ABSTRACT

Title of Dissertation: STRAP CAPACITIVE SENSOR MEASUREMENTS OF ARM DEFORMATION CROSS SECTIONS FOR GESTURE RECOGNITION
Louis J Dankovich IV
Doctor of Philosophy, 2021

Dissertation Directed by: Professor Sarah Bergbreiter
Department of Mechanical Engineering

Wearable Human Machine Interfaces (wHMI) for gesture recognition have the potential to enable and improve a variety of fields including virtual reality, prosthetics, assistive exoskeletons, and therapeutic robotics. Challenges faced by current technologies include lack of suitable manufacturing infrastructure for newer technologies, cost, and difficulties with modeling forces with in individual grasps/fingers. Many of these technologies also suffer from degraded accuracy due to shifting of the sensors on the users body. As a result, their predictive power is often lowered by long term wear, user motion, and donning/doffing devices between uses. Typical methods to get around this include algorithms which attempt to correct for changes in the position of sensors on the user, using adhesive to attach sensors directly to the users body, and using a large number of sensors to generate a map of the limbs activity while lowering the dependence on any single sensors position. In this thesis, the Capsense, a new wHMI constructed from conventional off the shelf (COTS) parts is explored. This technology simplifies traditional multi-point sensor architectures

by using capacitive sensor straps to measure the shape change of individual cross sections of the arm due to muscle use. Because these straps are only dependent on radial location, they should be more robust to rotation around the arms circumference. The principle behind the operation and construction of the device is explained in detail; and a multi-gesture trial is performed on 5 subjects. Due to COVID era restrictions, further trials are limited to a single subject. These trials explore how robust the gesture recognition models generated with the Capsense are to long term wear, inter-day use, and when the elbow is added as an additional degree of freedom. Finally, the ability to model and predict individual finger forces and forces in full hand grasps is explored. Results obtained are shown to be comparable in accuracy to existing state of the art technologies for selected gestures, and point towards greater robustness to user motions and long term wear than other technologies, though difficulty with tracking thumb motions was found to be a significant limitation of Capsense.

STRAP CAPACITIVE SENSOR MEASUREMENTS OF ARM DEFORMATION CROSS SECTIONS FOR GESTURE RECOGNITION

by

Louis J Dankovich IV

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2021

Advisory Committee:
Professor Sarah Bergbreiter, Chair/Advisor
Assistant Professor Axel Krieger
Professor Miao Yu
Associate Professor Mark Fuge
Professor Jae Kun Shim

© Copyright by
Louis J Dankovich IV
2021

Acknowledgments

To Dr. Bergbreiter for giving me a long enough leash to follow my ideas, but yanking on the chain when I ran too far off course. And for having an almost superhuman level of patience with me during this process. To my committee, without whose helpful feedback and insights I would not have made it this far. To Elaine for not killing me even when I wanted to kill me. To Mom and Dad for not selling me to the circus when it still would have been profitable. To Jonathan and Teri for their insufferable optimism and useful advice. To James and Anne who trust me to be the cool and questionably sane uncle for my nieces. To Cat, because she and her family believed in me long before I did. To Steve, the best friend/rival a maker could ask for and who helped immensely by giving me someone to bounce ideas off and helping with software writing. To Ari for suffering through the painfully incoherent first draft. To everybody who hung out with me/dated me/supported me in my homeless street performer phase you have my deepest thanks and apologies. To Jack Terricloth whose writing, music, and addiction to bad ideals kept me sane during this process. To the Math Teacher who failed me in 4th grade . . . who has the PhD now? To all the people in my life I couldn't find something witty and succinct to say about know that I love and value you. And to Katherine . . . you know what you did!!!

Table of Contents

Acknowledgements ii

Table of Contents iii

List of Tables vi

List of Figures vii

List of Abbreviations xi

Chapter 1: Introduction 1
- 1.1 Wearable Human Machine Interfaces . . . 2
  - 1.1.1 What are wearable human machine interfaces? . . . 2
  - 1.1.2 Use Cases for wHMI . . . 3
  - 1.1.3 Contemporary wHMI Sensor Types . . . 4
  - 1.1.4 Challenges in wHMI . . . 7
  - 1.1.5 Placement . . . 8
- 1.2 Machine Learning Overview . . . 9
  - 1.2.1 Data Preparation . . . 10
  - 1.2.2 Machine Learning Protocols . . . 11
  - 1.2.3 Training and Validation . . . 15
- 1.3 Gestures Used . . . 21
- 1.4 Proposed Technology . . . 23
- 1.5 Outline of Thesis . . . 24

Chapter 2: Gesture Recognition via Flexible Capacitive Touch Electrodes 25
- 2.1 Introduction . . . 25
- 2.2 Principle of Operation . . . 27
- 2.3 Materials and Methods . . . 31
  - 2.3.1 Armband . . . 31
  - 2.3.2 Experiment . . . 33
  - 2.3.3 Analysis . . . 34
- 2.4 Results and Discussion . . . 37
  - 2.4.1 Prediction accuracy of different algorithms . . . 37
  - 2.4.2 Individual subject Accuracies . . . 37
  - 2.4.3 Intra-subject accuracy . . . 39
- 2.5 Conclusions . . . 40

Chapter 3: CapSense Sensor Design 46
3.1 Overview . . . 46
3.2 Design . . . 47
3.2.1 Materials Selected . . . 49
3.2.2 Connection Method . . . 51
3.2.3 Electronics . . . 54
3.2.4 Construction . . . 55
3.3 Experimental Validation . . . 56
3.3.1 Gestures . . . 57
3.3.2 Data Collection . . . 58
3.3.3 Machine Learning . . . 61
3.3.4 Combinations Tested . . . 61
3.3.5 Experiment . . . 62
3.4 Results And Discussion . . . 62
3.5 Final Design and Future Work . . . 65
3.5.1 Final Design . . . 65
3.5.2 Future Work . . . 65
3.5.3 Conclusions . . . 66

Chapter 4: Exploring the Limitations of Stationary Gesture Recognition with Capacitive Straps 68
4.1 Introduction . . . 68
4.2 Methods . . . 69
4.2.1 Microcontroller Settings . . . 69
4.2.2 Machine Learning . . . 73
4.3 Experiment . . . 73
4.3.1 Data Collection . . . 73
4.3.2 Data Processing . . . 74
4.4 Results . . . 76
4.4.1 General . . . 76
4.4.2 Transitional Class . . . 78
4.4.3 Gesture Limitations . . . 79
4.5 Conclusions and Future Work . . . 80
4.5.1 Conclusions . . . 80
4.5.2 Future Work . . . 81

Chapter 5: Extended Wear and Inter-Day Tests of Stationary Gesture Recognition with Capacitive Straps 83
5.1 Introduction . . . 83
5.2 Methods . . . 85
5.2.1 Placement . . . 85
5.2.2 Machine Learning . . . 87
5.3 Experiment . . . 88
5.3.1 Data Collection . . . 88
5.3.2 Data Processing . . . 89

5.4 Results and Discussion . . . 90
5.4.1 Results . . . 90
5.4.2 Discussion . . . 92
5.5 Conclusions . . . 95

Chapter 6: Gesture Recognition with Elbow Motion as an additional Degree of Freedom 97
6.1 Introduction . . . 97
6.2 Materials and Methods . . . 99
6.2.1 Equipment Used . . . 99
6.2.2 Experiment . . . 101
6.2.3 Machine Learning . . . 103
6.3 Results and Discussion . . . 108
6.3.1 Regression Results . . . 108
6.3.2 Classification Results . . . 111
6.4 Conclusions and Future Work . . . 118
6.4.1 Conclusions . . . 118
6.4.2 Future Work . . . 119

Chapter 7: Finger and Whole Grasp Forces 120
7.1 Introduction . . . 120
7.2 Full Hand Grasps . . . 121
7.2.1 Materials and Methods . . . 121
7.2.2 Experiment . . . 122
7.2.3 ML Protocol . . . 123
7.2.4 Results and Discussion . . . 124
7.3 Independent Finger Forces . . . 125
7.3.1 Materials and Methods . . . 126
7.3.2 Experiment . . . 127
7.3.3 ML Protocol . . . 128
7.3.4 Results and Discussion . . . 130
7.4 Conclusions and Future Work . . . 133
7.4.1 Conclusion . . . 133
7.4.2 Future Work . . . 134

Chapter 8: Conclusions and Contributions 135
8.1 Overall Contributions . . . 137
8.2 Published Papers . . . 138
8.3 Future Work . . . 138

Bibliography 140

List of Tables

1.1 Short title 18

2.1 Sensor strap feature importance in model by subject 40

3.1 Combinations Tested and Their Results 63

4.1 Software Configurable Settings 72
4.2 Combinations Tested and Their Results 78

6.1 Combinations Tested and Their Results 109

7.1 Combinations Tested and Their Results 124
7.2 Sensors Used in Individual Finger Force Tests 126

List of Figures

1.1 Placement Strategies with the sensors show in grey a) Targeted, b) Distributed. 8
1.2 a) k-fold analysis b) stratified shuffle splits. Both images modified from. [72] 17
1.3 a)Interpreting a Confusion Matrix, b) A normalized confusion matrix. 20

2.1 Previous approaches using CT for gesture recognition. A) Dual electrode methods use the human body as a variable dielectric. B) Single electrode methods treat the human as a negative electrode and measure change in capacitance as a function of skin proximity to the positive electrode. Arm cross-sections adapted from [41]. 29
2.2 The proposed electrode architecture encircles the entire arm. As a result it can spatially integrate the activity of multiple muscles at the same time, and generates a change in capacitance as a function of their combined contraction and relaxation. Two generic grasps are illustrated. In Grasp 1, the top muscle shown in blue is contracted and the left muscle shown in yellow relaxes. In Grasp 2 the top blue muscle relaxes, and the left yellow muscle contracts. Because this changes the distance between the skin and sensor strap, the two grasps result in different capacitive 'profiles'. Arm cross-section adapted from [41]. 30
2.3 The armband includes 7 capacitive touch electrode straps connected to a Teensy 3.2 and Bluetooth LTE radio. The armband is designed to fit over the subject's forearm and strap 1 is aligned with the knob of ulna bone on wrist to provide an anatomical landmark. 31
2.4 Grasps G1-G16 were selected from CTG [23]; F1-F10 represent individual finger motions; W1-W6 represent wrist motions. 42
2.5 Test rig used for consistent grasps between subjects. 43
2.6 Raw data from selected grasps taken from Subject 1. 43
2.7 Prediction accuracy for each subject as a function of algorithm. . . . 44
2.8 (a) Confusion matrices for subjects 1-3 and 5. These subjects all had very high prediction accuracies. (b) Close-up of confusion matrix for subject 4. This subject had poorer accuracy as seen in grasps, finger motions, and wrist motions. 45

3.1 The neoprene fabric used in these experiments. 50

3.2 The configuration of a) unshielded sensors, and b) shielded sensors. The shield layer is shown in green, the sensor layer in copper, the base dielectric layer in black, and the outer dielectric layer in white. 51
3.3 a) Loop connection the white portions represent conductive thread b) snap connection, the grey portions represent snap prong, c) Shielded snap connection, green represents the shield layer. Note the hole in layer to prevent shorting. Cartoons are not to scale 52
3.4 An example of the harness used to insure consistent placement of electronic components and minimize wire stress. The harness pictured was for snap connections. For loop connections, the wires would be threaded through holes punched in the same location as the upper row of snaps. In the shielded connections discussed in Sec. 3.2.3, the lower row of snaps would be connected to the sensors shield layer. . 53
3.5 The electronics subsystems, and the path taken by a single capacitive sensor signal in a) Baseline configuration (PCB1), b) shielded configuration (PCB2), and c) shielded and multiplexed configuration (PCB3). 56
3.6 An example of the finished sensors, electronics, elbow brace, wiring harness, and positioning strap. 57
3.7 The order in which gestures were performed in this experiment. . . . 59
3.8 a) the chair and desk setup for experiment, b) the companion application. 60
3.9 An example of the finished sensors, electronics, elbow brace, wiring harness, and positioning strap. 66

4.1 Internal Reference in blue vs oscillations observed in black [56]. . . . 70
4.2 The gestures used for the experiment in this chapter included:neutral, spread, fist, pincer, pinch, point, thumbs up, and transitional 74
4.3 Leave n out training scheme. The training set (green) was used for training a model and the model was tested on each of the remaining data sets (grey). The average result of training on all test/train pairs was recorded as the presumed real world accuracy. 75
4.4 Confusion matrix showing raw results for static classification. 77
4.5 Results of removing gestures involving the thumb a) pinch removed, b) thumb up removed, c) pinch and thumb up removed. 79
4.6 a) The intrinsic muscles of the thumb image from motionplus.com [63], b) the extrinsic muscles of the thumb image from geekymedic.com [64]. 80

5.1 Snaps used to align between trials are circled in orange a) tightness on arm and b) position on elbow, c) the sensors worn on arm. 86
5.2 Setup a) Tape outlines were used to insure position of chair and laptop between trial sessions b) the hand was rested on the desks mouse pad with middle finger touching edge of desk when hand was in neutral position. 87

5.3 Results of modelling with SVC. Globally on the X axis results are grouped by Gestures Used(GU) in training, and Y axis by the Session Label(SL). On individual confusion matrices, the x-axis shows the result of varying the *tr*, and the y-axis shows the results of varying the *tp*. In both cases they can be seen to have a fairly negligible impact compared to SL and GU. 90
5.4 The results of training with LDA. Globally on the x-axis results are grouped by Gestures Used(GU) in training, and y-axis by the Session Label (SL). On individual confusion matrices, the x-axis shows the result of varying the *tr*, and the y-axis shows the results of varying the *tp*. Raising the *tp* above 40% significantly lowered the accuracy. . 92
5.5 This plot shows all the datasets collected and their samples in sequential order. Of particular interest is the fact that the 93

6.1 The range of motion the users arm moved through in this experiment. For convenience a 90 degree bend of the elbow was considered the 'neutral' position and the room was used as a frame of reference. As such the neutral position corresponding to 0 degrees from the rooms frame of reference a) full flexion corresponds to 70 degrees b) neutral bend corresponding to 0 degrees, c) full extension corresponding to -90 degrees. 101
6.2 a) Gestures used included neutral, fist, spread hand, pincer, pinch, point, middle finger point, victory, pinky point, and horns; b)the range of motion for the elbow image adapted from [69]. 102
6.3 In this plot the class data has been removed from the capacitive measurements as discussed in Sec. 6.2.3.2, and for visual clarity they have been centered around 0 by having their means subtracted. When plotted against θ_y, a clear relationship can be seen between the arms angle and the capacitance measured. However it can be seen that this relationship displays non-linear behavior at the extreme angles. . 105
6.4 The top image is the raw data in the order it came in labeled with datasets and color coded by gesture. Data sets 1 and 2 show levels of noise, and extreme values not observed in the other datasets. The individual sensor values also drop over time in ways not observed in the other datasets, possibly indicating that the sensors shifted/settled on the arm over the course of these datasets. The bottom image shows the sensor readings after the portion associated with the elbows angle change is stripped from them. While dataset 1 and 2 still show some artifacts not observed in the other datasets, all datasets now have observable patterns that could be attributed to individual gestures. . 110
6.5 Results from classification with $\text{CAP}_{\text{n,normalize}}$ *features*. 112
6.6 Results from classification with $\text{CAP}_{\text{n,corr}}$ *features*. 113
6.7 The most commonly confused groups of gestures were a) victory vs pinky and middle finger extension, b) horns vs pinky extension and victory c) the pincer vs middle finger extension and victory. 115

6.8 The results of dropping multiple DOF gestures. 116
6.9 The results of dropping single DOF gestures. 117

7.1 Setup used in full hand grasp experiments a) palm view of glove instrumented with force sensitive resistors, b) grasping 1" diameter steel pipe used in the experiment. 122
7.2 Data from the two trials used for grasp force analysis. The top image shows the values recorded by CapSense. Note that sensors CAP2, CAP6 and CAP7 display a drift in values over the course of this experiment.The lower image shows measurements from the force sensitive resistors on the fingertips. 123
7.3 The by finger results of modelling normalized force sensor data with BR (on left) and SVR (on right). When using the more limited S_{Lim} feature set both achieve reasonable accuracy. 125
7.4 Independent finger forces setup a) sensor equipped rig b) arm resting on chair and position of rig relative to subject. 127
7.5 Independent finger forces setup a) sensor equipped rig b) arm resting on chair and position of rig relative to subject. 128
7.6 Bins used vs accuracy of prediction (averaged over all fingers in a grouping). 131

List of Abbreviations

A
ARD Automatic Relevance Determination

B
BR Bayesian Ridge

C
CGT Cutkosky Grasp Taxonomy

D
DG Dynamic Gestures
DOF Degree of Freedom
E
EMG Electromyogrpahy

F
FSR Force Sensitive Resistors
FMG Force Myograpy

G

H
HMI Human Machine Interface

I
ID Inter Day
IMU Inertial Motion Unit
IS Inter Session
IT Inter Trial

J
JA Joint Angles
L
LDA Linear Discriminant Analysis

LF Limb Force
LNO Leave-N-Out

ML Machine Learning

N

O

P

R
RF Random Forest

S
SG Static Gestures
SVC Support Vector Machine

T
TP Threshold Probability
tr Train Fraction

U

V

wHMI Wearable Human Machine Interface

Chapter 1: Introduction

Ever since we booted up the first computer, the question of how best to communicate with them has been asked iteratively with each generation of advances. Each advance in computing and machines has depended on changes in the Human Machine Interface (HMI). Early on, punch cards were a tedious but adequate HMI for programming, but the renaissance of computing occurred when video display screens and keyboards became commonplace in the 1970s. The growth of Graphical User Interfaces in the 1980s would have been impossible without the mainstream availability of computer mice, and the growth of the video game industry was made possible by joysticks. Without the development of touch screen interfaces, it is unlikely that smartphones would have succeeded as a mainstream technology. A current HMI technology challenge is wearable devices for real time Human Gesture Recognition (HGR), and stretches across a host of fields including robotic therapy [68, 74] , exoskeletons [1, 3], prosthetic limbs [14, 70], teleoperation [59], sign language recognition [2] and virtual reality (VR)[38]. This chapter will briefly review the current expectations of a wearable Human Machine Interface (wHMI), look at the advantages and challenges of current technologies, give a brief overview of the Machine Learning methods used in this work, propose capacitive sensor straps as a

low-cost, simple-to-use wHMI, and introduce the remaining chapters of the dissertation.

1.1 Wearable Human Machine Interfaces

1.1.1 What are wearable human machine interfaces?

The purpose of a Human Machine Interface (HMI) is to act as a transducer that takes some measurable signal from a subject, model information about the users physical actions based on that signal, and pass them to a machine to act on. For purposes of this work, hand gestures are the user input being measured. Actions that could potentially be of interest include static gestures (SG) such as an open or closed hand, joint angles (JA, e.g. how far the elbow is bent / extended), the amount of force that limbs (LF) are using to hold those angles / make gestures, and dynamic gestures (DG) which could be looked at as a sequence of gestures over time such as circling a hand or waving.

What distinguishes wearable Human Machine Interfaces (wHMI) from technologies such as proximity measurement and cameras is that they are worn on the users' body. The three basic components of a wHMI are the sensor technology used to capture user motion, a placement strategy for how sensors will be applied, and a machine learning algorithm to interpret sensor inputs. Previous studies have identified a few requirements for useful wHMI; the maximum delay between a user moving and the interface recognizing it should not exceed 150 - 300 ms [60] [84], and the accuracy of gesture recognition needs to be greater than 85% - 92% [70] [76].

1.1.2 Use Cases for wHMI

While there have been great advances in the use of cameras for gesture recognition, there are areas in which wearable technologies may be more suitable for daily use. The continued development of wHMI is motivated by use cases in areas such as physical rehabilitation, sign language recognition, prosthetics, assisted daily living, and sports training [38]. As an example of therapeutic needs, stroke patients in recovery may not have the full range of motion needed to make gestures or suffer from neurological difficulties preventing them from controlling the sequence in which muscles are used [54]. For these cases, understanding and retraining the underlying muscle use is often of more interest than the final position of the limb [38]. In the case of an amputee, there may not be any limbs to form gestures with. A device which can take inputs from muscles in the residual limb (e.g., force levels or combinations of flexed muscles) and act as a layer of abstraction is a desirable feature [15, 38]. For sports training, other players on the field and the position of the body itself may block camera angles. Clothing worn may also obscure information needed by cameras to make a precise determination of joint angles for both sports training and VR applications [20]. For sign language recognition, power demands of cameras, low light performance limitations, and privacy concerns have all been cited as arguments against computer vision based approaches [38].

1.1.3 Contemporary wHMI Sensor Types

An excellent review of the current state of wHMI for hand gesture recognition was published by Jiang et al in 2021 and goes far more in depth than there is space for in the confines of this thesis [38]. That said, this section gives a very brief overview of current sensor technologies being used both in research and in the field along with some of their strengths and weaknesses. All of the sensors in this section have shown accuracies greater than 90% for multiple class gesture recognition in studies, and most have, to some extent, had adoption outside the lab. Where applicable, the cost of commercial versions of the sensors is provided.

Electromyography (EMG) is a technology which reads the +/-5 mV nerve signals that trigger muscle activity. EMG is a mature technology about which many papers have been written and is the gold standard that other technologies compare their capabilities to. Some of its strengths include being able to observe nerve activity in users lacking limbs or with significantly weakened muscles, the ability to diagnose neurological issues, the fact that it can determine fatigue, and accuracies ranging from 92%-99% in the lab. Deployment of EMG faces a host of challenges outside of the lab however. Electrode shift over the course of a day reduces the accuracy of trained models [70], it is prone to picking up environmental noise [70, 82], user motion artifacts are often incorrectly recognized as gestures [82], nonlinear signal drift over the course of the day occurs due to muscle fatigue and perspiration [48], sensors over small muscles often pick up cross talk from neighboring muscles [48], obesity is known to decrease the accuracy of EMG sensors [47], and typical ap-

plications require 2-8 medical grade sensors at a cost of $600-$800 per sensor [14][8]. Typical applications involve sampling at 1kHz and require significant preprocessing to smooth the signal over a window of multiple samples prior to extracting a host of features that can then be fed into models [84]. There is also a significant hardware and computational overhead associated with filtering out signal drift, environmental noise, and motion artifacts [44].

Force Sensitive Resistors (FSR) have seen use in laboratory settings as an alternate wHMI known as Force Myography (FMG). In principle, FMG measures the pressure of a muscle against a sensor when the muscle flexes and relaxes [93]. It has performed comparably to EMG in offline studies, and has been investigated for uses including grasp counting in stroke patients, gesture recognition, finger force recognition, and measuring relative finger motions [92, 93]. Individual sensors are far less expensive than EMG electrodes at $10 per pressure sensor, no specialized hardware is required to filter environmental noise, and there is a lower computation overhead as models use sensor signals directly. FMG is however still sensitive to motion artifacts resulting in models that lose accuracy if sensors shift over the course of the day, and sensors can suffer hysteresis ranging from 4-10% and measurement errors of up to +/-6% for force measurements [93].

Flex sensors are electronic devices that change resistance as a function of curvature, and have been used to track individual joint angles for wHMI. Individual sensors can be obtained for as little as $12 per sensor and tracking the angle of a joint is biomechanically intuitive [2]. Like FMG they are far less expensive than EMG, do not pick up environmental noise, and do not require feature extraction. On the

downside, there is no way of tracking the location of a bend in a given sensor, so one sensor per joint is required for each joint to be tracked. Flex sensors also display nonlinear responses to bend angles [27], and possess saturation curvatures above which resistance does not increase which may vary from unit to unit. Together these factors lead to an increase in model complexity. Their typical application is in data gloves and they have been used in rehabilitation, teleoperation control interfaces, [59] and sign language translation [2]. These gloves may be difficult for users facing physiological impairments to don, are reported to be uncomfortable to wear for long durations; but can in theory can obtain far more sophisticated models of finger and hand than wrist mounted sensors, in part because many of the muscles controlling the smaller motions of the fingers and thumb are intrinsic to the hand [97]. At the low end, data gloves cost approximately $600 and range up to $15000 for professional grade gloves.

Inertial Motion Units (IMUs) are used to track acceleration and motion in Cartesian space. A 10 degree of freedom (DOF) IMU has a gyroscope to track pitch, yaw, and roll angular rates, an accelerometer to track linier acceleration, a magnetometer that can provide compass directions, and an onboard barometer. IMUs in stand-alone applications typically measure motions of entire limbs and have seen use in exoskeleton control and workout training devices [6, 13]. IMU sensors are relatively inexpensive at $30 per unit appear and are often used with other technologies in gesture fusion applications. For hand gesture recognition, fusing IMU data with other technologies has been shown to boost prediction accuracy for static gestures and allow better recognition of starting points for dynamic gestures

[30].

1.1.4 Challenges in wHMI

A number of areas that are commonly quoted as challenges to adoption include cost, lack of infrastructure/manufacturing methods to produce bleeding edge devices, motion artifacts, robust gesture recognition over a range of positions, and electrode contamination [14, 38, 54].

Motion artifacts and gesture recognition over a wide variety of limb positions are intimately tied challenges in gesture recognition [38]. Often the muscles used to change hand and arm position are the same ones being used to make gestures which can lead to confusion between changes in position and changes in gesture. Classification error rates have been shown to increase to an average of 35% when attempting grasps in positions different from training data [38, 83]. Typical methods used to correct for this involve training in multiple positions to allow for models to better recognize gestures made in different postures [12, 61].

Sensor shift is closely related, but rather than being a challenge associated with the limb being in a different position, it is a challenge associated with sensors sliding over the surface of the limb or being placed in a different position between trials. A review by Campbell in 2020 cites differences in sensor position on the arm as being a confounding factor that can lower model accuracy by as much as 13% [18]. Methods used to overcome this challenge typically use either many sensors to allow for the ones more prone to slippage to be ignored [38], or specialized algorithms in

order to correct for sensor shift [17].

Another continuing challenge is in determining force output of muscles. For prosthetics, continuous force level control, i.e. reading the relative force being applied by muscles in the residual stump, is the dominant form of control [38]. However it still faces many challenges in application to rehabilitation and measurement of forces within more dexterous finger motions [9, 50]. Muscle fatigue can also play a significant role in confusion of force regressions for long term wear [38].

1.1.5 Placement

There are two basic strategies for placing sensors on a subject as shown in 5.2. The earliest papers used targeted sensors in which there was a one-to-one

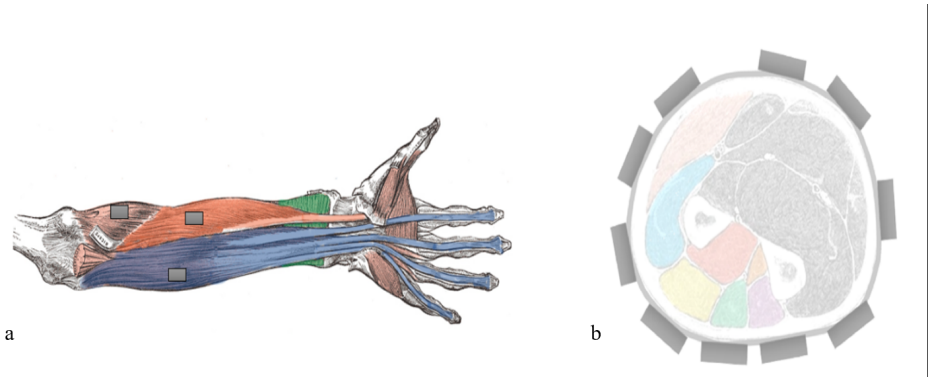


Figure 1.1: Placement Strategies with the sensors show in grey a) Targeted, b) Distributed.

relationship between a sensor and a muscle of interest. This is a very intuitive approach and does not require many sensors, but it does require precise placement of sensors [96]. More recent studies have taken a more shotgun approach in which

sensors are distributed over the limb evenly with the assumption that they will cover enough area to create useable patterns for model training. While this has the advantage of simplifying placement, it requires many more sensors [67]. Both strategies depend on 'point sensors', in that individual sensors are trained over a given point along the radial length of arm and around circumference. A sensor's location can be described as a 2 DOF coordinate; this means that there are two different directions in which a sensor can move which are likely to decrease its overall accuracy as a function of the sensor shift and motion artifacts discussed earlier.

1.2 Machine Learning Overview

Machine Learning (ML) in the broadest sense is the process in which models are built so that some set of *features* are used to make predictions regarding some *target* value. The *features* used in this work are readings from the capacitive straps worn on the subjects' arms, and the *targets* may include elbow angles, gestures being made, and forces being applied.

For purposes of this work, all models will fall into the category of Supervised Learning. This means that each sample provided to the model will have a label attached to it for the model to use as a *target* for training [36]. There are two flavors of Supervised Learning which are pertinent to this work. Classification models are used when there are discrete categories into which the *target* may fall. Regression is used when the *target* is a continuous value [36].

The vast majority of models require some preparation of data as discussed in

Sec.1.2.1. Once the data has been prepared there are a variety of models which can be used for classification, all of which have some tradeoffs in terms of their capabilities. These models are discussed in Sec.1.2.2. Of course merely building a model is just the first step. The methods used to train and validate models in this work are discussed in Sec. 1.2.3. For purposes of this work, the machine learning was implemented using Scikit-Learn, a library built to be run in the Python computing environment[72].

1.2.1 Data Preparation

For many models, raw data can lead to sub-optimal performance. Some models explicitly expect a Gaussian distribution of *feature* values, and even those which do not may place more importance on *features* which have a higher range of values and/or larger numeric values [36, 42]. Two common methods used to prepare data are *Normalization* and *Standardization*. Normalization, also known as min-max scaling, normalizes data to a range between 0 and 1 by using the following equation [49]:

$$X_{normalized} = \frac{X_{feature,sample} - X_{min,feature}}{X_{max,feature} - X_{min,feature}} \tag{1.1}$$

In this equation $X_{feature,sample}$ is the value of a feature within a given sample, $X_{min,feature}$ is the minimum recorded value for that *feature* within the available samples, and $X_{max,feature}$ is the maximum observed value for that *feature* in the available samples.

Many machine learning protocols assume a Gaussian distribution of values in

the *features* they use for prediction. Not all real world data actually follows Gaussian distribution. Standardization (also known Standard Scalar) corrects this using the following formula [73]:

$$X_{standardized} = \frac{X_{feature,sample} - \mu_{feature}}{\sigma_{feature}} \tag{1.2}$$

In this formula $\sigma_{feature}$ is the standard deviation of the feature as observed from available samples, and $\mu_{feature}$ is the mean of the feature as observed in available samples.

1.2.2 Machine Learning Protocols

An overview of the ML algorithms used in this work are presented in the following subsections. The 'no free lunch' theorem indicates that there is no universal model suitable for all tasks and datasets. Generally within papers reviewed, researchers compared several models prior to selecting one that performed best according to their requirements, and that approach was also followed in this work.

For each model that was considered in this work, an overview of how it works, whether it can perform classification and/or regression model, its strengths, its weaknesses, and its time complexity will be presented. A full discussion of Time Complexity (*O()*) is outside the scope of this work. In essence, *O()* is a descriptor of how long a model takes to run, and how much computing power is needed to run a model. It can be useful in selecting between models in platforms where computational power may be a limitation of the system [72].

1.2.2.1 K-Nearest Neighbor

K-Nearest Neighbor (KNN) is one of the simplest models conceptually. KNN models can perform both regression and classification. They work on the principle of 'Feature Similarity'; the k points that have *feature* values closest to that of the sample being classified are polled [36]. Whatever value the majority of them have is assumed to be the value the sample should have. This has the advantage of being very good at classifying data sets which have a complex boundary, and making sense intuitively [16]. The drawbacks include the fact that the performance of KNN declines as the number of samples increases because it has no mechanism to ignore irrelevant features [16]. The time complexity can be modeled as $O(n_{(features)} * n_{(samples)} * min\ (n_{(features)},\ n_{(samples)})$ [72][33].

1.2.2.2 Random Forest

Random forest (RF) models use *ensembles*. This means that instead of a single model, many models are constructed and then each one is used to generate a prediction. When a new sample is received, each model makes a prediction, and the predictions are polled to determine the likely *target* value. Each decision tree uses some m of p available *features* to build its model. As implied by the name, each decision tree has multiple branches that split off potential end values until the tree reaches a likely *target* value [36]. While any given tree may not be a very strong predictor, many trees will generally settle on a reasonable approximation of a samples value. Advantages of these models include the fact that they work well

without regularization, and can give details about the importance of features [72]. Unfortunately these models can become very large, which puts significant demands on computing resources, and are prone to over training which can produce poor accuracy on real world data [25, 36].

