DESIGN AND IMPLEMENTATION OF PATIENT SPECIFIC VIRTUAL REALITY SYSTEMS FOR MOTOR REHABILITATION

A Dissertation Presented

by

Mark Langdon Sivak

to

The College of Engineering

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in the field of

Interdisciplinary Engineering

Northeastern University
Boston, Massachusetts

August 2012
ABSTRACT

There is a need for low-cost, in-home rehabilitation systems with increasing numbers of strokes per year. Therefore a virtual reality system for home use in rehabilitation of the hand and wrist that included hardware, middleware, and software was created and tested with healthy subjects. The Angle Tracking and Location At-home System (ATLAS) was created with potentiometer bend sensors to track finger angle and inertial measurement units (IMUs) to track hand orientation and position. To facilitate the communication between the ATLAS and a computer the Device Agnostic Virtual Environment (DAVE) was created as the middleware solution. Patient specific virtual environments have been created for the ATLAS with the goal of increasing subject motivation and outcomes. The ATLAS was found to fit the requirements for a low-cost, in home system for motor rehabilitation.
# Table of Contents

Abstract ................................................................................................................................. i  
List of Figures ....................................................................................................................... iv  
List of Tables ....................................................................................................................... viii  
List of Equations .................................................................................................................. viii  
List of Code Fragments ...................................................................................................... viii  
Acknowledgements ........................................................................................................... ix  
Dedication ............................................................................................................................ ix  

CHAPTER 1: Introduction ..................................................................................................... 1  
1.1 Virtual Reality Systems ............................................................................................... 2  
1.2 Significance ................................................................................................................... 3  
1.3 Definitions and Nomenclature .................................................................................... 4  
1.4 Dissertation Outline .................................................................................................... 7  
1.5 Contributions to the Field .......................................................................................... 7  

CHAPTER 2: Physical Therapy and Rehabilitation Background Information .................... 9  
2.1 Introduction .................................................................................................................. 10  
2.2 Stroke ......................................................................................................................... 10  
2.3 Human Information Processing Models ...................................................................... 11  
2.4 Motor Control ............................................................................................................. 14  
2.5 Motor Learning ........................................................................................................... 18  
2.6 Motivation and Feedback in Rehabilitation ............................................................... 20  
2.7 Games for the Elderly, Serious Games and Games for Health .................................. 23  
2.8 Summary .................................................................................................................... 26  

CHAPTER 3: Virtual Reality and Game Design Background Information .......................... 27  
3.1 Introduction ................................................................................................................ 28  
3.2 Game Components ..................................................................................................... 29  
3.3 Interest Curves ........................................................................................................... 32  
3.4 Teaching Cycle .......................................................................................................... 34  
3.5 Pleasure in games ..................................................................................................... 39  
3.6 Flow in Games ........................................................................................................... 42  
3.7 Game Feel .................................................................................................................. 47  
3.8 Player Types .............................................................................................................. 50  
3.9 Usability Heuristics ................................................................................................... 52  
3.10 Current Video Game Motion and Gesture Hardware - Review .................................. 54  
3.11 Summary .................................................................................................................. 61  

CHAPTER 4: VR Systems Software Communications Middleware ..................................... 62  
4.1 Introduction ................................................................................................................. 63  
4.2 Background Information ............................................................................................ 63  
4.3 LabVIEW SubVI for UDP Communication ................................................................. 66  
4.4 Serial Communication Algorithm ............................................................................. 67  
4.5 Device Agnostic Virtual Environment ....................................................................... 68  
4.6 Implementation into VR Systems for Rehabilitation .................................................. 71  
4.7 Future Work ............................................................................................................... 74
LIST OF FIGURES

Figure 1: Global Schematic of a Virtual Reality System ......................................................... 2
Figure 2: Schematic of Types of Virtual Reality Input ............................................................ 3
Figure 3: Schematic of the “Black Box” Human Information Processing Model ...................... 12
Figure 4: Model of the Stages of Human Response ............................................................... 13
Figure 5: Model of the Closed-Loop Human Information Processing .................................... 13
Figure 6: Closed-Loop Model for Movement Control .............................................................. 15
Figure 7: Schematic of Mechanics, Dynamics, Aesthetics Game and Design Components ...... 29
Figure 8: Schematic of the Mechanics, Dynamics, Aesthetics Framework from the Prespectives of Player and Designer .................................................................................. 30
Figure 9: Schematic of the Elemental Tetrad ......................................................................... 31
Figure 10: Schematic of Distilled Game Elements ................................................................... 32
Figure 11: Schematic of a Theoretical Interest Curve for a Positive Experience ....................... 33
Figure 12: Chart for a generic Teaching Cycle for a Game ...................................................... 35
Figure 13: Screen Capture of the Portal Locations in the First Puzzle in Portal ....................... 36
Figure 14: Screen Capture of the Portal Locations in the Second Puzzle of Portal .................. 37
Figure 15: Screen Capture of the First Player Placed Portal in the Third Puzzle of Portal ......... 38
Figure 16: Diagram of an Interest Curve and Teaching Cycle for a Well-designed Experience ... 39
Figure 17: Schematic of Flow and the Optimal Experience ...................................................... 42
Figure 18: Schematic of the User-System-Experience Model of Player Enjoyment ................ 46
Figure 19: Venn Diagram of the Components of Game Feel .................................................... 48
Figure 20: Image of the Nintendo Wii Remote and Nun’ Chuck ............................................. 54
Figure 21: Image of the Microsoft Kinect .............................................................................. 56
Figure 22: Image of the Playstation Move for the Playstation 3 ............................................. 57
Figure 23: Image of The Novint Falcon ................................................................................. 58
Figure 24: Image of two SensAble Phantom Devices ............................................................ 60
Figure 25: Sample Block Diagram for a LabVIEW SubVI for UDP Communication ............... 66
Figure 26: Screen Capture of the Device Agnostic Virtual Environment Setup Menu ............ 68
Figure 27: Illustration of Simple Geometric Representations of Device Data ...................... 69
Figure 28: Left: Screen Capture of the DAVE IMU Orientation Visualization, Right: Screen Capture of the DAVE Digital Input Visualization ...................................................... 70
Figure 29: Schematic of the Task Order and Loop for DAVE Software ................................. 70
Figure 30: Screen Capture of the VRack SubVI for UDP Communication to Game ................ 71
Figure 31: Screen capture of the DAVE Calibration Testing ................................................... 72
Figure 32: a: Accelerometer perpendicular to Gravity. b: Accelerometer on a 45° with Gravity. 78
Figure 33: Illustration of the Gyroscope rotational axes ....................................................... 78
Figure 34: Image of the 5DOF IMU used in the VRack and 9DOF Razor used in the ATLAS ... 79
Figure 35: Flow Diagram of the Kalman Filter Global Steps .................................................. 81
Figure 36: Illustration of the Rotational change between two states of an IMU ...................... 84
Figure 37: CAD Rendering and Photograph of the Test Bed V1 .......................................... 85
Figure 38: CAD Renderings of the IMU Testbed v2 ................................................................. 86
Figure 39: Left: IMU Testbed v2 with 5DOF IMU mounted, Right: IMU Testbed v2 with Arduino Microcontroller ....................................................................................................... 87
Figure 40: IMU Test Bed Potentiometer Characterization Graph ......................................................... 87
Figure 41: Image of the IMU Testbed v2 setup for VRack Pedal Angle Calibration .......................... 91
Figure 42: Graph of VRack Pedal Angle Calibration 55° Kalman Filter vs. Potentiometer .......... 92
Figure 43: Graph of Left VRack IMU 45° Test vs. Peak ................................................................. 94
Figure 44: Graph of the Left VRack IMU 90° Test vs. Peak .............................................................. 94
Figure 45: Graph of Right VRack IMU 45° Test vs. Peak ............................................................... 95
Figure 46: Graph of Right VRack IMU 90° Test vs. Peak ............................................................... 95
Figure 47: Photograph of IMU Test Bed v2 In Roll-Pitch Mode ....................................................... 98
Figure 48: Images of the IMU Test Bed v2 with Mounted 9DOF IMU for Pitch-Roll .................. 99
Figure 49: Photograph of the IMU Test Bed v2 in Yaw Mode ........................................................... 99
Figure 50: Images of the IMU Test Bed v2 with Mounted 9DOF IMU in Yaw Mode ..................... 100
Figure 51: Graph of the Fifty Degree Characterization of the Roll Axis ........................................... 101
Figure 52: Graph of the Fifty Degree Characterization of the Pitch Axis ......................................... 101
Figure 53: Graph of the Fifty Degree Characterization of the Yaw Axis ......................................... 102
Figure 54: Graph of the Joint Fifty Degree Characterization of the Pitch and Roll Axes ............. 102
Figure 55: Graph of Healthy Subject Supination-Pronation of the Forearm to Roll .................... 103
Figure 56: Graph of Healthy Subject Flexion-Extension of the Wrist to Pitch ............................... 104
Figure 57: Graph of Single Subject IMU Pitch with Wrist Flexion-Extension Range ..................... 105
Figure 58: Graph of Healthy Subject Radial-Ulnar Deviation to Yaw ........................................... 105
Figure 59: Flow Diagram of Steps in Position Calculation Algorithm ........................................... 106
Figure 60: Image of the Characterization Testing Assembly for ATLAS Hand Position .............. 109
Figure 61: Graph of Linear Potentiometer for IMU Test Bed Distance vs. Voltage .................... 109
Figure 62: Graph of the Linear Potentiometer Position and the Raw Accelerometer Data for Single Movement Test ........................................................................................................ 110
Figure 63: Graph of the Corrected Raw Acceleration Data and the Velocity for the Single Movement Test .................................................................................................................... 111
Figure 64: Graph of the Position from the Linear Potentiometer and IMU for the Single Movement Test ......................................................................................................................... 111
Figure 65: Graph of the Raw Acceleration for a Lateral Position Test of a Healthy Subject .......... 113
Figure 66: Graphs of the Raw Velocity and Corrected Velocity for a Lateral Position Test of a Healthy Subject .................................................................................................................. 113
Figure 67: Graphs of Raw Position and Corrected Position for a Lateral Position Test of a Healthy Subject ........................................................................................................................ 114
Figure 68: Schematic of the Skeletal Layout of the Left Hand ......................................................... 120
Figure 69: Schematics of the Hand Joints Measured by the ATLAS Bend Sensors ..................... 121
Figure 70: Diagram of the Global Motions of the Wrist ................................................................. 122
Figure 71: Image of the P5 Glove and Infrared Tower ................................................................. 124
Figure 72: Image of the CyberGlove II ......................................................................................... 125
Figure 73: Image of the 5DT Data Glove 5 Ultra ................................................................. 126
Figure 74: Image of the Accelaglove ......................................................................................... 127
Figure 75: Image of the HandTutor ......................................................................................... 128
Figure 76: Image of the Flexpoint Bend Sensor ........................................................................ 129
Figure 77: CAD Rendering of the ATLAS v3 .............................................................................. 131
Figure 78: Images of the First (left) and Second (right) versions of the ATLAS ....................... 132
Figure 79: Data Flow Diagram of the ATLAS v3 ................................................................. 133
Figure 80: Photograph of the ATLAS v3 Microcontroller Unit (left) and the IMU in case (right) 134
Figure 81: Photograph of all ATLAS v3 Components .......................................................... 134
Figure 82: Images of the Bend Diameter Test bed (left) and the Bend Angle Test Bed (right) 135
Figure 83: Circuit Diagram of the Resistive Sensor Test Bed Circuit .................................. 136
Figure 84: Circuit Diagram of the Passive Filter for Differential Bend Sensor Signals ........ 137
Figure 85: Graph of the Hour Long Low-Pass Filter Only Static Bend Sensor Test .............. 138
Figure 86: Circuit Diagram of the Bend Sensor Amplification and Filter Circuit ................ 138
Figure 87: Graph of the Resistance Testing of the Bend Sensor for Different Radii .............. 139
Figure 88: Graph of Theoretical Voltage from the Bend Sensor Amplification and Filter Circuit .......................................................... 140
Figure 89: Graph of the Hour Long Static Bend Sensor Test ............................................. 140
Figure 90: Graph of Mechanical Testing for the Right Hand Bend Sensors ...................... 141
Figure 91: Graph of Mechanical Testing for the Left Hand Bend Sensors .......................... 141
Figure 92: Scanned images of Hand Tracings from Two Subjects ..................................... 143
Figure 93: Images of the static Testing Templates (left) and Goniometers (right) ............... 143
Figure 94: Image of the Right Hand Index Finger Baseline Bending Test ........................... 144
Figure 95: Photographs of Right Hand Thumb to Small Finger Bending Test (left) and the 45°-45° Bending Test (right) ................................................................. 144
Figure 96: Photograph of the Right Hand 45 MCP Bending Test ...................................... 145
Figure 97: Chart of Hand Size Data by Gender .................................................................. 146
Figure 98 Charts of Goniometric Data Average and Standard Deviation ......................... 147
Figure 99: Graph of the Exponential Angle Trend vs. Voltage for Right Hand .................. 148
Figure 100: Raw Data Chart of Right Hand Index Finger Bending by Test ....................... 149
Figure 101: Raw Data Chart of Right Hand Middle Finger Bending by Test ..................... 149
Figure 102: Raw Data Chart of Right Hand Ring Finger Bending by Test .......................... 150
Figure 103: Raw Data Chart of Right Hand Thumb Bending by Test ............................... 150
Figure 104: Raw Data Chart of Right Hand Index Finger Bending by Gender .................. 151
Figure 105: Raw Data Chart of Right Hand Middle Finger Bending by Gender ................ 151
Figure 106: Raw Data Chart of Right Hand Ring Finger Bending by Gender .................... 152
Figure 107: Raw Data Chart of Right Hand Thumb Bending by Gender ............................ 152
Figure 108: Circuit Diagram of the New Bend Sensor Amplification and Filter Circuit ...... 153
Figure 109: Diagram of Multiple User Virtual Interactions .............................................. 157
Figure 110: Final Prototype Virtual Scene Movement Schematic ..................................... 159
Figure 111: Screen Capture of the Multiple User Virtual Environment for Rehabilitation Virtual Scene Rendering During a Session ......................................................... 160
Figure 112: Software Communication Global System Schematic ...................................... 161
Figure 113: Screen Capture of Final Software Prototype Simple Maze ............................ 162
Figure 114: Screen Capture of the Advanced Maze for the HERRI Software .................. 163
Figure 115: Direct Motion Schematics for ATLAS VR Scenes ......................................... 165
Figure 116: Indirect Motion Schematics for ATLAS VR Scenes ....................................... 166
Figure 117: Diagram of A Generic Teaching Cycle for a Game ....................................... 167
Figure 118: Screen Capture of the ATLAS Calibration Virtual Scene .............................. 171
Figure 119: Screen Capture of the Indirect A (Penguin Panic) ATLAS Virtual Scene ....... 171
Figure 120: Screen Capture of the Direct B (Grasp and Relase) ATLAS Virtual Scene ............... 172
Figure 121: Schematic of Distilled Game Elements ................................................................ 175
Figure 122: Schematic of a Teaching Cycle for a Generic Game ........................................ 176
Figure 123: Image of the 5DOF IMU and 9DOF Razor IMU .................................................. 177
Figure 124: CAD Rendering of the Current Version of the ATLAS ..................................... 178
Figure 125: Screen Capture of the Calibration Virtual Scene for the ATLAS ..................... 179
Figure 126: Image of VRack System Components .................................................................. 190
Figure 127: Hand Enhancement Robotic Rehabilitation Interface Global System Schematic ... 191
Figure 128: Image of the Hand Enhancement Robotic Rehabilitation Interface .................... 192
Figure 129: Two Degree of Freedom Robotic Hand Rehabilitation Device ......................... 193
Figure 130: Squid system Communication Schematic ............................................................ 194
LIST OF TABLES

Table 1: Game design criteria for stroke rehabilitation of elderly users............................................. 24
Table 2: Stroke Game Design Lessons. ................................................................................................. 25
Table 3: Mental States of flow............................................................................................................. 43
Table 4: Criteria for GameFlow ........................................................................................................... 44
Table 5: Player Types based on quality perceptions ........................................................................... 51
Table 6: Steps in the MATLAB Algorithm for VRack Pedal Angle ..................................................... 89
Table 7: Steps in the Kalman Filter Function for the VRack Pedal Angle ........................................... 90
Table 8: 5DOF IMU Characterization Values from Test Bed Mounted IMU ......................................... 92
Table 9: 5DOF IMU Characterization Values from VRack IMUs .......................................................... 93
Table 10: Steps in the ATLAS Hand Orientation Setup Function .......................................................... 96
Table 11: Steps in the ATLAS Hand Orientation Loop Function ........................................................... 97
Table 12: New Steps in the Position Algorithm for the ATLAS ............................................................. 107
Table 13: Normal Range of Motion Values for Finger Joints ................................................................. 121
Table 14: Normal Range of Motion Values for Wrist Motion ............................................................... 123
Table 15: Game Components of ATLAS Virtual Scenes ...................................................................... 167
Table 16: Patient Specific User Feedback based on Player Type............................................................ 169

LIST OF EQUATIONS

Equation 1: Fitts’ Law .......................................................................................................................... 16
Equation 2: The Power Law of Practice .............................................................................................. 19
Equation 3: UDP Data string Format .................................................................................................. 66
Equation 4: Equations for the Prediction Step of the Discrete Kalman Filter ..................................... 81
Equation 5: Equations for the Measurement Update Step of the Discrete Kalman Filter ................. 82
Equation 6: Rotation Matrices for X, Y, and Z unit vectors ............................................................... 83
Equation 7: Orthogonal Left-Handed Rotation Matrix ........................................................................ 83
Equation 8: Direction Cosine Matrix .................................................................................................. 84
Equation 9: Voltage Differential Equation .......................................................................................... 137
Equation 10: Circuit Equation for the Amplifier of the ATLAS Bend Sensors ................................. 139

LIST OF CODE FRAGMENTS

Code Fragment 1: Examples of Arduino Serial Communication......................................................... 67
Code Fragment 2: Main Task in the ATLAS Calibration Software ..................................................... 73
Code Fragment 3: IMU Test Bed Sweep Algorithm ........................................................................... 88
Code Fragment 4: Single Axis Kalman Filter for Hand Position ....................................................... 107
Code Fragment 5: Band-pass Filter for Hand Position ....................................................................... 108
ACKNOWLEDGEMENTS

It is my pleasure to acknowledge the people who contributed to the completion of this work.

First I would like to acknowledge Dr. Mavroidis, my adviser, and my co-adviser, Dr. Holden. Their insight and assistance throughout the duration of this work was invaluable. I would also like to acknowledge Dr. Mourant for his contributions as part of my committee.

Next I would like to recognize and thank the members of the Biomedical Mechatronics Laboratory, in particular Richard Ranky, Ozer Unisarjikli, and Maciej Putrizinski. I never would have finished without their support.

The work presented here would not have been possible without the help of the student groups for the ATLAS and Squid projects, the students from Dr. Holden’s laboratory, and the other graduate students I worked with on all my projects.

DEDICATION

I would like to dedicate this work to my family who supported me throughout its duration. Thanks Dad, Mom, Scott, Seth, and Erin.
CHAPTER 1: INTRODUCTION

GLOBAL SCHEMATIC OF A VIRTUAL REALITY SYSTEM
1.1 VIRTUAL REALITY SYSTEMS

Recent technological advancements in the fields of computer processing power, displays, and graphics have blended with input device technologies to create the foundation for virtual reality (VR) systems. VR systems have found uses in many industries including entertainment (computer and videogames), defense, aviation, machine operation, medicine, and rehabilitation.

There are four main components in every VR system. Figure 1 below presents a flowchart of the interactions of these components and how they relate to the user and practitioner.

![Figure 1: Global schematic of a virtual reality system](image)

There are three hardware components of a VR system: the input device, display, and computer. The input device is used by the user to interact with the virtual environment. The display gives visual feedback to the user by displaying the virtual environment. The computer is connected to both the display and input device. The computer runs the VR software and any other necessary programs that are needed in the VR system. The focus of this research project is on the design and use of VR software and virtual environment design for use in rehabilitation.
The input devices used in rehabilitation systems only have certain degrees of freedom to reduce complexity and cost. Only linear or rotational degrees of freedom are possible with these devices. Digital (high or low) degrees of freedom can also be incorporated into these input devices. Figure 2 below presents the most common degrees of freedom.

![Figure 2: Schematic of Types of Virtual Reality Input](image)

The linear degrees of freedom are x, y, and z and the rotational degrees of freedom are roll, pitch, and yaw, all labeled above. These degrees of freedom are usually sensitized through the use of analog sensors. Digital sensors can be used which give a Boolean (true/high or false/low) signal. This information is important because these limitations are inherent in every motor rehabilitation device. With this knowledge it is possible to abstract the input device and design systems that can be used with any device.

1.2 SIGNIFICANCE

Stroke has significant impact on the lives of many Americans each year (discussed further in Section 2.2). Substantial time is spent by therapists setting up home rehabilitation regiments after stroke because of the limited number of total visits to clinical facilities [1]. With increasing
occurrences of stroke each year having effective in-home devices could assist with rehabilitation. Home based stroke rehabilitation games have the potential to help patients rehabilitate after a stroke [2]. This can relieve some of the pressure of the patient traveling to a facility for treatment. New technologies are allowing for low-cost immersive VEs to be developed [3]. These include low cost input devices (discussed further in Section 3.10) and more off-the-shelf game technology. Also robotic rehabilitation devices have the potential to revolutionize delivery and availability of motor therapy [3]. The creation of a low-cost in-home system for rehabilitation could be beneficial to many patients post stroke.

1.3 DEFINITIONS AND NOMENCLATURE

This section has an alphabetical glossary of terms that are used throughout the rest of the document. Project names and certain terms are defined within the document.

**Avatar**

An “avatar” is the virtual representation of the user in a virtual environment, scene, or game. If there is no graphical representation of the avatar the term represents the presence of the user in the virtual environment, scene, or game.

**DOF**

The acronym DOF means “degrees of freedom” and refers to a number of axes or points of actuation of a sensor or device.

**Event**

“Events” is used in this thesis to describe reactions of the virtual environment to the avatar and the mechanics.
The term “game” has many different definitions depending on the scope of the subject. In this thesis the term “game” is used to describe virtual environments and virtual scenes where the user has a specific task and is given a score based upon how they perform that task. Note that for this thesis the terms “game”, “video game”, and “computer game” are interchangeable and identical.

A game engine is a software program or group of programs that is designed to aid in the development of video or computer games. More specifically this means that the game engine includes classes, libraries, or methods to create and display geometry as well as control physics properties.

The term “goal” is used to describe the positive endpoint of the mechanic. It is also used as motivation for the user to interact with the virtual environment.

The term “input device” as used in this thesis means a device that provides data to a software program. This includes rehabilitation devices with sensors and video game console controllers.

“Interface” as used in this thesis means the software that the practitioner or patient uses to control the input device and/or the virtual reality, scene, or environment. The term “interface” is also synonymous with “user interface” or “graphical user interface” in this thesis.
**IMU**

IMU is the acronym for “inertial measurement unit” which is a family of inertial sensors that combine accelerometers, gyroscopes, and magnetometers.

**Mechanic**

The term “mechanic” is used to describe a single component of a virtual scene, environment, or game such as the player avatar’s movement or the scoring. It may also be used as “game mechanic”.

**Player**

The term “player” is synonymous with “user”, “patient”, and “subject” within the context of this document. That means that the “player” is the person who is using the virtual system.

**Practitioner**

This thesis uses the term “practitioner” to mean the person who is running the rehabilitation session. This can include physical therapists, occupational therapists, doctors, or engineers.

**User**

The term “user” as used in this thesis means the person that is controlling the input device. In general this means the patient using the rehabilitation device but the term user is used to cover all possible users including healthy people and other practitioners.

**Virtual Reality, Environment, and Scene**

The term “virtual reality” is used in this thesis to denote a software program is controlled by a user via an input device and that has a graphical component. “Virtual environment” refers to the graphical component of the virtual reality software, it is also where the user’s avatar inter-
acts with other virtual objects or performs task. A “virtual scene” is a section of the virtual environment and is used to describe a specific mechanic or task that takes place in the larger virtual environment.

1.4 DISSERTATION OUTLINE
This document contains eight chapters. The first is an introduction which contains this section as well as information on nomenclature, a glossary, significance, and contributions to the field (the next section). The second and third chapters present background information on relevant topics in physical therapy, psychology, game design, and virtual reality. The fourth chapter presents the middleware used in the work presented. The fifth chapter discusses the use of IMUs in two specific devices, the VRACK and ATLAS, including the hardware, algorithms, and testing. Chapter six and seven both pertain to the ATLAS system, the first presents the hardware and the second the virtual environments. The last chapter has the conclusions found during this work.

