

DEVELOPING A MULTI-DEGREE-OF-FREEDOM BODY-MACHINE INTERFACE FOR
CONTROL OF ASSISTIVE DEVICES

By

Sanders Wainwright Aspelund

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Mechanical Engineering - Master of Science

2017

ABSTRACT

DEVELOPING A MULTI-DEGREE-OF-FREEDOM BODY-MACHINE INTERFACE FOR CONTROL OF ASSISTIVE DEVICES

By

Sanders Wainwright Aspelund

Assistive devices are revolutionizing the way people with severe motor impairments are able to interact with their environments and perform activities of daily living. Brain-machine interfaces are not well-suited for the changing body and brain of a growing child and therefore we focused our efforts on further developing body-motion interfaces (BoMIs). Current BoMIs are based on principal-component analysis (PCA) and are well-suited for 2D tasks. We extended the PCA position control mode to velocity control mode to enable children to control the positions of the end-effector of a robotic arm. Testing showed that this method was not well-suited for younger children, nor was it easily expandable to higher degrees of freedom. We also developed a novel BoMI technique utilizing a finite element model to produce a virtual representation of the user's body, referred to as the virtual body model. This enables us to project the movements of the subject to a set of distinct anatomical motion patterns. Using natural body motions of the torso and head, the distinct motion patterns are intuitive and easily learned, even when considering numerous degree-of-freedom (DOF) control. The proposed method can take small, inexpensive, wireless sensors and reliably produce accurate pose estimates of the user. The estimates can be interpreted by the software to generate commands for the assistive device. In testing, a user with limited training was able to utilize the interface to control five DOFs of a robotic arm in order to conduct a pick-and-place task as well as to drink from a water bottle.

ACKNOWLEDGMENTS

Many people have helped me on this journey, and I am truly grateful for all their support:

My parents, Curtis and H  l  ne Aspelund, and my siblings, Keegan, Morgan, and Stanton, for having encouraged me to continue pursuing the next level in my education

Dr. Ranjan Mukherjee, my wonderful and patient advisor, for all of his guidance and wisdom throughout this entire process

Dr. Daniel Segalman and Dr. Rajiv Ranganathan, for being on my advisory committee

Malakiva Padmanabhan, lead experimenter for Principal Component Analysis study

Sheryl Chau, primary developer of the Virtual Body Model method

Connor Boss, for staying up late to help edit my thesis

TABLE OF CONTENTS

LIST OF FIGURES	vi
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Current methods and interfaces	2
1.3 Proposed solution: BoMI overview	5
1.3.1 Preliminary Solution: Principal Component Analysis	6
1.3.2 Second Solution: Virtual Body Model	7
1.4 Proposed Solution: BoMI implementation	8
1.4.1 Basic Experimental Setup and Equipment	8
1.5 Thesis Outline	9
2 PRINCIPAL COMPONENT ANALYSIS: EXPERIMENTAL SETUP AND PROGRAM DESCRIPTION	11
2.1 Experimental Hardware	11
2.2 Experiment and Program Description	13
2.2.1 Calibration	13
2.2.2 Initialization of Testing Program	14
2.2.3 Task Description and the Graphical Interface	17
2.2.4 Testing Process	20
2.3 Robot Control with User-in-the-Loop	20
3 PRINCIPAL COMPONENT ANALYSIS - EXPERIMENTAL RESULTS	22
4 VIRTUAL BODY MODEL: THEORETICAL DEVELOPMENT	30
4.1 Finite Element Model	30
4.2 A Set of Five Anatomically Distinct Motion Patterns	32
4.3 Virtual sensors	35
4.4 Mapping of User Movements to Matching Body Movements of VBM	37
5 VIRTUAL BODY MODEL: EXPERIMENTAL SETUP AND PROGRAM DESCRIPTION	39
5.1 Experimental Hardware	39
5.2 Experiment and Program Description	39
5.3 Robot Control with User-in-the-Loop	41
6 VIRTUAL BODY MODEL: EXPERIMENTAL RESULTS	42
6.1 Virtual Body Model of the User	42
6.2 Verification of matching of user body movements to Virtual Body Model	43
6.3 Experimental Task: Pick and Place	47

7 CONCLUSIONS	51
7.1 PCA Experiment	51
7.2 Development and Experimental Validation of the VBM Method	52
7.3 Further work	52
REFERENCES	54

LIST OF FIGURES

Figure 1:	JACO ² arm with open end-effector	11
Figure 2:	Location and orientation of wireless IMUs for PCA testing [25]	12
Figure 3:	Goal Space with boundary for the PCA reaching task	17
Figure 4:	Semicircle paths for tracing task	19
Figure 5:	Block diagram of PCA-based robot end-effector position control	20
Figure 6:	Block diagram of PCA-based robot end-effector velocity control	20
Figure 7:	Mean movement time for 2D robotic arm testing sessions. The error bars correspond to the standard error of the set. Learning is very apparent between the PreTest and the MidTest while a ceiling effect is observed between the MidTest and the PostTest. The rise in movement time of the 9 year old participants can be attributed to mental and physical fatigue. Only results for the three of eleven 9 year olds and eight of eleven 12 year old who completed the experiment are included. All twenty-six adults completed the experiment.	23
Figure 8:	Mean normalized path length for 2D robotic arm testing sessions. The error bars correspond to the standard error of the sample set. Learning is very apparent between the PreTest and the MidTest while a ceiling effect is observed between the MidTest and the PostTest. The rise in normalized path length of the 9 year old and 12 year old participants can be attributed to mental and physical fatigue. Only results for the three of eleven 9 year olds and eight of eleven 12 year old who completed the experiment are included. All twenty-six adults completed the experiment.	24
Figure 9:	Mean movement time for 2D robotic arm training sessions. The error bars correspond to the standard error of the sample set. The adults show a nearly monotonic decline in their movement times while the 9 year old and 12 year old participants trend slightly downward with much more variability. Only results for the three of eleven 9 year olds and eight of eleven 12 year old who completed the experiment are included. All twenty-six adults completed the experiment.	25

Figure 10:	Mean normalized path length for 2D robotic arm training sessions. The error bars correspond to the standard error of the sample set. The adults show a nearly monotonic decline in their normalized path lengths while the 12 year old participants trend downward for the first four practice sessions but decrease in accuracy as fatigue sets in. Only results for the three of eleven 9 year olds and eight of eleven 12 year old who completed the experiment are included. All twenty-six adults completed the experiment.	26
Figure 11:	This Figure was reproduced with permission from Children show limited movement repertoire when learning a novel motor skill [25]. Mean movement time for 2D cursor position control sessions. The error bands correspond to the standard error of the sample set. There is very clear learning occurring between the PreTest and MidTest while a ceiling effect is observed between the MidTest and PosTest	28
Figure 12:	This Figure was reproduced with permission from Children show limited movement repertoire when learning a novel motor skill [25]. Mean normalized path length for 2D cursor position control sessions. The error bands correspond to the standard error of the sample set	29
Figure 13:	VBM dimensions: head width ℓ_1 , distance from top of the head to base of the neck ℓ_2 , shoulder span ℓ_3 , torso height ℓ_4 , length from under arm to waist ℓ_5 , width of waist ℓ_6 , chest span ℓ_7 , torso thickness ℓ_8 , neck thickness and width ℓ_9 , and height of the lower portion of the head ℓ_{10}	31
Figure 14:	(a) Undeformed configuration of the VBM. (b), (c), (d), (e), and (f) show five deformed configurations of the VBM corresponding to five anatomically distinct motion patters of the upper body. The five anatomically distinct motion patters are: (b) torso flexion and extension, (c) torso lateral flexion, (d) torso rotation, (e) neck lateral flexion, and (f) scapular protraction and retraction.	33
Figure 15:	Placement of the j -th virtual sensor, $j \in 1, 2, \dots, k$, on the triangular face of a tetrahedral element of the VBM. The nodes of the triangular element are marked as a_j , b_j , and c_j ; a_j is arbitrarily selected as the base node, and $\zeta_{j,orig}$ and $\eta_{j,orig}$ are unit vectors from the base node to the other two nodes in the initial undeformed configuration.	36
Figure 16:	Five wireless IMUs attached to the cap and vest of the user are circled in the (a) front and (b) back views of the user: three are located on the vest and the other two are placed on the top and back of the cap.	40
Figure 17:	Block diagram of VBM robot control system with user-in-the-loop	41

Figure 18:	User posed in first orientation along with matching VBM configuration. The resultant forces describe a positive force in P_1 causing the torso flexion and a positive force, P_3 , causing the torso rotation. The head remains neutral and is thus aligned with the shoulders. This would have caused the robot to move forward and upwards as described in Section 4.2	44
Figure 19:	User posed in second orientation along with matching VBM configuration. The resultant forces describe a negative force in P_2 causing the lateral torso flexion and a negative P_4 causing the head to remain upright. This would have caused the robot to move to the left and end-effector to spin in a negative direction as described in Section 4.2	45
Figure 20:	User posed in third orientation along with matching VBM configuration. The resultant forces describe a positive force in P_3 causing the torso rotation and a positive force in P_5 causing the scapular protraction. This would have caused the robot to move upward and the end-effector to close as described in Section 4.2	46
Figure 21:	End-Effector trajectory in Cartesian coordinates during user trial	48
Figure 22:	Variation of Cartesian coordinates of the end-effector, wrist rotation, and end-effector configuration with time for the user trial	49

1 INTRODUCTION

This chapter begins by discussing the motivation for this work and providing a literature review of the current state of interfaces that are available to assist those with severe physical disabilities. Then the current need for development in this area, in order to complement what today's methods lack, is discussed. Finally, the solution presented in this thesis is introduced.

1.1 Motivation

In the Greater Lansing community, there is a child with congenital absence of all four limbs. While he is able to navigate using a powered wheelchair that he controls using all that developed of his right arm, which is only an inch long, he requires assistance in almost all activities of daily living (ADLs). These ADLs include activities such as grasping a cup and bringing it to his mouth in order to drink, or turning a doorknob to open a door so as to allow him to pass through with his wheelchair. These are among many other tasks that currently require a caregiver to accomplish. Providing the child with the means to carry out these tasks on his own will allow for a much greater sense of independence and improvement in the quality of life due to no longer needing to rely on a caregiver to provide the needed help. This is especially important as he becomes a teenager and seeks to individuate himself among his peers and acquaintances as more than his disability and the constant assistance he requires. Additionally, a semi-permanent solution could provide cost savings for many years to come by reducing the amount of time a caregiver is required to be present.

On a much larger scale, the 2010 census estimates that approximately 300,000 children un-

der the age of 15 require some form of assistance when performing ADLs [1]. In this survey, ADLs were defined to include “difficulty getting around inside the home, getting into/out of bed, bathing, dressing, eating, or toileting”; all of which are related in some significant way to mobility and movement. There exist many different piecemeal solutions for providing assistance to these individuals beyond the direct intervention of caregivers. Examples include powered wheelchairs, robotic devices, braces for limbs and joints, ergonomic tools specifically designed to assist the individual with specific tasks [2], etc. that enable the user to function at a more adept and independent level. Each of these tools are matched with the user based on the person’s needs and abilities.

1.2 Current methods and interfaces

Each of the more complex aforementioned assistive devices requires a specific interface to interact with or control the device; an interface that may not be ideal for every user. Some devices can be operated using a physical interface such as a keyboard, a joystick, or a mouse, which all require the applicable appendages (arms, hands, fingers) and motor learning capabilities. An effort has been made to create interfaces that are verbally controlled [3, 4] or track eye and head movements [5] to sense what the user is attempting to communicate. This information may be in reference to a physical target, either in local space or on a monitor for remote work, that the interface must interpret. Verbal control methods encounter problems of filtering out noise from the environment and extracting the user’s verbal instructions to the robot from interactions with others. Eye and head tracking methods face challenges of their own ranging from dealing with lighting conditions that make it difficult to track the user’s gaze [6], to handling errant glances that the user does not intend to use as input.

As such, a tremendous amount of work has been conducted in the realm of human-in-the-loop robotic systems in which the robotic system attempts to assist in the decision making of the task based on input from the user [7]. This work involves using sensors such as stereo cameras and forces sensors for building a model of the environment and interacting with the objects of interest in it. In this study, the user was given varying degrees of control in a pick-and-place task using a joystick for input. Motion planning for large and fine movements, collision avoidance, optimal grasping strategies, etc. were calculated by the robotic system to reduce the control load on the user. User surveys showed these to be very beneficial in easing the cognitive effort. The outcomes of the task for both grasping and pick-and-place tasks also improved as determined by the success of the grasp, the accuracy and speed of the movement, and the number of unwanted collisions. However, even with the tremendous advances in this area of research, there must still be a robust level of interaction for user control during times when the robotic system misinterprets the environment, incorrectly understands the user's intentions, or makes a choice that the user deems to be undesirable.

A type of general-purpose interface that is being used and studied are Brain-Machine Interfaces (BMIs) which function by measuring either electrical signals from the surface of the scalp using electroencephalography (EEG) or measuring the signal emanating from individual neurons or very small groups of cells in the brain through use of a cortical implant. Through an intensive training process, the user can learn to control their brain signals as recorded by the BMI in order to operate the associated device [8–12].

There are several disadvantages that make BMIs unsuitable for children. Non-invasive BMIs, which involve using a mesh of electrodes on the scalp, have a low signal-to-noise ratio [13] and therefore are limited in controllable degrees of freedom (DOFs) and speed of control even with

extensive training. They provide slow rates of data (up to 5.32 bits/sec [14]) as well as drifting over time [15], therefore requiring periodic recalibration. Additionally, the daily task of setting up the EEG is a burdensome process that can take 20-30 minutes for configurations using gel for good signals, though dry electrode technology is advancing well.

Invasive BMIs require surgery to implant the sensors under the scalp such that they are in direct contact with the neurons. This surgery is expensive, dangerous, and importantly, temporary. The danger arises due to the risks of the operation as well as the increased risk of subsequent infection. The temporary nature of the implant is due to the fact that current materials decay after a few years [16], the devices are enveloped with biologic material that damps the signal, and the structure of a child's brain continues to change. Neither invasive nor non-invasive BMI methods use any of the body's natural motions that the user may still retain, which can cause these abilities to further decay in functionality. Also, children are very rarely in situations such that they have lost all movement of the body, in which case these brain-machine interfaces would be more suitable.

A related interface is the muscle-machine interface which uses electromyography (EMG) to measure the electrical signals traveling along a neuron away from the brain or the activity involved in the flexing of muscles, and correlating them to commands to the device. This can be used to assist working muscle that is too weak to accomplish a necessary task [17] or to generate independent control signals such as to operate a drone [18]. This method suffers issues related to the noise of the signal, the ability of the user to purposefully flex the muscles associated with specific commands, and the need for suitable attachment points which can be lacking for individuals with congenital or surgical absence of limbs.

1.3 Proposed solution: BoMI overview

As an alternative to BMIs and other interfaces previously discussed, Body-Machine Interfaces (BoMIs) are also general-purpose interfaces that allow the user to actively control a device using the gross movements of their body instead of tools such as joysticks, gamepads, computer mice, etc. We will first discuss the current state of BoMIs, specifically those that use inertial measurement units (IMUs) to measure movements of the torso and head, and how they compare to the alternatives. Then we will present two proposed methods of implementing these BoMIs; first, using an expansion of a PCA approach, and then introducing a novel approach that uses a VBM. These two methods were the basis for our experiments that are described later.