1.2.2.3 Linear Discriminant Analysis

Linear Discriminant Analysis LDA is a classification (LDA) model. As implied by its name, it looks for a linear combination of *features* which best separate the *target* classes. It has the inherent assumptions that each class will have the same covariance, and a Gaussian distribution of data. If assumptions can be met, LDA is extremely accurate. It works by finding means vectors of the class and attempting to maximize distance between classes while minimizing the covariance of *features* between classes. Then Bayes Theorem is used to estimate the probability that a new sample encountered belongs to a given class. Features with large outliers can significantly lower its accuracy. LDA implementations normally use dense matrices and require eigen-decomposition which can be computationally challenging and lead to a fairly high time complexity. Its time complexity can be expressed as $O(n*m*t + t^3)$ where m is the number of samples, n is the number of features, and $t = min(m,n)$. [17, 36, 72].

1.2.2.4 Support Vector Machine

Support Vector Machines (SVM) are available in both Regression (SVR) and Classification (SVC) versions. They work by casting data onto a hyper-plane using only the training points which are closest to decision boundaries and attempts to maximize the center distance between categories. The points closest to or on the boundaries are known as support vectors and the space between is the margin. Because these are the only important features to the model, SVM can often use a surprisingly small number of samples to make a good model and handle large numbers of features by ignoring those which do not improve the model [16]. Support vectors from each class are maximized to increase the margin to its largest possible size [72]. While SVM is a nominally linear algorithm, it can accept different kernels. These kernels can cast data to a higher plane and allow otherwise inseparable data to be divided cleanly by support vectors. This allows it to find relationships that would be missed by LDA and SVC is popular in gesture recognition papers [40, 43, 66]. While the processor overhead of SVM is lower than LDA, it is still fairly high at $O(n*m^3)$ where n is the number of features and m is the number of samples [72].

1.2.2.5 Automatic Relevance Determination and Bayesian Ridge

Because of how similar they are and how little they appear in this work, both Automatic Relevance Determination (ARD) and Bayesian Ridge (BR) share a single listing. They are both regression models which are based off of Bayesian methods that develop a probabilistic distribution based on observed *features* rather

than looking for specific linear combination of *features*. In these models the starting probability distribution shape is assumed and each available sample is used to create posterior probability distributions either supporting or changing the shape of that probability function. Bayesian regression methods are good at adapting to data at hand, very good at using posterior probability to ignore redundant features, and can incorporate regularization into models [42, 87, 90]. The primary difference between BR and ARD is that BR assumes a spherical Gaussian as its prior distribution shape as opposed to ARD which assumes an elliptical distribution and has more flexibility regarding standard deviation of individual features [87, 90]. The time complexity of ARD is $O(n^3)$ and the time complexity of BR is $O(m*n^2)$ where n is the number of features and m is the number of samples [72, 75].

1.2.3 Training and Validation

While selecting a model is important, the model needs training and validation for practical application. A machine learning model has to be trained before it can be used. This work focuses on supervised learning models. This means that some quantity of *features* (in this work capacitive sensor readings) and matching *targets* are available. These features and targets are fed to an ML algorithm which attempts to build a model which if given a new set of *features* can predict what the *target* should be. In general models in this work are trained using test/train splits as discussed in Sec. 1.2.3.1.

Once trained, a model must be validated. How this validation is accomplished

is in part a function of how many datasets are available. In this work a dataset is defined as data collected from a single subject during a single session. If only a single dataset is available, then the only validation method available is n-fold validation as discussed in Sec.1.2.3.2. This validation method also can be used within the training data to determine if a model is *overtrained* as will be discussed in 1.2.3.4.

If multiple datasets are available, then leave-n-out validation schemes which are discussed in Sec.1.2.3.3 can be employed. These models are also useful for looking at the performance of inter-session (IS) and inter-day (ID) trials.

1.2.3.1 Train/Test Splits

In a Train/Test split some fraction of the data (tr) is used to train a model, and the remaining portion of the data *(1-tr)* is reserved for validation. For example in a train/test split of a 70/30, 70% of data would be used for training a model and the remaining data would be used for testing that model. In general machine learning applications, the rule of thumb is to use 70% of available data to train models. This is generally enough to account for variability in data without *overtraining* [72]. Overtraining occurs when a model is so tuned to the subset of data it trained on that it inadvertently builds a model on features which do not generalize well.

1.2.3.2 k-Fold-Validation

When only a single dataset is available, k-fold validation is the common method of validating the anticipated real world performance of the model. In k-fold vali-

dation some portion of the data is held out for testing. Then the remaining data available is split into k equally sized folds. In each cycle one group is held out for validation and some tr of the other k-1 are used to train a model which is validated on the held out data. A visual representation of a k-fold method is shown in Fig. 1.2a. A further refinement of this method shown in Fig. 1.2b. uses a random selec-

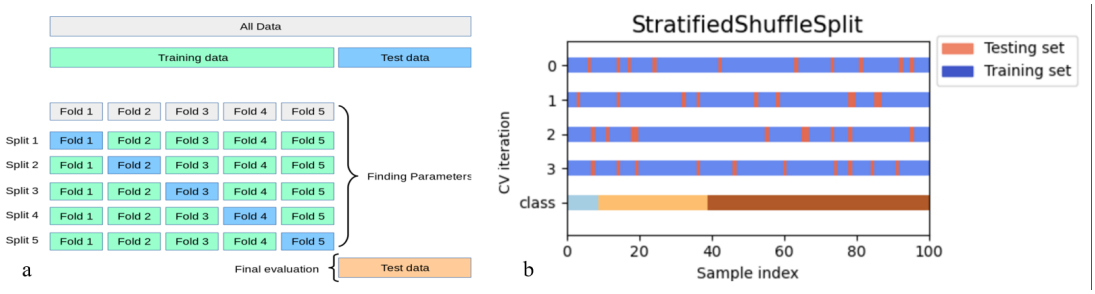


Figure 1.2: a) k-fold analysis b) stratified shuffle splits. Both images modified from. [72]

tion of data stratified by the relative representation of each class to generate a more accurate estimate of real world performance. This is fairly simply to implement using the built in tools in scikit-learn.

1.2.3.3 Leave-P-Out Validation

Leave-P-Out (LPO) validation schemes are used when more than one dataset is available for testing. In LPO there are some k datasets available. Some number of datasets j are selected to train models on. Some tr of these datasets are used to build a model. The remaining p=k-j models are reserved to validate the built model. This method is a more robust way to predict the performance of a model on

unseen data. For purposes of worn devices LPO testing has some additional benefits. For IT testing where visual examination does not show obvious user or electronics error, LPO results can be compared to the results of n-fold validation of the train set to help uncover instances of *overtraining* and instances where either the amount of data available for for training was inadequate or the model selected was a poor match. While poor model selection and insufficient data show the same pattern of

Condition	K-Fold Metric	Leave-P-Out Metric
Ideal	High	High
Overtraining	High	Low
Insufficient Data	Low	low
Poorly Selected Model	Low	Low

Table 1.1: Diagnosing with k-Fold and LPO.

errors, generally increasing the amount of data available for training by raising tr or increasing p will resolve insufficient data issues. Model algorithms which are a poor match for the type of data being presented will generally continue to display bad performance.

1.2.3.4 Metrics

The metrics used to validate an ML algorithm are in part a function of the type of algorithm. The most basic metric for classification is accuracy, which is defined as :

$$Accuracy = \frac{CorrectPredictions}{TotalPredictions} \tag{1.3}$$

While accuracy is a good starting point, it does not tell the whole story. There are two types of errors which can occur in classification. Type 1 error occurs when a model fails to recognize samples that belong in a class. Samples displaying Type 1 error are referred to as False Negatives (FN). Type 2 error occurs when a model places samples in a class they do not belong in and is known as a False Positive (FP).

If one considers a simple model to describe an open or closed hand, Type 1 error occurs when the model fails to recognize an open hand. Type 2 error occurs when the model classifies a closed hand as open. The metrics of Recall (Type1) and Precision (Type2) identify these types of errors [5]. The Precision metric describes the proportion of positive identifications in a class was correctly identified:

$$Precision = \frac{TP}{TP + FP} \tag{1.4}$$

Ideally this number would be 1, but increases in Type 2 error will lower it. Recall on the other hand describes the proportion of a given class which was captured by a model:

$$Recall = \frac{TP}{TP + FN} \tag{1.5}$$

If a model has significant Type1 error, it will fail to identify members of a class, the FN will rise and this value will be less than 1. Finally, F1 scores can be used to combine precision and recall into a single measure. They represent the harmonic

mean of Precision and Recall and take the form of:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{1.6}$$

While all these metrics are useful for describing single class problems, they get a bit more complicated when multiple classes exist. For multiclass models, the question of which classes a model has trouble with and whether there is a specific overlap of classes that generate most of the FP or FN becomes important. A common metric used for multiclass metrics is a confusion matrix (CM). The central diagonal of a confusion matrix is all the TP for a class. The FP is the sum of all values on the same row as a class which are not on the diagonal. The FN is the sum of all values in the same column as a class that are not in the central diagonal. An example of how a confusion matrix can be read is shown in Fig. 1.3a. Typically the rows

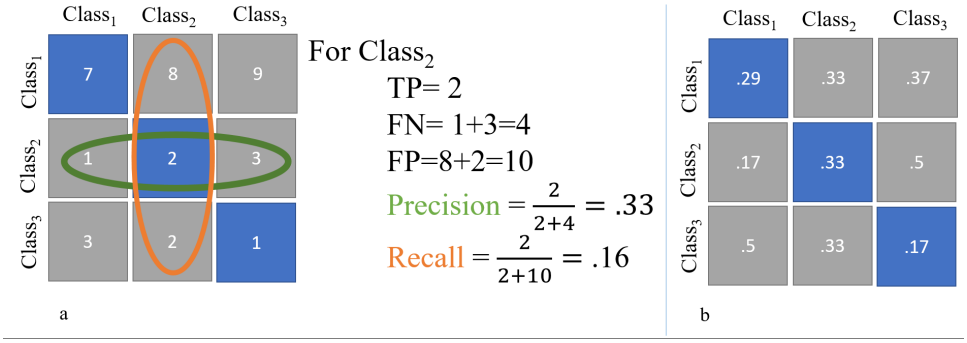


Figure 1.3: a)Interpreting a Confusion Matrix, b) A normalized confusion matrix.

of a confusion matrix are normalized by the number of samples in a given class, allowing the accuracy of classification to simply be read off the main diagonal. An

example of a normalized confusion matrix can be seen in Fig. 1.3b. In this thesis all classification models will use raw accuracy and when practical confusion matrices as their metrics.

Accuracy does not work well for regression problems due to the continuous nature of their output. To give an example, if the actual value being modeled is 4.0 a model output of 4.1 and 41 would both be penalized the same if accuracy was used as a metric. For this work, R^2 will be used as the regression metric. R^2 is a common metric for goodness of fit which explains how well the independent variables (*features*) describe variability in the dependent variable (*target*). Also known as the coefficient of correlation this metric can be expressed as: [65]

$$R^2 = \frac{RegressionSumOfSquares}{TotalSumOfSquares} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y})^2 - \sum_{i=1}^{n}(y_i - \hat{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1.7}$$

In this equation y_i is the actual value, $\bar{y}$ is the mean value, and $\hat{y}_i$ is the model's predicted value.

1.3 Gestures Used

There are a wide variety of sources cataloging potential hand motions. The Cutkostky taxonomy proposes a simplified list of grasps used in everyday life for use with robot hands [23]. Another classic method is the NinaPro database which attempted to codify gestures that may be of interest from a rehabilitation or a prosthetics perspective [7]. These databases suffer from having far more entries than most devices can reasonably accommodate, or than most studies can find

subjects willing to perform. The Cutkosky taxonomy has 16 grasps, and the NinaPro database has 52. In the case of the NinaPro database, 12 of these gestures are individual finger flexion and extension, and another 10 are motions of the wrist (i.e. pronation/supination, extension, flexion, ulnar/radial deviation) as distinct gestures.

This leads to two problems. The first is subject fatigue. It is not a trivial problem to find subjects willing to perform multiple data collection sessions composed of 52 different gestures. The second lies in the fact that many of the gestures are either redundant, or not actual gestures. For example, the individual finger motions, if accurately identified could be combined to recreate gestures.

Wrist motions are included in many studies, and I used them in my first set of experiments as well. However I came to realize that they are not independent gestures, but are instead additional degrees of freedom added within a hand gesture. For example one can make a fist with their wrist in a wide variety of positions depending on whether they are carrying groceries, swinging a hammer, or cutting food for dinner. Rather than being independent motions, the wrist positions are degrees of freedom superimposed over motions.

In reality for most practical uses, a much smaller set of gestures is required than those seen in full grasp taxonomies [38]. The typical goal in physical therapy and prosthetics is to accomplish activities of daily living (ADL), and the grasps which are needed to accomplish that vary significantly from one study to the next [18, 38, 53]. That being said, there were a few common gestures of that showed up in the majority of papers reviewed across fields including rehabilitation, prosthetics,

training, and VR interface. Most papers had some whole hand grasping gesture (closed fist), an index finger pinching gesture (pinch), a whole hand pinching gesture (pincer), an open hand (or neutral) gesture, a spread hand gesture, a single finger pointing gesture, and and a single finger gesture.

For my purposes I considered Closed Hand, Open Hand, Pinch, Pincer, Point, and Spread Hand to be top level gestures and included them all the studies performed for my thesis except the finger force analysis in Chap. 7. I also included a Thumbs Up gesture in Chaps. 4 and 5 because I wanted to determine if independent thumb motions could be tracked. As it turned out, thumb motions were a significant challenge for the capacitive strap sensors to pick up. For Chap. 6 the thumbs up gesture was dropped, but more single finger and multiple finger gestures were added to determine if they could be classified reliably.

1.4 Proposed Technology

While the technology proposed in this paper is discussed in far more depth in Chap. 2 and Chap. 3, the difference between the proposed architecture in this thesis and traditional methods of wHMI can be summed up as follows. Rather than using 'point' sensors which depend on both radial and angular positioning on the subjects arm, this technology uses straps that wrap all the way around the arm and measure the shape of the cross section. As such, rotation and minor shifts should effect the sensors far less than conventional technologies. Because this technology uses capacitance rather than pressure or EMG it is expected that the sensors will

also suffer less from signal drift and be very inexpensive to construct and implement with conventional off-the-shelf parts.

1.5 Outline of Thesis

In Chap. 2, the first foray into testing the 1D CapSense system on human subjects is discussed. This early iteration did not use inter session (*IS*) testing and while it showed promising results, some iterative hardware modifications discussed in Chap. 3 were required to develop the system used in the remainder of this thesis. Chap. 4 discusses the first attempts at *IS* gesture classification in between sessions on the same day, and in Chap. 5 this is expanded to look at the results of inter day *ID* testing. Chap. 6 adds in an additional DOF in the form of extension and flexion of the elbow, and expands the classification model used to correct for motion artifacts attributable to those motions. Chap. 7 presents the results of the first attempts at modelling whole hand and individual finger forces using both regressions and force bin classifications. Finally Chap. 8 recaps the results and outcomes, and gives some directions for continuing work on this device.

Chapter 2: Gesture Recognition via Flexible Capacitive Touch Electrodes

2.1 Introduction

Exoskeletons and wearable robotics are emerging as a technology capable of enhancing the lives of the injured and elderly. They offer an attractive method of providing mobility assistance, stroke rehabilitation, and general physical therapy. While physical actuation has improved over recent years, there is still significant room for growth in systems for recognizing user intent and, specifically for this work, gesture and motion recognition [52, 74, 94].

For wide adoption, gesture recognition systems must be robust, low cost, accurate, easy to implement, and comfortable enough for daily wear. The current landscape for gesture recognition has several competing technologies attempting to fill this niche including surface electromyography (sEMG), force myography (FMG), and electrical impedance tomography (EIT). All of these technologies have some limitations within these metrics.

sEMG uses small electrodes to monitor the electrical signal of muscle neurons which control the muscles. While this is the clinical gold standard, the technol-

ogy faces many challenges for widespread use outside the lab. Electrode pads are expensive, and obtaining clean signals requires significant preparation of skin and very precise placement. Even under ideal scenarios, significant signal processing is required. sEMG also suffers signal drift over course of the day, picks up environmental noise from electronics as well as neighboring muscle groups, and sweat can alter electrical signals [26, 40, 79].

FMG measures radial pressure of muscles as they contract and has been used with machine learning algorithms to achieve promising results in recent years. However, it is dependent on many as 32 discrete sensors being placed over thickest portion of forearm. This has several drawbacks including cost, the requirement of sensors to be worn directly against users' skin, and the potential for model accuracy to be diminished as sensors slip to different positions during normal motion [40, 43].

EIT has also seen some use in gesture recognition. In this context a bracelet carrying an array of electrodes is worn on the arm and a signal is sent out by one of the electrodes. The other electrodes read the signal and compare it to the input to determine impedance. While this method can be highly accurate, the authors indicate that it is highly dependent on placing sensor band at a specific location on the arm and requires direct skin contact [91].

This work presents a novel application of capacitive touch (CT) as an alternative method to sense recognize and classify human motion. The design we focus on uses continuous (versus many discrete) electrodes wrapped around the circumference of the arm to spatially integrate signals from a large number of muscles. The wristband format used is easy to don and comfortable enough for long term wear.

Because CT measures proximity rather than direct pressure it also has the potential to be worn over clothing or bandages.

The primary contribution of this work is an exploration of CT using these circumferential electrodes for gesture recognition. To accomplish this, an armband was developed and experiments were conducted to explore accuracy of the system in recognizing a variety of grasps and individual finger and wrist motions. Data collected from subjects was processed using several common and well documented machine learning models.

2.2 Principle of Operation

Capacitive touch is a mature technology that has been used in a wide array of touch interfaces and proximity sensing devices. It is robust, low cost, and non-invasive, but the goal of this work was to determine both comfort and accuracy for gesture recognition.

An excellent review detailing taxonomy of capacitive sensing and various methods used to interface humans with machines via CT was compiled by Grosse-Puppendahl et. al. in 2017 [34]. Capacitance in its most basic form is a measure of energy stored in an electric field between a positive and negative electrode. This is described by the equation:

$$C = \epsilon \frac{A}{d} \tag{2.1}$$

In this equation ϵ represents the dielectric constant of the material separating the electrodes of overlap area A and d is the distance between the electrodes. Ca-

pacitance changes proportionally to ϵ and A, and inversely proportional to d. CT technologies measure how capacitance changes as a function of the human subject interacting with the electrodes, but can be complex since all of these variables can be changing at once.

Early investigations into CT as a means of gesture recognition focused on the use of a dual electrode active CT architecture (Fig. 2.1). This approach is similar to EIT in that it excites an electrode attached to the body and captures data from ground electrodes elsewhere on the body. Changes in capacitance were a function of both changes in distance between electrodes, and changes in dielectric value of the human body due to muscle activity [21, 34, 98]. While these methods can be very accurate, they are very sensitive to the position of the sensors, can be prone to environmental noise, and have a relatively high computational overhead.

In more recent research, a simpler single electrode architecture was adopted (Fig. 2.1B) in which the human body was treated as negative electrode, and capacitance changed as a function of distance between body and positive electrode. Zheng et al looked at controlling prosthetics with many pads placed directly over specific muscle groups; they achieved good results, but pads had to be mounted on a rigid surface and required placement over specific muscles of interest [99].

Other uses of CT to classify gestures or motion include the use of a positively charged ring on one finger to measure changes in capacitance caused by motion of neighboring fingers [88]. These results showed good accuracy but limited utility to finger-based gestures as they could only determine relative position of neighboring fingers. Baldwin et al used an electrode worn on one leg to count strides and

determine distance between legs [10]. Capacitance was changed by the proximity of the neighboring leg. While effective for these purposes the requirement of measuring an external limb prevented shielding electrodes from external signals, and detailed information about muscle use or joint positions was not available [10].

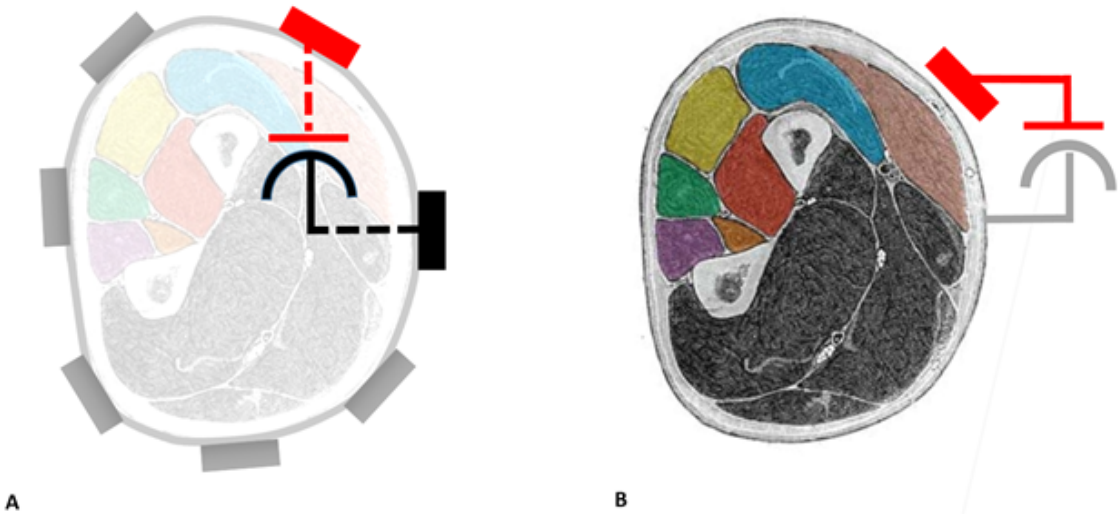


Figure 2.1: Previous approaches using CT for gesture recognition. A) Dual electrode methods use the human body as a variable dielectric. B) Single electrode methods treat the human as a negative electrode and measure change in capacitance as a function of skin proximity to the positive electrode. Arm cross-sections adapted from [41].

This paper proposes a simple and low-cost approach to using CT for classifying gestures. In order to move or make gestures, many muscles must be contracted or relaxed in concert. The capacitance measured by a flexible electrode wrapped around the circumference of a limb will therefore change as the total amount of skin at a given distance from the electrode changes with each grasp (Fig. 2.2). The combination of muscles contracting and relaxing should produce a specific capacitive signature for each individual gesture. The hypothesis for this work is that the combined data from several electrodes wrapped around the limb will allow for

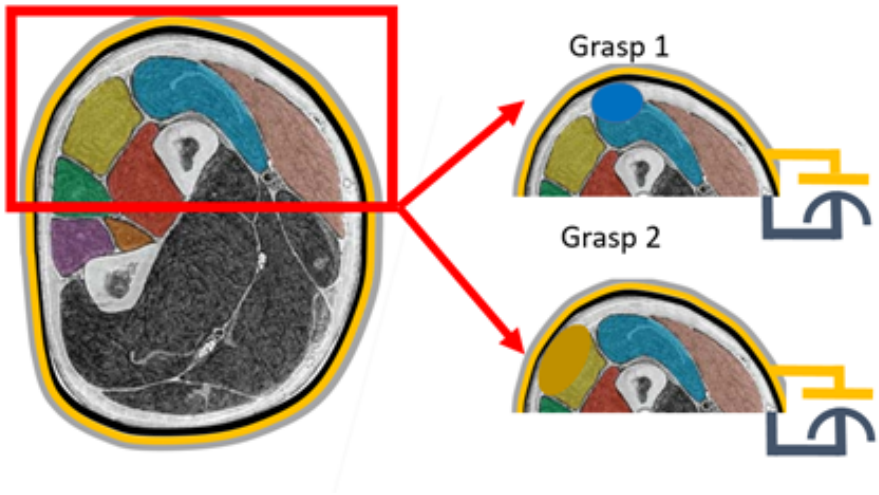


Figure 2.2: The proposed electrode architecture encircles the entire arm. As a result it can spatially integrate the activity of multiple muscles at the same time, and generates a change in capacitance as a function of their combined contraction and relaxation. Two generic grasps are illustrated. In Grasp 1, the top muscle shown in blue is contracted and the left muscle shown in yellow relaxes. In Grasp 2 the top blue muscle relaxes, and the left yellow muscle contracts. Because this changes the distance between the skin and sensor strap, the two grasps result in different capacitive 'profiles'. Arm cross-section adapted from [41].

accurate gesture recognition. Wrapping the sensors entirely around the limb should mitigate the effects of motion artifacts and eliminate need to place sensors precisely over individual muscle groups. Machine learning will be used to help classify unique capacitive signatures as different grasps. Given the physical simplicity of the approach as well as computational simplicity, this technique should allow for gesture recognition at a far lower cost in terms of parts and computational power than existing technologies.

2.3 Materials and Methods

A wearable armband (Fig. 2.3) was constructed to assess the accuracy of CT sensing in a series of grasping experiments described here. It should be noted that the armband was designed to be placed only on the forearm of each user with the goal of classifying grasps and gestures through muscle activation in the forearm only.

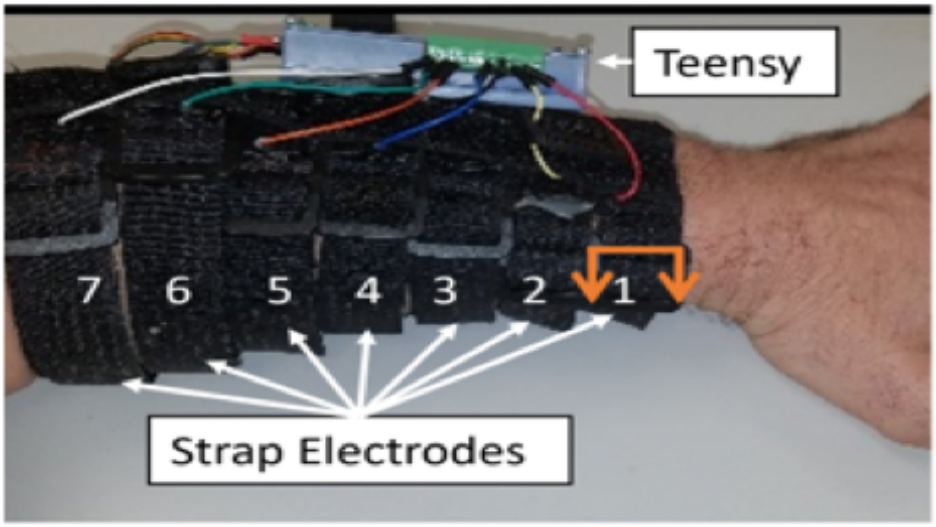


Figure 2.3: The armband includes 7 capacitive touch electrode straps connected to a Teensy 3.2 and Bluetooth LTE radio. The armband is designed to fit over the subject's forearm and strap 1 is aligned with the knob of ulna bone on wrist to provide an anatomical landmark.

2.3.1 Armband

The electrodes making up the positive electrodes of the capacitive sensor need to be conductive, flexible enough to wrap around the arm, and adaptable and comfortable for human subjects of varying sizes. While several approaches were tested including conductive fabric and conductive elastomers, the simplest and most robust approach used copper tape. To allow for adjustment of straps to fit various human

subjects, flexible electrodes were constructed by applying 6.3 mm wide copper tape to 19 mm wide snag-free Velcro tape that allowed each electrode to be tightened and fixed into place. The copper electrodes were sandwiched on the opposite side by nylon straps that were sewed to the velcro and used to create an additional dielectric barrier between the electrode and the human subject.

Seven of these electrodes were arrayed by sewing them to a piece of fabric that was used to adhere the on-board electronics. Spacing between electrodes was designed to be approximately equal at 22 mm apart. A Teensy 3.2 microcontroller board was selected due to its low energy consumption, 16-bit ADC resolution, and the presence of 12 integrated pins capable of capacitive measurement via the dual oscillator method. In this armband prototype, wires were soldered to the copper straps and connected to a header on a custom printed circuit board that also included the Teensy. In order to track arm angle for future work, a BNO055 IMU was also connected to the microcontroller. To allow for wireless communication between the armband and a computer (or phone), an HC-05 Bluetooth module was attached and used as a serial connection.

A companion application was built in Labview to collect data from the armband in real time at 20 Hz. This application also indicated the gesture that a subject should perform and enabled the authors to manually indicate when grasp events had occurred and ended using a button for labelling purposes.

2.3.2 Experiment

For initial proof of concept testing, five volunteers containing three men and two women were recruited to test the ability of CT to distinguish between 32 selected gestures (Fig. 2.4). 16 of these gestures were made up of grasps in the Cutkosky Grasp Taxonomy (CGT) [23]. This set was selected because it contains most grasps used in daily life and is relevant to both robotic grasps and physical therapy. An additional 16 motions were selected as representative of the typical range of motion for the hand and wrist. These motions included: flexion and extension of wrist, ulnar and radial deviation of wrist, supination and pronation of the hand, and individual flexion and hyperextension of each finger.

The sensor system was placed on a subject's arm, and the lowest strap was aligned to lie just behind the head of ulna to provide a constant anatomical landmark (Fig. 2.3). Buckles for the velcro straps were aligned with the radius bone on the inner arm. Subjects were then seated at a desk and placed their arms in a rig equipped with an elbow rest to help the subject maintain position, and sliding stage to allow the various objects grasped in experiment to be moved to a comfortable distance from subject (Fig. 2.5).

When cued from the software, subjects would grasp an object or make the requested hand motion, press button to indicate their hand was in position, hold the position for 3-5 seconds, release button to indicate they were done performing gesture, and return their hand to a neutral position. For each grasp or gesture in Fig. 2.4, five repetitions were obtained giving 300-500 discrete samples. This

experiment was performed once on each subject.

2.3.3 Analysis

Saved data was processed offline using Anaconda Python 3.6 with the scikit-learn library. A sample of the raw data collected for several gestures can be see in (Fig 2.6). Since the nominal capacitance for each electrode is different, the capacitance data was normalized to values between -1 and 1 and different machine learning algorithms were applied.

The goal of machine learning is to develop models relating the gestures (categories) to features (individual capacitive readings from the strap electrodes). Because the specific gestures being made were known through manual labeling, a selection of supervised machine learning algorithms detailed below were used. The particular features of these experiments important to consider when selecting algorithms are the large number of categories, likely nonlinearity of the capacitance data, and practicality toward a future embedded classification solution.

Linear Discriminant Analysis (LDA) is good at categorization problems with linear data. It also has a very low overhead which may make it suitable for embedded systems. LDA implicitly assumes a Gaussian density to each category, and that all categories share the same co-variance matrix. The sample data provided to the algorithm is used to find a line which can split between categories so as to maximize the ratio of between class variance vs out of class variance [72]. While effective for systems which are linear separable, it performs poorly on nonlinear data. For LDA,

the default scikit-learn settings were used.

Under the assumption that data may not be linear, the Support Vector Machine (SVM) algorithm with a radial basis function (RBF) kernel was also tested. SVM casts data onto a hyper-plane using only the training points which are closest to decision boundaries and attempts to maximize the center distance between categories. The points closest to, or on, the boundaries are known as support vectors and the space between is the margin. Support vectors from each class are maximized to increase the margin to its largest possible size [72]. Because it is not constrained to linear relationships, it can find relationships that would be missed by LDA and has been used in a number of gesture recognition papers [40, 66]. The drawback of SVM is that it has a high processor overhead, which makes it less suitable for embedded systems. For SVM the commonly accepted default values of a C value equal to number of sensors and a γ of $1/C$ were used in testing.

The K-nearest neighbor (KNN) algorithm was utilized to determine whether cluster based methods would produce a better model versus linear or hyper-plane separations. KNN works by finding the mode value for each individual class based on finding nearest neighbors of each point in a training set. It works well when there are a reasonably small number of features and its models are easy to update over time with new information [72]. However, because it holds all data in direct memory, it can be very computationally expensive. The default setting of five nearest neighbors was used for this paper.