1.5 CONTRIBUTIONS TO THE FIELD
In terms of the design of virtual environments the contributions to the field are a new model for the components of a virtual scene and the teaching cycle that are presented in Sections 3.2 and 3.4 respectively. Another contribution is the DAVE software for prototype rehabilitation devices presented in Chapter 4. Chapter 5 presents new algorithms and uses for inertial sensors in rehabilitation. Chapter 6 and 7 explain the contribution to the field of a low cost glove rehabilitation device, the ATLAS. There have been numerous publications and patents as a result of this work, they can be seen below.
Publications:

- Development of a Low-Cost Virtual Reality-based Smart Glove for Rehabilitation. Accepted to the International Conference Series on Disability, Virtual Reality and Associated Technologies in September 2012.
- Patient Specific Ankle-Foot Orthosis Using Layered Manufacturing [4].
- Design of a Haptic System for Hand Rehabilitation Integrating an Interactive Game with an Advanced Robotic Device [5].
- NUVABAT: Northeastern University Virtual Ankle and Balance Trainer [6].
- Modular Stationary Bicycle Mechatronic Kit Interfaced with a Virtual Environment for Rehabilitation of Individuals with Movement Asymmetry [7].
- A Robotic Hand Rehabilitation System With Interactive Gaming Using Novel Electro-Rheological Fluid Based Actuators [8].
- Dielectric Electroactive Polymer Actuator as a Haptic Interface [9].
- Design of a Low Cost Multiple User Virtual Environment for Rehabilitation (MUVER) of Patients with Stroke [10].

Patents:

CHAPTER 2: PHYSICAL THERAPY AND REHABILITATION BACKGROUND INFORMATION

FLOW DIAGRAM FOR THE HUMAN PROCESSING MODEL
2.1 INTRODUCTION
To create virtual reality systems it was important to understand topics in physical therapy and rehabilitation. This chapter includes information on the relevant topics and includes a review of current research in the field.

2.2 STROKE
In the United States alone, nearly 800,000 people experience a new or recurrent stroke each year [16]. At six months post stroke, 55-75% of survivors still have impaired function in the arm, and in cases with initial upper extremity paralysis, complete motor recovery has been reported at <15% of cases [17]. In the US estimated total costs for stroke care are now near $30 billion/year [18]. Strokes cause 9% of all deaths around the world, making it the second most common cause of death behind ischemic heart disease [19]. Strokes accounts for approximately 2-4% of worldwide health-care costs [19].

Strokes cause high stress in patients [20]. Rehabilitation, for patients, is fundamentally a process of relearning how to move to carry out their needs successfully [21]. Another trend in rehabilitation is the study of constraint-induced physical therapy. In these studies patients who are many months post stroke have their unaffected arm constrained in some way and are required to do tasks with their affected upper-limb [22, 23]. Taub states that patients post stroke learn to replace the goal-oriented movements of the affected limb with the unaffected resulting in less need to use the affected limb. By constraining the unaffected limb the patient is forced to use and increase practice with the affected limb.
2.2.1 LIFE POST-STROKE
Stroke obviously affects the life of a patient. In this section information is presented about a patient’s life post stroke. This includes information on long-term rehabilitation as well as leisure as playing games is considered a leisure activity.

The final effect of a stroke on a patient depends on factors such as genetic characteristics, psychological defenses, state of health, family environment, community involvement, economic resources, standard of living, and religious faith [20].

Leisure activities can provide important health benefits such as promoting general fitness, strength, coordination and posture [24, 25]. Passive activities become more popular post-stroke [26]. Leisure activities decline with age and the rate of decline increases due to stroke [25]. There is also social deterioration for most patients post-stroke [27]. Along with other problems including more psychiatric symptoms and lower functional ability all these issues give patients post-stroke lower life satisfaction [27]. Social disintegration was significantly associated with reduction in global life satisfaction [27]. Stroke can lead to decreased socialization outside the home with all but close family members [27]. If leisure is to be believed to be an overall goal of rehabilitation then practicing and transferring skills vital to leisure pursuits should be implemented into rehabilitation [25].

2.3 HUMAN INFORMATION PROCESSING MODELS
Virtual Reality systems engage many human senses, usually visual, auditory, and haptic or touch. It is therefore important to understand how these senses will react in real-time to a VR system.
The way in which humans process information is analogous to the way computers process information. In the most basic form, a “black box” of human cognition can be broken into three phases: input, processing, and output [28]. Figure 3 below reproduced from [28] displays these three phases visually.

**FIGURE 3: SCHEMATIC OF THE “BLACK BOX” HUMAN INFORMATION PROCESSING MODEL [28]**

The input comes from an outside stimulus, then the “black box” of the human processes that stimulus, once processed an output is performed by the human in the form of a reaction. For motor control purposes that output is usually a movement. Conveniently, this model is very similar to an open loop engineering system [29]. This model is too basic to use beyond an introduction of the topic, thus a more advanced model is needed.

To create that more advanced model it is important to look into the “black box” and decipher what parallel or serial operations are occurring. Three stages can be discerned from the “black box”: stimulus-identification, response-selection, and response-programming. Together these stages form a human’s processing of a stimulus and creation of a response, the time it takes to complete these phases can also be referred to as reaction time [28]. Figure 4 reproduced from [28] on the following page presents this model visually.
In this model the input leads directly to the stimulus identification stage. The more advanced model presented above is missing another important part of human information processing, feedback. Humans react to their environment through a closed loop, not an open loop [28, 30]. To complicate further humans also use their memories (both working and long-term) to process the stimuli. So another model is needed that incorporates these features, the figure of that model reproduced from [28] is presented below.
This model now includes the feedback that a human receives from their environment and can use to modify their previous movements or decisions. Of note long term memory is used before working memory. For the creation of game for rehabilitation this means that the design must account for previous biases or learning that the player may have as it can affect how intuitive a game is to learn and play [31].

2.4 Motor Control

Motor control is an important part of rehabilitation. With VR systems for rehabilitation visual and auditory feedback is often used to enhance the experience and provide feedback to the player. Motor control relies heavily on the concept that humans use sensory information to regulate our movements [28].

Vision is the most critical receptor for supplying information about the movement of objects in space [28]. Games (and rehabilitation) rely on correct and accurate movement perception, the work of Keele and Posner [32] showed that 150-190 ms is when visual perception gains an advantage. Games used for motor rehabilitation are real-time so knowledge of when feedback should occur and the speed at which a human can use their visual senses is very important. Building on the concept of closed loop control Schmidt and Lee presented a model for use in motor control to understand the flow of movement and information, which can be seen in the figure on the next page, Figure 6, reproduced from [28].
Much like the closed loop model in the previous section this model relies on input to begin the loop and then continues to use the previous iteration to adjust. However, this model accounts for feedback from muscles, movement, and the environment making it better suited to the real-time of a virtual environment. It is important to note that many patients with stroke suffer many deficits, and that virtual reality can help substitute for this loss by providing augmented feedback to enhance motor learning.

When making input movements for a virtual environment it is important to address both the speed to complete and also the accuracy of the movement. Fitts’ Law has been used to under-
stand this tradeoff [33]. It describes the relationship between amplitude of the movement, the target width, and the resulting average movement time. The equation for Fitts’ law is below.

**EQUATION 1: FITTS’ LAW**

\[ MT = a + b \left( \log_2 \frac{2A}{W} \right) \]

\( MT \) is the average movement time, \( A \) is the amplitude of the movement, and \( W \) is the target width. It is derived from the general form of the linear equation [28]. The empirical constants, \( a \) and \( b \), are required for the mathematical model to fit the experimental data. The \( y \)-intercept, the value of \( MT \) where the line best fit crosses the \( MT \) axis, corresponds to \( a \). The constant \( b \) is the slope. Fitts’ law implies that there is an inverse relationship between the difficulty (or required accuracy) and the speed [33]. When choosing movements to be used in virtual environments for rehabilitation this relationship must be accounted for in order to maintain motivation and immersion for the player.

In Chapter 6 the Angle Tracking and Location At-home System for Bimanual Rehabilitation (AT-LAS-BR) is introduced. The system focuses on reaching and grasping tasks so it is important to look at the factors that contribute to coordination. There have been two views on reaching and grasping: temporal planning and spatial planning [28]. Temporal planning was based on work by Woodworth and then expanded by Jeannerod and Arbib [34]. According to that work reaching and grasping are considered to be two distinct components, transport and grip formation. Furthermore the extrinsic attributes of an object are used to plan the transport movement while the grip formation movement is determined from the intrinsic attributes. However, there have been many studies to show that this model is not correct and that the movements are not tem-
poral based but instead are spatially based and linked [28]. Spatial planning considers the reaching and grasping to be a single aiming task [35, 36]. In both models the visual information is used to assess both reaching and grasping where the difference is that in the model presented by Rosenbaum et al. a goal posture is compared to the current configuration of the arm.

As the ATLAS-BR project moves forward another consideration will be for bimanual tasks that have certain caveats in reaching and grasping, mainly that the two hands may do different movements and tasks simultaneously, including interacting with one another. Stroke impacts one arm thus further complicating bimanual coordination. Goodman et al. found that if the same task is done with each arm while one must navigate an obstacle that the time to complete for both arms suffers [37]. While Schmidt and Lee note that more research is needed in this area the work to date would indicate that a temporal coupling of the limbs is constrained more for a reach and grasp task than a Fitts’ aiming only task [28]. Another interesting finding is that when doing different unimanual rapid tasks bimanually the arms seem to have coordination [38].

Dynamic-Pattern Theory is another theory in motor control of note for bimanual input into virtual environments. The theory makes an assumption that a system changes state temporally to find stability [28]. The key to this theory is variability, something that is assumed to give a linear decline in the efficiency of a motor control system. In Dynamic-Pattern Theory variability is used as a trigger to change the organization of movements [28].
Motor control has many important topics for the work presented in this document. However, in rehabilitation the goal is not simply motor control but also motor learning. A topic discussed in the next section.

2.5 Motor Learning

Motor learning pertains to the process with which humans learn new motor skills over time by practicing. This section presents the most notable theories from the field of motor learning in healthy subjects as well as pertinent information for use with virtual reality and games. Schmidt and Lee define motor learning using four characteristics [28]:

1. **Motor learning is a set of processes**: Learning is a set of internal changes that occur when a person practices.

2. **Learning produces an acquired capability for movement (habit)**: Learning produces a state or capability for moving more skillfully.

3. **Motor learning is not directly observable**: Changes due to learning must be inferred by experiments or assessment.

4. **Motor learning is relatively permanent**: This means that easily reversible alterations in mood, motivation, or internal state like fatigue are not thought of as due to learning.

Through these characteristics it is obvious that practice is the most important part of learning. More learning will occur if there are more practice trials [28]. The Power Law of Practice can be used to model how practice can change the time to complete a task. The equation for the Power Law of Practice can be seen below.
In that equation $T$ is the time to complete a task, $a$ and $b$ are constants and $P$ is the measure of the amount of practice (e.g. number of trials). The importance of this equation is the inverse relation between $T$ and $P$, as $P$ becomes larger (and it modified by $a$ and $b$) $T$ becomes smaller. In plain terms this means that as practice increases, time for the task decreases. When a power function is graphed logarithmically for time the number of trials will yield a linear function [28, 33].

Motor learning has been described as occurring in phases of learning [28, 39]. These phases include a cognitive phase, associate phase, and autonomous phase. This work began with Bryan and Harter in the late nineteenth century with their creation of a hierarchy of habits [40]. They found in their work that subjects would have plateaus of skill before acquiring a higher skill. This reinforces the ideas of phases instead of a linear or steady increase in skill by learning.

When a new task is presented to a learner the first problem that must be solved is finding the best motor strategy to achieve the goal of the task[28]. Most learners adopt a “guess and check” method of finding the strategy, they try multiple strategies and discard those that do not work. In this initial phase is where the largest gains in performance are found. During this phase the use of many training aids (including instructions and augmented feedback discussed in the following section) is very effective.

Once the cognitive phase is over the learner knows the most effective way of doing the task and now focuses on refining that method. In this phase performance gains slow and become
more gradual. There is no longer a verbal link to the task and the learner focuses on how to do it instead of what to do [39].

In the last phase of learning the learner enters the autonomous phase. This can take weeks, months, or years depending on the task and the type of practice [28]. As the name for the phase would suggest, learners in this phase can perform the task automatically leading to very slight or no gain in their skill. This allows the learner to focus on another task or another aspect of the task.

Learning of a new task is applicable to both game playing and rehabilitation as they both have the goal of higher skill with practice. Another important consideration for motor learning in rehabilitation is that patients are often re-learning previously well-learned movements. A topic not addressed here but in the following section is how feedback and motivation can enhance learning.

2.6 Motivation and Feedback in Rehabilitation

Motivation is important for learning and rehabilitation and feedback is a cornerstone to motivation. In this section research and relevant work in these areas is presented. Motivation and feedback are also integral components of good game design.

It is important to maximize motivation so that more learning can be achieved. In the field of psychology there are thought to be two types of motivation, intrinsic and extrinsic [41, 42]. Intrinsic motivation means that a person does a task for the enjoyment of the task itself, whereas extrinsic motivation means that a person does a task because of a separate outcome or goal [42]. A combination of both types of motivation is needed for rehabilitation as the experience
should be enjoyable and have the separate goal of increasing motor skill. Two important considerations in motivation are giving weight to the task by feedback [28] and setting proper goals.

Maslow’s Hierarchy of Needs [43], has been held up for over 50 years as a possible theory of motivation. In his seminal paper, *A Theory of Human Motivation*, Maslow lays out the hierarchy with the most basic of human needs at the lower levels and more complex needs at the higher levels. Lower level needs include physiological needs such as breathing, food, water, sleep as well as safety needs such as security of body, employment, and resources. Higher level needs include self-actualization and esteem needs such as confidence, creativity, and problem solving. The hierarchy is not without criticism as several researchers have stated that human needs have no definite hierarchy [44, 45].

Schmidt and Lee contend that there are two different kinds of feedback, inherent and augmented. Inherent feedback is what a person receives as feedback from their own senses internally. In contrast augmented feedback comes from supplemental information about the task [28]. The focus in game design and rehabilitation is augmented feedback because it can be used to overcome the sensory impairments some patients may have. Augmented feedback can occur in many ways, a list of the possible dimensions can be seen below [28]:

- **Concurrent**: Presented during the movement
- **Terminal**: Presented after the task
- **Immediate**: Presented immediately after the sub-task
- **Delayed**: Delayed in time after the sub-task
- **Verbal**: Presented in spoken form
- **Nonverbal**: Presented in a form not capable of being spoken
- **Accumulated**: Feedback to represent an accumulation of past performance
- **Distinct**: Feedback that represents each practice repetition of a task
- **Knowledge of Results (KR)**: Communicated post task information about the movement in the environment
- **Knowledge of Performance (KP)**: Communicated post task information about the nature of the movement pattern

KR and KP are two important and well researched areas of feedback for motor learning. KR is essentially feedback after the task about the movement in terms of the environment goal [28]. KP is often referred to as biofeedback as it pertains directly to the accuracy of the movement pattern, not its accuracy in space [46]. Feedback (specifically immediate feedback) is a tenet of Game Feel and is discussed further in section 3.7.

Goals and goal-setting are important factors in learning. There has been research into different types of goals, ranging from specific difficult goals to simple or “do-your-personal-best” goals [47]. There is still debate on what types of goals are best for learning. Goals of any form have been shown to improve motivation for learning and also retention in rehabilitation [48]. Ideal goals should be specific, measurable, activity-related (intrinsic), realistic (achievable), and time-specific [48].

There has been research into motivation specifically for stroke rehabilitation with interesting conclusions. Research has shown that motivation is affected by the rehabilitation environment,
suggesting that interaction with other patients and observing their progress in a well-maintained physical space can increase motivation [49]. This is a potentially important finding as work in virtual environments moves to allowing users to interact with each other in real-time. Favorable comparisons with other stroke patients can increase motivation while unfavorable comparisons can have the opposite effect [48]. Age is commonly thought by professionals to have a negative impact on motivation for rehabilitation which is important for the populations of stroke patients [49]. Awareness of factors that affect motivation for rehabilitation can only have positive effects on outcomes [50]. Information about the rehabilitation plan can increase motivation [48].

Motivation is a large factor in why VR is thought to aid rehabilitation and motor learning. The general information and findings presented in this section are revisited during the design of the virtual scenes presented in Chapter 7.

2.7 GAMES FOR THE ELDERLY, SERIOUS GAMES AND GAMES FOR HEALTH

Research has shown that virtual environments and games can provide significant motivation to post-stroke patients [51] and can promote motor rehabilitation [52-56]. Patients prefer using exercises augmented with virtual reality to doing exercises in only a physical environment [57]. Patients post-stroke can benefit from repetitive, goal-oriented, robotic therapy [58]. While there have been many studies into the effectiveness of VR rehabilitation they are not far reaching or standardized [53]. The table on the following page contains information on game criteria specifically for stroke rehabilitation.
Three popular active activities for patients post-stroke are gardening, cooking, and indoor games (bingo, cards, etc.) [26]. These activities would possibly make good themes for games designed for stroke patients because of their popularity with the population. Research has shown that very few patients (less than 15%) can make their own meals independently post-stroke [59]. Therefore cooking becomes an even more advantageous choice for theme because of the possibility of overcoming this deficit.

Design attributes for stroke rehabilitation games are: social context, motion type and cognitive challenge [2]. Social context refers to interaction with others in the virtual environment. This can be accomplished by using one of the virtual interactions described in Section 7.2.1 or by using a computer player programmed into the game. Motion type refers to games that use the physical motion of the patient as an input, usually a specific rehabilitation exercise. Cognitive challenge refers to the type of games created to stimulate a patient cognitively. Because many stroke patients have cognitive issues the level of difficulty required in a game aimed at rehabilitation is very important. Lessons from designing games for stroke rehabilitation are in the table on the following page.

### TABLE 1: GAME DESIGN CRITERIA FOR STROKE REHABILITATION OF ELDERLY USERS [51].

<table>
<thead>
<tr>
<th>Criteria for Stroke Rehabilitation</th>
<th>Criteria for Elderly Entertainment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptability to motor skill level</td>
<td>Appropriate Cognitive Challenge</td>
</tr>
<tr>
<td>Meaningful tasks</td>
<td>Simple objective/interface</td>
</tr>
<tr>
<td>Appropriate feedback</td>
<td>Motivational feedback</td>
</tr>
<tr>
<td>Therapy-Appropriate range of motion</td>
<td>Element of social activity</td>
</tr>
<tr>
<td>Focus diverted from exercise</td>
<td>Appropriateness of genre</td>
</tr>
<tr>
<td></td>
<td>Creation of new learning following guidelines of experts</td>
</tr>
<tr>
<td></td>
<td>Sensitivity to decreased sensory acuity and slower response</td>
</tr>
</tbody>
</table>

Three popular active activities for patients post-stroke are gardening, cooking, and indoor games (bingo, cards, etc.) [26]. These activities would possibly make good themes for games designed for stroke patients because of their popularity with the population. Research has shown that very few patients (less than 15%) can make their own meals independently post-stroke [59]. Therefore cooking becomes an even more advantageous choice for theme because of the possibility of overcoming this deficit.

Design attributes for stroke rehabilitation games are: social context, motion type and cognitive challenge [2]. Social context refers to interaction with others in the virtual environment. This can be accomplished by using one of the virtual interactions described in Section 7.2.1 or by using a computer player programmed into the game. Motion type refers to games that use the physical motion of the patient as an input, usually a specific rehabilitation exercise. Cognitive challenge refers to the type of games created to stimulate a patient cognitively. Because many stroke patients have cognitive issues the level of difficulty required in a game aimed at rehabilitation is very important. Lessons from designing games for stroke rehabilitation are in the table on the following page.
There has also been research into the patterns of design for serious games. The list below presents this information.

Design Pattern Taxonomy [60]:

- **Right level of difficulty** – the game must not exceed the abilities of the patient, nor put them in danger of discomfort or aggravation.
- **Time limits / Quick games** – the game will have a set time limit in which the patient should attempt to score as highly as possible. The time limit should be fairly short to avoid exhausting the patient.
- **Direct feedback** – any points scored should be clearly indicated with both visual and auditory feedback. Feedback should also be given at the end of each session as to the patient’s progress in comparison to previous sessions.
- **Precise maneuvering** – the game should provoke precise range movements in the patients arm, and should not award points for incorrect motions.
Beyond design concerns there are other important factors for the success of a virtual scene including how the patient interfaces with the VR [53]. Patients are more interested in games that automatically adapt to their abilities than a larger selection of games [54]. Subjects preferred games that mimicked real world activities slightly [54]. Providing an appropriate amount of challenge and success is important [54].

2.8 SUMMARY

This chapter contains information on various topics regarding rehabilitation and physical therapy. Of note is the information on Stroke, particularly the statistics that prove the need and significance of this work. The subsequent topics on human information systems, motor control, and motor learning inform how a subject interacts with a virtual reality system. Motivation and feedback and game design bridge this chapter with the one that follows.
CHAPTER 3: VIRTUAL REALITY AND GAME DESIGN BACKGROUND INFORMATION

TEACHING CYCLE & INTEREST CURVE

COMBINATION OF A TEACHING CYCLE AND AN INTEREST CURVE
3.1 INTRODUCTION

The field of video game design and particularly serious video game design is new and exciting. It is only in recent years that game design has become an academic field and its research base is in the early stage. This chapter serves as an introduction to several key concepts in game design generally and in serious game design for stroke rehabilitation specifically.

The definition of a game is something that has been discussed by several scholars and it is beyond the scope of this document to explain all the definitions. To give some insight into what a game is and isn't Schell has a list of game qualities which can be applied to virtual environments and experiences [61].

Qualities of games:

1. Games are entered willingly.
2. Games have goals.
3. Games have conflict.
4. Games have rules.
5. Games can be won and lost.
6. Games are interactive.
7. Games have challenge.
8. Games can create their own internal value. (scoring systems)
9. Games engage players.
10. Games are closed, formal systems.
11. Games are problem solving activities.
This chapter presents theories on games specifically about the player experience. As stated in the previous chapter, motivation is paramount to help aid motor learning in rehabilitation.

### 3.2 Game Components

As a method to design or analyze a game it is prudent to break down a game into components. There is some previous research in this area that helps to classify game components and their importance.

The Mechanics-Dynamics-Aesthetics (MDA) framework was originally introduced by Marc Le-Blanc in a workshop and presentation and the later expanded by an academic paper with Robin Hunicke and Robert Zubek [62]. In their definitions mechanics are various actions, behaviors, and control mechanisms afforded to the player within a game context. Dynamics work to create aesthetic experiences and aesthetics are the appearance of the game world to the player. By using this framework the authors came to the conclusion that game design components could be linked to gameplay components as seen in Figure 7 below [62].

![Diagram of Game Components and Design Components](image)

**Figure 7: Schematic of Mechanics, Dynamics, Aesthetics Game and Design Components [62]**
The three game components: rules, system, and fun and connected to mechanics, dynamics and aesthetics in the design of a game. Mechanics corresponds to the underlying algorithms in the game, the systems. Dynamics describes how the mechanics act on a player of the game. The aesthetics invoke emotional responses in the player during the experience. The set of design components is where the MDA framework draws its name. A figure showing the perspective of the player and designer can be seen below [62].

![FIGURE 8: SCHEMATIC OF THE MECHANICS, DYNAMICS, AESTHETICS FRAMEWORK FROM THE PERSPECTIVES OF PLAYER AND DESIGNER [62]](image)

With this framework the authors correlated a relationship between the design of a game and the player. The roles of those individuals and how the components connect them are important in understanding what is needed to design a successful game experience.

Schell uses a different structure to categorize the elements of a game called the Elemental Tetrad, as seen in Figure 9, on the following page [61]. The visibility of the elements relates to the player’s view, again showing the relationship between player and designer. Mechanics are the procedures and rules of a game. Story is the sequence of events that unfolds in a game. Aesthetics is how a game looks, sounds, smells, tastes, and feels. Technology is the tool that makes a game playable.
Schell stresses that none of the elemental is more important than any other, thus the element’s diamond arrangement. He also stresses that each element is essential to successful game design. Beyond sharing the perspective of the player the Elemental Tetrad and the MDA framework share two terms: mechanics and aesthetics. While the terms are shared the author’s definitions for the terms are different. Mechanics in the MDA framework is analogous to the technology element in the Tetrad. Mechanics in the Tetrad is analogous to Dynamics in the MDA Framework. Theme and Aesthetics in the Tetrad are combined into only Aesthetics in the MDA framework.

With knowledge of these models a new model was needed to simplify the design and analysis of games specifically for rehabilitation. Both of the models do not reference the player directly, which belies the importance of the player for interaction. Terms from both models were generic enough to fit most games without being specific enough to be truly usable for design. A new
model must correct these issues and begin with focus on the role of the player and use the definitions of games and play to find the other necessary components: world, goal, and mechanic as seen in the figure below.

![Figure 10: Schematic of Distilled Game Elements](image)

The player is the person that interacts with the game elements. The player exists within the game world, a place where the game will take place. The player must complete the mechanic in order to achieve the goal. This is a very player-centric model, in contrast to the two previous models presented the world and goal correspond to the aesthetics/theme and the mechanic corresponds to the dynamics/mechanics. What this new model does not address is the technology (Elemental Tetrad) or mechanics (MDA) because each element (including the player) has underlying rules and algorithms.

