

Intuitive Human-Machine Interfaces for Non-Anthropomorphic Robotic Hands

Cassie Meeker

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2020

© 2020

Cassie Meeker

All Rights Reserved

Abstract

Intuitive Human-Machine Interfaces for Non-Anthropomorphic Robotic Hands

Cassie Meeker

As robots become more prevalent in our everyday lives, both in our workplaces and in our homes, it becomes increasingly likely that people who are not experts in robotics will be asked to interface with robotic devices. It is therefore important to develop robotic controls that are intuitive and easy for novices to use. Robotic hands, in particular, are very useful, but their high dimensionality makes creating intuitive human-machine interfaces for them complex. In this dissertation, we study the control of non-anthropomorphic robotic hands by non-roboticists in two contexts: collaborative manipulation and assistive robotics.

In the field of collaborative manipulation, the human and the robot work side by side as independent agents. Teleoperation allows the human to assist the robot when autonomous grasping is not able to deal sufficiently well with corner cases or cannot operate fast enough. Using the teleoperator's hand as an input device can provide an intuitive control method, but finding a mapping between a human hand and a non-anthropomorphic robot hand can be difficult, due to the hands' dissimilar kinematics. In this dissertation, we seek to create a mapping between the human hand and a fully actuated, non-anthropomorphic robot hand that is intuitive enough to enable effective real-time teleoperation, even for novice users.

We propose a low-dimensional and continuous teleoperation subspace which can be used as an intermediary for mapping between different hand pose spaces. We first propose the general concept of the subspace, its properties and the variables needed to map from the human hand to a robot hand. We then propose three ways to populate the teleoperation subspace mapping.

Two of our mappings use a dataglove to harvest information about the user's hand. We define the mapping between joint space and teleoperation subspace with an empirical definition, which requires a person to define hand motions in an intuitive, hand-specific way, and with an algorithmic definition, which is kinematically independent, and uses objects to define the subspace. Our third mapping for the teleoperation subspace uses forearm electromyography (EMG) as a control input.

Assistive orthotics is another area of robotics where human-machine interfaces are critical, since, in this field, the robot is attached to the hand of the human user. In this case, the goal is for the robot to assist the human with movements they would not otherwise be able to achieve. Orthotics can improve the quality of life of people who do not have full use of their hands. Human-machine interfaces for assistive hand orthotics that use EMG signals from the affected forearm as input are intuitive and repeated use can strengthen the muscles of the user's affected arm. In this dissertation, we seek to create an EMG based control for an orthotic device used by people who have had a stroke. We would like our control to enable functional motions when used in conjunction with a orthosis and to be robust to changes in the input signal.

We propose a control for a wearable hand orthosis which uses an easy to don, commodity forearm EMG band. We develop an supervised algorithm to detect a user's intent to open and close their hand, and pair this algorithm with a training protocol which makes our intent detection robust to changes in the input signal. We show that this algorithm, when used in conjunction with an orthosis over several weeks, can improve distal function in users. Additionally, we propose two semi-supervised intent detection algorithms designed to keep our control robust to changes in the input data while reducing the length and frequency of our training protocol.

Table of Contents

List of Tables	v
List of Figures	vii
Acknowledgments	x
Chapter 1: Introduction	1
1.1 Teleoperation for Dexterous Robot Hands	1
1.2 Contributions	5
Chapter 2: Background and Related work	7
2.1 Teleoperation Controls	7
2.1.1 Teleoperation Controls Using Datagloves as Input	8
2.1.2 Teleoperation Controls Using Low-Dimensional Spaces	10
2.1.3 Teleoperation Controls Created Algorithmically	11
2.1.4 Teleoperation Controls Validated with Online Experiments	12
2.1.5 Teleoperation Controls Using EMG as Input	13
2.2 Rehabilitation Robotics	14
2.2.1 Supervised EMG Controlled Orthoses	15
2.2.2 Supervised EMG Controlled Prosthetics	17

2.2.3	Semi-Supervised EMG Controlled Prosthetics	18
2.2.4	Disagreement-Based Semi-Supervised Learning	19
Chapter 3:	Teleoperation Subspace	21
3.1	Definition	22
3.2	Teleoperation Subspace Mapping	24
3.3	A Complete Teleoperation Algorithm	26
3.4	Summary	26
Chapter 4:	An Empirical Mapping for the Teleoperation Subspace	28
4.1	Empirically Defining the Subspace Mapping	29
4.2	State-of-the-Art Comparisons	31
4.3	Experiments	33
4.4	Results	35
4.5	Summary	38
Chapter 5:	An Algorithmic Mapping for the Teleoperation Subspace	39
5.1	Algorithmically Defining the Subspace Mapping	40
5.1.1	Object Set	43
5.1.2	Grasp Generation	44
5.1.3	Fitting a Subspace to a Grasp Dataset	46
5.1.4	A Complete Teleoperation Algorithm	50
5.2	State-of-the-Art Comparisons	51
5.3	Experiments	52

5.4	Results	56
5.5	Summary	62
Chapter 6: Using the Teleoperation Subspace with an EMG Input		64
6.1	Mapping EMG into Teleoperation Subspace	65
6.2	State-of-the-Art Comparisons	68
6.3	Experiments	70
6.4	Results	75
6.5	Summary	77
Chapter 7: Wearable Orthotics for Stroke Subjects		79
7.1	Challenges	80
7.2	Summary	82
Chapter 8: Supervised EMG Control for Wearable Hand Orthotics		83
8.1	Hardware	84
8.2	Intent Detection	86
8.3	Training with the Exotendon Device	90
8.4	Experiments	91
8.5	Results	94
8.6	Summary	99
Chapter 9: Rehabilitative and Assistive Performance of a User-Controller Hand Orthosis . .		100
9.1	Hardware	101
9.2	Intent Detection	102

9.3	Training with the Exotendon Device	103
9.4	Experiments	104
9.5	Results	107
9.6	Summary	109
Chapter 10: Semi-Supervised EMG Control for Wearable Hand Orthotics		111
10.1	Hardware	113
10.2	Intent Detection	114
10.3	Semi-Supervised Learning for Intent Detection	116
10.3.1	Disagreement Based Oracle	117
10.3.2	K-means Oracle	119
10.3.3	LDA Updater	121
10.4	Experiments	122
10.5	Results	125
10.6	Summary	131
Chapter 11: Conclusion		133
11.1	Contributions	133
11.2	Challenges	135
11.3	Future Work	136
References		150

List of Tables

4.1	Process to empirically define the projection matrix for the teleoperation subspace mapping	31
5.1	Object set	43
5.2	Joint mapping from the Cyberglove to the Schunk SDH	53
5.3	Joint mapping from the Cyberglove to the two finger gripper	54
5.4	Pick and place experiment results	58
5.5	In-hand manipulation experiment results	59
6.1	Normalized root mean square error of trained regressors and global accuracy of trained classifiers	72
6.2	Pick and place statistics	76
8.1	Training protocol and assigned labels. For each combination of instruction given to the subject and state of the exotendon device, the table shows the ground truth label $Intent_t^s$ assigned to EMG data. Training begins with the tendon extended and the subject asked to relax (top row, middle column) and proceeds in counter-clockwise fashion	90
8.2	Subject demographic and clinical information	92
8.3	Results for non-functional motions	94
8.4	Results for functional motions (pick and place)	96

9.1	Training protocol and assigned labels. For each combination of instruction given to the subject and state of the exotendon device, the table shows the ground truth label $Intent_t^s$ assigned to EMG data. Training begins with the tendon extended and the subject asked to relax (top row, middle column) and proceeds in counter-clockwise fashion	104
9.2	Rehabilitative and assistive results, showing mean gains and standard error between pre- and post-intervention. Bold results indicate gains (or losses) outside the range of standard error	108
10.1	Intersession results	126
10.2	Intrasession results	126