In order to investigate relative importance of each electrode feature, a Random Forest (RF) algorithm was employed. RF is an ensemble method which combines

bagging and random feature selection. It operates by growing many decision trees in parallel. Each tree is built on a random selection of observations created through bootstrap sampling of the data and at each node the tree is split using a random selection of features with the goal of maximizing similarity of categories within the resulting subsets of data. This is repeated until a stopping condition is reached. A prediction phase is then entered where the same test data is presented to each tree. At each end leaf node of the tree, the most common category in the node is calculated. The predictions of all trees are then combined via a voting algorithm to provide a final selection model [29]. Advantages of the RF algorithms include the fact that they work for both classification and regression problems, have fairly low processor overhead, and that the analysis of many feature combinations in parallel allows for determination of relative importance of individual features. A downside is that RF has difficulty integrating new data, but researchers are currently investigating methods to overcome that limitation [25]. An ensemble of 250 trees was used in this paper's analysis.

When models trained on a given set of data are exposed to real world data they were not trained on, out-of-sample errors may occur. The 10-fold validation method is a common way of validating machine learning models against 'unknown' data.[39]. In 10-fold models, all samples from a single class are combined and then randomly shuffled. In this case, all data from the five iterations of a single grasp or gesture for each subject were combined before splitting into ten stratified samples. Stratified means that within each sample, the number of members in a given class is proportional to the overall data set. For each iteration, 9 of the stratified samples

are used to teach a machine learning model, and the 10th is used to test the model. This is then repeated for 9 more iterations until each subset has been used once as a test model. The mean of prediction accuracy and variance are calculated. These values represent the expected performance of the algorithm on real world data.

Each of these algorithms was applied to the collected data for each subject. To test intra-subject accuracy grouped fold analysis was applied. In grouped fold analysis, one subject is held out to test models formed using the data from remaining subjects.

2.4 Results and Discussion

2.4.1 Prediction accuracy of different algorithms

The average accuracy versus algorithm for the 10-fold validation tests are shown in Fig. 2.7 . For all subjects tested, RF proved the most accurate algorithm, and the KNN model came in as a very close second. Given the large number of categories, this is not a surprise. As mentioned previously, instead of looking at boundaries between categories or distances between them, the randomness of RF does a good job with data that does not have clear linear or polynomial boundaries.

2.4.2 Individual subject Accuracies

For purposes of demonstrating predictive capabilities, confusion matrices were built using the RF model. In confusion matrices, the x-axis represents the predicted class, and y-axis represents the actual class. The diagonal down the center represents

correct predictions, and the intersecting squares give data about what proportion of incorrect predictions fall into each class. The model generated by Random Forest was then used to build a confusion matrix for each subject showing which gestures the model had difficulty in predicting. These matrices can be seen in Fig. 2.8. The only one to show any significant error was Subject 4, whose plot has been expanded for easier visualization.

In general, RF was able to generate accurate predictive results for grasps used. However, there are steep hurdles to be faced when developing a model for human subjects. It is very rare for humans to gesture in precisely the same manner repeatedly unless it is a gesture that they have been trained to do. Indeed, even though the on-board 9 DOF IMU was not used in the models, its data indicated that in some grasps, the angle at which objects were grabbed varied significantly between recorded instances, which was reflected by differences observed in the capacitive readings. It is possible that an additional 9 DOF IMU mounted on the hand could be used to examine differences in hand position as related to forearm and correct for or provide more accurate measurement of grasp and position.

Aside from variability of grasps within a test, some subjects had specific patterns of misclassification which could be related to their individual attributes. For subject 2, there was slight difficulty in interpreting grasps G13, G14, and G15 (fingertip on puck, palm on puck, and palm on sphere respectively). It is possible that the subject's hand size in relation to that of the puck and sphere was such that the grasps used were very similar. For subject 3, it was noted that some confusion existed between F7 and F8. The subject indicated that they had difficulty indepen-

dently hyperextending their middle finger without moving index finger. In future tests, finger hyperextension are done from a closed fist may prove easier for subjects and present a more accurate signal.

The one subject for which significant confusion was seen was Subject 4. Of the subjects tested, they had the smallest circumference at all points along the arm, and in order to obtain a snug fit, many of the electrode straps had to be pulled tight enough to overlap themselves. This may have led to interference in obtaining a clean signal. It is possible that future work with this individual using shorter straps could yield a more accurate result. They also had the shortest forearm, which could indicate that strap placement may need to be based on percentage of overall arm length rather than the fixed spacing used in this study.

2.4.3 Intra-subject accuracy

Universal modelling between subjects proved to be very inaccurate with SVM generating the highest accuracy at 12%. It was noted however that grasps from the Cutkosky Grasp Taxonomy had a higher accuracy than total set of data. This is not entirely a surprise as other technologies such as IMU [35] and EMG [11] have also had difficulty in identifying grasps and gestures between individuals. This other work has focused on motion primitives as a possible way to overcome the limitations of subject-specific grasp models. To evaluate why difficulty occurred in finding a global model, the importance assigned to individual sensor straps in each subject by the RF algorithm was investigated. The importance assigned to each strap is

shown in Table 2.1.

Subject	Straps listed in order of importance						
	1	2	3	4	5	6	7
1	5	7	4	6	3	2	1
2	1	6	2	3	7	5	4
3	4	2	7	3	1	6	5
4	2	3	1	5	6	7	4
5	1	2	6	7	3	5	4

Table 2.1: Sensor strap feature importance in model by subject

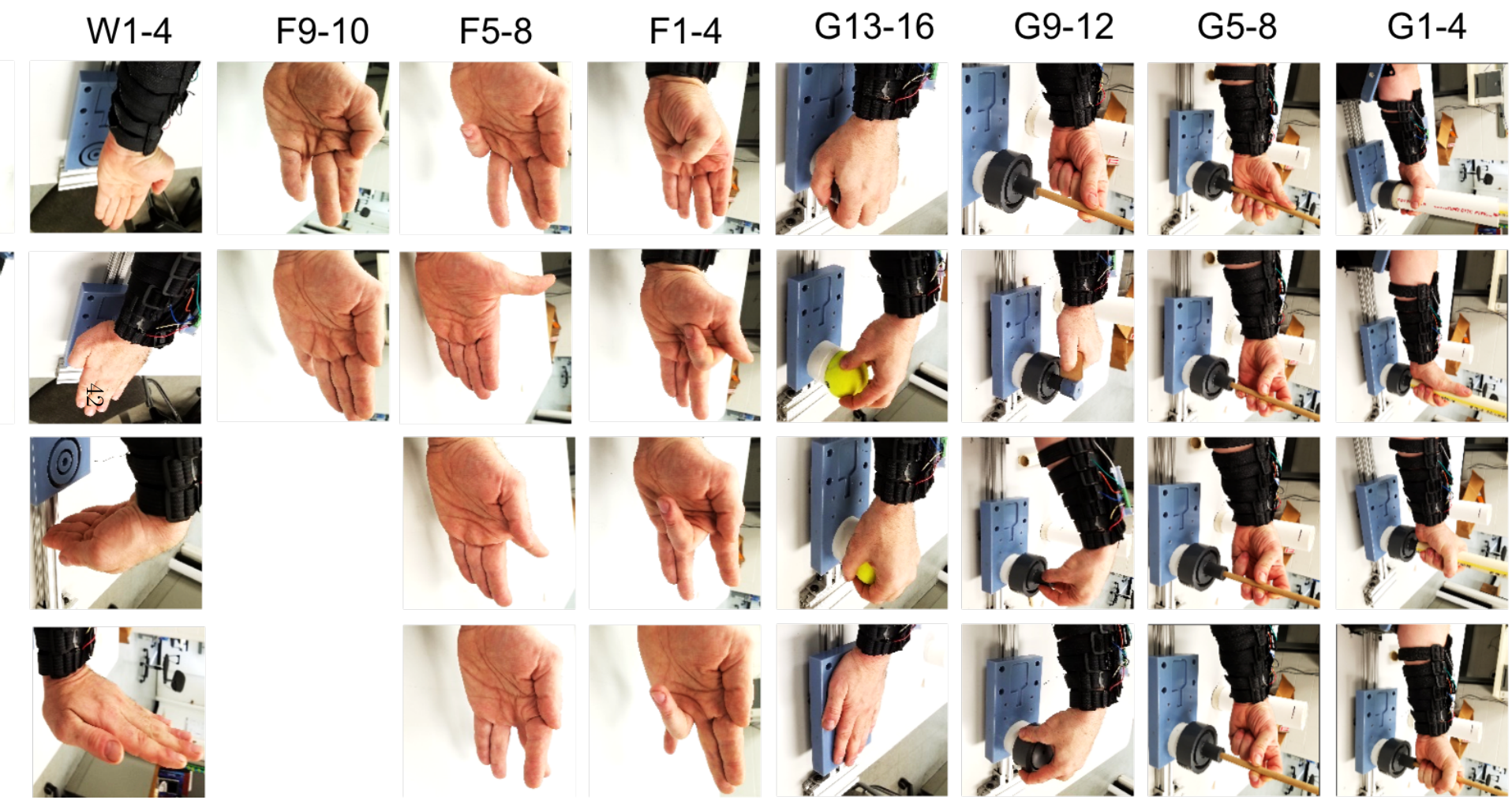
There does not seem to be any specific pattern regarding which sensor straps are most important in predicting grasps. Two possible explanations for this include: a) the different subjects used their muscles in very different manners making it difficult to translate models between subjects, and b) the sensor straps were tightened to a point which was deemed qualitatively comfortable by individual subjects resulting in variability between individual straps on subjects. One possible way to verify whether the underlying difference was due to strap tightness or muscle utilization would be to do two separate trials per subject on different days and evaluate similarity of models.

2.5 Conclusions

A proof of concept for a novel method of performing gesture recognition using CT has been created and demonstrated to have high ($95.6 \pm 0.06\%$) accuracy for individual subjects within a range of 16 grasps and 16 individual joint motions. The armband is simple, low cost, easy to use, and robust. This initial study demonstrates

accuracy comparable with recent studies using conventional technologies such as FMG ($94.2 \pm 3.0\%$) and EMG ($84.1 \pm 8.6\%$ accuracy) [40].

While the device has performed well in categorizing end points of gestures, it remains to be seen whether real time analysis is within its capabilities. The protocol in this experiment involved manually labelling data recorded while holding a static gesture, and is unlikely to answer this question. Future studies using the IMU to determine relative hand and arm positions and buttons to automatically record finger contact would allow a more robust investigation of gesture recognition in a a continuous data set, and could allow for joint angles to be calculated using linear regression methods. Other things of interest in the near future include determining whether results of trials on an individual subject remain consistent between trials, and whether models constructed from one arm of a healthy individual can accurately predict gesture on the other arm. If this is the case, then it may be possible to use the healthy arm as a trainer for a therapeutic robotics and assistive devices as it would indicate specific differences between motion in healthy and impaired limbs. Testing on a variety of different sized subjects could shed light as to whether fixed spacing is viable, or whether sensor spacing needs to be a function of subjects arm size.


W1-4
F9-10
F5-8
F1-4
G13-16
G9-12
G5-8
G1-4

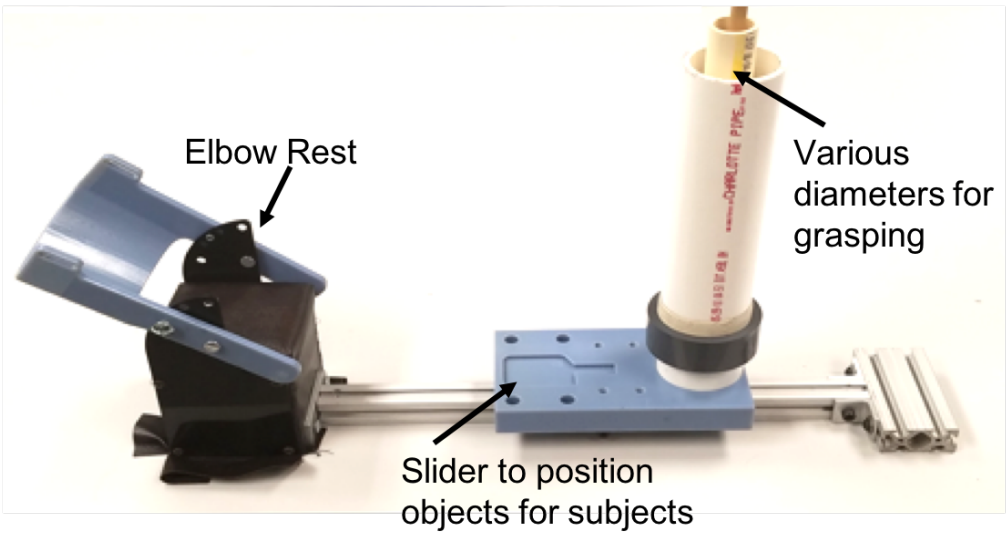


Figure 2.5: Test rig used for consistent grasps between subjects.

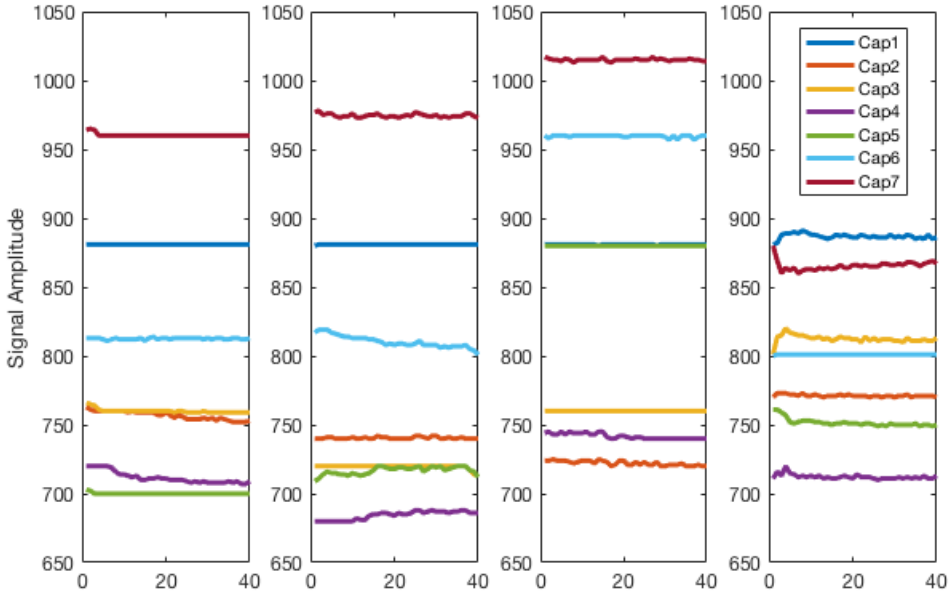


Figure 2.6: Raw data from selected grasps taken from Subject 1.

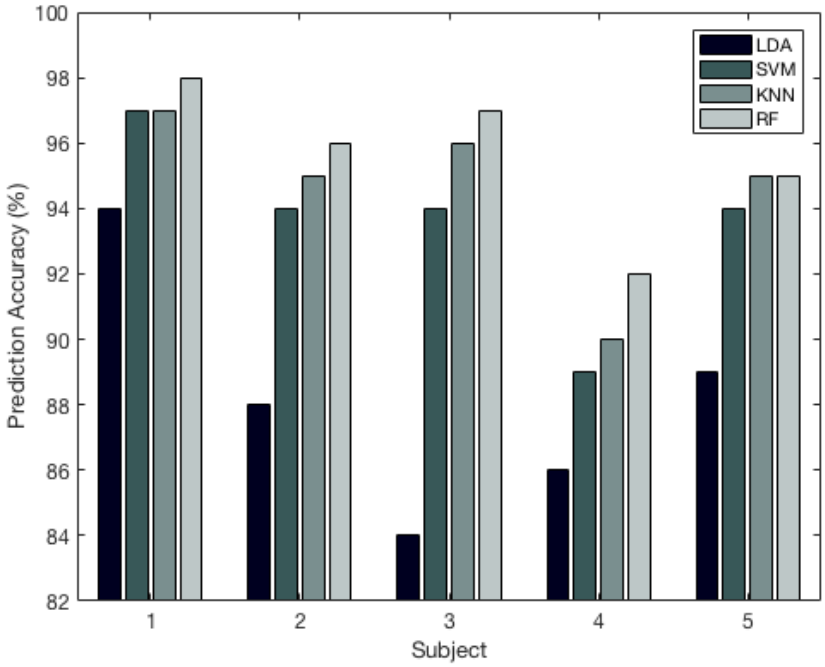


Figure 2.7: Prediction accuracy for each subject as a function of algorithm.

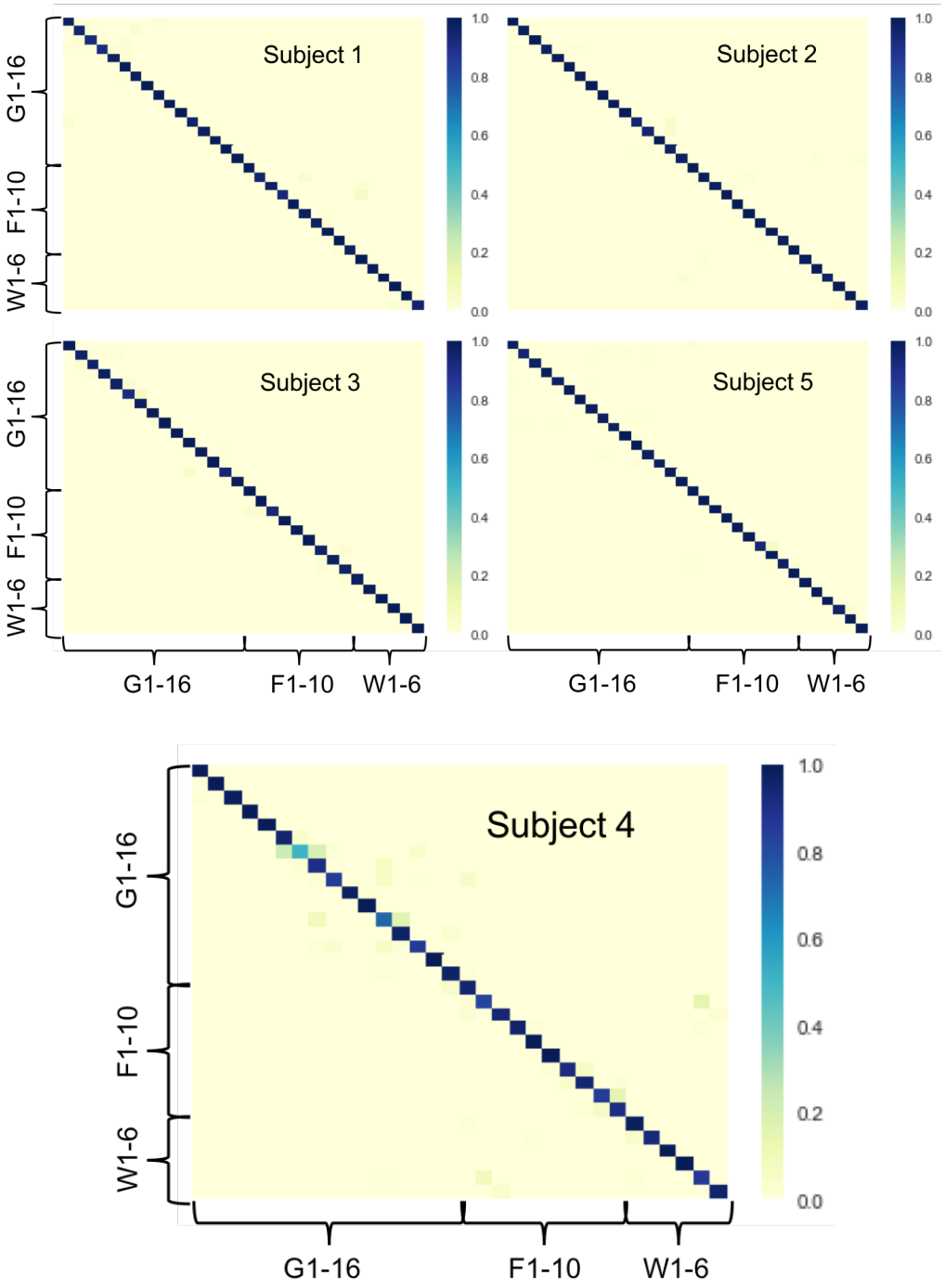


Figure 2.8: (a) Confusion matrices for subjects 1-3 and 5. These subjects all had very high prediction accuracies. (b) Close-up of confusion matrix for subject 4. This subject had poorer accuracy as seen in grasps, finger motions, and wrist motions.

Chapter 3: CapSense Sensor Design

3.1 Overview

Chapter 2 established the potential of flexible capacitive touch electrodes (from here on, titled CapSense) as a wHMI for gesture recognition. Using an Extra Trees Forest model with no preprocessing of the data on data collected from five users and ten-fold cross validation yielded an average accuracy of 95.6% over 32 tested static gestures made by seated users with their arms fixed in position by a rig.

In order to explore real world use inter-trail (IT) tests were initially performed. In these tests multiple data sets were collected from a single subject, and models trained on one data set were used to predict gestures on the other data sets from the same user. It was found that the IT accuracy was on average in the low 30% range.

A number of potential fixes for this were considered including changing the machine learning models used, introducing shielding in order to eliminate environmental noise as a source of error, and adding additional sensors to provide for more inputs to the models.

To test these fixes and further validate this novel sensing technology for gesture recognition and measuring grasp force while maintaining comfort and cost for a

wearable device, it was important to further explore the hardware design space, particularly around materials and electronics. Given the goal of improving the accessibility of this sensing technology, it was important to focus on a design that simplified construction, was easy for users to don, maintained position over the course of a days use, and could be constructed from comfortable easily accessible materials. Sec. 3.2 discusses the decisions made around that design and gives a description of the device used in the remainder of the studies.

The various configurations from Sec. 3.2 were validated using the experimental protocol discussed in Sec. 3.3. To facilitate the significant number of tests required for this experiment a more limited set of gestures were selected as discussed in Sec. 3.3.1. Because it has been noted that the models used in Chap. 2 lost reliability when used outside of the data set they were trained on, a new protocol described in Sec. 3.3.3 was adopted to validate the hardware. The results of the experiments and the hardware configuration selected going forward are discussed in Sec. 3.4

3.2 Design

The physical design of the CapSense is composed of three layers i) a bottom dielectric, ii) conductive sensing layer, and iii) an external dielectric. Sensors using shielding also have iv) a dielectric spacer, and v) a conductive shielding layer.

The original sensors used in Chap. 2 were constructed on a bottom dielectric of 15.87 mm nylon webbing (Joanne 1890086) with a conductive sensor layer of 6.35 mm copper foil tape (Selizo CFT-02), and an external dielectric of No-Snag

Sew On Velcro tape (Velcro 90666). While updating the design, new materials were considered both to replace the dielectric and to simplify construction of the base layer that the sensors were layered over. Given that in the US 1 in 5 adults with mobility issues live below the poverty line and half of users pay for mobility assistance devices entirely out of their own pocket or with help from their family [55], it was important to select reasonably inexpensive and readily available off the shelf materials. Section 3.2.1 looks at the materials that were explored.

On the original CapSense from Chap. 2 each sensor was connected to the electronic components of the system by a wire soldered directly to the conductive tape. The sensor wiring often failed mechanically and silently at this point copper tape and tin solder are maladapted to human test subjects. Soldering wire directly to copper strips created a brittle point that would fail after short periods of use, and the copper tape used in construction was prone to developing stress fractures and cracking over time. These internal cracks were difficult to detect before they became a sudden source of noise/error and there was no means of ensuring that placement was consistent between trials. Two alternate methods of connecting the wires were explored as means of creating more robust connections as discussed in section 3.2.2

Electronic design is discussed in Sec. 3.2.3. There were two proposed modifications to the CapSense PCB. The first was intended to test whether creating a barrier to environmental electrical noise would increase accuracy and repeatability of tests. In order to do so, versions of the PCB were created that allowed on board shielding as an option. The logic of this shielding is discussed in section 3.2.3 and the implementation of this shielding is discussed in Sec. 3.2.4. The other change

tested on the electronics was adding in a multiplexer to allow for more capacitive sensors to be connected than the native hardware allows as discussed in Sec. 3.2.3

Finally how to actually put all the parts together is addresssed in Sec. 3.2.4. On the original CapSense, the individual sensors were aligned by a rectangular piece of fabric with channels sewn that the sensors were inserted through. The tightness of sensors on the arm was adjusted by plastic buckles (Shapenty B077GLCCC9), and held at selected tightness by the self adhesion of the velcro exterior layer. While this insured that the tightness of straps would be consistent between uses, it did not insure that relative position of sensors or electronic components relative to the wrist or elbow would stay constant. Nor did it allow adjustment of the spacing of the straps or precise control over that spacing.

3.2.1 Materials Selected

The electrodes in the CapSense are composed of two primary materials the electrode itself and the electrode sheathing material. The sheathing material is particularly important because it provides both a dielectric between the electrode and the skin that can affect sensitivity as well as a comfortable interface to the skin.

The two materials chosen as candidates for this sheathing material were an 85% polyester blend 1.2 mm thick craft felt (Kunin 10000393)and a 2.8 mm thick perforated neoprene(FoamOrder Grade 1 perforated). The craft felt was chosen because it was inexpensive, relatively comfortable, and stiff enough to not allow sensors to collapse along the length of arm. The neoprene material was selected

because it is commonly used in both sportswear and medical braces. The specific neoprene selected is composed of three layers as shown in fig; a thin nylon top layer and a Velcro loop layer sandwich a neoprene layer perforated with air holes to allow the wearers skin to breathe.While this is not an exhaustive list of materials that could be used, these particular materials were chosen due to their ready availability.

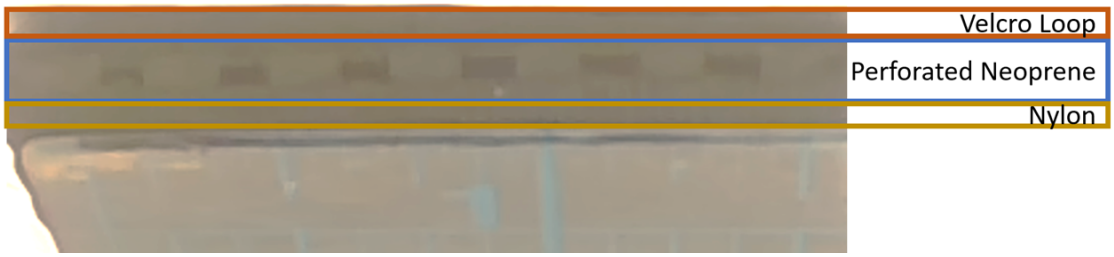


Figure 3.1: The neoprene fabric used in these experiments.

Given the success achieved with copper tape electrodes in Chap 2, the decision was made to use conductive fabric tape for the electrodes. This material is often used in computers for grounding electronic components. The tape is inexpensive, widely available, does not become brittle upon repeated bending, has conductivity similar to copper, and the double sided adhesive simplified assembly of sensors. In order to allow for testing of the connection methods discussed in 3.2.2 10 mm double-sided conductive tape (MEC CSTK -010) was used for the conductive sensing layer.

Both shielded and unshielded electrodes were tested in this chapter. For the shielded units, a layer of 22.2 mm fabric ribbon was used as a dielectric spacer layer with a second 16.0 mm layer of conductive tape (MEC CT-015) 3.2 mm longer than

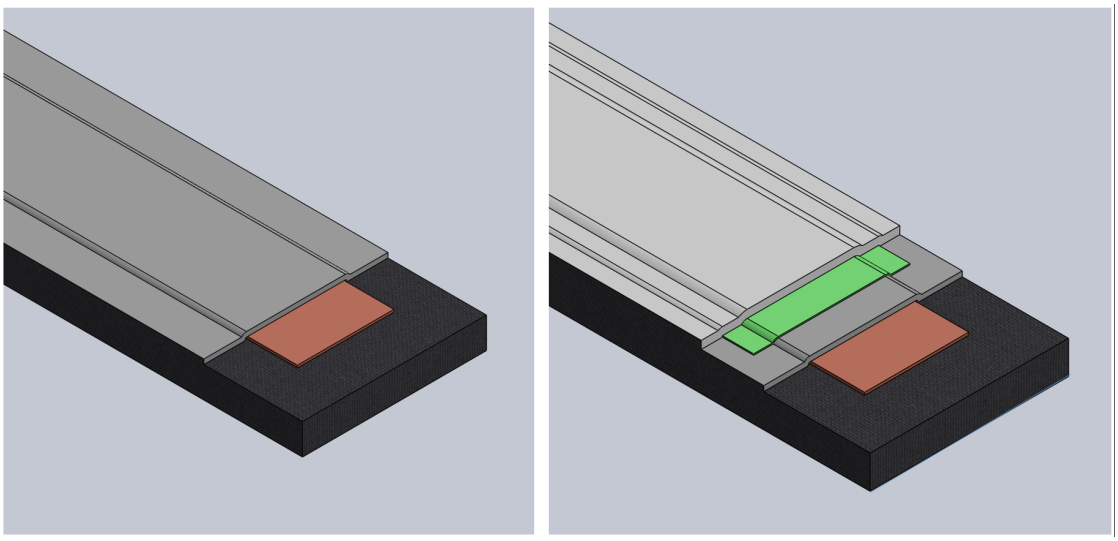


Figure 3.2: The configuration of a) unshielded sensors, and b) shielded sensors. The shield layer is shown in green, the sensor layer in copper, the base dielectric layer in black, and the outer dielectric layer in white.

the original sensor tape placed above it to act as a shield. For shielded sensors, 10 mm diameter holes were punched in the electrodes to allow connections to be made to the electrode layer. In both cases, a top layer of 19.1 mm wide Snag Free Velcro (Velcro 90666) running the full length of the electrode was used as the top layer to encapsulate sensors, allow adjustment of straps on a subjects arm, and as an attachment point for the systems electronic components.

3.2.2 Connection Method

As alluded to in Sec. 3.1 significant issues cropped up with work hardening and failure of sensors at the point where they connected to wires. In order to resolve this, two possible methods of connection were considered for connecting the wires to the sensors:the loop method as shown in Fig.3.3 a and the snap method in Fig3.3 b.

In the loop method, loops were twisted and soldered in the end of connector wires before using conductive thread (Adafruit 2-ply Stainless Steel) to connect them to sensors strips by sewing through the top Velcro layer and into the conductive material below. A dab of hot blue was applied on top of the sewn connection both to shield the exposed wire, and prevent the hard point where the solder connection transitioned to the wire from acting as a point of failure.

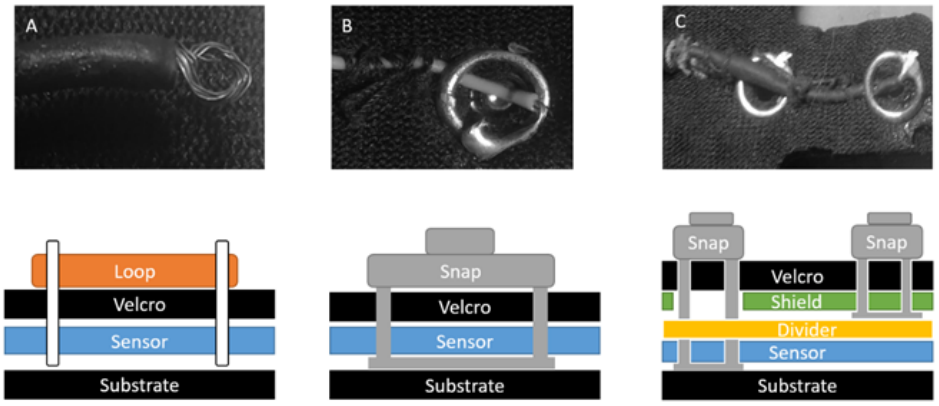


Figure 3.3: a) Loop connection the white portions represent conductive thread b) snap connection, the grey portions represent snap prong, c) Shielded snap connection, green represents the shield layer. Note the hole in layer to prevent shorting. Cartoons are not to scale

The second method used snaps to connect wire leads to the electrodes. While snap connections are common in settings ranging from anti-static harnesses to medical devices, concern remained that loose tolerances in the snap connection could become a source of capacitive noise. For sensors using the snap connection method, the crown portion of a size 12 prong snap (Amazon B0061B52FA) was pushed through the electrode side of the strip and the male snap was hammered down on top to form a contact to the electrode.

For both types of connections, the wires were sewn down to a Neoprene harness as shown in Fig. 3.4. This harness served the functions of insuring that wires would maintain constant positions during tests so that their motion could not become a source of noise; and insuring that the electronic connections would be in the same relative position across different platforms tested.

