A Machine Learning Approach to Intended Motion Prediction for Upper Extremity Exoskeletons

Justin Berdell
justinberdell@gmail.com

Follow this and additional works at: https://huskiecommons.lib.niu.edu/allgraduate-thesesdissertations

Part of the Artificial Intelligence and Robotics Commons, Electrical and Computer Engineering Commons, and the Robotics Commons

Recommended Citation
https://huskiecommons.lib.niu.edu/allgraduate-thesesdissertations/6853

This Dissertation/Thesis is brought to you for free and open access by the Graduate Research & Artistry at Huskie Commons. It has been accepted for inclusion in Graduate Research Theses & Dissertations by an authorized administrator of Huskie Commons. For more information, please contact jschumacher@niu.edu.
ABSTRACT

A MACHINE LEARNING APPROACH TO INTENDED MOTION PREDICTION FOR UPPER EXTREMITY EXOSKELETONS

Justin Berdell, M.S.
Department of Electrical Engineering
Northern Illinois University, 2022
Hasan Ferdowsi, Director

A fully solid-state, software-defined, one-handed, handle-type control device built around a machine-learning (ML) model that provides intuitive and simultaneous control in position and orientation each in a full three degrees-of-freedom (DOF) is proposed in this paper. The device, referred to as the “Smart Handle”, and it is compact, lightweight, and only reliant on low-cost and readily available sensors and materials for construction. Mobility chairs for persons with motor difficulties could make use of a control device that can learn to recognize arbitrary inputs as control commands. Upper-extremity exoskeletons used in occupational settings and rehabilitation require a natural control device like the Smart Handle that can detect and provide position and orientation trajectories to their kinematic models. Aerial and submersible vehicles that often require multiple inputs for positioning and throttling could see their control systems simplified by a technology like the Smart Handle which can offer both with the dedication of only one hand from the user. This study has shown that the Smart Handle device can learn to output a continuous range of translation and rotation information from simple training sets that consist only of examples where intended motion was restricted to varying magnitudes in a single DOF at a time. With personalized calibration, the Smart Handle can consistently classify movement with over ninety-five percent accuracy. Proof-of-concept experiments were successfully conducted on exoskeleton control applications as well as wheeled robot control.
NORTHER ILLINOIS UNIVERSITY
DEKALB, ILLINOIS

AUGUST 2022

A MACHINE LEARNING APPROACH TO INTENDED MOTION
PREDICTION FOR UPPER EXTREMITY EXOSKELETONS

BY
JUSTIN BERDELL
©2022 Justin Berdell

A THESIS SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE
MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

Thesis Director:
Hasan Ferdowsi
TABLE OF CONTENTS

LIST OF TABLES..................................................................................................................v

LIST OF FIGURES................................................................................................................vi

Chapter

1. INTRODUCTION.................................................................1

1.1 Background..............................................................................1

1.1.1 History.................................................................2

1.1.2 Uses.................................................................5

1.1.3 Future...............................................................6

1.2 Critical Dimensions of Exoskeletons.................................7

1.2.1 Hardware Arrangement.............................................8

1.2.2 Power Source.........................................................9

1.2.3 Actuators.............................................................10

1.2.4 Intended Motion and Control....................................13

1.2.5 Signal Processing...................................................14

1.3 Void Area: Intended Motion.............................................16

1.3.1 EMG Problems.....................................................16

1.3.2 Force Sensor Problems..........................................19

1.3.3 Cost Issues........................................................20

1.3.4 Setup Issues.......................................................21

1.3.5 Proposed Solutions................................................22
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4</td>
<td>28</td>
</tr>
<tr>
<td>1.4.1</td>
<td>29</td>
</tr>
<tr>
<td>1.4.2</td>
<td>30</td>
</tr>
<tr>
<td>1.4.3</td>
<td>30</td>
</tr>
<tr>
<td>1.4.4</td>
<td>31</td>
</tr>
<tr>
<td>1.5</td>
<td>32</td>
</tr>
<tr>
<td>1.5.1</td>
<td>32</td>
</tr>
<tr>
<td>1.5.2</td>
<td>33</td>
</tr>
<tr>
<td>1.5.3</td>
<td>34</td>
</tr>
<tr>
<td>1.5.4</td>
<td>35</td>
</tr>
<tr>
<td>1.5.5</td>
<td>36</td>
</tr>
<tr>
<td>1.6</td>
<td>36</td>
</tr>
<tr>
<td>2.</td>
<td>38</td>
</tr>
<tr>
<td>2.1</td>
<td>38</td>
</tr>
<tr>
<td>2.2</td>
<td>39</td>
</tr>
<tr>
<td>2.3</td>
<td>41</td>
</tr>
<tr>
<td>2.4</td>
<td>41</td>
</tr>
<tr>
<td>2.4.1</td>
<td>42</td>
</tr>
<tr>
<td>2.4.2</td>
<td>44</td>
</tr>
<tr>
<td>2.4.3</td>
<td>46</td>
</tr>
<tr>
<td>2.5</td>
<td>47</td>
</tr>
<tr>
<td>3.</td>
<td>49</td>
</tr>
</tbody>
</table>

2. PROBLEM STATEMENT

2.1 Research Question

2.2 Hypothesis

2.3 Reflection

2.4 Testing

2.4.1 Neural Network Modeling

2.4.2 Data Collection

2.4.3 Evaluation

2.5 Design of Experiment

3. METHODOLOGY
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Updates to Existing Hardware</td>
<td>49</td>
</tr>
<tr>
<td>3.2 Software</td>
<td>52</td>
</tr>
<tr>
<td>3.3 Data Collection</td>
<td>56</td>
</tr>
<tr>
<td>3.4 Neural Network Development</td>
<td>57</td>
</tr>
<tr>
<td>3.5 Generation of Arbitrary Output Vectors</td>
<td>61</td>
</tr>
<tr>
<td>3.6 Operational Scenario: With User Calibration</td>
<td>63</td>
</tr>
<tr>
<td>3.7 Operational Scenario: Without User Calibration</td>
<td>64</td>
</tr>
<tr>
<td>3.8 Additional Application: Wheeled Ground Vehicle</td>
<td>66</td>
</tr>
<tr>
<td>4. RESULTS</td>
<td>68</td>
</tr>
<tr>
<td>4.1 Operational Scenario: With User Calibration</td>
<td>68</td>
</tr>
<tr>
<td>4.2 Operational Scenario: Without User Calibration</td>
<td>69</td>
</tr>
<tr>
<td>4.3 Multi-Subject Scenario Inference Tests</td>
<td>77</td>
</tr>
<tr>
<td>4.4 Additional Application: Wheeled Ground Vehicle</td>
<td>78</td>
</tr>
<tr>
<td>5. CONCLUSIONS</td>
<td>79</td>
</tr>
<tr>
<td>5.1 Conclusion</td>
<td>79</td>
</tr>
<tr>
<td>5.2 Future Additions</td>
<td>80</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>89</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1.</td>
<td>Network Features by Accuracy Tested on a Random Sample of Input Data</td>
<td>59</td>
</tr>
<tr>
<td>4-1.</td>
<td>Control with Calibration</td>
<td>68</td>
</tr>
<tr>
<td>4-2.</td>
<td>One Training Subject</td>
<td>70</td>
</tr>
<tr>
<td>4-3.</td>
<td>Group Averages</td>
<td>71</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1. A technical drawing of Yagin’s 1890 passive lower-limb exoskeleton concept</td>
<td>2</td>
</tr>
<tr>
<td>1-2. Examples of exoskeletons that have been developed</td>
<td>4</td>
</tr>
<tr>
<td>1-3. (Left) An example diagram of an exoskeleton leg actuated by hydraulic cylinders. (Right) An actualized example of a pneumatic artificial muscle in a lower leg assistive exoskeleton</td>
<td>11</td>
</tr>
<tr>
<td>1-4. (Left) A cross section diagram of a DC motor designed for use in powered exoskeletons. (Right) An external view of the same motor design</td>
<td>12</td>
</tr>
<tr>
<td>1-5. A simple representation of the basic steps in preparing an EMG muscle activity signal for use in an exoskeleton control system</td>
<td>15</td>
</tr>
<tr>
<td>1-6. A sample controller design for an action-based force control exoskeleton</td>
<td>16</td>
</tr>
<tr>
<td>1-7. (Left) A depiction of the powered assistive wrist device reported in. (Right) An engineering drawing of the same assistive wrist device</td>
<td>17</td>
</tr>
<tr>
<td>1-8. A mapping of the locations EMG electrodes were applied to pick up intended motions for SUEFUL-7</td>
<td>18</td>
</tr>
<tr>
<td>1-9. A simple block diagram of proposed solution</td>
<td>23</td>
</tr>
<tr>
<td>1-10. (Left) A CAD rendering of the Smart Handle device. (Right) The prototype Smart Handle device mounted to extrusion for analysis</td>
<td>23</td>
</tr>
<tr>
<td>1-11. The six types of components comprising the Smart Handle</td>
<td>24</td>
</tr>
<tr>
<td>1-12. The physical construction of the Smart Handle</td>
<td>25</td>
</tr>
</tbody>
</table>
1-13. Two examples of the Wheatstone bridges that the strain gauges can be wired into.

1-14. An ANSYS simulation that shows the stresses on the mount when the user is making an
upward or downward motion while gripping the Smart Handle.

2-1. A visualization of gradient descent in performance learning neural network training algorithms.

3-1. A labeled photo of the full experimental hardware setup.

3-2. Electrical hardware diagram of the Smart Handle system.

3-3. A screenshot of the primary GUI used for the experiment.

3-4. A labeled screenshot of the different functions of the GUI used for the experiment.

3-5. Typical effective ANN topology.

3-6. A plot of ANN accuracy versus number of neurons in hidden layer.

3-7. A plot of network accuracy versus number of hidden layers.

3-8. A diagram of the training classes for a two-DOF wrist exoskeleton.

3-9. Four views of a small radio-controlled wheeled robot used for application tests.

3-10. A diagram of the training classes and output of the Smart Handle system in two DOF in
Cartesian coordinates.

4-1. A plot of classification accuracy versus number of subjects in the training set.

4-2. A diagram of the training classes for six-DOF exoskeleton arm applications.

4-3. Raw data for each of the different six translation classes plus “grip” (Grp) and “rest” (R).

4-4. Raw data for each of the six rotation classes plus “grip” (Grp) and “rest” (R).
4-5. An example of the network test output including all twelve translation and rotation classes plus “grip” and “rest”....................................................................................................................76

4-6. The two inference test scenarios for multi-subject, no calibration operation, with and without feedback to user................................................................................................................................78

(Middle/Right) Physical prototype........................................................................................................81

5-2. Two views of a CAD model for a “next-generation” 2-DOF actuated wrist exoskeleton...........82

5-3. Five interactions with the physical prototype of the “next-generation” 2-DOF wrist exoskeleton...83

5-4. Five more images of a different perspective of a user interacting with the “next-generation” 2-DOF wrist exoskeleton.................................................................84

5-5. Two views of a CAD model of an upper extremity exoskeleton with a 2-DOF actuated shoulder and a 1-DOF actuated elbow.................................................................85

5-6. Four views of the physical prototype of the upper extremity exoskeleton intended to test the Smart Handle technology.................................................................86

5-7. Six examples of the range of motion of the upper extremity exoskeleton prototype.............87
CHAPTER 1
INTRODUCTION

1.1 Background

Dating into prehistory, human beings developed the concept of augmenting their physical and mental abilities, as evidenced by ancient cave paintings depicting humans with the attributes of animals. This concept was widespread and many early pieces of writing detail examples of humans or human-like entities with unnatural, enhanced, or altered capabilities. These imply a deep-seeded instinct to investigate tool-use for the purpose of extending what an individual can accomplish. Over the centuries, tools were developed that would actualize many instances of that concept.

Ancient hunters discovered the atlatl, a device that lengthened the human arm and allowed spears to be thrown over longer distances. In the Middle Ages, glass blowers noticed that spheres of glass, when placed on a parchment, increased the apparent size of the words underneath. By the time of the Scientific Revolution, these spheres had matured into lenses that greatly magnified a human’s ability to perceive distant or very small objects. Humans were augmenting their capabilities with technology, and soon, this concept would be applied to human strength and endurance, and the idea of exoskeletons would come into being.
1.1.1 History

In 1890 a Russian engineer living in the United States named Nicholas Yagin invented a suit fashioned from longbow springs that could be strapped to the wearer’s lower back and feet and would run down the sides of the legs. The idea was that the energy stored in these springs while the wearer lowered toward the ground would then be released upon rising, thus allowing them to jump higher and run faster than normal [1]. The effectiveness of this device was questionable, but the idea was revolutionary. For the first time an attempt was made to construct a wearable device that would increase the physical strength of the wearer. Yagin’s suit is now recognized as the world’s first exoskeleton design.