### 3.3 Interest Curves

The interest level of an experience can be expressed by using an interest curve [63]. An interest curve is a graphical representation of a participant’s interest in an experience over time. While this generic definition can be used for all types of experiences (roller coasters, movies, plays,
first dates, etc.) it correlates very well with good game design. While interest has no quantity to measure there is ongoing research in games on using bio-sensors to add quantifiable data to experience elements like fun, enjoyment, immersion, and interest [64]. A sample theoretical interest curve is in the figure below.

![Interest Curve](image)

**FIGURE 11: SCHEMATIC OF A THEORETICAL INTEREST CURVE FOR A POSITIVE EXPERIENCE**

Interest curves have three distinct parts: the hook, lesser trials, and major trial. The hook is the first part of the experience that grabs the player and draws them in, immersing them into the experience. The lesser trials follow the hook and fluctuate in interest on a steady rise towards the third part of the curve, the major trial. During the lesser trials the immersion of the experience grows as the player invests in the experience. The major trial is the big finish, the final quest, the climax, where the player accomplishes the goal. Other work has theorized that interest curves are fractal and can be broken down from large experiences into game level interest curves all the way to game moment or game encounter interest curves [61, 65].
In order to maintain motivation during rehabilitation a game must keep interest and immersion for the player. The interest curve gives a visual way to model an experience again from the player’s perspective before they even turn it on.

### 3.4 Teaching Cycle

One of the most important aspects of any game is teaching the players how to play. Without knowledge of what they can do and how they do it the player is lost in the experience and will lose motivation and immersion. This section presents the idea of a teaching cycle to instruct players on the mechanics of the game [66].

Section 2.5 presented the phases of motor learning from a behavioral and physical therapy perspective. There are three phases: cognitive, associative, and autonomous. In game design and writing there is a theory around the number three and its importance. Three part structures are seen in many stories and it is thought that because three is the smallest number to create a pattern, that many jokes require three items to be funny [67]. In game design the rule of three also persists as design point to teach players a new mechanic. The mechanic is shown to the player, then they complete it in relative safety, then they are challenged with it.

Many critically acclaimed games have an observable method of teaching gameplay mechanics to players and it requires that two more phases be added to the phases of motor learning to create a teaching cycle, an introduction phase and a reuse phase. The theoretical teaching cycle for a game can be seen in Figure 12 on the following page.
The curve that is created by the values of player enjoyment in the teaching cycle also corresponds to the values needed to create a good interest curve for an experience. This relies on the fractal nature of interest curves and the phases of motor learning discussed earlier. The first phase in the teaching cycle is that the player is introduced to the skill or mechanic. This is usually done by cues or told to the player directly in some form of tutorial. The player is curious about how to use the skill so their anticipation leads to a medium level of enjoyment. The second step of the cycle is the learning phase. In this phase the player is given a very simple and obvious use of the skill that was just introduced to them. In a well-designed game this should not be challenging to the player and therefore has a lower enjoyment level. Testing is the third step of the learning cycle. This is a harder use of the skill than was first presented and because of this increase in difficulty it has a higher enjoyment level. The next step is the most important of the cycle, challenge. In this step the player needs to use their knowledge of the skill to ex-
tend its use beyond the obvious uses shown to them previously. The challenge phase of the cycle has the highest level of enjoyment because if the player succeeds they have mastered the skill. The final phase of the cycle is reuse, where the player has now mastered the skill or mechanic and is ready to incorporate it into their next challenge.

Teaching cycles appear in many games, one of the most pronounced examples is in the critically acclaimed game Portal [68]. In this game the player must pass a series of tests by simply getting to the end of each level. The puzzles involved in the tests use portals, a game mechanic that allows the player to pass from one wall to another instantly as if teleporting.

As the teaching cycle presented here suggests in the first level of the game the player is shown what a portal is by forcing them to walk through one to leave their starting point. During this first level the player does not have the ability to move or control the portals only pass through them. The figure below is a screen capture from the first level where the blue and orange portals first appear to the player.

![FIGURE 13: SCREEN CAPTURE OF THE PORTAL LOCATIONS IN THE FIRST PUZZLE IN PORTAL [68]](image-url)
In the second puzzle of Portal the player is now introduced to the idea that the portals are not permanent and can shift locations (Figure 14). To complete this puzzle the player must pass through a portal a minimum of five times, reinforcing the mechanic. The player learns that the crates used to trigger switches can pass through portals. The blue portal changes position at set time increments allowing the player access to three separate areas. Those three areas can be seen through the orange portal in the figure below. This puzzle correlates to the learning step of the learning cycle because the player still has no control over the portals and must react to them to complete the puzzle. Also there is no punishment for the player as they can just keep trying combinations of portals until they have found the solution.

FIGURE 14: SCREEN CAPTURE OF THE PORTAL LOCATIONS IN THE SECOND PUZZLE OF PORTAL [68]

The actual portal gun (which gives the player the ability to aim and shoot portals) appears in the third puzzle in the game. This corresponds to the testing phase of the teaching cycle. In this puzzle the player must again use timing to move through the portals with the goal of collecting
the portal gun. Once the player possesses the portal gun they now have control over the placement of the blue portal. Figure 15 is a screen capture of the puzzle once the player acquires the portal gun and places their first blue portal.

![Figure 15: Screen capture of the first player placed portal in the third puzzle of Portal [68]](image)

The next phase of the cycle, challenge, is elongated over the next nine puzzles because during this time several other mechanics are introduced and go through their own teaching cycles (switches and crates, light balls, moving platforms, and momentum). Once the player completes these puzzles they move into the reuse phase for portals because they are now able to create and place both the orange and blue portals. Thus other mechanics take their place and begin or continue their learning cycles.
Interest curves were presented in the previous section and they match with teaching cycles in an interesting way. Both of these phenomena have similar patterns when used in an effective game. Figure 16 (below) presents a well-designed interest curve overlaying a teaching cycle.

![Figure 16: Diagram of an Interest Curve and Teaching Cycle for a Well-Designed Experience](image)

The hook phase of the interest curve matches the introduction phase of the teaching cycle. The next two phases of the teaching cycle, learning and teaching, make up the lesser trials phase of the interest curve. The major trial of the interest curve is mirrored by the challenge phase of the teaching cycle. Teaching cycles were used to design the virtual scenes presented in Chapter 7.

### 3.5 Pleasure in Games

There is still no definitive work on how to make an experience enjoyable but several members of the field have attempted to create a framework or list of pleasures associated with playing a game. These can be useful to influence design as well as feedback for virtual environments for rehabilitation. Leblanc’s 8 pleasures can be seen on the next page [62, 69]:

---

[Image 173x427 to 451x586]
1. **Sensation**: Game as sense-pleasure

2. **Fantasy**: Game as make-believe

3. **Narrative**: Game as unfolding story

4. **Challenge**: Game as obstacle course

5. **Fellowship**: Game as social framework

6. **Discovery**: Game as uncharted territory

7. **Expression**: Game as soap box

8. **Submission**: Game as mindless pastime

These pleasures can be used to craft the game experience but also give insight into what kind of player would enjoy the experience (discussed further in Section 3.8). Sensation is a pleasure from intrinsic motivation, for the sake of the experience. Fantasy is pleasure from escapism, something discussed further in Section 3.7. Narrative is the pleasure in the dramatic unfolding of a sequence of events [61]. Challenge relates to learning of a skill, something explored further is several sections (2.4, 2.5, 3.4, 3.6, and 3.7). Fellowship as a pleasure pertains to interacting with others. Pleasure from discovery in games comes from a player’s curiosity. Expression and customization for the player in the game is a pleasure. Submission again pertains to escapism, to the player being immersed in the experience as it was designed. Schell comments that any list like this is not comprehensive and therefore he suggests other pleasures in games to be considered on the next page[61]:

---

40
- **Anticipation**: when the player knows a pleasure is coming, the wait can be pleasurable.
- **Delight in Another's Misfortune**: The player can feel this in competitive games for their opponents.
- **Gift Giving**: This pleasure comes from surprise but also if the player is the gift-giver it pertains to knowing the surprise for the recipient.
- **Humor**: players find pleasure in humorous things.
- **Possibility**: this is a pleasure of possible choices.
- **Pride in an Accomplishment**: as achievement systems in games continue to expand players can take more pride in past accomplishments.
- **Purification**: players can take pleasure in clearing the screen of goals or items, cleaning each level completely.
- **Surprise**: much like gift giving, players enjoy being surprised.
- **Thrill**: Schell uses a particularly good quote from roller coaster designers to describe this pleasure, “fear minus death equals fun”.
- **Triumph over Adversity**: Many games pit the player against impossible odds and portray their avatar as the underdog, because players take pleasure in being successful when chance is not in their favor.
- **Wonder**: players can take pleasure in awe and amazement.

Like so many topics in game design with further research more will be learned about pleasure in games. For the purposes of my work I wanted to incorporate these pleasures in to virtual scene design so that I could potentially increase motivation for the player.
3.6 FLOW IN GAMES

The psychological concept of flow was first proposed by Mihály Csíkszentmihályi [70]. Flow pertains to creating an optimal experience or order in consciousness. It is much more likely that flow will arise from a structured activity than from a random one [71]. Play is considered to be a flow activity, meaning that flow can come naturally from experiencing it [71]. Much like interest curves and teaching cycles flow can be applied to any experience and pertains to the player’s skill and the challenge of the experience. The figure below is a graphical representation of flow and how it is achieved [71].

![Flow Zone Graph]

**FIGURE 17: SCHEMATIC OF FLOW AND THE OPTIMAL EXPERIENCE [71]**

The theory of flow centers on the idea that if a player shifts between the edges of anxiety and boredom, by manipulation of the challenge of the experience and skill of the player, the player will have an optimal experience. However, there are other mental states possible on the flow spectrum that should be designed around. These states can be seen in Table 3 on the following page.
TABLE 3: MENTAL STATES OF FLOW

<table>
<thead>
<tr>
<th>Mental State</th>
<th>Challenge</th>
<th>Skill</th>
<th>Mental State</th>
<th>Challenge</th>
<th>Skill</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apathy</td>
<td>Low</td>
<td>Low</td>
<td>Boredom</td>
<td>Low</td>
<td>Moderate</td>
</tr>
<tr>
<td>Worry</td>
<td>Moderate</td>
<td>Low</td>
<td>Relaxation</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Anxiety</td>
<td>High</td>
<td>Low</td>
<td>Control</td>
<td>Moderate</td>
<td>High</td>
</tr>
<tr>
<td>Arousal</td>
<td>High</td>
<td>Moderate</td>
<td>Flow</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

The understanding of how a player achieves flow is important for game design but so are the characteristics of the Optimal Experience. Csíkszentmihályi outlines these characteristics and they are listed below [71]:

- Sense that the user's skills are adequate for the task
- Task is in a goal-directed, rule-bound system
- There is adequate instruction on the system
- There is adequate feedback to the user from the system
- Concentration during the task is intense
- The user's mind focuses only on the task
- Self-consciousness disappears
- Sense of time is distorted
- The task is so gratifying that it requires no outside motivation

Play is a flow activity but physical movement can also be linked to flow. The listing below includes the characteristics of physical motion and flow [71]:

a. Set a realistic goal and as many subgoals as possible
b. Find ways of measuring progress in relation to these goals
c. Keep concentration on the task being performed, keep making finer and finer distinctions in the challenges involved

d. To develop the skills necessary to interact with the opportunities available

e. To keep raising the stakes if the activity gets boring

This list bears a resemblance to information presented in several other sections of this document (sections 2.5, 2.6, 3.3, 3.4, 3.5, and 3.7). These elements should be implemented into virtual environments for rehabilitation to try to induce flow in a player and raise their motivation.

Specific research has been done into using flow in games. Sweetser and Wyeth present a model for evaluating player enjoyment in games called GameFlow [72]. In this model the authors outline criteria for GameFlow, which has been reproduced in the table below.

<table>
<thead>
<tr>
<th>TABLE 4: CRITERIA FOR GAMEFLOW [72]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Concentration</strong></td>
</tr>
<tr>
<td>▪ games should provide a lot of stimuli from different sources</td>
</tr>
<tr>
<td>▪ games must provide stimuli that are worth attending to</td>
</tr>
<tr>
<td>▪ games should quickly grab the players’ attention and maintain their focus throughout the game</td>
</tr>
<tr>
<td>▪ players shouldn’t be burdened with tasks that don’t feel important</td>
</tr>
<tr>
<td>▪ games should have a high workload, while still being appropriate for the players’ perceptual, cognitive, and memory limits</td>
</tr>
<tr>
<td>▪ players should not be distracted from tasks that they want or need to concentrate on</td>
</tr>
<tr>
<td><strong>Challenge</strong></td>
</tr>
<tr>
<td>▪ challenges in games must match the players’ skill levels</td>
</tr>
<tr>
<td>▪ games should provide different levels of challenge for different players</td>
</tr>
<tr>
<td>▪ the level of challenge should increase as the players progress through the game and increases their skill level</td>
</tr>
<tr>
<td>▪ games should provide new challenges at an appropriate pace</td>
</tr>
<tr>
<td><strong>Player Skills</strong></td>
</tr>
<tr>
<td>▪ players should be able to start playing the game without reading the manual</td>
</tr>
<tr>
<td>▪ learning the game should not be boring, but be part of the fun</td>
</tr>
<tr>
<td>▪ games should include online help so players don’t need to exit the game</td>
</tr>
<tr>
<td>▪ players should be taught to play the game through tutorials or initial levels that feel like</td>
</tr>
</tbody>
</table>
- playing the game
  - games should increase the players’ skills at an appropriate pace as they progress through the game
  - players should be rewarded appropriately for their effort and skill development
  - game interfaces and mechanics should be easy to learn and use

<table>
<thead>
<tr>
<th>Control</th>
<th>Players should feel a sense of control over their actions in the game</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>▪ players should feel a sense of control over their characters or units and their movements and interactions in the game world</td>
</tr>
<tr>
<td></td>
<td>▪ players should feel a sense of control over the game interface and input devices</td>
</tr>
<tr>
<td></td>
<td>▪ players should feel a sense of control over the game shell (starting, stopping, saving, etc.)</td>
</tr>
<tr>
<td></td>
<td>▪ players should not be able to make errors that are detrimental to the game and should be supported in recovering from errors</td>
</tr>
<tr>
<td></td>
<td>▪ players should feel a sense of control and impact onto the game world (like their actions matter and they are shaping the game world)</td>
</tr>
<tr>
<td></td>
<td>▪ players should feel a sense of control over the actions that they take and the strategies that they use and that they are free to play the game the way that they want (not simply discovering actions and strategies planned by the game developers)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clear Goals</th>
<th>Games should provide the player with clear goals at appropriate times</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>▪ overriding goals should be clear and presented early</td>
</tr>
<tr>
<td></td>
<td>▪ intermediate goals should be clear and presented at appropriate times</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feedback</th>
<th>Players must receive appropriate feedback at appropriate times</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>▪ players should receive feedback on progress toward their goals</td>
</tr>
<tr>
<td></td>
<td>▪ players should receive immediate feedback on their actions</td>
</tr>
<tr>
<td></td>
<td>▪ players should always know their status or score</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Immersion</th>
<th>Players should experience deep but effortless involvement in the game</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>▪ players should become less aware of their surroundings</td>
</tr>
<tr>
<td></td>
<td>▪ players should become less self-aware and less worried about everyday life or self</td>
</tr>
<tr>
<td></td>
<td>▪ players should experience an altered sense of time</td>
</tr>
<tr>
<td></td>
<td>▪ players should feel emotionally involved in the game</td>
</tr>
<tr>
<td></td>
<td>▪ players should feel viscerally involved in the game</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Social Interaction</th>
<th>Games should support and create opportunities for social interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>▪ games should support competition and cooperation between players</td>
</tr>
<tr>
<td></td>
<td>▪ games should support social interaction between players (chat, etc.)</td>
</tr>
<tr>
<td></td>
<td>▪ games should support social communities inside and outside the game</td>
</tr>
</tbody>
</table>

Another model was introduced by Cowley et al. to attempt to understand flow in games [73].

The model presented is called the USE model, short for user-system-experience. The goal of this model is to outline the relationship between flow and gameplay. The model is represented by the figure on the following page [73].
For a player to experience flow in an activity a balance must be achieved between the external complexity of the system and the internal model that a player develops of that system. A player attains an optimal internal model of the system through learning and practice. The attainment of a strong correlation between internal and external complexity is dependent both on the capability and willingness of the player to master the system and on the quality of system design to meet the requirements and capability of the player. Games have an ordered environment, with opportunities for action in the form of controls [73]. Mastery of a structured challenge within a virtual world requires the dedication to learn and apply skills, but as it is a virtual world, the level of collateral investment is entirely discretionary to the world’s designer.

One of the methods in design that can be used to make sure the player remains in the flow zone is to have dynamic or adaptive game mechanics. Carefully chosen actions that a player makes during gameplay can be used to determine their skill level accurately [74]. Adaptive
game mechanics can increase success and progression rate for players of all competence levels [74].

While more research into flow is required for specifics of flow in rehabilitation, the underlying theories and research provide strategies that can be integrated into the design of virtual environments for motor rehabilitation.

3.7 Game Feel

Another way to look at design for virtual environments is to implement and analyze the design by its game feel. Game feel was introduced by Swink and relies on much of the theory behind interest curves (Section 3.3), motor control (Section 2.4), motor learning (Section 2.5), and human information processing (Section 2.3). Swink proposes that game feel means different things to different game designers, thus is cannot be defined [31]. Game feel is composed of a combination of three different elements: real-time control, virtual (simulated) space, and polish [31].

Real-time control relies on human information processing in a closed loop. Swink explains the interaction between the game and the player as two closed loops that can interconnect to create a larger loop or as a conversation [75]. If a game has real-time control it must respond to the player constantly and timely. Virtual space, as mentioned in section 3.2 every game must have a world. It provides context to the real-time control and allows the player to make comparisons between their avatar and the world for meaningful interaction [31]. This includes the response between the world and the avatar as well as immersion for the player in the experience. Polish refers to elements of a game that enhance the interaction without changing the
underlying simulation [31]. Elements of polish are added to both the real-time control and the virtual space to give higher immersion and enjoyment to the interactions. This includes elements like sound effects, animations, and particle effects. The Venn diagram below presents the components of game feel and numbers the areas of overlap in the types of game feel [31].

![Venn diagram of the components of game feel](image)

**FIGURE 19: VENN DIAGRAM OF THE COMPONENTS OF GAME FEEL [31]**

The different types of Game Feel [31] are:

1. **True game feel**: games that have all three components
2. **Raw game feel**: games composed of virtual space and real-time control but lacking polish
3. **The pure aesthetic sensation of control**: games that lack virtual space but have polished real-time control
4. **Physical simulation**: games here either do not occur or respond in real-time
5. **Naked real-time control**: games here lack virtual space and polish and usually not released until they acquire at least one other component
6. **Naked virtual space**: as with 5 these games in this state are usually not completed
7. **Naked polish**: as with the two previous types, this type is for games in progress
The power of game feel to help design comes from the experiences a player has with true game feel. These experiences can lead to a flow experience with the game and cause intrinsic motivation for the game. Swink presents the most common experiences that are listed below.

The five most common experiences of game feel [31]:

- **The aesthetic sensation of control**: this experience is the sense of puppetry or control over the avatar and/or the virtual space of a game.

- **The pleasure of learning, practicing and mastering a skill**: as the learning cycle or the phases of motor learning or intrinsic motivation show, humans enjoy learning and mastery.

- **Extension of the senses**: this experience is the blending of the physical interface (input device) with the human controlling it. It corresponds to the autonomous phase of motor learning.

- **Extension of identity**: the extension of the senses can proceed further and allow players to have emotional ties to an experience

- **Interaction with a unique physical reality within the game**: escapism is another human trait and source of pleasure. Games allow players to be more powerful or different people, impossible in the real world.

Game feel is a powerful tool for game design because it is focused on player experience. Games that exhibit true game feel can enhance the intrinsic motivation of their players, a goal of virtual environments for rehabilitation.
3.8 Player Types

There has been recent research into a player’s personality within a game. This relates to their personal goals and what they find fun or enjoyable within a game. Attempting to make a virtual environment patient specific requires calibration from both the hardware and the software side, so it is important to understand a player’s type or personality and how that can be used to enhance their experience within the game.

Bartle’s research into players types based on multi-user dungeons (MUDs) has long been held up as a resource for determining player type [76]. He describes four different player “suits” (achievers, explorers, socializers, and killers) and how and why they play [76]. A list of the suits and their attributes can be seen below.

♦ Killers: These types of players use the game to cause distress to other players. They enjoy head-to-head competition and want to be the best, by destroying their competition.

♦ Achievers: These players either impose goals upon themselves or use in-game goals and strive vigorously to achieve them. They are interested in accumulation of in-game items and in being the best, by being the best player they can.

♥ Socializers: Players in this category are more interested in (positive) interaction with others than in-game goals or being the best.

♦ Explorers: As the name implies, these players want to learn as much as they can about the virtual world. They want to push the envelope on what the player can do within the game.
While these player suits are useful for multiplayer games they do not address what goals are important to all players in a quantitative sense. Weber and Shaw proposed a social cognitive theory based model for player types [77]. Within this model there are six player types as seen in the table below [77].

<table>
<thead>
<tr>
<th>Player Type</th>
<th>Characteristics</th>
</tr>
</thead>
</table>
| Hedonist    | - Strong enjoyment incentives  
- Self-improvement  
- Approachability |
| Competitor  | - Self-improvement  
- Competition with others  
- Positive mood |
| Organizer   | - Self-improvement  
- Positive mood  
- Approachability  
- Self-awareness  
- Self-efficacy  
- Judgment |
| Rebel       | - Competition with others  
- Flexibility  
- Social judgment  
- Winning above all else |
| Team Player | - Self-improvement  
- Approachability  
- Positive mood  
- Self-efficacy  
- Social judgment  
- Competition with others |
| Socializer  | - Very social  
- Self-improvement  
- Positive mood  
- Approachability  
- Not interested in competition |

The six player types described in this work all have goals that lead to high enjoyment in the experience. In the study the players were categorized using a survey before playing off the shelf games. When used on custom games this taxonomy might not work for certain player types.
However, this information was used to help design the goals and interfaces of the work presented in Chapter 7.

3.9 **Usability Heuristics**

For many aspects of good game design (immersion, flow, and game feel) a well-designed graphical user interface (GUI) is required. There has been extensive research into what makes a good GUI ([78], [79], and [80]) and presented below are the usability heuristics of Nielsen [81, 82].

- **Visibility of system status:** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

- **Match between system and the real world:** The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

- **User control and freedom:** Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

- **Consistency and standards:** Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

- **Error prevention:** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
• **Recognition rather than recall**: Minimize the user’s memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

• **Flexibility and efficiency of use**: Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

• **Aesthetic and minimalist design**: Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

• **Help users recognize, diagnose, and recover from errors**: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

• **Help and documentation**: Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user’s task, list concrete steps to be carried out, and not be too large.

These heuristics or rules of thumb become more important with users who have suffered a stroke because they may have cognitive deficits. In Chapter 7 these heuristics are used to create the user interfaces design for the virtual scenes based on the player type discussed in Section 3.8.
3.10 CURRENT VIDEO GAME MOTION AND GESTURE HARDWARE - REVIEW

There are motion and gesture devices currently on the market for consumers and several of these devices have been used in stroke rehabilitation. In the following section I will outline the major devices and their use in rehabilitation.

3.10.1 NINTENDO WII

The Nintendo Wii was released in the fall of 2006 as a personal video game console [83]. The Wii controller has two components: a remote and nunchuk. The Wii remote is normally held in the player’s dominant hand. If the player is using both the nunchuk and the remote, the remote is usually held in the right hand. The nunchuk plugs into the remote and is usually held in the left hand and not a requirement for all Wii games. A figure of the Wii remote and nunchuk can be seen below.

![Image of the Nintendo Wii remote and nunchuk](image.png)

**FIGURE 20: IMAGE OF THE NINTENDO WII REMOTE AND NUN' CHUCK [83]**

The Wii console connects directly to a television (either via component video and audio or some kind of converter) and the Wii controllers communicate with the console using Bluetooth...
wireless technology. There is a large library of Wii games, however at this time there is no licensed software development kit available to the public.

The Wii was used in a case report for rehabilitation of an adolescent with Cerebral Palsy [84]. Deutsch et al. chose to use the Wii because of the age of the subject as well as the low cost of the system. They also found that using the off the shelf software was sufficient in aiding rehabilitation.

The largest drawback of the Wii system is lack of a readily available software development kit. However, since the remote uses Bluetooth for communication it is possible to use it with a Bluetooth capable computer but this is far from ideal. Another drawback of the Wii is that the user is required to grip the remote therefore making exercise of the fingers difficult or impossible. These finger controls on the remote and nun’ chuck require a high level of finger dexterity, a significant challenge for many patients with stroke.

3.10.2 MICROSOFT KINECT

The Microsoft Kinect was released in the US in November 2010 as a new input device for the Xbox 360 console. The technology used in the Kinect is based around four possible methods of interaction: motion sensor, skeletal tracking, facial recognition, and voice recognition. A picture of the Kinect can be seen on the following page.
Within the sensor bar there are several sensor systems including an infrared camera, a microphone array, a tilt servo (to change pitch), and a visual depth sensor [86]. Microsoft has released a software development kit and tools to create Kinect applications.