BoMIs exist in multiple variations that use body signals such as muscle activity, head control, eye-tracking, etc. to control assistive devices [19–21]. In children, there is not only the need for assistive devices, there is also the need for the child to develop socially through interactions with family, teachers, peers, and friends. In this regard, it is important that the interface not be intrusive with regular communication, thereby disqualifying most eye and tongue trackers. The interface devices should be small enough to not be overtly noticeable and obtrusive. Otherwise they end up dominating every introduction the child has, not allowing them to feel normal.

The solution we propose is to use wireless IMUs to measure the movements of the torso and head of the individual. These measurements can then be transcribed using various algorithms to control a variety of assistive devices such as computers, wheelchairs, robotic arms, etc. Wireless IMUs are quite small, inexpensive, and very simple to relocate on the user's body as he/she grows physically. Currently, these devices are very robust, with high signal to noise ratios and long-lasting batteries; characteristics that are only expected to improve in the coming years. The upper body is

an ideal location to situate sensors because the upper body is usually unaffected by conditions, such as birth defects or amputations, that require children to need assistive devices. Studies conducted using adult participants with tetraplegia have shown that patients were able to quickly learn to control a cursor on a screen or a robotic wheel chair [22, 23] in one to two sessions. Importantly, the ability to successfully use their devices continued over time, without the need for recalibration of the sensors.

Currently, limitations of the BoMI method include the difficulty of controlling high numbers of degrees of freedom in a manner that is intuitive and comfortable for the user. This was the problem that was investigated for this thesis and for which our novel approach seeks to provide a solution.

1.3.1 Preliminary Solution: Principal Component Analysis

PCA has been used to describe how the nervous system reduces complex movements into relatively few body motions that account for the vast majority of the variance of the movement [24]. In this study of hand and finger coordination, participants were asked to imagine grasping a large number of everyday objects (e.g. banana, calculator, Frisbee, needle, etc.) and it was found that the two principal components with highest variance accounted for 80-85% of all the variance in the data. Our preliminary BoMI solution, as described in Chapters 2 and 3, involved utilizing the PCA method to find the two principal components with highest variance of the shoulder and torso movement of a participant as measured during a period of unrestricted movement. This method was previously utilized by our group for 2D position cursor control and has now been adapted for 2D robotic velocity control. To determine these two principal components, the user was asked to explore their full comfortable range of motion as recorded by a set of four wireless IMUs. From this data, the two principal components associated with the largest variances were extracted. When

being used to control a cursor or robot, the system mathematically projected the wireless IMU output onto these two principal components and then interpreted the match as commands to be sent to the device.

This method was much simpler to implement at any number of DOFs, but learning became difficult very quickly as the DOFs were increased. This is due to the fact that principal components are ordered in terms of greatest to least variance, and therefore, additional included principal components describe smaller variances that are harder to control. Additionally, learning was difficult at even low degrees of freedom for participants in which the mathematical basis found during the period of unrestricted motion was not compatible with natural body motions. Another aspect of PCA is that it ensures that the basis movements in their mathematical representations are orthogonal to each other, while natural body motions are often not. Therefore, as DOFs increased, the ability to separately control independent degrees of freedom became increasingly difficult as natural body movements incorporated multiple basis vectors.

1.3.2 Second Solution: Virtual Body Model

Our second solution, presented in Chapter 4, involved creating a virtual body model (VBM) which uses a finite element model to represent the physical body of the user. Given the natural movement patterns of a human torso and head, such as leaning forward and backwards, and twisting or turning side to side, these distinct movement patterns can be mapped to commands controlling the movements of the device. In implementation, wireless IMU signals are captured which are used to define the current body pose of the user. Then a small set of virtual forces are defined as pertaining to each distinct movement pattern. These were applied to the VBM in order to cause the VBM to deform so that it closely matches the orientation of the user's physical torso

and head. These forces are considered to be a mapping of the body movements to commands to be sent to the robot interface. In the new mapping, the magnitudes of these forces can be scaled as necessary to control sensitivity along particular degrees of freedom of the assistive device. While our current implementation works well for most users, in cases of restricted ability to move parts of the body, a study of the patient's movement patterns can be used to adapt the VBM and forces to match what the user is capable of.

This method was more difficult to implement as it is tailored to each user as well as being computationally slower. However, because the commands correspond to natural movement patterns that are both distinct and simple to combine, the system was easier to control at higher degrees of freedom. Using this method, a test-user was able to control a robotic arm and use it to pick up a water bottle from a table and set it down on a raised surface half a meter away, successfully releasing it and pulling the end-effector away.

1.4 Proposed Solution: BoMI implementation

For this research, we proposed utilizing a body-machine interface (BoMI) to map the output from a set of wireless IMUs as a measure of the user's body movements to controls for a JACO² Robotic Arm from Kinova Robotics. More thorough descriptions of the PCA method and VBM method experiments are described in Chapter 2 and Chapter 5, respectively.

1.4.1 Basic Experimental Setup and Equipment

For the testing of the two experiments, multiple wireless IMUs (3-SpaceTM Wireless 2.4 GHz DSSS, TSS-WL Handheld Sensor Unit) from Yost Labs Inc. were attached to the user's vest using hook and loop fasteners. The wireless IMUs communicated with a laptop through a wireless USB

dongle also purchased from Yost Labs Inc. which then ran the respective algorithms to send the appropriate velocity commands to the JACO² robotic arm. Based on the desired configuration of the end-effector of the robotic arm, the user would need to change the position of their torso, or in the case of the VBM experiment, their head, to cause the robot to move as required. For the PCA testing, the experiment was based off of previous work by the lab of Dr. Mei-Hua Lee in which the user would directly control the position of a 2D cursor on a screen using body movements. The task in this experiment was to direct the cursor into various targets as they appeared on the screen [25]. In the new experiment however, the user would need to control the end-effector of the robotic arm using velocity control in order to move it to targets on a projector screen. One concern was the user's ability to learn a mapping from controlling the velocity of the cursor to the resulting position of the cursor. For verification of the VBM model, a pick-and-place task using a water bottle was attempted as validation. Additionally, it was seen if the user could grasp the water bottle using the end-effector and drink from it using a straw.

1.5 Thesis Outline

The remainder of this thesis is divided into two parts. The first part presents the PCA experiment and results as follows: Chapter 2 discusses the experimental setup used to conduct the PCA learning study, with the PCA experimental results presented in Chapter 3. The second part presents the VBM theoretical development, experiment, and results as follows: Chapter 4 discusses the development of VBM method and the theory of how it takes the outputs from the wireless IMUs and converts them into commands to send to the robot; Chapter 5 discusses the experimental setup used to conduct the VBM verification testing; and the VBM experimental result are presented in

Chapter 6. Chapter 7 gives the conclusions arrived at from this work as well as future work to explore additional avenues of this work.

2 PRINCIPAL COMPONENT ANALYSIS: EXPERIMENTAL SETUP AND PROGRAM DESCRIPTION

2.1 Experimental Hardware

The robot used for the experiments was a JACO² robotic arm from Kinova Robotics with six DOFs, a reach of 900 mm, and unlimited joint rotation. The end-effector was a three-fingered grasper with under-actuated fingers that adapt to size of a grasped object. The robotic arm was located about 0.5 meters to the left of the center of the user and 0.25 meters in front. The robot was configured to initialize in a left handed configuration, as shown in Figure 1, to maximize the



Figure 1: JACO² arm with open end-effector



Figure 2: Location and orientation of wireless IMUs for PCA testing [25]

viewable area of the workspace.

The main program was run in a Linux Ubuntu 14.04 environment on a Lenovo Thinkpad with an i5 Intel processor that communicated with the robot through USB. Four wireless IMUs, described above, were used to capture body movements. They were configured to return the three Euler angles roll, pitch, and yaw.

For the PCA test, the wireless IMUs were placed on the front and back of both shoulders and tilted at approximately 45 degrees from vertical as shown in Figure 2. The PCA wireless IMU locations were chosen to capture redundancy of the gross motions of the shoulders and torso as there were only two PCAs that were considered for this testing. This concept was developed for previous testing done by collaborators for 2D cursor position control [25], and was replicated so as to better compare the two sets of results. In keeping with the original experiment, the yaw values were discarded due to drifting of the yaws sensors in the original experiment.

2.2 Experiment and Program Description

The testing scheme for this experiment was developed as an analog to the 2D cursor position control that was presented in [25]. The experiment began with the attachment of the four wireless IMUs to the vest of the participant as described above. The output of the wireless IMUs was recorded while having the user explore their full range of motion to find their principal components and variances. The principal components and variances were transferred to the other computer running the robot control program. Using the calibration matrix, the user was to control the robot through a series of two tasks, repeated through 13 sessions: PreTest, Tracing1, Practice1-4, MidTest, Practice5-8, PostTest, Tracing2. The goal was to see how well the user learned to control the robot through the course of the program; specifically, if their times improved between testing sessions. This overview is explained in more detail in the following subsections.

2.2.1 Calibration

To begin, the user donned a well-fitted vest upon which the wireless IMUs were attached as described above. After beginning the PCA calibration program that would record the wireless IMU data and return the principle components and variances, the user was instructed to find a comfortable position referred to as their “happy place”. This would serve as their origin about which their wireless IMU angles would be zeroed. Then the calibration program was run for 70 *sec* at 50 *Hz*, gathering the pitch and roll of the four wireless IMUs and combining them into an eight-dimensional vector space. In keeping with the 2D cursor control program, the yaw was ignored. This portion was referred to as the “calibration dance” during which the user was asked to move their shoulders and torso through a range of motion that they would feel comfortable using

for an extended period of time. They were encouraged to use a large and diverse set of motions so that a calibration would be created with variances of reasonable magnitude. A set of restricted motions during calibration would result in small variances which would make control very sensitive and therefore difficult. A very large set of motions during calibration would require the participant to use comparatively large motions during the control portion. However, the variances were only checked after the participant concluded the experiment.

The MATLAB[®] function *pca* was applied to this eight-dimensional set of points, returning the principal components as eight orthogonal unit vectors along the principle axes as well as their corresponding variances. The two 1×8 vectors with the highest variances were then saved to a file, along with their corresponding variances, as a 2×8 matrix for use in the control portion of the experiment.

2.2.2 Initialization of Testing Program

The user was seated at the testing setup slightly behind and to the right of the robot, facing a large projector screen. At this point the experimenter would initialize the testing program. After automated checks were performed to ensure communication is successful, the wireless IMUs were primed with their output formats and frequency and the robot arm was sent to a predetermined initial position. This starting position was chosen such that the orientation of the end-effector made it simple for the participant to visually associate the tips of the fingers with the robot's coordinate system. This initial orientation ensured that the end-effector maintained a very stable orientation during the testing as it traveled throughout the task space. As the arm moved into place, the graphical interface that the user would be interacting with, as described in the next section, would appear on the screen using a back-lighting projector to avoid shadows cast by the arm.

When everything was in position, the operator asked the user to assume their comfortable “happy place” position and commence the main portion of the program. This began with the wireless IMUs being sent a command to zero their positions and then commence streaming their orientations relative to that zeroed position. Subsequently, a new thread was created to handle the parsing of the data from the wireless IMUs and pack it into a shared vector that could be accessed in the main loop of the program, running at 100 Hz. The vector of eight wireless IMU values was multiplied by the 8×2 matrix from the calibration session, resulting in a 2×1 vector which corresponded to the raw X and Y velocities as shown in Equation (1).

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} \end{bmatrix} \cdot \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} \quad (1)$$

These were divided by the square root of their corresponding variances, so as to normalize them, and multiplied by a gain factor. The magnitude of the resulting velocity vector was then compared to a predefined buffer-zone size. If the magnitude was inside the buffer-zone, then the velocities were set to zero. If the magnitude was outside the buffer-zone, then a vector in the same direction as the velocity, but with a magnitude equal to that of the buffer-zone, would be subtracted from the

velocity vector. This set the boundary of the buffer-zone to zero velocity. Outside the buffer-zone, the velocity grew linearly. This buffer zone was an important addition to the system as the user would not be able to return exactly to their “happy place” and remain absolutely still. This would result in the robotic arm always receiving non-zero velocities and continuously moving. The buffer zone provided a simple means for the user to have comfortable range of motion to return to and have the robot stop moving.

The position of the robot was queried and compared to see if it was in the correct goal or if it was against the virtual border of the task space. These are described in more detail in the next section. If the fingertips entered a goal area, a timer was started to see if they had remained in the goal long enough. A new goal was presented if this requirement was met. If they left the goal before the timer reached the required value, then the timer was reset until they re-entered.

Before the first testing session began, an orientation session was performed where the user was able to ensure that their body movements, instructed to be random, were able to control the robot. Additionally, during this orientation session, a marker was presented on the screen between the fingertips fo the end-effector, that the user was informed corresponded to robot’s position and to remember this for the remainder of the study. The marker was not shown during the testing and practice sessions because it was found that the user only focused on the marker rather than the robotic arm as a whole. This orientation session was only allowed to be very short so as not to provided the user with time to do any substantial learning of the mapping of their body motions to the velocity commands sent to the robot.

2.2.3 Task Description and the Graphical Interface

There were two primary types of tasks that the user had to accomplish: the reaching task derived from the 2D cursor position control experiment, and a novel tracing task. For the reaching task, the user was presented with a series of target goals located at eight radially symmetric points as shown in Figure 3. The distance from the center of the entire area to the center of one of the outer circles was 22 cm. This value was chosen based to be twice that of the value used in the cursor control program in [25] to allow for more direct comparison. The radius of the circles was determined to be 2.14 cm using Fitts' Law as done in [25]. In kinesiology literature, Fitts' Law is

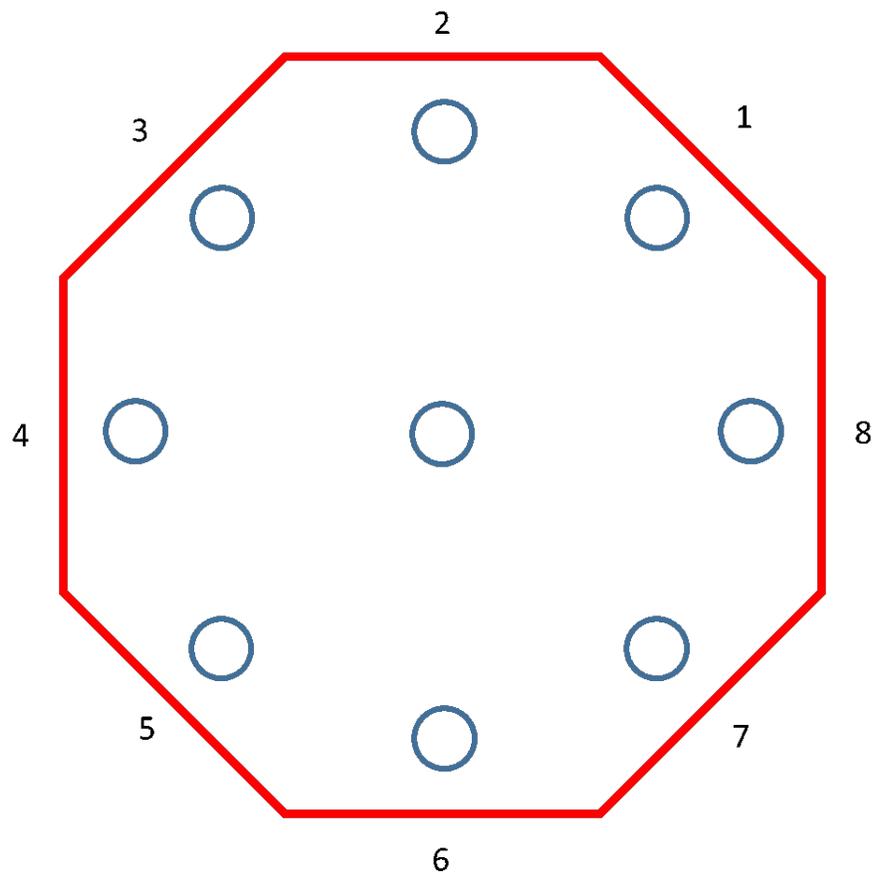


Figure 3: Goal Space with boundary for the PCA reaching task

a predictive model of human-computer interaction [26].