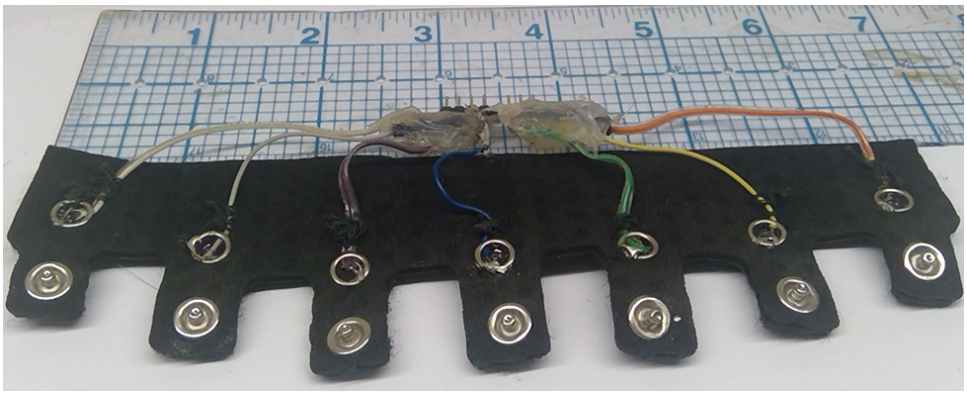


Figure 3.4: An example of the harness used to insure consistent placement of electronic components and minimize wire stress. The harness pictured was for snap connections. For loop connections, the wires would be threaded through holes punched in the same location as the upper row of snaps. In the shielded connections discussed in Sec. 3.2.3, the lower row of snaps would be connected to the sensors shield layer.

For both types of connectors, the ends of wires intended to interface with the electronics were soldered to a standard 2.54 mm spaced pin header and reinforced with hot glue to minimize wire strain. For this stage of the project, all electronics connections were made with Sullins connectors (PPTC082LJBN-RC, Digikey) to allow for the various PCBs and sleeves to be swapped out as needed. For sensors to be shielded, the wires connected to the electrode were wrapped in EMI-shielding heat shrink tubing (Amazon B019OTRL5O) with separate leads connecting to the

shield planes on the straps and the shield ports on the electronics as discussed in Sec. 3.2.3

3.2.3 Electronics

As discussed in Chap. 2, the original CapSense sensors were connected directly to the capacitive sensor pins on a Teensy microcontroller board (using the MK20DX256VLH7 Cortex M4 microcontroller from NXP Semiconductors). The original design used 7 pins, lacked a simple way to add more ports and was unshielded. In order to validate concerns including the lack of shielding as a source of noise and give the option of adding more electrodes, alternate designs were explored. Several approaches were pursued to improve the electronics design. A universal model of the PCB was designed that used the following common components: a Teensy 3.2 with an embedded MK20DX256VLH7 microcontroller, an HC-O5 Bluetooth module for communication, a LION charger for battery maintenance (MCP73831T-2ACI/OT), and a BNO055 IMU to measure forearm orientation for later experiments. Using this basic PCB footprint as a starting point, three variations were made. The first version simply connected the native capacitive sensor pins on the Teensy 3.2 directly to inputs for the CapSense as seen in Fig. 3.5 a.

A second version (PCB2) was made to allow for shielding. Manufacturer recommendations for shielding capacitive touch sensors typically involve either a passive shield using a ground plane behind the electrode side not intended to interface with the body [22], or using an op amp configured as a unity gain buffer in order

to block unwanted signals[86]. Some researchers caution that active shielding can itself become a source of noise[80]. For PCB2 an op-amp (MCP6024T-I/ST) was connected to the Teensys capacitive touch pins in a unity gain configuration. A switch controlled power to the op amp so that the performance of the system with the op amp present but powered off could be compared to that with the op amp powered on. The op-amp was connected to a row of outputs that allowed it to be independently connected to sensors so that the performance of configurations with shielding turned on but not connected could also be compared. The basic format for the shielded units is shown in Fig. 3.5 b.

Finally a third version (PCB3) was created to potentially explore the increasing the number of electrodes available. Using 14 channels for capacitive sensing would allow strap electrodes to be placed on both the anterior and posterior of the forearm. Because only 11 capacitive sensing ports are available on the native hardware platform of the Teensy 3.2, the proposed solution was to use a 16 pin multiplexer (74HC4067PW,118) to allow more electrode connections. This multiplexer was connected to a single pin on the Teensy 3.2 and had 14 possible connections available, all of which were connected to individual op am connections as shown in Fig. 3.5 c.

3.2.4 Construction

A laser cutter (Universal VLS3.50 Laser Cutter) was used to cut the harnesses discussed in Sec. 3.2.2 from neoprene and a base layer template from the materials

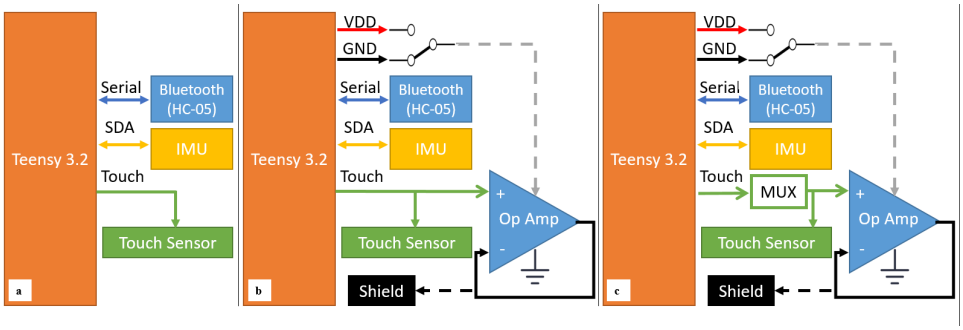


Figure 3.5: The electronics subsystems, and the path taken by a single capacitive sensor signal in a) Baseline configuration (PCB1), b) shielded configuration (PCB2), and c) shielded and multiplexed configuration (PCB3).

discussed in Sec. 3.2.1. The individual sensors were assembled initially using the double sided tape to position parts, and then stitched along edges of external dielectric with a sewing machine (Janome HD-3000BE). Snaps were used to position a 7.5" long 1.5" wide Velcro 1 Wrap Tape (B00404ULFQ). The other end of the tape used snaps to position it on an elbow brace (Tomight B07R5CSJN8) to insure that the position of the CapSense relative to the elbow would stay relatively constant when components were switched out between trials. The labeled components can be seen in Fig. 3.6.

3.3 Experimental Validation

The combination of components available led to a total of 19 combinations of settings and components needing to be validated. In order to streamline the process, a smaller subset of gestures was selected to validate. In Sec. 3.3.1 the gestures selected are listed and described, and the method used to collect data

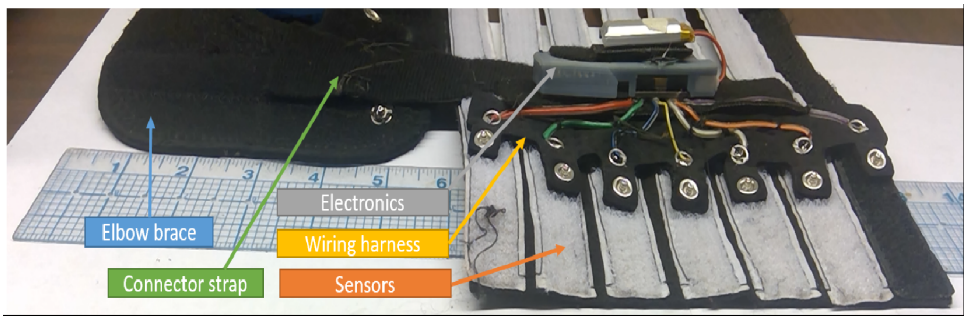


Figure 3.6: An example of the finished sensors, electronics, elbow brace, wiring harness, and positioning strap.

is discussed in 3.3.2. As discussed in the introduction when moving from in-trial tests to inter-trail tests, the accuracy of the RF model used in Chap. 2 decreased significantly. Given that both KNN and SVM came in as close seconds for accuracy in they were tested on the data. The algorithms and the procedure used to test the data is described in Sec. 3.3.3.

3.3.1 Gestures

In Chap. 2 32 separate gestures were used. This selection was based off a combination of individual wrist positions representing the 3 axes of motion the wrist is capable of, individual finger flexions and extensions, and the set of gestures put forth in the Cutkovsky taxonomy of grasps. The number of gestures used in this chapters' experiments was pruned for two reasons. The first was that it would have taken a significantly longer time than was available to test all of these grasps and positions across all available hardware combinations; and to some extent pointless. As discussed in Chapter 1, there are about 5-7 gestures that come up repeatedly as

being important across the various fields in which gesture recognition is used. The second reason lay in the fact that the 3 DOF of the wrist should not be taken to be separate gestures, but instead should be treated as features that occur in parallel with a gesture. In Chap 2, the wrist motions (W1-W6) actually represent different positions that gesture G13 can be in. Meanwhile for gestures G1-G9, the wrist is in a different position than the one used for gestures G10-G16 which may have made differentiating these gestures a more trivial task than it would be in reality. In order to simplify testing, a fixed wrist position was used in this chapters' experiments and the gestures selected for this chapter were those likely to be meaningful for as many potential applications of the technology as possible. They included: Neutral, Spread Fingers, Fist, Pincer, Pinch, Point, and Thumbs Up following the pattern shown in Fig. 3.7 to insure an equal representation of all gestures and to record transitions between all gestures of interest for future review.

3.3.2 Data Collection

Due to COVID 19 restrictions, the only subject available for the experiments in this chapter and the ones following was myself. The IRB had a complete lockdown on early stage human testing in place. The elbow brace was donned first by the subject to insure that sensors would be in the same relative position on the arm for each test. The positioning strap was then snapped onto the positioning snaps on the elbow brace and the sensors being used for the data set. The sensors were secured around the subjects arm using Velcro. A chalk pencil was used to label the desired

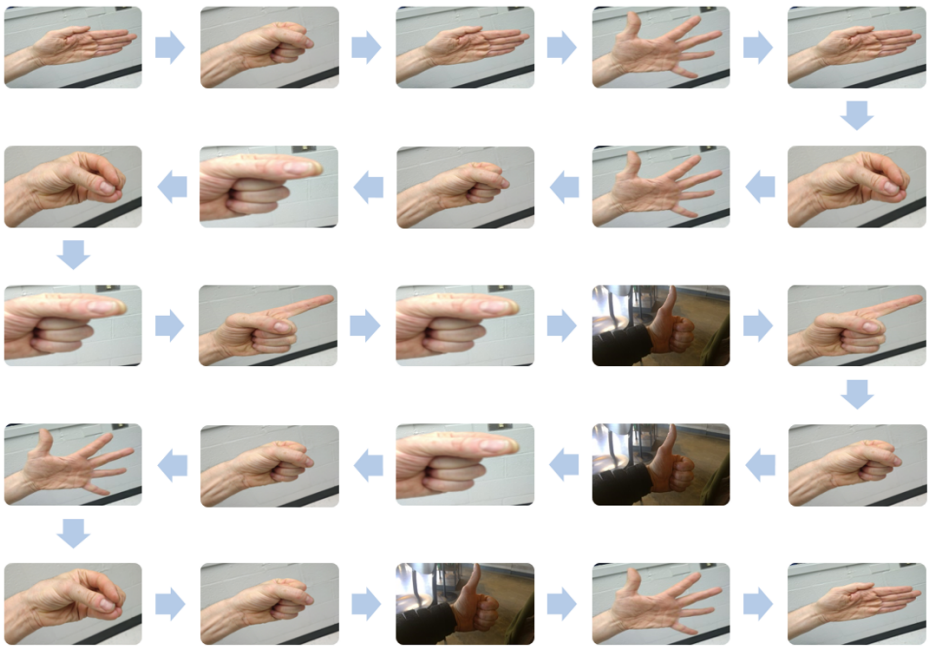


Figure 3.7: The order in which gestures were performed in this experiment.

location of the straps end points on each set of sensors so that variability in how tightly the sensors were wrapped around the arm could be eliminated as a factor between trials. The PCB for a given set of experiments were then connected to the sensors wire harness; the length of the wire harness itself constrained the electronics to roughly the same position on the arm for each set of experiments. The subject was seated on a straight backed wooden chair in front of a desk with their back against the back of the chair, and their feet placed at the front feet of the chair to insure consistent body posture across trials, and attempted to hold their arm against their side with their elbow bent at a 90 degree angle. To insure environmental uniformity between tests, tape was used to outline the position of the chair the subject was seated on relative to the desk, and a tape outline as provided on the desk to insure

consistent placement of the computer that the data collection application ran on as seen in Fig. 3.8. A companion application was used to establish communication

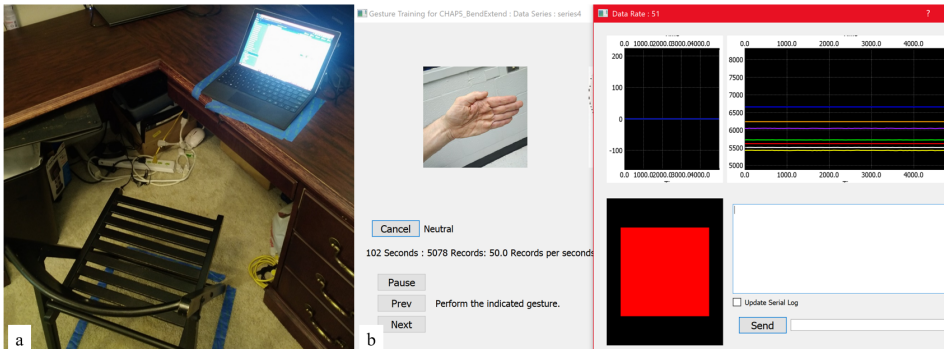


Figure 3.8: a) the chair and desk setup for experiment, b) the companion application.

with the electronics which streamed capacitive readings from each sensor in a comma separated string at a 20 Hz frequency. The companion application cued the sequence of gestures shown in Fig. 3.7 by providing an image of the desired gesture, showed duration of time for which gesture had been held, displayed real time incoming data from sensors, and appended a label associated with the cued gesture to the end of the data string before saving to a .csv file for offline processing. The subject instructions were to make the prompted gesture as naturally as possible, hold for 8 seconds, and then click the button on the companion application to go to the next gesture. When the 'next' button was clicked, the companion application gave 3 seconds to transition between gestures during which incoming data was labelled as transitional.

3.3.3 Machine Learning

All datasets were first scaled using the standard scalar method as discussed in Sec. 1.2.1. Both KNN and SVC were tested as alternate algorithms to RF using a train/test split of 0.7. Inter trial accuracy for a given combination was established by training a model on the first data set from a day and testing the model on the days second data set. As will be discussed in Sec.3.3.5 paired datasets were collected on two different days to in case user error or fatigue on a given day became a source of error. The average of both days accuracy as taken as the average accuracy for the combination. If a difference of more than 10% was found between the accuracy calculated on the two different days, human error was assumed and a third pair of data sets was collected. Whichever two data sets accuracies' were closest together were then averaged to generate the assumed accuracy for the combination.

3.3.4 Combinations Tested

There were three types of connections explored as discussed in Sec. 3.2.2: wired, snap connections (Snap), and shielded snap connections (Dual Snap). There were 5 physical configurations of the sensors tested wire loop connections on felt base bottom dielectric(FeltWire), snap connections on felt bottom dielectric(FeltSnap), snap connections on neoprene bottom dielectric(NeoSnap), shielded snap connections on felt bottom dielectric(NeoShield), and shielded snap connections on neoprene bottom dielectric (NeoShield). In order to save material, construction time, and test time the wire loops were only implemented on the felt sensor sleeve. It

was assumed that the trend in relation to loop vs snap connections would hold no matter what the material substrate was. As discussed in Sec. 3.2.3 there were 3 PCB tested, the original version (PCB1), the shielded version (PCB2), and the shielded and multiplexed version (PCB3). Both PCB2 and PCB3 had a switch that allowed power to be disconnected from the shielding op amp so that any changes in performance caused by the presence of op amp could be accounted for. A summary of the combinations tested and their averaged results can be found on table 3.1.

3.3.5 Experiment

For each sensor/pcb/setting combination two data sets were taken prior to moving to the next combination. In order to insure that user fatigue was not an issue, a second group of data sets were taken on another day and the combinations of sensors/pcb/setting were run in a different order.

3.4 Results And Discussion

SVC significantly outperformed KNN for all IT tests by a margin of 10%-20%. Given the significant difference in performance, only SVC will be reported on in this section.

As can be seen on Table 3.1, no matter what the material or connector, PCB2 outperformed both PCB1 and PCB3, averaging an accuracy of 85.7% accurate with a standard deviation of 7.4% across all trials. This was significantly better than PCB1 (77.6% with a standard deviation of 7.3%) and PCB3 (76.7% with a standard

Sensor Set	Electronics	Connection	Shield On	Average Accuracy	Standard Deviation
PCB1	Felt Wire	Wire	N/A	87%	5%
PCB1	Felt Snap	Snap	N/A	76%	2%
PCB1	Neo Snap	Snap	N/A	70%	10%
PCB2	Felt Wire	Wire	No	90%	7%
PCB2	Felt Wire	Wire	Yes	87%	1%
PCB2	Felt Snap	Snap	No	90%	1%
PCB2	Felt Snap	Snap	Yes	87%	4%
PCB2	Felt Shield	Snap2	No	82%	2%
PCB2	Felt Shield	Snap2	Yes	68%	1%
PCB2	Neo Snap	Snap	No	81%	2%
PCB2	Neo Snap	Snap	Yes	83%	4%
PCB3	Felt Wire	Wire	No	79%	1%
PCB3	Felt Wire	Wire	Yes	81%	3%
PCB3	Felt Snap	Snap	No	73%	1%
PCB3	Felt Snap	Snap	Yes	79%	3%
PCB3	Felt Shield	Snap	No	73%	1%
PCB3	Felt Shield	Snap	Yes	79%	3%
PCB3	Neo Snap	Snap	No	72%	3%
PCB3	Neo Snap	Snap	Yes	77%	4%
PCB3	Neo Shield	Snap	No	72%	4%
PCB3	Neo Shield	Snap	Yes	83%	2%

Table 3.1: Combinations Tested and Their Results

deviation of 3.8%)

When comparing results of various connection types, the Loop connections were the most accurate of the pack at 84.9% with a standard deviation of 4.2% as compared to 80.0% for the snap connectors with a standard deviation of 7.3%. However several of the wire loop connections failed over the course of the various tests due to work hardening and fracturing of wires. It was decided that the convenience of the snap connectors was slightly more important than the wire loops slight accuracy improvement. The least accurate tests were the ones using Dual Snap connectors

with an average accuracy of 76.6% and a standard deviation of 3.8%. It is possible that this was caused by gradual failures inside the shielded portions allowing the external shielding to short to the sensor layer and may bear revisiting at some future point. However given the time required to build the shielded harnesses, the extra materials needed for them, and their relatively low accuracy, they were abandoned for purposes of this work.

Across all tests compared, the felt dielectric had an accuracy of 83.0% with a standard deviation of 6.8% as compared to the Neoprene dielectric which had an average accuracy of 78.1% with a standard deviation of 7.2%. This difference in performance was not expected, and several potential sources could be responsible. The performance of the neoprene came as something of a surprise. Felt has a listed permittivity of 1.22 [78], as compared to the permittivity of neoprene which is 6.7 [28]. Capacitive measurements vary proportionally with permittivity and inversely with distance. The neoprene fabric used was double the thickness of felt, but also had nearly six times the permittivity. This should have led to nearly three times the change in signal as muscles flexed and relaxed. Which in turn would presumably have led to higher sensitivity. Two possible explanations for this include the fact that the neoprene was perforated for comfort. The relative permittivity of air is roughly 1 at standard temperature and pressure, so the perforations would likely lower the permittivity of the bulk fabric significantly. This also would allow sweat vapor out, which had the potential to saturate the nylon exterior layer leading to a significant reduction in its perimittivity over time. In some of the least successful tests, significant drift of the signal over time was observed. A potential future

workaround would be to encapsulate the area around sensors in a waterproof dielectric, or at least wrap the arm in a waterproof membrane prior to donning sensors. The second possible cause was the comparative stiffness of the two fabrics. Craft felt is stiff and fairly difficult to compress; neoprene on the other hand is both easy to compress and elastic in nature. As muscles flexed and relaxed, hysteresis could have played a role in the fabrics behavior and added noise to the electrodes. Further research is needed to improve the capacitance model in this sensing system.

3.5 Final Design and Future Work

3.5.1 Final Design

Because the snap connection method was less prone to failure due to wear (and easier to rebuild when it did fail), the snap connector was selected as the method to use for the remaining chapters of this dissertation. A final version of the V6 PCB was constructed for use in future chapters that did not use any on-board op-amps given that shielding did not seem to affect accuracy. To eliminate the socket connection as a point of failure, the wire connectors were soldered directly to the PCB, and both they and the PCB were sewn onto a fabric sleeve as pictured in Fig. 3.9.

3.5.2 Future Work

In recent work, using strain measurement on resistive elastomers has become a popular area of research for wHMI devices [4, 77]. A Lycra substrate with conduc-

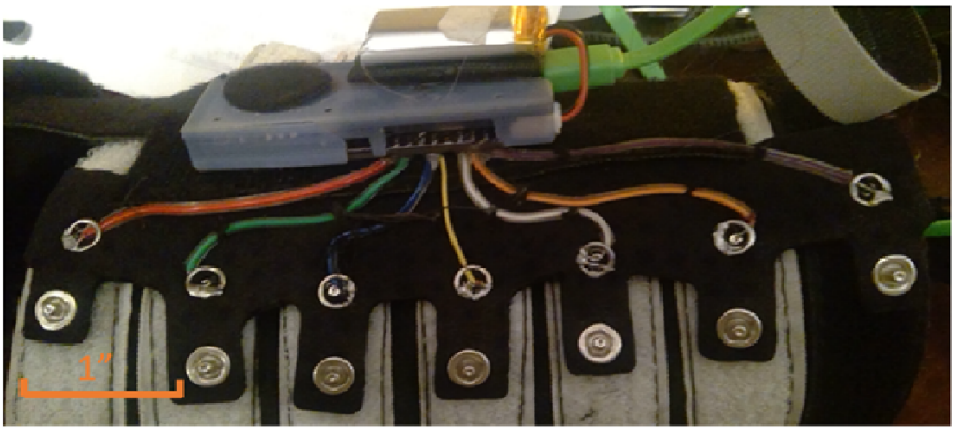


Figure 3.9: An example of the finished sensors, electronics, elbow brace, wiring harness, and positioning strap.

tive elastomers could act to resolve both issues. The Lycra material would ensure consistent pressure on a subjects arm between trials, and tracking both resistance and capacitance on the same sensor strap may provide more data and allow for higher accuracy over a wider range of gestures and motions. Currently a spray on sensor version is in the process of being built. It would also be worth exploring encapsulation of the sensors in a waterproof membrane to mitigate any potential effects from subjects sweat saturating the sensors.

3.5.3 Conclusions

The raw materials for the variety of the CapSense constructed in this chapter comes to $17 and can be constructed at home from off the shelf components without any specialized equipment. To put this in context, the primary technology used for wHMI in commercial devices is Electromyography (EMG). While it is the clinical gold standard; it requires very accurate placement, suffers from signal drift over the

course of the day, picks up significant environmental noise, and professional grade EMG equipment costs thousands of dollars [85]. Hobby grade EMG exists and is cheaper at $38, but is generally limited to prototyping and educational use by low accuracy and high signal to noise ratios [57]. As described in Chap. 1 a significant adoption barrier for technologies is cost, and as such creating less expensive sensors while maintaining accuracy is a good step forward. That said, the average inter trial accuracy of 90% is at the lower range of contemporary work. It does however represent a significant improvement over the 30% IT accuracy reported at the start of this chapter. The next chapter will look at what can be done to improve those results further.

Chapter 4: Exploring the Limitations of Stationary Gesture Recognition with Capacitive Straps

4.1 Introduction

In Chap. 3 a best practices hardware design for the sensor substrate was selected and further ML algorithms were tested. The best inter-trial accuracy was only 90% over 7 gestures which was a substantial improvement over Chap. 2 results, but is not a particularly high value by modern standards.

In a survey of current work in wearable human machine interfaces no study was reported that achieved less than a 90% accuracy [38]. At the low end 92% accuracy for 4 gestures was reported using FMG sensors with an SVM algorithm [51]. The Leone group accheived a 98.8% accuracy on 7 classes of gesture. However it should be noted that three of the classes were open hand, supination with open hand, and pronation with open hand [50]. So it could be argued that they really could identify 5 classes, but that one class had 2 additional degrees of freedom. On the high end, an array of 64 sEMG electrodes using a High Density Computing model on a Field Programmable Gate Array achieved 97% accuracy over 13 gestures. This did however require a far more expensive hardware configuration to achieve [61].

Further improving the accuracy of the sensing system would require either 1) improving the sensitivity of the capacitive sensors to pick up smaller deformations in the forearm, 2) improving algorithm tuning, or 3) revisiting the biomechanics of the forearm to improve our understanding of the limits of this sensing technology for gesture recognition. In this chapter, we explore all three of these improvements, ultimately resulting in a much greater understanding of the sensing technology. Sec. 4.2.1 explores the fundamentals of the CapSense system on the Teensy 3.2 microcontroller board to improve sensor sensitivity. Sec. 4.2.2 looks into the possibility of changing how data is labelled, which algorithm is used for classification, and introduces probability thresholds as means of rejecting classes the model is not confident on. In Sec. 4.3.1 the procedure used to collect data is discussed, and Sec. 4.3.2 follows up with a brief description of how the data was processed. In Sec. 4.4, the results of both improving sensor sensitivity and algorithm tuning are compiled and implications toward the class of gestures that can be reliably classified within inter-trial testing is further elucidated using these results. Sec. 4.5 offers conclusions and possible future work.

4.2 Methods

4.2.1 Microcontroller Settings

As mentioned previously, one possible approach to increasing the sensitivity of the capacitive sensors was to tune the capacitive sensing electronics on the Teensy 3.2. The Teensy 3.2 uses an NXP Semiconductors MK20DX256VLH7 processor.

This processor has 12 pins that are capable of capacitive measurements within a range of C_{in} = *0 to 1200 pF*. These measurements are represented digitally as counts rather than Farads. The count is a ratio of how many times a capacitor with a known value can be charged/discharged in the time it takes to charge/discharge the unknown capacitor[56]. The microcontroller uses a 16-bit ADC resulting in capacitance measurements between 0 and 65535 counts [45].

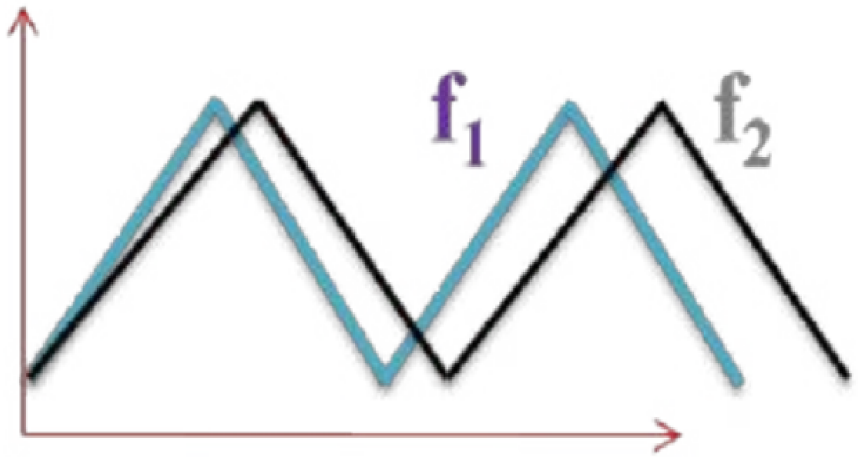


Figure 4.1: Internal Reference in blue vs oscillations observed in black [56].

The manufacturer provides a formula to convert digital counts to a capacitance value in Farads [45], and this formula indicates that the measured capacitance is a function of several user-adjustable settings including: External Current (I_{ext}), Number of Scans per Read (NSCAN), and a frequency scaling factor (PSCALE) that adjusts how much time each reading takes.

Equation 4.1 is used to calculate the maximum capacitance that the system

can read as a function of the user adjustable variables. For example, use of a larger external current will allow for a larger maximum capacitance to be measured because the unknown capacitance can be charged and discharged more quickly. Similarly, increasing the frequency scaling factor to slow down the timer increments will result in a smaller maximum capacitance that can be read.

$$C_{max} = \frac{65535 * I_{ext}}{4 * NSCAN * PSCALE} \tag{4.1}$$

The highest value observed in any combination of hand and wrist positions tested was 47 pF which is far lower than the maximum 1200 pF that the microcontroller is capable of reading. As such, the system can be fine-tuned to match the maximum readable value C_{max} with the maximum expected value C_{expect} of approximately 47 pF so that more precision could be achieved in measurements, potentially giving better inputs to the machine learning algorithms.

Per the manufacturer, the variables that effect the maximum value that can be read also have other effects on capacitive readings including: changing the sensitivity of capacitive measurement, increasing robustness to noise, or shortening read time. Table 4.1 gives some details on each of these inputs, its formula, its accepted range of values, and how it can be expected to effect sensitivity, read time, and noise resistance.

There are many different combinations of variables which can lead to a given C_{max}. The manufacturer states explicitly that increasing I_{ext} should make the system more robust to noise. It also seems reasonable that multiple scans should allow

Setting	I_{ext}	NSCN	PSCALE
What it is	External Current	Number of Scans	Oscillator Frequency Divider
Formula	$2 * (CURR + 1)$	$NS + 1$	$[2^{PS+1}]$
Range of Inputs	$0 < CURR < 16$	$0 < NS < 32$	$0 < PS < 28$
C_{max}	Linear Scaling	Inverse Linear Scaling	Inverse Exp. Scaling
Sensitivity	Inverse Linear Scaling	Linear Scaling	Exp. Scaling
Read Time	Linear Scaling	Linear Scaling	Exp. Scaling
Noise Resist	Improves	Should Improve	Should Improve

Table 4.1: Software Configurable Settings

averaging to cancel out noise, and that a shorter read time would be less affected by transient noise in environment. By maximizing different variables, four different settings were generated that would allow use of the full-scale range on the ADC to measure C_{expect}, and optimize for slightly different conditions. A default setting was also used as a control resulting in four total settings tested:

Default: settings used the default parameters from the manufacturer and did not attempt to use the full range

Fast Read: settings focused on minimizing the read time

Noise Reduction: settings focused on noise reduction

Sensitivity: settings focused on sensitivity of readings

Because of the nonlinear interaction between the various user controllable variables (and the speed of modern computing), a brute force method was selected to find candidate values.

4.2.2 Machine Learning

Both of the data preparation methods discussed in Chap. 1; *Standardization* Eqn. 1.2 and *Normalization* Eqn. 1.1 were tested as data scaling methods. The candidate algorithms selected for comparison were KNN, SVC, and LDA, all of which are discussed in Chap. 1, Sec. 1.2.2.

4.3 Experiment

The experiment consisted of two phases, data collection which is explained in Sec. 4.3.1 and processing that data via machine learning as discussed in Sec. 4.3.2.

4.3.1 Data Collection

The procedure from Chap. 3 was used again to collect data in this chapter. In short, the subject was seated upright in a chair with the instruction to hold their arm against their side bent at a 90-degree angle wearing the sensor assembly and positioning brace. A companion app cued the subject to perform gestures, and recorded both label and sensor data. The gestures used are recapped in Fig. 4.2 and were performed in the order shown in Chap. 3 on Fig. 3.7.

For each of gesture other than transitional, 8 seconds of data was collected at 20 Hz. Three seconds was given to transition between gestures, and that transitional period was labeled as transitional so that it could be recorded. Transitional was a catchall class that represented the transition between gestures and was included to determine whether the ML algorithm could reject data that could not be classified



Figure 4.2: The gestures used for the experiment in this chapter included:neutral, spread, fist, pincer, pinch, point, thumbs up, and transitional .

within a given gesture. The collected data was labelled with the gesture class being prompted and saved for offline analysis.