List of Figures

3.1	Teleoperated manipulation where the operator's hand movements are recorded and used to drive a non-anthropomorphic robotic hand. To achieve this, a mapping between the human and robot hand kinematics is required.	22
3.2	Steps to enable real time teleoperation using teleoperation subspace with a data-glove as input.	24
4.1	The teleoperation subspace used as an intermediary between the pose spaces of different hands. Here we show how the motions that are associated with the subspace basis vectors can be intuitively defined by a user, based on the hand's kinematics. .	29
4.2	Origin poses of two example hands.	30
4.3	Poses which demonstrate the human hand's kinematics limits along the basis vectors of T . To see the poses in the context of the teleoperation subspace, refer to Figure 4.1.	30
4.4	Joint mapping scheme between the Cyberglove and the custom robotic hand.	32
4.5	Setup for pick and place experiments.	34
4.6	Average time (in seconds) to pick and place objects using different teleoperation controls in (top) graphical and (bottom) numerical format. * denotes that the average was calculated with a smaller sample size for some of the mapping methods of that object.	36
5.1	We can formalize the hand motions that define the subspace with objects, then use this definition to predict where a grasp for that object will lie in the subspace, regardless of the hand's kinematics. This figure shows the human hand and the Schunk hand grasping the same set of objects. When an object is held by either hand, the resulting grasp will lie at the same location in the subspace T	42
5.2	Object set visualizations.	44

5.3	Teleoperation mappings generated for the human hand, Schunk SDH, and two finger gripper, both empirically and algorithmically. Each of the spokes represents a degree of freedom for the hand, and the blue (spread), red (size) and green (curl) values along those spokes indicate the values in the α_H , σ_H , and ϵ_H , respectively, at that degree of freedom.	55
5.4	Pick and place experimental setup.	56
5.5	In-hand manipulation experimental setup.	57
6.1	Control scheme for the proposed EMG teleoperation.	67
6.2	Steps to enable real time teleoperation using forearm EMG as a control input. . . .	68
6.3	Comparison of control methods which map between forearm EMG and a robot pose space.	69
6.4	Pick and place experiments - setup and results.	75
7.1	Two examples of concept drift in the EMG signals of a stroke patient. Each graph shows eight signals from the sensors of our EMG sensing suite. These signals are produced by a single subject under different conditions.	81
8.1	Prototype hand orthosis in extension configuration with EMG armband.	84
8.2	Top: Prediction of classifier (dotted blue), ground truth (solid red) and filtered probability before thresholding (solid orange) vs. time for Subject A without the device operating. Classification value of 1: open intention, classification value of 0: no open intention. Bottom: Raw EMG of Subject A and open close events which correspond to the top graph.	95
8.3	Top: Prediction of classifier (dotted blue), ground truth (solid red) and filtered probability before thresholding (solid orange) vs. time for Subject A with the device operating. Classification value of 1: open intention, classification value of 0: no open intention. Bottom: Raw EMG of Subject A and open close events which correspond to the top graph.	96
8.4	Illustration of pick and place tasks by 2 subjects. Each row shows a complete task execution by one subject, progressing from left to right.	97
8.5	Force results for Subject A. Each peak corresponds to one hand extension for grasping.	97

9.1	Updated hand orthosis prototype in extension configuration.	101
9.2	Real world objects used for treatment.	106
10.1	Updated hand orthosis prototype in extension configuration with multimodal sensing suite.	113
10.2	Example of both stable and unstable intersession intent prediction. The motor command predicted by the classifier is blue and the ground truth motor command is red. Plot labels indicate the type of classifier used, as defined in Section 10.4. A vertical black line denotes the classifier was updated at that time step.	127
10.3	Example of stable intrasession intent prediction. The motor command predicted by the classifier is blue and the ground truth motor command is red. Plot labels indicate the type of classifier used, as defined in Section 10.4. A vertical black line denotes the classifier was updated at that time step.	128

Acknowledgments

First and foremost, I would like to thank my advisor, Matei Ciocarlie. He took a chance on a biomedical engineer who knew next to nothing about robots or coding and was very patient as I learned both. I became a better scientist during my time in his lab and hope that I came away with at least a fraction of his ability to distill very complex, high-dimensional ideas into simple and clear statements. He has a vision for how he wants to change the field of dexterous robotic manipulation and I look forward to seeing what he and his lab come up with next.

I couldn't have done a lot of the work in this thesis without the support and expertise of the clinical team at Columbia's medical center. Joel Stein, Lynne Weber, Michaela Fraser, and Lauri Bishop kept us focused on the priorities of our population, always had suggestions for improving my controls, powered through IRB forms, and always kept our sessions with subjects fun and upbeat. I am very grateful for our collaboration and couldn't have asked for a more invested team of doctors and therapists.

I would also like to thank all the subjects who participated in my studies for their patience and their feedback. My subjects' enthusiasm and determination made every user study worth the fact that they somehow found every corner case in my code. They were absolutely instrumental in my ability to validate all of the controls in this dissertation and I would not be here without them.

I would like to thank my committee, Sunil Agrawal, Matei Ciocarlie, Joel Stein, Peter Allen, and Gerard Ateshian, for reading my thesis, their support, and their feedback. I couldn't have asked for a better, or more varied, group of experts to review the work in this dissertation.

I'm grateful I got to spend my PhD with the four other students who were in Matei's first cohort with me: Sangwoo, Pedro, Tianjian, and Max. I appreciate all of the hardware and server expertise

they lent to me - I wouldn't have been able to create or test most of my controls without it. I also learned more about soccer (football) than I ever hoped or wanted to know from them. Emily came in a year later, but she was a great roommate, our walks around campus lowered my stress levels, and I stayed young listening to stories of her caving adventures. I appreciate the hard work of the master students and undergrads who worked on these projects with me - Abigail, Rami, and Tom. I also really enjoyed working with our lab postdocs, Long and Bing. As for the group of PhDs who started in the lab in the past year, I didn't get to work with them for very long, but I'm excited to see what they come up with. I especially want to mention Ava, who is taking over our orthosis project - she is passionate about wearable robotics and I know she'll take our device into the next phase with zeal.

I couldn't have made it through these past five years without the amazing group of friends I found in grad school. Danielle, Emily, Nicole, Rand, Erin, and Tatiana, among others - thank you for the trivia, the stitching, the TV binge nights and all the casual hangouts. They are some of the best women and best engineers I know and it was a pleasure to go through this journey with them. I wish them the best of luck.

I'd like to thank my Mom, my Dad, and my sister Becca, their unconditional love, for proof-reading my papers, for their support, and for always giving me a place to come home, no matter where I lived.

And finally, to Sam, who was here for all of it, who stayed up with me most nights, who vacationed wherever my latest conference was, who constantly made me laugh, and who brought me chocolate when I needed it, you kept me sane through all of this. Thank you for all the adventures, the love, the support, and for listening to all my engineering talk. We made the most of our time in NYC and I'm glad I got to spend it with you.

As I near the end of my PhD and reflect back on my five years here, it may surprise some (even me) to learn that I would do it all again in a heartbeat. I fell, quite by accident, into a field that I love, and which I believe can change the world. So, as I push onward to the next adventure, I hope to carry with me the lessons, the people, and the passion that I found in my PhD.

Chapter 1: Introduction

From industry [1] to the home [2] to our workplaces [3], robots are becoming more and more prevalent. In hospitals, surgeons and robots operate together, in disaster zones, robots help humans diffuse and clean up dangerous situations, in homes and in clinics, rehabilitation robots allow people to regain function of their limbs, and in workplaces, robots supervised by humans perform assembly and manipulation tasks. These new paradigms have made human-robot collaboration the subject of much study in both academia and industry.

As these collaborative robots become more common, effective communication between the human and robotic agents is important. Seamless communication helps ensure the human's safety and enables an effective workflow. We need communication and control methods which do not significantly interfere with the human's responsibilities within the collaborative workspace. Human machine interfaces need to be intuitive, fast, and easy to use.

1.1 Teleoperation for Dexterous Robot Hands

This thesis focuses specifically on human-machine interfaces to control robotic hands. Robotic hands are extremely useful, but extremely complex. Dexterous manipulation in autonomous systems is still an open question, due to the high dimensionality of the joint space which must be controlled. As such, it is conceivable that users will need to interface with robotic hands in order to guide dexterous manipulation tasks more frequently than they will need to interface with other robotic systems.

The high dimensionality of robotic hands is also what makes creating human machine interfaces for them so complex. Interfaces using joysticks or cursors are insufficient to enable control of such devices, especially if the user requires the robot hand to perform dexterous manipulation

tasks.

Fortunately, the human hand is equally high dimensional and can serve as a controller for robotic hands. Using the human hand as a control input can allow users to teleoperate using the same motions they would have used to pick up the object themselves. Since the control leverages motions the user is already familiar with, and does not require knowledge of external hardware, this control input provides an intuitive and user-friendly interface [4]. This helps users, particularly novice users, complete tasks in a safe and timely manner.