Due to the low cost (under $200 depending on included software) and sensors available the Kinect has been piloted for use as a rehabilitation device [87] and [88]. Lange and Rizzo et al. have been using the Kinect with several populations [89] and [90]. Suma, Lange, and Rizzo et al. developed a middleware that can be used to connect a virtual environment for rehabilitation with the Kinect called Faast (Flexible Action and Articulated Skeleton Toolkit) [91]. The middleware can also allow use of the Kinect with off-the-shelf games.

The Kinect is a very new technology and more research is needed in its use as well as assessing its capabilities. The device does have some flaws for rehabilitation including a low bandwidth (under 60Hz), no way to sense fine finger movement, lack of tactile feedback, and a high learning curve due to the recent release of accompanying software like Faast and the SDK from Microsoft.
3.10.3 PlayStation Move

The PlayStation Move is an input device for the PlayStation 3 that is used in conjunction with the PlayStation Eye Camera. The Move was released in September 2010 in the US. In many ways the interaction a player has with the Move is very similar to the controller of the Nintendo Wii (section 3.10.1). The device includes a remote for each hand that communicates with the PlayStation 3 via Bluetooth. At the end of the main Move wand is an RGB LED that can change color based on game functionality and needs. This sphere is tracked by the camera based upon its color, allowing up to four moves to be tracked at any time. The in-plane position of the orb is calculated from the EyeToy image and then the relative size of the orb is used for depth. The figure below displays the left and right hand controllers of the Move.

FIGURE 22: IMAGE OF THE PLAYSTATION MOVE FOR THE PLAYSTATION 3 [92]

Researchers have used the PlayStation Eye camera in motor rehabilitation research since its release [93-95]. The Move has been explored as well but due to its recent release it is still a very active area of research [96]. One of the most notable developments from November 2011 is a patent application from Sony including biometric sensors (i.e. heart rate monitor) in their input
devices including the Move [97]. This is evidence in a larger trend that input devices will include more sensors so that more player data can be used in the real-time dynamic experience of a game.

3.10.4 NOVINT FALCON

The Falcon device from Novint is a computer input device that is designed for playing video games. The Falcon has a 4” x 4” x 4” workspace and a two pounds force feedback capability. It is a 3 degree of freedom device in terms of movement, however by using different grips more degrees of freedom can be added by buttons or dials [98]. A figure of the device can be seen below.

![Figure 23: Image of the Novint Falcon](image)

The Falcon interfaces with a PC using a USB 2.0 port. The Falcon has several games already available for it as well as driver software to play other games. The Falcon software development kit is available for download [98].

The device has been chosen for use in rehabilitation research in the past because of its simple programming interface and relatively low cost. One such study was conducted in 2008 in the UK
The researchers picked the Novint Falcon device because of low cost, three degrees of freedom, force feedback, and other factors. The study was setup to examine the rehabilitation of the upper extremity of stroke patients. The researchers used some of the built in or off the shelf virtual environments in conjunction with the Falcon. Subjects who had experienced a stroke at least six months prior were split into two random groups four. One of these groups performed tasks in virtual environments using the Falcon and the other group did virtual reality “relaxation” exercises that did not include using the software with the Falcon. The researchers found that the group that used the Falcon in conjunction with the virtual environment showed improvement after only six of the twelve sessions [100].

The Falcon device has many advantages but because of its design only having three degree of freedom and not having a system to sense finger kinematics it is not optimal for motor rehabilitation of the hand and wrist.

3.10.5 PHANTOM DEVICES

Sensible Technologies has a product line of haptic input devices designed to gather input and give feedback to the user’s fingers, hand, and arm. The devices most suited to be used in rehabilitation VR systems are the six degree of freedom PHANTOMs. Users grasp the phantom stylus (device the shape of a pencil or pen) in either hand and have control of x-direction, y-direction, z-direction, roll, pitch, and yaw. A figure of two SensAble PHANTOM devices can be seen below [101].
FIGURE 24: IMAGE OF TWO SENSABLE PHANTOM DEVICES [101]

The PHANTOM six degree of freedom device is interfaced with the user’s PC via a parallel, Firewire, or USB port. The device also comes bundled with several software demos and a software development kit specific to the inputs and limitations of the device. SensAble devices have been used in rehabilitation studies but their main use is as input devices for computers [101].

The PHANTOM was used as a haptic handwriting aid for training and rehabilitation at Deakin University in Australia [102]. In this study the researchers used the Omni PHANTOM device as a handwriting aid to assist in rehabilitation post stroke. The researchers used the SensAble software development kit along with their own code to create the virtual environments. In house testing was performed on the VR system and the use of the Omni showed great promise as a rehabilitation tool [102].

The main advantage the PHANTOM devices have over other devices presented is force feedback. The main disadvantage of is that the stylus must be grasped at all times so the user cannot exercise their fingers using the device.
3.11 SUMMARY

The field of game design is new and expanding very quickly. Many of the topics presented are borrowed or modified from other fields because of these growing pains. However, there is still pertinent information that is usable in the design of patient specific virtual environments in the field. This is evidenced by the competing models for game components and the murky definitions of what a game is.

The theories of game feel, flow, interest curves, and teaching cycles all relate to how players learn in games and the impact of that learning on the experience. This is one of the reasons serious games have become such a necessary topic of research as the power of teaching via games is still mostly undiscovered.

Pleasure in games and how different players react and enjoy different experiences is another way in which customization of virtual environments can be achieved. As more research becomes available on the psychology of players and their virtual personalities, designers will be able to tailor experiences to players for more motivation and enjoyment.

The growing size of the games industry and its role in recreational entertainment is leading to new devices with innovative and advanced input options. The Kinect, Move, and Wii are consumer level devices with sensor fidelity that is approaching that of research devices. This trend looks to continue and allow more empirical data for designers to use to further customize game experiences to players.
CHAPTER 4: VR SYSTEMS SOFTWARE COMMUNICATIONS MIDDLEWARE

SCREEN CAPTURE OF THE DEVICE AGNOSTIC VIRTUAL ENVIRONMENT SETUP MENU
4.1 INTRODUCTION
Prototype research devices pose several challenges to engineers and designers. One of these challenges is how to have communication between the sensors on the physical prototype and the prototype software. Devices that were created within the body of this work either relied on National Instruments LabVIEW software or a microcontroller with communication via a USB and serial connection. For efficiency a single calibration and visualization software was designed with capabilities for both UDP and serial communications. This work began in June of 2008 building upon the first version of the Hand Enhancement Robotic Rehabilitation Interface (HER-RI) (see Section 4.6.3).

4.2 BACKGROUND INFORMATION
The following sections contain information on the software used to create the components of the middleware solution. This includes information on Panda 3D, LabVIEW, and software communication protocols TCP and UDP.

4.2.1 PANDA 3D
The Panda 3D game engine (Panda) was created by Disney and is used by Carnegie Mellon University. It is open source and freely available, any software created with it can be sold without giving royalties or credit to the creators of Panda 3D. The engine uses Python as the scripting language. Panda3D also has input methods for the direct input of Head Mounted Displays (HMD) and VR trackers.

The Panda 3D game engine was used for several versions of the glove device VR software as well as the device calibration and testing software. The greatest advantage of Panda 3D is prototyping speed for a virtual environment. The greatest drawback is the performance of the en-
engine. Because the VR systems in general run multiple software programs during a rehabilitation session it is necessary to have an engine that has optimum performance.

The main systems within Panda that were used are the event system, collision detection, loader, timer, and user interface text. The event system within Panda is used to control mechanics and events within the virtual scene. The collision detection system triggers events when models collide with each other or with other collision geometry. The Panda loader uses .egg files to load in 3D models and other assets. The timer in Panda is based off the Windows system timer and it is used to time events and mechanics in the virtual scenes. The user interface text system is used to display the score and instructions for the user without issues of window size or resolution.

4.2.2 LabVIEW
LabVIEW (short for Laboratory Virtual Instrumentation Engineering Workbench) is a sensor data acquisition and device control prototyping software. The original version was released in 1986 for the Apple Macintosh. The current version, LabVIEW 8.6.1, was released in February of 2009.

LabVIEW uses a visual programming language to create virtual instruments (VIs) that can be used to control devices and acquire data from sensors. This programming style is built upon creating block diagrams for the information flow of the VI the LabVIEW VIs discussed in this document uses a while loop to iterate a block diagrams functionality at a frequency that is determined by the hardware and sensor speed.

Many devices created in the Biomedical Mechatronics Laboratory (BML) at Northeastern University [103] use LabVIEW as a middleware between the sensors and hardware and the VR
software. All data acquisition was also done in LabVIEW for these devices and not done in the VR software. The main use of LabVIEW in this project is to create a UDP SubVI as described in Section 4.3.

4.2.3 COMPUTER COMMUNICATION
The two major types of computer communication are transmission control protocol (TCP) and user datagram protocol (UDP), both members of the internet protocol suite along with internet protocol (IP).

TCP was first introduced by Cerf and Khan in 1974 as a method for packet switching between nodes [104]. A TCP segment consists of a header and data component with the header consisting of 10 mandatory fields and an optional extension field. The header is verified for each data segment making the data very reliable at the cost of speed and ordering the packets so that packet loss can be confirmed. In order to create a socket connection three packets are required [105].

UDP was created in 1980 by Reed as a way to augment current communication protocols. In contrast to TCP, UDP does not have a verified header or handshake when a socket is connected. Thus there is no method to verify the data or order of packets, giving UDP less overhead. However, these differences allow UDP communications to be much faster than TCP communications. The middleware described here required fast communication and because of a highly controlled network (either communication on the local host or a LAN) UDP was chosen for communication.
4.3 **LabVIEW SubVI for UDP Communication**

One of the goals of the research project is to create a software platform that can use any device created in the BML. Most of the devices are controlled by using LabVIEW. It then follows that a LabVIEW SubVI capable of cross-platform communication is needed to simplify the software implementation for current and future devices. The figure below is an example SubVI for basic UDP communication.

![Sample Block Diagram for a LabVIEW SubVI for UDP Communication](image)

**Figure 25: Sample Block Diagram for a LabVIEW SubVI for UDP Communication**

For the communication to be setup correctly some information must be chosen by the user.

The first piece of information needed is the port that will be used as a socket for communication. This is usually a four digit number and some ports are reserved for common programs such as internet browsers and for operations on the computer (ports numbered 0-1024) so a higher numbered port should be used. Once the socket is set up for receiving the data the transmission of the data may begin. Data for UDP communication is sent as a string and sent continuously using a while loop. The string format of the data is most easily used if sent as a comma separated list, as seen below.

**Equation 3: UDP Data String Format**

\[num_a, num_d, num_s, analog_1, ... analog_{num_a}, digital_1, ... digital_{num_d}, string_1, ... string_{num_s}\]
In the format above the number of different signals is sent allowing the virtual environment (described in Section 4.5) to split and use the data regardless of the quantity or type of signal. The values $num_a$, $num_d$, and $num_s$ correspond to the quantity of analog, digital, and string signals respectively. After the quantities the actual data is sent based on the type of data. Depending on the device that used the LabVIEW SubVI modifications were made to the data string format including adding titles for data or specifically grouped data, such as what is used for IMUs. This is discussed in further detail in section 4.6.

4.4 SERIAL COMMUNICATION ALGORITHM

Several devices in the Biomedical Mechatronics Laboratory use microcontrollers instead of LabView for data acquisition. This required a different solution for communication than UDP, so serial communication was chosen and implemented for those devices.

Serial communication from the microcontroller (in all current devices they are based on the Arduino platform, explained further in section 6.3.2) is very straightforward. Arduino programs must consist of two functions: setup and loop. During the setup function the serial communication is started and a baud rate is chosen. Then in the loop function the data is written to the serial (COM) port on the computer as seen in the code fragment below.

**CODE FRAGMENT 1: EXAMPLES OF ARDUINO SERIAL COMMUNICATION**

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println(millis());
}
```
In the code fragment above the serial communication is initialized with a baud rate of 9600 on line 2. On line 6 the current time elapsed since the program began (calculated by the millis function) is then sent to the COM port. Serial communication data is sent as a series of bytes, unlike the strings required for UDP. However, the Arduino software will do the conversion for the data automatically so that a data format like the one presented in Equation 3 can be used for consistency. The data for both UDP and serial communication was sent to the Device Agnostic Virtual Environment which is presented in the following section.

4.5 DEVICE AGNOSTIC VIRTUAL ENVIRONMENT

The Device Agnostic Virtual Environment (DAVE) was designed in the summer of 2010 and underwent improvements until spring of 2012. The software was written in Python using the Panda 3D game engine. The goal of the software was to create a virtual environment that could visualize the data output of any prototype robotic rehabilitation device to streamline testing and design. The figure below is a screen capture of the DAVE setup menu.

**FIGURE 26: SCREEN CAPTURE OF THE DEVICE AGNOSTIC VIRTUAL ENVIRONMENT SETUP MENU**
DAVE was designed to allow for both UDP and serial communication (using pySerial) either separately or in unison. Before the visualizations begin the user can modify the default values for serial port, baud rate, UDP port, UDP IP, and choose to print the raw data to the program console. The communication can be turned on or off at will without resetting the program. Once setup is complete the panel is removed and can be accessed by button in upper right corner of the screen. Simple virtual representations of different device axes can be seen in the figure below.

**Figure 27: Illustration of simple geometric representations of device data**

The devices that use or have used DAVE have the same types of sensor output, namely linear, rotary, and digital signals. Linear and rotary data can be collected by many types of sensors including infrared sensors, potentiometers, IMUs, and rate gyroscopes. Digital sensors that are common on these devices include buttons, Hall Effect sensors, heart rate monitors, and switches. To accommodate these signals DAVE has several different visualization options including cubes for orientation of IMUs, scaling bars for linear signals, and color changing blocks for digital signals. These visualizations can be seen in Figure 28 on the following page.
The DAVE software uses tasks in Panda 3D to loop over the data and create the visualizations. Tasks are a special kind of function that is repeated based on the type of task. For the DAVE software the main task is called every frame (60 or 75 times per second). Python also allows for simple file I/O so DAVE allows saving data to text or spreadsheet files. Figure 29 below displays the data flow within the tasks for the DAVE software.

The DAVE software supports 12 digital inputs, 12 analog inputs, 2 IMU inputs, and 8 string inputs. It has been used in four projects to date and more information about the implementation is in the next section.
4.6 IMPLEMENTATION INTO VR SYSTEMS FOR REHABILITATION

DAVE has been implemented into four systems to date. This includes use of the LabVIEW SubVI and/or the serial algorithm. The following sections outline the implementation and specifics of each of the systems.

4.6.1 VIRTUAL REALITY AUGMENTED CYCLING KIT (VRACK)

The VRack is a low cost sensitized kit that can be fitted onto any stationary bicycle in order to measure symmetry between a patient’s affected and unaffected sides post stroke. For more information on the VRack see Appendix Section A.1. The VRack was the second project that used the DAVE software. Specifically the VRack used the LabVIEW SubVI to send data to a bike simulation game, a screen capture of the SubVI can be seen below.

FIGURE 30: SCREEN CAPTURE OF THE VRACK SUBVI FOR UDP COMMUNICATION TO GAME
The firmware of the VRack cycles at 100Hz and receives data at 1kHz which is averaged and decimated before being filtered and modified. Once the data has been conditioned it is sent to the game via UDP. The data was sent to the game at 50Hz in the format of a comma separated string. The data was then used to control the bicycle rider in the game and control the pace rider by using the heart rate data and comparing it to a target rate.

4.6.2 Angle Tracking and Location At-home System for Bi-Manual Rehabilitation (ATLAS)

The first two versions of the ATLAS (section 6.3.4) used LabVIEW to send data to the first versions of the DAVE software. The current version of the ATLAS (section 6.3.5) uses serial communication with the latest version of the DAVE software. The figure below is a screen capture of the calibration function of DAVE used for the characterization testing discussed further in sections 5.5, 5.6, and 6.5.

![Screen Capture of the DAVE Calibration Testing](image)

**FIGURE 31: SCREEN CAPTURE OF THE DAVE CALIBRATION TESTING**
Data was entered manually for the goniometer reading and then data from the bend sensors was collected via serial communication. The main task of the calibration testing can be seen in the code fragment below.

**CODE FRAGMENT 2: MAIN TASK IN THE ATLAS CALIBRATION SOFTWARE**

```python
1. def updateTask(self, task):
2.     self.dataListS=self.sd.getDataList()
3.     if self.ct!=None:
4.         if self.ct.testActive:
5.             self.ct.calibFile.write(self.sd.dataString)
6.             self.ct.dataCount=self.ct.dataCount+1
7.         if self.ct.dataCount>=500:
8.             self.ct.dataCount=0
9.             self.ct.testActive=False
10.            self.dataSound.play()
11.            self.ct.nextTest()
12.            print "Test Complete"
13.     elif self.ct.imuTestActive:
14.         self.ct.calibFile.write(self.sd.dataString)
15.     if self.viewMode<2 and self.numIMU>0:
16.         self.updateIMU()
17.     elif self.viewMode==2 and self.numAna>0:
18.         self.updateAnalog()
19.     elif self.viewMode==3 and self.numDig>0:
20.         self.updateDigital()
21.     else:
22.         print "ERROR: Invalid view mode"
23.     return Task.cont
```

The main task in the above code fragment was used as the main loop of the program. The first step of the task is to read the data from the IMU (line 2). The next lines (3-14) dictate the logic of the calibration testing and how the program interprets the protocol and runs the specific tests. Lines 15-22 of the task adjust the view mode of the device ambiguous visualization for any device. The last step of the algorithm (line 23) signifies that this Panda task will be run every frame of the visualization (between 60-120Hz).

The pySerial module contains a read line function that will continue to read from the serial buffer until a new line ("\n") character is found. This function is simple to use but can have a
large impact on efficiency. Its use in this algorithm did not impact the 100Hz that was the target frequency for the software.

4.6.3 HAND ENHANCEMENT ROBOTIC REHABILITATION INTERFACE (HERRI)
The HERRI is a tabletop system for hand and wrist rehabilitation. Sections A.2 and 7.2.3 contain more information about the project. The HERRI was the first system to use the LabVIEW SubVI and DAVE software. Data for the location of the player in the game was sent via UDP to a Panda 3D game. The game then calculated scoring and timing based on the location data from the system.

4.6.4 ELECTRO-ACTIVE POLYMER DEVICE FOR PULSE AND BLOOD PRESSURE
This system was used for telemedicine and communication of health information over the internet. For more information about this system see [9]. This system used the LabVIEW SubVI from the VRack to send blood pressure data over the internet so that the pulse could be simulated using an electro-active polymer actuator.

4.7 FUTURE WORK
There are three possible improvements to the components of the DAVE system. The first is to implement a configuration file, the second is to create a web app version of DAVE, and the third is to create a mobile version.

One improvement to the DAVE system would be the implementation of a configuration file for the communication variables. This would be a text or XML file that would be read when the program was started so that the default values for variables such as baud rate, IP, serial port, and UDP port can be modified and set outside the program.
Another improvement for the DAVE system is a web app version of the DAVE software. This will allow for users to have access to DAVE on any computer with any operating system without needing to install the program and instead installing a plug-in for their browser.

The last improvement is to create a mobile version of DAVE for Android, Windows, and iOS phones and devices. This would allow devices to interface with DAVE without a PC and allows the devices to be Bluetooth compatible.

4.8 SUMMARY

Communication between custom rehabilitation devices and virtual environments can be accomplished by serial or UDP communication. It is advantageous to standardize the communication in order to reduce the time needed in prototyping. This chapter introduced the two communication methods as well as the DAVE software and how it has been implemented into four different prototype systems.
CHAPTER 5: INERTIAL MEASUREMENT UNITS IN REHABILITATION DEVICES

CAD RENDERING OF THE INERTIAL MEASUREMENT UNIT TESTBED V2
5.1 INTRODUCTION

Inertial sensors have become ubiquitous in devices today. Many smart phones, cameras, laptops, and other electronics include one. They are a low cost solution to potentially gather kinematic data during rehabilitation. In this chapter two different systems with integrated inertial sensors are discussed, the VRack and the ATLAS. This work was completed between September 2008 and January 2012.

5.2 BACKGROUND INFORMATION

In conjunction with the material in chapter 2 this section presents other background information that is pertinent to the work presented here. This includes information on the type of sensors used, their use in other rehabilitation systems, Kalman filtering and direction cosine matrices.

5.2.1 INERTIAL SENSOR COMPONENTS

Inertial Measurement Units (IMUs) consist of single or multi-axial accelerometers, gyroscopes, and possibly magnetometers. Due to microelectromechanical (MEM) manufacturing methods IMUs have become smaller and less expensive. This had led them to be integrated into everything from cars and laptops to iPods and smart phones.

Accelerometers are used to measure proper acceleration meaning that when in free fall they register zero and when stationary and parallel to gravity they read 1g. Accelerometers can be used to measure shock or tilt when the sensor is static. The sensors suffer when in a dynamic state because they read both the acceleration due to gravity and the component of the acceleration in the axis. The figure on the following page presents the basic equations behind angle calculations with accelerometers.
Gyrosopes are rate sensors, meaning that they read the velocity around an axis. Rate sensors have the disadvantage of being relative and requiring an absolute sensor like an accelerometer or magnetometer to provide context to the signal.

The main reason magnetometers are included in IMUs is to measure yaw (also called heading). Accelerometers cannot calculate yaw because changes in yaw do not result in a change to the gravity component of the force. Therefore if yaw is needed a magnetometer is used to calculate...
the angle of true north versus the angle of the IMU. This requires that there is no local magnetic field that could mask true north and can be a sensor prone to drift.

5.2.2 IMUs Used on the VRack and ATLAS
Two different IMUs were used in this research: the 5DOF IMU and the 9DOF Razor IMU. Both of these IMUs have a small form factor and low cost which made them the optimal choices for the projects. A photograph of the IMUs can be seen below.

FIGURE 34: IMAGE OF THE 5DOF IMU USED IN THE VRACK AND 9DOF RAZOR USED IN THE ATLAS

The VRack used the 5DOF IMU which contains a 3-axis accelerometer and a 2-axis gyroscope. The accelerometer on the IMU is the ADXL335 which has a range of ±3.6g. The IDG-500 is the gyroscope on the IMU and it has a maximum velocity of 500°/sec as well as an amplified signal that has a maximum velocity of 110°/sec with a 4.5x higher sensitivity. The IMU has a low voltage and current draw.

The IMU used in the ATLAS project was the 9DOF Razor IMU. As the name implies this IMU has a three-axis accelerometer, three-axis gyroscope, and three-axis magnetometer. Unlike the IMU used in the VRack, this IMU had an onboard microcontroller (ATmega328) that allowed for filtering and data acquisition to be done on the sensor board. The figure above (right) shows
the IMU. The accelerometer on this 9DOF IMU is the ADXL345 with a ±16g range. The gyroscope on the IMU is ITG-3200, as mentioned it is three-axis with ±2000°/sec range. The magnetometer integrated circuit on the IMU is the HMC5883L with a ±80e range. The entire sensor package has a low current (approx. 35mA) and voltage (3.3 or 5V) requirement and communication between the microcontroller and sensors can be completed using I²C, commonly called two wire communication.

5.2.3 USE OF INERTIAL SENSORS FOR REHABILITATION

The use of inertial sensors in rehabilitation began in the 1970s with research into the uses for measuring movement during gait [106]. Inertial sensors are still used today for gait [107] but have been found to be more problematic than infrared systems due to their relative sensing technologies. Many pedometers today use accelerometers to determine gait velocity and cadence.

The other major area in rehabilitation where inertial sensors are used in is judging activity level or patterns of activity during the day [108]. These systems use inertial sensors in mostly static positions to determine posture and when the inertial sensor has a dynamic response an aggregate of the total motion is used to determine activity level.

Inertial sensors are also used in upper limb rehabilitation and tracking [88, 109-111]. Many of these systems use inertial sensors with visual or infrared tracking to create a more robust solution. Others use only inertial sensors much like the ATLAS which is described in this and the following chapter. The continually lowering cost of inertial sensors and their increasingly ubiquitous nature will only lead to their increased use in rehabilitation as well as other fields.
5.2.4 **Kalman Filtering with IMUs**

In his seminal paper Kalman presented a state-based recursive filter that could be used for IMU calculations [112]. There is plenty of research on using IMUs with Kalman filters in various fields including navigation of autonomous vehicles, camera calibration, and rehabilitation [113-116]. The Discrete Kalman filter (DKF) uses a state model that is updated based on sensor readings of different reliabilities and the previous time-state of the system. The global steps of a Kalman Filter are outlined in the figure below.

![Flow Diagram of the Kalman Filter Global Steps](image)

**FIGURE 35: FLOW DIAGRAM OF THE KALMAN FILTER GLOBAL STEPS**

The prediction and update steps are where the calculation takes place. The sensors may need additional filtering and conditioning before they can be used in these steps. The generic version of the DKF has five main equations, two in the prediction step and three in the update step [117]. These equations are presented below.

**EQUATION 4: EQUATIONS FOR THE PREDICTION STEP OF THE DISCRETE KALMAN FILTER**

\[
\hat{x}_k^- = A \hat{x}_{k-1} + Bu_{k-1} \quad \text{Project the next step of the state (i)}
\]

\[
P_k^- = AP_{k-1}A^T + Q \quad \text{Project the error covariance ahead (ii)}
\]

Equation 4.i is used to predict the coming state of the system. The variable \(\hat{x}_k^-\) signifies the state estimation at step \(k\) and \(\hat{x}_{k-1}\) is the state from the previous step. The variables \(A\) and \(B\) are matrices of \(n \times n\) and \(n \times l\) size respectively. \(A\) describes the state of the system at steps \(k-1\) and \(k\) and \(B\) describes the optional control input of the previous step, \(u_{k-1}\), of the state \(x\). The
Equation 4.ii is used to predict the coming error covariance (an estimation of the accuracy of the state of the system) in the step, $P_k^-$. The previous error covariance is the variable $P_{k-1}$ which is multiplied by $A$ and its transpose. $Q$ is the process noise covariance. The equations for the measurement update step of the DKF are below.

**EQUATION 5: EQUATIONS FOR THE MEASUREMENT UPDATE STEP OF THE DISCRETE KALMAN FILTER**

\[ K_k = P_k^-H^T(HP_k^-H^T + R)^{-1} \] Calculate the Kalman Gain (i)

\[ \hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \] Update estimate with measurement (ii)

\[ P_k = (I - K_kH)P_k^- \] Update the error covariance (iii)

Equation 5.iii is used to calculate the Kalman Gain of the system at this step which is an n x m matrix, $K_k$. The n x m matrix $H$ is used to relate the state to the measurement $z_k$. $R$ is the measurement noise covariance. The next equation, Equation 5.ii, is used to update the estimate for the state found in the first equation of the prediction step (4.i) with the measurement for the current step. The innovation or residual is signified by the difference between the measurement and state relation ($z_k - H\hat{x}_k^-)$. The last equation updates the error covariance in anticipation of the next step prediction (ii) by using an identity matrix, $I$, minus the Kalman Gain for the step then multiplied by the previous error covariance.

The DKF continues this cycle of predict and correct for each step of the system. A DKF was used on the SDOF IMU for the VRack project which is further described in the later sections of this chapter.

5.2.5 **DIRECTION COSINE MATRICES WITH IMUS**

The purpose of direction cosine matrices (DCMs) is to create rotation matrices that can be used in conjunction with other filtering to find estimations for orientation and position of an IMU.
Another method commonly used to create rotation matrices is the use of Euler angles. DCMs are most commonly used for navigation of drones and watercraft \[118\]. The calculations for DCMs begin with the rotation matrices which can be seen below.

**EQUATION 6: ROTATION MATRICES FOR X, Y, AND Z UNIT VECTORS**

\[
A_x = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \phi & -\sin \phi \\
0 & \sin \phi & \cos \phi
\end{bmatrix} \\
A_y = \begin{bmatrix}
\cos \theta & 0 & \sin \theta \\
0 & 1 & 0 \\
-\sin \theta & 0 & \cos \theta
\end{bmatrix} \\
A_z = \begin{bmatrix}
\cos \psi & -\sin \psi & 0 \\
\sin \psi & \cos \psi & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

These rotation matrices correspond to gimbal type motion in the cartesian axes (with \( \phi \), \( \theta \), and \( \psi \) corresponding to roll, pitch, and yaw respectively). They can be combined to create a larger matrix that can be a representation of a orientation in all three axes. This is a very useful matrix for use with IMUs because it allows for the different orientation axes to be calculated based on data available from rate gyroscopes, accelerometers, and magnetometers. The combined matrix can be seen below.

**EQUATION 7: ORTHOGONAL LEFT-HANDED ROTATION MATRIX**

\[
A = \begin{bmatrix}
\cos \theta \cos \psi & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\
\cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \\
-\sin \theta & \sin \phi \cos \psi & \cos \phi \cos \theta
\end{bmatrix}
\]

A matrix similar to this is used in the ATLAS algorithm for both orientation and position. A DCM can be formed by looking at the two positions of the state of the system and finding the cosine between the values. A diagram further explaining the state change and the DCM can be seen on the following page.
The DCM is used because it allows the calculation of the Euler angles without having to constantly reference the origin values for each angle. The cosines are calculated between steps in the state of the system and used to determine the angles. A generic DCM can be seen in the equation below.

\[
DCM = \begin{bmatrix}
\cos(x_1, x_2) & \cos(x_1, y_2) & \cos(x_1, z_2) \\
\cos(y_1, x_2) & \cos(y_1, y_2) & \cos(y_1, z_2) \\
\cos(z_1, x_2) & \cos(z_1, y_2) & \cos(z_1, z_2)
\end{bmatrix}
\]

DCMs offer a simple method for calculating rotations and can be easily programmed into most microcontrollers due to having smaller matrices than a Kalman Filter which would require a 9x9 matrix for the number of angles being calculated.

## 5.3 IMU Test Bed

A test bed was needed in order to characterize the IMUs so that the orientation and position estimation algorithms could be created. The following sections outline the design and fabrication of this test bed. The test bed was designed in collaboration with Richard Ranky and Alan Angridozza.
Characterization of the IMUs would require that the IMUs orientation could be changed in a controlled manner that had independent sensing. This lead led to the first version of the test bed which used two high precision potentiometers (3852, Bourns) to measure the roll and pitch of the IMU, a CAD rendering and photo of the test bed can be seen below.

The first version of the test bed was fabricated using rapid prototyping (SLA) and was mounted on a tripod and used a bubble level and perpendicular guides to determine the starting angles. This proved to be adequate for initial testing and characterization of the IMUs. However, a new version of the test bed was needed so that velocity during the change in orientation could be controlled. This second version used two servomotors (SM-S3317S, SpringRC) in conjunction with the two potentiometers to control the orientation of both IMUs. CAD renderings of the second test bed can be seen on the following page.
The second version of the test bed was also created using additive manufacturing (SLA) and was mounted on a tripod or a linear guide for testing. Once again a bubble level and perpendicular guides were used to zero the system. The servomotors were used to rotate the IMU in roll and pitch as the default configuration. In order to power and control the test bed an Arduino microcontroller and power supply were added after the fabrication of the parts was complete. Photos of the completed v2 test bed can be seen on the following page.
The potentiometers used in the test bed were characterized to insure that they were linear and would give an accurate angle change calculation. A protractor was used to measure angle in the test bed and then checked against the potentiometer signal. The graph of this test can be seen in Figure 40 below.

\[ \text{Angle} = 0.2473 \times \text{(Pot. Value)} - 41.981 \]

\[ R^2 = 0.9998 \]
The R2 value of 0.9998 showed that the potentiometers were linear and the equation created from the experimental data was used to calculate the angle in all the tests using the test bed. Software was written to control the test bed using the Arduino and Processing. The program is written so that the test bed changes orientation a set number of repetitions with set velocity, angle change, and step size. The Processing code controls the Arduino to modify the position of the servomotors and then writes the data to a text file. The code fragment that modifies the servomotor position is below.

**CODE_FRAGMENT 3: IMU TEST BED SWEEP ALGORITHM**