4.3.2 Data Processing

To better simulate real world performance a leave n out cross validation scheme discussed in the introduction as adopted. For each cycle of testing, Of the available n datasets, *n-1* were reserved for testing and 1 was used for training. The training data was divided into a 70/30 train /test fractions to build and validate a model prior to validating the generated model against each of the remaining *n-1* datasets by using their capacitive data as input *features* for the model and comparing its predictions to the known *target* labels. These inter-trial (IT) accuracy's were used to compare performance between models. An image of the training scheme is shown

in Fig. 4.3. Because I was interested in determining whether non-gesture data could be easily rejected, tests were run first with all class data, and then with the transitional class pruned from the data.



Figure 4.3: Leave n out training scheme. The training set (green) was used for training a model and the model was tested on each of the remaining data sets (grey). The average result of training on all test/train pairs was recorded as the presumed real world accuracy.

For each train / test pair, a number of possible settings for the machine learning pipeline were swept through changing one setting value at time while holding other settings fixed. These settings included data labelling method (Raw Labels, Expanded Labels), scaling method (MinMax vs StandardScalar), and ML Classifier (KNN, SVC, LDA). The results of each combination were saved to a spreadsheet for offline analysis. Each row of the spreadsheet recorded: the dataset trained on, the dataset validated against, the labelling method, the scaling method, the ML classifier, the raw accuracy for all data including transitional class, and the in class accuracy after pruning out the transitional class. After evaluating the spreadsheet to determine which settings produced the most accurate results on average across

the tests, the LNO test method was used a second time with the best settings to determine if specific gestures posed a problem. To do so, a list of the the models predicted class labels vs labeled ground truth class labels was compiled, appending each train/test dataset pair. This list was used to generate a confusion matrix that could be examined to determine which classes the model had trouble identifying.

4.4 Results

4.4.1 General

Sweeping through all the parameters listed in Sec. 4.3.2 generated over 19000 lines of data for each of the microcontroller settings tested. To get a general sense of which microcontroller settings should be used, the results were filtered to find test - train pairs which achieved over 90% accuracy. All of the examples meeting this criterion used the default microcontroller settings with SVC and a slight majority of them 69% used standardization to normalize data. The average Raw Accuracy was still low, coming in at 70% even for SVC. In order to investigate why a confusion matrix Fig. 4.4 was generated as described in Sec. 4.3.2.

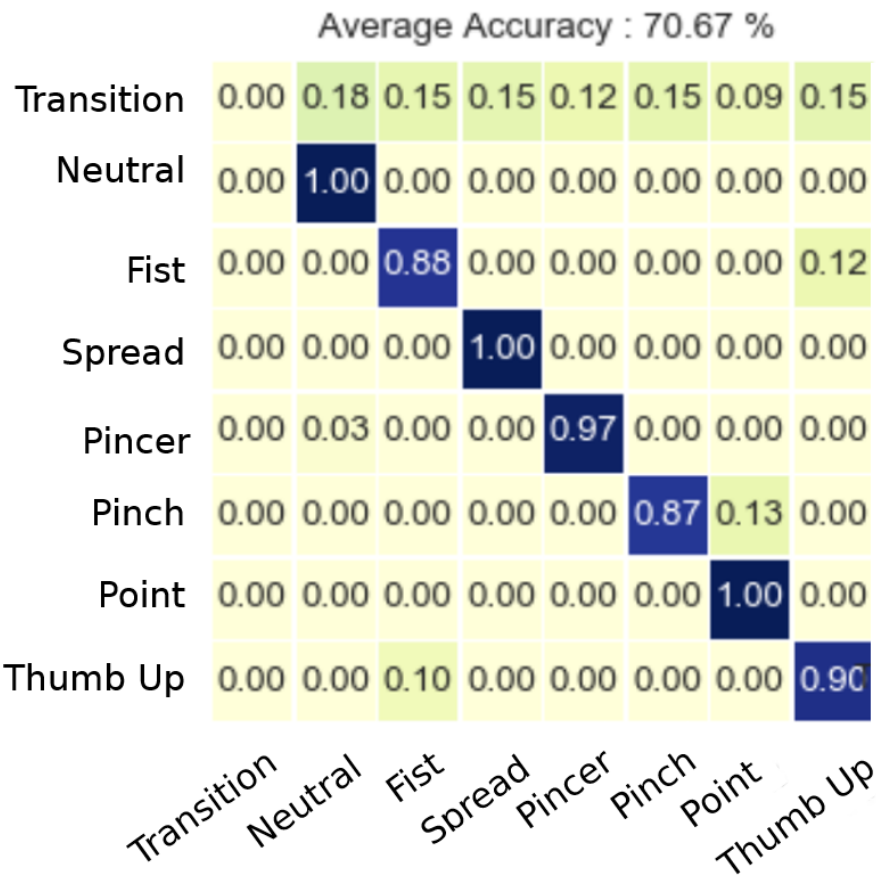


Figure 4.4: Confusion matrix showing raw results for static classification.

As can be seen the majority of error from incorrect classification falls into a few specific categories. Firstly the transitional class is often mistaken for whichever class the gesture is closest to being at a given moment. Secondly there is some confusion between Pinch and Point, and Fist and Thumbs Up. The confusion surrounding transition will be discussed in section Sec. 4.4.2. This leaves the remainder of the confusion centered around gestures involving use of the thumb, which will be discussed in Sec. 4.4.3. After removing the gestures the depended on thumb position, SVC still was by far the most accurate of the algorithms tested. The average

accuracy of across all train/test pairs for default settings is shown on Table 4.2 for the various algorithms considered. The tables first row shows the accuracy of all remaining gestures when the transitional class is removed, and the remaining rows look at the results of removing some of the other gestures which the model had difficulty with.

Non Transitional Gestures Used	SVC	LDA	KNN
All	91%	86%	88%
All But Thumb Up	93%	90%	90%
All But Pinch	95%	91%	83%
All But Pinch and Thumb Up	99%	97%	87%

Table 4.2: Combinations Tested and Their Results

4.4.2 Transitional Class

As seen in Fig. 4.4, the majority of classification errors observed were on data labeled as transitional. The transitional class had been included in tests initially because I was interested in determining whether the models would reject data that did not fit a specific class. This approach did not prove successful for two reasons. First, there is a lot less transitional data given the relative time it takes to transition between gesture states versus time spent in the state which leads to an imbalance in data available to train. This imbalance is further complicated by the fact that samples in this class are inherently moving between actual gestures and is likely to bear a strong resemblance to them at any given moment. After dropping these transitional classes, the accuracy of classification using SVC as a model rose to 91%.

A set of confusion matrices showing how accuracy changes as various problem

gesture classes are dropped is shown in Fig. 4.5.

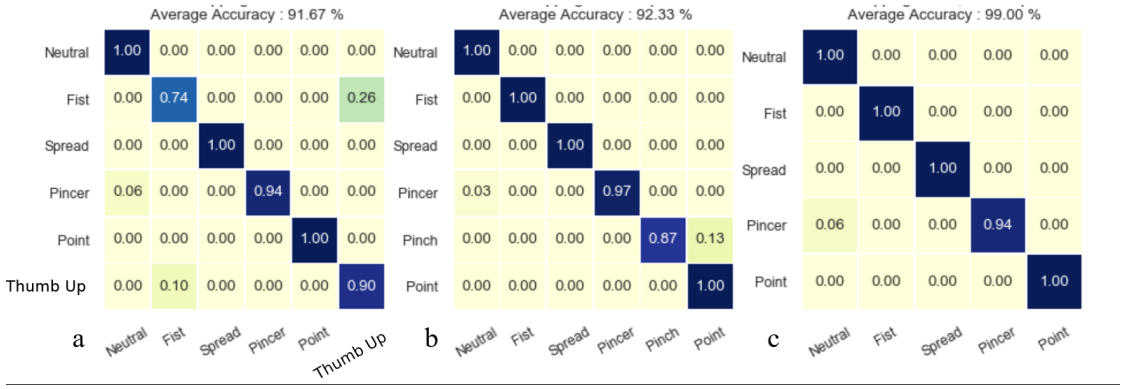


Figure 4.5: Results of removing gestures involving the thumb a) pinch removed, b) thumb up removed, c) pinch and thumb up removed.

4.4.3 Gesture Limitations

To better understand the limitations around gestures that can be detected using this sensing technology, the above analysis was repeated while dropping certain gestures. The other primary source of error involved thumb positioning. Of the eight muscles that insert on the thumb, five of them connect to other bones in the hand as shown in Fig. 4.6a. The remaining three are considered deep muscles, meaning that their insertion points are buried deep under other larger forearm muscles. As seen in Fig. 4.6b of those three only one, the Abductor Pollicis Longus (ABP) which controls Abduction of the thumb is of a significant size [62]. This may explain why the thumbs up gesture has somewhat higher accuracy in classification than the pinch gesture. In thumbs up and fist, the sole difference is flexing or relaxing the ABP. In pinch vs is most often mistaken for point it is likely the signal of the larger Extensor

Policis muscle (which is responsible for extending the index finger) obscures the signal from the ABP.

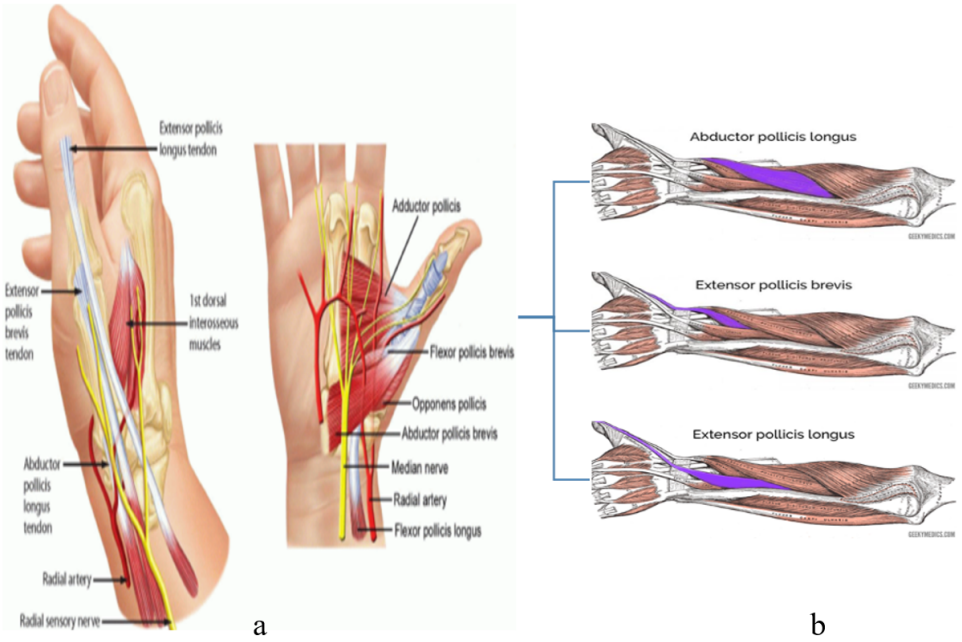


Figure 4.6: a) The intrinsic muscles of the thumb image from motionplus.com [63], b) the extrinsic muscles of the thumb image from geekymedic.com [64].

4.5 Conclusions and Future Work

4.5.1 Conclusions

It was found that when the arm is in a stationary position, 7 classes of gestures can be distinguished with 91% accuracy for inter-trial testing. When gestures depending heavily on thumb motion are removed, the remaining 5 classes of gestures can be predicted with 99% accuracy. These results fit will within the range of accuracy (92%-98%) discussed in the introduction of this chapter for sEMG [50]. It is

likely that this sensor system could recognize a larger number of arbitrary gestures if the thumb position was not a significant factor in differentiating them. The performance of the various algorithms also gives some insight into the data they were receiving. For LDA to work well feature data needs to be normally distributed, the classes need similar covariance, and features need to be linear separable between classes. LDA does not do well with data that has significant overlap between class features, and is heavily effected by outliers in the data sets. Its poor performance on this data set likely implies that the data is not linear separable and may have significant outliers. The poor performance of KNN could indicate that not all of the sensors share equal importance. KNN often suffers from poor performance when redundant and non-contributing *features* exist because it does not have any means of penalizing irrelevant *features* [36].

4.5.2 Future Work

A possible workaround for the difficulty experienced in tracking the thumb could involve a design using a half glove with a thumb sleeve incorporating a flex sensor. Not only could this sensor fusion allow for a broader range of gestures to be classified, it would also provide a mechanism for insuring consistent placement of sensors on arm between sessions of use. Adding even a partial glove would impede hand motion however. A unique aspect of sensors only on the forearm is that the hands are completely free for grasping and gesturing tasks. It would be of interest to determine how differences in placement and pressure between sessions on different

days impacts measurements and whether shifting of sensors on the users arm during same-day trials is a significant source of error. An initial attempt is made at this in Chap. 5. Finally, COVID limited access to subjects with different body sizes and compositions. Early tests prior to COVID showed good results on individuals with higher BMI, but seemed to show difficulty on subjects with shorter and slenderer arms. As vaccines become more widespread and society enters its post COVID phase more subjects with a variety of different body types should be recruited.

Chapter 5: Extended Wear and Inter-Day Tests of Stationary Gesture Recognition with Capacitive Straps

5.1 Introduction

In Chap. 4 the experiments focused on datasets collected in one sitting. While this did help elucidate the limitations of the CapSense system, it did little to indicate how well the system would perform with extended wear or if trained models could perform useful predictions the next day. In many existing technologies, when the device is worn for extended periods of time the accuracy can be diminished by a number of factors. Signal drift can change the actual values being read by the sensors, and if the sensors are able to shift in position due to the motion of a user, they can wind up over different tissue than they were originally trained on resulting in lower accuracy from models [18, 38].

Many studies point to changes in the position of sensors in between days as a reason that models require some amount of training data to be taken at the beginning of each new use session [38, 61]. A review by Campbell in 2020 cites differences in sensor position on the arm as being a confounding factor that can lower model accuracy by as much as 13% [18]. Moin, et. al. found that on High

Density sEMG, long term wear or removing and replacing sensors on a different day could lower accuracy of a trained model by on average 11.89% [61].

In theory, rotation on the users' arms should not have an impact on the CapSense's measurements due to the fact that they are measuring an entire cross section of the arm rather than a discrete point. However, shifts in radial position of sensors could lead to a significant decrease in the accuracy of models. Removing the device and putting it on again on a different day could also introduce changes both in the position of sensors on the arm, and the actual pressure with which the sensors wrap around the arm.

This chapter looks at both inter-session (IS) trials in which models are trained on a data set early in the day and then tested against a data set taken significantly later in the day, and inter-day (ID) models in which the model is trained on a data set taken on an entirely separate day. A challenge faced in this chapter is that differences in arm position between trials could become a confounding factor that lowers accuracy in both the IS and ID tests. The ID tests face an additional challenge because replacing the device on a different day could lead to a difference in how tightly the straps are adjusted between days, which could in turn change the baseline signal of the sensor.

Sec. 5.2.1 discusses approaches used to mitigate these challenges. It was also worth addressing some open questions regarding how models were trained and used. These questions include: a) how much data is needed at the start of a session to train an accurate model?, and b) how confident are models in their predictions? These are both discussed further in Sec5.2.2. In Sec. 5.3 the methods used to collect and

process the data are discussed. This is followed up in Sec. 5.4 with a discussion of the results and conclusions/future work in Sec. 5.5.

5.2 Methods

5.2.1 Placement

There are two types of variables that need to be controlled to insure uniform signals in a stationary test. The first type is variation in placement of the sensors on the body, and the second type is the variation of body posture and arm position between sessions. The use of a single subject in the experiments in this chapter simplified consistent placement of the CapSense on the arm. The elbow strap used in Chap. 3 was modified with snaps to insure that it's snugness on the arm would be consistent between trials. This constrained the position of the sensors along the radial length of the arm. In order to insure consistent pressure between trials, short straps (50.8 mm Nylon) were sewn to the ends of the sensor straps, and outfitted with snaps to connect them to the row of snaps on the end of the harness discussed in Sec. 3.2.2. An image of the sensor on the arm with adjustment snaps is shown in Fig. 5.1.

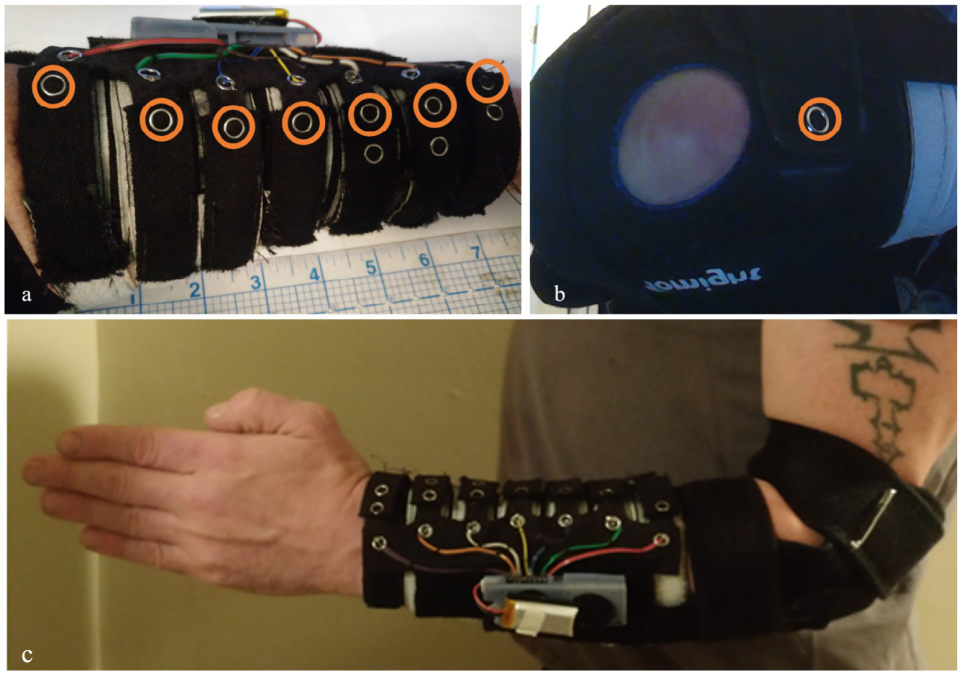


Figure 5.1: Snaps used to align between trials are circled in orange a) tightness on arm and b) position on elbow, c) the sensors worn on arm.

The second area of possible variation between session is body positioning. If the subject's elbow joint is at a different position, that implies that the muscles around the elbow joint are flexed or relaxed differently, which could potentially change the values read from other sensors. In an attempt to mitigate the possible effects of changes in posture the subject was seated in a straight backed chair and instructed to rest their back against the chair while resting their hand on the desks keyboard tray with their middle finger touching the desks edge when the hand was in the neutral position. Tape outlines were used to insure that the placement of both the computer running the companion application that the subject interfaced with and the chair the subject was seated on would remain constant between trials

as shown in Fig. 5.2.

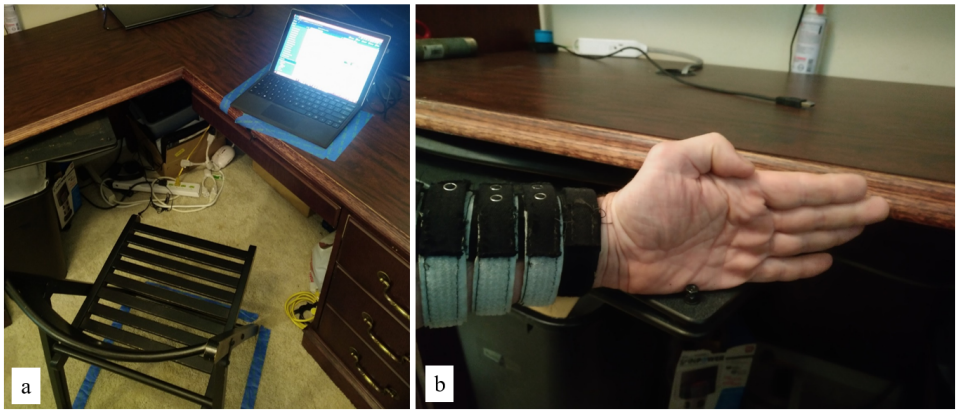


Figure 5.2: Setup a) Tape outlines were used to insure position of chair and laptop between trial sessions b) the hand was rested on the desks mouse pad with middle finger touching edge of desk when hand was in neutral position.

5.2.2 Machine Learning

The eventual goal of this device is use in long term wear. While in Chap. 4 by default the models were trained on 70% of the data available from a given trial, knowing how little data could be used to generate an accurate model was of interest. In order to do so, the train fraction (tr) was varied as described in Sec. 5.3.1.

Another hyper-parameter varied in this chapter was threshold confidence. All of the ML algorithms considered for use with the CapSense are capable of providing a percentage likelihood that a sample is in any of the classes the model can predict. Normally the default behavior is to select the model with the highest likelihood. This means that if a gesture is as little as 1% more likely than any other class, it will be selected as the correct gesture. This implies that a model which is barely able

to distinguish between classes can look good in offline testing, but display very poor accuracy in the real world. One way to test for this is to set a threshold probability (tp) for accepting that a gesture is actually in the class predicted by a model. If the probability gesture is lower than the threshold, the gesture can be rejected into a garbage class. This can be used to eliminate less confident or stable models early on.

In order to explore the model's robustness, a specific tp was fixed. This allowed two cases to be explored in models. In the case where a model is very uncertain and assigns near equal probability to many classes, then no class will have a high enough probability to exceed the assigned tp resulting in many samples will be placed into the garbage class. In the case where the model is very confident that a sample belongs in the wrong class, the number of samples placed in the garbage class will not necessarily rise, but the model will continue to incorrectly classify samples in spite of a rise in the tp.

5.3 Experiment

5.3.1 Data Collection

The gestures used and manner in which they were collected was very similar to that in Chap. 3. The only difference was that 3 sessions of data collection were used. For each section 4 data sets were collected. The first and second sessions were on the same day, but separated by approximately 1 hours time during which the subject was free to get up and wander about. The CapSense was then removed.

The next day the subject re-donned the CapSense sensors and took 4 more datasets. The methods described in Sec. 5.2.1 were used to minimize the effects of the subject getting up and moving about between IS tests, and replacing the sensors on the arm for ID tests.

5.3.2 Data Processing

The leave-one-out test method described in Sec. 4.3.2 was used again in this chapter. However three more variables were recorded. For each data used to train, multiple models were built using a range of tr and tp values. The tr was varied from $0.65 < tr < 0.85$ in 0.05 increments and the tp was varied from $0 < tp < 0.5$ in 0.10 steps. In order to assess the effect of user motion, and taking off /replacing the sensors each test train pair was given a Session Label (SL). These labels were used to distinguish between between the results of same day IS trials and inter-day ID trials. The data recorded from each test-validation set was as follows: Test set, Validation set, Session Label, Train Fraction, Confidence, ML Model, Gestures Used, Overall Accuracy. For analysis, the results were grouped by SL, ML model and Gestures Used (GU). Within these groups, a global average of all results in the group was generated to give a rough overview, followed by a more granular look at how tr and tp affected accuracy within a given grouping.

5.4 Results and Discussion

5.4.1 Results

The most obvious and immediate result is that SVC outperformed LDA significantly. A plot of the results using SVC can be seen in Fig. 5.3 and a plot of the results of LDA can be seen in Fig. 5.4.

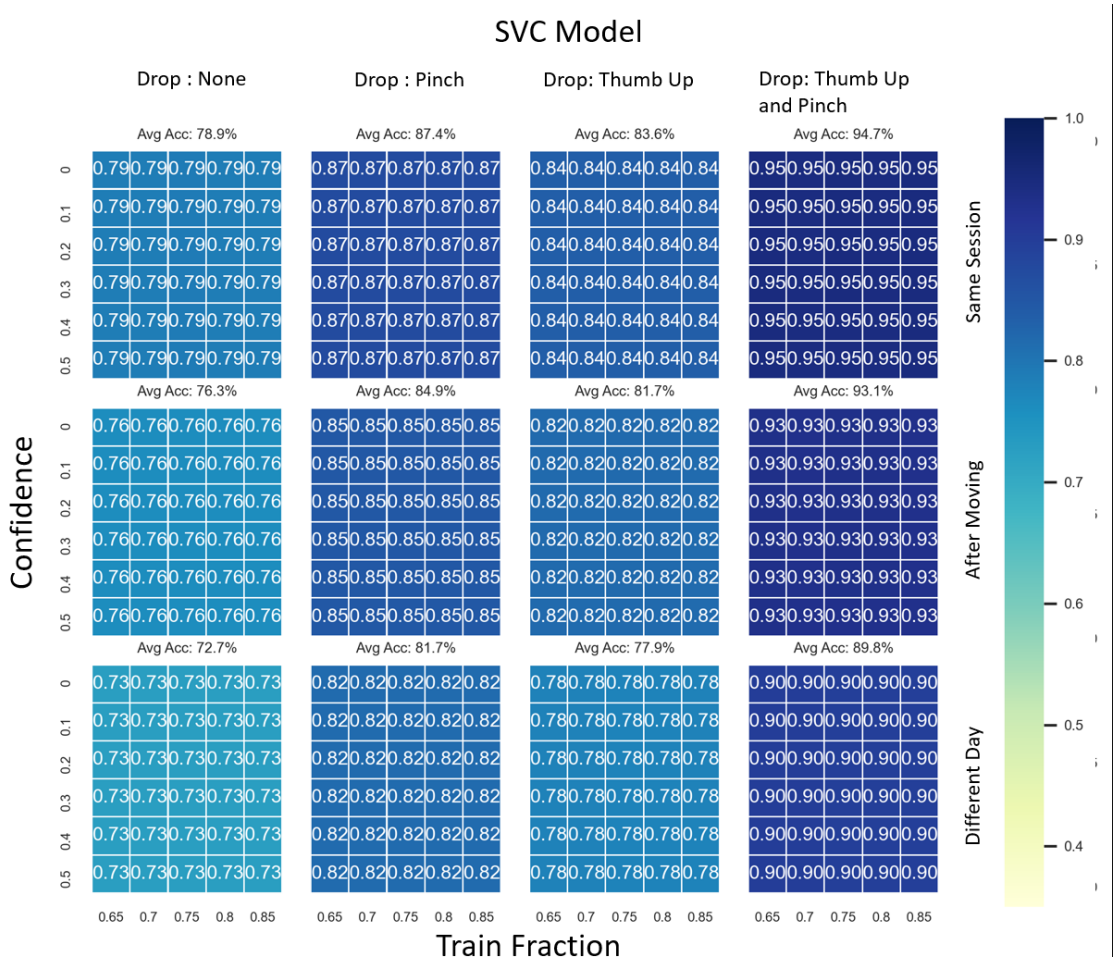


Figure 5.3: Results of modelling with SVC. Globally on the X axis results are grouped by Gestures Used(GU) in training, and Y axis by the Session Label(SL). On individual confusion matrices, the x-axis shows the result of varying the *tr*, and the y-axis shows the results of varying the *tp*. In both cases they can be seen to have a fairly negligible impact compared to SL and GU.

For SVC, accuracy was most impacted by the selection of gestures used and the difference between session/day. When averaging the effect on accuracy across all gesture sets, the accuracy for different session trials was 1.9% lower and accuracy for trials on different days decreased accuracy by 5.7%. When averaging the effect of removing gestures across all session groupings, dropping the Pinch gesture increased accuracy by 8.7%, dropping Thumb Up increased accuracy by 5.1%, and dropping both Pinch and Thumb Up raised accuracy by 16.6%. For SVC the results of varying *tr* and *tp* were negligible, and changed the average accuracy by less than 0.1%.

LDA was significantly less successful predicting gestures. As can be seen in Fig. 5.4, within all groupings it had a far lower accuracy than that observed from SVC. Raising *tp* had a significant deleterious effect on models, but varying the *tr* did not lower the models accuracy.

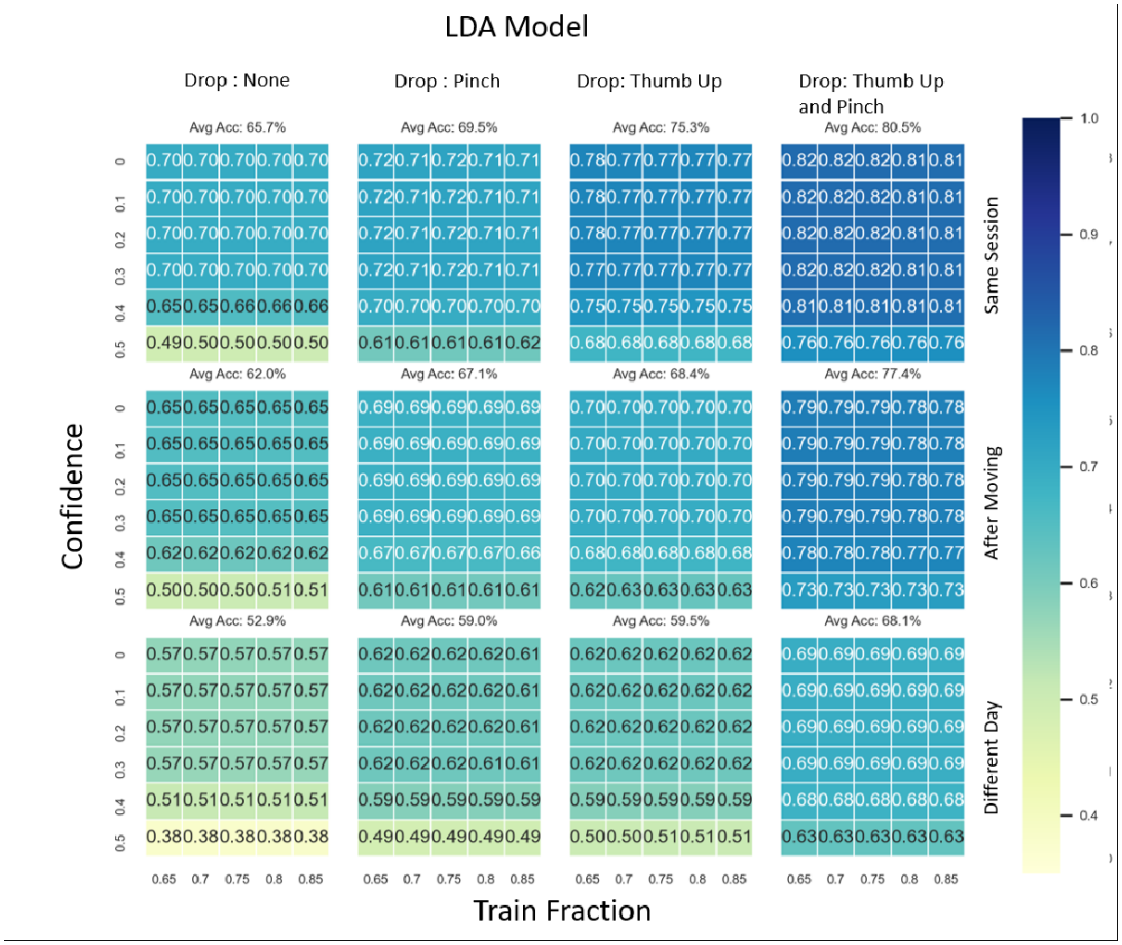


Figure 5.4: The results of training with LDA. Globally on the x-axis results are grouped by Gestures Used(GU) in training, and y-axis by the Session Label (SL). On individual confusion matrices, the x-axis shows the result of varying the *tr*, and the y-axis shows the results of varying the *tp*. Raising the *tp* above 40% significantly lowered the accuracy.

5.4.2 Discussion

While some pairs of datasets reached 100% IT accuracy, the average IT accuracy between same day data sets was 94.7%. As a first step towards exploring this lack of accuracy, all of the data sets were compiled sequentially and each reading was plotted. The results is shown on Fig. 5.5.