The robot started in the center of the space and one at a time, in a random order, the outer goals would appear on the screen. The user was tasked with controlling the robotic arm such that the tips of the fingers entered the denoted outer circle and remained in the area for 500 *msec*. This time was chosen to ensure that the control was purposeful and the entering of the area was not simply the user conducting a “driveby” with the robotic arm. Given that the maximum speed of the robotic arm was 20 *cm/s*, this length of time necessitated that the user slowed down upon entering the target area and didn’t just pass through a thin slice of the area.

When the outer target was successfully entered and the position was held for the specified amount of time, the center goal reappeared and the user was instructed to direct the robotic arm towards that circle just as before. This process would repeat until the session was complete. For the Test sessions, all eight outer targets were used two times. For the Practice sessions, only the 2,4,6,8 numbered targets were used, but three times each. The order was randomized within each set without overlap. For example, the Practice session outer goal order might be [3,2,1,4,2,4,1,3,4,3,1,2].

In order to enable the user to receive additional feedback regarding their location and progress, multiple cues were used. Always present on the black background during Testing sessions, was the number of goals left; while for Practice sessions, the most recent score pertaining to how close the user was to the center of the target goal, normalized to the radius of the goal, was always present. When a new goal appeared, its outline would appear in green for three seconds before transitioning to red. When the robot entered the goal area, the goal would be filled in with yellow to signify this.

Additionally, to prevent the user from traveling too far out of the desired workspace, an octagonal boundary was programmed into the system such that when the robot was at the boundary, a

command to move further outward would be canceled. Therefore, the user could only return to the workspace. When this feature was activated due to the position of the robot, the octagon would appear on the screen in thick red lines to signify the user that they would only be able to move in certain directions.

For the tracing task, the user was instructed to follow a curved path as closely as possible. The paths were defined as semi-circles as shown in Figure 4a and 4b. The first four curves were semi-circles with a diameter of 22 cm, while the second set of double semicircles each had a diameter of 11 cm. In the interest of time, if the user had not successfully reached the end point of the given curve after 30 sec, the robot would return to center and prompt the next curve. Only one curve was shown at a time. Due to the precise control in multiple degrees of freedom required to complete this task well, and in order to provide better feedback to the user, the location of the robot cursor was always present on the screen. Additional bands would sequentially appear as the cursor moved farther and farther away from the desired curve.

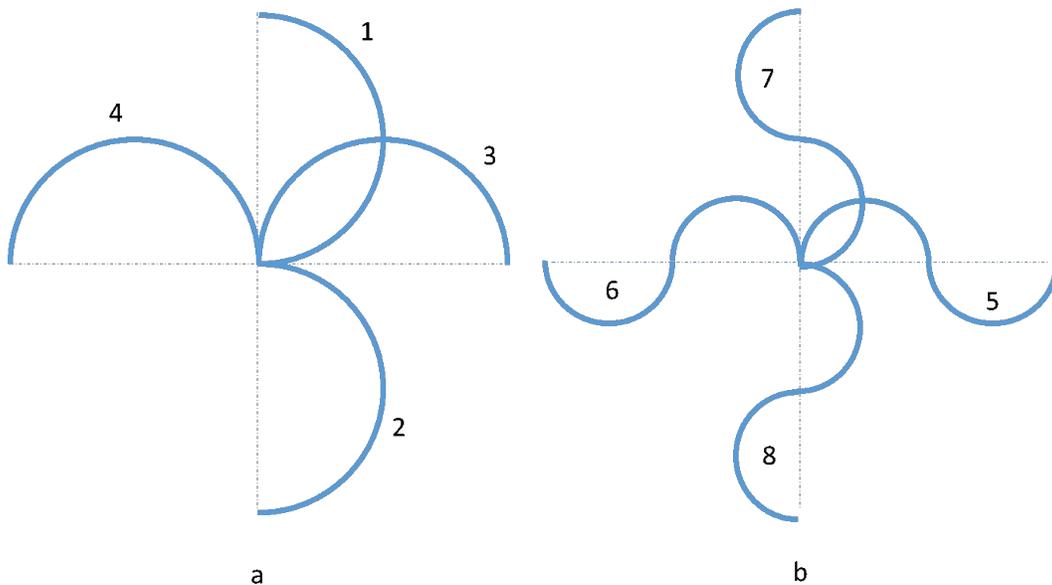


Figure 4: Semicircle paths for tracing task

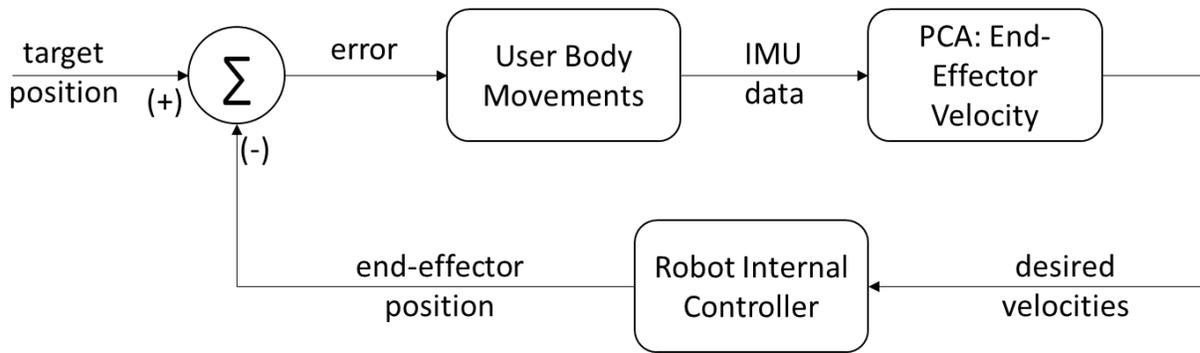


Figure 5: Block diagram of PCA-based robot end-effector position control

2.2.4 Testing Process

As previously mentioned, the order of the sessions was PreTest, Tracing1, Practice 1-4, MidTest, Practice 5-8, PostTest, Tracing2. After each session, the user was allowed a small break to rest before the next session. Plenty of encouragement and praise was provided throughout the sessions in order to maintain motivation and enthusiasm.

2.3 Robot Control with User-in-the-Loop

A block diagram of the feedback loop used to control the position of the robot is shown in Figure 5. In this control structure, the current configuration of the robot is compared to its desired configuration by the user. The user performs a movement of their torso as recorded by the wireless

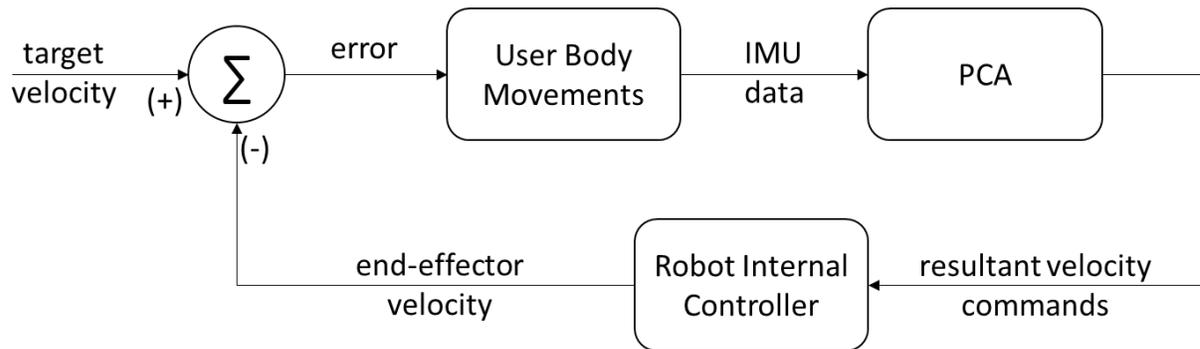


Figure 6: Block diagram of PCA-based robot end-effector velocity control

IMUs. This data is then compared to their principal vectors to generate the velocity commands that are sent to the robot. Figure 6 shows a block diagram of the feedback loop used to control the velocity of the robot. This more clearly shows how the user needs to learn and adapt their movements in order to place their body in the correct orientations to generate the required velocities in order to move the end-effector of the robot to the goal locations.

3 PRINCIPAL COMPONENT ANALYSIS - EXPERIMENTAL RESULTS

For studying the ability of participants of different ages to learn to control the robotic arm in two dimensions, 48 participants were recruited to complete the set of tasks described in Chapter 2. The participants ranged in age from 8 to 37, grouped into sections of 9 year old participants ± 1 year, 12 year old participants ± 1 year, and adults of age 18 or older. There were no subjects invited between the ages of 14 and 17. There were eleven 9 year old participants, with four females and seven males; eleven 12 year old participants, with four females and seven males; and twenty-six adults, with fifteen females and eleven males. None of the participants were cognitively or physically impaired.

Of the 9 year old participants, three successfully completed the full experiment, while eight did not; of the 12 year old participants, eight successfully completed the full experiment, while three did not; and all of the adults successfully completed the experiment. If a participant did not complete the full experiment for any reason, they were considered an attrition case. The reason for attrition among the younger participants was an inability to complete the task either before losing interest and asking to stop or failing to make sufficient progress in the allotted time as determined by the experimenter. Multiple reasons have been postulated to explain why this attrition occurred, but they are still being studied. The data from the attrition cases were not included in the following results due to their incompleteness.

In order to most closely be able to compare the data for the 2D velocity control experiment with the robot, to that of the previous work for the 2D cursor position control, the movement time

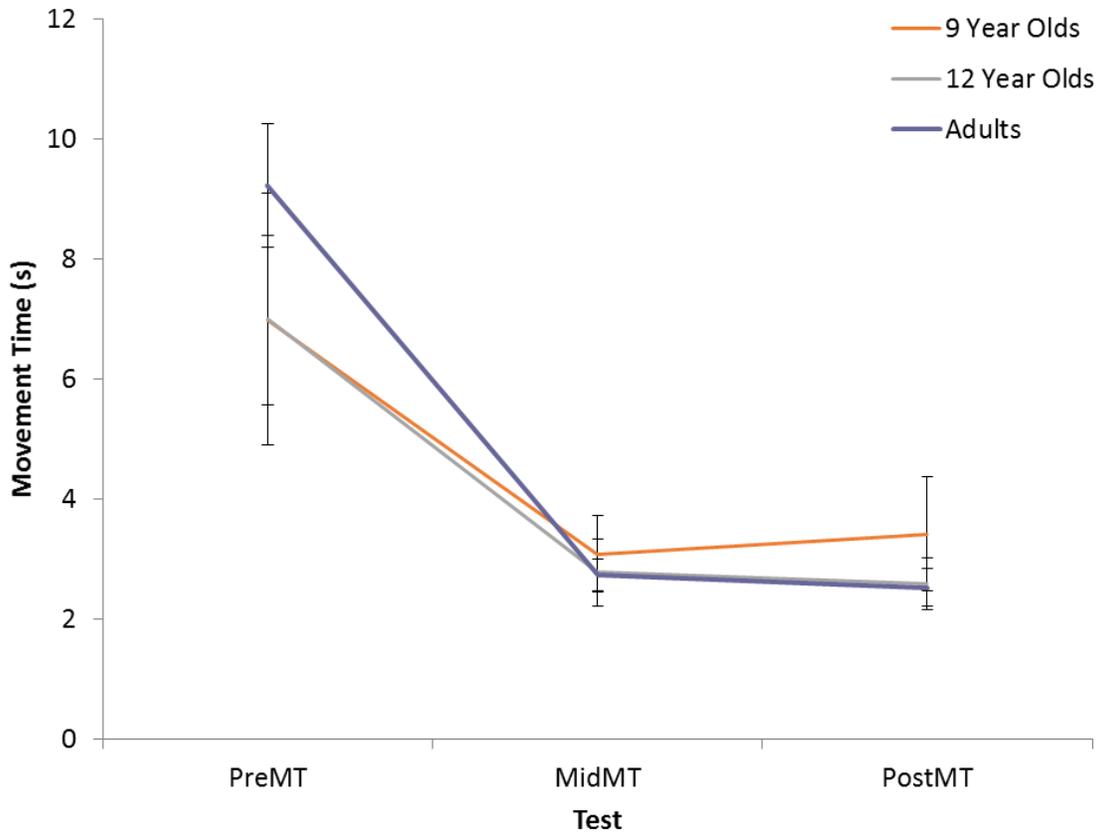


Figure 7: Mean movement time for 2D robotic arm testing sessions. The error bars correspond to the standard error of the set. Learning is very apparent between the PreTest and the MidTest while a ceiling effect is observed between the MidTest and the PostTest. The rise in movement time of the 9 year old participants can be attributed to mental and physical fatigue. Only results for the three of eleven 9 year olds and eight of eleven 12 year old who completed the experiment are included. All twenty-six adults completed the experiment.

and normalized path length for each session was calculated. The movement time was defined as the average time it took for the participant to direct the robot from the center goal to the outer goal after it had appeared. The normalized path length was defined as the total distance traveled by the robot, normalized to the distance between the center goal and the outer goal. These values were tabulated and plotted to see how learning occurred in each individual participant as well as within each age group as shown in Figures 7-10. While three successful 9 year old participants is not a statistically significant sample size, their values are included here for completeness.