```
1. //sweep mid to max
2. for(pos = midDeg-step; pos < maxDeg; pos += step) {
3.     writeDelayReadSpit(pos, pos2);
4. }
5. // sweep max to mid
6. for(pos = maxDeg; pos >= midDeg; pos -= step) {
7.     writeDelayReadSpit(pos, pos2);
8. }
9. // sweep mid to min
10. for(pos = midDeg; pos >= minDeg; pos -= step) {
11.     writeDelayReadSpit(pos, pos2);
12. }
13. //sweep min to mid
14. for(pos = minDeg-step; pos < midDeg; pos += step) {
15.     writeDelayReadSpit(pos, pos2);
16. }
```

The code fragment uses four for loops to create a triangle wave signal that is sent to the servomotors. Each for loop moves the servo between either min and max and the origin. Inside the for loops (lines 3, 7, 11, and 15) calls the function writeDelayReadSpit, that function uses the two position variables, pos and pos2, to change the position of the two servomotors. It then delays for a certain period of time to allow the servo to move and then reads data from the IMU and the potentiometers and prints that data to a text file. This code is run within the Processing draw function which functions as a loop.
5.4 VRack Pedal Angle

The VRack (see Appendix Section A.1) is a low-cost modular sensor system that can be installed on any stationary bicycle and the system used a 5DOF IMU to calculate the pedal angle the rider. The following sections outline the algorithm and testing process to calculate the pedal angle.

5.4.1 5DOF IMU Algorithm for Pedal Angle

In order to calculate the angle of the pedal the sensor signals needed from the 5DOF IMU were the Y-axis and Z-axis accelerometers and the Y-axis gyroscope (pitch). This calculation did not need to be real-time so it was completed in post processing of the data by using a MATLAB program that implements a Kalman filter. This algorithm was created in collaboration with Oussama Drissi.

The VRack system acquires data at 100Hz and creates a large data file that includes raw data from all the sensors including the IMUs on the pedals. This data file is used with the MATLAB code to determine the pedal angle. The MATLAB algorithm is contained in two separate files, one for the setup and output of the algorithm and the other for the actual Kalman filter calculations. The algorithm steps for both files are presented in the tables below.

<table>
<thead>
<tr>
<th>TABLE 6: STEPS IN THE MATLAB ALGORITHM FOR VRACK PEDAL ANGLE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Steps in the MATLAB Algorithm</strong></td>
</tr>
<tr>
<td>Import the appropriate data file</td>
</tr>
<tr>
<td>Unit conversions for Y-accel, X-accel, and Y-gyro</td>
</tr>
<tr>
<td>Initialize the change in time for the calculation</td>
</tr>
<tr>
<td>Set the covariance of the error matrix</td>
</tr>
<tr>
<td>Set the state equation parameters for matrices A, B, and C</td>
</tr>
<tr>
<td>Set initial state for angle and gyro bias</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Set the measurement covariance noise

\[ R = 0.30000 \]

Found from the accelerometer jitter

Set the process covariance noise matrix

\[ Q = \begin{bmatrix} Q_{\text{angle}} = 0.2000 & 0 \\ 0 & Q_{\text{gyro bias}} = 0.3000 \end{bmatrix} \]

Create the output array

The output array is \( X \)

Apply the Kalman Filter to the previous signal recursively

This is done using a for loop based on the length of the data file

Set the observations for \( Y \)-accel and \( Z \)-accel

\[ y = \text{atan2}(-Z_{\text{accel}}, -Y_{\text{accel}}) \]

Set the deterministic input

\[ u = \begin{bmatrix} Y_{\text{gyro}} \\ 0 \end{bmatrix} \]

Update the state using the Kalman Filter function

The Kalman Filter function is in a second MATLAB file discussed below

Place the calculated values in the output array

The values are placed in the \( X \) array

Graph the results

The output array is graphed against the time

---

The MATLAB algorithm contains a function call to the second file which has the code for the actual Kalman Filter function. The function uses the variables defined in the MATLAB algorithm: \( A, B, C, x, P, u, y, Q, \) and \( R \). The table below contains the steps and equations in the Kalman Filter function.

### TABLE 7: STEPS IN THE KALMAN FILTER FUNCTION FOR THE VRACK PEDAL ANGLE

<table>
<thead>
<tr>
<th>Steps in the Kalman Filter Function</th>
<th>Value/Equation/Note</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State Prediction</strong></td>
<td></td>
</tr>
<tr>
<td>Prediction of the Angle</td>
<td>( x = A \ast x + B \ast u )</td>
</tr>
<tr>
<td>Prediction of the covariance matrix</td>
<td>( P = A \ast P \ast A^T + Q )</td>
</tr>
<tr>
<td><strong>State Update</strong></td>
<td></td>
</tr>
<tr>
<td>Calculate angle using accelerometers</td>
<td>( w = y - C \ast x )</td>
</tr>
<tr>
<td>Correct the angle and bias</td>
<td>( E = C \ast P \ast C^T + R )</td>
</tr>
<tr>
<td></td>
<td>( K = A \ast P \ast C^T \ast E^{-1} )</td>
</tr>
<tr>
<td>Calculate the covariance of the prediction error</td>
<td>( P = P - K \ast C \ast P )</td>
</tr>
<tr>
<td>Correct the prediction of the state</td>
<td>( x = x + K \ast w )</td>
</tr>
</tbody>
</table>

The algorithm and filter function provide the angle output using the \( Y \)-accel, \( Z \)-accel, and \( Y \)-gyro data from the 5DOF IMU. However, in order to get peak accuracy the 5DOF IMU needed to be characterized a process that is discussed in the next section.
5.4.2 Pedal Angle Characterization Testing Setup and Protocol

In order to use the algorithm effectively the 5DOF IMUs had to be characterized because their specified values were not accurate enough. The specification sheet for the accelerometer states the initial voltage value to be 1.5V and the voltage difference for a g is 300mV. The specification sheet for the gyroscope states that the initial voltage is 1.35V and the sensitivity is 2mV/°/s. The v2 test bed was used to compare the specified values against experimental values. The setup for the v2 test bed can be seen below.

![Image of the IMU Testbed V2 Setup for VRack Pedal Angle Calibration](image)

The 5DOF IMU was tested using different angle ranges and step sizes (speeds). Each trial contained ten or eleven repetitions, depending on the angle range. The potentiometer data was used to compare the calculated angle from the MATLAB code. The results of the characterization can be found in the next section.

5.4.3 Characterization Testing Results

The test bed was used to examine five angle ranges: 55°, 83°, 110°, 138°, 165°. These ranges were chosen based on the pedal angle changes found in healthy patients [119]. The mounted
5DOF IMU was also held at 90° and 0° to determine the correct voltage range and initial voltage offset for the Y-accel, Z-accel, and the initial offset for the Y-gyro, the values of which can be seen in the table below.

<table>
<thead>
<tr>
<th>Value</th>
<th>Y-accel</th>
<th>Z-accel</th>
<th>Y-gyro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage Range</td>
<td>0.3376</td>
<td>0.3176</td>
<td></td>
</tr>
<tr>
<td>Initial Offset</td>
<td>1.62325</td>
<td>1.6568</td>
<td>1.3427</td>
</tr>
</tbody>
</table>

The characterization testing was very successful with every test showing very similar results. The IMU calculation was very accurate compared to the potentiometer angle. The graph for the 55° test displaying these results can be seen below.

![Graph of VRack Pedal Angle Calibration 55° Kalman Filter vs. Potentiometer](image)

**FIGURE 42: GRAPH OF VRACK PEDAL ANGLE CALIBRATION 55° KALMAN FILTER VS. POTENTIOMETER**

Once characterization of the 5DOF IMU was complete and the algorithm was proven experimentally characterization of the pedal angle on the VRack system was needed. That testing is discussed in the next section.
5.4.4 **Healthy Subject Characterization Testing IMU vs. Peak System**

The 5DOF IMUs installed on the VRack had to be characterized in the same way as the mounted 5DOF IMU in the previous section. However, instead of using potentiometers a Peak system (IR position) was used. The pedals were held at 0° and 90° to determine the initial voltage offset and range for the accelerometers and gyroscopes. A table containing the found values is below.

<table>
<thead>
<tr>
<th>Value</th>
<th>Y-accel</th>
<th>Z-accel</th>
<th>Y-gyro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right voltage range</td>
<td>0.3012</td>
<td>0.2987</td>
<td></td>
</tr>
<tr>
<td>Right initial offset</td>
<td>1.734696</td>
<td>1.65633</td>
<td>1.649</td>
</tr>
<tr>
<td>Left voltage range</td>
<td>0.3008</td>
<td>0.3014</td>
<td></td>
</tr>
<tr>
<td>Left initial offset</td>
<td>1.757696</td>
<td>1.628897</td>
<td>1.3512</td>
</tr>
</tbody>
</table>

Once the 5DOF IMUs were characterized data was collected by changing the angle of the pedal by hand and collecting data. While the 90° range is out of the expected range of angle changes while pedaling it became important to test to ensure the viability of the Kalman Filter algorithm. The graphs for the testing of the left and right pedal IMUs are on the following page.

The Peak system was setup with markers on either end of the pedal. This has the potential to cause some error because the distance between the markers is not very large. There were also issues involving the Peak losing the reflective signal of the markers causing spikes in the test data.
FIGURE 43: GRAPH OF LEFT VRACK IMU 45° TEST VS. PEAK

FIGURE 44: GRAPH OF THE LEFT VRACK IMU 90° TEST VS. PEAK
The tests were very successful as shown the data from the IMU is very close to the found angle from the Peak system. There were issues with the time of acquisition from the VRack which can be seen in these tests as an offset near the end of the tests. This has since been removed due to a modification of the VRack system (defragmenting the hard drive and increasing available...
RAM) as well as a change to the MATLAB algorithm to use the actual change in time \( dt \) between data points.

5.5 ATLAS HAND ORIENTATION

The ATLAS (Chapter 6) used a 9DOF Razor IMU mounted on the back of the hand to collect data and calculate the orientation of the hand. This IMU has an Arduino microcontroller that was used to condition the data before it was sent to the main ATLAS microcontroller. The following sections detail the algorithm, characterization, and results of the hand orientation calculation.

5.5.1 9DOF RAZOR IMU ALGORITHM FOR HAND ORIENTATION

As has been previously stated the IMU used in this project has a microcontroller onboard which allowed for the creation of a real-time algorithm to estimate the orientation of the IMU and therefore the hand. The algorithm is adapted from the algorithm presented in [118] and uses direction cosine matrices. Since the program for the algorithm is written using the Arduino IDE there are two main functions: setup and loop. The setup function initializes all the sensors, variables, and matrices used in the algorithm. It also collects a baseline reading that is used to offset the initial orientation of the IMU. The table below presents the steps in the setup function of the Arduino program.

<table>
<thead>
<tr>
<th>TABLE 10: STEPS IN THE ATLAS HAND ORIENTATION SETUP FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Steps in the Setup Function</strong></td>
</tr>
<tr>
<td>Initialize the serial connection</td>
</tr>
<tr>
<td>Initialize the analog signals and accelerometer</td>
</tr>
<tr>
<td>Initialize the magnetometer</td>
</tr>
<tr>
<td>Initialize the gyroscope</td>
</tr>
<tr>
<td>Collect a series of samples from sensors</td>
</tr>
<tr>
<td>Calculate and apply offset to sensors</td>
</tr>
</tbody>
</table>
The default orientation of the IMU had to be modified for this algorithm due to its mounted orientation on the back of the hand. Prior to the setup function the program does declaration and instantiation of constant, variables, and data structures (vectors and matrices) used in the algorithm. Once the setup function has completed successfully the loop function is called continuously until the program terminates. A table outlining the steps of the loop function is below.

**TABLE 11: STEPS IN THE ATLAS HAND ORIENTATION LOOP FUNCTION**

<table>
<thead>
<tr>
<th>Steps in the Loop Function</th>
<th>Value/Equation/Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify the change in time since previous loop</td>
<td>The $dt$ is used in several of the calculations as well as to control the rate of output sent over the serial connection</td>
</tr>
<tr>
<td>Acquire data from the accelerometer, gyroscope, and magnetometer</td>
<td>Data is placed into a matrix or vector where appropriate</td>
</tr>
</tbody>
</table>
| Calculate compass heading from the magnetometer | $x_{heading} = x_{mag} \cos(\theta) + y_{mag} \sin(\phi) + z_{mag} \cos(\phi) \sin(\theta)$  
$y_{heading} = y_{mag} \cos(\phi) - z_{mag} \sin(\phi)$  
$yaw = atan2(-y_{heading}, x_{heading})$ |
| Update direction cosine matrix with new sensor data | An integrator and proportional term are added to the matrix for control of the gyroscope |
| Normalize the direction cosine matrix | Using dot product |
| Correct the drift from the previous step | The magnitude of the accelerometer vector is calculated for a reliability filter  
Yaw is corrected with the new step  
An integrator and proportional term are used on the heading |
| Calculate the Euler angles | $roll: \phi = -\arcsin(DCM_{2,0})$  
$pitch: \theta = \arctan2(DCM_{2,1}, DCM_{2,2})$  
$yaw: \psi = \arctan2(DCM_{1,0}, DCM_{0,0})$ |

At the end of each loop the data for the appropriate sensors and/or calculations was printed to serial. As with the VRack system this algorithm and the 9DOF Razor IMU had to be character-
ized to ensure accuracy of the orientation estimation. This characterization is described in the next sections.

5.5.2 **HAND ORIENTATION CHARACTERIZATION TESTING SETUP AND PROTOCOL**

The characterization testing used the v2 IMU test bed that was used for the VRack. It required a new mounting for the 9DOF Razor IMU as it is larger than the 5DOF. The testing was done at ±50° and ±70° angle ranges with five repetitions. Once again the v2 test bed would be mounted on a tripod and a bubble level would be used to ensure a neutral starting angle. Photographs of the testing setup can be seen below.

![IMU Test Bed V2 in Roll-Pitch Mode](image_url)

**FIGURE 47: PHOTOGRAPH OF IMU TEST BED V2 IN ROLL-PITCH MODE**
In this orientation the 9DOF Razor IMU was tested in both pitch and roll (separately and together), however with the design of the test bed there is no way to test the third rotation (yaw) without modification. A metal angle was used to mount the v2 test bed vertically which allowed for testing of yaw. Figure 49 and Figure 50 are images of the test bed in the yaw configuration.
As with the previous characterization testing using this test bed, Processing and Arduino were used to program the microcontroller to control the servomotors and acquire data from the IMU and potentiometers. The results of the characterization testing are presented in the next section.

5.5.3 Characterization Testing Results

The outcome of the testing was very positive. The most accurate axes were roll and pitch. Both axes had a very tight fit to the angle found from the potentiometers. Variations in speed and angle range did not affect their accuracy (testing was performed at 2°, 5°, 10°, and 20° per second with a range of ±50° or ±70°). For this testing, 180° is considered to be horizontal for both roll and pitch and true north for yaw. The graphs for the positive-negative fifty degree single axis tests for pitch and roll are on the next page.
The yaw axis is more prone to drift because, as mentioned previously, accelerometers cannot detect changes in yaw as they are perpendicular to gravity. The yaw calculation is therefore less accurate than the pitch and roll. Possible modifications to reduce this error can be seen in the following section and the future work section of this chapter. The graph of the fifty degree yaw test is on the following page.
To ensure that there was no crosstalk between the sensors or issues with algorithm during combined motion tests were completed with pitch and roll together and with yaw and roll. Once again pitch and roll were very accurate, within two degrees error. The graph below is the joint test of pitch and roll for a fifty degree range.
As with the VRack once characterization testing was completed successfully the device was tested with healthy subjects. That testing is discussed in the next section.

5.5.4 Healthy Subject Testing for Orientation

Healthy subject testing for orientation was completed as part of larger testing done for the AT-LAS device which is further explained in the next chapter. First, subjects (n=10) completed the bend sensor protocol using the right hand. Then, they completed the IMU hand in space protocol using the same hand. This consisted of moving the wrist and/or forearm in six different ways: roll (supination/pronation), pitch (wrist extension/flexion), yaw (radial/ulnar deviation), medial-lateral (x), forward-backward (y), and up-down movements (z). The results of the orientation tests are in the graphs below.

![Graph of IMU Roll from Supination-Pronation of the Forearm](image)

**FIGURE 55: GRAPH OF HEALTHY SUBJECT SUPINATION-PRONATION OF THE FOREARM TO ROLL**

The goal of the testing for roll was to see approximately a 180° change based on the flexibility of the subject. The subjects were asked to start in full supination (0°) and then rotate to neutral (180°). There was consistency between all subjects for the test and the testing very clearly
shows the five repetitions of the testing. Subjects varied 30° on the maximum range and 20° on the minimum. The results of the second test, pitch, can be seen in Figure 56.

Subjects started in full flexion and then moved to full extension, neutral is considered to be 180°. Only eight subjects are represented on the graph due to an error in the pretest calibration algorithm for two subjects. The pitch data was also consistent and it is easy to discern the five repetitions of the test. The range of the values is as expected, each healthy subject had their wrist flexion and extension measured prior to the test and the calculated values fell within these ranges. The pitch data and range for a single subject is in the graph on the following page.

This subject had a wrist flexion value of 59° and an extension value of 60°. Their pitch data falls within this range which is expected because due to the cadence of the testing the subjects were not always able to reach their maximum and stay with the timing.
The yaw test (Figure 58 above) had the largest range of start values due to positioning of the subject (for their comfort and largest range of motion) during the test and the yaw measurement being an absolute measurement. The ranges of the values are very consistent with all subjects falling between $35^\circ$ and $50^\circ$. The five repetitions can once again be seen easily showing that the subjects were able to use the cadence during the test.

The second half of this testing that pertains to the position estimation is discussed in the section below. For more information on the protocol and testing presented here see section 6.5.
5.6 ATLAS HAND POSITION

In concert with the calculations for hand orientation, a hand position calculation was also derived. The calculation was completed with an algorithm on the 9DOF Razor IMU microcontroller and based on the orientation calculation. The following sections explain the algorithm and the implementation and testing.

5.6.1 9DOF IMU ALGORITHM FOR POSITION

The algorithm is based off the same direction cosine matrix method used in the hand orientation. In order to find a calculation for position the accelerometers must be used. However, this is problematic because not only are accelerometers affected by impacts and they have high jitter, they measure two forces simultaneously: contribution from gravity and linear acceleration in axis. To accommodate this there are several new steps to the algorithm which can be seen in the algorithm flow diagram below.

![Algorithm Flow Diagram](image)

**FIGURE 59: FLOW DIAGRAM OF STEPS IN POSITION CALCULATION ALGORITHM**

The first six steps of the algorithm are the same as the hand orientation algorithm. The last two steps, gravity calculation and position estimation, are specific for the calculation of position. The first step is to estimate the accelerometers contribution from gravity from the orientation calculation and then remove it. The next is to use the corrected acceleration to find the position. The new steps in this algorithm are presented in detail in the table on the following page.
TABLE 12: NEW STEPS IN THE POSITION ALGORITHM FOR THE ATLAS

<table>
<thead>
<tr>
<th>New Steps in the Position Algorithm</th>
<th>Value/Equation/Note</th>
</tr>
</thead>
</table>
| Find expected contribution of gravity in accelerometer signals from orientation | \[ x_{\text{gravity}} = \sin(\theta) \cdot g \]
| | \[ y_{\text{gravity}} = \sin(\phi) \cdot g \]
| | \[ z_{\text{gravity}} = (x_{\text{gravity}}) \times (y_{\text{gravity}}) \]
| Subtract the expected gravity from the accelerometer signal | |
| Apply dead zone to the accelerometer signal | This step reduces drift during periods of static operation |
| Double integrate the accelerometer signal to find position | \[ \text{pos} = \int_0^t (\text{accel} \cdot dt) \cdot dt \]
| Alternatively equations of linear motion can be used to find position from acceleration | \[ \text{pos}_k = v_{k-1}t_k + a_k \frac{t_k^2}{2} \]

The contributions from gravity are found using trigonometry and then the result is subtracted from the acceleration. Next a dead zone is applied to reduce the drift from jitter when the accelerometer is static. Then the calculation for position estimation is completed.

Initial testing of this algorithm proved that the algorithm worked but that the error was so large that the data was unusable for position estimation. Several methods of filtering were attempted to compensate for the error. The first was a single axis Kalman Filter, the code of which can be seen below.

**CODE FRAGMENT 4: SINGLE AXIS KALMAN FILTER FOR HAND POSITION**

