGain
    elif xv2 < 0:
        y = -10*JointRateGain
    if xv3 > 0:
        z = 10*JointRateGain
    elif xv3 < 0:
        z = -10*JointRateGain

    if xv4 > 0:
        yaw = 1*JointRateGain
    elif xv4 < 0:
        yaw = -1*JointRateGain
    if xv5 > 0:
        pitch = 1*JointRateGain
    elif xv5 < 0:
        pitch = -1*JointRateGain
    if xv6 > 0:
        roll = 1*JointRateGain
    elif xv6 < 0:
        roll = -1*JointRateGain

elif autoControlMode == 2: # proportional image space controller
    x = 100*xv1*JointRateGain
    y = 100*xv2*JointRateGain
    z = 100*xv3*JointRateGain
    pitch = 10*xv5*JointRateGain
    yaw = 10*xv4*JointRateGain
    roll = 10*xv6*JointRateGain
    [pitch,yaw,roll]=limit([pitch,yaw,roll], -1*JointRateGain, 1*JointRateGain)
    [x,y,z]=limit([x,y,z], -10*JointRateGain, 10*JointRateGain)

elif autoControlMode == 1: # adaptive controller
    framerate = 30

    # calculate xvd (time derivative of xv), and get new xv
    xvd = (numpy.mat([xv1,xv2,xv3,xv4,xv5,xv6]).T -
           xvbuff[:,(xvbuffi+1)%xvbuffsize])*framerate/xvbuffsize
    xv1d = xvd[0]; xv2d = xvd[1]; xv3d = xvd[2]; xv4d = xvd[3]; xv5d = xvd[4];
    xv6d = xvd[5];
    xv = numpy.mat([xv1,xv2,xv3,xv4,xv5,xv6]).T

    # add xv to the buffer (to get a smoother xvd)
    xvbuff[:,xvbuffi] = xv
    xvbuffi += 1
    if xvbuffi == xvbuffsize:
        xvbuffi = 0

    # estimate of distance to plug into C matrix. Either actual z (pretend like this
    # is a laser rangefinder reading, for example, or calculated Kz/xv6 where Kz is
    # the distance from the target measured when xv6 = 1

```

```

#
zh = z
zh = 1.2*xv6

# calculate C matrix, the target/end-effector coupling matrix
C[0] = [1, 0, 0, 0, -zh, 0]
C[1] = [0, 1, 0, zh, 0, 0]
C[2] = [0, 0, 1, 0, 0, 0]
C[3] = [0, 0, 0, 1, 0, 0]
C[4] = [0, 0, 0, 0, 1, 0]
C[5] = [0, 0, 0, 0, 0, 1]

# This is empiricially the actual C matrix
C[0] = [19, 0, 0, 0, -zh/4.7, 0]
C[1] = [0, 19, 0, zh/4.7, 0, 0]
C[2] = [0, 0, -19, 0, 0, 0]
C[3] = [0, 0, 0, .014, 0, 0]
C[4] = [0, 0, 0, 0, -0.014, 0]
C[5] = [0, 0, 0, 0, 0, 0.014]

# calculate image Jacobian
Jinv[0] = [float(ah[0])*xv3, 0, float(ah[0])*xv1, 0, 0, 0]
Jinv[1] = [0, float(ah[1])*xv3, float(ah[1])*xv2, 0, 0, 0]
Jinv[2] = [0, 0, ah[2], 0, 0, 0]
Jinv[3] = [float(ah[3])*xv3+float(ah[4]), float(ah[5])*xv3, float(ah[3])*(xv1-
xv4)+float(ah[5])*xv2, -float(ah[3])*xv3-float(ah[4]), 0, 0]
Jinv[4] = [float(ah[8])*xv3, float(ah[6])*xv3+float(ah[7]), float(ah[6])*(xv2-
xv5)+float(ah[8])*xv1, 0, -float(ah[6])*xv3-float(ah[7]), 0]
Jinv[5] = [0, 0, 0, 0, 0, 1]
J = numpy.linalg.pinv(Jinv)

# remove controller influence from xvd to get just xvd error
xvd = xvd-J*numpy.linalg.pinv(C)*xcd

# calculate Y matrix of knowns
Y1 = xv3*xv1d+xv1*xv3d
Y2 = xv3*xv2d+xv2*xv3d
Y3 = xv3*xv1d+xv1*xv3d-xv4*xv3d-xv3*xv4d
Y4 = xv3*xv2d-xv5*xv3d+xv2*xv3d-xv3*xv5d
Y[0,0] = Y1
Y[1,1] = Y2
Y[2,2] = xv3d
Y[3] = [0,0,0,Y3,(xv1d-xv4d),Y2,0,0,0,0]
Y[4] = [0,0,0,0,0,0,Y4,(xv2d-xv5d),Y1,0]
Y[5,9] = xv6d

# calculate gain matrix K based on current JointRateGain
K[0,0] = JointRateGain*100
K[1,1] = JointRateGain*100
K[2,2] = JointRateGain*100
K[3,4] = JointRateGain*10
K[4,3] = JointRateGain*10
K[5,5] = JointRateGain*10

# update parameter estimates
ahd = -P*Y.T*J.T*K.T*xv
ah += ahd/framerate
print (Y*ah).T

# calculate control output
xcd = K*xv
[x, y, z, pitch, yaw, roll] = xcd # get outputs in standard variables

# output saturation to mimick joysticks
[pitch,yaw,roll]=limit([pitch,yaw,roll], -1*JointRateGain, 1*JointRateGain)
[x,y,z]=limit([x,y,z], -10*JointRateGain, 10*JointRateGain)

return [pitch,yaw,roll,x,y,z]

```

Appendix C: Demonstrative Videos

Video 1: Proportional Image-Space Controller, Vernier Rate

For the first several seconds, the target is left to drift in 6 degrees of freedom. Then, the proportional image-space controller is engaged, and the end-effector begins to track the target, with control velocities saturated to low values (vernier rate), similar to the rate used aboard the actual Space Station.

<http://www.youtube.com/watch?v=U3rZ8NGcwgl>

Video 2: Proportional Image-Space Controller, Fast Rate

For the first several seconds, the target is left to drift in 6 degrees of freedom. Then, the proportional image-space controller is engaged, and the end-effector begins to track the target, with control velocities saturated to values much higher than the limits used in real life for this kind of teleoperation task.

<http://www.youtube.com/watch?v=wT6E9UBq5n4>

Video 3: Adaptive Visual Anchoring

The "adaptive component" of the adaptive image-space controller is used here. In other words, the K^*x_v component of control velocity is removed, leaving only the $C^*Y^*a_h$ component which in the ideal case rigidly couples the end-effector to the target. For the first several seconds in this video, the target is left to drift in 6 degrees of freedom. Then the "adaptive anchor" is engaged. After a period of adaptation (during which the view wobbles), the controller's internal estimate of the target's drift velocity allows the end-effector to (almost) rigidly couple with the target, such that the center camera view becomes close to stationary. After another several seconds, the adaptive anchor is disengaged and the target is seen to drift off.

<http://www.youtube.com/watch?v=2hcc2c3w0dM>