Fig. 1-1. A technical drawing of Yagin’s 1890 passive lower-limb exoskeleton concept. Credit [1].
Although Yagin’s efforts took place in the nineteenth century, the development of practical exoskeletons would depend on twentieth and twenty-first century technologies. The first modern exoskeleton to advance beyond the concept stage was developed by General Electric (GE) from 1965 to 1971. They called it “Hardiman”, and it was powered by a hydraulics system which aimed for a load capacity of 680 kg. The design was not successful, however. It lacked a suitable control system and never achieved practical lower limb strength enhancement [2].

After Hardiman, exoskeleton development stagnated, awaiting advancements in electric drive technologies and faster more capable electronics. By the beginning of the twenty-first century, inspired by enhancing the battlefield performance of soldiers, the U.S. Defense Advanced Research Projects Agency (DARPA) took up the reigns of exoskeleton design with the Exoskeleton for Human Performance Augmentation project (EHPA). DARPA began funding a myriad of exoskeleton projects including BLEEX (Berkeley Lower Extremity Exoskeleton) by The Robot and Human Engineering Laboratory of the University of California Berkeley, WEAR (Wearable Energetically Autonomous Robot) by SARCOS, and a design by The Media Laboratory’s Biomechatronics Group at MIT, among others [1][3].

Following the numerous contributions made by the DARPA-funded projects, additional groups entered exoskeleton development. Lockheed Martin, led by Professor Homayoon Kazerooni of the BLEEX team, created HULC (Human Universal Load Carrier). Raytheon purchased the work done previously by SARCOS and built a refined, next-generation model of their WEAR exoskeleton. These designs, due to their creation through DARPA’s EHPA program, had a military or industrial motivation. These are only two uses, and there but many other applications for exoskeleton devices.
At around the same time that DARPA revived interest in exoskeleton research, several Japanese groups began developing projects aimed at assisting disabled individuals in recovering lost or diminished motor functions. In 2004, Tsukuba University began development on the HAL (Hybrid Assistive Leg) line of exoskeleton suits. HAL exoskeletons offered very desirable performance specifications, and a model from this line became the first commercially available exoskeleton in history [1][4]. Many of the DARPA suits made use of hydraulic power, much like Hardiman, but the HAL suits adopted DC motors and harmonic gear reducers in each of the joints. Additionally, HAL used a network of electromyography sensors (EMG) to measure electrical activity in the muscles of the wearer and determine intended movements. The Kanagawa Institute of Science and Technology, Tokyo University, and Honda have also produced exoskeletons, aimed primarily at assisting the disabled [1][5].

Fig. 1-2. Examples of exoskeletons that have been developed. From right: Hardiman, BLEEX, HAL. Credit [1].
1.1.2 Uses

These works, and many others, paved the way for groups around the world working on all aspects of exoskeleton technology today. Although groups in the United States, Russia, and China are still working to develop suits for combat roles, much current development focuses on medical applications. In addition to restoring mobility or strength to the elderly and providing new capabilities to those born with disabilities, exoskeletons can play an important part in rehabilitation. Stroke victims, people with ailments which cause long-term immobility such as comas or spinal injuries, or even astronauts who spend months in microgravity environments can use exoskeleton devices to assist in their exercise routines.

Beyond military and medical uses, exoskeletons could play a role in the future of industry. It has been shown that strenuous repeated motions, exactly like those performed in many manufacturing, construction, and agricultural jobs, cause debilitating conditions later in life such as carpal tunnel syndrome, Raynaud’s Syndrome, tendinitis, thoracic outlet syndrome, epicondylitis, degenerative disc disease, DeQuervain’s Disease, among many others [6]. People will continue to suffer through these conditions if another method for performing complex industrial tasks is not found. Exoskeletons, which combine the stamina of robots with the intelligence of human beings, are a potential solution. Additionally, exoskeletons could positively contribute to safety in an industrial environment. Workers frequently tasked with taking heavy loads, loads which they could lift in improper ways, or potentially fall on them or otherwise harm them if contact is made in undesired ways, could employ exoskeletons which would help shoulder
those loads and potentially decrease the probability of accidents, or lessen the impact of accidents should they occur.

1.1.3 Future

The future of exoskeletons is not certain. Development has been slow, and current models suffer from several critical shortcomings. High sensor cost, cumbersome mechanisms, lengthy unit calibration, unintended human-machine conflict, impractical donning and doffing, and insufficient battery life must all be addressed before exoskeletons can become common. However, recent advances in artificial intelligence, battery technologies, and wearable biomechanical sensors could provide solutions to some of these issues and help step exoskeletons closer to practicality. Neural networks allow for circumventing some of the long-standing signal processing issues fought by roboticists, who up until recently have largely relied on combinational logic to interpret sensor inputs. Lithium metal and lithium sulfur batteries, which are more power-dense than current technologies, could help exoskeletons function all day for the disabled, or through an entire shift for industrial workers, without the need to carry spares or chargers [7]. Electric busses and trucks, long range electric cars and drones, and even electric airplanes have been touted as possible with these new batteries. Adapting them to exoskeletons for all-day uses would be simple task. Very small wearable sensor nets that continuously monitor gait, as well as trunk and upper extremity positions, are providing data that could help designers create more comfortable and fluid exoskeletons and reduce the risk of human-machine conflict.
1.2 Critical Dimensions of Exoskeletons

An exoskeleton is a highly complex confluence of many different technologies and methodologies usually attributed to separate disciplines. What is most readily noticed when looking at an exoskeleton is its mechanical design, or how the hardware is arranged around the user. This includes joints, linkages, motors, hydraulics and other things typically associated with mechanical engineering. In addition to this basic framework, every exoskeleton needs a power source to provide energy to the actuators and electronics. These actuators must have a safe and reliable control system which is itself dependent upon some form of sensor network for picking up the user’s intended motions. These technologies are often associated with electrical engineering. Further, complex signal processing that takes the raw data from the sensors and creates a set of instructions that can operate the actuators is needed. These signals are often used as reference values for digital control systems that work alongside trajectory planning algorithms, equations of motion, and inverse kinematic models. All this is done in software, which is usually associated with computer science. Most importantly, these devices must support the user and do no harm under any circumstances. A sound working knowledge of anatomy, physiology, and injury is required before any hardware design can begin. These topics are most often associated with biology and medicine. Each of these varied technologies can be viewed as belonging to a separate critical aspect, or critical dimension, of an exoskeleton. It is the task of researchers belonging to these different fields, or multidisciplinary exoskeleton specialists, to investigate incremental improvements in each, step-by-step, as well as high-level conceptual revolutions in exoskeleton technology in general.
1.2.1 Hardware Arrangement

In ordinary humanoid robotics, the goal is to replicate the system of joints and linkages displayed in the human body. This is desirable because the entire human infrastructure of the planet was designed around easily accommodating this form. Designing humanoid robots presents a set of challenges in and of itself, but compared to exoskeleton hardware design, it is notably simplistic [1]. Designing a fully articulated robotic structure that operates safely around a human user is a highly complex task. Each degree of freedom displayed by the human anatomy the exoskeleton encapsulates must be accounted for and present in the design. In fact, it is often the case that extras are needed to accommodate the human inside, which increases the complexity of the equations of motion and kinematic model.

For example, the human shoulder is similar to a simple ball joint. However, a ball joint cannot be used for an exoskeleton shoulder, because it would have to be mounted alongside the human, and thus the axes of rotation for the human and the robot would not coincide. This would make it impossible for the exoskeleton elbow to remain in the same relative position to the human’s elbow. The shoulder must be approximated by three revolute joints whose axes of rotation intersect at the center of rotation of the human shoulder. Practically, this cannot be done because some linkages would impact the human over a portion of their range of motion. Prismatic joints in the back are often used to bring the linkages in question away from the human. This is why in-depth knowledge of physiology and mechanical engineering are required.
Additionally, considering the range of motion that the human occupant is capable of is of paramount importance from a safety perspective. A humanoid robot could safely have any of its joints’ range of motion extended to enhance its capabilities. However, this would be dangerous if implemented in an exoskeleton. Exceeding the range of motion of the wearer could cause serious injury to the soft tissues of the body, and perhaps break bones. Therefore, care and deliberation must be taken by the engineers responsible for creating the hardware for exoskeletons. The idea is to make life safer for the user, not to introduce danger.

1.2.2 Power Source

As of the time of writing, it could be said that exoskeletons largely fall into two broad categories: grounded and mobile. Grounded exoskeletons are those often found in research laboratories that focus on one part of the body. These are scientific tools and, due to being fixed to one spot, can simply be energized from wired mains power. No special power considerations are needed for these designs [8].

Mobile exoskeletons, however, represent the general vision of what an exoskeleton is. A mobile device that a person can wear, and with which one can move about freely. This requires an onboard power source and, given the nature of the actuators utilized by exoskeletons whichever type they may be, this source must be of very high energy density in order to get a full day’s use. The problem is energy density is the primary failing of chemical batteries. Electric vehicles have been so long coming due to this fact. Liquid fuels are often cited as being two or more orders of magnitude more energy dense than lithium-ion batteries [9]. This means you would need up to
100 volume units of batteries to get the same energy supply as would be present in one volume unit of gasoline. This is precisely why manned quadcopters, electric planes, long range EVs, and all-day exoskeletons are not yet common.

Right now, with some notable exceptions such as the exoskeleton “Body Extender” which has a standalone operation time of eight hours, many of the mobile suits found in literature have about three hours of operation per charge [1][5]. This could work for a therapeutic exoskeleton that only needs to be used for a short period, but for military, industrial, or use for disabled mobility assistance, some new development is going to be needed. As mentioned in section 1.1.3, new battery technologies like lithium metal or lithium sulfur models could overcome these shortcomings, but for now power sources remain an open issue.

1.2.3 Actuators

An exoskeleton requires components that will allow it to move, and, as explained in section 1.2.1, those movements must fall within a certain range [1][10]. In addition to falling within a certain range, movements must be precise and delicate to safely operate around the human inside. Therefore, they require some type of actuator and some form of actuator control. Actuators could be anything from hydraulics, to electric, to pneumatic.

Hydraulics makes use of the incompressibility of liquids to move the exoskeleton’s joints. The liquids are pumped into tubes, referred to as a barrel, which holds a piston and a piston rod. The liquid presses against the piston and, since the liquids cannot compress, they force the piston
out of the barrel and actuate the linkage attached to the rod [11][12]. Many early exoskeletons utilized this form actuator but it is less commonly seen today.

Pneumatics is often very similar to hydraulics, only using gasses instead of liquids, and thus the dynamics of movement are changed. Unlike the liquids used in hydraulics, the gasses used in pneumatics are compressible. This compressibility causes a delay in movement in pneumatics compared to hydraulics [13][14]. Pneumatics can also refer to what are called pneumatic artificial muscles (PAMs), which contract or extend when a bladder is filled with pressurized air. In this way they attempt to mimic the functionality of biological muscles and present distinct advantages in exoskeleton design [15]. For instance, they are often comprised of soft materials that make the exoskeleton relatively safe to use around humans. Further, PAMs can

Fig. 1-3. (Left) An example diagram of an exoskeleton leg actuated by hydraulic cylinders. Credit: “Precision Interaction Force Control of an Underactuated Hydraulic Stance Leg Exoskeleton Considering the Constraint from the Wearer”, Shan Chen, et. al. (Right) An actualized example of a pneumatic artificial muscle in a lower leg assistive exoskeleton. Credit: “Altering gait variability with an ankle exoskeleton”, Prokopios Antonellis, et. al.
be very narrow when extended. This is desirable for use in permanent assistive devices. A 2012 Ph.D. thesis titled “Intelligent Assistive Knee Orthotic Device Utilizing Pneumatic Artificial Muscles” showed that PAMs could be placed around thigh and shin braces with a minimal profile, thus being ideal for commercial products.

The most common form of exoskeleton actuation is an electric drive. Systems exist for using AC motors, servos, and brushed (DC) and brushless DC (BLDC) motors, although BLDC motors are most common [16][17]. DC motors can be easily paired with buck/boost converters for speed control, and display four quadrant operation [18]. BLDC motors are commonly controlled with three-channel half-bridge controllers or full-bridge controllers. Power can be transmitted in either gear systems or cable systems, although each have their strengths and weaknesses. Gearboxes come in many varieties, most notably planetary gearsets, cycloidal drives, or harmonic drives.

![Fig. 1-4. (Left) A cross section diagram of a DC motor designed for use in powered exoskeletons. (Right) An external view of the same motor design.](image)
Controlling an exoskeleton is extremely difficult. There are many ways to do it, but few seem ideal. In general, there are two broad categories of exoskeleton control: intention-based control systems and action-based control systems. As implied by the names, intention-based control and action-based control are differentiated by how the human interacts with the wearable. If a human is required to perform an action to which the suit will respond, it is called an action-based system. If the system is capable of picking up on signals from the user that anticipate movement, this is an intention-based system. The most common examples of each would be EMG control and force control for intention and action-based systems respectively.