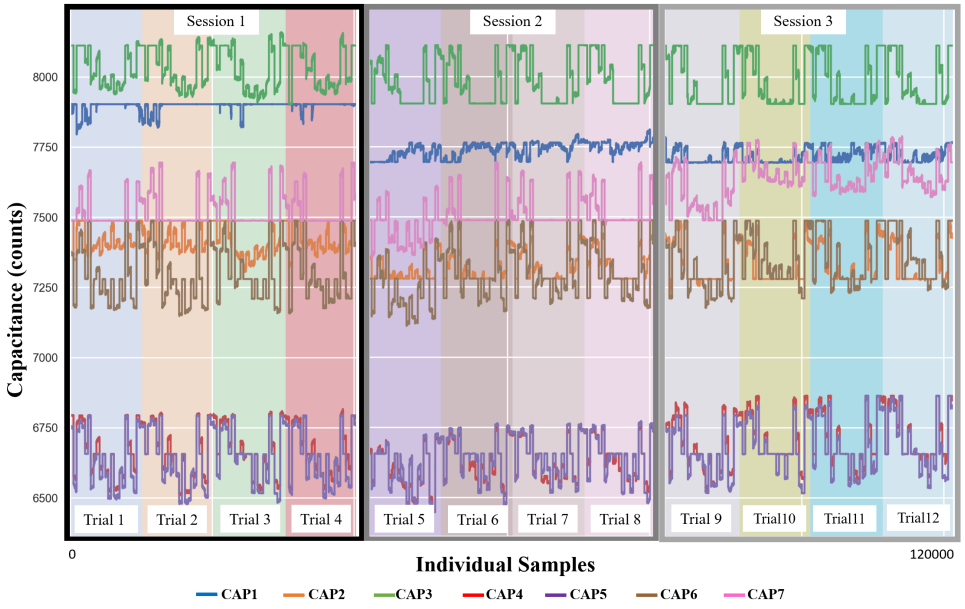


Figure 5.5: This plot shows all the datasets collected and their samples in sequential order. Of particular interest is the fact that the .

Given that the order in which gestures are made and the duration they are held for is constant over the course of all trials, it is reasonable to assume that samples for the individual trials should display the same pattern and range of values. The two sensors closest to the wrist (CAP_1 and CAP_2) show significant variations in their behavior both between sessions and during data sets in the same session.

In the case of CAP_1 during Session 1 significant unexplained variability in its values are observed between data sets. It also displays less dramatic, but equally significant variations in behavior in data sets 5 and 8 of Session 2. It can also be seen that both its minimum and maximum measured values drop significantly on Session 2 and Session 3 as compared to Session 1. CAP_2 is one step further away from the wrist than CAP_1 and also displays some variability that seemingly has little

to do with actual gestures or arm positions in data set 3 where it has notably lower values than the other data sets in Session 1. It is also noted that while its values are relatively constant both between sessions and between data sets for Sessions 2 and 3, the whole range of values displayed in Session 1 from minimum to maximum recorded are significantly higher than those displayed in Sessions 2 and 3

CAP_7 also displays significant fluctuations of its range of values (minimum measured and maximum measured) between Session 1 vs Session 2 and Session 3. As will be discussed further in Chap. 6, these 3 sensors (CAP_1, CAP_2, and CAP_7) are the ones most likely to be significantly effected by changes in elbow angle. It is possible that despite my best efforts, the keyboard platform (which was not marked in tape) shifted between trials. Or that not marking the point on the desk where my middle finger made contact allowed for variations in wrist position to occur without me noticing.

Attempts were made to rerun the modelling with subsets of sensors. For each subset one of the potential problem sensors was removed to observe its affect on the global averages. Removing CAP_2 and CAP_7 actually lowered the accuracy of models. However removing CAP_1 increased the accuracy of all SL and GU groupings by 1% to 2%. This would seem to imply that implies that the fluctuations on CAP_1 could have been responsible for the generally lower SS inter trial accuracy than was observed in Chap. 4. While removing the malfunctioning sensor from models could improve their accuracy, it also gave less data to be used in generating models and so still could not reach the 99% accuracy observed in Chap. 4.

5.5 Conclusions

In this section it was shown that models can persist both over extended wear and still be fairly accurate across sessions on different days. After pruning CAP_1 from the features it was found that for same session IT results an average accuracy of 96% could be achieved. For different sessions IS results were still averaging 95% accuracy. And ID trials could achieve 92% accuracy. This amounts to a 1% degradation of accuracy for IS trials and a 4% degradation of accuracy for ID trials, both of which are considerably lower than the values cited by alternative sensing methods described in the beginning of this chapter. The improvements when CAP_1 was removed from the model could point to the beginning of hardware failure on CAP_1, or just indicate that slight changes in the wrist position were effecting CAP_1 in ways that were less noticeable on the other sensors, or caused by the edge of the desk pressing on the sensor without me noticing.

This highlights the importance of finding a means to determine whether changes in a sensors value are a function of user position, external forces, actual errors occurring on the sensor. User position could be further eliminated by use of an arm brace such as that used in post surgical settings to prevent motion of the subjects elbow. While outside the scope of this study, reliability engineering principles could be used to detect when a given sensor is starting to fail so that information from that sensor could be rejected or the sensor could be replaced. It is also possible that by intentionally moving the joints through their range of motion, models could be made which would account for changes caused by changes in joint angle and used

to differentiate between changes in gesture vs changes in position. This may allow for value changes attributable to motion to be removed prior to running classification models. Given this need and the significant change in baseline values seen on sensors between sessions, Chap. 6 will explore how adding flexion and extension of the elbow as a degree of freedom affects the sensors.

Chapter 6: Gesture Recognition with Elbow Motion as an additional Degree of Freedom

6.1 Introduction

All of the previous results were captured with a seated, stationary user with the elbow angle constrained to approximately a 90 degree bend as seen in Fig. 5.1. This limited muscle activation in the forearm to only what was required for the gestures. However, it is unreasonable to expect users to remain stationary during use of a wHMI device. As seen in Chap. 5, even when users do remain relatively stationary, it is possible that interaction between eternal devices and sensor straps or shifting of their joint position over time can result in a change of the measured capacitance.

This is a known problem in existing technologies and many researchers are still investigating the challenges associated with motion artifacts [12, 48, 82]. Classification error rates have been shown to increase to an average of 35% when attempting grasps in positions different from training data [83]. Current researchers often train devices in multiple positions [12], and/or add data from different limb positions to update the model [61].

A possible different approach would be to gather training data while sweeping a limb through a range of positions while collecting capacitive and IMU data, and develop two models in parallel. One model could correct for changes in sensor values associated with positions, and the corrected data could be used to model gestures.

The goal of this chapter is to test the ability of the capacitive sensors to model the noise caused by elbow motions, reject that noise, and accurately model gestures across multiple arm positions. To do this, an extra degree of freedom is introduced through motion at the elbow joint. The full range of the elbow angles and arm position used in this experiment is discussed in Sec. 6.2.2 and the range of elbow position used in the experiment can be seen in Fig. 6.1. In order to account for this range of motion a hybrid sensor model approach is used. This approach incorporates both the CapSense and an Inertial Motion Unit (IMU) attached to the electronics on the forearm as discussed in Sec. 6.2.1. IMUs are commonly used for motion tracking purposes in a variety of research fields, including sports medicine and rehabilitation [32].

The hypothesis going into this experiment was that the sensor sleeve measurements could be compensated for by knowledge of the elbow angle. In other words, for a given capacitive strap measurement ($\mathrm{CAP_n}$), the portion attributable to the gesture ($\mathrm{CAP_{n,gest}}$) and the portion attributable to change in elbow angle (CAP_{n,θ_y}) would be linear separable. This takes the form of an equation:

$$CAP_n = CAP_{n,gesture} + CAP_{n,\theta_y} \tag{6.1}$$

While neither $CAP_{(n,gesture)}$ nor CAP_{n,θ_y} can be known a priori, it is reasonable to assume that an approximation how the muscle changes associated with elbow angles would effect $\widetilde{CAP}_{n,\theta_y}$ could be modelled via regression techniques. And that the values obtained in these regressions could partially correct for the elbow motion:

$$\widetilde{CAP}_{n,corr} = CAP_n - \widetilde{CAP}_{n,\theta_y} \tag{6.2}$$

This corrected value in turn would allow far more accurate classifications of gestures to be made despite additional motion at the elbow.

6.2 Materials and Methods

In this section, the IMU used will be described in Sec. 6.2.1, the gestures selected for the experiment will be described in Sec. 6.2.2.1 and the collection protocol will be described in Sec. 6.2.2.2. The ML pipeline is described in Sec. 6.2.3. Because it was noted that there was some variation in the elbow angles collected in across the datasets collected as described in a two stage method was used to prepare the data described in Sec. 6.2.3.1 prior to testing. The test protocol used is described in Sec. 6.2.3.

6.2.1 Equipment Used

The IMU selected for this work was a Bosch BNO-055, which has a triaxial 14-bit accelerometer, a triaxial 16-bit gyroscope, and a triaxial geomagnetic sensor. This particular IMU can provide outputs in either quaternions or used the onboard

computational resources to fuse data from the geomagnetic sensor, accelerometer and gyroscope to provide Euler Angles. For this experiment, only the Euler angles $(\theta_x, \theta_y, \theta_z)$ were collected. Fig. 6.1 shows the sensor unit attached to the sleeve and the arm in three different positions which represent the two extremes of the arms elbow motion and the experiments neutral position.

The IMU was integrated into the electronics of the CapSense sleeve, offsetting it somewhat from the center of the both radially and axially. The y component of its Euler angle output (θ_y) corresponded fairly well with the range of motion expected from a normal human arm. θ_y ranged from -87 degrees at full extension to 70 degrees when fully bent for a total sweep of 157 degrees as measured by the IMU. While not perfectly aligned with the arms range of motion (which should not exceed 150 degrees) this was close enough to stand in as a descriptor of the elbow angle.

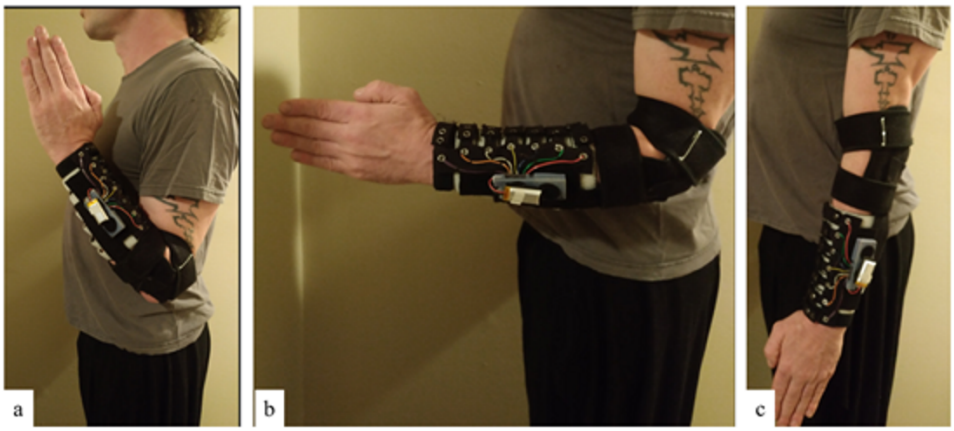


Figure 6.1: The range of motion the users arm moved through in this experiment. For convenience a 90 degree bend of the elbow was considered the 'neutral' position and the room was used as a frame of reference. As such the neutral position corresponding to 0 degrees from the rooms frame of reference a) full flexion corresponds to 70 degrees b) neutral bend corresponding to 0 degrees, c) full extension corresponding to -90 degrees.

6.2.2 Experiment

6.2.2.1 Gestures

In Chap. 4 and 5, it was found that motions involving the thumb were far more difficult for this system to classify. Given that the only digit involved in the thumbs up gesture is the thumb it was dropped from the list of gestures used in this experiment. In its place several more individual finger motions were added in order to evaluate the accuracy of the system in identifying individual finger gestures, and multiple finger gestures. A total of 10 gestures was used in this experiment as shown in Fig. 6.2.

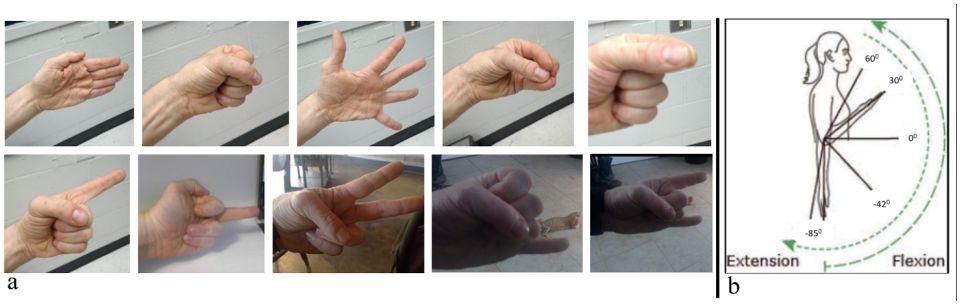


Figure 6.2: a) Gestures used included neutral, fist, spread hand, pincer, pinch, point, middle finger point, victory, pinky point, and horns; b)the range of motion for the elbow image adapted from [69].

6.2.2.2 Protocol

Again limited by COVID-19 restrictions, a single subject was used in this chapter. The subject wrapped their arm in saran wrap as a barrier prior to donning the sensors because it was very humid that day and there was a concern that sweat wicked into the fabric could change its conductivity over the course of testing resulting in significant signal drift. As in previous chapters, data was collected in a seated position. Because the goal of this chapter was to collect data with the added degree of freedom from elbow motion, no elbow support was provided the subject was instructed to hold their upper arm against the side of their body in a natural position and keep their wrist in a neutral position to avoid introducing noise from shoulder and wrist motions.

A companion application was connected to the CapSense via Bluetooth. This application cued gestures and added labels indicating the gesture being prompted. recorded. The incoming sensor data from the CapSense and IMU were recorded at

a 50 Hz frequency This increased frequency was an attempt to compensate for the fact that gestures were no longer stationary for this set of experiments. For each gesture, the subject was instructed to make the gesture prompted by the companion application, and then fully flex and extend their elbow joint 10 times while keeping their upper arm stationary before returning to the neutral elbow position shown in Fig. 6.2b. Each gesture was only cued for 1 time, gesture changes were done in the neutral position, and three seconds were given to transition between gestures. During this three seconds all incoming data from the sensors was labeled as Transitional so that it could be dropped prior to analysis. A total of 7 data sets were collected to gather data for testing.

6.2.3 Machine Learning

The core hypothesis tested in this chapter is that a model using $\widetilde{CAP}_{n,corr}$ as described in Eqn.6.2 can generate a more accurate model of gestures made when the elbow joint is added as a DOF than models using CAP_n and θ_y as features. Testing this hypothesis required a four step protocol: i)first data was observed visually for patterns and irregularities as discussed in Sec. 6.2.3.1, ii) data was normalized as discussed in Sec. 6.2.3.2, iii) Regression models to using various features from the normalized data were tested as predictors for the effect of elbow angle change on the capacitive sensors outputs as descried in Sec. 6.2.3.3, and iv) classification models using CAP_n and $\widetilde{CAP}_{\text{n,corr}}$ were built and compared as discussed in Sec. 6.2.3.4.

6.2.3.1 Data Preparation

As can be seen in Fig. 6.4 the first two datasets displayed significant shifts in values over time between start and end of dataset. It was assumed the source of these shifts were user and/or hardware error and they were pruned.

6.2.3.2 Normalization

Inspection also revealed some variability between datasets observed maximum ($\theta_{y,max}$) and minimum ($\theta_{y,min}$) elbow angles. In spite of the range of motion seeming the same to the subject, these values could vary by as much as 10% between trials. Datasets in which the maximum flexion of the elbow reached a more acute angle displayed higher maximum readings from the capacitive sensors, and datasets in which the elbow joint came closer to maximum extension had smaller minimum readings from the capacitive sensors. As can be seen in Fig. 6.3, the behavior at the maximum ($\theta_{y,max}$) and minimum ($\theta_{y,min}$) elbow angles displayed somewhat non-linear behavior.

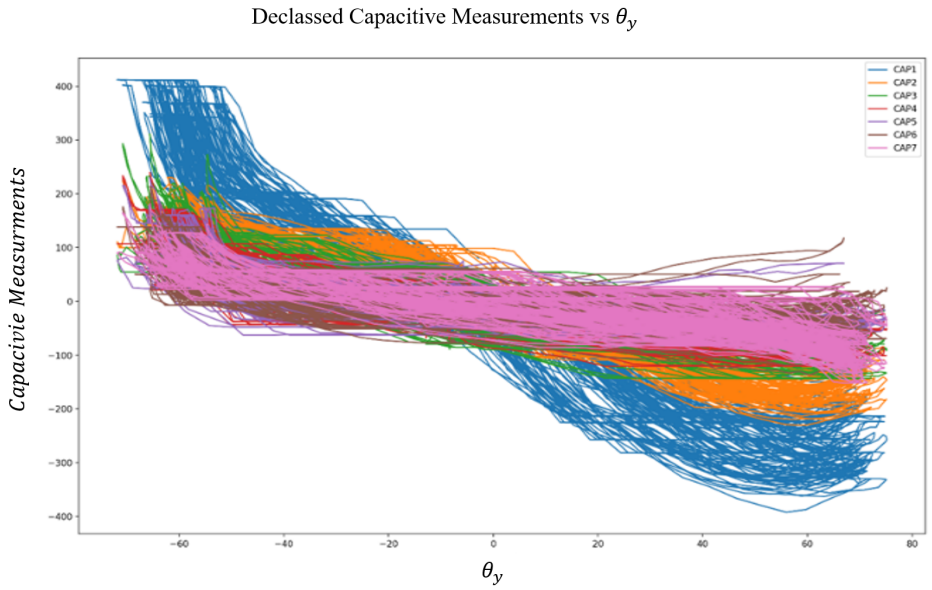


Figure 6.3: In this plot the class data has been removed from the capacitive measurements as discussed in Sec. 6.2.3.2, and for visual clarity they have been centered around 0 by having their means subtracted. When plotted against θ_y, a clear relationship can be seen between the arms angle and the capacitance measured. However it can be seen that this relationship displays non-linear behavior at the extreme angles.

An assumption was made that the variation in angles in the dataset was responsible for the variation in maximum and minimum values observed. In order to account for the possible behavior around extreme angles, capacitive and angle measurements were normalized using the global maximum and minimum values across all datasets used to generate normalized capacitance ($\text{CAP}_{\text{n,normalized}}$) and normalized angular measurements ($\theta_{y,normalized}$). For example, among the θ_y samples the maximum and minimum angles observed across all datasets were $\theta_{y,max}$ =69.2 degrees and $\theta_{y,min}$=-86.6 degrees respectively and these values were used for normalization. In order to explore the relationship between $\theta_{y,normalized}$ and $\text{CAP}_{\text{n,normalized}}$ the com-

ponent of the capacitive measurements attributable to gesture ($CAP_{n,normalized,class}$) had to be removed. While the values cannot be known a priori for new data, it is reasonable to assume that a small training set could be used to generate a mean value for each combination of gesture class and sensor ($\overline{CAP}_{n,class_m}$). With these means can then be used to obtain the scaled capacitive measurements stripped of their class data ($CAP_{n,declass}$) using the following piecemeal equation.

$$CAP_{n,declass} = \begin{cases} CAP_{n,normalized,Class_1} - \overline{CAP}_{n,normalized,Class_1} \\ ... \\ CAP_{n,normalized,Class_m} - \overline{CAP}_{n,normalized,Class_m} \end{cases} \tag{6.3}$$

6.2.3.3 Regression

Support Vector Regression(SVR) is a variation of SVM that performs regressions. Given the success of SVC in the previous chapters, SVR was selected as a candidate model for this chapter. For each capacitive sensor strap two different features sets were tested with SVR. For each sensor, some portion (tr) of the available data from all combined datasets was used to build and test models for $\widetilde{CAP}_{n,\theta_y}$ -the assumed value of the capacitance sensor as a function of θ_y. Two different feature sets were tested for modelling. The first feature set (F_1) tested the assumption that $CAP_{n,\theta_{y,normalized}}$ could be modeled purely a function of $\theta_{y,normalized}$ as in Eqn. 6.4.

$$\widetilde{CAP}_{n,\theta_{y,normalized}} = f(\theta_{y,normalized}) \tag{6.4}$$

The second feature set (F_2) tested the hypothesis that there was useful information about $\widetilde{CAP}_{n\theta_y,normalized}$ embedded in the other capacitive straps and used normalized angular values and normalized values of all the other capacitive straps on the subjects arm as features.

$$\widetilde{CAP}_{n,\theta_y} = f(\theta_{y,normalized}, CAP_{m!=n,normalized}) \tag{6.5}$$

The resulting $\widetilde{CAP}_{n,\theta_y}$ values were compared with $\mathrm{CAP_{n,declass}}$ to validate the models. For each model, the R^2 values of the individual straps as determined via 5-Fold cross validation were recorded, and the average was used to determine overall success of the model.The results are discussed in Sec. 6.3.1. The values obtained from these regression models were used to generate values for $\widetilde{CAP}_{n,corr}$ using Eqn. 6.2 as described in Sec. 6.1. It is worth noting that the *features* fed to these models are normalized rather than rather than declassed. That implies that once a functional model has been developed it should work on new features without a priori knowledge of their gesture class.

6.2.3.4 Classification

Given the results of previous chapters SVC was selected as the model for classification. The results of models using raw capacitive data ($\mathrm{CAP_n}$) and $\widetilde{CAP}_{n,corr}$ were both tested as possible features for the classification models. Given the significant noise observed on $\mathrm{CAP_1}$ and $\mathrm{CAP_2}$ even after applying the methods discussed in Sec. 6.2.3.3 the effect of removing these sensors was also tested, as was the effect

of providing IMU data as an additional feature.

In order to determine how well classification performed outside of the set it was trained on an LPO training scheme was used. Models were trained on a train fraction (tr) of some set (j) of the available trials. Inset accuracy was calculated on the remaining (1-tr) fraction of the data from the j datasets used to train. Out-of-set accuracy was obtained by using the model to classify all data in the (p=1-j) remaining datasets. All possible combinations of j train sets were used with any given tr, and the average in-set and out-of-set accuracy were recorded as metrics along with the confusion matrices for the most accurate and least accurate attempts. In order to determine the minimum quantity of data needed to produce accurate classification models values for tr ranging from 0.1 to 0.8 in steps of 0.05 and values of j ranging from 1 to j-1 were tested. The results of are discussed in Sec. 6.3.2.

6.3 Results and Discussion

6.3.1 Regression Results

The models of $\widetilde{CAP}_{n,\theta_y}$ using the Model$_1$ *features* had mediocre results with only CAP$_1$ and CAP$_2$ achieving R^2 values of over 0.85. Those using the Model$_2$ features had far better predictive power. For all CAP$_n$ the R^2 was significantly higher than Model$_1$. The results achieved by each model are shown on Table 6.1.

Some sensitivity testing was done on tr and it was found that lowering the amount of training / test ratio as far as 10/90 ($tr = 0.1$) would only decrease average R^2 value to 0.832 and the classification accuracy using the resulting CAP$_n$ was only

Sensor	R^2 $Model_1$	R^2 $Model_2$
CAP_1	0.906	0.968
CAP_2	0.835	0.966
CAP_2	0.666	0.931
CAP_4	0.531	0.872
CAP_5	0.040	0.611
CAP_6	0.435	0.72
CAP_7	0.505	0.734
Average	0.562	0.851

Table 6.1: Combinations Tested and Their Results

lowered by 1%.

This seems to imply that the once the sensors have settled into position on the arm, a small set of data collected early in day could be used to create regression models to correct for elbow motion over a long period of time. Of the sensors, CAP_5 had the lowest R^2 value under both methods, a glance at Fig. 6.4 indicates that ironically it also had the least noise to begin with and for some gestures displayed no appreciable noise at all. Attempts to improve the quality of noise removal achieved by regression included adding $\theta_{x,normalized}$ and $\theta_{z,normalized}$ as features, and testing with other models.

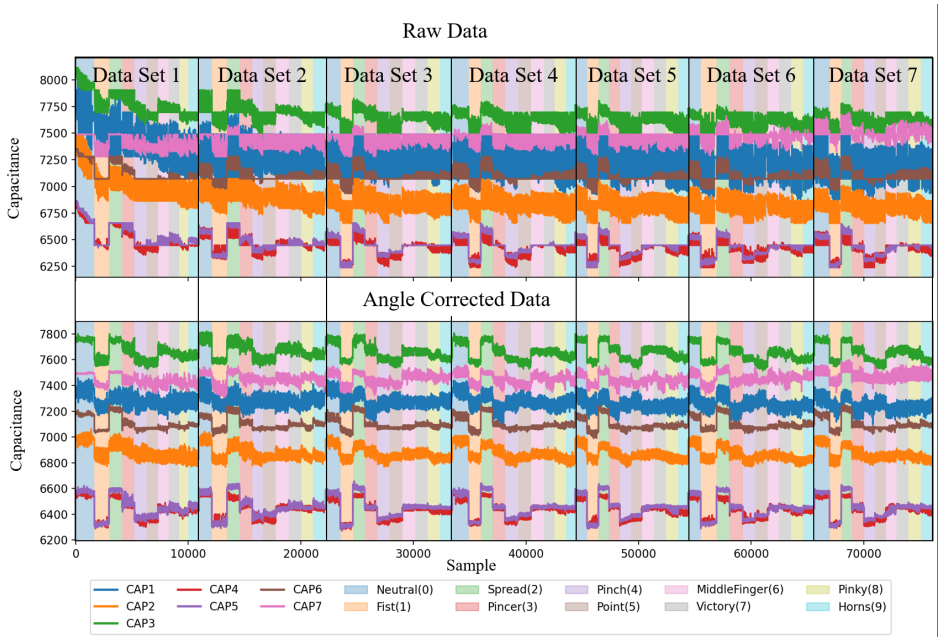


Figure 6.4: The top image is the raw data in the order it came in labeled with datasets and color coded by gesture. Data sets 1 and 2 show levels of noise, and extreme values not observed in the other datasets. The individual sensor values also drop over time in ways not observed in the other datasets, possibly indicating that the sensors shifted/settled on the arm over the course of these datasets. The bottom image shows the sensor readings after the portion associated with the elbows angle change is stripped from them. While dataset 1 and 2 still show some artifacts not observed in the other datasets, all datasets now have observable patterns that could be attributed to individual gestures.

As can be seen on Tab. 6.1, the sensors closest to the wrist (CAP_1, CAP_2) and elbow (CAP_6, CAP_7) had the highest R^2 values. It makes intuitive sense that the effect of elbow position on the sensors closest to the elbow would be relatively easy to model given that the insertion point of the bicep on the forearm is almost directly under them. It makes less intuitive sense that they had lower R^2 values than the sensors closest to the wrist. The elbow brace portion of the hardware may explain the lower R^2 values. At the more acute elbow angles, the elbow brace

impeded motion somewhat and it required more work from the bicep to achieve further flexion of the elbow. The amount by which the sensors closest to wrist (CAP_1, CAP_2) were affected by elbow motions was unexpected. There are three possible root causes which could explain this.

Firstly the elbow brace and strap configuration used to maintain position (pictured in Fig. 6.1) may have transferred forces applied at the elbow all the way down length of sensor. A second potential explanation is that the sensor assembly is non-elastic, and not skin-tight, so it is possible that it slid over surface of the arm that during each cycle of flexion and extension that was also measured and modelled. A final possibility to consider is that the wrist has 3-DOF of its own, flexion/extension, pronation/supination, and ulnar/radial deflection. While tracking wrist motions fell outside the scope of this work, and motions of the wrist may have contributed additional noise to the system and using parallel models to correct for wrist motion should be explored in future works.

6.3.2 Classification Results

Ultimately, the goal is still to be able to do accurate gesture classification. In order to test whether the regressions done to correct for elbow angles improved classification accuracy models using $CAP_{n,normalized}$ as their *features* were compared with models using $\widetilde{CAP}_{n,corr}$. In Fig. 6.5 and 6.6 the results of models classifying all ten gestures using three datasets (j=3) and a ($tr = 0.5$) split are shown.

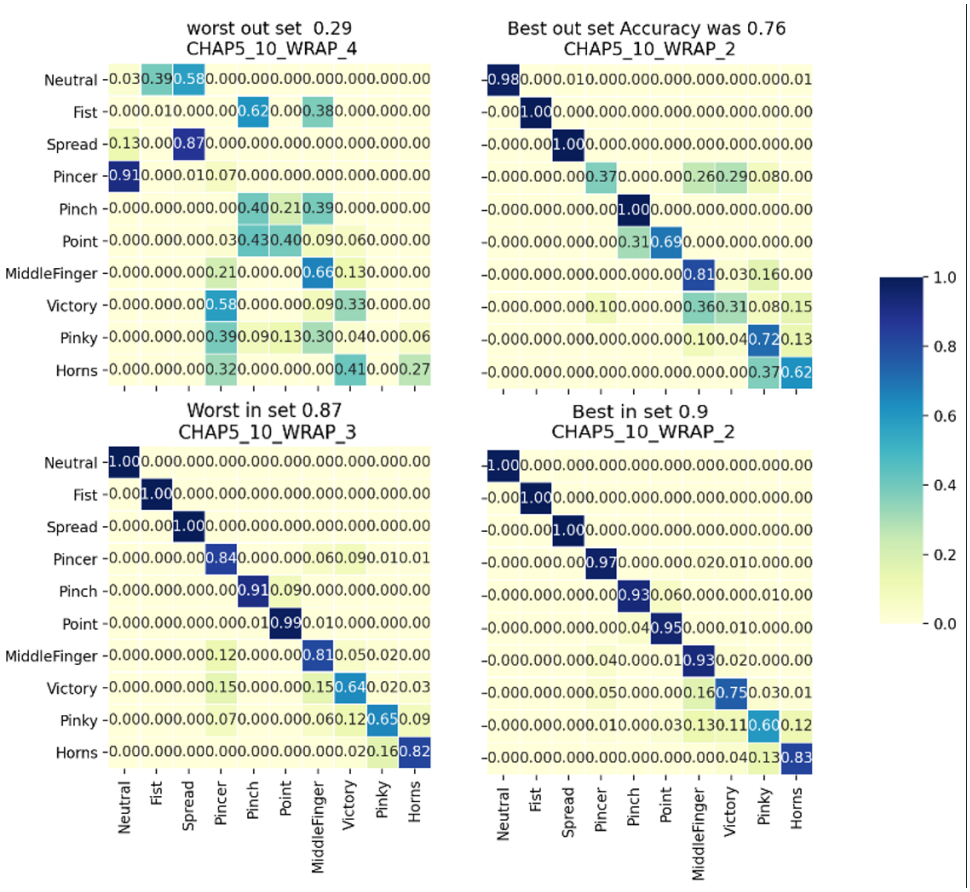


Figure 6.5: Results from classification with $\text{CAP}_{\text{n,normalize}}$ *features*.

The first model results shown in Fig. 6.5 was trained only on the $\text{CAP}_{\text{n,normalized}}$ *features* and only achieved an average accuracy of 73%. While the second model was trained on the corrected $\widetilde{CAP}_{\text{n,corr}}$ *features* achieved an out of set accuracy of 82% as shown in Fig. 6.6. In these figures the top row displays confusion matrixes out-of-set predictions and the bottom row displays inset predictions. The left column contains the least accurate results and the right contains the most accurate.

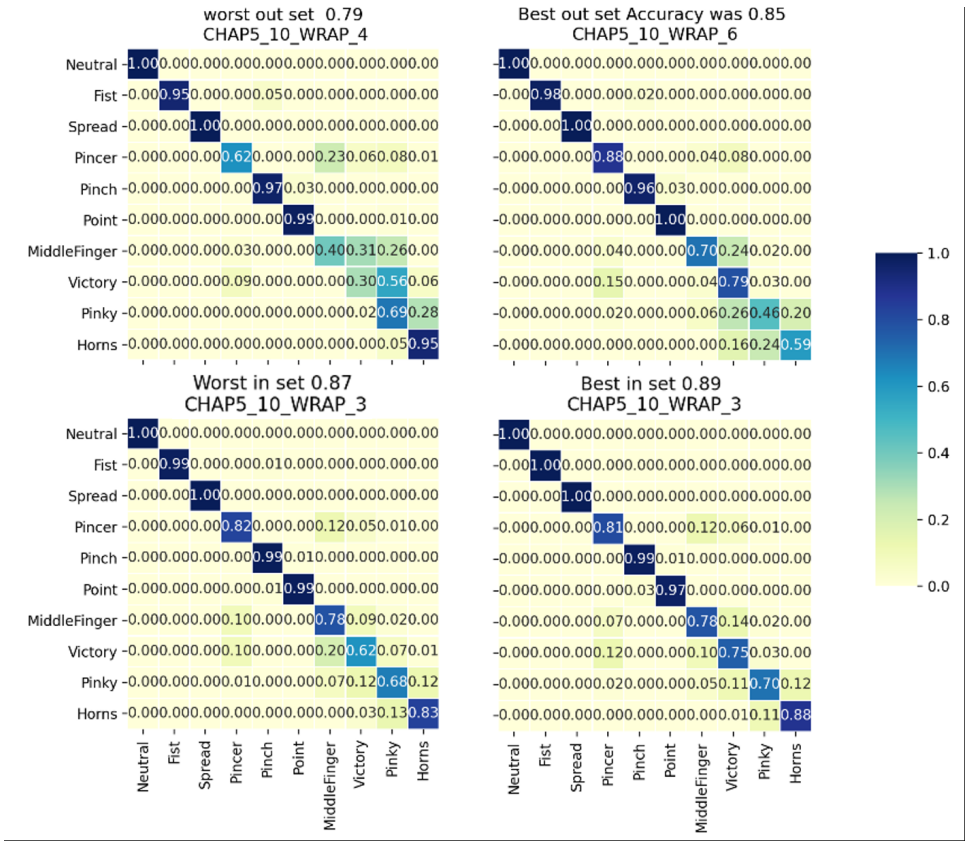


Figure 6.6: Results from classification with $\text{CAP}_{\text{n,corr}}$ *features*.