The human-machine interfaces we propose in this thesis harvest the motion of the human hand to control robotic hands. We collect information about the users' hands using two devices: a dataglove, which measures the joint angles of the hand, and an electromyography (EMG) armband, which measures muscle activity in the user's forearm.

With controls that harvest the motion of the user's hand, interfacing with robotic hands becomes particularly difficult when the robot is non-anthropomorphic, or when the user does not have full use of their hand. When the robot hand is non-anthropomorphic, the different joints, different axes, different numbers of fingers, or any number of dissimilarities between the hands make it complex to define how the robot should respond to movements of the human hand. When the user does not have full use of their hand, either because of environmental or physical constraints, this lowers the dimensionality of the input we can give to the robot. In these cases, we can often use EMG signals to approximate for hand position.

We study the control of non-anthropomorphic robotic hands in two contexts:

- **Collaborative (side-by-side) Manipulation:** In human-robot teams, ideally, the robot would be able to complete its part of the task autonomously. However, this is not always possible. Dexterous manipulation for robot hands has not yet been solved because of the high dimensionality of the hands, the wide array of objects that can be encountered in unstructured environments, and the almost infinite possibilities for how to approach, grasp, and manipulate an object [5].

When the robot encounters a situation where it cannot solve how to grasp or manipulate an

object autonomously, the human team member can be asked to teleoperate the robot until the grasping task becomes tractable for the autonomous program. In these corner cases, human cognition is faster, more reliable, and more robust than autonomous planners.

The field is currently lacking intuitive, wearable, and unobtrusive controls for robotic systems designed for side-by-side or collaborative manipulation that are robust and dexterous enough for complex scenarios. This is, in part, because robotic hands which have proven themselves to be useful and versatile in unstructured environments are often non-anthropomorphic, and therefore unintuitive to control.

This thesis develops controls for able-bodied users who wish to teleoperate non-anthropomorphic hands during side by side collaboration. We introduce a teleoperation subspace as an intermediary between hands with dissimilar kinematics (Chapter 3). A linear projection allows us to project from the human joint space or forearm EMG to the teleoperation subspace, then from the teleoperation subspace to the robot joint space.

We use both datagloves and EMG signals as control inputs for this teleoperation scheme. Datagloves are robust and consistent, and forearm EMG is inexpensive and does not compromise the user's ability to perform manipulation on their own.

We introduce two methods for creating the mapping between joint space and the teleoperation subspace. One of these methods requires users to provide intuition about hand kinematics (Chapter 4), and the other is kinematically independent (Chapter 5). We also provide a way to map between EMG signals and the teleoperation subspace (Chapter 6). Through experiments with novices, we show that the subspace we propose is relevant to teleoperation for hands of multiple kinematics, that the value of the subspace does not derive exclusively from human intuition, and that novices find the control intuitive with both EMG and dataglove control inputs.

- **Assistive Robotics:** Wearable assistive orthotics can help improve the quality of life of people who do not have full use of their hands, like people who have had a stroke. Yet,

without full use of their hand, these are the people who are perhaps least able to interface with the conventional robotic devices which could help them regain function in their hand or perform activities of daily living.

The field is lacking in human-machine interfaces for wearable hand orthoses which have been studied as the orthosis moves the subject's hand in order to elicit functional tasks. As the orthosis elicits the desired movement pattern for the user's hand, the user will fatigue, the sensors may shift, and, if the user has had a stroke, the motion may activate the abnormal muscle coactivation present in many stroke subjects.

In this thesis, we develop human-machine interfaces for stroke subjects, whose most common impairment pattern (spasticity) prevents them from extending their fingers. Because the target users do not have full use of their hand, we develop a control for a robotic rehabilitation device using EMG signals from the affected forearm as input. This control input has the additional benefits of being intuitive, and strengthening the muscles of the user's affected arm.

We present a EMG based control for an underactuated hand orthosis which we show can correctly classify the user's intended movement during functional motions while the orthosis is operating (Chapter 8). To compensate for the changes in input signals caused by the interaction between the orthosis and the hand, we develop a training procedure which compensates for the altered EMG signals during these functional movements. We show that, when used in conjunction with an orthosis over several weeks, our control can provide functional gains for chronic stroke subjects (Chapter 9). We also explore semi-supervised learning as a way to compensate for changes in the input signal (Chapter 10). We present two semi-supervised learning algorithms and demonstrate that, since they can compensate for these signal changes, they can reduce the training burden placed on the user by data-driven controls.

1.2 Contributions

In this thesis, we focus specifically on creating human-machine interfaces that are robust to novice users. We validate all our proposed human-machine interfaces on experiments with people who are not robotic experts. As robots become more prevalent in our everyday lives, it becomes more and more likely that the people who need to interact with robots will not be experts in the field. It is therefore important to make controls intuitive, with a low cognitive burden on the user.

Our experiments show that the controls introduced in this thesis are robust to novice users - they require little or no formal training, while remaining intuitive. Our experiments in the field of collaborative manipulation show that our proposed methods allow novice users to teleoperate faster than state-of-the-art methods. Our experiments in the field of assistive robotics show that our proposed methods allow novices to perform functional motions with a relatively short training time.

This thesis is the first to:

- Propose a low dimensional subspace relevant to teleoperation that can enable online teleoperation for novices faster than state-of-the-art teleoperation methods.
- Show that an algorithm that produces teleoperation mappings without kinematic-specific intuition can enable online teleoperation.
- Demonstrate teleoperation of a non-anthropomorphic, multi-degree-of-freedom (multi-DOF) robot hand using forearm EMG as a control input.
- Study pattern classification of functional movements for stroke subjects in conjunction with a physical orthosis that enables grasping.
- Propose a semi-supervised control for a hand orthosis; this control aims to reduce the training burden placed on the user by intent detection classifiers trained on forearm EMG.

Overall, in this dissertation, we aim to develop human-machine interfaces that allow people who are not robotic experts to easily control non-anthropomorphic robot hands. We introduce

novel methods to control robot hands during collaborative manipulation and to control an assistive robotic device used for stroke rehabilitation. In both contexts, we perform user studies with novices to validate our controls and ensure they are intuitive. Our work in collaborative manipulation represents a complete pipeline to create and deploy a teleoperation control using either a dataglove or EMG as a control input. Our work in rehabilitation robotics shows that ipsilateral controls using EMG as a control input can enable functional movement with relatively little training data, but also highlights areas of future study. *We believe that the contributions in this dissertation show that even complex, non-anthropomorphic robots can be accessible to novice users.*

Chapter 2: Background and Related work

Human machine interfaces come in all shapes and sizes. They can be controlled by voice [6], by touchscreen [7], or by biological signals [8, 9], to name a few. In this dissertation, we focus on controls which harvest the motions of the human hand because the human hand is high dimensional and therefore an ideal control for high dimensional robotic hands. We focus primarily on studies which use datagloves or forearm EMG as control inputs, since this is what we use for our controls.

A human-machine interface should be tailored to its application. Even if two controls use forearm EMG as input to execute a grasping task, the control must be tailored to both the target user population and to how the hardware is intended to interface with the user. Depending on the control, robot kinematics can also shape how the interface should be designed. In this chapter, we review controls for two applications: teleoperation controls and rehabilitation robotics.

That being said, we believe that lessons learned from one field of robotics can inform the controls for another robotic field. Although we focus on two applications, we also attempt to cast a wide net for controls which are relevant to these applications - for our review of teleoperation literature, we include relevant controls from side-by-side manipulation, learning from demonstration, autonomous, and even prosthetic applications. For our review of the rehabilitation robotic literature, we include relevant controls from prosthetics, orthotics, and multi-view learning applications.

2.1 Teleoperation Controls

Human machine interfaces in side-by-side manipulation must give the user effective control of the robot, since they will be asked to teleoperate in complex corner cases where the robot is unable to complete the task alone.

We review literature which teleoperates robotic devices using two control inputs: datagloves

and forearm EMG. For the dataglove input, we review classic teleoperation controls, teleoperation controls which use low dimensional spaces, and teleoperation controls which are created independent of hand kinematics. For EMG-based controls, most teleoperation literature only studies single degree of freedom robots, so we also review controls from prosthetic applications.

The human-machine interfaces in this dissertation are designed for and tested on novice users. In the literature, it is relatively rare to do any testing with novices. We review literature which validates their teleoperation controls with online experiments and highlight those who perform studies with novices.