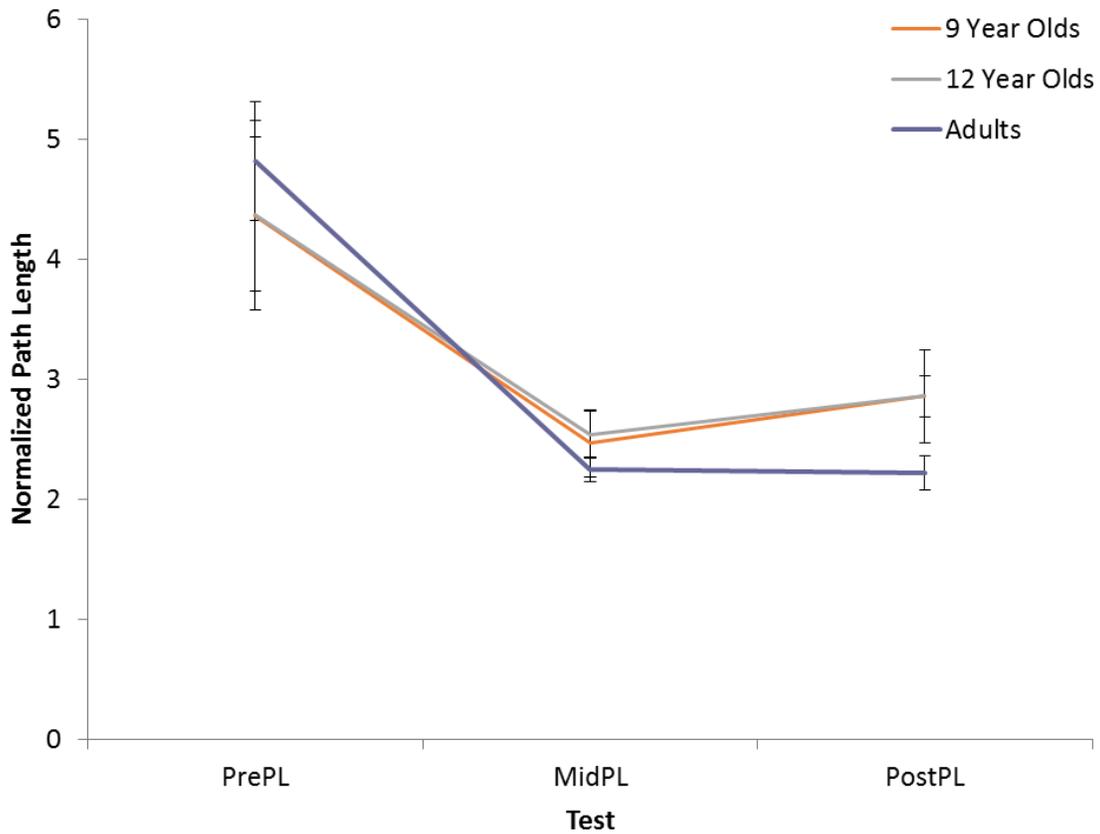


Figure 8: Mean normalized path length for 2D robotic arm testing sessions. The error bars correspond to the standard error of the sample set. Learning is very apparent between the PreTest and the MidTest while a ceiling effect is observed between the MidTest and the PostTest. The rise in normalized path length of the 9 year old and 12 year old participants can be attributed to mental and physical fatigue. Only results for the three of eleven 9 year olds and eight of eleven 12 year old who completed the experiment are included. All twenty-six adults completed the experiment.

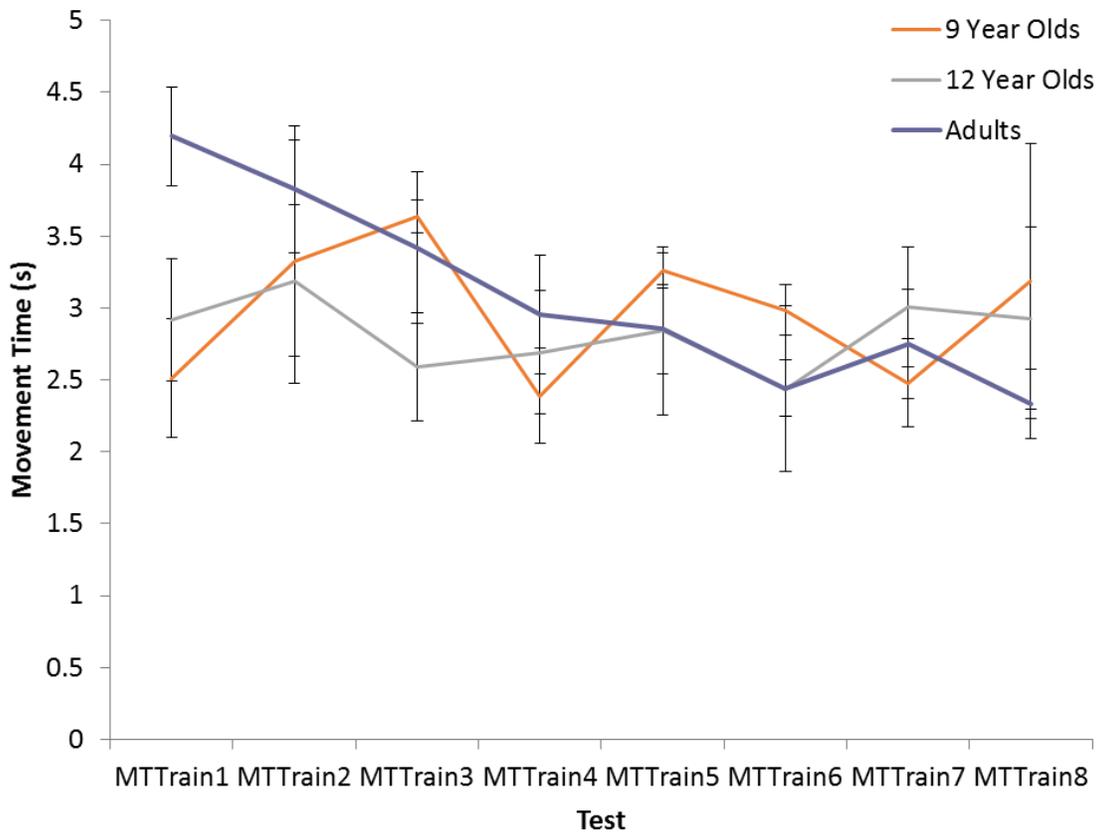


Figure 9: Mean movement time for 2D robotic arm training sessions. The error bars correspond to the standard error of the sample set. The adults show a nearly monotonic decline in their movement times while the 9 year old and 12 year old participants trend slightly downward with much more variability. Only results for the three of eleven 9 year olds and eight of eleven 12 year old who completed the experiment are included. All twenty-six adults completed the experiment.

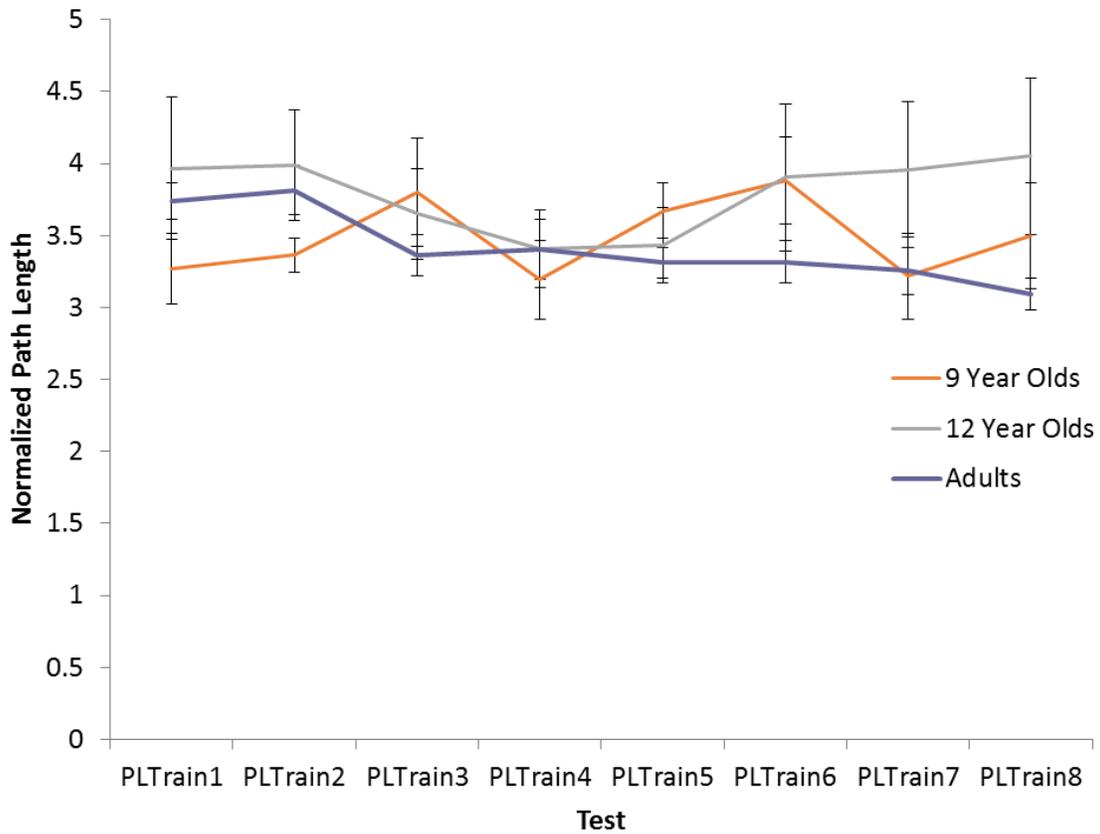


Figure 10: Mean normalized path length for 2D robotic arm training sessions. The error bars correspond to the standard error of the sample set. The adults show a nearly monotonic decline in their normalized path lengths while the 12 year old participants trend downward for the first four practice sessions but decrease in accuracy as fatigue sets in. Only results for the three of eleven 9 year olds and eight of eleven 12 year old who completed the experiment are included. All twenty-six adults completed the experiment.

These figures all show that learning was definitely occurring between the PreTest and the MidTest, but that a ceiling effect was present between the MidTest and the PostTest. For the robotic control, this ceiling effect could have been exacerbated by the fact that the robotic arm was limited in the firmware to travel at a maximum speed of 20 *cm/s*. This means that after the participant learned how to direct the robotic arm in the desired direction and how to slow down as the target area was approached, the minimum movement time was still limited by the physical speed of the robot. This was not the case for the cursor control as there was a direct mapping from the orientation of the user to the 2D position of the cursor on the screen.

Part of the reason that the 9 and 12 year old sample groups had shorter mean initial movement times and mean normalized path lengths than the adults is explained by the observation that only non-attrition cases are included. All of the adults completed the task, even with very non-intuitive PCA matrices that were difficult to learn and therefore control initially. In comparison, children with slightly difficult configurations were likely to ask to stop or reach the 1.5 hour time limit set by the experimenter, and so their extremely high movement times and normalized path lengths did not skew their group results upwards.

These figures were then compared to those from the 2D cursor position control (Figures 11 and 12) [25]. The results were very similar in regards to the adults having shorter mean movement times and normalized path lengths than the 12 year olds who then had shorter mean movement times and normalized path lengths than the 9 year olds. It was also apparent in both cases that substantial learning occurred between the PreTest and the MidTest, while a ceiling effect was noticeable between the MidTest and the PostTest. This ceiling effect was visible in that times and path lengths leveled out as improvement slowed down. The 9 year olds in both studies exhibited a decrease in performance, especially when compared to the results of the 12 year olds and adults.

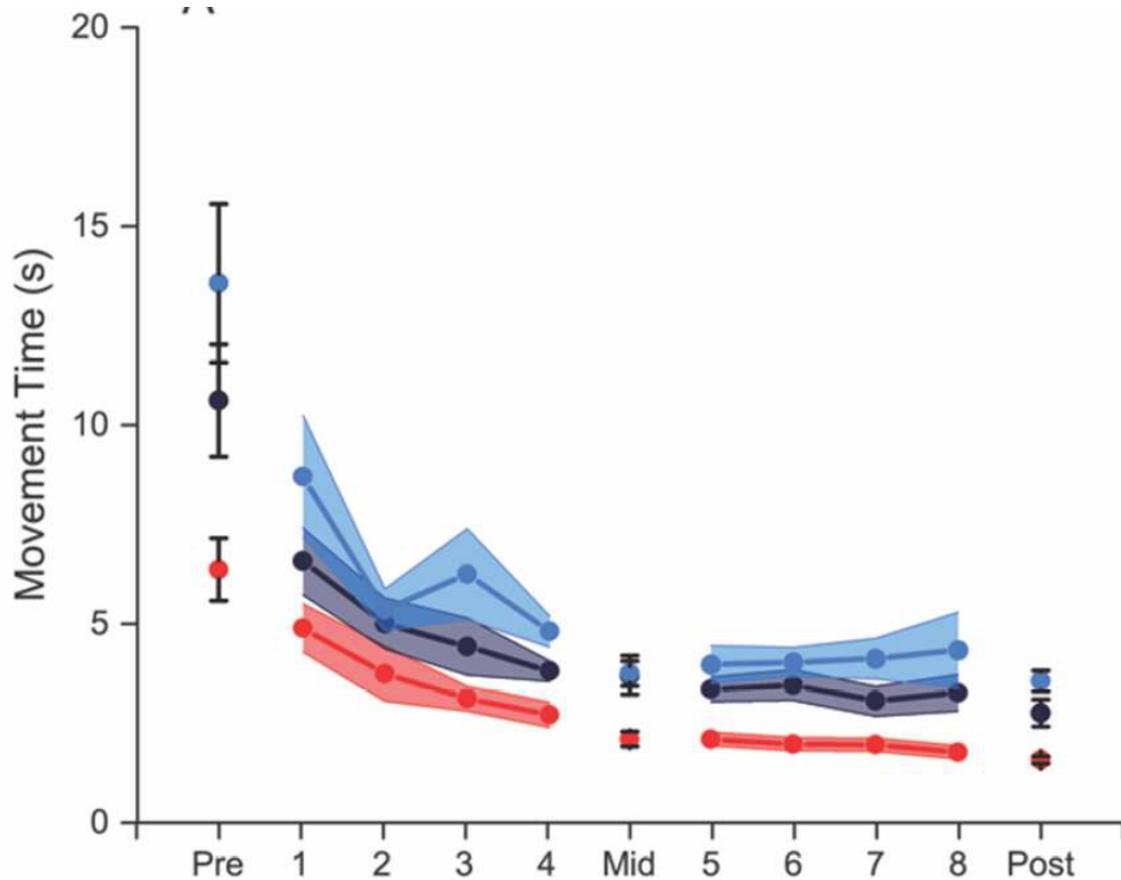


Figure 11: This Figure was reproduced with permission from Children show limited movement repertoire when learning a novel motor skill [25]. Mean Movement Time for 2D Cursor Position Control sessions. The error bands correspond to the standard error of the sample set. There is very clear learning occurring between the PreTest and MidTest while a ceiling effect is observed between the MidTest and PosTest

This was likely attributed to fatigue in the young subjects.