```c
1. void kalmanUpdateX(){
2. float measurement;
3. measurement = bpAccel[0];
4. //prediction update
5. //omit x = x
7. //measurement update
10. kfaX[2] = (1 - kfaX[4]) * kfaX[2];
11. }
```

The Kalman Filter calculations become very simple in a single axis because there are no matrices. The prediction and update steps are still used in a single axis version of the Discrete Kalman
Filter. The results for the filter were mixed depending on time duration and speed of movement so another filter was implemented, a band-pass filter.

CODE FRAGMENT 5: BAND-PASS FILTER FOR HAND POSITION

```c
//High-pass Filter for the Accelerometer Data
float highPassFilter(float currentAccel){
    float hpData;
    float hpData = alphaHP * prevHPX + alphaHP * (currentAccel - prevHPX);
    prevHPX = hpData;
    return hpData;
}

//Low-pass Filter Function for the Accelerometer Data
float lowPassFilter(float currentAccel){
    float lpData;
    lpData = prevLPX + alphaLP * (currentAccel - prevLPX);
    prevLPX = lpData;
    return lpData;
}

//Band Pass Filter
void bandPassFilter(){
    for(int i = 0; i < 3; i++){
        float bpd = lowPassFilter(Accel_Vector[i], i);
        bpAccel[i] = highPassFilter(bpd, i);
    }
}
```

A band-pass filter is made up of a low-pass and high-pass filter coupled together. This filter is used for a single axis and first the low-pass filter is used, then the result is run through the high-pass filter. The bandwidth was determined based on the movement of the specific experiment. This filtering also had mixed results in initial testing. Characterization testing using post-processing of the data was performed (presented in the next section) to try to determine possible solutions to the issues of position calculation.

5.6.2 CHARACTERIZATION TESTING SETUP AND PROTOCOL

The v2 IMU test bed was once again used for the testing. However, instead of mounting on a tripod the test bed was mounted on a linear guide with an attached linear potentiometer. The test bed was then moved linearly (manually) and data was collected from the linear potentiom-
eter, the rotary potentiometers on the test bed, and the sensors on the IMU. An image of the test bed in this configuration can be seen below.

![Image of test bed](image)

**FIGURE 60: IMAGE OF THE CHARACTERIZATION TESTING ASSEMBLY FOR ATLAS HAND POSITION**

Before testing could begin in earnest the linear potentiometer needed to be tested for linearity. A scale was used to measure distance and compare that to the voltage of the linear potentiometer. A graph of this test can be seen below.

![Graph of linear potentiometer](image)

**FIGURE 61: GRAPH OF LINEAR POTE NTIOMETER FOR IMU TEST BED DISTANCE VS. VOLTAGE**

The linear potentiometer proved to be very linear with an $R^2$ value of 0.9999. Several tests were completed including single movement linear axis tests and more complicated dynamic tests...
with the test bed servomotors rotating the IMU. The results of these tests are presented in the next section.

5.6.3 Characterization Testing Results
The first test attempted with the linear test bed setup was a single change in distance while the IMU was in a static orientation. The distance change was performed manually in a horizontal direction at a high rate of velocity with a very fast deceleration. The potentiometer distance graph of this movement can be seen below along with the matching raw accelerometer signal.

![Graph of Linear Potentiometer Position and Raw Accelerometer Signal](image)

FIGURE 62: GRAPH OF THE LINEAR POTENTIOMETER POSITION AND THE RAW ACCELEROMETER DATA FOR SINGLE MOVEMENT TEST

The test took approximately two seconds although the actual movement lasted for less than a quarter of a second. The raw data for the accelerometer looks as expected except for the large impact data point that occurred at the end of the movement. For this movement that is linear a third order acceleration signal was acquired from the IMU. Before the post processing on the acceleration could be performed some manipulation to raw data had to be completed. The graphs on the following page display the corrected data and the velocity for the test.
The raw accelerometer data had to undergo a sign change because the IMU is mounted backwards. The impact data point was removed using a band-pass filter. Point-by-point integration was then performed on the acceleration data to estimate the velocity. Once the velocity was calculated the position could be estimated, the graph of the position estimation is below.

FIGURE 63: GRAPH OF THE CORRECTED RAW ACCELERATION DATA AND THE VELOCITY FOR THE SINGLE MOVEMENT TEST

FIGURE 64: GRAPH OF THE POSITION FROM THE LINEAR POTENTIOMETER AND IMU FOR THE SINGLE MOVEMENT TEST
The accelerometer position calculation did not adjust for bias in velocity or position, therefore the longer the accelerometer acquired data the more erroneous the estimation will become. The direction and duration of the movement can be inferred from the data but no accurate position can be estimated. The position estimation was not accurate enough from this testing and a new sensor scheme and algorithm needs to be developed.

5.6.4 Healthy Subject User Testing for Position

As described in the hand orientation sections and the next chapter the testing for the hand position was part of overall testing of the ATLAS system. The subjects (n=10) were asked to move through a distance of ±30cm for 2 seconds, hold the end position for one second, and return a total of five times for each test. Tests were conducted in x (medial-lateral), y (forward-backward), and z (up-down) axes.

The data suffered from the same issues as the characterization testing presented in the previous section namely the bias from the accelerometers cannot be corrected algorithmically. This leads to the sensor having a large drift that overtakes the signal making it unreliable. There has been research into the possible methods of correction of accelerometer and gyroscope bias and it can be applied here to show further show the issues of using accelerometers for position estimation [120]. The raw acceleration from a subject for the lateral test can be seen on the following page.
As with the characterization testing the acceleration data was conditioned to increase accuracy. The next step was to calculate the velocity using point-by-point integration. The graphs for the velocity of the lateral test can be seen in Figure 66 below.
The velocity calculation introduced a bias that had to be corrected to increase the accuracy. This is a manual operation that requires finding the linear fit to the data and using that to adjust the x axis and remove the bias. The graph on the right on the previous page in Figure 66 is the now corrected velocity data that still shows the five distinct repetitions that are shaped like parabolas due to the first integration of the third order accelerometer data. The position could be estimated from the corrected velocity and the graphs of the outcome are below.

![Graphs of Raw Position and Corrected Position](image)

**FIGURE 67: GRAPHS OF RAW POSITION AND CORRECTED POSITION FOR A LATERAL POSITION TEST OF A HEALTHY SUBJECT**

The position estimation also suffered from a bias that could be corrected to some extent manually. The accuracy of the calculation is not within acceptable levels for use in rehabilitation systems. Possible solutions to these calculation issues are presented in the next section.

**5.6.5 FILTERING AND DATA ISSUES WITH POSITION CALCULATION AND POSSIBLE SOLUTIONS**

The results found from both the characterization and healthy subject testing were expected and they necessitate a better solution for position estimation. Two possible solutions are the implementation of an absolute position sensor system to the ATLAS such as infrared or magnet-
ic tracking or placing two more IMUs on the arm (one forearm, one upper arm) and using the known lengths of the arm segments and orientation calculation to estimate position.

Infrared and magnetic tracking have been used in upper arm rehabilitation previously and have been proven to be very accurate, so there exists a body of research to draw from [121-124]. A solution using one of these technologies would require that more hardware be added to the ATLAS system, increasing cost and complexity of the system and requiring a completely new version of the software. These technologies also have limitations; infrared tracking requires line of sight and both technologies are prone to artifacts and interference from other equipment.

The reverse kinematics solution using more IMUs has also been previously researched and found to be much more accurate than the single IMU method detailed previously [125, 126]. This solution would also require more hardware to be added to the ATLAS but would not require an overhaul of the software because the algorithm for orientation could be used again. This method is still a relative calculation, meaning that the end point of the shoulder would be used as an origin point but there is no requirement for line of sight and the possibility of artifacts and interference is lower.

5.7 Future Work

The future work for both systems, VRACK and ATLAS, includes more testing with healthy and impaired patients. Specifically for the VRACK more characterization of the IMUs needs to be completed to increase the accuracy of the 5DOF IMUs. This will include static and dynamic testing of the pedal at known angles. The ATLAS orientation and position calculations would also benefit from further characterization testing and perhaps a transition to Kalman Filtering over
the direction cosine matrix method. The position calculation requires more research into possible solutions. Michael Conry and Matthew Jamula completed preliminary research into using two more IMUs to estimate position of the upper limb in the spring of 2012 which included the creation of an algorithm and pilot testing. This will be explored further with a new version of the ATLAS. For more information on future work for the ATLAS see section 6.7.

5.8 SUMMARY

Inertial sensors have been used in rehabilitation systems for over 30 years. In recent years inertial measurement units (IMUs) have become ubiquitous and rise in popularity. This chapter presented three different uses for IMUs in two projects: the VRack (Appendix Section A.1) and the ATLAS (Chapter 6). Two different methods for calculation were also presented: the Discrete Kalman Filter and the Direction Cosine Matrix. A test bed was fabricated for characterization of IMUs that includes servomotors and rotary potentiometers.

For the VRack project a five degree of freedom IMU (three-axis accelerometer and two-axis gyroscope) was used in the pedal assembly to measure pedal angle. A post-processing algorithm was derived for the calculation using a Discrete Kalman Filter. Position results were collected in both the characterization and healthy subject testing. More testing is planned for the device including additional characterization testing and impaired subject testing.

The ATLAS project used a nine degree of freedom IMU (three-axis accelerometer, gyroscope, and magnetometer) to calculate hand orientation and position estimation. The algorithms used were programmed into a microcontroller on the IMU for use in real-time. Both algorithms use a direction cosine matrix approach to complete the calculations. The orientation calculation
proved to be accurate in both characterization and healthy subject testing. The position estimation was inaccurate due to drift, bias, and impacts on the accelerometer. There are several possible solutions for more accurate position estimation that are being currently researched. Future work for the system includes more characterization testing and impaired subject testing.
CHAPTER 6: THE ANGLE TRACKING AND LOCATION AT-HOME SYSTEM FOR BIMANUAL REHABILITATION

CAD RENDERING OF THE ATLAS V3
6.1 INTRODUCTION
The Angle Tracking and Location At-home System for Bimanual Rehabilitation (ATLAS-BR or ATLAS) started as a senior capstone design project in the fall of 2009. The goal of the project was to create a low-cost bimanual glove system to measure the kinematics of the hand and wrist of patients post stroke and allow them to interact with virtual scenes for rehabilitation. A low-cost at-home device that is simple enough for the patient to use alone could have significant economic impact on both the supply side of therapy as well as the demand [127].

The author was a co-advisor on the project with Dr. Mavroidis and Dr. Holden. The team of five undergraduate engineering students (Avi Bajpai, Caitlyn Bintz, Jason Chrisos, Andrew Clark, and Drew Lentz) was charged with creating the first and second versions of the device which is summarized in Section 6.3.4 and was tested by two physical therapy students (Sarah Hines and Alyson Jodoine). Once the students finished the author continued the project and created the current third version discussed in Section 6.3.5. The virtual scenes created for the ATLAS are discussed in the next chapter.

6.2 BACKGROUND INFORMATION
This section presents information on the biomechanics of the hand and wrist, hand devices currently available, and potentiometer bend sensors. The ATLAS relies on the additional background from chapter 2 and 3 as well as the material presented below.

6.2.1 BIOMECHANICS OF THE HAND AND WRIST
The human hand has twenty-seven degrees of freedom: four in each of the four fingers totaling sixteen, the thumb has five degrees of freedom, and the wrist has six degrees of freedom. The hand consists of twenty-seven bones and four articulations: interphalangeal articulations of
The four fingers have three degrees of freedom for extension and one for abduction and adduction. A diagram of the hand skeleton is below. [128]

The hand skeleton consists of eight carpal bones, five metacarpals, five proximal phalanges, four intermediate phalanges, and five distal phalanges. The carpal bones connect to the two bones of the forearm: radius and ulna. The joints between the finger bones that are measured using the ATLAS can be seen in the figure on the following page [128].
The metacarpophalangeal (MCP) joints of the fingers and thumb are between the metacarpal bones and the proximal phalanges. The proximal interphalageal (PIP) joints of the fingers are between the proximal and intermediate phalanges. The thumb has no intermediate phalange so the second joint from the metacarpal is the interphalageal (IP) joint between the proximal and distal phalanges. The table below presents the normal range of motion for the joints measured by the ATLAS bend sensors from Mallon et al. [129].

<table>
<thead>
<tr>
<th>Movement</th>
<th>Digit</th>
<th>Men MCP</th>
<th>Women MCP</th>
<th>Men PIP</th>
<th>Women PIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexion</td>
<td>Index</td>
<td>94</td>
<td>95</td>
<td>106</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>Long</td>
<td>98</td>
<td>100</td>
<td>110</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>Ring</td>
<td>102</td>
<td>103</td>
<td>110</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>Little</td>
<td>107</td>
<td>107</td>
<td>111</td>
<td>111</td>
</tr>
<tr>
<td>Extension</td>
<td>Index</td>
<td>29</td>
<td>56</td>
<td>11</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Long</td>
<td>34</td>
<td>54</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Ring</td>
<td>29</td>
<td>60</td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Little</td>
<td>48</td>
<td>62</td>
<td>13</td>
<td>21</td>
</tr>
</tbody>
</table>
The wrist, while not as complex as the fingers in terms of degrees of freedom, is important for the sensors in the ATLAS, namely the IMUs. For many rehabilitation exercises wrist motion is coupled with finger bending [130]. A figure displaying the different axes of global motion for the wrist is below.

![Diagram of the global motions of the wrist]

**FIGURE 70: DIAGRAM OF THE GLOBAL MOTIONS OF THE WRIST**

The wrist has three axis of global motion which can be seen in the figure above. Later in this chapter and in the previous chapter these motions are referred by Euler angle rotations. Extension and flexion is analogous to pitch, radial-ulnar deviation is analogous to yaw or heading and supination-pronation is analogous to roll (pitch, yaw, and roll defined in Chapter 1). There have been several studies to determine the normal range of motion for these degrees of freedom for different populations which can be seen in the table on the following page.
Each of the studies presented in Table 14 have different sample sizes, age ranges, and gender ratios. Of note for the work presented here are the US Army/Air Force results as that study was used to determine hand size in Section 6.5. Ryu et al. had a balanced sample in terms of gender but none of these studies matches the age range used in testing explained in Section 6.5 [139].

The ATLAS has been designed to quantify specific degrees of the freedom in the hand and wrist. The bend sensors are placed to measure the MCP and PIP joints of the index, middle, and ring fingers as well as the MCP and IP joints of the thumb, the next version of the system will include sensors on the small finger. The IMU mounted on the back of the hand is used to measure radial-ulnar deviation of the wrist, supination-pronation of the forearm, and extension-flexion of the wrist.

### Commercial Hand and Glove Devices

There are currently many devices available commercially and in research that have comparable to traits to the ATLAS. In order to create a unique device that serves the stated goals of the project it is important to understand the weaknesses and advantages of these devices. In the following sections summarize the technology and uses for other hand and glove devices.
6.2.2.1 THE P5 GLOVE
The P5 Glove was released from Essential Reality in 2002 aimed at the home personal computer (PC) video game market [140]. The hand device itself is attached to an Infrared tower via a PS/2 cable, that tower is attached to the PC via a USB cable. The P5 comes with a CD that has device drivers that allow the device to work on any computer running a Microsoft operating system and also on any Mac using OS version 9 or below [141]. The P5 can emulate a mouse for any application but can also provide 3D interaction with applications compatible or designed for the P5. A picture of the P5 is below.

FIGURE 71: IMAGE OF THE P5 GLOVE AND INFRARED TOWER [141]

The P5 glove was not designed to be a rehabilitation device. However, its low cost and easy to use software has made it popular for rehabilitation uses. There is no commercially available rehabilitation software for the P5, but many researchers have created their own software or used compatible game software.
One such use of the glove was conducted in 2006 at Rutgers University [142]. The researchers used a P5 Glove and an Xbox console to display virtual environments that were created using Java 3D. The hardware and software for the VR system was chosen based upon cost and what a patient could afford in their home. The researchers found that the low cost setup including the P5 did have much lower functionality than a traditional hand rehabilitation VR system [142].

6.2.2.2 CyberGlove II
The CyberGlove II is a wireless device and has a capacity of twenty-two high-accuracy joint-angle measurements [143]. The glove uses a proprietary resistive bend-sensing technology to capture real-time digital joint-angle data. A figure of the CyberGlove II can be seen below.

![Image of the CyberGlove II](image_url)

**FIGURE 72: IMAGE OF THE CYBERGLOVE II [143]**

The CyberGlove II connects to a computer by a wireless USB receiver. The device has batteries in a forearm-mounted compartment. Software does come bundled with the glove which is for evaluation of the hand and not for virtual reality. Also there is no publicly available software development kit.

The CyberGlove boasts very good accuracy and data resolution but the large cost of the system and lack of sensor data for absolute hand position are the main drawbacks of the device. Also lack of software and expensive software support are an issue in its use in a VR system.
6.2.2.3 5DT Data Glove 5 Ultra
The 5DT Data Glove 5 Ultra is designed for use in motion capture and animation [144]. The device measures finger flexure and interfaces with a PC via a USB cable. The device features automatic calibration and has an on-board processor. A figure of the 5DT Data Glove 5 Ultra can be seen below.

![Figure 73: Image of the 5DT Data Glove 5 Ultra](image)

The 5DT Data Glove 5 Ultra uses Bluetooth technology to make it wireless. It also has a cross-platform software development kit. The bundled software that comes with the device has no rehabilitation applications.

The 5DT Data Glove 5 Ultra lacks sensors for absolute hand position. The cost of the device is moderate but still too high for in-home use. The wireless technology is an advantage for use in a large workspace and for compatibility.

6.2.2.4 Acceleglove
The Acceleglove was developed by AnthroTronix and includes six three-axis accelerometers, one on the back of each fingertip and one on the back of the hand [146]. The device is low-cost and connects to a computer via USB. It comes bundled with an SDK that is cross-platform. The
accelerometers have a range of ±1.5g and the sampling rate of the device is 35Hz. An image of the Acceleglove can be seen in the figure below.

![Image of the Acceleglove](image)

**FIGURE 74: IMAGE OF THE ACCELAGLOVE [146]**

The main use of the Acceleglove is static gesture recognition (for sign language) and software has been developed for this use [147]. The device could be used for simple and static rehabilitation exercises but due to the nature of the accelerometers it cannot be used to measure finger bending while the wrist and arm are moving.

6.2.2.5 **HandTutor**
The HandTutor is a hand and wrist device that was created by MediTouch. The device uses bend sensors to capture finger and wrist bending and the data is then sent to a computer via USB [148]. The device comes with rehabilitation software and uses novel training matching to help subjects improve motor control. An image of the device is on the following page.
The HandTutor is higher cost than the expected cost of the ATLAS system because it is not a bimanual system and it lacks the ability to calculate dynamic hand movement and orientation.

6.2.3 Hand and Glove Devices in Research
There have been numerous devices developed for rehabilitation that have never been commercialized. Most of the devices use either bend sensing or inertial sensors or use an off-the-shelf sensor system with a custom system.

There are several devices that use only bend sensors to capture finger bending data [149, 150]. One of these devices, the Shadow Monitor, was implemented with a pneumatic glove to measure finger bending of impaired subjects [151]. Other systems use camera tracking or other absolute tracking technologies to gather information about hand and arm position [152]. Another research device was the Rutgers Master Glove II [153]. The device uses custom pneumatic actuators and sensors placed on the palm to avoid wires at the fingertips. The direct-drive configuration of the actuators provides force to the tips of the fingers that connected to the device. The device has several advantages including force feedback and possibility of actuated finger exercises. For more information on the use of inertial sensors in rehabilitation see Section 5.2.3.
6.3 SYSTEM HARDWARE AND SENSORS

The ATLAS-BR has three distinct components: the gloves, the data acquisition system and the resting base. The gloves are the most important part of the system. They are the actual interface between the user and the rest of the VR system. The gloves consist of two sensors systems: bend sensors for finger angle and inertial measurement units for orientation and position estimation. The following sections outline the specifics of the system components.

6.3.1 BEND SENSORS

The ATLAS uses bend sensors to collect data on the bend angle of the user’s fingers. There are many off-the-shelf bend sensors that have been tested for use in wearable devices [154].

Bend Sensors from Flexpoint Inc. were chosen for the ATLAS because they are thin, low-cost and can be made into custom shapes [155]. Flexpoint sensors are potentiometers that consist of substrate (usually polyimide) coated with a bend sensitive ink. This proprietary ink is very hard and brittle. When the sensor is bent the ink on the sensor cracks but remains on the substrate, making measurements with the bend sensors very reliable with little noise. The sensors are typically 0.005” thick and can come in various lengths and widths. An image of the Flexpoint Bend Sensors is below.

![Image of the Flexpoint Bend Sensor]

**FIGURE 76: IMAGE OF THE FLEXPOINT BEND SENSOR [155]**

The sensors have a finite base resistance that can vary depending on geometry and design to range between 100Ω and 500kΩ. By changing the geometry of the sensor the resistance will
change, an increase in length will have a proportional increase to resistance. For more information about how the Bend Sensors were used in the ATLAS see Sections 6.3.4 and 6.3.5. For more information on the characterization testing for sensor see Section 6.4.

6.3.2 **ARDUINO MICROCONTROLLERS**

Arduino is an open-source electronics prototyping platform. The intended market of users for the Arduino includes artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Recently the Arduino has become more popular with researchers and product designers. The biggest advantage the Arduino affords the ATLAS is that the hardware is low cost and the software is open source.

Arduino microcontrollers can be programmed to receive input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with a computer via USB, Bluetooth, or Wi-Fi. The Arduino boards can be built by hand or purchased preassembled; the software can be downloaded for free. The hardware reference designs (CAD files) are available under an open-source license.

6.3.3 **ATLAS DESIGN OVERVIEW**

From its inception the ATLAS was based on some core ideas and sensor technologies. From research and interviews with area experts the design prototype was defined as a low-cost device for at-home use. Other attributes of the device include native bimanual configuration, sensors for finger bending, and sensors for hand and wrist orientation and position. A CAD rendering of the device can be seen in the figure on the following page.
The ATLAS design would include the sensors mounted on the hand by use of a garment or strap. The device would also have a base for a subject to rest between exercises and an electronics enclosure to contain the microcontrollers and filter and amplification circuits needed for the device.

6.3.4 ATLAS Version 1 and 2

The first and second versions of the ATLAS were created in the fall and spring of 2009 respectively by the capstone team mentioned previously. These first versions of the hardware were a necessary step in the evolution of the system. Images of the first and second versions of the ATLAS are on the following page in Figure 78.
The first version of the system had several limitations including bulky cables that connected the gloves to the computer, exposed wiring on several of the sensor connections, less flexible bend sensors, and the need for an external power supply. The second version of the system addressed several of these issues. The second version had all wiring covered and smaller, more flexible cabling. It also had a new glove design that allowed for more protection of the bend sensors. This system was used to create the current version of the system which is presented in the next section.

6.3.5 ATLAS VERSION 3

The third (and current) version of the ATLAS was designed fabricated in the fall of 2010 and spring of 2011. It contains several significant upgrades including: new microcontrollers, new cabling, new IMU, and streamlined software. The goal of this version was to reduce the size, cost, and power consumption of the device while maintaining and improving the features of previous versions.

The most significant change and improvement made between the second and third versions of the ATLAS was the introduction of microcontrollers to do the calculations and data acquisition.
for the system. The previous system required the use of a National Instruments DAQ card that was mounted on the mother board of a PC. That DAQ card was the most expensive component of the system. It also required that the system run off a desktop computer with a tower and had a large electronics box to house pin-outs for the cables, simple filter circuits, and power supply wiring because the system was powered externally. In the current version there are three microcontrollers. There is one on each glove contained in the IMU mounted on the back of the hand. These microcontrollers calculate the orientation and position values of the hand from the IMU data. The other microcontroller in the system is in the tabletop enclosure and receives the input from the gloves (microcontrollers and bend sensors), does the necessary calculations and then outputs the complete system data to the computer via USB the figure below is a diagram of the global system layout.

![Diagram of the global system layout](image)

**FIGURE 79: DATA FLOW DIAGRAM OF THE ATLAS V3**

The ATLAS v3 software needed to be completely rewritten to accommodate the change in data acquisition and calculation hardware. This change reduced the size and power consumption of the device and lowered the cost of the components significantly. New enclosures had to be designed for the hardware and they were fabricated using rapid prototyping. Images of the new enclosures are on the following page.