2.1.1 Teleoperation Controls Using Datagloves as Input

Conventional teleoperation methods are divided into three main categories: joint mapping, fingertip mapping, and pose mapping.

Joint mapping (also called joint-to-joint mapping) is used when the slave hand has similar kinematics to the human controller [10]. If the human and robot joints have a clear correspondence, the human joint angles can be imposed directly onto the robot joints with little or no transformation [11]. This mapping is most useful for power grasps [12], and is limited if the robot hand is non-anthropomorphic.

Fingertip mapping (also called point-to-point mapping) is the most common teleoperation mapping method. Forward kinematics transform human joint angles into Cartesian fingertip positions. These undergo scaling to find the desired robot Cartesian fingertip positions and then inverse kinematics determine robot joint angles. This mapping is useful for precision grasps [12]. When the robot has less than five fingers, the extra human fingers are ignored [13]. However, teleoperation using this method is difficult when the workspaces of the human and robotic fingers are not similar, as is the case for non-anthropomorphic hands.

An alternative to fingertip mapping is virtual object mapping, which uses the relative distances between fingertips. Virtual object mapping uses the distance between the master fingertips and inverse kinematics to calculate joint angles that place the slave fingertips at the same relative dis-

tance. This method can be used in both 2-D [14] and 3-D [15][16] grasping scenarios. The relative distances between fingertips are often calculated based on Cartesian fingertip positions, so virtual object mapping is similar to fingertip mapping. Like fingertip mapping, virtual object mapping can be limited when the human and robot workspaces are incompatible.

For fingertip and joint mapping, how to reconcile kinematic differences between the human and robotic workspaces is an open question. Humans use only 3.6% of the potential workspace for the thumb during grasping [17]. So, if a robot finger maps to the human thumb, that finger will not move significantly during grasping tasks, unless the teleoperator adapts their grasping in unintuitive ways. To solve this problem, researchers combine different types of mappings [18, 12], use virtual object mapping with special considerations for the workspace differences [14], optimize distances in task space [10], use error compensation [19], and alter the robotic hand frame for individual grasp postures to minimize the workspace differences [20].

Pose mapping attempts to replicate the pose of the human hand with a robot hand. Unlike fingertip and joint mapping, it attempts to interpret the function of the human grasp rather than replicate literal hand position. Sometimes, this requires identifying the human pose before mapping between the human and the robot [21, 22]. Usually a neural network is used to identify the human pose and then the mapping occurs through another neural network [21] or pre-programmed joint-to-joint mapping [22]. Outside of a discrete set of known poses, pose mapping can lead to unpredictable hand motions and is typically used when only simple grasping poses are required. Furthermore, if the human hand pose is misidentified, the robot hand will move in undesirable ways. Others perform pose mapping in an end-to-end fashion, either with transformation matrices, using least squared error compensation when this transformation is not exact [23] or, more recently, neural networks [24].

It is worth noting that all of the above mapping techniques can use vision as a control input as well [25, 26, 27, 24]. However, vision-based methods are less robust than dataglove-based methods because vision requires that the human hand be well-lit and poised in the field-of-view of a camera. Furthermore, vision-based controls suffer when part of the human hand is obfuscated,

either by other fingers or external objects that the teleoperator is manipulating. These conditions cannot always be guaranteed in unstructured environments. Because of this, we do not focus on vision based methods in this work.

2.1.2 Teleoperation Controls Using Low-Dimensional Spaces

Because of the high-dimensionality of the human hand, some teleoperation controls use low dimensional spaces to map between the human hand and robotic hands.

‘Postural synergies’ make up a low dimensional grasping space found by Santello et al. [28]. These synergies are specific to the human hand. Synergies move the description of human grasping from discrete, static poses [29] into a continuous space.

The concept of postural synergies has been applied to robots in different ways. Some works find synergies of robot hands by finding robot poses that resemble grasping poses for human hands, and then performing principal component analysis (PCA) on those poses. The poses are either found through joint mapping [30], pose mapping [31], or human intuition [32] [33]. Others use postural synergies to underactuate anthropomorphic hands, like the Pisa/IIT SoftHand 2 [34], using human intuition to build synergies into the mechanical design of the robot.

Synergies can be an effective control for teleoperation. Jenkins demonstrated a low dimensional control which could potentially be used to teleoperate robotic systems with cursor control [35]. To our knowledge, this has only been tested in simulation. Brygo et al. translated postural synergies from joint space to fingertip Cartesian space to control teleoperation [36]. This work only considers the first postural synergy, and is most appropriate for underactuated hands. Kim et al. demonstrated a synergy level controller which uses multiple postural synergies to enable teleoperation [31]. This method calculates the synergies of the robot hand through pose mapping, which could have discontinuities. They calculated robotic synergy coefficients based, in part, on the rate of change for each synergy coefficient. The low dimensional space we introduce in this thesis does not have discontinuities and does not have a temporal component.

Synergies have also been used in the field of autonomous grasp planning. Finding synergies

for robotic hands based on human synergies can inform grasp planning [32][37][20].

Other works use low dimensional latent spaces which are not based on synergies to approximate human poses in non-anthropomorphic models. These latent variables have enabled both the animation of non-anthropomorphic creatures [38] and teleoperation.

Gaussian process latent variable models (GP-LVM) can enable teleoperation of humanoid robots. In some formulations, the latent space changes with every different master-slave pairing [39]. In other formulations, multiple robots and a human share the same latent space [40].

Training data driven mappings, like some pose mappings [32], or GP-LVMs, requires the user to create many corresponding poses between the human and robotic hands. Creating these corresponding poses is often tedious and time consuming. We are inspired by a similar desire to find shared subspaces between robotic and human hands, but our methods for generating teleoperation mappings reduce the burden placed on the user by eliminating the requirement to create tens or even hundreds of corresponding poses between the human and robot.

2.1.3 Teleoperation Controls Created Algorithmically

There are works which, like the algorithmic mapping introduced in Chapter 5, try to create teleoperation mappings without requiring that the user provide intuition about hand kinematics.

Kheddar et al. proposed high level abstraction teleoperation, where the operator manipulates a virtual environment and a bilateral transform translates changes in the virtual environment into commands for the robot, such that the slave mimics the change in the real environment [41]. The gripper control is object based, i.e. the robot must manipulate and transport an object in the real world in the same way it is being manipulated in the virtual environment [42]. Although they describe several possible ways to transform between the human and robotic hands, their ultimate solution is autonomous, and does not include a teleoperation mapping. This idea of learning high level tasks from demonstrations is reviewed extensively elsewhere [43].

Kang et al. also introduced an object based approach to identifying human grasps using a contact web [44]. Once the human grasp has been identified, the robot hand is shaped based on

virtual fingers and the human grasp. However, the user still provides an understanding of how the robot functions - for each new robot, they assign the fingers as being a primary finger, a secondary finger, or a palm.

Finally, Gioioso et al. defined an object based approach for mapping between hands with dissimilar kinematics using virtual objects [45, 46, 47, 48, 49]. This work replicates the deformation of the virtual object in the human hand with the virtual object in the robot hand. This is the first time a (virtual) object set was used to define a teleoperation mapping. However, the authors have reported varying performance for the same hand with different numbers of virtual points and different numbers of synergies, meaning that creating the mapping for each hand requires the user to tune control parameters, including how many contact points to use, where to place these contacts, and which synergies to use. Of this body of work, only two publications show the method to be feasible for robotic hand manipulation with online teleoperation experiments [48, 49]. The first work only considers spherical virtual objects [48]. The method was later extended to a virtual object of any shape, but does not consider scenarios where the slave has fewer contact points than the master, and it assumes the the movement of the master contacts can be represented by a homogeneous transform [49]. In our experiments, neither of these assumptions hold true.

All of the above work requires the user to give some input which provides an understanding about the hand's kinematics or function. Some groups have moved towards teleoperation mappings which reduce the burden on the user, but our algorithmic mapping is the first to create a teleoperation mapping completely automatically, using a subspace definition which is independent of hand kinematics.

2.1.4 Teleoperation Controls Validated with Online Experiments

As a final note about the pose mapping and synergy based methods, to our knowledge, only three publications in the literature demonstrate a pose or synergy mapping to be feasible for hand manipulation with online teleoperation experiments [49, 48, 24]. None of these methods were shown to be effective for a non-anthropomorphic hand, which is the focus of this thesis.