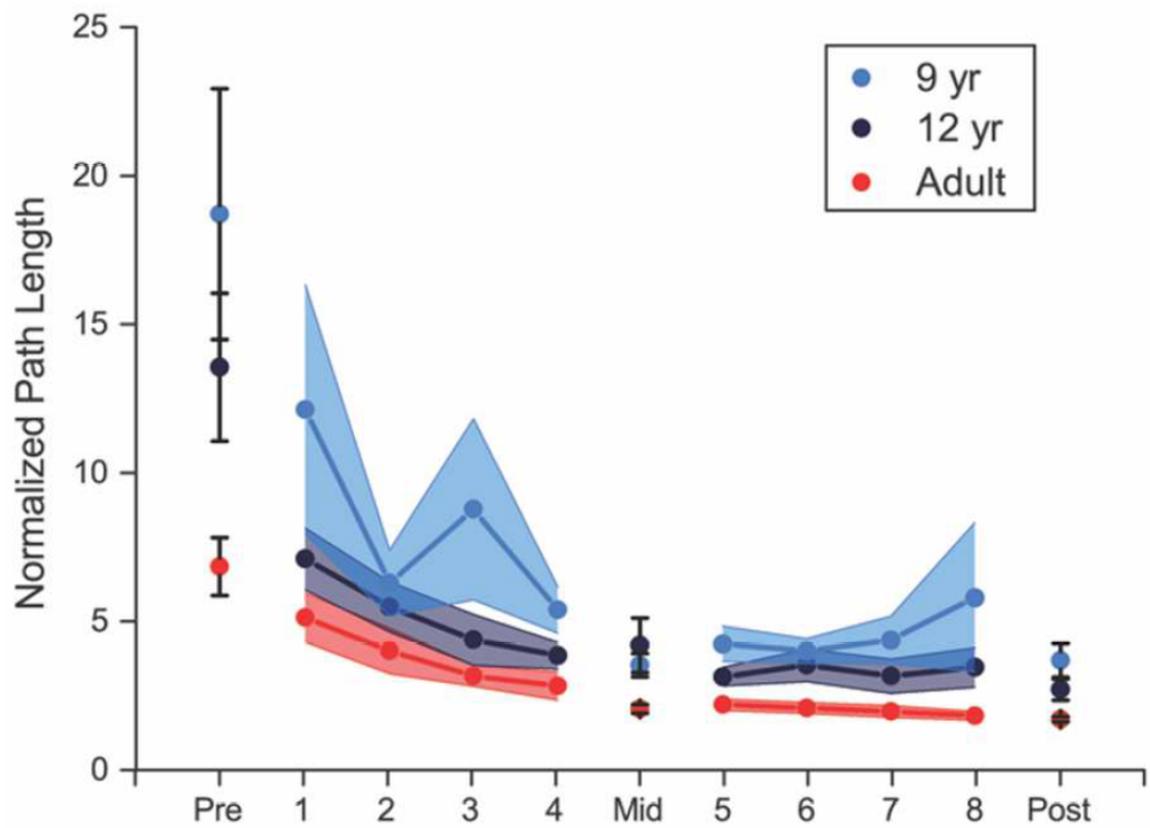


Figure 12: This Figure was reproduced with permission from Children show limited movement repertoire when learning a novel motor skill [25]. Mean normalized path length for 2D cursor position control sessions. The error bands correspond to the standard error of the sample set

4 VIRTUAL BODY MODEL: THEORETICAL DEVELOPMENT

As mentioned in the introduction, a Virtual Body Model was developed as a BoMI control method for translating the user's motions into values to be sent to the robot controller. This control interface was developed as a group effort and then integrated into the larger program and tested. Results of this work were submitted to the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017) [27]. This chapter is adapted from Section III of that submission which is under review at the time of this publication.

4.1 Finite Element Model

The VBM system is comprised of a finite element model (FEM) of the applicable body segments of the user. Depending on which portions of the body the user still has adequate control over, the model is created with dimensions corresponding to the user's physical measurements as can be seen in Figure 13. Because this model was developed with the express aim of helping a child without arms or legs and confined to a wheelchair, only a torso, neck, and head were modeled in the VBM. However, if a user were to have control of additional appendages such as an upper portion of an arm, then additional segments could be added to the model with appropriate sensors. On the other hand, if a user has more severely limited control of certain parts of the body, then additional constraints can be applied to the model.

Additionally, while the model was treated as an isotropic structure, it can be made more complex in the future to better represent the natural stiffness properties of the user's body. Both anatom-

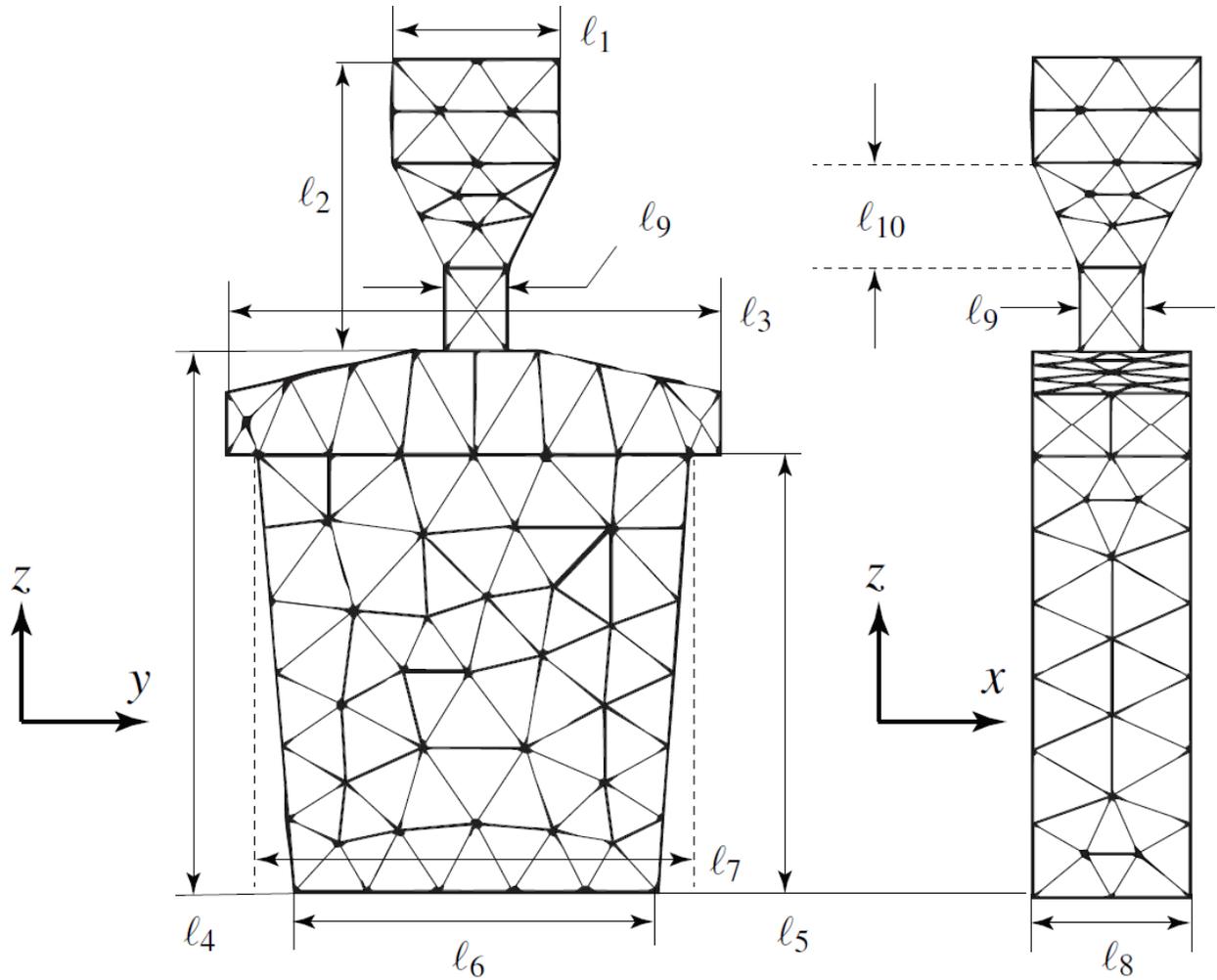


Figure 13: VBM dimensions: head width l_1 , distance from top of the head to base of the neck l_2 , shoulder span l_3 , torso height l_4 , length from under arm to waist l_5 , width of waist l_6 , chest span l_7 , torso thickness l_8 , neck thickness and width l_9 , and height of the lower portion of the head l_{10}

ical and physiological properties may be taken into account due to potential conditions such as muscle control degeneration, weakness, or paralysis.

The finite element model of the structure was meshed using algorithms in ANSYS, a commercially available software. The element type was chosen as tetrahedral with each node having three DOFs (x, y, z) in the inertial Cartesian space. Following usual FEM convention and using a total number of n nodes, the displacements of the i -th node could be represented by (u_{xi}, u_{yi}, u_{zi}) , $i = 1, 2, 3, \dots, n$, and the components of external forces acting on that node could be represented

as (f_{xi}, f_{yi}, f_{zi}) . With these forces and displacements in the inertial x, y, z Cartesian system and assuming a linearly elastic deformation model, the equilibrium equation can be written as

$$F = KU \quad (2)$$

where

$$F \triangleq [f_{x1} \ f_{y1} \ f_{z1} \ \cdots \ f_{xn} \ f_{yn} \ f_{zn}]^T$$

$$U \triangleq [u_{x1} \ u_{y1} \ u_{z1} \ \cdots \ u_{xn} \ u_{yn} \ u_{zn}]^T$$

and K is the $3n \times 3n$ stiffness matrix with elements derived by standard finite element methods.

In order to more easily replicate a human user's movements and because K is guaranteed to be invertible, Equation (2) can be modified as

$$U = CF, \quad C \triangleq K^{-1} \quad (3)$$

where C is the compliance matrix. Because K is a function of the non-displaced model, it is a constant matrix and so it can be computed offline, and subsequently, so can C .

4.2 A Set of Five Anatomically Distinct Motion Patterns

In order to control the robotic device, we define a set of anatomical movement patterns that will be mapped to specific commands. These body movement patterns are chosen such that the user will find them intuitive as well as distinct and able to be combined. This will enable the user to more easily control single and multiple DOFs as they choose. By having the movement patterns match natural macro-body movements, the user should be able to easily learn the mapping between a body movement and a resulting robot motion. They should also consider the movements to be relatively simple to use for extended periods of time. We have defined the five movement patterns as shown in Figure 14 with (a) defining the undeformed state:

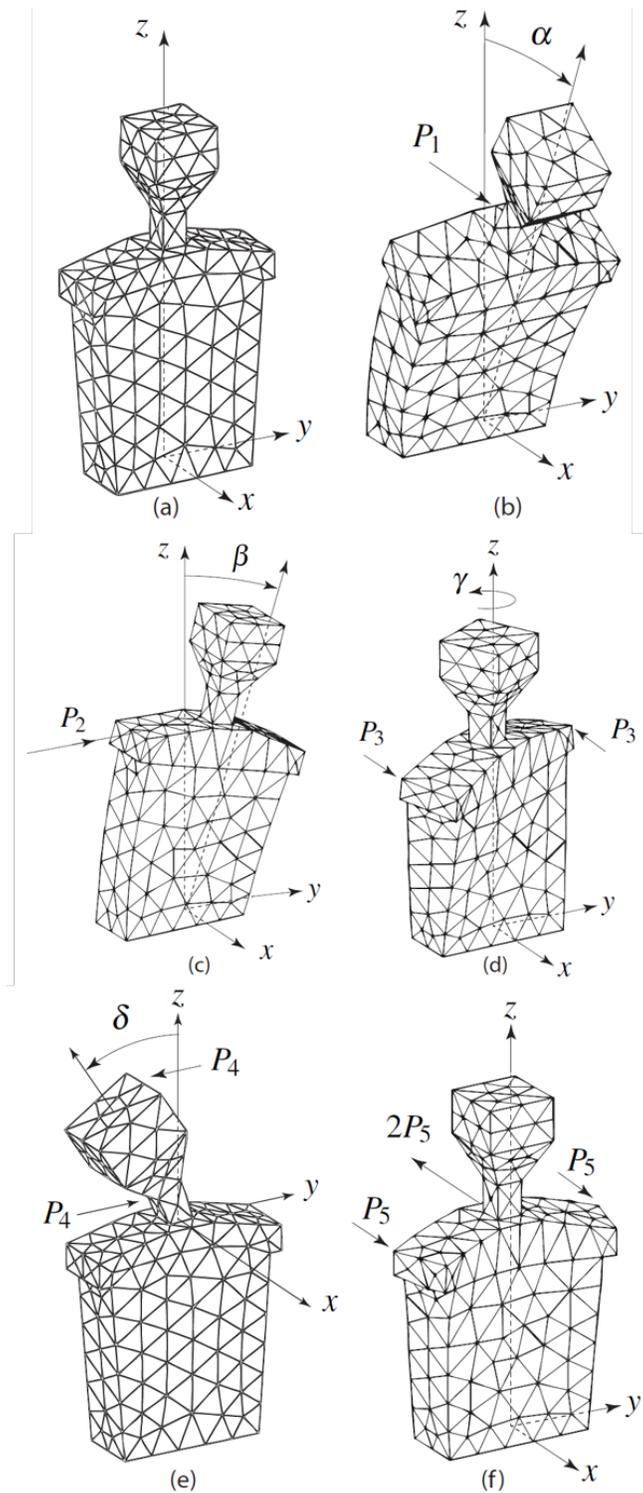


Figure 14: (a) Undeformed configuration of the VBM. (b), (c), (d), (e), and (f) show five deformed configurations of the VBM corresponding to five anatomically distinct motion patterns of the upper body. The five anatomically distinct motion patterns are: (b) torso flexion and extension, (c) torso lateral flexion, (d) torso rotation, (e) neck lateral flexion, and (f) scapular protraction and retraction.

1. Torso flexion and extension - This motion pattern relates to leaning forward and backward; it can be produced in the VBM by applying the single force P_1 , as shown in Figure 14(b). Since the virtual body bends forward in the x direction by angle α , this motion will be mapped to the translational movement of the end-effector in the x direction.
2. Torso lateral flexion - This motion pattern relates to leaning to the left and right; it can be produced in the VBM by applying the single force P_2 , as shown in Figure 14(c). Since the virtual body bends laterally in the y direction by angle β , this motion will be mapped to the translational movement of the end-effector in the y direction.
3. Torso rotation - This motion pattern relates to twisting the torso about the spinal column axis; it can be produced in the VBM by applying the couple P_3 shown in Figure 14(d). Since the virtual body rotates about the z axis by angle γ , this motion will be mapped to the translational movement of the end-effector in the z direction following the right hand rule.
4. Neck lateral flexion - This motion can be produced in the VBM by applying the couple P_4 shown in Figure 14(e). Since the head rotates about the x axis by angle δ , this motion will be mapped to the rotational movement of the end-effector about the instantaneous local x axis of the end-effector, known as the “roll” angle.
5. Scapular protraction and retraction - This motion can be produced by the set of three equilibrating forces P_5 shown in Figure 14(f). This motion of the virtual body will be mapped to opening and closing motion of the end-effector.

Two further motions that this process can easily be extended to are neck flexion and extension and neck rotation. Neck flexion and extension can be applied with a couple of forces at the node

located at the back of the head and center of neck. This can control the rotational movement of the end-effector about the instantaneous local y axis of the end-effector, known as the “pitch” angle. Neck rotation can be applied with a couple of forces at nodes located at the sides of the head which can control the rotational movement of the end-effector about the instantaneous local z axis of the end-effector, known as the “yaw” angle. These would complete the set for controlling the seven degrees of freedom of the robotic manipulator. However, for this proof-of-concept experiment, we limited our analysis to five body motions defined for controlling five DOFs of the robotic arm: movement in inertial x, y, z directions, rotation of the end-effector about the inertial z (vertical) axis, and opening and closing of the end-effector.