EMG stands for electromyography, and it works by sensing electrical signals in the user’s muscles that correspond to limb movements. This method has the main benefit of being able to cater to individuals who cannot physically move their own body, and it therefore holds promise for products aimed at aiding the severely disabled [10]. However, EMGs come with quite a few drawbacks. The sensors are costly, the signal processing required is of notable complexity, and the sensors themselves often require precise placement and laboratory calibration in order to function, although this is beginning to change with two caveats described in sections 1.3.3 and 1.3.4.

Force control makes use of force sensors placed throughout the interior of the exoskeleton, where the exoskeleton contacts the wearer. It works by using a control system that tries to bring the force against the sensors to zero [1]. For example, if the user of an upper limb exoskeleton is
standing with their arms to their side and wants to raise their right arm forward, they would begin by simply doing so. This would bring their arm into contact with the force sensors inside the suit, and thus, those sensors would begin registering a non-zero force. A PID controller could be used to tell the actuators to begin movement on a specified trajectory that would move the exoskeleton arm away from the direction in which force is being applied. As the robot’s arm begins to move with the human’s, the force, if applied constantly, would become less due to the relative movement. The control system will move the actuator only until it has reached a state of zero applied force once more, and then the suit and wearer will again be in equilibrium.

Force control is a very elegant technology, but still requires complex sensor networks distributed throughout the exoskeleton. The preceding description included sensors on the interior of the exoskeleton, but they are often placed on the outside as well. Such systems try to transfer force from interior sensors to exterior ones. This complexity is not ideal and since movement on the part of the user is required, therapeutic models would be limited to patients capable of moving independently [10].

1.2.5 Signal Processing

Whether using intention-based or action-based control systems, complex signal processing techniques are required. In EMG intention-based systems, processing involves several steps. First, statistical detection theory is used to design a detector to differentiate the presence of muscle activity from background conditions. Probabilistic information regarding the sample distributions of these conditions is needed. Once a signal is detected it must be full-wave rectified to allow for
calculations of mean and max values, as well as allowing for more intuitive observation. Next, due to the random nature of EMG signals the data must be smoothed to create what is called a linear envelope. The moving average and root-mean-square algorithms are often used for this. Additionally, due to the microvolt nature of the signal, outside conditions can cause wild variations in readings even on the same muscles of the same subject. Amplitude normalization is needed to resolve this [19]. It is important to note, the preceding steps are needed simply to prepare the signals for use in a logic scheme, which will itself be highly complex.

![Fig. 1-5](image)

**Fig. 1-5.** A simple representation of the basic steps in preparing an EMG muscle activity signal for use in an exoskeleton control system. Credit [20].

Force control systems also require complex processing. The signals picked up by the force sensors are not as labor intensive to process as with EMG, but here, feedback control techniques are used to allow the system to track the desired force on the sensors [1]. One such technique is called inner-position loop control, and an example diagram is presented below:
1.3 Void Area: Intended Motion

So far in this work, different aspects of exoskeletons have been described to set the backdrop for the research presented below. To show the necessity for contributions in exoskeleton research, a single critical dimension will be explored in terms of its current developmental shortcomings, followed by a short description of how one or more of those shortcomings might be overcome. Realizing that the issues of cost, complexity, and intensive calibration could all be impacted by a new intended motion determination system, it was chosen as the focus of this work.

1.3.1 EMG Problems

EMG sensors are very commonly used in exoskeleton designs. They work due to the electrical nature of the human nervous and muscular systems. Neural activity in the brain transmits
electrical signals through the spinal cord and to a muscle via motor neurons. This causes a release of calcium ions in the muscle resulting in a process called depolarization, which changes its electromechanical gradient. These changes can be registered from electrodes affixed to the user’s skin at separate points along the electromechanical gradient. If those electrodes are placed properly, a repeatable pattern will be present in the signal they pick up that accurately represents the expansion and contraction of a human muscle [19]. This is exactly the sort of thing that is needed for a robot to mirror a human’s movements.

That said, EMG sensors are not without their problems. The signals they produce are notoriously noisy, and complex signal processing techniques are required to clean them up [20].
And once the signals are cleaned, they are often used in complex sets of fuzzy logic rules. For a three DOF (degree of freedom) wrist device capable of pronation/supination, radial/ulnar deviation, and wrist flexion/extension, six separate types of EMG signals had to be defined and the movements were attained with a set of 46 compound if/then rules [10]. It is not hard to imagine how quickly such a system will explode into immense complexity when including the elbow and shoulder, let alone the rest of the body.

![Fig. 1-8. A mapping of the locations EMG electrodes were applied to pick up intended motions for SUEFUL-7. Credit [21].](image)

In addition to the complexity displayed by the wrist motion assist exoskeleton SUEFUL-7, a seven DOF upper-limb exoskeleton was designed with EMG intention-based control [21] and further illustrates the complexity of EMG systems. As the name would suggest, it is a fully
articulated exoskeleton arm. To control the arm with EMG sensors, 16 separate channels had to be established. Each of these channels corresponds to a muscle that will have its activity tracked and utilized for exoskeleton actuation. Muscle activity was sampled at 2kHz and an RMS calculation was done to clean the signals. This worked but, if possible, why not avoid it?

1.3.2 Force Sensor Problems

Overall, force control is an effective and proven way to generate trajectories for an exoskeleton. From the review of the available literature that was done in preparation for this work, it seemed like force control systems did not have that many downsides. But, as mentioned earlier, exoskeleton models that employ force sensors must install many such sensors throughout the design for it to function. The more sensors there are, the more processing there must be, and the more computing resources are used. Complex software must then be written to handle the data from all these sensors and to implement the control algorithms. This presents the classic problem that having many components increases the opportunities for failures.

Ideally, there would be a solution that would allow for a centralized cluster of a few sensors so as to avoid these problems. Still, it would seem overall, even with this issue, the argument could be made that force control is superior to EMG control. Force control cannot help individuals who lack mobility entirely however, which is probably why EMG systems remain so popular even considering their drawbacks.
1.3.3 Cost Issues

As it stands, powered exoskeletons are very expensive. In order to make powered exoskeletons commercially viable, they must be produced and sold for a price that is accessible to people. First, EMG sensors can vary widely in price and quality. It is possible to buy a set online for less than one-hundred US dollars. However, the chances that these sensors would work reliably in an exoskeleton are small. On the other end of the spectrum, laboratory-grade EMG sensors could cost as much as several thousand dollars for a set [22].

To reliably equip an exoskeleton with EMG sensors, especially in potentially hazardous conditions as are often found in military or industrial settings, high quality sensors are required. The same goes for delicate medical situations. This drives the overall cost of an EMG-based unit up and draws it further away from the realm of commercial actualization. A similar argument could be made for force sensor-based models, however, to a lesser extent. The hardware used in these models is not as expensive as EMGs, but many are still needed.

It should be noted however, that a group of Italian and German scientists from the University of Trento and the German Aerospace Research Center respectively, has made interesting progress in developing high quality, low-cost EMG support electronics. They endeavored to construct a better solution for EMG-based prosthetic hand control. The result of their work was a small, easily wearable, multichannel device with a maximum cost of 150 EUR at the time of writing, or roughly 180 US dollars [22]. This could dramatically reduce the cost of an exoskeleton, but this new technology has not yet been broadly embraced.
1.3.4 Setup Issues

In section 1.3.1, it was said that EMG sensors must be placed properly for them to function well as an intended motion system for exoskeletons. This was not said lightly. By many accounts exoskeletons that make use of EMG sensors require expert technicians to calibrate electronics and place the electrodes [23]. In certain medical settings, for applications such as rehabilitation for example, this might not be much of an issue. In those situations, the user is often already in a medical facility in the presence of experts. However, for other medical applications such as everyday mobility or strength enhancement, complex setup is not practical. If the exoskeleton is intended to be for someone’s everyday use, such as a pair of glasses or a wheelchair is today, it must be something that the user can easily don and doff by themselves and at their convenience.

Further, if the use is industrial or military in nature, the issue of simple setup is even more critical. Military personnel absolutely must be able to begin exoskeleton operation at a moment’s notice. When in a combat setting, it is never known when contact with an enemy might be made, and so a suit that cannot be utilized without warning serves limited purpose. Lastly, no industry will be quick to adopt new equipment if their employees require an hour or more of entirely unproductive time on the clock to simply prepare for work. The increase in productivity the equipment affords would need to offset this loss, and more, to be profitable. Ideally, any powered industrial exoskeleton would be no more difficult to get into than a forklift, or perhaps a safety harness. So, the issue of setup alone would seem to invalidate EMG models for consideration for this type of deployment. Coupled with the cost issue mentioned earlier, it is clear that for certain applications EMG-based exoskeletons stand a low probability of broad acceptance. It should be
noted that it does not appear that exoskeletons that use force-control suffer from setup issues to this extent.

Interestingly, another Italian team set out to combat the issue of setup directly. To avoid the usual precise time-consuming setup process, they designed a simpler one that could be easily fitted to the wearer. There was a marked decrease in accuracy in picking up the intentions of the user. However, they believed that the natural ability of the human central nervous system to adapt to external disturbances would be enough to compensate for a lower accuracy. Their results showed that it is possible to employ a system like this and still noticeably reduce the effort exerted by the user [23].

1.3.5 Proposed Solutions

Using the outline drawn by the issues regarding current intended motion methodologies stated above, one can begin constructing a concept for a technology to overcome them. For instance, if EMG sensors are complex, expensive, and unwieldy, avoid them. If EMG sensors and force sensors distributed throughout a design create a cumbersome situation, begin thinking of ways to make use of a consolidated sensor system that can utilize cheaper hardware that is easy to maintain. If it is not feasible to implement any system that requires an involved setup routine, focus on general purpose solutions that lend themselves to being used “out of the box” so to speak.
The team at the Biodynamics Laboratory at Northern Illinois University have developed a device known as the “Smart Handle” that seeks to fill the role described above, for use in upper extremity exoskeletons [24]. It is a handle whose purpose is to detect minute deviations in force applied by the hand of a user and use that data to determine precisely what translations or rotations
in 3D space the hand is attempting to make. The movements of the user’s hand can be processed and used to determine what type of movements the exoskeleton is supposed to make.

The Smart Handle is comprised of six classes of components as shown below. Two mount bracket extensions are attached to a mount bracket which will attach to an exoskeleton or static mount. Fit between the mount bracket extensions is an extension column around which the handle exterior is constructed. At the top and bottom of the extension column, where it meets the mount bracket extensions, are fixed two hexagonal base components. These will act as a foundation for cantilever columns. Each side of the hexagonal bases will have one cantilever attached to it, making twelve in all, six from the top and six from the bottom. A 3D printed plastic grip is added to the cantilever mounts for user comfort. To get a more complete picture of the Smart Handle device, some images showing how it is constructed are provided.

Fig. 1-11. The six types of components comprising the Smart Handle.
The Smart Handle uses strain gauges mounted inside the handle between the cantilever columns and the extension column in order to pick up the slight actions of the user gripping it. Strain gauges work via piezoelectric effects, or effects that relate mechanical deformation with electrical properties. The electrical resistance of the gauge material changes based on its mechanical deformation. Since the cantilever columns are fixed at one end and free at the other, they will be able to bow very slightly as the user applies pressure to them. This, in turn, deforms the strain gauge, which alters its electrical resistance. If the strain gauges are part of an electric circuit known as a Wheatstone bridge, an electrical signal can then be created that attenuates predictably along with the deformation of the cantilevers.
In addition to the strain gauges mounted on the cantilever columns, there are eight more strain gauges on the Smart Handle, making twenty in total. These are located on the mounting bracket components. The purpose of these gauges is to pick up user movements that affect more than just the cantilevers. A translation movement upwards or downwards can be further analyzed by measuring the stresses the mounting bracket is experiencing. Two pairs of differentially wired strain gauges are mounted at these points for this purpose. The team that created the Smart Handle created this ANSYS model of the mounting bracket to investigate its response to users making these sorts of movements. A rotation movement in the yaw dimension can be analyzed by measuring stresses felt by the thin tabs on which the handle basses are mounted. Two more pairs of differentially wired strain gauges are placed on these tabs.
Fig. 1-14. This is an ANSYS simulation that shows the stresses on the mount when the user is making an upward or downward motion while gripping the Smart Handle.

What this does is offer exactly two of the features described as the possible solutions to the intended motion void area. A centralized, easy to maintain cluster of sensors that is relatively cheap and relatively simple. This, however, is only part of the solution. It would still require complex signal processing, possibly using combinational logic, statistical detection theory, or perhaps fuzzy logic. What is needed is another elegant solution that eliminates these concerns and offers the ideal fix to exoskeleton control needs. Artificial intelligence techniques such as neural networks can do just that. Neural networks are parallel computing processors that do not need to be told, in detail, every single action they must take [25]. The advantage of neural networks is that, upon being exposed to the desired behavior, they modify their own structure and learn how to obtain the desired result by themselves. They store information in what are called “synaptic weights” that connect multiple neurons, or computational centers [25]. Neural networks have been shown to be able to perform immensely complex tasks using this technique. Of course, there are hundreds of different types of neural networks architectures, and many ways these networks can
be trained. A large focus of this work was to discern which combinations of these things are ideal in upper extremity exoskeleton control solutions.