The noise removal seen in Fig. 6.4 was not just cosmetic as can be seen from the nearly 10% rise in accuracy for models using $\widetilde{CAP}_{\text{n,corr}}$ for classification. Focusing on the angle-corrected results in Fig. 6.6 it can be seen that there were clusters of gestures that the models had difficulty distinguishing between. It was noted that often the multiple finger motions were confused with their individual finger components. The Victory gesture was often confused with middle finger extension (MFE) and little finger extension (LFE). All three gestures are shown in Fig. 6.7a.

Horns was often incorrectly classified as either LFE or Victory. This group of gestures is shown in Fig. 6.7b. The confusion between Horns and LFE is not

surprising as the little finger is used in both gestures.

The difficulty in distinguishing between Victory and LFE is a little surprising, but could be explained in part by the fact that the little finger has limited independent motion, and to some extent both the ring and middle finger move during LFE.

Finally, the Pincer gesture was sometimes confused the MFE and Victory. This grouping can be seen in Fig. 6.7c. The middle finger and index finger are both components of the Pincer gesture, which could in part explain this confusion.

It was also observed anecdotally that when moving the arm to full extension, the thigh somewhat blocked both pinching and pincer motions which led to some flexion and extension of wrist to complete full range of elbow motion.

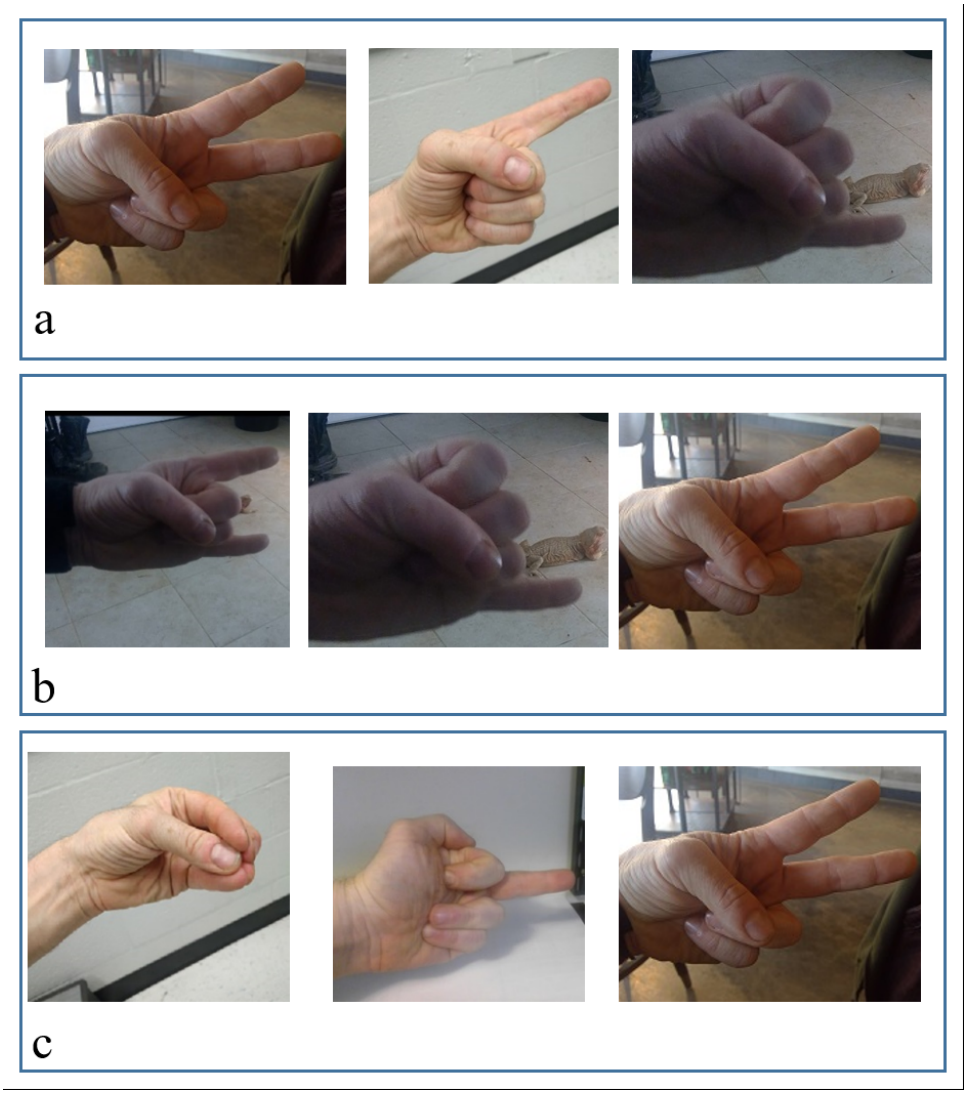


Figure 6.7: The most commonly confused groups of gestures were a) victory vs pinky and middle finger extension, b) horns vs pinky extension and victory c) the pincer vs middle finger extension and victory.

In order to further clarify results, two subsets of gestures were created and tested among. In Gesture Set 1 multi-finger gestures such as Pincer, Victory, and Horns were removed. This resulted in a 95% average accuracy and is shown in Fig. 6.8.

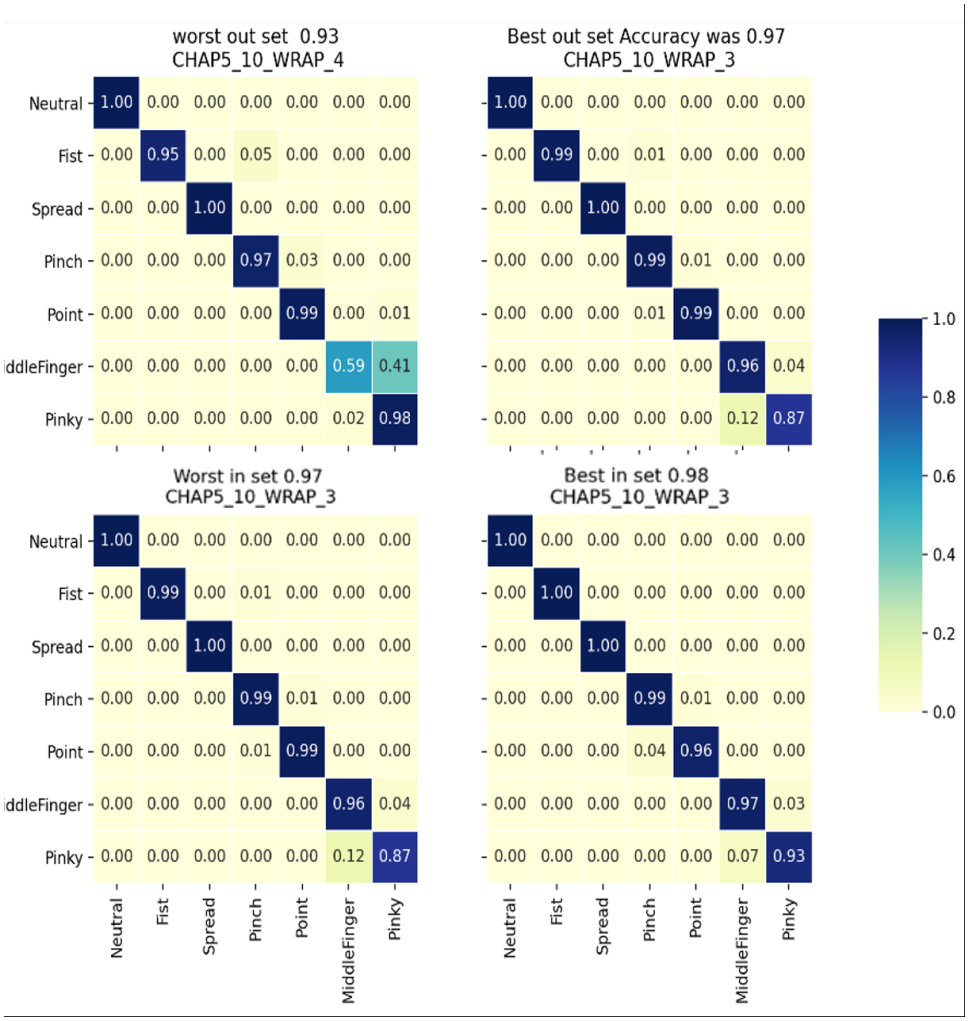


Figure 6.8: The results of dropping multiple DOF gestures.

In Gesture Set 2 the single finger gestures such point, MFE, and LFE were removed. This resulted in only modest improvements, achieving an overall accuracy of 93%. The results of modelling this gesture set are displayed in Fig.6.9.

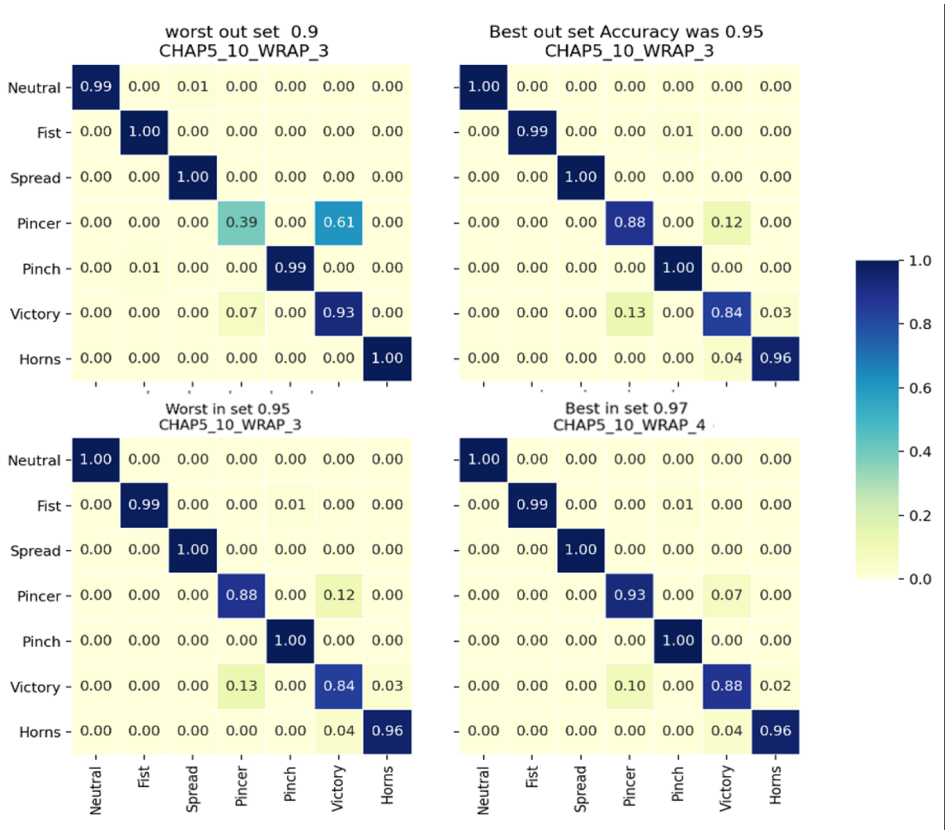


Figure 6.9: The results of dropping single DOF gestures.

Significant accuracy improvements were achieved by limiting the model to just the top level gestures and individual finger motions, though the model still had difficulty with the little finger. This may pave the way for classifying the motion of individual fingers and building more complex top-level models of gesture and motion from these finger classifications.

In order to assess how much data was needed, the values for both tr and j were varied. It was found that at j=*2* reduced accuracy to 93% and j=*1* reduced out-set accuracy to 91%. So long as tr was in the range of $0.5 < tr < .75$ there was no change in accuracy. Adding in the IMU data as extra features had no effect

whatsoever on the accuracy of models.

6.4 Conclusions and Future Work

6.4.1 Conclusions

In this chapter flexion and extension of the elbow was added as a degree of freedom. An SVR model was developed to predict the component of capacitive measurements associated with elbow motions and achieved an average R^2 of 0.851. Using this SVR model to correct for elbow motion increased the classification accuracy by 9% for models examining all 10 classes as compared to models trained on the uncorrected capacitive measurements. Further examination showed some of the new classes added in this chapter had difficulty in distinguishing between multi-finger gestures and single finger gestures. Narrowing classes examined down to 7 classes composed of top level gestures and individual finger gestures led to an accuracy of 95% accuracy for 7 classes of gestures.

Looking back to Chap. 4, in static gestures with 7 classes an accuracy of 91% was achieved, and with 5 classes an accuracy of 99% was achieved. The results of this chapter represent a 4% improvement in IT classification with 7 selected classes as compared to Chap. 4, even with a DOF of elbow motion added in as a confounding factor. To put this in the context of current work, the results of some contemporary studies will be briefly compared.

Beaulieu et al were able to decrease multi-position classification error rates among 5 gestures tested in different positions from 17.2% (model trained in a single

position) down to 10.3% by introducing examples of a gesture from different training positions[12].

Moin et al saw accuracy decrease by as much as 30% when the position of the arm was changed and were able to partially overcome this by injecting data from the new position to update the model. This data was given a 50% heavier relative weight in the new model which was retrained. After this the model could recognize gestures with only a 3% loss of accuracy [61]. This proof of concept may pave the way for real world applications of this technology.

6.4.2 Future Work

Interesting directions for future work include adding more degrees of freedom for arm motion including wrist and shoulder motion, building individual models for finger positioning, testing multiple output algorithms to determine if combined finger outputs can be used to model more complex gestures, and applying domain adaption learning models to see if trials performed on different days can be used to classify results on each other [31].

Chapter 7: Finger and Whole Grasp Forces

7.1 Introduction

The previous chapters focused on recognizing a particular gesture or grasp, but focused only on the kinematics and not the strength of the grasp. For example, a fist can be closed lightly, or tightly with much more force. For many applications, determining the pressure being exerted in grasps is an important capability. The amount of force a muscle is applying is often an important feature in many systems that require human machine interaction. In exoskeleton applications it has been investigated as a means of applying proportional assistance [12, 38], in physical therapy as an input to teach patients recovering from stroke to modulate force [35, 37, 58, 71], and in prosthetics to allow a single gesture pattern to control multiple grasp states [15, 82].

While EMG and FMG have both shown success in identifying individual finger force components of grasp, a variety of challenges make them difficult to apply for this purpose outside of the lab [9, 19, 50, 89]. A 2020 review by Campbell et al on sEMG control for prosthetics pointed to a 20% reduction in accuracy in gesture classification when force levels within the gestures were varied [18].

This chapter will explore using the capacitive sensors to create models for a)

the overall measured grasp force of the closed hand gesture as measured by force sensitive resistors (FSR) as described in Sec. 7.2. and b) the forces created by fingers acting independently of each other as described in Sec. 7.3.

7.2 Full Hand Grasps

7.2.1 Materials and Methods

The full grasp force tests were an initial proof of concept to show that regressions could be performed to determine the total force in a grasp and the component forces of individual fingers. For these experiments, a glove instrumented with Force Sensitive Resistors (FSR) (Interlink 30-81794) attached by Velcro to allow for repositioning as needed provided ground truth on finger force and whole palm force as shown in Fig. 7.1. A 1" diameter pipe was mounted in a vice clamped to the desk as seen in Fig. 7.1a. The datasheet for these FSR indicates that these sensors can detect a range of forces ranging from 0.1 N (0.023 lb-f) to 100 N (27 lb-f) in 0.1N increments.

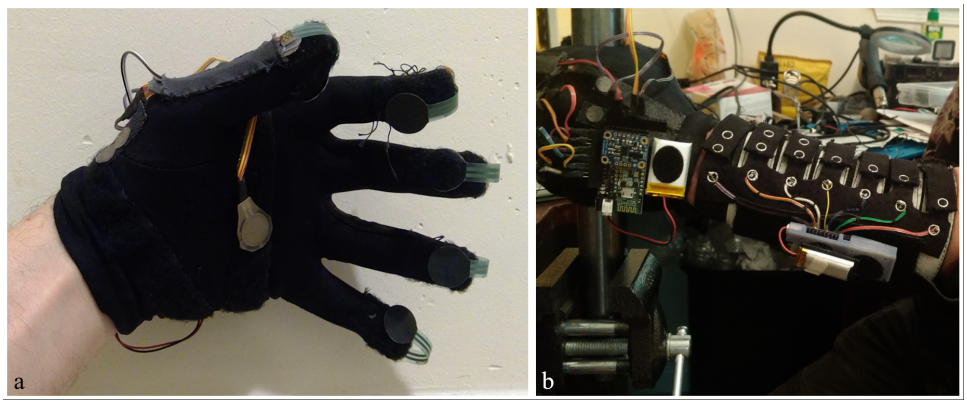


Figure 7.1: Setup used in full hand grasp experiments a) palm view of glove instrumented with force sensitive resistors, b) grasping 1" diameter steel pipe used in the experiment.

7.2.2 Experiment

A 1" diameter pipe was clamped to the desk with a vice as shown in Fig. 7.1b to provide a solid surface to grasp. The user was seated at a comfortable distance from the desk grasping the pipe which was squeezed and released 10 times while data was recorded for offline processing using the application built for Chap. 3. Sample data from the full grasp tests is shown in Fig. 7.2.

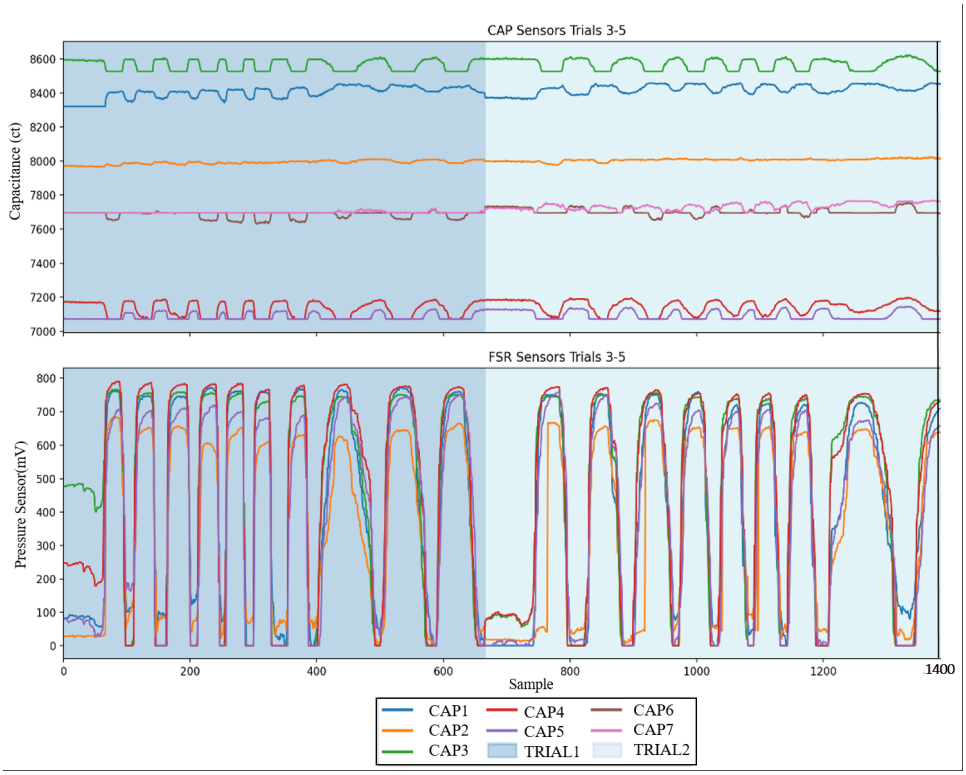


Figure 7.2: Data from the two trials used for grasp force analysis. The top image shows the values recorded by CapSense. Note that sensors CAP2, CAP6 and CAP7 display a drift in values over the course of this experiment.The lower image shows measurements from the force sensitive resistors on the fingertips.

7.2.3 ML Protocol

A model was created for each individual finger and for the palm. A 75/25 train/test split (tr=.75) was used to train Standardization Scalars and models on the first data set, the second data set was then scaled on the trained scalars and the scaled data fed to the models as features in order to validate performance. SVR was used again in this chapter as a candidate model due to its success in Chap. 6, and Automatic Relevance Determination (ARD) and Bayesian Ridge (BR) are described

in the introduction in Sec. 1.2.2. Given that we are working with regressions rather than classification problems R^2 was selected as a metric.

7.2.4 Results and Discussion

Looking at Fig. 7.2 slight, but continuous drift over the course of both trials can be observed on sensors CAP_2, CAP_6, and CAP_7. This may be indicative of slight elbow motion over course of tests. These tests were performed chronologically prior to Chap. 5 and Chap. 6 and IMU data was not recorded.

Because models using all sensors (S_{All}) performed somewhat poorly; a second set of models (S_{Lim} was created using a limited feature set that dropped the previously mentioned sensors on which drift was observed. The S_{Lim} feature set used included: CAP_1, CAP_3, CAP_4, and CAP_5). The results of both models in predicting full grasp forces are shown on Tab. 7.1.

ML	Sensors	Ind. R^2	Mid. R^2	Ring R^2	Lit. R^2	Palm R^2	Avg R^2
BR	S_{Lim}	0.939	0.837	0.796	0.819	0.920	0.862
ARD	S_{Lim}	0.938	0.836	0.799	0.815	0.918	0.861
SVR	S_{Lim}	0.902	0.759	0.839	0.866	0.910	0.856
BR	S_{All}	0.807	0.344	0.830	0.809	0.926	0.734
ARD	S_{All}	0.797	0.317	0.832	0.811	0.916	0.733
SVR	S_{All}	0.526	0.435	0.574	0.557	0.589	0.536

Table 7.1: Combinations Tested and Their Results

Note the especially poor performance of SVR for both feature sets. A possible explanation is that both ARD and BR use posterior probabilities probability to eliminate/lower the importance of unreliable features. This may have allowed both of the Bayesian methods to ignore sensors which were noisy or undependable.

The S_{Lim} feature set can produce R^2 values above 0.9 for the palm of the hand, and R^2 values of over 0.85 when averaged across all fingers and the palm for all algorithms considered. Fig. 7.3 shows a sample set of real values versus predictions using the S_{Lim} feature set.

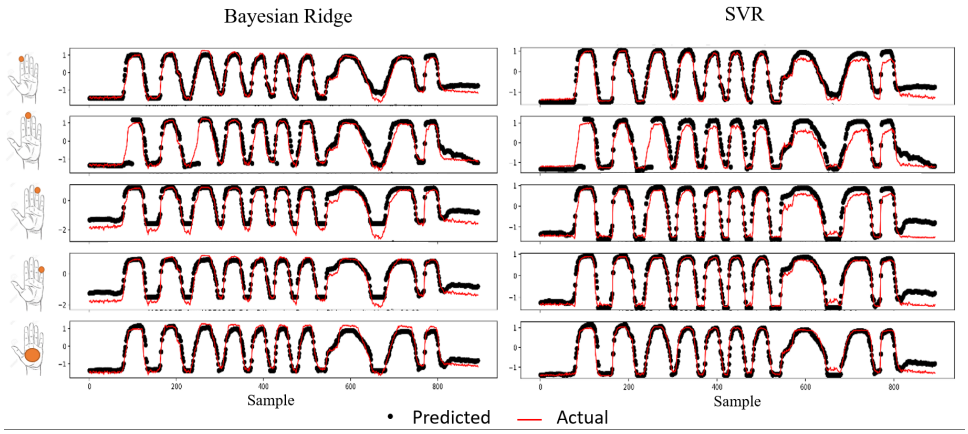


Figure 7.3: The by finger results of modelling normalized force sensor data with BR (on left) and SVR (on right). When using the more limited S_{Lim} feature set both achieve reasonable accuracy.

For the most part, the predicted forces using S_{Lim} are quite close to the measured ground truth from the FSRs. While these results show promise, the fact that all fingers are acting in unison may have made prediction a simpler task.

7.3 Independent Finger Forces

The goal of these tests was to determine if the finger forces could still be mapped to regressions, or at least to bins of force levels when the fingers were acting independently of each other.

7.3.1 Materials and Methods

A set of PCB Piezotronics piezoelectric force sensors were mounted on a rig shown in Fig. 7.4. Unfortunately, 5 sensors of the same model were not available; Table 7.2 lists the sensors used for each finger and their parameters.

Finger	Model	SN	Range	Sensitivity
Thumb	208M182	22789	Unavailable	Datasheet Unavailable
Index	208C01	29102	44.48 N (10 lb)	$(\pm 15\%)500\frac{mV}{lb}(112410\frac{mV}{kN})$
Middle	208C01	29030	44.48 N (10 lb)	$(\pm 15\%)500\frac{mV}{lb}(112410\frac{mV}{kN})$
Ring	208M182	22785	Unavailable	Datasheet Unavailable
Little	208C01	17593	44.48 N (10 lb)	$(\pm 15\%)500\frac{mV}{lb}(112410\frac{mV}{kN})$

Table 7.2: Sensors Used in Individual Finger Force Tests

The rig as seen in Fig. 7.4a had grooves to ensure constant finger positions over the course of multiple trials and was clamped to a desk so that it would remain stationary when finger forces were applied. A NI USB-6221 DAQ was used to connect the sensors to a computer for data collection.

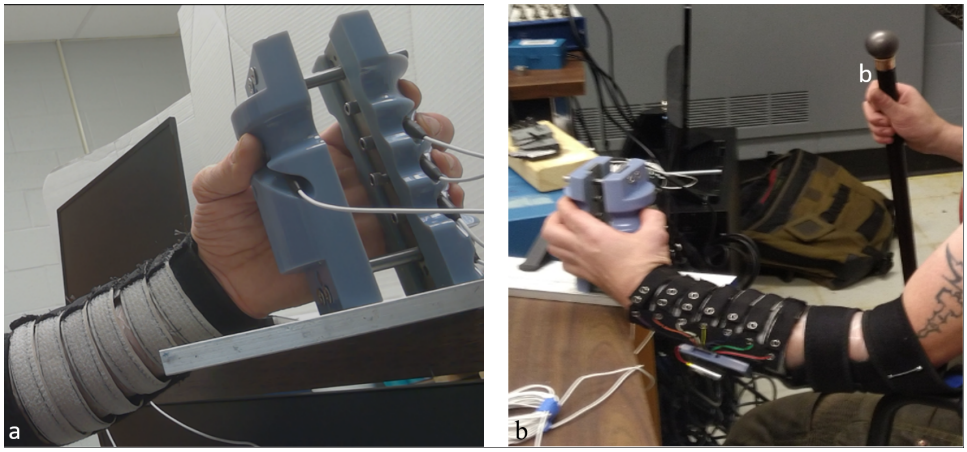


Figure 7.4: Independent finger forces setup a) sensor equipped rig b) arm resting on chair and position of rig relative to subject.

7.3.2 Experiment

The subject was seated at the desk with their arm resting on the chairs elbow as seen in Fig. 7.4b, and a comfortable distance from the sensor with their fingers resting on the sensors. A custom NI LabView 2017 application created for this experiment cued the subject to apply force with a finger or combination of fingers. The application then recorded samples from the Piezzoelectric and CapSense sensors at a 50Hz rate, labeled them with the cued gesture, and stored them in a .csv file for offline analysis. This study collected data on a total of ten combinations including: the individual fingers (including thumb) by themselves, each individual finger used in opposition to thumb, and all finger grasps.

7.3.3 ML Protocol

The data required some preparation before it could be used in modelling. As can be seen in Fig.7.5 the piezoelectric sensors are prone to signal drift.

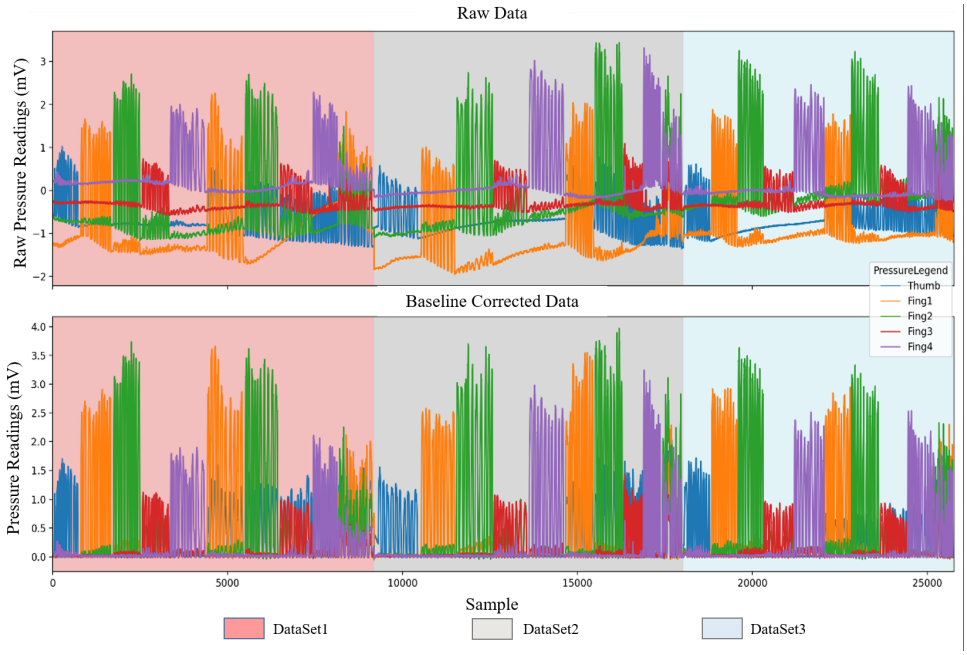


Figure 7.5: Independent finger forces setup a) sensor equipped rig b) arm resting on chair and position of rig relative to subject.

Given the observation that the FSR base values rose continuously over the course of a given day all datasets taken on a given day were imported. The method proposed by Zhang et al for spectral correction was then used to remove baseline drift from the piezoelectric data prior to using it as ground truth [95]. Within the independent finger force (IDF) experiments, two distinct groupings of fingers were considered. Group I models were trained using the force outputs of the 4 fingers as *targets* because of how poorly models had tracked the thumb in previous chapters.

Group II used the 4 fingers and the thumb as targets while training models.

For both groups, a model was trained for each finger in the group using only the IDF samples, ie those in which finger acted individually. The test data used to validate models included samples of both IDF and the combined finger force (multiple finger grasps, pinches, etc) in order to test the hypothesis that the summed individual finger forces from models would be equal to the summed ground truth forces measured. Eqn. 7.1 shows how models were summed, and Eqn. 7.2 shows how ground truths were summed.

$$Total_{model} = \sum_{n=min}^{Little} model_{force} \tag{7.1}$$

$$Total_{forces} = \sum_{n=min}^{Little} FSR_{zhang,normalized} \tag{7.2}$$

In these equations 'Little' represents the little finger force, and 'min' represents the furthest finger from the little finger included in the group (i.e. the index finger for Group 1 and thumb for Group 2).

Two different approaches were used to model IDF. Both approaches used the groupings discussed above as targets, drew their features from the CapSense measurements, and followed the LNO method discussed in previous chapters. In other words, models were trained using *features* and *targets* from one dataset, and used the remaining datasets to validate, repeating until each dataset had been use for training and tested against the remaining datasets.

The difference between the methods was in the type of modelling used. The first approach used regression based modelling with the CapSense as features to predict the individual finger forces as continuous values. Based on the results of the whole hand force trials, both SVR and BR were tested as possible models for individual finger force regressions and R_2 was used as a performance metric.

A second approach divided the FSR data for each finger divided into N equal sized bins representing discrete force levels levels within range of force measured from the finger over the course of testing. Candidate values for N included 2,3,4,5. For these binned classification models SVC and LDA were both investigated as candidate algorithms and accuracy was used as a metric.