In the literature on hand teleoperation, most works validate their proposed teleoperation methods on one or two expert users or perform their experiments in simulation (e.g. [25, 19, 23, 12, 18, 14, 21, 22, 45, 46, 47]). We were only able to find two works that validate a proposed teleoperation method with novice users on a physical robot [24, 48]. Both of these works validated their method with five novices users teleoperating a single robotic hand. In this thesis, we validate our work over two different manipulation tasks with a total of fifteen novice users, using two robotic hands with different kinematic configurations. These experiments show that the subspace mappings we propose are intuitive and encode information relevant to teleoperation for hands with different kinematics.

2.1.5 Teleoperation Controls Using EMG as Input

EMG control in teleoperation has been primarily used for two applications: grippers and control of anthropomorphic hands, principally in the context of prosthetics.

EMG teleoperation of a one degree of freedom (DOF) gripper usually uses proportional control to determine the gripper aperture [50][51]. Using EMG, it is possible to estimate the aperture of a gripper in conjunction with the position of the elbow and the wrist [52]. Choudhary et al. developed a three DOF hand with EMG control, but the EMG only provides a binary open/close signal [53]. EMG control for grippers is consistent, intuitive and reliable, but grippers do not give the sort of versatility that many complex grasping scenarios require.

The other context in which EMG teleoperation has been studied is prosthetics. All of the studies below describing prosthetic EMG controls were tested on able bodied humans. Prosthetic controls can use an agonistic/antagonistic muscle pair to control a single DOF underactuated hand [54] or they can estimate the motion of one of the joints of the human hand [55] [56]. Studies which control more than one DOF usually control the wrist, often in a continuous and proportional manner [57][58][59][60].

One interesting approach combines force control and gesture control. Yoshikawa et al. classified hand position, and then used an empirical EMG-joint angle model to estimate wrist or metacar-

popphalangeal (MCP) joint angles [61]. Yamanoi et al. developed models of the relationship between force and EMG for multiple postures. They used EMG to classify hand pose, and then determined grip force based on the force model [62]. Castellini et al. used a similar strategy in a data based approach [63]. Gijssberts et al. created an entirely force driven EMG control for a prosthetic hand [64].

As with dataglove controls, EMG-based teleoperation controls for multi-DOF anthropomorphic robots commonly use low dimensional synergy spaces and dimension reduction [65]. Rossi et al. mapped EMG signals to the synergies of an underactuated robot hand [66]. Matrone et al. used agonistic/antagonistic muscle pairs to determine wrist position, and then control a robotic hand using the synergies of the robot [67]. Artemiadis and Kyriakopoulos used principal component analysis (PCA) to find low-dimensional representations of both kinematic and EMG data for the human arm. They mapped between the two spaces to teleoperate a robot arm [68], and, later, a hand-arm system [10].

Fully anthropomorphic robot hands are complex, fragile and expensive. At the other end of the spectrum, open-close binary EMG controllers are easier to implement and map directly to one-DOF grippers, but provide less versatility and dexterity.

To the best of our knowledge, we are the first to develop an EMG control that can control multiple DOFs for non-anthropomorphic robot hands, combining the dexterity of multi-DOF control with the robustness and cost effectiveness of non-anthropomorphic hands.

2.2 Rehabilitation Robotics

Rehabilitation robotics is another field in which human-machine interfaces play a critical role in the performance of the system. In this field, the robotic device is actually attached to the user's hand and/or arm. Perceived device performance is strongly correlated with use for both orthoses [69] and prostheses [70]. It is important that the device meets the user's expectations in its effectiveness, ease of use, and reliability, or the user will not wear it.

Proposed control methods for rehabilitation robotics include brain control interfaces [71],

bilateral controls [72], therapist/user driven controls [73] and ipsilateral EMG controls. Of these possible interfaces, only ipsilateral EMG uses the same muscles and movement patterns that would have been used before the disability/injury occurred, and has the potential to be easy to do. For that reason, in this dissertation, we focus on ipsilateral EMG-based control mechanisms. EMG-based controls have been used successfully for the knee [74], upper arm [75], and wrist [76], as well as the hand.

EMG-based human machine interfaces can be *supervised*, where all of the training points have a corresponding label or where heuristics are used, *unsupervised*, where none of the training points have a label, and *semi-supervised*, where a small subset of the training points have a corresponding label. Below we will discuss both supervised and semi-supervised paradigms, since they have been successfully applied to the field of rehabilitation robotics. We also discuss disagreement based semi-supervised learning, which has not been applied to rehabilitation robotics before, but which we believe has the potential to improve controls in this field.

Rehabilitation robotics for hands can, broadly speaking, be divided in to two categories: orthotics and prosthetics. Orthotics provide assistance to people who have lost function of their hand, and prosthetics provide assistance to people who have lost the hand itself. We will discuss controls for both below.

A large part of this thesis deals with how the presence of hardware, when it is attached to the human body, can change the input signals from the user. This has not been studied extensively in the literature for orthoses, but we do call out a few studies which have taken this into consideration.

2.2.1 Supervised EMG Controlled Orthoses

A number of proposed EMG controls for hand orthoses include the use of bilateral muscles [77] or bicep muscles [77, 78]. We have chosen to focus on ipsilateral muscles of the forearm, aiming for more intuitive control for the user, and also leaving the bilateral hand free to operate independently. Furthermore, most forearm muscles are used specifically to manipulate the hand, whereas bicep muscles are not.

Hand orthoses using ipsilateral forearm EMG controls are largely developed with either the goal improving hand function or assisting grasping [79]. A number of studies have shown that EMG data can be processed in order to predict hand position [80, 81, 82]; the armband we use to collect EMG signals has a built-in ability to extract hand control signals based on typical EMG patterns of a healthy user.

None of the above studies, nor the armband, demonstrate the feasibility of using their control scheme in conjunction with an orthosis that enables the execution of functional tasks. Our experience indicates that methods which predict hand motions in absence of an orthosis need significant adaptation to be effective when an orthosis is present (Section 8.3). This thesis presents a method for training an EMG control in the presence of an orthosis to better predict hand position while the orthosis is operating.

Other studies do discuss EMG-driven controls that are developed in conjunction with orthoses [83, 84, 85]. In these studies, two sensors are placed by trained experimenters on specific pairs of muscles. The control then uses a threshold to determine when to open and close the orthosis. Our control does not require placement on specific muscles to function, resulting in a much easier donning process. Our commodity sensors are fully encapsulated and wearable, with no separate amplification or power electronics. Both of these features are important for building wearable orthotics that patients can use at home without direct medical supervision. Furthermore, instead of looking only at pairs of muscles and thresholding activation signals (as in the above studies), we investigate patterns within the EMG signal of the entire forearm. This allows for future development of the EMG control to include multiple and varied hand positions. It has been shown that pattern classification can identify multiple hand positions in stroke patients [86]; therefore, we predict our control approach will be able to grow and develop with our orthosis design as, in future studies, we continue to enable more hand positions in stroke patients.

As stated above, pattern classification of functional movements for stroke subjects has been studied with the ultimate goal of controlling an orthosis [86]. However, to our knowledge, this kind of classification has not been studied in conjunction with an orthosis that enables functional

movements. Our experience indicates that EMG controls for stroke patients need to be developed in conjunction with orthoses. We explore the feasibility of using pattern classification while simultaneously using an orthosis to enable hand extension.

2.2.2 Supervised EMG Controlled Prosthetics

EMG control in the field of prosthetics is well documented and can provide insights for EMG control of orthotics. In the previous section, we discussed techniques for EMG control of prosthetics that were relevant to teleoperation. We provided examples of both proportional EMG controls [54] and pattern recognition based EMG controls [63, 87]. These prosthetic control paradigms are also relevant to orthotics.

Below we review additional prosthetic EMG controls that are particularly relevant for the field of orthotics. Since the EMG signals of stroke subjects are more irregular than that of able bodied subjects, prosthetic studies which consider how to train the device and how to process the EMG signals are important.

In prosthetics, the two main approaches to train devices are system training and user training. System training adapts the control to be more accurate and can involve gains, thresholds, computer-guided training or bilateral training to provide ground truth for the system [88]. User training teaches the user to produce control signals that are easily distinguishable for the system and, in the context of EMG controls, involves teaching the user how to create consistent and distinguishable muscle patterns [87]. We believe that this latter approach can also prove valuable for the types of signals we use here, and plan to apply it in future iterations of our control.