In summary, the solution proposed by this work to the problem of upper-extremity exoskeleton control systems in affordable and easy-to-use exoskeletons is as follows. A Smart Handle device comprised of a centralized cluster of cost-effective sensors and driven by an artificial intelligence (AI) system that can be taught how to behave thus eliminating high-cost, distributed sensor networks, complex signal processing, and laborious supervised setup and calibration.

1.4 Potential Outcomes

As mentioned in the previous section, the Smart Handle hardware existed before the start of this work. It was produced by a Senior Design group working with the Biodynamics Laboratory in the 2018/2019 academic year. What the team wanted to see with the conclusion of this work can be seen as four-fold. First, a suitable AI-based control system needed to be selected. Second, it was desired that workable accuracy be achieved in multiple operation scenarios, including operation with end-user calibration and operation without end-user calibration. Third, a technique for minimizing the amount of data required for AI training while maintaining continuous three DOF translation vector outputs and three DOF orientation vector outputs. Fourth, expanding the test hardware to a wrist prototype device capable of at least the pronation/supination and radial/ulnar deviation degrees of freedom that incorporates the Smart Handle as its control mechanism.
1.4.1 AI Models

The first of the desired outcomes were the AI models that needed to be developed to satisfy the performance goals in the different operation scenarios. There are many forms such models could take. Since what was needed was basically a classifier, many frameworks could have been employed. These include K-nearest-neighbor classifier (KNN), probabilistic classifiers such as Naive Bayesian classifiers, support vector machines, a wide variety of neural networks, and many more. For this use, KNNs would almost certainly be too simplistic. Probabilistic classifiers would require a deep knowledge of the possible signals and a lot of specialized code. For this purpose, something general and light on pre-existing knowledge of the behavior of the inputs would be preferable. This is exactly what neural networks can provide, making them the ideal selection.

Many organizations offer neural network modeling software that could be used for this purpose, MATLAB, Tensorflow, and PyTorch feature prominently among them. These are high-powered professional and academic software libraries that allow fast, easy, and optimized modeling with little effort. Each have been cited in numerous publications and have performed very impressively [26].

Although they offer a large library of learning algorithms with which to train a network, their options are not exhaustive. MATLAB cannot be implemented in products, and the others lack in dynamic predictive networks support, which left some possibility for customized software solutions.
1.4.2 Performance in Operational Scenarios

As mentioned above, there are two primary types of use that were to be investigated: operation with end-user calibration, referred to from here on as “single-subject” use, and operation without end-user calibration, referred to from here on as “multi-subject” use.

Single-subject use is the case where the Smart Handle device, more specifically the control computer, does not come pre-programmed with any information about how it is to be used. This can be done since the Smart Handle system is fully software defined and able to be tailored to many use cases. Here, calibration is required in the form of a short procedure where the user provides examples of the different inputs that they wish for the AI to associate with different outputs. The user is directly providing the training data for the AI. This being the case, it was expected that extremely high accuracy would be achieved.

Multi-subject use is the case where the Smart Handle device and accompanying control electronics is intended for use in a specific way, and therefore can be provided already trained on previously collected data. Therefore, there is no calibration procedure needed as in the single-subject case. Since in this scenario there is no specific tailoring to the end-user, accuracy is expected to be less than that of the single-subject scenario, but still high enough to be workable.

1.4.3 Minimizing Training Data

An exoskeleton control system should be capable of generating end-effector trajectory vectors for direct movement from any arbitrary point in the task-space to any other arbitrary point
in the task-space. Further, the same must be true for end-effector orientation. This presents a problem for neural networks since they are required to be provided with examples of all possible output classes. How would one design the output classes if the output must be an arbitrary vector in six DOF? The obvious and limited solution would be to cut up the task-space into discrete chunks, and then collect training data that is supposed to correspond to movement in the direction of each chunk. This solution is limited because it isn’t actually providing arbitrary output. It would also require two distinct translation phases in many cases, instead of one direct movement. Further, the amount of data collection would be huge and very time consuming for the subject providing the data in order get adequate input-space coverage. Some technique for providing truly arbitrary output on a small, discretized input set was needed.

1.4.4 Actuated Wrist Prototype

It was hoped that over the course of the research the current Smart Handle hardware could be used as the nucleus of an exoskeleton wrist proof-of-concept to showcase these new ideas. This device would be capable of replicating wrist pronation and supination, radial and ulnar deviation, and potentially wrist flexion and extension. It would mount the Smart Handle within it and allow for testing the Smart Handle in non-static situations. Before this work, the Smart Handle was only tested and used to gather AI training data when fixed. Since this is a non-realistic situation, the team planned to collect data from dynamic situations in order to ensure the feasibility of the concept.
1.5 Quantification

Having erected a framework of ideas and concepts to support the proposed outcomes of this research, it would serve well to now flesh out this framework with a set of quantified goals. It would be good to state a level of accuracy that should be attained, along with a maximum allowable setup time, these being critical to the success of the work. Additionally, it would make sense to specify a form factor by which to adhere to, so the device won’t rely on hardware with an unrealistic footprint or volume. Finally, since training neural networks often takes time, a limit to how much training data is to be used and how much time can be allotted for training might help.

1.5.1 Accuracy

Classification neural networks are usually judged by their accuracy in classifying their inputs correctly. If an exoskeleton is to be accepted for military, industrial, or medical applications it will need to be highly reliable. Any control system that produces the incorrect motions will not be acceptable. To that end, a preliminary goal of 95% classification accuracy for each degree of freedom shall be set in the single-subject use case. Since the testing and inference data should match extremely closely with the training data this was a reasonable expectation.

In the multi-subject case, it is more difficult to tell what accuracy would be acceptable. In the original Smart Handle research phase, conducted by NIU electrical engineering graduate student Christopher Wolfe, seven subjects provided data for three states of interaction with the handle. These were pronation, supination, and simply gripping. (The handle at rest was another
state, but those samples were not needed from the subjects.) With the data from one randomly selected subject as the testing data and the data from the remaining six subjects as the training data (simulating the multi-subject use case) roughly 80% - 85% classification accuracy was achieved. In the beginning of the research when planning was being done, it was known that two more classes would be added for the multi-subject data collection, those being radial and ulnar deviation. At that time, it was also thought that somewhere between ten and twenty subjects would be recruited to provide the data. It is not possible to predict exactly how these extra movement classes and subjects would affect the classification performance, but it was thought that a reasonable expectation would be between 85% and 95% accuracy.

1.5.2 Data Collection

The team discussed on several occasions how training data for the network could be gathered. The general consensus is that more human subjects will have to be recruited than in the original study. That experience showed that the number will most certainly have to be increased to be successful in classifying five movements (plus “rest”) as opposed to three (plus “rest”), and at a higher accuracy. See section 4.2 for details.

A “back-of-the-envelope” type calculation can be done to try and put a rough number on how many subjects might be needed for the same accuracy as the original study but with the extra movements. If 85% was achieved with seven subjects on four classes, and it is assumed that one class can have at least 85% accuracy with zero subjects (a safe assumption since a network with only one class can’t be wrong), then a simple line (another assumption, but not as safe) would
suggest that around eleven people would be needed to get 85% accuracy with six classes. Therefore, it was decided that at least ten individuals would be recruited in order to try and keep the same accuracy. A rough goal of ten to fifteen subjects was then set.

To create the most controlled situation possible, and the best chance of proving the AI powered handle concept is workable at all, only male subjects would be chosen at first. This would give the most uniform test data. Then, the recruits after that would be female, rounding out the sample and allowing the team to test differences in grip characteristics from males to females.

1.5.3 Training

The next logical area to consider was the training data itself. From just the seven subjects of the first run, over fifty-thousand dataset elements were gathered, each sixteen values, one representing each of the strain gauge channel values from the Smart Handle. These were compressed down into groups of five, whose average was taken. This amounted to roughly twelve-thousand training set elements. Depending on which computer, library, and training algorithm was used, training a network on this dataset, split into 85% for training and 15% for testing, took anywhere from a few seconds to an hour. Efficient algorithms run on high-performance PCs achieved the lowest times, whereas more basic algorithms run on ARM SBCs took considerably longer. This range of devices will be explained in more detail in the next section. But given this information, it was safe to make some ballpark estimates regarding dataset size and training times. The team wanted to aim to collect no more than perhaps one million training elements. Enough
to have a reasonable chance at success, but also enough to keep training time on practical devices to no more than a few hours to one night.

The primary concern for data collection was keeping the time each subject was required to interact with the handle to an absolute minimum. Not only was this needed for the comfort of the subjects and for the practical concerns regarding recruiting and guiding them, but also for keeping network simplicity and single-subject user calibration manageable. Since the original Smart Handle hardware had sixteen outputs needing analysis, that meant most scenarios would have those sixteen outputs as the inputs to the neural network. It is not often the case that there are more outputs than inputs in a neural network. There would need to be drastic variation in the characteristics of the signals in order for that to work. So, right away the number of output classes is capped. This would mean seriously restricting discretization of the trajectory vectors. Since this is not acceptable, the most obvious solution was to decide there should be one network output corresponding to each movement class plus a “rest” state when there is no interaction with the handle. Human trials will collect the five previously mentioned classes plus “rest”, and some form of postprocessing would interpolate between each of the classes for compound movements, or movements between each of the classes which would give a continuous range of arbitrary outputs.

1.5.4 Form Factor

Since any computer device that is intended to be part of an exoskeleton control system must go where the exoskeleton goes, it must be capable of conveniently mounting onto the exoskeleton. Thus, the form factor and performance specifications will be limited. As another
ballpark estimate, it was thought that the power source, computer, support hardware and any peripherals necessary should fit into an area roughly the size of a human’s lower or upper back. Something like a maximum volume of 7,500 cm$^3$ was considered a reasonable benchmark.

1.5.5 Setup

As mentioned above, a critical stumbling block exoskeletons face regarding market penetration of any meaningful depth, is setup time. No one will want an exoskeleton that takes lengthy times to don or doff. Ideally, it would take no longer to put on the proposed wrist unit than it takes to simply grasp a handle. So, the time goal for that was set at essentially zero. The same could be said for any calibration in the multi-subject case. There, it is desired that there be no calibration at all, however this may not be realistic. Or perhaps calibration could be needed only periodically and on initial use. At any rate, for the purposes of this paper, the goal for setup will be set at zero for the multi-subject training scenario, and twenty minutes or less for the one-time calibration procedure required for the single-subject training scenario.

1.6 Report Organization

What follows in this report will be divided into four chapters: Chapter Two, Problem Statement, Chapter Three, Methodology, Chapter Four, Results, and Chapter Five, Conclusions. The Problem Statement will be split into five sections: stating of the research question, hypothesis, reflection, testing, and design of experiment. Chapter Three will discuss the different aspects of
this work’s methodology, updates made to the handle hardware between the original study and this one, the software developed for this study, the final data collection procedure, neural network development, generation of arbitrary output vectors, and how different scenarios were tested. Chapter Four will discuss the results of the research performed for this work, including both operation scenarios, further applications tested, and a review of the data collected. Finally, Chapter Five will draw conclusions based on the results found in Chapter Four, and will conclude with a discussion of possible future additions to the project.
CHAPTER 2
PROBLEM STATEMENT

Now, it would serve well to begin grounding the information and solutions, put forward in Chapter One, firmly in the scientific method. The scientific method begins with an observation. One could consider sections 1.1 to 1.3.4 of Chapter One a long list of observations regarding the current general state of exoskeleton technology. The next step in the scientific method is to consider the observations that have been made, and pose a question motivated by gaps in those observations. This question is called the research question. Once the observations and a thoughtfully designed research question are well understood, one might seek to use any specialized knowledge they possess to propose an answer in the form of a hypothesis. Now, consider sections 1.3.5 to 1.5.5 of Chapter One as a selection of specialized knowledge of electrical engineering and computer science that can inform the construction of a hypothesis. With a hypothesis proposed, it must be tested. How each aspect of the hypothesis is tested must be carefully considered, and once this is done, an experiment can be designed that isolates a specific knowledge gap and attempts to fill it in. Each of these steps will be examined in turn over the following chapter.

2.1 Research Question

In order to formally state a research question, a census of all the reported information will be taken. It is known that EMG sensors are costly, complex, and are difficult to setup on an
exoskeleton user. Similarly, if force control is used, force sensors must be placed in complex arrangements and significant work is needed in their control scheme. Additionally, hardware exists in the form of the Smart Handle that can eliminate the issues present in traditional setups, if only it could be properly supported with software. Ideally, software that circumvents the problems present in EMG and force control models, namely, the control issues outlined in section 1.3.1. Finally, what implications might there be for a research question stemming from the goal of constructing a two or three DOF wrist prototype? Considering all the above, a set of research questions could be stated as follows:

1. Can a solid-state, software-defined (AI), human-control-input device be created to output arbitrary translation and rotation vectors for trajectory planning in all six spatial DOF?