The change to the microcontrollers required all new wiring for the gloves to the microcontroller unit but allowed for the system to be powered via the USB connection to a laptop or desktop computer. The image below shows the entire ATLAS v3 system.
The ATLAS v3 has four bend sensors on each glove (thumb, pointer, ring, and middle finger). The IMU on the back of the hand is the 9DOF Razor IMU (discussed further in the previous chapter). The system interfaces with a computer via USB.

6.4 BEND SENSOR FILTERING, AMPLIFICATION, AND MECHANICAL TESTING

This section contains information on the characterization testing that was performed on the bend sensors. The bend sensors mounted on the fingers of the ATLAS change in resistance when bent in a single direction. The following sections present the theory, equipment, protocol, and results of this testing.

6.4.1 BEND SENSOR TEST BEDS

To characterize and test the bend sensors two test beds were developed in collaboration with Richard Ranky. The first test bed was designed for the sensor to be wrapped around different bend diameters from 1-6 cm. This distance was chosen based upon anthropomorphic data for the hand. Images of the test beds are shown below.

FIGURE 82: IMAGES OF THE BEND DIAMETER TEST BED (LEFT) AND THE BEND ANGLE TEST BED (RIGHT)
The second test bed that was created was a modified goniometer. It allowed for the bend sensor to be mounted and then bent to a desired angle instead of a bend diameter. Both of these test beds were fabricated using rapid prototyping (stereolithography).

### 6.4.2 Bend Sensor Data Filtering and Amplification

The Flexpoint bend sensors require filtering and amplification to be used in the ATLAS system.

The first step of this process was to characterize the bend sensors resistance at varying bend radii. To find the initial resistance of the bend sensors and test the linearity of their response a voltage divider was created. In order to get an accurate measurement of resistance without noise a test bed circuit was designed with a high accuracy resistor (59k Ω with a 0.1% deviation), a diagram of the circuit can be seen below.

![Circuit Diagram of the Resistive Sensor Test Bed Circuit](image)

**FIGURE 83: CIRCUIT DIAGRAM OF THE RESISTIVE SENSOR TEST BED CIRCUIT**

The circuit uses the differential voltage measurement of both a known and unknown (in this case the sensor) resistance. The knowledge of the input voltage and the chosen resistance allows for the calculation of a ratio between the voltages and resistances because the current over each resistor remains equal. The final equation for calculating the sensor resistance is below.
EQUATION 9: VOLTAGE DIFFERENTIAL EQUATION

\[ R_{\text{sensor}} = \frac{V_2 R_{\text{known}}}{V_1} \]

The resistance of the sensor can be calculated as the multiplication of the differential voltage over the sensor and the known resistance divided by the differential voltage of the known resistance. This circuit and the equation were implemented into a data acquisition system using LabVIEW and a NI USB-6216 DAQ card. The bend sensor test measures an analog voltage and then calculates the resistance of the bend sensor. Due to the nature of the analog signal it is important that it is filtered both by a hardware passive low-pass filter and a software filter. The circuit diagram of the low-pass filter is below.

![Circuit Diagram of Passive Filter](image)

FIGURE 84: CIRCUIT DIAGRAM OF THE PASSIVE FILTER FOR DIFFERENTIAL BEND SENSOR SIGNALS

The hardware filter is a simple passive low-pass filter. The two filters are connected at a point P that is not a shared ground. This is due to the nature of the differential signal. The capacitance and resistor values will vary based upon the acquisition rate of the data.

The Flexpoint bend sensors are known to have an issue with holding their resistance [149]. A static test was performed on a sensor to find the resistance loss over time. The chosen duration for the test was 60 minutes and the results of the raw signal with the low-pass filter can be seen in the graph on the following page.
The sensor lost close to half of its resistance over the course of the 60 minutes. The initial drop between 5 and 7 seconds can be attributed to the force that was applied to the sensor to bend it in place on the test bed. However, the resistance still dropped by approximately 25% over the rest of the 60 minutes. Thus a new circuit was needed to amplify and filter the signal. A differential amplifier was chosen based on the specifications for the bend sensor from the manufacturer. The circuit diagram for the differential amplifier is shown below.

The circuit contains a differential amplifier and a passive low-pass filter. The values of the resistors needed to be tested to determine the best configuration. Testing of the circuit with different resistances (380Ω to 2mΩ) in the R1 position can be seen in the graph on the following page.
Using the curves of the test bed it was found that the sensor saturated at high bend radii. Because of the nature of the human hand and the possibility of spasticity in users it was thought to be important that the sensors do not saturate with large bending. The equation that determines the output of the circuit is below.

**EQUATION 10: CIRCUIT EQUATION FOR THE AMPLIFIER OF THE ATLAS BEND SENSORS**

\[
V_{\text{out}} = \frac{(R_3 + R_2)R_B}{(R_B + R_1)R_2} V_{\text{in}}
\]

In the equation above the \(V_{\text{out}}\) is the voltage out from the sensor and \(V_{\text{in}}\) is the input voltage from the source. \(R_B\) is the bend sensor resistance which varies with finger movement. The \(R_2\) and \(R_3\) resistances remain constant in the circuit and are used as a ratio. \(R_1\) creates a voltage divider with the bend sensor and must be tuned to the resistance range of the bend sensor. Using the above equation resistance values were calculated using a scale of 0-5 volts and 25-3000\(\text{k}\Omega\). The theoretical values for the voltage can be seen in the graph on the following page.
This curve can be fit very easily with a polynomial of power two as was expected. The bend sensor was once again tested statically on the bend diameter test bend for 60 minutes. The results of this test are seen in the graph below.

This test was completed using the Arduino microcontroller for data acquisition leading to the less uniform signal (due to the analog to digital converter). The new circuit removes the re-
istance (or voltage) drop and provides a more controllable range of bend diameters. The next step was to test the sensors mechanically, which is discussed in the next section.

6.4.3 Bend Sensor Mechanical Testing

The bend sensors were mechanically tested using a modified goniometer. The sensor was mounted on the goniometer and then data was collected at each angle for 5 seconds and then averaged. Each sensor was tested at 5° intervals between 0° and 120°. The graphs below contain the results for the left and right hand sensors.

FIGURE 90: GRAPH OF MECHANICAL TESTING FOR THE RIGHT HAND BEND SENSORS

FIGURE 91: GRAPH OF MECHANICAL TESTING FOR THE LEFT HAND BEND SENSORS
The difference in the resistance between the sensors on both hands can be attributed to the mounting of the sensor position on the glove and also the sensors themselves. The sensors are created in batches so the calibration of the sensors can be done during manufacturing for many devices at once. These results match the theoretical values found from the amplification circuit (Figure 88) in terms of shape and amplitude of the sensor response and were used to analyze the healthy subject data collected in the testing described in the next section.

6.5 Healthy Subject Bend Sensor Characterization Testing

Once the mechanical tests were complete with test beds characterization, testing was completed (IRB# 10-12-14) with a total of 24 subjects (11 males, 13 females; mean age=25.5±6.9 years) with a broad range of hand sizes. The following sections contain information on the protocol for the testing and the results from the testing for the bend sensors for more information on this testing see the following two sections. The testing was completed in the Neurorehabilitation Research Laboratory at Northeastern University directed by Dr. Holden with the assistance of Doug Murray and Lindsey K. Dick.

6.5.1 Bend Sensor Testing Protocol

After subjects completed the inclusion questionnaire (Appendix Section B.2) and the informed consent form (Appendix Section B.1) hand tracings were obtained and marked with the following landmarks: styloid processes of the radius and ulnar; heads of the first and fifth metacarpals, metacarpal-phalangeal (MCP), proximal-interphalangeal (PIP) and distal-interphalangeal (DIP) joints of each finger. A sample of a hand tracing is in the figure on the following page.
This testing consisted of placing each subject’s hand on a series of templates for static angle measurements. Each of these templates were designed to place the subject’s fingers in a desired position at which time measurements of the MCP and PIP joints were taken of the index, middle, and ring fingers using a manual goniometer. Measurements of the thumb were also performed where the MCP and IP were measured. The image below is of the static templates and goniometers used in the testing.
Once the goniometer measurements were collected, the subject was asked to maintain the static position for five seconds while the bend sensor voltage was recorded. All twenty-four subjects completed the bend sensor protocol with their right hands and twelve subjects were asked to complete it with their left hand as well. Images of this static testing are below and on the following page.

FIGURE 94: IMAGE OF THE RIGHT HAND INDEX FINGER BASELINE BENDING TEST

FIGURE 95: PHOTOGRAPHS OF RIGHT HAND THUMB TO SMALL FINGER BENDING TEST (LEFT) AND THE 45°-45° BENDING TEST (RIGHT)
For each of these static measurements the subject was asked to place the target joint as close to the correct angle as possible and have all other joints be as close to neutral as possible. The list of tests for the bend sensor testing is below.

**List of bend sensor tests:**

1. Baseline: all joints fingers and thumb at full extension (0° or 180° on the goniometer)
2. 45° MCP: all fingers at 45° template for MCP Flexion
3. 45° PIP: all fingers at 45° on template for PIP Flexion
4. 90° MCP: all fingers at 90 degrees on template for MCP flexion
5. 90° PIP: all fingers at 90 degrees on template for PIP flexion
6. Max Thumb MCP: instruct subject to flex thumb maximally at MCP, the goal is isolated MCP (subject may use passive motion to obtain maximal flexion)
7. Max Thumb IP: instruct subject to flex thumb maximally at IP, the goal is isolated IP
8. Max Thumb MCP/IP: instruct subject to flex thumb maximally at MCP and IP
9. Thumb 45° MCP / 45° IP: thumb MCP and IP at 45 degrees flexion
10. Thumb-small finger: thumb opposition to pinkie fingertip
11. Thumb-index: Thumb opposition to index fingertip
12. Max MCP / IP: instruct subject to passively flex thumb maximally in IP and MCP
13. 90° MCP / 90° PIP: place subject into hand split device so that PIP and MCP of digits 1-3 are at 90°. Have this done in the vertical position.
14. 45° MCP / 45° PIP: place subjects hand so that PIP and MCP of digits 1-3 are at 45°. This will also be done in a vertical position.

15. Dynamic: dynamic open close hand: instruct subject to open and close hand fully for five seconds. The subject should start with the hand in a comfortable fist, all joints will then be measured before opening of the hand begins. The subject will then open and close their fist for five seconds.

The testing took place between September and November of 2011. The results for the bend sensor testing are contained in the following section.

6.5.2 Bend Sensor Testing Results

Using these tracings, hand sizes were calculated using a method similar to that used by the US Army in their development of an anthropomorphic database [132]. The method was a calculation of area based on the length measured from tip of the middle finger to the center point between styloid processes and the width which was measured from the fifth MCP to the second MCP. For males, mean width was 9.4±0.5cm and length was 20.2±0.5 cm; for females, width was 8.4±0.5 cm and length 18.1±1.1cm. Hand size distribution by gender can be seen in the figure below.

![Hand Size by Gender](image)

**FIGURE 97: CHART OF HAND SIZE DATA BY GENDER**
Subjects in the sample had a wide spread except for men with smaller hands which were under represented and one woman who had much larger than average hands. This information was used to better understand the bend sensor measurements.

An analysis was completed on the goniometric measurements to gauge the accuracy of the protocol and choices of static measurements as well as to determine potential errors in bend sensor tests. The chart below contains information about those measures.

The templates that were used for the testing proved to be accurate. The baseline measurements for both joints had averages within one degree of zero with standard deviations between one and two degrees. All MCP and PIP tests where the intended angle was 45° had accurate results except for the right ring finger in the 45° MCP / 45° PIP test which had a low average of $41^\circ \pm 4.58^\circ$. The most accurate test was for the 45° MCP which had averages of $45.14^\circ \pm 1.47^\circ$, $44.79^\circ \pm 2.28^\circ$, and $44.67^\circ \pm 4.03^\circ$ for the index, middle, and ring fingers respectively. The 90°
tests were lower due to the design of the template and the constriction of the glove on the subjects.

Due to the large range of hand sizes within the sample a trend line for all the raw data for the bend sensors was calculated to check against the amplification equation for the bend sensors. To that end an exponential trend was chosen and can be seen in Figure 99 below.

The exponential trend results follow the shape of the amplifier equation and the results found from the mechanical testing (Figure 90). However due to some of the data having artifacts the amplifier equation must be changed to create a more linear curve for the bend sensor angle. The raw bend sensor data was investigated for other trends due to gender or specific test. The charts on the following pages group the raw data by test for all the sensors on the right hand.

**FIGURE 99: GRAPH OF THE EXPONENTIAL ANGLE TREND VS. VOLTAGE FOR RIGHT HAND**
FIGURE 100: RAW DATA CHART OF RIGHT HAND INDEX FINGER BENDING BY TEST

FIGURE 101: RAW DATA CHART OF RIGHT HAND MIDDLE FINGER BENDING BY TEST
Each of the sensors on the right hand had very tight grouping for the baseline test as should be expected for the bend sensors. The tests also seem to vary based on the exponential nature of the amplifier equation. The right ring finger sensor was the noisiest, which is mostly likely due to a very poor fit of the sensor to the finger. Different sized gloves, better placement of the sensor, and a different method of closing the glove could solve the issue that is present with
the right ring finger sensor. As a pseudo analysis for size the data can also be view by gender in
the charts below.

FIGURE 104: RAW DATA CHART OF RIGHT HAND INDEX FINGER BENDING BY GENDER

FIGURE 105: RAW DATA CHART OF RIGHT HAND MIDDLE FINGER BENDING BY GENDER
As mentioned previously the fit of the ring finger led to unstable data values and these were seen in both the men and women subjects. The thumb sensor follows the same trend of previous sensors where there appears to be no correlation between hand size and the data collected by the sensor with the current filter circuit values.
The bend sensor characterization testing was required to validate the bench testing discussed previously. During the testing it became apparent that the ATLAS required redesign to address the filter and amplification circuit as well as the placement of sensors on the back of the hand. These modifications are presented in the next section.

6.6 POST SENSOR CHARACTERIZATION TESTING MODIFICATIONS

After the testing was completed several issues with the system needed to be addressed, the first was the bend sensor response at smaller angles. This was an issue because it meant that subjects bending in the 0°-45° range did not register a change in the bend sensor signal. This is due to the Arduino having a minimum voltage response to analog input signals. To increase the response at the lower angle levels an additional resistor was added in series with the bend sensor essentially creating a “pre-load” for the sensor and allowing it to have a higher effective range. The new circuit diagram can be seen below.

FIGURE 108: CIRCUIT DIAGRAM OF THE NEW BEND SENSOR AMPLIFICATION AND FILTER CIRCUIT

The other adjustment made to the system was to reinforce the mounting of the bend sensors as several bend sensors became loose during the healthy subject testing. The stronger mount-
ing will give a more reliable fit to the sensors and reduce the change in signal based on the hand size that was found.

6.7 Future Work

Future work for the ATLAS requires another version of the hardware so that significant improvements can be made. These improvements include a custom printed circuit board for the hand IMUs and base microcontroller. This would reduce cost (with quantity) and make the system more robust. More bend sensors or a custom bend sensor design would give the system better sensing capabilities. Another possible improvement would be to use a faster communication protocol such as SPI or I2C for the communication between the microcontrollers. If the next version of the ATLAS introduces more IMUs (as discussed in the previous chapter) then making the system wireless (Bluetooth) would be advantageous. Bluetooth would allow for a serial pipe or the use of a Human Interface Device (HID) driver which would make the device compatible with more computers, tablets, and even smart phones. Another iteration of the hardware would also necessitate new enclosures for the IMUs to make the connections and sensors more robust on the back of the hand.

6.8 Summary

In this chapter the low cost in-home glove system for rehabilitation, the ATLAS, was introduced. The system has sensors to monitor finger bending, hand orientation, and hand position which were discussed in the previous chapter. The bend sensors from Flexpoint chosen for the system had to be characterized before they could be implemented into the system. This required the testing of the resistance of the sensors which lead to the design and implementation of a circuit for amplification and filtering. The sensors were mechanically tested before being tested by 24
healthy subjects as part of the ATLAS testing. The testing showed some issues in the mounting of the bend sensors on the hand and a lack of response from the sensors at low angles. This has been addressed since the tests have been completed. However, the ATLAS will need another version of the hardware because there are several large improvements that can be made to the system.
CHAPTER 7: PATIENT SPECIFIC VIRTUAL ENVIRONMENTS

SCREEN CAPTURE OF A POSITIVE FEEDBACK ANIMATION IN AN ATLAS VIRTUAL SCENE
### 7.1 Introduction

This chapter outlines the VR software that was created for the ATLAS. The creation of this software started in the summer of 2011 and continues at the time of this writing. This chapter relies on background from Chapters 2 and 3 as well as the information in Chapters 5 and 6 about the ATLAS.

### 7.2 Previous Research

There has been research previously completed in this area by the author. The following sections outline this research and how it can be applied to the ATLAS system.

#### 7.2.1 Virtual Interactions for Multiple Users

Chapter 3 presented several game design paradigms but one not addressed was multiplayer games. There are four categories of virtual interactions: competitive, cooperative, versus, and mixed. A figure displaying virtual interactions can be seen below.

![Diagram of Multiple User Virtual Interactions](image)

**Figure 109: Diagram of Multiple User Virtual Interactions**
Either singly or in unison these interactions appear in every multiplayer game. Cooperative is the type of interaction where two or more players work together to complete a common goal or task. Counter-operative or versus interactions are defined by one user working against another user for a common goal only one can obtain. Competitive interaction is where users complete the same tasks and the same goals for competitive score, without any direct interaction between the players. Mixed interactions can be defined as interactions where users work together to complete a common goal but are scored competitively or any other combination of the first three interactions.

7.2.2 Virtual Reality Scenes with the P5 Glove

One proof of concept virtual environment was developed. The virtual scene used a competitive virtual interaction between two users. This scene is designed to allow practice of active grasp (pincer, 2-5 fingers with thumb) and release and maintained grasp with active supination (hand turning toward palm up). These movements are essential to improved hand function in users with stroke.

The mechanic was divided into phases which were timed and counted separately, in order to provide more feedback to subjects about their performance if the entire trial was not successful. These phases were 1) start position to grasp of lid; 2) grasp plus supination to threshold value and transport toward the right side of screen; 3) pronation, transport back to left to position lid on pot, then release of fingers (finger extension to a threshold value). The ability to count successful phases and trials and record time for each was incorporated in the design for the scoring function. Each element can be displayed separately to subjects to provide feedback about performance either during the session or after. These elements are 1) count for number
of successes for each phase; 2) count for number of successful trials (all 3 phases completed); 3) time for each phase, and 4) time for each trial. Time can also be displayed as a mean for block of trials, with the number in block adjustable. A schematic of this movement mechanic can be seen below.

In this scene, the user must first grasp the lid (at the correct grasping position the lid turns blue, once grasped the lid turns green), next s/he turns the lid (hand supination) while maintaining grasp and transports the lid to the right. Upon successful grasp plus supination and transport, the lid turns red. If the user loses grasp while turning, the lid drops back to the start position on the pot and returns to the neutral grey color. Supination threshold for success is currently set to 45 degrees, but is adjustable. Finally, the user pronates and transports the lid back to the pot and releases the grasp by extending the fingers, at which point the lid turns grey, and the trial is counted as a success. The final prototype software was created using the Panda 3D game engine. A screen capture of the final prototype virtual scene can be seen on the following page.
The virtual scene systems include a collision detection system for the lid of the front left pot and the plane that can be seen in the screen capture. The text and score mechanic was displayed on the screen in 2D while the rest of the scene is presented in 3D. For more information see [10].

7.2.3 CUSTOMIZABLE EXPLORATION GAME FOR THE HERRI

This software was created for use with the HERRI (Appendix Section A.2). In order to create a software package that gives the practitioner the most flexibility it is necessary to have a way for the practitioner to create unique virtual reality scenes for users. To facilitate this, a separate software program is needed.
The virtual scene design software needs to be simple for the practitioner to use and also have the data from it be read by both the virtual reality software and the LabVIEW control software. In order to do this and also have the data be savable the design software must create a file that is read by the other programs. The simplest solution is to have the design software write a text file that contains the data necessary for the other programs to recreate the scene. The data that each scene will need includes the position of the start and end points for each line segment as well as their “feel” (spring, damper, etc.), the equation for each line segment, and the “feel” of each line segment.

The practitioner will be able to create the scene in the design software much like a paint or art program. Once the practitioner is satisfied with the design they will save it in the design program. The design program will then create a text file that contains the necessary data for that
scene. This will be done by calculating the start and end points of each segment as well as its equation. The “feel” was chosen by the practitioner during design.

Once the data is calculated the design software will write it to text in a way that is both readable in a normal text format and also readable by the other programs. The virtual reality software will read in the text file and then create the scene geometry based on the data in the file. The control software will also read in data from the text file of the scene and create the necessary feel and forces that the scene requires. Because the design software creates a text file for each scene it is savable and reusable by the practitioner.

The final prototype virtual scene was created in the Panda 3D game engine. A system in Panda 3D was used to create the background of the maze dynamically. This allows for multiple mazes to be setup and used easily without extensive changes to the code. Also an algorithm was created to place the red and green items on the maze in a random position so that each time the maze is used it offers a different experience to the user. A screen capture of the final prototype can be seen below.

![Figure 113: Screen Capture of Final Software Prototype Simple Maze](image_url)
The player’s avatar in this virtual scene is represented by a blue circle with a lighter blue rectangle on top. The blue color was chosen to be different from the red and green items that are spread throughout the maze. The lighter blue rectangle on the avatar is used to show the direction that the user is moving.

The green items on the maze are placed either randomly or using a separate text file of x and y coordinates. The avatar is meant to collide with the green items to increase the user’s score and motivate them to do movements using the device. The red items act in the opposite way of the green ones. They too are placed in the maze either via random generation or by a separate text file. The red items lower the score of the user and the avatar should avoid them, forcing the user to go around and perform rehabilitation movements on the device.

The user interface for the prototype software is basic and simple so that the user can easily understand both their score and the time remaining. The time is displayed as a rectangle that decreases in size and changes color based on the time remaining in the session. Below is a screen capture of the advanced maze in the final prototype software.

![Screen Capture of the Advanced Maze for the HERRI Software](image)

**FIGURE 114: SCREEN CAPTURE OF THE ADVANCED MAZE FOR THE HERRI SOFTWARE**
The advanced maze above was the first maze implemented into the final prototype software. The design was created from practitioner feedback on what path density and level of complexity they wanted. The green and red items that are displayed can be randomly placed or placed by an eternal text file (further explanation in the next section). The diagonal paths that were implemented into this maze presented the largest challenge for subjects and it is because of this high level of difficulty that simple maze was created and implemented.

7.3 VIRTUAL SCENE DESIGN
Four virtual scenes for rehabilitation have been designed for the ATLAS. The two exercises chosen for the ATLAS are flexion-extension of the wrist and grasp and release of the hand. The scenes are designed to be paired so two scenes use the same rehabilitation exercise but different virtual representations. The two different representations are direct and indirect, which along with the scene components and design are explained in the following sections.

7.3.1 DIRECT “LINKED” MOTION
Direct or linked motion is a virtual representation of a movement mechanic that matches the movement as realistically as possible including a realistic avatar for the player. Direct is the most common method of movement mechanics in rehabilitation systems. Several previous projects from the BML have used direct motion. The figure on the following page shows a generalization of the motion for the direct virtual scenes of the ATLAS.
Scene A depicts the movement for wrist flexion-extension and scene B depicts the movements for grasp and release. In scene A the user is represented by a virtual forearm and hand and their flexion-extension is replicated in the virtual scene. Scene B is designed in the same way except with grasp and release of the hand. The user is again represented by a virtual forearm and hand and when grasping the player crumples paper and then on release it is tossed into a waste basket.

Direct motion has been proven effective because of its simplicity both in design and how intuitive it is for the user. However, the possibility for boredom and the narrow scope of the direct VRs can detract from rehabilitation as there is insufficient research on indirect scenes, explained in the next section.

7.3.2 INDIRECT OR “ABSTRACT” MOTION