Processing EMG signal for optimal performance has also been well studied in prosthetics. Control optimization studies have informed both signal processing [89, 90] and control hierarchy [91] of prostheses.

2.2.3 Semi-Supervised EMG Controlled Prosthetics

Traditionally, controls for prosthetics and orthotics have been supervised - trained on a relatively small test set and then that classifier is used to control the device during a longer testing session [63, 87, 86]. However, leveraging the unlabeled data from the test set to improve the classifier may make the control algorithm more robust to fatigue, different arm positions and abnormal muscle coactivation.

Semi-supervised learning has been shown to be an effective control paradigm for hand prosthetics. He et al. showed that an entropy function could be used to test the confidence of an LDA classifier and the classifier could be updated when confidence was high. They showed that this could improve classification of wrist movement across several hours of EMG testing [92]. Jain et al. also used entropy as a metric for when they should retrain their classifier, and additionally corrected their classifier using principal component analysis (PCA) cross validation and majority label correction, which assumes that the user will perform gestures as a low frequency [93]. Amsüss et al. trained an LDA classifier on EMG data, then trained another post-processing neural network which calculates the probability the the output of the classifier is correct. They use this to correct the LDA classifier, but do not actually update the original EMG classifier [94]. Chen et al. assume all predictions of the classifier are correct and use all new datapoints to update the classifier [95]. Zhai et al. assume that neighboring EMG segments are likely to belong to the same hand movement and correct the classifier predictions using a median filter on the past predictions, updating the CNN classifier when a correction occurs [96]. Sesinger et al. developed an adaptive LDA classifier with different updating schemes. They describe formulations where the learner is updated when confidence is high, when the confidence is low, and when there were rapid changes in the prediction stream [97]. Zhang et al. expanded this adaptive LDA classifier to include cycle substitution and probability weighting when updating the training dataset [98]. Liu et al. took EMG readings collected across several days, and found common characteristics among the classifiers trained on these different sessions, which allowed them to eliminate classifier retraining in later sessions [99]. Even reinforcement learning has been used for EMG based controls - Edwards et al. used general

value functions (GVFs) to predict which joint of a robotic arm the user would want to control next in order to reduce the amount of time the user spent switching between joints [100].

All of the above works are specific to EMG. Many of them leverage the low frequency nature of human hand motions in order to correct the classifiers [97, 96, 93]. However, all the above works are designed for prosthetic controls and only tested on either healthy subjects or amputees. Only three of the studies look at intersession accuracy [94, 99, 96] - for the rest, the classifier is trained on data which was collected during the same session as the test data. To our knowledge, semi-supervised learning has never been applied to a control for a hand orthosis.

One disadvantage of semi-supervised learning is that it isn't safe. If unlabeled datapoints are assigned an incorrect label and added to the training set, then performance can degrade. There have been several attempts to guarantee safe semi-supervised learning [101, 102, 103, 104]. Though these avenues are interesting, most works in the literature choose not to guarantee safety, instead leveraging the time series component of the EMG data and low frequency nature of the hand motions to avoid performance degradation.

2.2.4 Disagreement-Based Semi-Supervised Learning

Most of the semi-supervised learning for prosthetics leverages the low frequency nature of hand motions for supervision. However, there is another family of supervision which we believe could also be applied to rehabilitation robotics. Disagreement based semi-supervised learning asks multiple learners to collaborate in order to exploit unlabeled data [105]. If a learner is very confident in its prediction, it can be used to train the other learners.

Some formulations only require that one of the learners be confident about its prediction [106], while other formulations require disagreement between the different learners, and then add the example to the training set if one of the learners is much more confident about the disagreed upon example than the others [105]. Disagreement based learning relies on the differences between the learners. As the learners train each other, they become more similar, so after a time, they will not be able to improve each other any further [107].

This family of research started with the seminal work of Blum and Mitchell, which requires that there exist different views (features) which are sufficient to train a strong learner and are conditionally independent given the class [106]. These conditions are often difficult to meet with real world data. As a result, there are proposals which seek to relax these requirements [108, 109, 110].

Disagreement based supervised learning is well suited to co-exist with ensemble based learning methods. Javed et al. used cotraining to update learners, but an ensemble of all the learners was used to provide the final prediction [111]. Kolter and Maloof used an ensemble of weak learners and reduced the weight of the weak learners when they produced incorrect predictions. This is a variant of cotraining, where instead of updating the learners, the weight of incorrect learners is simply reduced. When the global prediction was wrong, a new weak learner was added [112]. Jiang and Zhang co-train learners on labeled data and unlabeled data on which the previous learners were confident and add the new learners into the ensemble [113]. Wang and Luo used ensembles to relax the requirements of co-training by training on random feature spaces [110].

To our knowledge, this family of disagreement-based semi-supervised learning has never been applied to an orthotic or prosthetic control. However, if a wearable orthosis had a sufficient number of sensors, these algorithms could improve classification accuracies for orthotic intent detection.

Chapter 3: Teleoperation Subspace

As we state in Chapter 1, teleoperation can be a faster and more effective control for robotic manipulators in highly unstructured environments than fully autonomous manipulation planners. As robots and teleoperation become more common in our everyday lives, it becomes increasingly likely that the people who will need to work with robots will not be robotic experts, but novices. Therefore, an important research direction for robot teleoperation aims to make the controls available to the operator as intuitive as possible: intuitive controls minimize the training time required for human teleoperators and can make teleoperation more accessible to novices.

For manipulation, teleoperation controls which harvest the user's hand motions, rather than using a joystick or a point-and-click interface, can provide an intuitive and user friendly interface [4], because they harness motions which are already natural to the teleoperator. An example of this workflow is shown in Figure 3.1.

Teleoperating a robot hand using a human hand as input requires a teleoperation mapping, which defines how the robot hand should move in response to movements of the human hand. Robot hand designs that are fully-actuated and highly anthropomorphic allow for a direct joint mapping to the human hand and thus are intuitive for a human to teleoperate; however, the hardware tends to be fragile and expensive. In contrast, non-anthropomorphic hands have proven to be robust and versatile in unstructured environments. However, finding an easy or intuitive mapping between the human hand and a non-anthropomorphic robot hand can be difficult, due to dissimilarities between the hands.

In this thesis, we seek to create a mapping between the human hand and a fully actuated but non-anthropomorphic robot hand that is intuitive enough to enable effective real-time teleoperation, even for novice users. In the context of collaborative manipulation, our target users are able-bodied users who are deployed in a human-robot team to complete manipulation tasks.

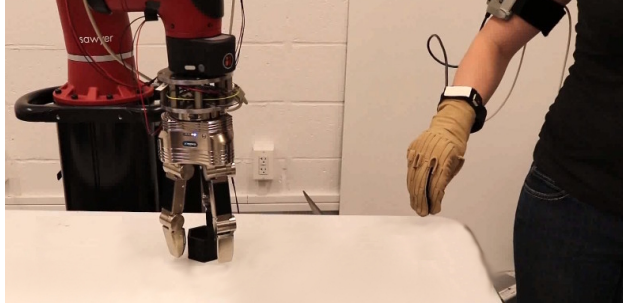


Figure 3.1: Teleoperated manipulation where the operator’s hand movements are recorded and used to drive a non-anthropomorphic robotic hand. To achieve this, a mapping between the human and robot hand kinematics is required.

We propose a subspace relevant to teleoperation. This subspace is an intermediary which allows us to map between the pose spaces of different hands. By projecting the pose of the master hand into the teleoperation subspace, which it shares with the slave hand, and then projecting from the teleoperation subspace into the pose space of the slave hand, we can enable real-time teleoperation.

Our mapping is independent of the master-slave pairing, so the mapping between teleoperation subspace and pose space of a robot does not have to be redefined with every new human teleoperator. Furthermore, the teleoperation subspace is low dimensional, which allows us to explore EMG-based controls (Chapter 6).

In this chapter, our main contribution is the introduction of a continuous, low-dimensional teleoperation subspace as an intuitive way to map human hand poses to robot hand poses for teleoperation. We posit that this method allows for intuitive teleoperation as long as both the master and the slave hand poses can be projected into this subspace.

Here, we introduce the teleoperation subspace and present a general formulation to map between the human hand and the robot hand. In the next three chapters, we explore different methods of populating the variables required to create a teleoperation subspace mapping.