4.3 Virtual sensors

In order to translate movements made by the user into displacements of the VBM, virtual sensors are placed on the VBM to represent the physical wireless IMUs on the user. The virtual sensors are placed so that their location and orientation are most closely aligned with those of the physical wireless IMUs. The location of each virtual sensor is defined by selecting a triangular face of one of the tetrahedral elements on the surface of the VBM and defining one of its vertices, a_j , as the location of the virtual sensor. As the wireless IMUs only return values for rotations, the virtual sensors must also deal with rotations as opposed to simple displacements, as is the norm for finite-element models. The orientation of the virtual sensor is defined by selecting the location node of the triangle on the surface of the VBM, and defining unit vectors $\zeta_{j,orig}$ and $\eta_{j,orig}$ as those pointing from the location node, a_j , to the other two nodes of the triangle b_j and c_j , as shown in Figure 15. The rotation of the virtual sensor can then be defined as the change in the unit vectors

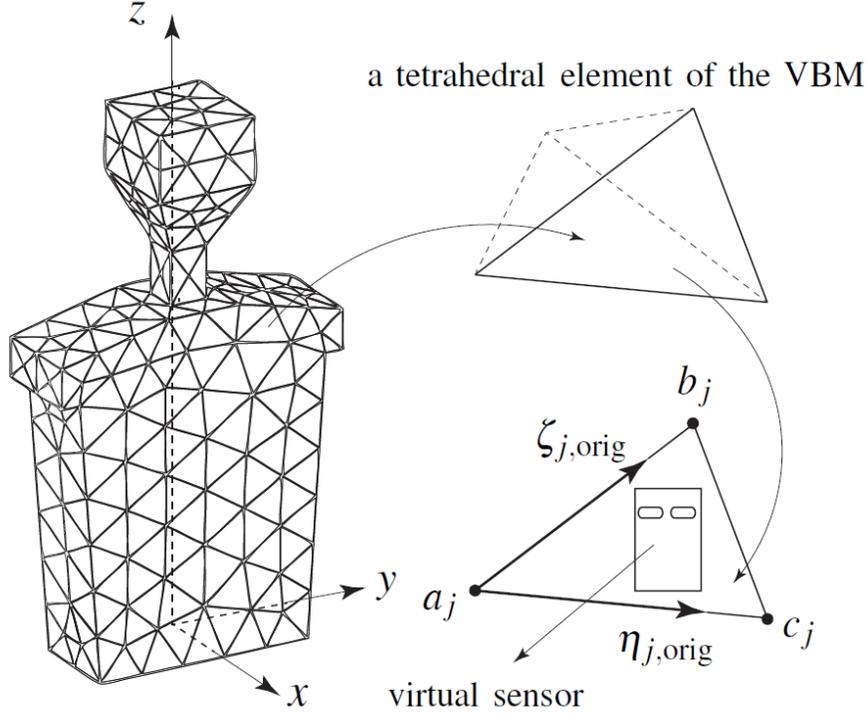


Figure 15: Placement of the j -th virtual sensor, $j \in 1, 2, \dots, k$, on the triangular face of a tetrahedral element of the VBM. The nodes of the triangular element are marked as a_j , b_j , and c_j ; a_j is arbitrarily selected as the base node, and $\zeta_{j,\text{orig}}$ and $\eta_{j,\text{orig}}$ are unit vectors from the base node to the other two nodes in the initial undeformed configuration.

ζ_j and η_j as a function of the relative displacement of the nodes of the triangular face.

To capture the three displacements of the three nodes of each virtual sensor, \bar{U} is defined as the vector of $9k$ elements containing the displacements of the k virtual sensors. To capture the fact that only five elements of F , the force vector defined previously, are independent, P is defined.

Therefore we can define

$$\bar{U} = \bar{C}P, \quad P \triangleq \begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 \end{bmatrix}^T \quad (4)$$

with \bar{C} as the simplified compliance matrix with $9k \times 5$ elements that captures the reduced number of displacements and forces relevant to the VBM. The five P_i elements can be considered a direct

mapping of the user's body movements to the anatomical body movements described previously. These forces can then be sent to the robotic arm as commands after scaling. This can be done because each of the forces independently causes one of the anatomical body movements.

4.4 Mapping of User Movements to Matching Body Movements of VBM

In order to map the user's movements into the anatomical movement patterns previously defined, two steps needed to be taken. In the first step, the system used the measurements from the wireless IMUs to determine how the VBM should deform. In the second step, the system determined what forces P_i , for $i = 1, 2, \dots, 5$, were needed to cause this deformation, or at least minimize the error between the deformation of the VBM caused by the forces and that as defined by the wireless IMU input. For the first step, it is noted that each wireless IMU provided a set of three Euler angles that could be used to compute a rotation matrix that transformed an initial orientation vector to a deformed vector. This rotated vector could be used to determine the desired orientation of unit vectors in the local neighborhood for the location nodes defined for each virtual sensor described above. If a_j , b_j , and c_j denote the three specific nodes of the VBM associated with the j -th virtual sensor, $\zeta_{j,orig}$ and $\eta_{j,orig}$ define the unit vectors in the initial undeformed configuration as shown in Figure 15, and R_j denotes the rotation matrix obtained from the j -th wireless IMU, the desired unit vectors $\zeta_{j,des}$ and $\eta_{j,des}$ can be obtained using the relations

$$\zeta_{j,des} = R_j \zeta_{j,orig}, \quad \eta_{j,des} = R_j \eta_{j,orig}, \quad j = 1, 2, \dots, k \quad (5)$$

To determine the forces P in the second step, we start with the initial guess $P = 0$. Equation (4) is used to obtain a functional description of the unit vectors as follows

$$Y = g(P), \quad Y \triangleq [\zeta_1 \quad \eta_1 \quad \dots \quad \dots \quad \zeta_k \quad \eta_k]^T$$

This function $g(P)$ represents the change in each unit vector given a unit input of each force P_i . As the system is linear and isotropic, this vector function can be computed offline. The Jacobian of the vector function is computed numerically and the Newton-Raphson method is used along with the method of least-squares [28] to estimate P as follows

$$P = (J^T J)^{-1} J^T (Y_{\text{des}} - Y_{\text{orig}}), \quad J \triangleq \left[\frac{\partial g}{\partial P} \right] \quad (6)$$

In the above equation, J is the Jacobian matrix and Y_{des} and Y_{orig} are defined as

$$Y_{\text{des}} \triangleq \left[\zeta_{1,\text{des}} \quad \eta_{1,\text{des}} \quad \cdots \quad \cdots \quad \zeta_{k,\text{des}} \quad \eta_{k,\text{des}} \right]^T$$

$$Y_{\text{orig}} \triangleq \left[\zeta_{1,\text{orig}} \quad \eta_{1,\text{orig}} \quad \cdots \quad \cdots \quad \zeta_{k,\text{orig}} \quad \eta_{k,\text{orig}} \right]^T$$

The forces P obtained from Equation (6) are the projections of the movement of the user onto the different anatomically distinct motion patterns, as discussed earlier in this section. These are the values that will be scaled and sent to the robotic arm as velocity commands.

5 VIRTUAL BODY MODEL: EXPERIMENTAL SETUP AND PROGRAM DESCRIPTION

5.1 Experimental Hardware

For the VBM verification testing, the same hardware was used as in the PCA testing. However, five wireless IMUs were used instead of four. The VBM locations on the torso were chosen to fully capture the necessary independent DOFs for the first four body movement patterns. The two wireless IMUs mounted on the cap were set to redundantly capture the orientation of the user's head. Assuming that the cap, fitted to the head, was a rigid body, then the rotations matrices derived from the Euler angles output from the two wireless IMUs on the head, should be the same. However, having two wireless IMUs provided a buffer against errant signals due to the averaging of the two sets of values. Figure 16 shows the locations of the five wireless IMUs for VBM.

5.2 Experiment and Program Description

The core of the PCA program was adapted for use in the VBM control method. Due to the tasks and the user input to the robot command algorithm being different, much of the goal-testing and boundary checking subroutines were discarded. The startup procedure was similar to that defined for the PCA process, however, there was no need for the configuration matrix defining the principle axes. The JACO² arm initialization process was edited to have a new starting position and orientation, but otherwise remained the same. The IMU reader thread was simply adapted to

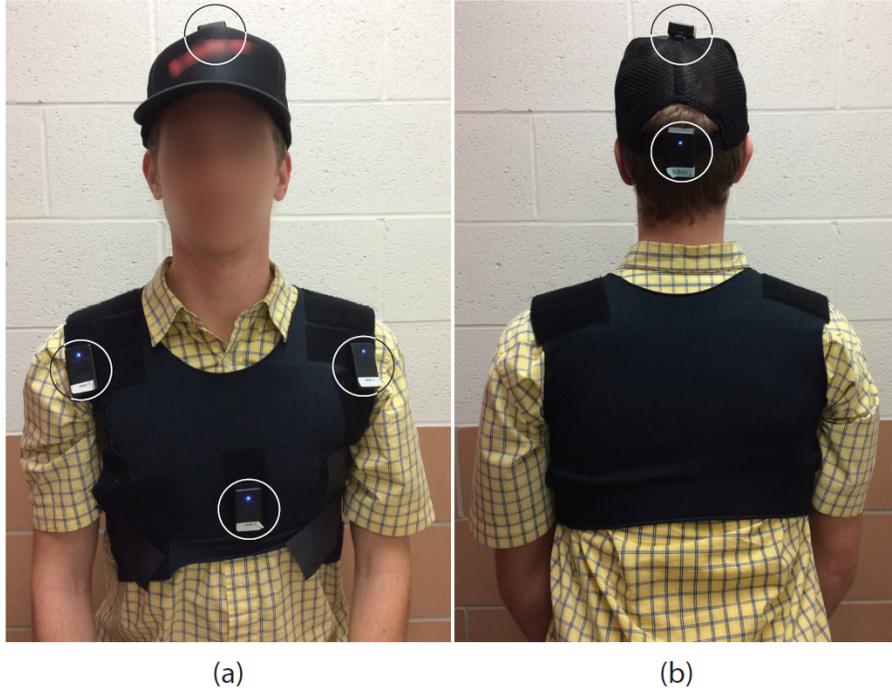


Figure 16: Five wireless IMUs attached to the cap and vest of the user are circled in the (a) front and (b) back views of the user: three are located on the vest and the other two are placed on the top and back of the cap.

handle the new number of wireless IMU inputs as well as incorporating the previously ignored yaw values. The data output thread was also changed to match the new values being used.

A new thread was created in order to run the VBM MATLAB[®] code through the MATLAB[®] engine. The 15 values from the wireless IMUs were passed as an array to the MATLAB[®] engine which, following the procedure outlined in the previous chapter, determined the five forces required to cause the VBM model to closely match the user's pose. These were then each scaled using gains determined experimentally. In order to create a buffer-zone of no movement, similar to that described in the PCA section, the magnitude of each scaled signal was then tested against a lower bound to see if it was large enough. This lower bound served a purpose similar to that of the buffer zone in Chapter 2. The Cartesian velocities were saturated at a level corresponding to the maximum velocity of the JACO² arm (20 cm/s) as prescribed by the manufacturer of the robot.

There was no longer a need to ensure that the hand of the robot remained in the plane of the screen, nor in the octagonal region. However, a lower bound in the z -direction was enforced to ensure that the robot would not contact the table surface. The program did not conclude after a specific task was accomplished according to the code, but rather when an interrupt was sent to the program by the person overseeing the testing.

5.3 Robot Control with User-in-the-Loop

Figure 17 shows a block diagram of the feedback loop used to control the robot with both the robotic elements and the user error control. The current configuration of the robot is compared to its desired configuration by the user. The user performs one of the predetermined body movement patterns, proportional to the error, in order to get the robotic arm to move towards the desired configuration and location. The VBM solves for the forces required to match the configuration of the wireless IMUs. These forces are then mapped to velocities that are relayed to the robot's internal controller that determines the inputs to provide to the joint actuators to realize the desired movement.

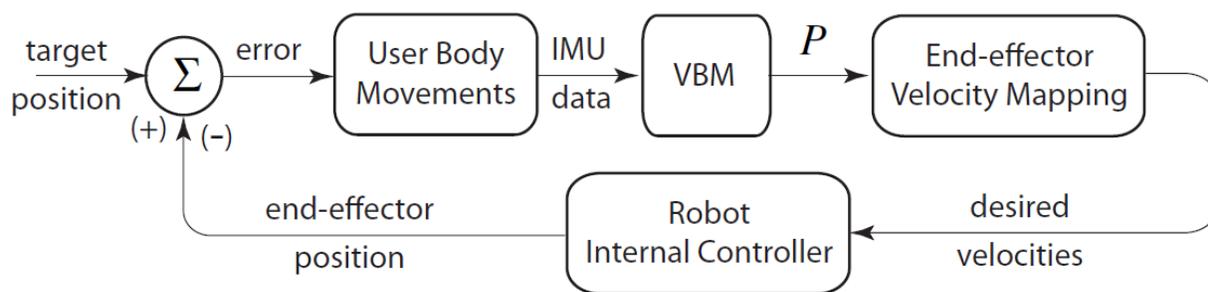


Figure 17: Block diagram of VBM robot control system with user-in-the-loop

6 VIRTUAL BODY MODEL:

EXPERIMENTAL RESULTS

This chapter presents the proof-of-concept validation of the VBM approach. As with Chapter 4, this chapter is largely based on the work submitted to 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017) [27]. This chapter is adapted from Section V of that submission.

6.1 Virtual Body Model of the User

The tests were conducted using the JACO² arm and the five wireless IMUs from Yost Labs Inc. mounted with a hook and loop on a vest and a baseball cap as described in Chapter 5 (see Figure 16). An able-bodied and cognitively sound adult user was chosen as the test subject. The measurements of his body, as defined in Chapter 4, were found to be

$$\ell_1 = 16.0 \quad \ell_2 = 32.0 \quad \ell_3 = 47.5 \quad \ell_4 = 52.0 \quad \ell_5 = 42.0$$

$$\ell_6 = 35.0 \quad \ell_7 = 41.5 \quad \ell_8 = 16.0 \quad \ell_9 = 6.0 \quad \ell_{10} = 12.0$$

with units of centimeters. These values were applied to the VBM model and the ANSYS generated mesh resulted in 883 elements and 283 nodes. Using an isotropic Young's Modulus of 4000 MPa and a Poisson's ratio of 0.33, the stiffness matrix K and the reduced compliance matrix \bar{C} were generated. Five virtual sensors, matching the location and orientation of the physical wireless IMUs of Figure 16 were placed on the VBM.