For both approaches, the resulting metric is presented as an average of all values within a grouping.

7.3.4 Results and Discussion

The attempts at using regressions on individual fingers were inaccurate at best. The inter-trial results reached a maximum average R^2 value of 0.286 across all of the Group II fingers with a modest improvement to 0.296 when only Group I fingers were observed. To explore whether results could be improved by using force levels and classification, each finger was divided into N equal sized bins denoting force levels from 0 to maximum force observed on a given force sensor. The accuracy using these various force bins is show in Fig. 7.6.

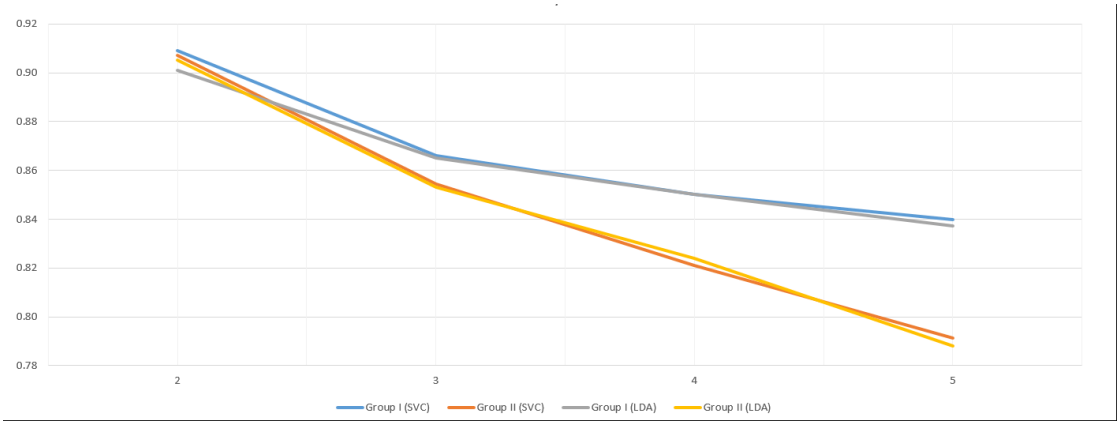


Figure 7.6: Bins used vs accuracy of prediction (averaged over all fingers in a grouping).

For the majority of bins, SVC was more accurate than LDA, and both algorithms had difficulty in accurately parsing forces coming from the thumb (which was not entirely surprising given the results of previous chapters). For both algorithms there is a significant decline in accuracy when going from N=2 to N=*3* and then a more gradual decrease in accuracy thereafter. Even at 5 bins accuracy is still above 80% for SVC based models.

To put these results into the context of existing work, another pair of studies will be briefly described. The Leone study used 6 sEMG sensors as *features* to estimate class, Force Sensitive Resistors (FSR) on the fingertips of a glove as ground truth, and employed a two stage classifier both stages of which used a Non-linear Logistic Regression algorithm. The first stage classified a gesture from among 7 different classes (i.e., Rest, Spherical, Tip, Platform, Point, Wrist supination, and Wrist pronation). In this study the Rest position corresponded to Open in my studies, Spherical grip equated to a Closed hand, the Tip grip equated to Pinch,

Platform equated to Spread, Point was the same as in my study, and supination and pronation were both performed in open hand position. If the gesture was judged to either be a 'Spherical' or 'Tip' grasp then forces within it were classified by a separate model into three different force levels (Low, Medium, Firm) corresponding to $< 45\%, 45\% - 75\%, and > 75\%$ of the total combined force of all fingertip FSR. They achieved a base level of 98.8% accuracy for gesture recognition. With three level Sphere and Tip force models their average accuracies were 98.8% and 97.64% respectively [50].

Baldaccino et al used publicly available EMG databases to build their models. They used a multivariate Bayesian Mixture of experts to analyze force and EMG data available for nine combinations of finger forces available in the NinaPro (Non-invasive Adapative Prosthetics) database. The Bayesian MOE model is a form of ensemble learning applying individual Bayesian models to each possible combination of forces. It uses an unsupervised learning approach in which a Gaussian Mixture Model (GMM) is used to select a number of clusters equivalent to the combinations expected. Two different types of results were presented. The first set gave metrics in accuracy which only considered if individual fingers or combinations of fingers were applying force. For single finger motions and accuracy of 87.89% was achieved, however the accuracy dropped to 62.39% for multifinger [9].

When limited to three bins, the work in this chapter was only 3% less accurate than the Leone study, and when limited to a binary measure of whether force was applied, it was 3% more accurate than the Baladaccino study.

It is unclear whether these results represent a limitation of the technology or

a problem in the actual experimental setup. The 208M182 sensors under the thumb and index finger required significantly more force than the other sensors to detect force. In addition, for all sensors it was observed that the real time response shown on screen by the data collection application often did not match the perceived force applied to the sensor, particularly in pinch and all finger grasps where sometimes the forces applied did not register at all.

All of the desks available to mount the sensor assembly on in the lab were fairly light, and when applying forces independently the table could be observed moving/shaking. This could have introduced further random noise to the data, and possibly led to inconsistent positioning over the course of trials.

Of the seven datasets initially collected, four had to be thrown out because significant table motion was observed. Even on the datasets that were retained, less obvious motions could have been a source of error. Had time permitted, recollecting these data sets with the equipment on a heavier table may have improved results. As would obtaining another pair of 208C01 sensors.

7.4 Conclusions and Future Work

7.4.1 Conclusion

Regression was fairly successful on the first set of experiments, and could obtain R^2 values of above 0.86 on average in spite of the drift observed in some of the sensors.

The results in the second experiment with independent finger motions were not

as promising. In these experiments, regression proved useless and only a binary classification of active versus inactive fingers yielded accuracies of above 90%. However, each finger could be separated into as many as five bins while still maintaining an average accuracy of over 84% across the entire hand.Further testing will be needed in order to determine whether the results of the second set of tests represents a limitation of the system, or a limitation of the experimental setup.

7.4.2 Future Work

These experiments should be revisited with new sensors, a stable table, and possibly a different set of gestures given the a priori knowledge that the thumb is a source of error. While equally sized bins were used this time around, it may be worth investigating whether accuracy could be improved by fine tuning where the cutoff points for each bin are.

Chapter 8: Conclusions and Contributions

In Chapter 2, CapSense was developed as a novel wHMI using cross sectional measurements of the forearm rather than point sensors. It was shown that these sensors can be constructed using readily available off the shelf components, and achieve accuracy comparable to state of the art research using sEMG sensors. This chapter was originally published in the IEEE ICRA conference proceedings [24] and awarded as the Best Paper on Human Robot Interaction. This work also led to a patent through the university Tech Transfer Office.

In Chapter 3, construction materials, shielding, and connection method were explored. It was found that shielding is largely unnecessary for this technology, and that snap connections do not introduce a source of noise, which should allow for configurations in wide variety of shapes and materials to be built and tested using same base electronic. It was also found that neoprene; while attractive on paper; is a poor substrate for constructing these sensors. A more rigorous inter-trial test method was introduced as a performance metric more suited to real world applications than n-fold cross validation. The results of this chapter were less accurate than those initially seen in Chapter 2, but the methods provide a path forward. Currently discussions are ongoing with Advanced Functional Fabrics of America (AFFOA), a

non-profit Institute headquartered near the Massachusetts Institute of Technology (MIT) to test additional elastomeric substrates and conductors for future versions of the CapSense.

In Chapter 4, the effect of microcontroller settings, learning models, and gesture selection was explored to improve our understanding of the systems limitations. A significant flaw was found in the systems inability to accurately model thumb motions and a hybrid sensor device was proposed. This hybrid device would make an interesting future study as COVID restrictions wind down and subjects become more readily available. Research on novel gesture recognition devices has seen publication in journals such as Nature [4], Nature Electronics [61], Journal Rehabilitation Research Development[89]. There are also several IEEE journals including Robotics Automation Letters [71], or Journal of Translational Engineering in Health and Medicine [77], or Sensors Journal[97]. Any of these could be reasonable outlets for the results of these experiments.

Chapter 5 took a look at the results of sessions after extended wear and on separate days to test how robust the models developed in Chapter 4 were. This is an important step toward application of this technology in the field.

Chapter 6 introduced an additional degree of freedom in the form of elbow motions. A procedure was developed to predict the likely change in signal caused by elbow motions and correct for them. When implemented this procedure raised inter-trail accuracy for tests in which elbow motions occurred by a 10%, and allowed 7 gestures to be classified with an accuracy of 95%. This study needs more subjects to produce a final written document. Work reviewing the effect of motion has seen

publication in journals such as IEEE Reviews in Biomedical Engineering [14] and the Journal of Prosthetics and Orthotics [12]. The journals mentioned previously for the results of Chap. 4 would also be reasonable avenues of publication.

Finally, in Chapter 7, preliminary results for grasp force regressions were explored and showed promising results, though further studies and more subjects will be needed to explore their potential.

8.1 Overall Contributions

The primary contributions of this work are as follows.

- Developed a method of using the capacitive noise generated in my attempts to use strain sensors as a signal to classify gestures
- Refined initial concept into a very inexpensive ($17 to build) sensor that can be created with conventional off the shelf parts
- Showed 99% accuracy on a selection of 5 gestures made while arm was stationary using conventional machine learning methods
- Determined that primary limitation of system in its current embodiments is classification of thumb gestures
- Created method for cancelling motion artifacts from elbow motion and showed 95% accuracy in classifying 7 gestures while arm was in motion
- Showed that independent motions of fingers can be recognized with capacitive straps

- Found that force regressions in whole hand grasps can achieve R^2 values of 0.92 for forces measured at palm and 0.86 average R^2 across all fingers
- Achieved 84% accuracy in modelling independent finger forces by dividing into 5 discrete bins of force levels

8.2 Published Papers

- L. Dankovich and S. Bergbreiter, Gesture recognition via flexible capacitive touch electrodes, in Proceedings - IEEE International Conference on Robotics and Automation, 2019, vol. 2019-May
- *(in preparation)* L. Dankovich and S. Bergbreiter, Using a fusion of capacitive sensing and IMU to compensate for elbow motion in gesture recognition
- *(in preparation)* L. Dankovich and S. Bergbreiter, Capacitive sensing as a non contact means of determining grip forces

8.3 Future Work

An interesting use case for CapSense that flows from this chapter lies in determining if muscles have relaxed. Discussions with the Sheikh Khalifa Stroke Institute Endowed Chair Dr. Raghavan at Johns Hopkins indicated a significant challenge in her work is recognizing when muscles prone to hypertonicity (a common condition underlying claw hand palsy) have relaxed. It is theorized that if muscle relaxation can be observed, then it would be a useful early indicator for the success of a given

therapeutic path even before the full effects of the therapy have been achieved. A joint research project is planned to investigate the ability of the deice to perceive muscle relaxation in healthy patients given an injected solution to relax specific muscles.

Outside of the upper body, work is currently being done with the CapSense system on the lower limbs by Owen Pearl along with Prof. Eni Halilaj at Carnegie Mellon University to determine feasibility of device for gait analysis.

Bibliography

[1] P. Agarwal, and A. D. Deshpande, Exoskeletons, in *Human Performance Optimization*, Oxford University Press, (2019).

[2] M.A. Ahmed, B. B. Zaidan, A. A. Zaidan, M. M. Salih, and M. M. Bin Lakulu, A Review on Systems-Based Sensory Gloves for Sign Language Recognition State of the Art between 2007 and 2017. *Sensors*, vol. **18(7)** (2018).

[3] K. Anam, and A. A. Al-Jumaily, Active Exoskeleton Control Systems: State of the Art, *Procedia Eng.*, vol.. **41**, (2012).

[4] O.A. Araromi, M. A. Graule, K. L. Dorsey, S. Castellanos, J. R. Foster, W. H. Hsu, A. E. Passy, J. J. Vlassak, J. C. Weaver, C. J. Walsh, and R. J. Wood, Ultra-sensitive and resilient compliant strain gauges for soft machines, *Nature*, vol. **587(7833)**, (2020).

[5] K. Al-jabery, T. Obafemi-Ajayi, G. Olbricht, and D. Wunsch II, Ed., *Computational Learning Approaches to Data Analytics in Biomedical Applications*, Academic Press, (2019).

[6] D. O. Amft, H. Junker, P. Lukowicz, G. Trster, and C. Schuster, Sensing muscle activities with body-worn sensors, *International Workshop on Wearable and Implantable Body Sensor Networks*, (2006).

[7] M. Atzori, A. Gijsberts, S. Heynen, A. M. Hager, O. Deriaz, P. Van der Smagt, C. Castellini, B. Caputo, and H. Muller, "Building the NINAPRO Database: A Resource for the Biorobotics Community", *Fourth IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, (2012).

[8] M. Atzori, and H. Mller , Control capabilities of myoelectric robotic prostheses by hand amputees: A scientific research and market overview, *Frontiers in Systems Neuroscience*, vol. **9**, (2015).

[9] T. Baldacchino, W. R. Jacobs, S. R. Anderson, K. Worden, and J. Rowson, Simultaneous Force Regression and Movement Classification of Fingers via Surface EMG within a Unified Bayesian Framework, *Front. Bioeng. Biotechnol.* , vol. **6**, (2018).

[10] R. Baldwin, S. Bobovych, R. Robucci, C. Patel, and N. Banerjee, Gait analysis for fall prediction using hierarchical textile-based capacitive sensor arrays, *Design, Automation Test in Europe Conference Exhibition (DATE)* , (2015).

[11] I. Batzianoulis, S. El-Khoury, E. Pirondini, M. Coscia, S. Micera, A. Billard, Emg-based decoding of grasp gestures in reaching-to-grasping motions, *Robotics and Autonomous Systems*, vol. **91**, (2017).

[12] R. J. Beaulieu, M. R. Masters, J. Betthauser, R. J. Smith, R. Kaliki, N. V. Thakor, and A. B. Soares, Multi-Position Training Improves Robustness of Pattern Recognition and Reduces Limb-Position Effect in Prosthetic Control, *J. Prosthetics Orthot.* , vol. **29(2)**, (2017).

[13] T. R. Bennett, J. Wu, N. Kehtarnavaz, and R. Jafari, Inertial measurement unit-based wearable computers for assisted living applications: A signal processing perspective, *IEEE Signal Processing Magazine*, vol. **33(2)**, (2016).

[14] J. L. Betthauser, C. L. Hunt, L. E. Osborn, M. R. Masters, G. Lvay, R. R. Kaliki, and N. V. Thakor, Limb Position Tolerant Pattern Recognition for Myoelectric Prosthesis Control with Adaptive Sparse Representations from Extreme Learning, *IEEE Trans Biomed Eng.*, vol. **65(4)**, (2018).

[15] J. Bowker, (J. I. Krajbich, M. S. Pinzur, B. K. Potter, and P. M. Stevens, Eds.), *Atlas of Limb Prosthetics: Surgical, Prosthetic, and Rehabilitation Principles*, American Academy of Orthopedic Surgeons, (2018).

[16] D. Bzdok, M. Krzywinski, and N. Altman, "Machine learning: Supervised methods, SVM and kNN," *Nature Methods*, (2018).

[17] D. Cai, X.He, and J. Han, Training linear discriminant analysis in linear time, *Proceedings - International Conference on Data Engineering*, (2009).

[18] E. Campbell ,A. Phinyomark and E. Scheme, "Current Trends and Confounding Factors in Myoelectric Control: Limb Position and Contraction Intensity," *Sensors (Basel)*, (2020).

[19] C. Castellini, and R. Koiva, Using surface electromyography to predict single finger forces, *Proc. of the IEEE RAS and EMBS Intl. Conf. on Biomedical Robotics and Biomechatronics*, (2012).

[20] Z. Chen, R. Wu, S. Guo, X. Liu, H. Fu, X. Jin, and M. Liao, 3D Upper Body Reconstruction with Sparse Soft Sensors, *Soft Robotics*, vol. **8(2)**, (2021).

[21] J. Cheng, O. Amft, and P. Lukowicz, Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition, *Pervasive Computing*, (2010).

[22] O. Comacho, and E. Viramontes, *Designing Touch Sensing Electrodes: AN3863 Electrical Considerations and Recommended Layout Patterns*, Freescale Semiconductor, (2011).

[23] M. Cutkosky, On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, vol. **5(3)**, (1989).

[24] L. J. Dankovich, and S. Bergbreiter, Gesture recognition via flexible capacitive touch electrodes, *Proceedings - IEEE International Conference on Robotics and Automation*, (2019).

[25] T. Dankowski, and A. Ziegler, Calibrating random forests for probability estimation, *Statistics in Medicine*, vol. **35**(22),(2016).

[26] C. J. De Luca, *Surface Electromyography: Detection and Recording*, Delsys Incorporated, (2002).

[27] L.E. Dunne, B. Smyth, and B. Caulfield, A comparative evaluation of bend sensors for wearable applications, *Proceedings - International Symposium on Wearable Computers, ISWC*, (2007).

[28] N/A Relative Permittivity - the Dielectric Constant. Engineeringtoolbox.com. $https://www.engineeringtoolbox.com/relative-permittivity-d_1660.html$, (accessed Jan, 15, 2021).

[29] F.Fabris, A. Doherty, and D. Palmer, A new approach for interpreting random forest models and its application to the biology of ageing, *Bioinformatics*, vol. **34(14)**,(2018).

[30] J. Fang, B. Xu, X. Zhou, and H. Qi, Research on gesture recognition based on sEMG and inertial sensor fusion. *Proceedings of 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference*, (2018).

[31] A. Farahani, B. Pourshojae, K. Rasheed, and H. R. Arabnia, "A Concise Review of Transfer Learning," *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, (2020).

[32] M. Georgi, C. Amma, and T. Schultz, Recognizing hand and finger gestures with IMU based motion and EMG based muscle activity sensing, *BIOSIGNALS 2015 - 8th International Conference on Bio-Inspired Systems and Signal Processing, Proceedings*, (2015).

[33] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov Neighbourhood Components Analysis, *Advances in Neural Information Processing Systems*, vol. **17**, (2005).

[34] T. Grosse-Puppendahl, C. Holz, G. Cohn, R. Wimmer, O. Bechtold, S. Hodges, M. Reynolds, and J. Smith, Finding common ground: A survey of capacitive sensing in human-computer interaction, *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, (2017).

[35] J. Guerra, J. Uddin, D. Nilsen, J. Mclnerney, A. Fadoo, I. B. Omofuma, S. Hughes, S. Agrawal, P. Allen, and H. M. Schambra, Capture, learning, and classification of upper extremity movement primitives in healthy controls and stroke patients, *International Conference on Rehabilitation Robotics (ICORR)*, (2017).

[36] T. Hastie, R. Tibshirani, J. Friedman, Ed. *Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction (12th ed.)*, Springer, (2007).

[37] F. C. Huang, and J. L. Patton, Movement distributions of stroke survivors exhibit distinct patterns that evolve with training, *Journal Neuroengineering and Rehabilitation*, vol. **13(1)**, (2016).

[38] S. Jiang, P. Kang, X. Song., B. Lo, and P. B. Shull, Emerging Wearable Interfaces and Algorithms for Hand Gesture Recognition: A Survey, *IEEE Reviews in Biomedical Engineering*, (2021).

[39] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. **2**, (1995).

[40] S. Jiang, B. Lu, W. Guo, C. Zhang, H. Wang, X. Sheng, and P.B. Shull, Feasibility of wrist-worn, real-time hand, and surface gesture recognition via sEMG and IMU Sensing, *IEEE Transactions on Industrial Informatics*, vol. **14(8)**, (2018).

[41] O. Jones, Fig 1.1 Cross section of the muscles of the distal forearm. Some extensor muscles, such as the anconeus, are not visible as they are situated proximally in the forearm. `http://teachmeanatomy.info/`, (Accessed March 2017).

[42] M. Jordan, J. Kleinberg, and B. Schlkopf, *Pattern Recognition and Machine Learning*, Springer, (2006).

[43] A. Kadkhodayan, X. Jiang, and C. Menon, Continuous Prediction of Finger Movements Using Force Myography, *Journal of Medical and Biological Engineering*, vol. **36(4)**, (2016).

[44] S. M. Khan, A. A. Khan, O. Farooq, Selection of features and classifiers for EMG/EEG based upper limb assistive devices - A review, *IEEE Reviews in Biomedical Engineering*, (2019).

[45] Freescale Semiconductor *K20 Sub-Family Reference Manual Supports: MK20DX64VLH7, MK20DX128VLH7, MK20DX256VLH7.*, Freescale Semiconductor, (2015).

[46] J. R. Koller, D. A. Jacobs, D. P. Ferris, and C. D. Remy, Learning to walk with an adaptive gain proportional myoelectric controller for a robotic ankle exoskeleton, *Journal Neuroengineering and Rehabilitation*, vol. **12(1)**, (2015).

[47] T. A. Kuiken, M. M. Lowery, and N.S. Stoykov, The effect of subcutaneous fat on myoelectric signal amplitude and cross-talk, *Prosthetics and Orthotics International*, vol.**27**, (2003).

[48] I. Kyranou, S. Vijayakumar, and M. S. Erden, Causes of performance degradation in non-invasive electromyographic pattern recognition in upper limb prostheses, *Frontiers in Neurorobotics*, vol. **12** , (2018)

[49] D. Lay, *Linear Algebra and its Applications (3rd ed.*, Adisson Wesley, (2002).

[50] F. Leone, C. Gentile. A. L. Ciancio, E. Gruppioni, A. Davalli, R. Sacchetti, E. Guglielmelli and L. Zollo, Simultaneous sEMg classification of hand/wrist gestures and forces, *Frontiers Neurorobotics*, vol. **13**, (2019).

[51] X. Liang, H. Li, W. Wang, Y. Liu, R. Ghannam, F. Fioranelli, and H. Heidari, Fusion of wearable and contactless sensors for intelligent gesture recognition, *Advanced Intelligent Systems*, vol. **1(7)**, (2019).

[52] H. S. Lo, and S. Q. Xie, Exoskeleton robots for upper-limb rehabilitation: State of the art and future prospects, *Medical Engineering Physics*, vol. **34(3)**, (2012).

[53] Z. Lu, K. Tong, X. Zhang, S. Li, and P. Zhou,"Myoelectric Pattern Recognition for Controlling a Robotic Hand: A Feasibility Study in Stroke," *IEEE Transactions on Biomedical Engineering*, vol. **66(2)**, (2019).

[54] P. Maceira-Elvira, T. Popa, A. C. Schmid, and F. C. Hummel, Wearable technology in stroke rehabilitation: towards improved diagnosis and treatment of upper-limb motor impairment, *Journal of Neuroengineering and Rehabilitation*, vol. **16(1)** (2019)

[55] A. Marques, For Medical Exoskeletons Access is the Next Step. Exoskeletonreport.com. `https://exoskeletonreport.com/2019/04/for-medical-exoskeletons-access-is-the-next-step/` (accessed June 2019).

[56] D. Martinez, *AN4330 Touch Sensing Software Using TSI Module.* Freescale Semiconductor, (2011).

[57] G. Marzloff, "Using Myoware a Low Cost Surface Electro Myography Sensor for Developing Rehabilitative Devices," *Association of Academic Physiatrists.*, Medium.com. `https://medium.com/physiatry/using-myoware-a-low-cost-surface-electromyography-sensor-for-developing-rehabilitation-devices-1d04a16f5396` (accessed June 2017).

[58] C. Meeker, S. Park, L. Bishop, J. Stein, and M. Ciocarlie, EMG Pattern Classification to Control a Hand Orthosis for Functional Grasp Assistance after Stroke, *IEEE International Conference Rehabilitative Robotics*, (2017).

[59] C. Meeker, T. Rasmussen, and M. Ciocarlie, Intuitive Hand Teleoperation by Novice Operators Using a Continuous Teleoperation Subspace, *Proceedings - IEEE International Conference on Robotics and Automation*, (2018).

[60] B.Milosevic, S. Benatti, and E. Farella, Design Challenges for Wearable EMG Applications, *Proceedings of the Conference on Design, Automation and Test in Europe*, (2017).

[61] A. Moin, A. Zhou, A. Rahimi, A. Menon, S. Benatti, G. Alexandrov, S. Tamakloe, J. Ting, N. Yamamoto, Y. Khan, F. Burghardt, L. Benini, A. C. Arias, and J. M. Rabaey, A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition, *Nature Electronics*, vol. **4**, (2021).

[62] N/A,Muscles of the thumb. Eastonhand.com. `http://www.eatonhand.com/mus/mus131.htm`. (accessed March 2021).

[63] N/A,Motion Plus Clinic-Stay Supple. Motionplutosteo.com. `https://www.motionplusosteo.com /blog/2017/9/25/troublesome-at-the-base-of-the-thumb ?rq=thumb`. (accessed March 2021).

[64] N/A,Muscles of the Posterior Forearm — Anatomy — Geeky Medics. Geekymedics.com. `https://geekymedics.com/muscles-of-the-posterior-forearm/`. (accessed March 2021).

[65] W. Navidi, *Statistics for Engineers and Scientists, 3rd ed.*. McGraw Hill, (2011).

[66] D. Novak and R. Reiner A survey of sensor fusion methods in wearable robotics., *Autonomous Systems*, vol.**73**,(2015).

[67] A. E. Olsson, P. Sager,E. Andersson, A. Bjrkman, N. Maleevi, and C. Antfolk Extraction of Multi-Labelled Movement Information from the Raw HD-sEMG Image with Time-Domain Depth, *Scientific Reports*, vol. **9(1)**, (2019).

[68] G. Onose, N. Popescu, C. Munteanu, V. Ciobanu, C. Sporea, M. D. Mirea, C. Daia, I. Andone, A. Spnu, and A. Mirea, Mobile mechatronic/robotic orthotic devices to assist-rehabilitate neuromotor impairments in the upper limb: A systematic and synthetic review, *Frontiers Neuroscience*, vol. **12**, (2018).

[69] A. M. Oosterwijk, M. K. Nieuwenhuis, C. P. van der Schans, and L. J. Mouton, Shoulder and elbow range of motion for the performance of activities of daily living: A systematic review, *Physiotherapy Theory and Practice*, vol. **34(7)**, (2018).

[70] N. Parajuli, N. Sreenivasan, P. Bifulco, M. Cesarelli, S. Savino, V. Niola,G. D. Gargiulo, Real-time EMG based pattern recognition control for hand prostheses: A review on existing methods, challenges and future implementation. *Sensors*, (2019).

[71] S. Park, C. Meeker, L. M. Weber, L. Bishop, J. Stein, and M. Ciocarlie, Multimodal Sensing and Interaction for a Robotic Hand Orthosis, *IEEE Robotics Automation Letters*, vol. **4(2)**, (2019).

[72] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M.Perrot, and E. Duchesnay,"Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, (2011).

[73] W. H. Press, *Numerical Recipes 3rd Edition: The Art of Scientific Computing (3rd ed.)*, Cambridge University Press, (2007).

[74] T. Proietti , V. Crocher, A. Roby-Brami, and N. Jarrass, Upper-limb robotic exoskeletons for neurorehabilitation: A review on control strategies, *IEEE Reviews in Biomedical Engineering*, vol. **9**, (2016).

[75] Y. Qi, T. Minka, R. Picard, and Z. Ghahramani, "Predictive Automatic Relevance Determination by Expectation Propagation," *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, (2004).

[76] A. Radmand, E. Scheme, and K. Englehart, High-density force myography: A possible alternative for upper-limb prosthetic control, *Journal Rehabilitation Research and Development* , vol. **53(4)**, (2016).

[77] A. Ramadan, A. Roy, and E. Smela, "Recognizing Hemiparetic Ankle Deficits Using Wearable Pressure Sensors", *IEEE Journal of Translational Engineering in Health and Medicine*, (2019).

[78] D. Rano, M. A. Chaudhary, and M. S. Hashmi, A New Model to Determine Effective Permittivity and Resonant Frequency of Patch Antenna Covered with Multiple Dielectric Layers, *IEEE Access*, vol. **8**, (2020).

[79] V Ravindra, and C. Castellini, A comparative analysis of three non-invasive human-machine interfaces for the disabled, *Frontiers in Neurorobotics*, vol.**8**, (2014).

[80] F. Reverter, X. Li, and G. C. M. Meijer, Stability and accuracy of active shielding for grounded capacitive sensors, *Measurement Science and Technology*, vol. **17**, (2006).

[81] G. Saggio, F. Riillo, L. Sbernini, and L. R. Quitadamo, Resistive flex sensors: a survey, *Smart Materials and Structures*, vol. **25**, (2016).

[82] E. Scheme, and K. Englehart, Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use, *Journal of Rehabilitation Research and Development*, vol. **48(6)**, (2011).

[83] E. Scheme, B. S. Hudgins, and K. B. Englehart, "Confidence-Based Rejection for Improved Pattern Recognition Myoelectric Control," *IEEE Transactions Biomedical Engineering* vol. **60(6)**, (2013).

[84] M. Simao , N. Mendex, O. Gibaru, and P. Neto, A Review on Electromyography Decoding and Pattern Recognition for Human-Machine Interaction. *IEEE Access*, vol. **7**, (2019).

[85] S. F. D. Toro, S. Santos-Cuadros, E. Olmeda, C. lvarez-Caldas, V. Daz, and J.L. San Romn, Is the Use of a Low-Cost sEMG Sensor Valid to Measure Muscle Fatigue?, *Sensors*, vol. **19(14)**, (2019).

[86] D. Wang, *SNOA926A Capacitive Sensing: Ins and Outs of Active Shielding.* Texas Instruments. (2015).

[87] F. Wilhelm, Explaining the Idea behind ARD and Bayesian Interpolation, forianwilhelm.info. `https://florianwilhelm.info/2016/03/explaining_the_idea_behind_ard`. (accessed April 2021).

[88] M. Wilhelm, D. Krakowczyk, F. Trollmann, and S. Albayrak, ering: Multiple finger gesture recognition with one ring using an electric field, *Proceedings of the 2nd International Workshop on Sensor-based Activity Recognition and Interaction, iWOAR 15*, (2015).

[89] M. Wininger, N.-H. Kim, and W. Craelius, Pressure signature of forearm as predictor of grip force, *Journal Rehabilitation Research Development*, vol. **45(6)**, (2008).

[90] D. Wipf, and S. Nagarajan,"A New View of Automatic Relevance Determination," *Advances in Neural Information Processing Systems*, vol. **20**, (2007).

[91] Y. Wu, D. Jiang, J. Duan, X. Liu, R. Bayford, and A. Demosthenous, Towards a high accuracy wearable hand gesture recognition system using EIT, *IEEE International Symposium on Circuits and Systems (ISCAS)*, (2018).

[92] Z. G. Xiao, and C. Menon, Performance of Forearm FMG and sEMG for Estimating Elbow, Forearm and Wrist Positions, *Journal of Bionic Engineering*, vol.**14(2)**, (2017).

[93] Z. G. Xiao, and C. Menon, A review of force myography research and development, *Sensors*, vol. **19(20)**, (2019).

[94] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello, Review of assistive strategies in powered lower-limb orthoses and exoskeletons, *Robotics and Autonomous Systems*, vol. **64**, (2015).

[95] Z. M. Zhang, S. Chen, and Y. Z. Liang, Baseline correction using adaptive iteratively reweighted penalized least squares, *Analyst*, vol. **135(5)**, (2010).

[96] X.Zhang. X. Chen, Z. Zhao, Y. Q. Tu, J. H. Yang, V. Lantz, and K. Q. Wang, Research on gesture definition and electrode placement in pattern recognition of hand gesture action SEMG, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol **4901**, D. Zhang, Ed. Berlin, Heidelberg Springer, (2008).

[97] Y. Zhang, Y. Huang, X. Sun, Y. Zhao, X. Guo, P. Liu, C. Liu, and Y. Zhang, "Static and Dynamic Human Arm/Hand Gesture Capturing and Recognition via Multiinformation Fusion of Flexible Strain Sensors," *IEEE Sensors Journal*, vol.**20(12)**, (2020).

[98] E. Zheng, B. Chen, W. Wang, K. Wei, and L. Wang, A wearable capacitive sensing system with phase-dependent classifier for locomotion mode recognition, *4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, (2012).

[99] E. Zheng, L. Wang, Y. Luo, K. Wei, and Q. Wang, Non-contact capacitance sensing for continuous locomotion mode recognition: Design specifications and experiments with an amputee, *IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, (2013).