3.1 Definition

As a general concept, we posit that, for many hands, a three dimensional manifold T can encapsulate the range of movement needed for teleoperation. The three basis vectors of T have

an intuitive correspondence with hand shape. One corresponds with how far apart the fingers are spread, another with the size of the object a hand can grasp, and another with how curled the fingers are. We refer to these basis vectors as α , σ , and ϵ , respectively.

We chose these bases on intuition, guided by Santello’s research of postural synergies [28]. Since Santello et al. used principle component analysis (PCA), a linear dimension reduction method, to find postural synergies, we also assume that mapping between pose space and teleoperation subspace is linear. We do not use the exact postural synergies found by Santello as the basis vectors for our subspace because we empirically found it easier to map to robotic hands (note that Santello’s synergies are strictly human specific) if we ‘decoupled’ Santello’s two synergies into three. It is also easier to explain the control to novices with decoupled movements.

We assume that many hands will be able to project their pose space into T . If this projection is possible, T is embedded as a subspace in the pose space of the hand. T is thus a subspace “shared” by all hands that can project their pose space into T . If the user can construct a projection matrix which projects pose space to teleoperation subspace in a meaningful way, our method will enable teleoperation. Teleoperation which uses this subspace as an intermediary between joint spaces enables the master and the slave hands to form similar shapes around a scaled object. Experimentally, we will show that this is the case for at least the human hand, the Schunk SDH, and a two-fingered gripper. We theorize that T is also relevant to teleoperation for other hands.

To teleoperate using T , there are two steps:

1. Given joint values of the master hand, find the equivalent pose ψ in teleoperation subspace T .
2. Given ψ computed above, find the joint values of the slave hand, and move the slave hand to these values.

These steps are illustrated in Figure 3.2. In order to enact the teleoperation steps, we must first define the mapping between T and the relevant pose spaces.

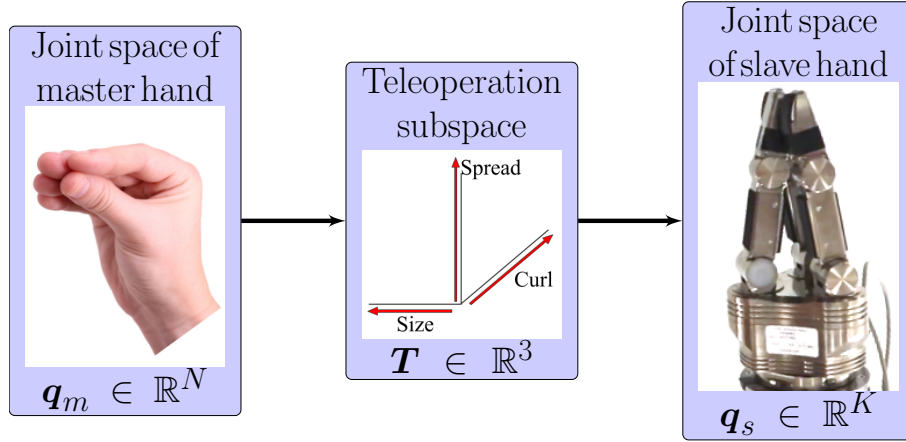


Figure 3.2: Steps to enable real time teleoperation using teleoperation subspace with a dataglove as input.

3.2 Teleoperation Subspace Mapping

For a given hand with N joints, projecting from joint space $q \in \mathbb{R}^N$ (we use pose space and joint space interchangeably) into teleoperation subspace T requires an origin pose $o \in \mathbb{R}^N$, a projection matrix $A \in \mathbb{R}^{N \times 3}$, and a scaling factor $\delta \in \mathbb{R}^3$.

Origin o To project between joint space and T , we require a hand-specific, “neutral” origin pose $o \in \mathbb{R}^N$.

$$o = [o_1, o_2, \dots, o_N] \quad (3.1)$$

This represents a hand position which will standardize the data as we project between joint space and T . The origin pose of the master is arbitrary; however, it is crucial that the origin pose of the slave corresponds to the master’s origin. The two hands should assume approximately the same shape while positioned at their respective origins.

Projection Matrix A The projection matrix $A \in \mathbb{R}^{N \times 3}$ is hand specific and consists of three basis vectors $\alpha_H, \sigma_H, \epsilon_H \in \mathbb{R}^N$. Whereas α, σ , and ϵ represent the general concept of a hand motion, α_H, σ_H , and ϵ_H are the projection of that motion into the pose space of a specific hand H .

$$\mathbf{A} = [\boldsymbol{\alpha}_H, \boldsymbol{\sigma}_H, \boldsymbol{\epsilon}_H] \quad (3.2)$$

$$\boldsymbol{\alpha}_H = [\alpha_{H1}, \alpha_{H2}, \dots, \alpha_{HN}]^\top \quad (3.3)$$

$$\boldsymbol{\sigma}_H = [\sigma_{H1}, \sigma_{H2}, \dots, \sigma_{HN}]^\top \quad (3.4)$$

$$\boldsymbol{\epsilon}_H = [\epsilon_{H1}, \epsilon_{H2}, \dots, \epsilon_{HN}]^\top \quad (3.5)$$

Scaling Factor δ We wish to normalize such that any configuration in pose space will project to a pose in \mathbf{T} whose value is less than or equal to 1 along each of the basis vectors. We therefore require a scaling factor $\delta \in \mathbb{R}^3$ to normalize the projection:

$$\boldsymbol{\delta} = [\delta_\alpha, \delta_\sigma, \delta_\epsilon]. \quad (3.6)$$

To calculate $\boldsymbol{\delta}$, we evaluate poses which illustrate the extrema of the hand's kinematic limits along the basis vectors. For example, the maximum and minimum values along $\boldsymbol{\sigma}$ are illustrated by projecting poses where the hand is holding the largest object possible and the smallest object possible from pose space into \mathbf{T} . Once we select the illustrative poses for the hand, we project these poses from pose space into \mathbf{T} using $\boldsymbol{\psi} = (\mathbf{q} - \mathbf{o}) \cdot \mathbf{A}$, where $\boldsymbol{\psi} \in \mathbf{T}$. From this set of poses in \mathbf{T} , we find the minimum and maximum values along each axis. Along $\boldsymbol{\alpha}$, the minimum and maximum are referred to as α_{min} and α_{max} , respectively. From these values, we calculate δ_α as:

$$\alpha_{range} = abs(\alpha_{max}) + abs(\alpha_{min}) \quad (3.7)$$

$$\delta_\alpha = \begin{cases} 0 & \text{if } \alpha_{range} = 0 \\ 1/\alpha_{range} & \text{otherwise.} \end{cases} \quad (3.8)$$

Finding δ_σ and δ_ϵ uses the same calculation.

$\boldsymbol{\delta}$ normalizes the projection from pose space to \mathbf{T} ; however, to project from \mathbf{T} back to pose

space, we require an inverse scaling factor δ^* :

$$\delta^* = [\delta_\alpha^*, \delta_\sigma^*, \delta_\epsilon^*] \quad (3.9)$$

$$\delta_\alpha^* = \begin{cases} 0 & \text{if } \delta_\alpha = 0 \\ 1/\delta_\alpha & \text{otherwise} \end{cases} \quad (3.10)$$

where we find δ_σ^* and δ_ϵ^* with similar calculations.

3.3 A Complete Teleoperation Algorithm

To project between teleoperation subspace \mathbf{T} and joint space \mathbf{q} , we use the hand-specific matrix \mathbf{A} , the origin \mathbf{o} , and the scaling factor δ :

$$\boldsymbol{\psi} = ((\mathbf{q} - \mathbf{o}) \cdot \mathbf{A}) \odot \delta \quad (3.11)$$

$$\mathbf{q} = ((\boldsymbol{\psi} \odot \delta^*) \cdot \mathbf{A}^\top) + \mathbf{o} \quad (3.12)$$

where \odot represents element-wise multiplication.

To use \mathbf{T} for teleoperation, Eq. 3.11 projects from the master hand's pose space into the shared teleoperation subspace and then Eq. 3.12 projects from the shared teleoperation subspace into the slave hand's pose space (Figure 3.2).