6.2 Verification of matching of user body movements to Virtual Body Model

To ensure that the VBM was able to closely match the configuration of the user as measured by the wireless IMUs, an initial verification of the system was conducted using various poses. These poses were chosen to be relatively complex in order to test whether the system would be able to produce reasonable outputs for the user's body configurations that were attempting to control multiple degrees of freedom simultaneously. Figures 18-20 depict a series of user orientations along with their corresponding VBM configurations. With the well-defined forces P_i , for $i = 1, 2, 3, 4, 5$, as described in Section 4.2, the control of the robot is intuitive and these gross body movements cause the expected reactions from the robot. The movements presented below were exaggerated so as to more easily show the quality of the match between the user configuration and that of the VBM. In practice, the motions were not required to be nearly as large due to sufficiently high gains of the system.

During live testing, the VBM portion of the program was able to cycle at 14 Hz on average. Due to the asynchronous nature of the entire program, as made possible through threading, this did not affect the 100 Hz rate at which the commands were communicated to the robotic arm. This rate was the result of the both inter-process communication as well as the MATLAB® computation time required to solve for the five P_i forces. It was determined that this rate was sufficient for the user to not be bothered by noticeable lag and still consider the response of the robotic arm to be "real-time." Additionally, the targeted users of this program would be focused more on the functionality that this system provides, rather than the speed at which the system functions.

No optimization has been done for either the computation time or the inter-process communication, though several possibilities have been discussed. A prime candidate for optimization of the

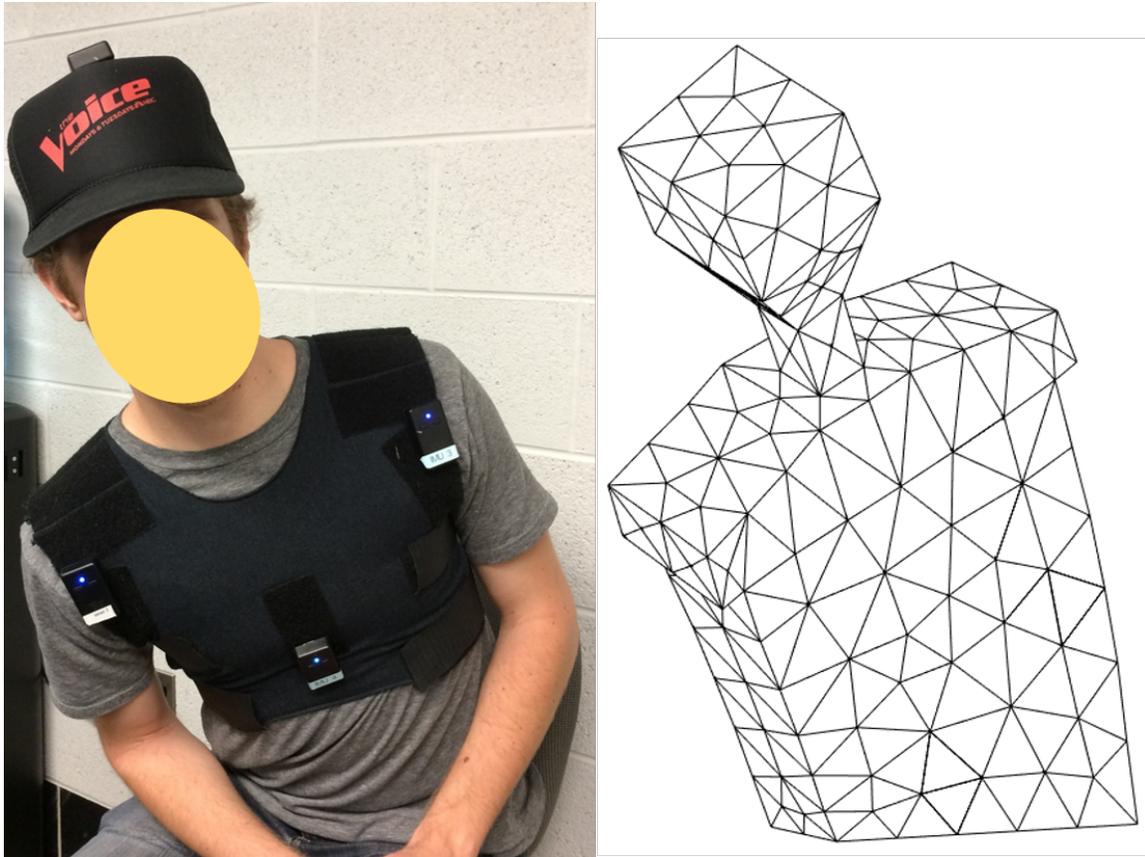


Figure 18: User posed in first orientation along with matching VBM configuration. The resultant forces describe a positive force in P_1 causing the torso flexion and a positive force, P_3 , causing the torso rotation. The head remains neutral and is thus aligned with the shoulders. This would have caused the robot to move forward and upwards as described in Section 4.2

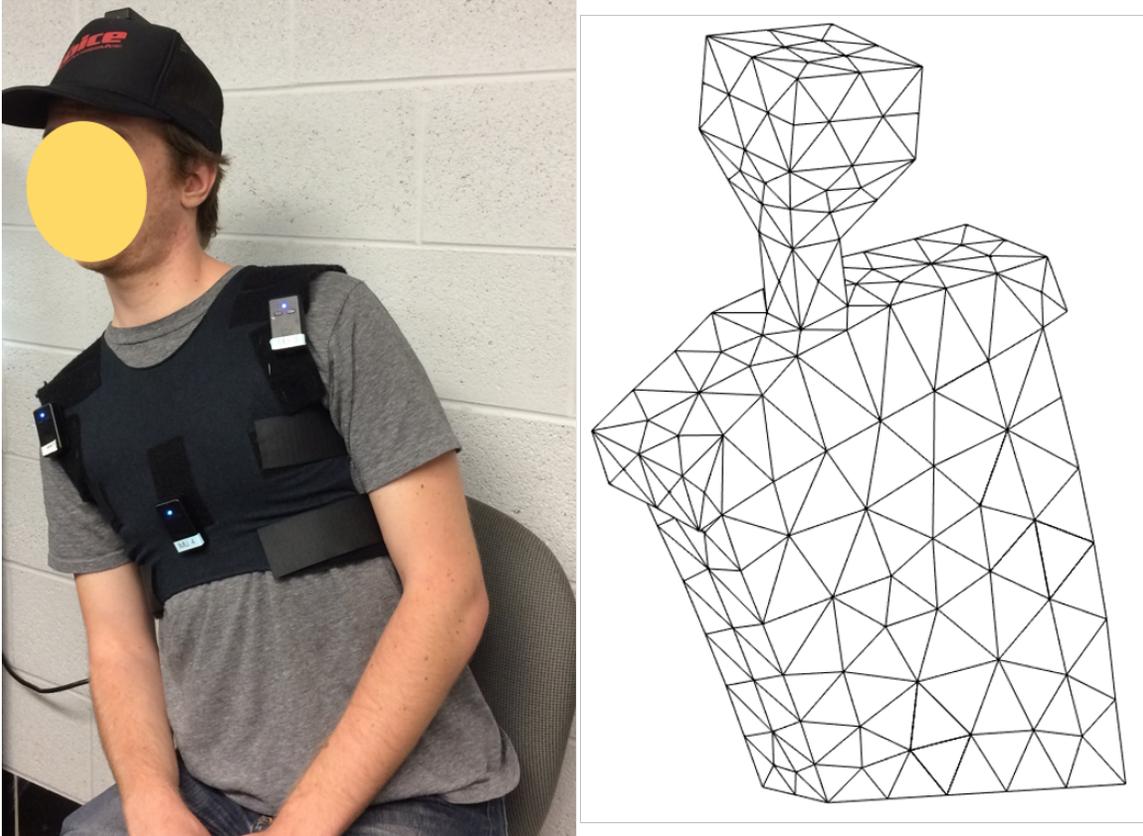


Figure 19: User posed in second orientation along with matching VBM configuration. The resultant forces describe a negative force in P_2 causing the lateral torso flexion and a negative P_4 causing the head to remain upright. This would have caused the robot to move to the left and end-effector to spin in a negative direction as described in Section 4.2

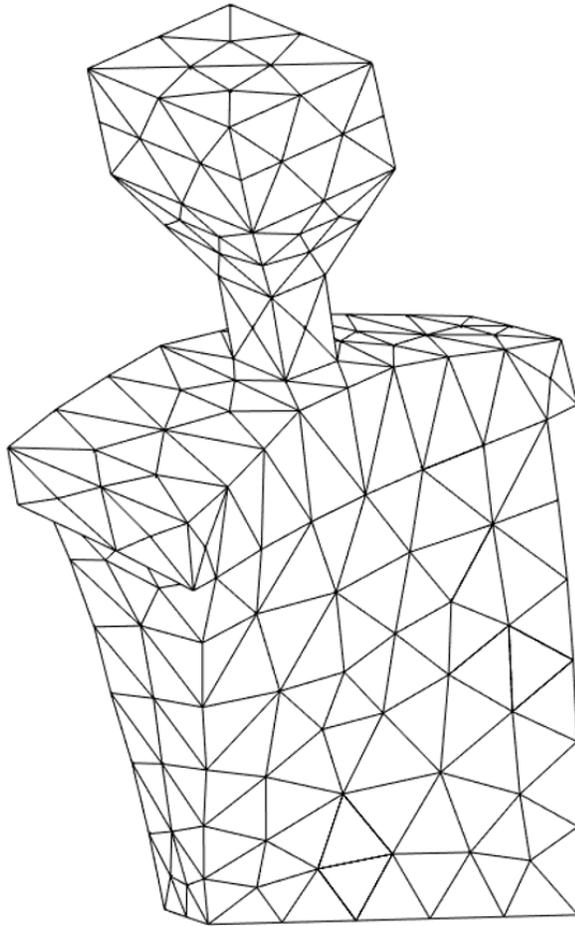
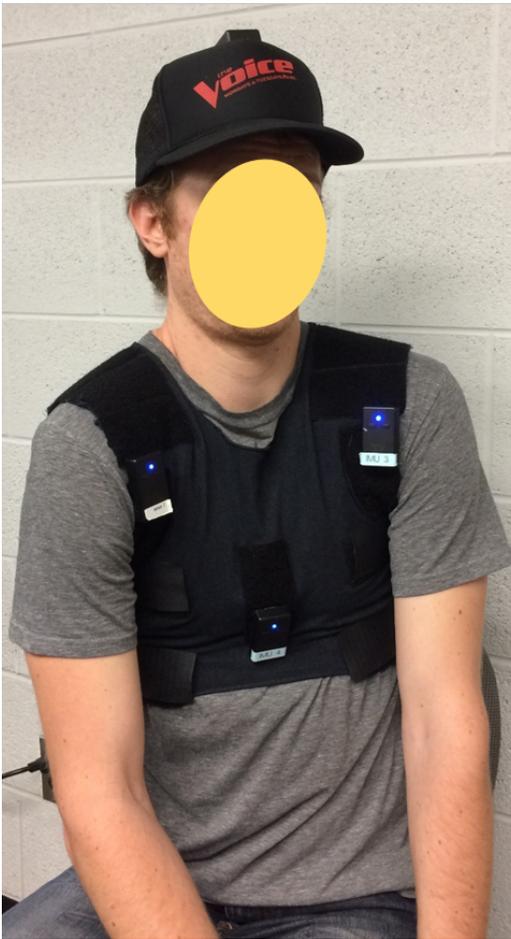


Figure 20: User posed in third orientation along with matching VBM configuration. The resultant forces describe a positive force in P_3 causing the torso rotation and a positive force in P_5 causing the scapular protraction. This would have caused the robot to move upward and the end-effector to close as described in Section 4.2

code with the linear finite element model with isotropic properties is to calculate $g(P)$ offline and therefore be able to calculate its Jacobian, J , and then $(J^T J)^{-1} J^T$ offline during initialization. This was not done for testing in order to ensure robustness of the program due to an error checking step being incorporated during the computation of J . Another possible improvement to the program is to provide an initial guess to the system based on the previous result. However, this could have integrative effects on error and would need to be studied more carefully before implementation. Increasing the number of threads utilized by the process could also improve speed, if appropriate. Irrespective of the level of code optimization, more powerful processors would also be able to increase the rate at which the forces are calculated.

6.3 Experimental Task: Pick and Place

The task designed for the user was to begin by moving the robot's end-effector from an initial position to where a plastic water bottle was located. After grasping the water bottle, the robot lifted the bottle and then set it down on a raised box 41 *cm* away and 8 *cm* up. The final portion of the task was to successfully release the bottle and move the end-effector away from the bottle without disturbing it significantly. The user was seated approximately 40 *cm* in the negative y -direction and 20 *cm* in the negative x -direction relative to the base of the robot while facing in the positive x -direction. Multiple practice runs were conducted before the recorded trial.

Figure 21 shows the trajectory of the end-effector and Figure 22 shows the x, y, z positions of the end-effector, wrist rotation angle, and end-effector grasping configuration as a function of time. The results of the user trial are described below with the help of Figure 21 and Figure 22:

- From $t_0 = 0.0 \text{ sec}$ to $t_1 = 7.0 \text{ sec}$, the BoMI was used to rotate the wrist to achieve “ready-

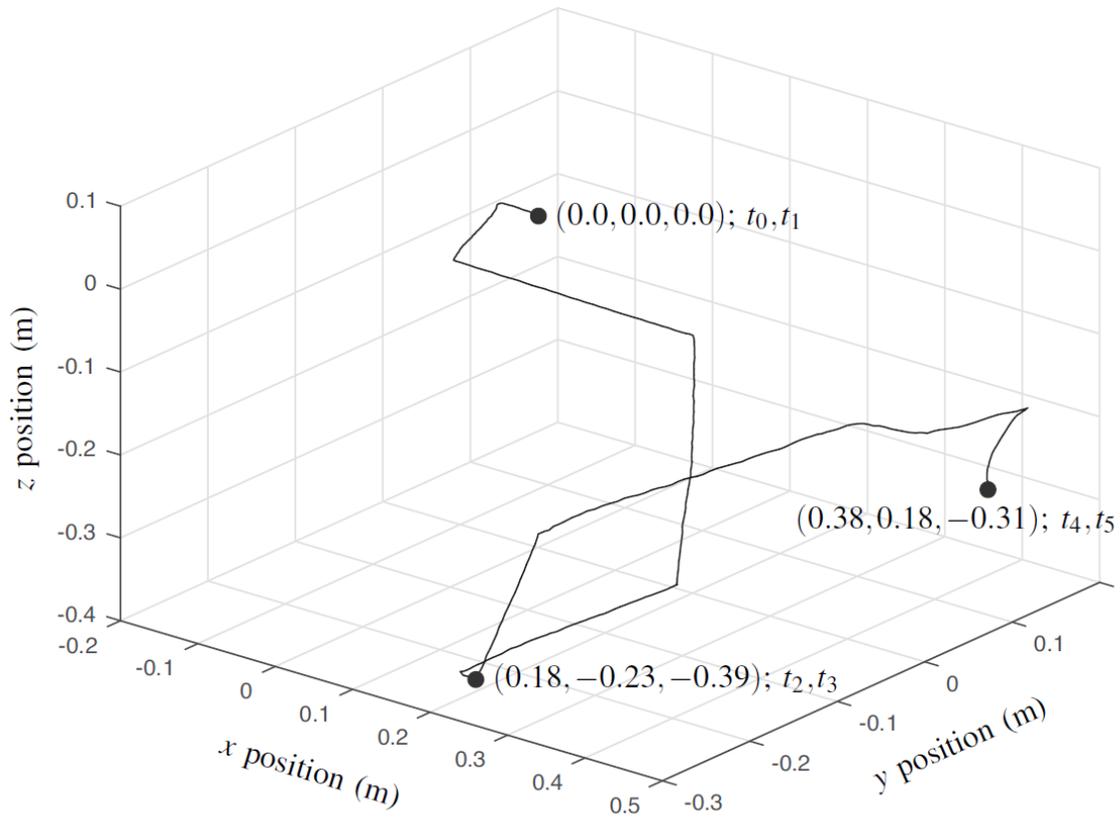


Figure 21: End-Effector trajectory in Cartesian coordinates during user trial

to-grasp” configuration - see Figure 22. During this time period, the end-effector position did not change significantly.