2. Is variation in natural human gripping characteristics low enough for a neural network to distinguish between fist position and orientation changes in a random individual, when only trained on data from a manageable number of separate individuals?

2.2 Hypothesis

Having now stated a research question, an effort must be made to propose a solution in the form of a hypothesis. Since the research question highlights the use of machine learning as a check on complex control, attention must be given to what sort of machine learning techniques are
available, and how they are implemented practically. Given the state of the Smart Handle, the number of outputs it has and the form that information takes, a simple multilayer perceptron neural network could conceivably be successful. There are probably too few inputs to make effective use of convolutions, and that being the case, a deep network would probably end up being counterproductive [25][27][28]. Considering preliminary work done in this area, at the time this hypothesis was conceived it seemed likely the case that additional inputs on top of simply the sensor outputs from the strain gauges will be needed. Also, considering the range of learning algorithms that have already been examined and their performance in tackling the problem, it is thought that more exotic algorithms or architectures could be considered. Or, to mitigate this need, very careful thought must be used in developing a procedure for collecting data from human subjects. Taking all this together, the following hypothesis could be stated:

1. It is possible to design and gauge a piece of hardware with no moving parts and design a ML approach to create a handle-type human input device that can control translation and orientation in six DOF.

2. It is possible to account for variation in human gripping characteristics, both in the mechanical design of the device and in software, such that practical control in six DOF can be accomplished by an operator not included in any datasets used in training the ML model.
2.3 Reflection

At this time, it would be good to look back at these questions and hypotheses in terms of what was specifically proposed in this work. To be clear, the Smart Handle already exists. And the two or three DOF wrist prototype was a proposed addition to the Smart Handle that would be developed in parallel with this work, but these were not what this work sought to show. This work was concerned with the software control system that goes along with the Smart Handle and the proposed wrist device. Most hardware additions to the existing device would require a decent knowledge of mechanical engineering and is generally outside the scope of what is proposed here. This is comprised of expansions to the neural network’s set of inputs, expansions to the neural network’s set of outputs, expansions of the capabilities of the neural network to categorize composite movements, addition of online training capability to the neural network, increasing of the accuracy of the classifications made by the network, modifications to the learning algorithms used in training the network, and modifications of the procedures employed to gather data with which to train the network. Each of these areas will be expanded upon in more depth in Chapter Three, Methodology.

2.4 Testing

Once a hypothesis is formulated in response to a research question, an experiment must be designed to substantiate or invalidate the hypothesis. But before that can be done, some method of testing the different areas that contribute to the hypothesis must be devised. Any experiment
designed to test the stated hypothesis would include neural network modeling, data collection, neural network training, and evaluation.

### 2.4.1 Neural Network Modeling

Neural networks are tested based on their performance on a given dataset. Usually, once a dataset is acquired, it is split into a training set and a testing set. It is common to use 75-85% of the data for training, and the remaining 15-25% for testing [25]. If the weights of a network have not converged to a set that will perform well on the testing set by the time all training set elements are exhausted, then the amount of data collected is insufficient, or the learning algorithm is not efficient enough. Simply running the training data through the network and optimizer more times is not enough to guarantee success.

There are many techniques that can help a network along. In the numerous variations on backpropagation, keeping the inputs low, between zero and one, is good for learning efficiency because high inputs often saturate the activation functions. For instance, with sigmoid activation, input values, \( p_l \), end up in the exponent of exponential functions.

The sigmoid activation function that takes the weighted inputs:

\[
\begin{align*}
    n &= \left( \sum_{i=1}^{N} w_i p_i \right) + b \quad \text{eq. 2.1} \\
    a &= \frac{1}{1 + e^{-n}} \quad \text{eq. 2.2}
\end{align*}
\]
Note, in the above equations, $w_i$ represents synaptic weights, and $b$ represents a neuronal bias, required for shifting decision boundaries away from the origin. If the inputs are large, the input space will place the network on a point of the performance surface whose gradient is very shallow. Common gradient descent algorithms rely on a noticeable gradient to converge quickly, or else a variable learning rate would be needed. At any rate, normalizing the inputs ensures that the algorithm enjoys an advantageous starting point for training [25][29].

Fig. 2-1. A visualization of gradient descent in performance learning neural network training algorithms. The surface represents the performance surface of the network, or a plotting of the overall network error in terms of the possible inputs. The point of the training is to move the error to a lower point in the performance surface. Credit: “Gradient Descent for Linear Regression”, Shreedhar Vellayaraj.

Another commonly used technique to encourage network convergence is some form of weight initialization. The network weights are usually randomly assigned values between zero and one for the same reasons the inputs are normalized. However, processes like the Nguyen-
Widrow weight initialization have been proven to help convergence by concentrating the input layer weight distribution to values determined by the size and connectedness of the network [29].

Nguyen-Widrow weight initialization:

\[ \beta = 0.7h^{\frac{1}{1}} \]

\[ n = \sqrt{\sum_{i=0}^{i<w_{max}} w_i^2} \]

\[ w_{t+1} = \frac{\beta w_t}{n} \]

In the above, \( h \) is the number of neurons in the first hidden layer, \( i \) is the number of inputs, and, again, \( w \) represents the weights. This process applies to the input weights only. Whether this was shown to be needed will be covered in Chapter Four: Results.

More sophisticated algorithms that do not rely on backpropagation are also available easily with neural networks software libraries provided by Tensorflow and PyTorch. These libraries were invaluable when modeling networks for this research.

### 2.4.2 Data Collection

The way the data was collected to form the datasets used in training and testing the neural networks played a huge roll in the success or failure of the experiments. Due to the fact that the current mathematical models for neural networks do not allow them to extrapolate, any training
data that was collected was confirmed to accurately represent the decision boundaries of the input space. Neural networks are capable of interpolation, so if a healthy portion of each class from the input space is gathered, the network should be able to learn to distinguish between them [30]. This being the case, when data was collected, special care was taken to ensure that an adequate sample of each class from the input space was taken.

It was thought that one way to do this was to attempt to determine where the decision boundaries were. After that was done, data taken could be focused from the entire periphery of those spaces, along with a sampling of their interiors. If this was needed, one way it could have been done was by sampling inputs from the entire range of pressures possible to be exerted on the Smart Handle for each possible movement. Another way of doing this might have been selecting human subjects from a wide range of body mass indexes, strengths, heights, weights, and genders. However, neither of these were needed. It was discovered that, in most cases, the character of each of the different inputs was sufficiently different than the others to make distinguishing them simple. It would turn out to be more of an issue during inference, but not enough to affect results.

Other options for ensuring good data were taken included testing each subject with a variety of different grasping positions. This was used in the first round of human trails. Each of the seven subjects performed a series of movements on the Smart Handle with grips at zero degrees, and at angles of positive and negative twenty degrees from zero. This attempts to account for any deviation in natural grasp that a population might display [24]. In this work the angle between the different grasping angles was lowered to plus and minus 15 degrees. Further information will be given describing what this means in Chapter Three.
2.4.3 Evaluation

Evaluating the success of the machine learning model was rather straightforward. Data collected from the subjects was split into training and testing sets as described above. However, it was important that these sets be chosen by splitting the data in the appropriate ways for each operational scenario. One for the multi-subject scenario, where the entire group of subjects was split into two, not simply splitting the entire dataset randomly. And another for the single-subject scenario where only data from one subject at a time would be considered, and the sets corresponding to each motion from that subject were split into two: a large one for training, and a small one for testing.

It was crucial that in multi-subject evaluation there was no data from individuals used to train the network allowed to contaminate the testing set. One of the points of this work was to determine whether this technology could be used effectively right “out of the box”. That situation assumes the latest user is a totally new person the network has never seen before. If the network was tested on data from a person it was trained on, this would not shed light on the system’s ability to fulfil that goal.

If the above was done successfully, and in multi-subject training the network was trained on one group of individuals and tested on another, evaluation in this case becomes simply a matter of testing the accuracy of its classifications. It was stated in section 1.5.1 that an accuracy of 85% and above was desired. If, after training, the network was able to classify 85% or more of the testing set correctly, it was counted as a validation of that aspect of the hypothesis. In single-subject training the splitting was less complicated, but still the general idea of clean data remained
the same. Care was taken to make sure all data was labeled correctly, and no incorrect data was used during any portion of the training. Doing so would have seriously compromised the results.

2.5 Design of Experiment

In order to substantiate or invalidate the hypothesis, an experiment was devised that would show if a neural network was capable of reaching the goal of 95% accuracy in single-subject training and 85% accuracy in multi-subject training using the Smart Handle device. Its controlling neural networks needed to learn wrist pronation and supination, ulnar and radial deviation, gripping, and rest at least individually at the stated accuracy.

First, a set of neural networks architectures was gathered and paired with an assortment of different learning algorithms. A software library was written to create the different architectures, as well as the different learning algorithms. They were tested and assessed one by one, in many combinations after the data was collected in order to be tested and evaluated. Possible architectures included those of varying numbers of hidden layers and those with varying degrees of connectedness.

In addition to the neural network models, the support hardware to allow PCs or SBCs to interact with the Smart Handle was developed. Originally, the system used an Arduino Mega as a set of ADCs and a multiplexer, and a laptop PC for support. Another that used a Raspberry Pi and MCP3008 ADCs was also considered. Both systems acquired data from the Smart Handle reliably. In the end, the majority of the research was done with a LattePanda Delta x86 SBC with MCP3008 ADCs taking data from the strain gauge amplifiers via an Arduino Micro using an Atmega 32U4.
These setups were a critical component in the experiment design and will be described further in Chapter Three.

After a suitable library of network models was compiled and the hardware and software worked reliably together, a program of human trials was designed. Full Internal Review Board (IRB) approval was sought and received. The number of participants and the type of motions and pressures they performed was debated at length. A highly automated procedure was developed to ensure the trials went smoothly and quickly. Dr. Ferdowsi, Simon Kudernatsch, and Christopher Wolfe, who had conducted the first set of trials, were questioned and consulted throughout this process. This process was difficult and complex and carried with it ethical considerations that were highly sensitive, as well as requiring lots of paperwork.

At this point, each of the models were trained on the dataset and had their final set of synaptic weights and biases saved for future testing and inference. This took a large amount of time and computing. There were concerns regarding the capability of SBCs in accomplishing this task, especially if they were ARM based. Once this was done, each of the trained networks was tested on the testing sets. Again, for the multi-subject scenario this set did not consist of any data gathered from subjects present in the training set. An accuracy score was easily assigned when this process was completed. Next, the networks were ranked in terms of their success.

Finally, this created a scenario in which the hypothesis could be meaningfully examined. If the models displayed 95% accuracy or greater in single-subject, and 85% accuracy or greater in multi-subject, then the hypothesis had been validated. Otherwise, either further work would have been done in tweaking the models, or perhaps there could have been flaws in the data collection procedure. This was not needed.
CHAPTER 3
METHODOLOGY

The methodology of an experiment to test a hypothesis is the core of a work like this. The following sections will describe in detail how each of the elements of the experiment were planned and accomplished. First, neural network development will be discussed in terms of what inputs were be used, why they had been chosen, how the outputs were be configured, what the network architectures would be, what learning algorithms were used, and what software was used, among other things. Following that will come an in-depth review of how the software and support hardware were tested to allow data collection to move forward. Next, a very brief review of NIH guidelines on human trials and how specifically the human trials should be conducted to collect the necessary data will be presented. Then, a plan will be proposed for training the numerous networks on the dataset, including how many elements there might be and how long it might take on different computing platforms. Finally, a description of precisely how the tested models would be assessed and ranked based on their performance is offered.

3.1 Updates to Existing Hardware

When the proposal for this work was written, it was thought amongst the team that the strain gauges on the Smart Handle alone would not be sufficient to make all the needed classifications. Early in the research much thought was given to finding new sensors that could be
added that would allow for a greater differentiation between regions of the input-space representing different output classes. Initial investigations into data collection and neural network testing showed that this thought was not correct. The strain gauge signals provided a sufficiently unique set of inputs for reliability differentiating classes. An explanatory example of this data will be shown in Figures 4-3 and 4-4.

The desire was to be able to distinguish all six DOF, three translations and three rotations (up/down, left/right, forward/back, roll, pitch, yaw), and when the Smart Handle hardware was designed the strain gauges were placed intentionally to achieve this. It would turn out that the tabs placed for mounting the strain gauges responsible for differentiating yawing movements were too thin, and therefore the signals off those sensors were erroneous due to material hysteresis. As a yawing movement was applied to the handle the tabs would deform to a point that they would not return to their original shape. As a result of this, yawing had to be omitted from consideration in inference tests in this research. The strain gauges placed to pick up the remaining movements worked well, and so the team was able to get workable data for five DOF (for a total of ten output classes, one for each direction of each DOF), along with the “grip” and “rest” classes.