Indirect or abstract motion is the opposite of direct motion. With indirect motion another virtual representation is chosen for the exercise with the intention of motivating the patient more than a direct motion scene. This is done by amplifying the feedback or using other game design
elements such as escapism. The figure below shows the motion representation of the indirect virtual scenes of the ATLAS.

![FIGURE 116: INDIRECT MOTION SCHEMATICS FOR ATLAS VR SCENES](image)

Scene A depicts the movement for wrist flexion-extension and scene B depicts the movements for grasp and release. In scene A the player is represented by 3D model of a penguin, in scene B the player is represented by a 3D model of a man doing push-ups.

The indirect method hinges upon creating a VR that is intuitive for the user but not over simplified so that it lowers motivation. In the virtual scene visualized on the left of Figure 116 a model moves laterally based upon the motion of the wrist. The right side of the figure shows an indirect motion that would correspond to grasp and release.

### 7.3.3 Game Components of ATLAS Virtual Scenes

Recall from section 3.2 the model for game components that was presented. The model consists of four elements: player, world, mechanic, and goal. The design of the scene in the previous sections follows this model with detail in the table on the following page.
TABLE 15: GAME COMPONENTS OF ATLAS VIRTUAL SCENES

<table>
<thead>
<tr>
<th>Virtual Scene</th>
<th>Player</th>
<th>World</th>
<th>Mechanic</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct A</td>
<td>Forearm and Hand 3D model</td>
<td>Desk and surrounding area</td>
<td>Flexion and extension of the wrist</td>
<td>Complete repetitions of the mechanic</td>
</tr>
<tr>
<td>Direct B</td>
<td>Forearm and Hand 3D model</td>
<td>Desk and surrounding area</td>
<td>Grasp and release of the hand</td>
<td>Complete repetitions of the mechanic</td>
</tr>
<tr>
<td>Indirect A</td>
<td>Penguin</td>
<td>Fictional iceberg and ocean</td>
<td>Flexion and extension of the wrist which moves the penguin right</td>
<td>Cause the penguin to dive into the water</td>
</tr>
<tr>
<td>Indirect B</td>
<td>Workout Guy</td>
<td>Fictional gym</td>
<td>Grasp and release of the hand which causes the guy to do a push-up</td>
<td>Cause the man to get pumped up</td>
</tr>
</tbody>
</table>

The differences in these scenes are based on the direct or indirect motion. The world, player, mechanic, and goal are dictated by that design.

7.3.4 Teaching Cycle Design for Indirect Scenes

The indirect scenes are a more innovative approach to adapt to a teaching cycle because they resemble mainstream consumer games. The learning cycle was introduced in section 3.4 and is composed of five steps: introduction, learning, testing, challenge, and reuse. The teaching can once again be seen in the figure below.
Other game mechanics must be introduced to design a teaching cycle for the indirect virtual scenes. For Indirect A, the penguin scene, modifications can be made to either time the cadence of the flexion-extension or the angle change that must be achieved to move the penguin to the goal. For Indirect B it also makes sense to modify the cadence of the exercise or the time the player must hold either the grasp or release segment of the movement.

The first step in the teaching cycle for Indirect A would be to introduce the mechanic by either an introduction animation or from the curator of the test. The learning phase would then be to allow the user to simply flex and extend their wrist until they have reached their goal (seeing the penguin dive). The testing step in the teaching cycle adds a timing mechanic so that the player now has to complete the scene within a time limit. This manifests itself in having the penguin begin to slide backwards on the ice if the user does not successfully do the exercise to the correct angles. The challenge phase would include the introduction of a gap in the ice where the player must time their movement to correctly reach the goal. Once the user had completed the previous phases they are now in reuse.

A teaching cycle for Indirect B would be similar to Indirect A. The introduction step of the cycle would be an introduction animation or curator explanation. The learning part of the cycle would allow the user to grasp and release their fingers until the scene is finished successfully. The testing step of the cycle for this virtual scene would include an accuracy mechanic where the user would have to follow a virtual guide grasping and releasing of the hand within an accuracy threshold. The fourth phase, challenge, would increase the difficulty of the scene by modi-
fying the speed and accuracy of the virtual guide. The last step of the cycle, reuse, will be achieved once the user successfully completes the previous steps.

### 7.4 Patient Specific User Interface Design

The interface for any virtual system is composed of the input from the user and the output from the system. The input for the ATLAS is the signals from the IMU and bend sensors. The output is the virtual scene including all visual and auditory feedback. The graphical user interface that is present with the virtual environment can be customized based on the player’s type of the user. The table below outlines the different types of feedback that could be used with different types of players.

<table>
<thead>
<tr>
<th>TABLE 16: PATIENT SPECIFIC USER FEEDBACK BASED ON PLAYER TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Player Type</strong></td>
</tr>
</tbody>
</table>
| Hedonist | ▪ Strong enjoyment incentives  
▪ Self-improvement  
▪ Approachability | ▪ Personal high score  
▪ Percentage completion |
| Competitor | ▪ Self-improvement  
▪ Competition with others  
▪ Positive mood | ▪ Overall high score  
▪ Time  
▪ Percentage completion  
▪ Position feedback |
| Organizer | ▪ Self-improvement  
▪ Positive mood  
▪ Approachability  
▪ Self-awareness  
▪ Self-efficacy  
▪ Judgment | ▪ Personal high score  
▪ Positive feedback  
▪ Ghost/guide mechanic |
| Rebel | ▪ Competition with others  
▪ Flexibility  
▪ Social judgment  
▪ Winning above all else | ▪ Overall high score  
▪ Timing |
| Team Player | ▪ Self-improvement and efficacy  
▪ Approachability  
▪ Positive mood  
▪ Social judgment  
▪ Competition with others | ▪ Ghost/Guide mechanic  
▪ Personal high score  
▪ Grading |
| Socializer | • Very social  
• Self-improvement  
• Positive mood  
• Approachability  
• Not interested in competition | • Personal high score  
• Overall community scores  
• Grading  
• Timing |

Player types were introduced in Section 3.8 and users could answer either a questionnaire or play a specific virtual scene to determine the type of player that they are. Then the interface and feedback can be customized to allow better immersion and motivation.

7.5 IMPLEMENTATION OF THE ATLAS VIRTUAL SCENES

Previous software for the ATLAS was created using the Panda 3D game engine. However, there were several limitations to its capabilities so another engine, Unity, was chosen to create the virtual scenes. The scenes presented in the previous sections were implemented into a new Unity project and a new addition to the DAVE software written in C# was used for serial communication between the ATLAS and the computer.

The first scene to be created was a simple calibration scene that allowed the user to see visual feedback of the ATLAS sensors in the form of a model of a hand and forearm. This scene is used to get the subject acclimated to the system and the types of input they can use. This scene is also used to help calibrate the system to the user. A screen capture of this scene is in the figure on the following page.
The next scene implemented was Indirect A, which included a penguin waddling across the screen as the user move their wrist in flexion and extension. A screen capture of this scene is below.

FIGURE 118: SCREEN CAPTURE OF THE ATLAS CALIBRATION VIRTUAL SCENE

FIGURE 119: SCREEN CAPTURE OF THE INDIRECT A (PENGUIN PANIC) ATLAS VIRTUAL SCENE
The last scene to be implemented to date is Direct B. In this scene the user grasps and releases their fingers and the motion is displayed on a virtual hand model. The figure below is a screen capture of this scene.

![Figure 120: Screen Capture of the Direct B (Grasp and Release) ATLAS Virtual Scene](image)

Unity has proven to be a very capable engine for the creation of virtual scenes and the other two scenes presented previously will be implemented. Other future work for the ATLAS software is presented in the next section.

### 7.6 Future Work

The next step in this work would be to test the virtual environments with first healthy and then impaired subjects. Healthy subjects could be tested first to ensure that the virtual scenes work and are intuitive enough for the users. Impaired subjects could then be tested on the virtual scenes to examine the patient specific user interface, the teaching cycle, and direct vs. indirect virtual environment design. To overcome improvement due to familiarity tests should include...
unstructured time for the subject to get comfortable with the system [156]. This should be done before the test of very specific tasks is started.

Improvements to the software include better graphics and more varied virtual scenes. Other features that could be added to the software are multiplayer scenes, an achievement system to track progress, and networked scenes to allow for telemedicine applications. There are some concerns about making the system online namely that more research into the areas of access of medical information, safety of all medical data on the internet, and standardization of virtual reality rehabilitation is needed.

7.7 SUMMARY

Previous work has been completed on the creation of patient specific virtual environments for both the P5 Glove with a MUVER and the HERRI with a customizable maze exploration game. This chapter presented the use of direct vs. indirect mechanics in virtual environments as well as the use of teaching cycles and patient specific user interfaces. Further testing needs to be completed on the virtual reality topics presented in this chapter to test their impact on rehabilitation of the user.
CHAPTER 8: CONCLUSIONS

ATLAS PATIENT SPECIFIC VR SYSTEM GLOBAL DATA SCHEMATIC
8.1 INTRODUCTION
The introduction chapter of this document presented the significance and need for the work. In this final chapter the following sections outline the major points and findings and summarize the body of work and the future work.

8.2 GAMES FOR REHABILITATION
The use of virtual environments and games for rehabilitation has increased in recent years with improving technologies and increasingly receptive patients. There are many issues with the creation of games for rehabilitation. One is the classification of what a game consists of, something addressed in Chapter 3 of this document. Other models of game components were presented and then a new model was defined which can be seen in the figure below.

![Figure 121: Schematic of Distilled Game Elements](image)

Game design is in its infancy as an academic field and as such there are many techniques that have been applied in the creation of games that have not been analyzed. Teaching cycles are an example of this. A teaching cycle is used by a designer to introduce the player to a new mechanic in a game. Teaching cycles are used in many games and are the product of understand-
ing how a player learns (motor learning in this case specifically) and what is needed for a positive experience (interest curves, flow, etc.). A teaching cycle for a well-designed game is below.

![Teaching Cycle Diagram]

**FIGURE 122: SCHEMATIC OF A TEACHING CYCLE FOR A GENERIC GAME**

Future work in the area of games for rehabilitation includes testing the concepts presented and implemented in Chapter 7 of this document which includes the patient specific feedback models and the teaching cycle. A more general look at the current state of the art would indicate that there is still much theory and results needed as everything thus far has only scratched the surface.

**8.3 INERTIAL MEASUREMENT UNITS IN REHABILITATION DEVICES**

Inertial measurement units (IMUs) have been used in rehabilitation for over thirty years. IMUs consist of accelerometers, gyroscopes, and/or magnetometers. They are used to estimate position and orientation of objects. The technology in the sensors has progressed to the point where many consumer electronic devices include IMUs or the sensor components of IMUs. In the work presented here two different IMUs were used in two devices. In the VRack the 5DOF IMU was used and in the ATLAS the 9DOF Razor IMU was used, they can be seen in Figure 123.
The VRack 5DOF IMU is called such because it has five degrees of freedom (three-axis accelerometer and two-axis gyroscope). The IMU was used to estimate the pedal angle. A Discrete Kalman Filter (DKF) was implemented to do the calculation. The 9DOF Razor IMU has nine degrees of freedom (three-axis accelerometer, three-axis gyroscope, three-axis magnetometer) which allows it to measure yaw. This IMU has an on-board microcontroller that was used to program an algorithm relying on direction cosine matrices to estimate orientation of the hand. Results from testing these IMUs were presented in Chapter 5.

Future work in this area includes further testing of both IMUs for robustness in their algorithms and repeatability. The VRack has the future work of a new hardware version that would allow for better sensor placement and new software to allow calculation of the DKF in real-time. The ATLAS will require a different solution for position estimation (possibly adding an IMU to the forearm and one to the upper arm) and research into the orientation of the hand should be performed to compare using a Kalman Filter with the direction cosine matrix currently implemented.
8.4 Rehabilitation with the ATLAS and Patient Specific Virtual Environments

The ATLAS is a low-cost, in-home, bi-manual glove rehabilitation device. It consists of potentiometer bend sensors on the fingers and the 9DOF Razor IMU mounted on the back of the hand. The system has had three iterations and its current version can be connected into any computer and is simpler and lower cost than previous versions. A CAD rendering of the current system is in the figure below.

![CAD Rendering of the Current Version of the ATLAS](image)

**FIGURE 124: CAD RENDERING OF THE CURRENT VERSION OF THE ATLAS**

The virtual reality software for the ATLAS was presented in Chapter 7. In that chapter direct and indirect mechanics were defined. From these designs several virtual scenes were implemented using the Unity game engine and algorithms from the DAVE software (Chapter 4). One of the virtual scenes is used to calibrate the device by giving the user visual feedback on their movements. A screen capture of this scene is in the figure on the following page.
These virtual scenes need to be tested with healthy and impaired subjects to verify their effectiveness and to better understand the indirect mechanics and learning cycles that can be implemented to improve outcomes. Other future work for the ATLAS includes a hardware revision that would include a more robust method of securing the bend sensors. More bend sensors also need to be added and other technical improvements include the IMU changes discussed in the previous section and making the system wireless.

Low-cost systems for in-home rehabilitation have the potential to improve outcomes post-stroke and this research presented the design and testing of the ATLAS with the hope of moving a few steps in that direction.
REFERENCES


S. Swink, Game feel: a game designer's guide to virtual sensation: Morgan Kaufmann, 2008.


[100] A. Chortis, P. Standen, and M. Walker, "Virtual reality system for upper extremity rehabilitation of chronic stroke patients living in the community."


5th Dimension Technologies Website. Available: [http://www.5dt.com/products/pdataglove5u.html](http://www.5dt.com/products/pdataglove5u.html)


APPENDIX A: PROJECT SUMMARIES

This appendix includes information about projects that the author was involved in from the Biomedical Mechatronics Lab at Northeastern University. Each section includes a summary and information about the device.

A.1 Virtual Reality Augmented Cycling Kit (VRack)

The virtual reality augmented cycling kit (VRack), shown in Figure 126 on the following page, consists of novel hardware components embedded with sensors that are used to enhance the use of a typical stationary exercise bicycle for a patient post stroke. The system is modular and can be mounted on any stationary bicycle. The complete instrumentation in the kit includes two sensitized handlebar modules (A in Figure 126), two instrumented pedal modules (C and D), a heart rate monitor (B), and additional electronics for signal conditioning, power, and connection to a laptop via a USB connection. The parameters monitored by these systems are communicated to a practitioner’s interface (E in Figure 126) to monitor quantitative performance of each parameter, as well as an interactive virtual environment (F) to provide visual feedback on the rider’s progress. The practitioner can also use their interface to manually adjust the gain of each sensor’s reflection in the virtual rider so as to attempt to keep up patient motivation or magnify the visual feedback to suit the patient’s kinetic or kinematic capabilities. Each parameter’s representation in the virtual rider can also be switched off or set to automatic pilot to suit the clinical needs.
In the virtual environment, a pace rider is displayed as a visual target to help motivate the patient and regulate their heart rate. The patient is instructed to remain next to the pace rider, which will sprint ahead or lag behind depending on the current heart rate of the patient relative to the exercise range set by the practitioner.

A.2 Hand Enhancement Robotic Rehabilitation Interface (HERRI)

The Hand Enhancement Robotic Rehabilitation Interface (HERRI) utilizes Electro Rheological Fluid (ERF) to create a hydraulic two degree of freedom (2DOF) device. The 2DOF that are possible with the device are grasp and release of the hand and supination and pronation of the wrist. A figure showing the global system is shown on the following page.
FIGURE 127: HAND ENHANCEMENT ROBOTIC REHABILITATION INTERFACE GLOBAL SYSTEM SCHEMATIC
The hardware of the HERRI consists of several custom pieces of hardware as well as off the shelf components. The system required two computers, a real time target system which ran a real time operating system and a host computer which ran the interface and the VR software. Connected to the real time target computer was the data acquisition hardware which was connected to the hand device’s sensors. The device had a strain gauge, load cell, and other sensors that collected data about the force, torque, and position of the user’s hand on the device. The device itself was composed of a rotary and linear hydraulic actuator and damper system. Other hardware included an amplifier, high voltage power supplies, and a custom made handle system. A figure showing the system setup as a desktop system can be seen below.

FIGURE 128: IMAGE OF THE HAND ENHANCEMENT ROBOTIC REHABILITATION INTERFACE

In the setup in the figure above the system is placed on the right side of the host computer. The user sits a comfortable distance from the device and uses their right hand to control the device.
The VR software outputs to the monitor to give the user and practitioner feedback on their performance. A close photograph of the device can be seen in Figure 129 below.

![Image of the device](image)

**FIGURE 129: TWO DEGREE OF FREEDOM ROBOTIC HAND REHABILITATION DEVICE**

The device consists of a linear and a rotary ERF actuator. The actuators were designed custom for the device by Brian Weinberg. Although it is possible to provide the assistive force with conventional actuators such as voice coils and hydraulic cylinders, the use of an ERF actuator has several superior features. First, it has a high force density (up to 500 N in a compact design) and second it has a high response rate to varying electric fields which enables smooth control of forces during rehabilitation sessions.

An ERF actuator was designed as the principle component in the device. The hydraulic actuator uses electro rheological fluid as its control fluid and performs exceptionally compared to standard hydraulic systems due to the proximity of its control valve. The control valve of the ERF actuator is situated inside the actuator (opposed to externally in a manifold) and it is controlled by varying the electric field across the valve. This change in electric field changes the yield stress
of the ERF fluid, which modulates the pressure drop across the valve. Since the valve is inside the actuator, close to the piston and the ERF reacts in milliseconds, precise force control is possible.

A.3 SQUID

The Squid system originated as a Capstone project in the fall of 2011 and was advised by Dr. Mavroidis and the author. The students involved in the project were Amy Schaffer, Alexandra Aas, Alexandra Moran, Trevor Lorden, Adam Morgan, Thomas Wilbur, Joseph Sheehan, and Kyle Peters. The project was a collaboration between the Mechanical and Industrial Engineering Department and the Creative Industries Program, both at Northeastern University. The goal of the project was to create a health monitoring system that interfaced with a smartphone.

FIGURE 130: SQUID SYSTEM COMMUNICATION SCHEMATIC
APPENDIX B: TESTING FORMS AND SURVEYS

The following sections present the testing forms and surveys discussed in this document.

B.1 ATLAS INFORMED CONSENT

Informed Consent Form - Northeastern University

Investigator Name: Maureen K. Holden, PT, MMSc, PhD, Department of Physical Therapy
Constantinos Mavroidis, PhD, Department of Mechanical & Industrial Engineering

Title of Project: “Development and Testing of a Low Cost Virtual Reality-based Glove (‘Smart Glove’) for Hand Rehabilitation”

Informed Consent to Participate in a Research Study
We are inviting you to take part in a research study. This form will tell you about the study, but the researcher will explain it to you first. You may ask this person any questions that you have. When you are ready to make a decision, you may tell the researcher if you want to participate or not. You do not have to participate if you do not want to. If you decide to participate, the researcher will ask you to sign this statement and will give you a copy to keep. Your signature means that you understand what you will be asked to do and agree to participate in the study.

Why am I being asked to take part in this research study?
We are asking you to participate in this investigation because you are a healthy adult, and can move your hand and arm easily. You must be at least 18 years old to be in this research project.

Why are you doing this research study?
The purposes of this study is to test the accuracy of the sensors in a ‘Smart Glove’ prototype when worn by subjects of different ages, genders and hand size. The ‘Smart Glove’ is a lightweight instrumented glove that is designed for use with computerized virtual reality games that will be designed to help patients who have had a stroke to relearn use of their paralyzed hand. The information from this study will be used to help us improve the accuracy of the glove measurements. Based on data from this study, we will develop a calibration routine to correct errors caused by differences in fit of the glove for different hand sizes.

What will I be asked to do?
If you decide to take part in this study we will ask you to attend one session of 1-2 hr in length. First, you will fill out a screening questionnaire to ensure that you have no conditions which might produce pain or soreness in your hand following participation. At the start of the session, we will measure your hand by drawing an outline of your hand on paper, and marking your joint locations on the tracing. Next, you will put on the ‘Smart-Glove’, and it will be connected to the computer so we can record your hand movements.

Next, you will be asked to move your hand, fingers, and/or arm in different ways, while the computer records the movement. For example, you may be asked to make a fist, open your fingers all the way, or lift your hand to shoulder height. A total of 15-20 movements will be tested.
Each movement will be repeated 3 times. Your arms and hands will be videotaped during these movements to help us later when analyzing the data from the glove.

Where will this take place and how much of my time will it take?
The location of the testing will be in Dr. Holden’s Laboratory at Northeastern University, Robinson Hall, Rm 008. Total time period for the study will be 1-2 hours. Frequent rest breaks will be offered to subjects during the testing, or given at any time upon request.

Will there be any risk or discomfort to me?
The risk involved in this study is minimal. There will be no risk greater than your everyday activities. There may be times when your hand and shoulder become fatigued from the motions. This should be minimal and will not cause lasting harm. If you experience any discomfort in your hand or arm the days following the session, you should let one of the investigators know.

Will I benefit by being in this research?
There is no direct benefit to you for taking part in this study. However the information gained from this study may help the investigators advance the use of the glove for patients who have had a stroke.

Who will see the information about me?
Only the investigators, Dr. Maureen K. Holden, Dr. Constantinos Mavroidis, and study personnel will have access to your information. Your information and data will be stored on the computer and coded by number. The key to the number code will be kept in a separate file, available only to the study investigators. Any personal information will be kept strictly confidential. Videotapes (digital) will also be stored by the investigators on a password protected computer, and identified by a number code. If any data from this study are used in publications, in scientific journals, or presentation at scientific conferences, classroom lectures, or laboratory websites, your identity will be protected.

If I do not want to take part in the study, what choices do I have?
You are free to leave the study at any time.

What will happen if I suffer any harm from this research?
Suffering harm is unlikely. In the event of an emergency we will contact 911 and Northeastern University emergency medical services.

Can I stop my participation in this study?
Your participation in this research is completely voluntary. Even if you begin the study you may quit at any time.

Who can I contact if I have questions or problems?
If you have any questions related to the research, you may contact Mark Sivak, MS, Dept of Mechanical Engineering, at (617)-373-7733 or Prof. Maureen Holden, Department of Physical Therapy, Northeastern University, at (617)-373-5274.

Who can I contact about my rights as a participant?
If you have any questions about your rights as a participant, you may contact Nan C. Regina, Director, Human Subject Research Protection, 960 Renaissance Park, Northeastern University Boston, MA 02115 tel. 617-373-4588, email: irb@neu.edu. You may call anonymously if you wish.

Will I be paid for my participation?
You will be given a ten dollar gift card for your participation.
Will it cost me anything to participate?
The only cost for your participation is any travel expenses and paying for parking if not handi-
capped.

I agree to take part in this research.

____________________________________________________  __________
Signature of person agreeing to take part                      Date

____________________________________________________
Printed name of person above

____________________________________________________  __________
Signature of person who explained the study                     Date
to the participant above and obtained consent

____________________________________________________
Printed name of person above
Subject #:_____________

Smart Glove Study
Screening Questionnaire

Name:__________________________________               Date:_______________
Address:__________________________________               Age: _______________
__________________________________               Gender: _______ 
Phone Number:_____________________________               Glove Size:__________
Alternate Phone number: _____________________               (small / medium/ large)
Email Address:_____________________________

Please Circle your response to each question below:

What is your dominant hand? ……………………………………………….. Right / Left
Are you fluent in English (both in writing and speaking)? ………………………Yes / No
Are you able to commute to Northeastern University, Robinson Hall? …………Yes /No
Are you able to commit to one session of 1 to 2 hours in length? ……………..…Yes / No
Have you had any fractures or sprains of your hand in the last year? …………………Yes/ No
Do you have any loss of feeling in your hand or wrist? (For example: tingling or
numbness due to diabetes, nerve injury, arthritis, or other cause) …….……………...Yes/No
Do you have any difficulty moving your hand in daily activities? (For example:
due to pain, weakness, arthritis, fibromyalgia or other cause).......................Yes /No
Have you had any diagnosis of a joint or muscle disorder in your hands or arm?...Yes /No
Do you believe that moving / holding your hand in various positions for ~ 10-15sec
at a time over a 1-2 hour test session might cause you any discomfort?..........Yes/ No
Do you have any arm/hand impairments or concerns that are not addressed above? Please de-
scribe:________________________________________________________________
__________________________________________________________________