- From $t_1 = 7.0 \text{ sec}$ to $t_2 = 25.2 \text{ sec}$, the BoMI was used to move the end-effector to the location of the water bottle along the trajectory shown in Figure 21. It can be seen from Figure 22 that x, y, z positions of the end-effector changed during this time period.
- From $t_2 = 25.2 \text{ sec}$ to $t_3 = 31.1 \text{ sec}$, the BoMI was used to grasp the water bottle; the end-effector was originally in its open configuration - see Figure 22.
- From $t_3 = 31.1 \text{ sec}$ to $t_4 = 50.0 \text{ sec}$, the BoMI was used to move the water bottle to its desired location; during this time, the end-effector remained closed.

- From $t_4 = 50.0 \text{ sec}$ to $t_5 = 56.6 \text{ sec}$, the BoMI was used to release the water bottle.

The user was then able to move the end-effector away without touching the bottle.

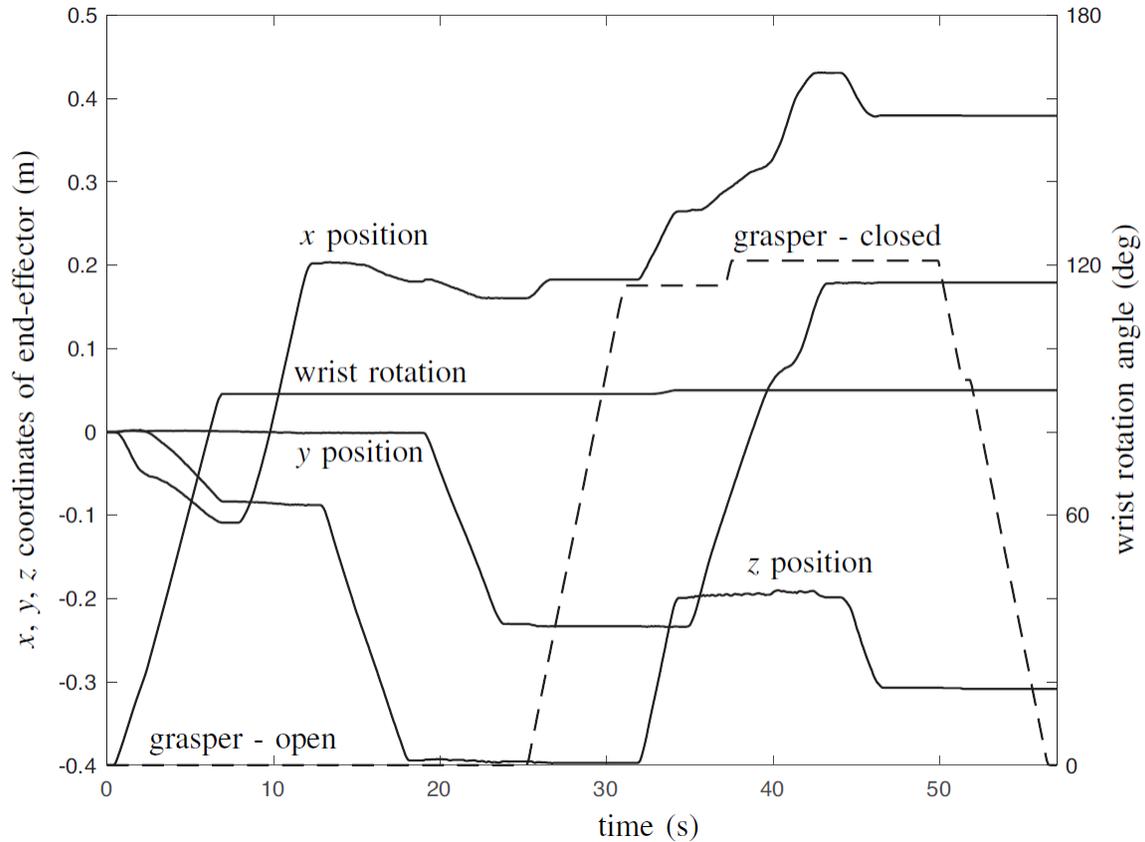


Figure 22: Variation of Cartesian coordinates of the end-effector, wrist rotation, and end-effector configuration with time for the user trial

The entire task was accomplished in 56.6 sec . The smooth trajectory with minimal overshoot shows that the user was able to exhibit a good understanding of the mapping from body movements to robot controls in regards to both direction and magnitude. Because the user only had a few times to practice with the VBM control method before the actual trial, it is expected that the time to complete the task would decrease as the user becomes more comfortable with the defined body movements. Notably, while in this example, the user primarily controlled the robotic arm in a single DOF at a time, with additional practice, a more experienced user will be able to effectively

control more DOFs simultaneously. Additionally, with feedback provided by the user, the gains and filters on the individual velocity commands could be tuned to best fit the user. It should also be noted that the maximum speed of the robot was limited by the firmware for safety reasons.

Further trials showed that the user was able to grasp a water bottle that had a straw coming out the hole for drinking and bring it to the user's mouth. The user was then able to drink from the straw while keeping the bottle stationary and then control the robot to bring the bottle away from the user's face and set it back down on the table.

7 CONCLUSIONS

For this thesis, a 2D position cursor control experiment from the literature [25] was adapted to 2D robotic end-effector velocity control to study the ability of participants at different ages to learn to control the robot. Additionally, this thesis presents the development and testing of a novel VBM method as an approach to controlling many DOFs of a robotic arm in an intuitive manner using only movements of the torso and head of the user as inputs.

7.1 PCA Experiment

For the PCA experiment, we replicated the 2D position cursor control experiment of [25] and adapted it to velocity control of the robot end-effector in the same two dimensions. This involved redeveloping the original MATLAB[®] program running in a Windows environment into a C++ program running in a Linux environment.

The users were instructed to move their shoulders and torso freely for a full minute to gather their range of motion as recorded by wireless IMUs. These values were used to generate the two principle components of the user's movement space. For the reaching task component of the experiment, participants were instructed to use a subset of these motions to control the tip of the robotic end-effector and move it into a set of goal areas, holding the position for 500 *msec*. Results from the 2D velocity robot control were similar to those from the 2D position cursor control task in that tremendous learning was shown between the PreTest and the MidTest before the participants reached a performance ceiling. Additionally, a tracing task was performed to measure the participants fine control of the robot along curved paths.

7.2 Development and Experimental Validation of the VBM Method

Through this work, an intuitive and effective method for controlling several DOFs of a robotic arm was successfully developed and validated. A finite element model was designed to match the physical dimensions of the user upon which virtual sensors were placed corresponding to physical wireless IMUs on the user. Forces were then defined that pertained to natural body movement patterns which were intuitively applied by the user to control the velocity of the robotic end-effector as well as its orientation and the opening of the fingers. Using this method, a user was able to conduct a pick-and-place task using a water bottle, as well as successfully grasping a water bottle and drinking from it using a straw. The experiment was done with control over five DOFs, but the system can easily be expanded to seven DOFs allowing full user control and grasping ability in 6D space. This is especially significant because BoMIs have never controlled more than two DOFs, let alone five DOFs that are easily expandable to seven DOFs.

7.3 Further work

For the PCA study, an investigation of the PCA matrix configurations of the participants, as related to their attrition or success, could provide useful insights into their success and rate of learning. We may discover that those who did not complete the experiment were presented with highly non-intuitive principle directions that made learning very difficult. Additionally, closer analysis of the tracing task can be conducted to study the fine control of the robotic arm along non-straight curves as opposed to the gross directional motions involved in the reaching task.

It is clear that the VBM model can easily be extended to seven DOFs with the addition of the extra two rotations controls of the end-effector associated with the tilting the head forward

and backward and turning it side-to-side. This would enable the user to have full control of the robot's full range of motion. Additionally, parameters such as gains and dead zones of control will continue to be tuned in order to achieve ease of control and comfort of the user. Currently, work is being undertaken to include computer vision and recognition capabilities in the system to assist the user in tasks involving precision or multiple and simultaneous DOFs of the end-effector such as opening a door. Touch, force, and proximity sensors could be added to enable the user to interact with objects that are partially occluded from view.

One of the downsides of the VBM method is that adapting the finite element model and defining the distinct motion patterns for each user is a work-intensive design process. For users with restricted motion capabilities due to conditions such as muscle control degeneration, weakness, arthritis, or paralysis, the PCA method can be leveraged as a means of determining potential distinct motion patterns that the user is easily capable of doing.

REFERENCES

REFERENCES

- [1] Matthew W Brault et al. Americans with disabilities: 2010. *Current population reports*, 7:0–131, 2012.
- [2] F. Marciano. Self leveling spoon, November 3 2016. US Patent App. 15/107,527.
- [3] Nikolaos Mavridis. A review of verbal and non-verbal human–robot interactive communication. *Robotics and Autonomous Systems*, 63:22–35, 2015.
- [4] Richard C Simpson. Smart wheelchairs: A literature review. *Journal of rehabilitation research and development*, 42(4):423, 2005.
- [5] Tiffany L Chen, Matei Ciocarlie, Steve Cousins, Phillip M Grice, Kelsey Hawkins, Kaijen Hsiao, Charles C Kemp, Chih-Hung King, Daniel A Lazewatsky, Hai Nguyen, et al. Robots for humanity: A case study in assistive mobile manipulation. 2013.
- [6] AMER Al-Rahayfeh and MIAD Faezipour. Eye tracking and head movement detection: A state-of-art survey. *IEEE journal of translational engineering in health and medicine*, 1:2100212–2100212, 2013.
- [7] Christine Bringes, Yun Lin, Yu Sun, and Redwan Alqasemi. *Determining the benefit of human input in human-in-the-loop robotic systems*. IEEE, 2013.
- [8] John P Donoghue. Connecting cortex to machines: recent advances in brain interfaces. *Nature neuroscience*, 5:1085–1088, 2002.
- [9] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791, 2002.
- [10] Jose M Carmena, Mikhail A Lebedev, Roy E Crist, Joseph E O’Doherty, David M Santucci, Dragan F Dimitrov, Parag G Patil, Craig S Henriquez, and Miguel AL Nicolelis. Learning to control a brain–machine interface for reaching and grasping by primates. *PLoS biol*, 1(2):e42, 2003.
- [11] Meel Velliste, Sagi Perel, M Chance Spalding, Andrew S Whitford, and Andrew B Schwartz. Cortical control of a prosthetic arm for self-feeding. *Nature*, 453(7198):1098–1101, 2008.
- [12] Leigh R Hochberg, Daniel Bacher, Beata Jarosiewicz, Nicolas Y Masse, John D Simeral, Joern Vogel, Sami Haddadin, Jie Liu, Sydney S Cash, Patrick van der Smagt, et al. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398):372–375, 2012.
- [13] Dennis J McFarland, Lynn M McCane, Stephen V David, and Jonathan R Wolpaw. Spatial filter selection for eeg-based communication. *Electroencephalography and clinical Neurophysiology*, 103(3):386–394, 1997.

- [14] Xiaogang Chen, Yijun Wang, Masaki Nakanishi, Xiaorong Gao, Tzyy-Ping Jung, and Shangkai Gao. High-speed spelling with a noninvasive brain-computer interface. *Proceedings of the national academy of sciences*, 112(44):E6058–E6067, 2015.
- [15] Stephen I Ryu and Krishna V Shenoy. Human cortical prostheses: lost in translation? *Neurosurgical focus*, 27(1):E5, 2009.
- [16] Rikky Muller, Hanh-Phuc Le, Wen Li, Peter Ledochowitsch, Simone Gambini, Toni Bjorninen, Aaron Koralek, Jose M Carmena, Michel M Maharbiz, Elad Alon, et al. 24.1 a miniaturized 64-channel 225 μ w wireless electrocorticographic neural sensor. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, pages 412–413. IEEE, 2014.
- [17] Kazuo Kiguchi and Qilong Quan. Muscle-model-oriented emg-based control of an upper-limb power-assist exoskeleton with a neuro-fuzzy modifier. In *Fuzzy Systems, 2008. FUZZ-IEEE 2008.(IEEE World Congress on Computational Intelligence). IEEE International Conference on*, pages 1179–1184. IEEE, 2008.
- [18] H Rex Hartson. *Advances in human-computer interaction*, volume 2. Intellect Books, 1988.
- [19] Rafael Barea, Luciano Boquete, Manuel Mazo, and Elena López. System for assisted mobility using eye movements based on electrooculography. *IEEE transactions on neural systems and rehabilitation engineering*, 10(4):209–218, 2002.
- [20] Todd A Kuiken, Guanglin Li, Blair A Lock, Robert D Lipschutz, Laura A Miller, Kathy A Stubblefield, and Kevin B Englehart. Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms. *Jama*, 301(6):619–628, 2009.
- [21] Anton Plotkin, Lee Sela, Aharon Weissbrod, Roni Kahana, Lior Haviv, Yaara Yeshurun, Nachum Soroker, and Noam Sobel. Sniffing enables communication and environmental control for the severely disabled. *Proceedings of the National Academy of Sciences*, 107(32):14413–14418, 2010.
- [22] Maura Casadio, Rajiv Ranganathan, and Ferdinando A Mussa-Ivaldi. The body-machine interface: a new perspective on an old theme. *Journal of Motor behavior*, 44(6):419–433, 2012.
- [23] Elias B Thorp, Farnaz Abdollahi, David Chen, Ali Farshchiansadegh, Mei-Hua Lee, Jessica P Pedersen, Camilla Pierella, Elliot J Roth, Ismael Seáñez Gonzáles, and Ferdinando A Mussa-Ivaldi. Upper body-based power wheelchair control interface for individuals with tetraplegia. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(2):249–260, 2016.
- [24] Marco Santello, Martha Flanders, and John F Soechting. Postural hand synergies for tool use. *Journal of Neuroscience*, 18(23):10105–10115, 1998.
- [25] Mei-Hua Lee. Children show limited movement repertoire when learning a novel motor skill.

- [26] R William Soukoreff and I Scott MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of fitts law research in hci. *International journal of human-computer studies*, 61(6):751–789, 2004.
- [27] Sheryl Chau, Sanders Aspelund, Ranjan Mukherjee, Mei-Hua Lee, Rajiv Ranganathan, and Florian Kagerer. A five degree-of-freedom body-machine interface for children with severe motor impairments.
- [28] Steven C Chapra. Applied numerical methods. *With MATLAB for Engineers and Scientists*, 2012.