Although not necessary for effective classification of the twelve possible output classes, one additional sensor was applied to the handle. This was an inertial measurement unit (IMU) for providing the software with the absolute orientation of the handle at all times. When the time comes for mounting the Smart Handle on an actual exoskeleton arm and beginning that test regime, knowing the relative position of the handle, in this case the end-effector, compared with the base of the exoskeleton will be crucial for the control algorithms. This was the only sensor added to the hardware setup, and it was not used in the actual classification of inputs. It may also be helpful
to note that in an actual exoskeleton arm application, the actuators in each joint will be equipped with an encoder for tracking real-time position angles of each link in the kinematic chain. This information will be used in the trajectory planning and inverse kinematic model. But an IMU such as the one discussed above provides absolute orientation, meaning it is not measured relative to any reference frame other than that of the Earth. So, it may still prove useful to include.

![Fig. 3-1.](image)

**Fig. 3-1.** A labeled photo of the full experimental hardware setup.

In the final hardware setup, sensor outputs are amplified by high-precision INA125 instrument amplifiers, before analog-to-digital conversion with two eight-channel MCP3008 chips, and then filtered and scaled in a preprocessing microcontroller, in this case a Atmega 32U4. The preprocessed data is sent to a computer where it is organized and directed for primary processing by the neural network on an Intel Celeron x86 processor on a LattePanda Delta that seeks to classify the movements into whatever classes of motion it has been trained to recognize. These classes are completely customizable and can be altered to fit many control applications. Classified data describing the desired state of the system and including things like magnitude and
direction of movement are then postprocessed by a postprocessing microcontroller, also an Atmega 32U4, which then sends the final commands to the controlled device. This technology is not solely useful in exoskeleton applications, these could be servos and motor drivers for wheeled vehicles, or wireless commands for remote control of land, aerial, or submarine vehicles. In this case the controlled devices are two brushed DC motors controlled with BTS7960 motor drivers.

Fig. 3-2. Electrical hardware diagram of the Smart Handle system.

3.2 Software

At all times when the experimental setup is active, three separate programs are running. There is a C++ script running on the preprocessing microcontroller, a Python script running on the primary computer, and another C++ script running on the postprocessing microcontroller. Every
20ms the preprocessing microcontroller is polling all sixteen channels of the ADC chips (there are two eight-channel chips) which are communicating with the microcontroller using the serial-peripheral-interface (SPI). The data from the ADC chips is smoothed with a running average of eight samples. On startup, an initial averaging buffer eight elements long is populated, then during operation each cycle begins with deleting the oldest of the eight samples and bringing in the latest sample. The eight samples from each of the sixteen channels is then averaged, and that set of values is saved to a variable and posted to the universal-serial-bus (USB) and picked up by the x86 Celeron processor on the primary computer. Since the preprocessing microcontroller is on an Arduino board and the Arduino integrated-development-environment (IDE) was used, posting data to USB also allowed for real-time plotting of each of the datapoints. This was useful in testing the Smart Handle since it allowed intuitive visual inspection of operation. In between each of the 20ms cycles the data from the IMU was also polled and included at the end of the list of amplified gauge readings and sent over USB to the primary computer.

The Python script on the primary computer was very complex. It, most importantly, held the code for the neural network, but also held the code for a robust graphical-user-interface (GUI) than allowed a user of the Smart Handle system to fully monitor and control all aspects of the system. A complete description of this software would not be practical in this paper. In short, a PyQt5 GUI was created that had eleven display panels, divided into six separate interfaces.

First, was what was called the “status dashboard” for real-time sensor monitoring. This was populated with the current values of each of the sensors on the handle from the data passed over USB from the preprocessing microcontroller. For easy troubleshooting, the highest valued
strain gauge was highlighted in red. Each of the gauge reading was combined into a 2-norm and plotted on an intensity meter. This will be explained more below.

Additionally, a data acquisition panel allowed full control of all data recording functionality that was required to collect training data for the neural networks. User biometrics could also be collected via this panel. Right before human trials, a new widget was added to this panel that linked to a text document that held a randomized list of movements for automating and smoothing data collection. Again, more on this below.

At the bottom of the GUI, direct manual control of the testing motors was made available. As mentioned above, the primary portion of this research revolved around the pronation/supination and radial/ulnar deviation DOF, so there are two separate motor control areas. From here each of the motors could have their speed, position, and direction of movement set. This was very useful for initial dynamic and actuated testing scenarios that were useful in probing practical operation and led to the development of the second generation two-DOF wrist prototype described in Chapter Five.

At the top middle of the GUI, network generation capabilities are found. These widgets generate feedforward multilayer perceptron type neural networks using the PyTorch machine learning framework, developed by Facebook’s machine learning research team. Network architecture, activation function, loss function, optimizer algorithm, learning rate, number of training epochs, locations of training and testing datasets, and helpful mode select push-buttons are all controlled through this area.
Fig. 3-3. A screenshot of the primary GUI used for the experiment.

At the extreme right, two groups relating to neural network operation are found. On the top are the network training and testing panels. These allow the user to specify a name for the network being created, and initiate the training process, after which the network parameters will be saved. The same can be done in order to test pre-trained and saved networks, simply by specifying the name of the parameter file. Below that, the inference panel is located. This panel takes in a network parameter file and syncs with the status-dashboard to allow for real-time inference, or real-time network testing on actual user interaction with the handle. It also displays the primary and secondary movements being classified, which was important for testing the software’s ability to generate arbitrary vectors from primary directions. At the bottom, there are six intensity meters that display the real-time output levels on each of the output neurons that
correspond to each of those movement classes. This also allowed testing of arbitrary vector
generation.

![Network generation](image)

**Fig. 3-4.** A labeled screenshot of the different functions of the GUI used for the experiment.

### 3.3 Data Collection

The data for the following experiments was collected in human-subject trials approved by the Institutional Review Board of Northern Illinois University, DeKalb, Illinois (# HS20-0244).

Human subject research is a delicate topic, and the current Covid 19 pandemic only exacerbated this. The National Institutes of Health, or NIH, has published a large catalog of information that explains how to properly conduct human trials that was referenced in preparation for collecting the data [31]. It should be noted, that due to the paperwork, difficulty, ethical concerns, and the Covid 19 pandemic, the way the data is collected was very carefully planned before any trials are
begun. Twelve volunteers performed gripping, rolling, pitching, yawing, forward/backward, left/right, and up/down movements on the handle. Not all participant performed all movements. Priority was placed on gripping, pitching, and rolling to simulate pronation, supination, radial deviation, and ulnar deviation movements in a wrist exoskeleton. The subjects grasp of the handle was placed in three orientations: at zero degrees, at +15 degrees, and at -15 degrees deviation from the line normal to the handle. The handle was placed at a height so that the subject’s elbow was at 90 degrees flexion during the collection. For each movement at each wrist orientation, six collection runs were performed that each lasted three seconds. Two were done at low intensity, to provide the neural network with exposure to samples with a low signal-to-noise ratio, two were done at high intensity to prevent the neural network from attempting to extrapolate, and two were done at a mid-range intensity to help fill the gap in the input space. Note that this means a total of 270 seconds of data was collected from each subject. The ordering of movements collected from each subject was randomized to prevent fatigue bias potentially caused by subjects always performing the same movements last. A widget on the GUI automatically imported these lists and displayed them for the researcher running the trials. This reduced the likelihood of error and data corruption.

3.4 Neural Network Development

Every neural network trained and tested with the Smart Handle data was a multilayer perceptron. The relatively low number of inputs and the fact that the primary task of the network was classification, meant anything more sophisticated was not required for most applications.
The variety of options present in selecting network topology, activation function, loss function, optimizer, and hyper-parameters is substantial. Given the primitive nature of current understanding of neural networks, appropriate selections were made by trial and error. In each application tested, the standard loss function Mean-Square-Error (MSE) was shown to be sufficient.

**Fig. 3-5.** Typical effective ANN topology.

In order to choose the best activation function, most efficient optimizer, and the best network topology, a random sample of the data collected was chosen for creating training and testing sets. This was used to see how well the different optimizers would learn the training sets in a given amount of time. These tests showed that the standard activation function Rectilinear Unit (ReLU) was all that was needed. Since most other activations are more computationally
expensive, ReLU was selected for all subsequent tests. Adagrad, an adaptive learning algorithm, was the quickest and best performing optimizer on the random dataset. To save time, this became the go-to standard setup for working with Smart Handle data. Only in cases where performance failed to meet expectations were these changed.

**TABLE 3-1:  NETWORK FEATURES BY ACCURACY TESTED ON A RANDOM SAMPLE OF INPUT DATA**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Name</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ReLU</td>
<td>97.10</td>
</tr>
<tr>
<td></td>
<td>Leaky ReLU</td>
<td>97.05</td>
</tr>
<tr>
<td></td>
<td>GELU</td>
<td>96.93</td>
</tr>
<tr>
<td></td>
<td>Sigmoid</td>
<td>96.07</td>
</tr>
<tr>
<td></td>
<td>SiLU</td>
<td>95.86</td>
</tr>
<tr>
<td></td>
<td>Hardshish</td>
<td>95.34</td>
</tr>
<tr>
<td></td>
<td>CELU</td>
<td>94.81</td>
</tr>
<tr>
<td></td>
<td>ELU</td>
<td>94.61</td>
</tr>
<tr>
<td></td>
<td>SELU</td>
<td>94.54</td>
</tr>
<tr>
<td></td>
<td>Softsign</td>
<td>94.04</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adagrad</td>
<td>97.22</td>
</tr>
<tr>
<td></td>
<td>Adamax</td>
<td>97.15</td>
</tr>
<tr>
<td></td>
<td>AdamW</td>
<td>97.05</td>
</tr>
<tr>
<td></td>
<td>Adam</td>
<td>95.43</td>
</tr>
<tr>
<td></td>
<td>Rprop</td>
<td>88.08</td>
</tr>
<tr>
<td></td>
<td>ASGD</td>
<td>44.20</td>
</tr>
<tr>
<td></td>
<td>SGD</td>
<td>40.21</td>
</tr>
<tr>
<td></td>
<td>Adadelta</td>
<td>25.89</td>
</tr>
<tr>
<td></td>
<td>RMSprop</td>
<td>16.75</td>
</tr>
</tbody>
</table>
Next, tests were planned to determine the number of hidden layers and the number of neurons per hidden layer. It was shown that after there were fifty neurons in the one hidden layer, accuracy in classification on the random dataset was consistently over 97% and did not perform any better with more neurons. Since the accuracy saturated very close to 100%, it was thought that for this application only one hidden layer would do fine. A small number of probing tests showed that accuracy only decreased with more hidden layers.

Fig. 3-6. A plot of ANN accuracy versus number of neurons in hidden layer.
3.5 Generation of Arbitrary Output Vectors

Since the Smart Handle relies on a neural network optimized with supervised learning methods, training data must be provided to the system before it is useful. One of the key advancements toward developing this technology was motivated by data collection. Since fluid continuous trajectories are desired to be output by the system, the networks must understand every possible movement, magnitude and direction, the user could input. Collecting data on every possible direction in perhaps as much as six DOF for robot arm applications would be prohibitive. Even collecting data for simple two DOF applications would pose issues with classifying the training data and avoiding discretization of the output. To solve these issues simultaneously, a
data collection/neural network output processing technique was developed that allowed for training data to be collected in only movement directions defined by the basis vectors that describe each desired DOF.

For example, in a wheeled vehicle, training data need only be collected for forward/backward and left/right. If the user should input a movement direction that is a combination of those two movements, the network can interpolate between the classes it understands to output the correct desired direction, even though it has never been trained on it directly. Or, for a wrist exoskeleton imitating a ball joint with three DOF, data need only be collected for a pure roll movement corresponding to wrist pronation/supination, a pure pitch movement corresponding to wrist radial/ulnar deviation, and a pure yaw movement corresponding to wrist flexion/extension. Post processing steps performed on the data output by the neural network have been shown to be capable of interpolating between the necessary basis vectors which are then scaled by an intensity parameter calculated from the raw sensor data and by what component of the output is dedicated to each classification. Not only does this greatly reduce the amount of training data needed, but it also provides fully continuous output trajectories. Intensity is calculated as the 2-norm of the sensor readings, which can be interpreted as the magnitude of the position vector locating that point within the input space:

\[
|\vec{x}| = \sqrt{x_0^2 + x_1^2 + \cdots + x_{n-1}^2}
\]

\textit{eq. 3.1}

Along with the physical handle device, the software developed for it includes a user-friendly graphical-user-interface (GUI) from which every operation needed for using the handle can be controlled. Data can be observed in real-time, saved in a text file, loaded from a text file
and observed and graphed, as well as other features. Designing, training, testing, and running neural networks as well as observing real-time outputs are also done through the GUI.

3.6 Operational Scenario: With User Calibration

The primary use case tested thus far was control of a robot arm for upper extremity exoskeletons. The Smart Handle was initially conceived with this application in mind. The work was motivated by the problems discovered by others trying to create a practical powered exoskeleton that maintains movement in the wrist without exposing the wearer to dangerous loads. Many exoskeletons either sacrifice the mobility of the wrist in order to safely decouple the wearer from any loads the robot is taking, or attempt to maintain that mobility by terminating the exoskeleton just before the wrist and allowing the wearer to interact with loads directly with their hands [24]. Ideally, a system would be devised that could achieve both simultaneously. Additionally, implementation of ML methods eliminates the need for complex combinational for fuzzy logic rules as seen in other wrist exoskeletons [10][21].

Initial testing of the system for upper-extremity exoskeleton applications was done in the pronation/supination and radial/ulnar deviation DOF of the wrist. Again, training and testing sets for the ANNs were compiled from the data collected from the twelve volunteers. In this case the gripping, rolling (pronation/supination), and pitching (radial/ulnar deviation) was used. Neural network training was done with the parameters mentioned above. This use case was meant to simulate a situation where the handle was not programmed before interacting with the operator, meaning a calibration process must be completed for each individual attempting to use the handle.
To quantify the performance of the system, the 270 seconds of data for each of the twelve subjects individually was split into training and testing sets. Neural networks were trained on the data from only one subject and then tested on separate data from the same subject.

![Diagram of training classes for a two-DOF wrist exoskeleton](image)

**Fig. 3-8.** A diagram of the training classes for a two-DOF wrist exoskeleton.

3.7 Operational Scenario: Without User Calibration

Since the above tests suggested that operator setup can be greatly reduced with this system compared to others, it was desired that it be determined whether setup could be eliminated entirely. In order to achieve this, the data collected from eleven of the twelve subjects, excluding Subject 5 due to errors in that dataset, was compiled into one large dataset. Ten of the subjects’ data would
be used for training the network, and the remaining subject’s data would be used for testing. This division would simulate a handle trained on many people, hopefully with enough variation in gripping characteristics to properly cover the input space and allow for generalization to any random person. If successful, the individual who did not provide training data would be able to operate the exoskeleton without any calibration whatsoever. The system could be used “out-of-the-box” and calibration/setup would be entirely eliminated.

First, every possible combination of one training subject and one testing subject was performed, for a total of 121 ANNs. This data provided a good description of the variation of the dataset. The results showed that substantial variation is present in the possible grips and movements a human can have interacting with a handle. It also showed that substantial variation existed between male and female grip characteristics. Therefore, the two female subjects were eliminated from the subsequent tests in an effort to minimize variation and maximize neural network classification accuracy.

Next, ten random datasets were created with two training subjects and one testing subject, and another ten with three training subjects and one testing subject, and so on until ten random sets with seven training subjects and one testing subject were created. Finally, every possible combination of eight training subjects and one testing subject was created. When the data organization was completed a neural network with the standard parameters and topology for the Smart Handle was trained and tested for each of the 190 datasets. More information on this is presented in Chapter Four.
3.8 Additional Application: Wheeled Ground Vehicle

The simplest use scenario that has been tested with the Smart Handle was the control of a wheeled vehicle that moved in only two DOF, forward/backward and left/right. This test was done as a proof-of-concept for further work in adapting this technology to controlling mobility chairs for those with compromised arm/wrist/hand motor function. A small, remote-controlled, differentially steered cart was created to act as the control device.

![Fig. 3-9. Four views of a small radio controlled wheeled robot used for application tests.](Image)

Training and testing sets were created from the gripping, forward/backward and left/right data collected from the volunteers. Neural networks of the types described above were trained on these sets until MSE was less than one percent. Afterward, the test sets were passed through the networks and accuracy in classification was calculated by number of correct classifications divided by total number of attempted classifications.
Fig. 3-10. A diagram of the training classes and output of the Smart Handle system in two DOF in Cartesian coordinates.

It was desired that the networks properly classify the inputs and meaningfully assign magnitudes to the forward/backward and left/right basis vectors, such that their vector addition generated an output vector whose direction match that of the intended motion of the user. This is diagramed in figure 3-10 above.
CHAPTER 4

RESULTS

4.1 Operational Scenario: With User Calibration

As in the first experiment, the relative simplicity of the data and the fact that the training and testing data came from the same individual, very high accuracies were achieved. The results were as follows.

<table>
<thead>
<tr>
<th>Table 4-1: Control with Calibration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sex</strong></td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>Male</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Female</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Use of the handle in this case was highly successful. Further tests showed that only the data from zero degrees was necessary for sufficient performance, meaning only 90 seconds of data could be used during a very brief and minimally cumbersome calibration process. Neural network training was completed for each subject in less than ten seconds due to the small number of inputs and network parameters. Also note that this process can easily be automated and be completed with only prompts from the software and no supervision by trained technicians. This process is not dependent upon a laboratory setting or any equipment beyond the handle and its supporting electronics.

4.2 Operational Scenario: Without User Calibration

Since this research began one of the primary questions needing to be answered was how many individuals would have to provide training data for the neural network in order for it to generalize to an arbitrary person with a given accuracy. During the human trials data was collected from twelve subjects. The data from Subject Five was unable to be validated so it was not used, giving a total useful dataset of eleven individuals. The first test done on this data was creating a table where both axes were a list of each of the subjects. The subject in each column provided a testing set, and the subject in each row provided a training set. Each subject was used to test one network trained on each and every one other subject’s data. This was done to check and average the accuracy of a network’s classifications when only trained on one other person. The following table was produced from this test.
## TABLE 4-2: ONE TRAINING SUBJECT

<table>
<thead>
<tr>
<th></th>
<th>S0</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
<th>S9</th>
<th>S10</th>
<th>S11</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>99.83</td>
<td>73.59</td>
<td>62.46</td>
<td>53.82</td>
<td>61.77</td>
<td>63.12</td>
<td>61.84</td>
<td>65.81</td>
<td>48.09</td>
<td>38.43</td>
<td>92.13</td>
</tr>
<tr>
<td>S1</td>
<td>88.19</td>
<td>100.0</td>
<td>71.26</td>
<td>59.16</td>
<td>66.61</td>
<td>69.26</td>
<td>56.84</td>
<td>66.51</td>
<td>38.88</td>
<td>43.08</td>
<td>91.72</td>
</tr>
<tr>
<td>S2</td>
<td>61.57</td>
<td>70.89</td>
<td>96.91</td>
<td>75.95</td>
<td>70.98</td>
<td>53.07</td>
<td>60.96</td>
<td>73.58</td>
<td>40.98</td>
<td>49.27</td>
<td>79.66</td>
</tr>
<tr>
<td>S3</td>
<td>80.71</td>
<td>75.99</td>
<td>79.89</td>
<td>95.14</td>
<td>72.84</td>
<td>65.87</td>
<td>68.96</td>
<td>79.84</td>
<td>50.63</td>
<td>47.06</td>
<td>93.03</td>
</tr>
<tr>
<td>S4</td>
<td>74.19</td>
<td>79.99</td>
<td>70.97</td>
<td>73.28</td>
<td>99.41</td>
<td>55.11</td>
<td>77.21</td>
<td>74.61</td>
<td>43.49</td>
<td>45.12</td>
<td>76.01</td>
</tr>
<tr>
<td>S6</td>
<td>52.49</td>
<td>50.04</td>
<td>55.96</td>
<td>63.90</td>
<td>57.28</td>
<td>99.56</td>
<td>48.24</td>
<td>56.63</td>
<td>23.70</td>
<td>28.63</td>
<td>71.58</td>
</tr>
<tr>
<td>S7</td>
<td>72.21</td>
<td>68.34</td>
<td>70.45</td>
<td>65.94</td>
<td>61.47</td>
<td>57.22</td>
<td>99.81</td>
<td>86.50</td>
<td>54.85</td>
<td>30.26</td>
<td>77.73</td>
</tr>
<tr>
<td>S8</td>
<td>78.41</td>
<td>59.23</td>
<td>68.76</td>
<td>58.14</td>
<td>70.89</td>
<td>57.48</td>
<td>72.66</td>
<td>98.80</td>
<td>55.58</td>
<td>51.31</td>
<td>83.27</td>
</tr>
<tr>
<td>S9</td>
<td>40.03</td>
<td>37.72</td>
<td>44.94</td>
<td>44.02</td>
<td>42.46</td>
<td>24.34</td>
<td>56.70</td>
<td>60.20</td>
<td>97.58</td>
<td>68.34</td>
<td>55.44</td>
</tr>
<tr>
<td>S10</td>
<td>53.51</td>
<td>58.07</td>
<td>49.78</td>
<td>61.57</td>
<td>48.36</td>
<td>31.36</td>
<td>60.87</td>
<td>67.94</td>
<td>63.92</td>
<td>94.98</td>
<td>59.22</td>
</tr>
<tr>
<td>S11</td>
<td>61.54</td>
<td>68.98</td>
<td>77.65</td>
<td>79.27</td>
<td>66.73</td>
<td>80.51</td>
<td>62.31</td>
<td>69.20</td>
<td>55.09</td>
<td>44.83</td>
<td>100.0</td>
</tr>
</tbody>
</table>

The diagonal highlighted in green represents when the training and testing data come from the same subject. This diagonal also represents a set of results for one type of network, described in 3.4, applied to the single-subject operational scenario. Similar to the results presented in 4.1 movement classification accuracy on average is greater than 95%.

All cells of the above table that represent a test where the training and testing data both came from a subject who responded “male” for sex are white. All cells of the above table that represent a test where the training and testing data both came from a subject who responded “female” for sex are yellow. All cells of the above table that represent a test where training data came from a subject who responded “male” for sex and testing data came from a subject who responded “female” for sex are blue. All cells of the above table that represent a test where training data came from a subject who responded “female” for sex and testing data came from a subject who responded “male” for sex are red. The averages of these groups are given below.
### TABLE 4-3: GROUP AVERAGES

<table>
<thead>
<tr>
<th>Group</th>
<th>Test</th>
<th>Average Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>Single-subject</td>
<td>98.37</td>
</tr>
<tr>
<td>Blue</td>
<td>Train: male Test: female</td>
<td>43.85</td>
</tr>
<tr>
<td>Red</td>
<td>Train: female Test: male</td>
<td>49.81</td>
</tr>
<tr>
<td>Yellow</td>
<td>Train: female Test: female</td>
<td>66.13</td>
</tr>
<tr>
<td>White</td>
<td>Train: male Test: male</td>
<td>69.17</td>
</tr>
</tbody>
</table>

Two things from the above table are immediately notable. First, the Blue and Red groups as well as the Yellow and White groups are very similar in average accuracy. Second, the values for the Blue and Red groups are much lower than the values for the Yellow and White groups. This would seem to suggest that there is some measurable difference between male and female gripping characteristics. Data from males alone will not be efficient in training networks for use by females, and vice-versa. This would also imply that there should be two separate networks for use in the multi-subject scenario, one for males and one for females. Given the numbers of each involved in this study, focus will from now on be on a network optimized for males.

The above data pertaining to networks trained on only one individual and then tested on a separate new individual represents the first datapoint that can be used in a regression analysis to determine an equation that defines the relation between number of training subjects and
classification accuracy on a new person. Next, datasets were created and tests done for the situations where there were two, three, four, five, six, seven, and eight training subjects. With nine male datasets available, there were far too many combinations of subjects for the training sets to do an exhaustive search. Therefore, ten combinations of subjects were randomly generated for each of the seven possible numbers of subjects for testing. Networks of the standard type were trained and tested of each of these sets. From these tests, the following plot of performance versus data was generated, and a power regression was performed to compute the best-fit curve for the data conforming the theoretical models of shallow neural networks.

![Network Performance](image)

**Fig. 4-1.** A plot of classification accuracy versus number of subjects in the training set (extrapolation in blue).

As can be seen above, with current methods, accuracy versus data performance saturates at around 90% accuracy. A prohibitively large sample of individuals would be required to achieve
greater than 95% accuracy, and it could be expected that at least 20-30 individuals would be required to achieve around 90% accuracy. Due to this result, it is believed that a more sophisticated type of neural network might be needed to see successful performance in this scenario, at least in the testing phase, without feedback available in real-time inference.

After the success of the above test, it was desired to find out in how many DOF the system could accurately classify movement. One subject provided data for ten movements in five DOF: forward/backward, left/right, up/down, roll, and pitch. This would allow position to be altered in three dimensions, as well as allowing for orientation to be altered in two dimensions. (It should be noted that the handle design should be capable of providing orientation in three DOF, however a flaw in a part of the hardware for the only existing prototype meant yaw had to be omitted for this stage of testing.)

![Diagram of training classes for six-DOF exoskeleton arm applications.](image)

*Fig. 4-2.* A diagram of the training classes for six-DOF exoskeleton arm applications.
With the five DOF listed above, along with the data collection and basis vector interpolation technique devised for the Smart Handle, inclusion of a kinematic model and equations of motion along with appropriate trajectory planning should allow this system to fully and fluidly operate an exoskeleton arm with up to seven DOF (the six DOF of the Smart Handle output can be converted into commands that actuate a robot arm with seven or more DOF). Neural network training, testing, and inference showed that higher accuracy, greater than 95%, can be achieved in this scenario, at least under static laboratory conditions.

For the sake of completeness, presented below are a few figures displaying raw data collected from the Smart Handle system. With the data that was collected, plots can be generated of the time varying signals on each of the 16 strain gauge channels. Two, thirty second collection runs were done where a sequence of translation movements and rotation movements, starting with a plain grip, were done separated by rest periods.

![Fig. 4-3. Raw data for each of the different six translation classes plus “grip” (Grp) and “rest” (R).]
The sequences were “grip”, “forward”, “backward”, “left”, “right”, “up”, “down” for translations, and “grip”, “positive-roll”, “negative-roll”, “positive-pitch”, “negative-pitch”, “positive-yaw”, and “negative-yaw” for rotations. In the figures for each thirty second sequence of movements, the different movement and rest periods are labeled and delineated by vertical black lines. Note the unique color sequences from lower magnitude to higher magnitude (y-axis). This is thought to be the primary characteristic the networks use to determine class. Also note in Figure 4-4, in the rest periods following “positive” and “negative” yaw, that the black and white signals do not return to their pre-activation state. This is due to the material hysteresis issue in the tabs on which the strain gauges for measuring yaw are mounted and prevented the yaw DOF from being tested extensively in inference. It should be noted that it is believed that this issue can be overcome simply by training the networks to recognize the error state as an example of “rest”, but further research is needed to prove this.

Fig. 4-4. Raw data for each of the six rotation classes plus “grip” (Grp) and “rest” (R).
Finally, shown in Figure 4-5, is one example of the network classification tests plotted on a graph for easier consumption. The desired output of the network determined by the labeled testing datasets are graphed in a solid line color-coded to movement class. The actual network output is plotted as a color-coded triangle, one for each input element. Since over 2500 input elements are tested and graphed in this figure, the triangles blur into a thick solid line. It can be seen that the thick solid line at output 1.0 is always matching the color of the thin vertical line immediately preceding it. That means the classification was correct for the vast majority of the input elements. Figure 4-5 shows how the network is still able to classify with greater than 95% accuracy in the single-subject scenario even when all twelve translations and rotations are tested.

Fig. 4-5. An example of the network test output including all twelve translation and rotation classes plus “grip” and “rest”.
4.3 Multi-Subject Scenario Inference Tests

As seen in Figure 4-1, average multi-subject classification accuracy only barely made it up to the target region of 80-85%. This was less than ideal. Large numbers of tests were done across a several week period to bring these numbers up, but sizable increases were not achieved. The only remaining test was to load the networks for real-time inference on the multi-subject training sets. Two types of interactions with the handle were observed during these inference tests. First, referred to as “blind” tests, were done by having the new individual interact with the handle with no information of the network outputs. The subject was blind to any errors. There was no feedback in this type of system, recreating the computer tests done on the corresponding datasets. Similar performance to those tests were expected and observed.

The second type of interaction observed were referred to as “observation” tests. In this case, the subjects were allowed to observe the output of their interactions with the handle and make subtle corrections to their actions based on feedback from the handle. This case more closely recreates any practical interactions with a controlled device. Within only a few minutes subjects were able to make small corrections to their use of the handle enough to notably eliminate errors. This observation was interpreted as a sign of validation for the Smart Handle control technology.

It is interesting to note that the Italian EMG research team that produced the EMG sensor network designed for simple and easy donning and doffing also made use of this phenomenon. In that case, the need for precision placement of the EMG sensors was ignored in favor of a more user-friendly and repeatably placeable sensor network with embedded and non-alterable sensor
positions. They observed that the plasticity of human adaptation was enough to largely bridge the gap in performance between the original technique and their new simplified technique.

**Test Scenario 1: blind**

```
   Human          Smart Handle System  →  Trajectory Vector
                ↑                            ↘
                ↖                            ↘
                Observation
```

**Test Scenario 2: Observation**

```
   Human          Smart Handle System  →  Trajectory Vector
                ↑                            ↘
                ↖                            ↘
                Observation
```

**Fig. 4-6.** The two inference test scenarios for multi-subject, no calibration operation, with and without feedback to user.

### 4.4 Additional Application: Wheeled Ground Vehicle

Given the simplicity of the data and the distinctive differences between the input data for the output classes grip, forward, backward, left, and right, over 98% accuracy was achieved on the test datasets from both subjects used. It was then desired to see how little data was needed to inform collection optimization. Only eighteen seconds of data per movement, which combined amount to only 72 seconds of data was shown to be perfectly sufficient to allow the individual from whom the training data was collected to fully control movement in a plane in any arbitrary direction and any arbitrary magnitude.
CHAPTER 5
CONCLUSIONS

5.1 Conclusions

The Smart Handle technology provides a solid-state, software-defined, low-cost, low-complexity human input control system that has many advantages. It is highly customizable, mechanically simple, and resilient. Compared to other handle-type human-input control systems, the Smart Handle can offer much more intuitive control in more DOF. In exoskeleton applications it may solve the problems of high cost, high complexity, and impractical donning and doffing that currently trouble the technology. This research concludes that both hypotheses posed are affirmed.

The development of the Smart Handle device is still in its early stages. Although promising, much work must be done to validate its usefulness. The first step would be to explore more sophisticated ANN types to determine whether a zero calibration, out-of-the-box mode of operation is possible. More subjects may be recruited as well to further validate the work in Sections 3.7 and 4.2. The second step would be to implement an actuated wrist model capable of moving in at least the pronation/supination and radial/ulnar deviation DOF of the wrist. A compliant and responsive prototype will be required. This provides a relatively simple proof of concept for practical exoskeleton and robot arm applications. Beyond that, a more complete arm prototype that includes elbow flexion/extension, shoulder flexion/extension and shoulder internal/external rotation DOF should be produced. Forward and inverse kinematic models and
the equations of motion will be needed for the arm, as well as a method of trajectory generation that can map the position vector output of the handle into changes of the arm’s state vector in joint-space. Once this is achieved, the logical next step would be to combine subsystems developed in the second and third steps to develop a full exoskeleton arm around the Smart Handle intended motion system.

5.2 Future Additions

One of the issues faced during this research was the design of the original actuated wrist. Since the mounting of the handle and radial/ulnar deviation motor was not balanced, a relatively complex control method was needed. As the handle and motor moved through the pronation/supination dimension, the lever arm that was created got bigger and smaller depending on the angle it made with the fixed mount. As such, the pulse-width-modulation (PWM) signal that was sent from the postprocessing microcontroller to the motor drivers had to be attenuated sinusoidally with the angle so as to give somewhat constant speed of rotation. This is not ideal. A better solution would be to design a wrist that was balanced in all DOF.

Another issue with the first wrist was the fact that it used non-backdrivable brushed DC motors. This meant that the collection of data from the handle on that wrist would be difficult for the dynamic case. Also, brushed DC motors are not known for their quick agile performance. Generally, they are sort of slow and lead to cumbersome movements for the device.

In order to solve these issues, a new wrist was desired. The first step for this would be to devise a drive system that would properly suit the application. Far better for this case than brushed
DC motors are brushless DC motors. They are backdrivable, compliant, and can provide very quick, agile, natural performance when paired with quality motor drivers. To that end, work on the new wrist began with investigations into custom planetary gearbox designs. Eight iterations were done on this drive, culminating in a design pictured below. These are 100mm diameter, two-stage, 25:1 reduction planetary gearboxes designed for use with 90mm brushless pancake motors.

![Fig. 5-1. (Left) Computer-aided-design (CAD) rendering of a two-stage 25:1 planetary gearbox intended for use in upper extremity exoskeleton projects. (Middle/Right) Physical prototype.]

Once a solid design was in hand, it was adapted to two new situations. First, the new wrist prototype, and second a shoulder and elbow upper extremity exoskeleton. The new wrist was comprised of a new handle design that used eight cantilevers around its circumference instead of twelve. These were oriented in only the principal directions, since over the course of the research it was determined that it was highly likely that these are all that is needed for operation. Additionally, new grips were designed to fit around the cantilevers in a snug way so as to create
an almost continuous smooth perimeter around the handle. This would correct certain user comfort issues faced with the first version of the handle.

This handle was mounted on a system of two brushless motors configured in-line with each other so the whole wrist is constantly in balance no matter what orientation it is in. A 90mm brushless motor is responsible for the pronation/supination DOF and is driving a 5:1 single-stage planetary gearbox. Mounted on the output of the pronation/supination gearbox is a smaller 60mm brushless DC motor that drives a 4.8:1 single-stage planetary gearbox. This is the drive responsible for the radial/ulnar deviation DOF. The motors and drives can both be mounted in-line because the energy of the output of the 60mm motor is transmitted 90 degrees by a bevel gear. This can be seen in several of the images below.

Fig. 5-2. Two views of a CAD model for a “next-generation” 2-DOF actuated wrist exoskeleton.
This wrist exoskeleton is much smaller, lighter, and of higher performance than the first. Since it utilizes brushless DC motors, the level of control and agility possible is much higher. High quality driver boards called ODrive 3.6 brushless motor drivers have been purchased for use in actuating this prototype. This driver board can operate in many control modes that can be tailored to this application. These drivers can also configure the motors to behave a virtual springs and allow for ideal dynamic data collection on the wrist.

Fig. 5-3. Five interactions with the physical prototype of the “next-generation” 2-DOF wrist exoskeleton.
It is not only desired to evaluate Smart Handle performance in the application to wrist exoskeletons. The rest of the DOF of the human arm should also be investigated. To that end, the two-stage drives developed for the 90mm motors were designed into a full upper extremity exoskeleton. This device has two DOF in the shoulder, flexion/extension, and internal/external rotation. It also has elbow flexion/extension. Note that the human shoulder is actually a ball joint and has three DOF. In addition to the two mentioned above there is also abduction/adduction.
However, designing three revolute joints for each of these DOFs that all intersect at the center of the shoulder ball joint’s rotation is impossible while avoiding one of the joints colliding with the wearer’s head. At least one additional DOF is needed in an exoskeleton shoulder to pull the internal/external rotation joint motor away from the head of the user when abduction/adduction movements are being performed. Because of this, the abduction/adduction DOF was left out of the upper extremity exoskeleton developed for Smart Handle testing. This device is pictured below.

![Fig. 5-5. Two views of a CAD model of an upper extremity exoskeleton with a 2-DOF actuated shoulder and a 1-DOF actuated elbow.](image)

The exoskeleton is constructed of aluminum rods and 3D printed polylactic acid (PLA) and mounted on a commercial rucksack frame. At about 12 kg it is not light, but is only intended for use in testing the translational capabilities of the Smart Handle output vectors when handed over to a trajectory planning algorithm and proper kinematic model. The arms of the exoskeleton were
specifically designed to conform to the “two-link-manipulator” equations so that determining the forward and inverse kinematics as well as the equations of motion would be highly simplified.

Fig. 5-6. Four views of the physical prototype of the upper extremity exoskeleton intended to test the Smart Handle technology.

With the two-DOF wrist and the three-DOF arm, the Smart Handle can now be tested on five of the seven DOF of the human arm in practical dynamic situations. This is of high importance in determining the real usefulness of the Smart Handle control concept. Each of the two hardware setups will be tested, verified, and optimized individually, starting with the wrist and then moving on to the arms. The electronics and control software for the wrist is largely complete due to the research presented in this work regarding the original actuated model, however, the electronics and control software for the arms have not yet been designed or constructed. It is anticipated that the arm application will be much more complex due to the inclusion of translations in that scenario, and as such, completion will require a sizable length of time. When finished, if successful, integration of the two setups will begin.
It should also be noted that, although classification was highly accurate in testing, in running inference on the datasets containing all three DOF of translation and all three DOF of rotation, the networks as tested were not able to produce reliable arbitrary vectors. The post
processing done on the networks’ classifications was not possible due to the networks’ outputs all falling to zero. When the input data is stretched so far, so-to-speak, to try and get fourteen outputs with only around fourteen inputs, and when the input is a mix of all six DOF, meaning it is not closely reminiscent of any single input class, the outputs all fall to zero. It is known that accurate arbitrary output vectors in inference is effective with up to three DOF, so it is thought that simply dividing the processing into two neural networks running simultaneously, one for translations and one for rotations, would easily solve this problem. Tests of this are planned. This issue is not expected to represent a fundamental problem with the Smart Handle technology. Once this fix is tested, this functionality will be added to the system.
REFERENCES