So, given the joint angles of the master hand, we are able to calculate the joint angles of the slave hand using:

$$\mathbf{q}_s = (((\mathbf{q}_m - \mathbf{o}_m) \cdot \mathbf{A}_m) \odot \delta_m \odot \delta_s^*) \cdot \mathbf{A}_s^\top + \mathbf{o}_s. \quad (3.13)$$

3.4 Summary

We describe a method for teleoperating a non-anthropomorphic, fully actuated robot. We use a subspace relevant to teleoperation as an intermediary between hand pose space and robot pose

space. Projecting from the control input into the teleoperation subspace, and then projecting from the teleoperation subspace into the pose space of the slave hand will enable teleoperation where the two hands make similar poses around a scaled object.

Our method makes several assumptions in order to teleoperate. Guided by postural synergies, we assume that the teleoperation subspace we have defined encapsulates all of the information needed to teleoperate a slave hand with only three basis vectors, and that mapping between joint space and teleoperation subspace is linear.

Despite these assumptions, the potential applications of our method are broad. Unlike other teleoperation methods which use low dimensional subspaces, our method does not require the user to generate a large number of corresponding poses for the human and the robot. For each hand, the user must only provide three variables to enable teleoperation. In our method, the mapping between T and pose space is independent of the master-slave pairing. If the mapping variables for multiple robots have been defined, then each human teleoperator only needs to provide the mapping for their specific hand (through a set of calibration poses) in order to teleoperate any of the robotic hands. It also is worth noting that, although we describe teleoperation in the context of a human controlling a robot, there is nothing in our method which requires this, and our method could theoretically be used to map poses between two robotic hands.

In this chapter, we define the teleoperation subspace at a conceptual level. We describe a complete algorithm for teleoperation and the variables needed to project between the pose spaces of different hands. The specifics of how these variables will be populated will be critical to the performance of our teleoperation control.

In the next chapter, we describe a concrete method for creating the teleoperation mapping empirically, leveraging the intuition of the person creating the mapping to populate the required variables.

Chapter 4: An Empirical Mapping for the Teleoperation Subspace

In the last chapter, we defined a subspace which we hypothesized was relevant to teleoperation. At a conceptual level, each of the basis vectors that define the subspace corresponds to a hand motion: hand opening, finger curl, and finger spread. While these concepts are natural for the human hand, we need to also define them in the context of non-anthropomorphic robot hands.

In this chapter, we show that this definition can be done empirically: in this formulation, the person creating the teleoperation mapping defines what the motions of ‘open’, ‘curl’, and ‘spread’ mean for a specific robot hand. In this way, the mapping is tied to hand kinematics, since the hand motions mean different things for different hands. We show that this empirical mapping can be created following a series of simple steps (Section 4.1) and leads to effective teleoperation for novices.

The general formulation we describe in the previous chapter defines the variables needed to map from pose space to the teleoperation subspace. Using a dataglove as a control input allows us to measure the pose (joint angles) of the human hand reliably and accurately. Unless otherwise stated, when mapping the human hand into the teleoperation subspace, we will use a dataglove as the control input.

Experimentally, we prove that our empirical method using a dataglove as input is effective and intuitive using a robot hand with highly non-anthropomorphic kinematics. The experiments in this chapter show that our method allows novice teleoperators to pick and place objects significantly faster than state-of-the-art teleoperation mapping methods. In the next chapter we will extend our validation of the empirical mapping and show that it can be applied to multiple robotic hands and multiple tasks.

Our main contributions are as follows: we provide an empirical method to define the teleoperation subspace and to create a projection into the subspace. This method requires the user to

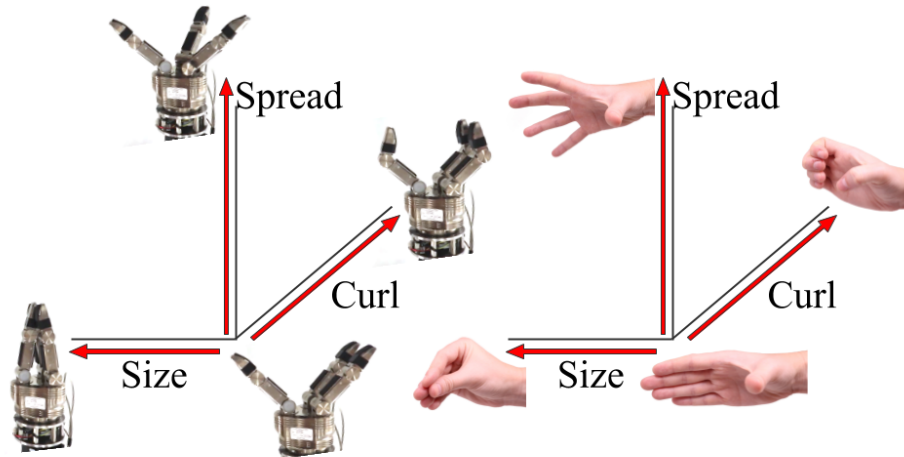


Figure 4.1: The teleoperation subspace used as an intermediary between the pose spaces of different hands. Here we show how the motions that are associated with the subspace basis vectors can be intuitively defined by a user, based on the hand’s kinematics.

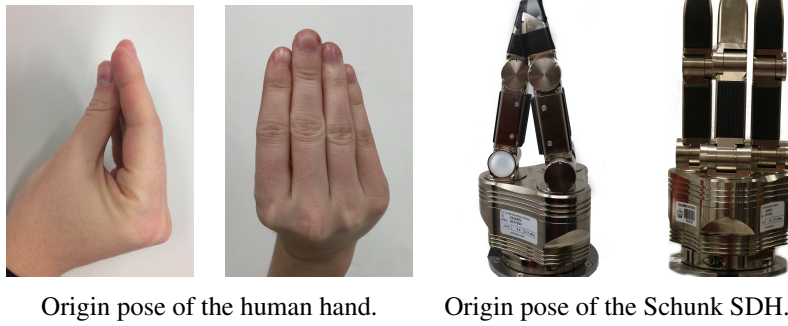
define the hand motions for the subspace using their intuition and based on the hand’s kinematics. Experimentally, we prove that this method is effective and intuitive using a robot hand with highly non-anthropomorphic kinematics.

4.1 Empirically Defining the Subspace Mapping

When defining the subspace mapping empirically, the user must define the motions associated with each basis vector in the teleoperation subspace using their intuition, based on the hand’s kinematics. Figure 4.1 shows this for both the human and the Schunk SDH hand.

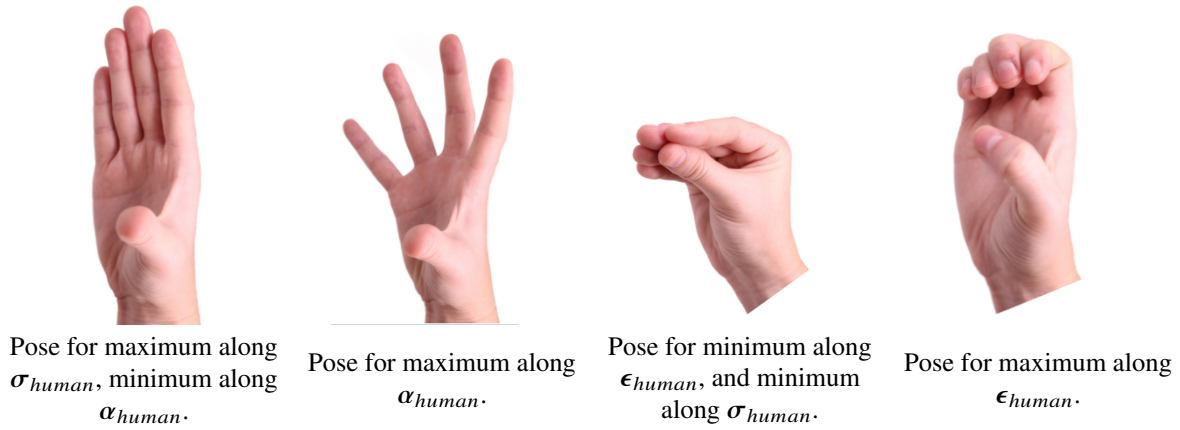
The teleoperation subspace mapping can be created empirically through a relatively simple process. For each hand, the user must:

- Select an origin pose. Figure 4.2 shows the pose we chose for the human hand and the Schunk SDH robot in our experiments.
- Determine poses which illustrate the extrema of the hand’s kinematic limits along the basis vectors. It is up to the user to determine poses which illustrate the full range of values for each basis vector. Figure 4.3 shows the poses which demonstrate these ranges for the human hand.



Origin pose of the human hand. Origin pose of the Schunk SDH.

Figure 4.2: Origin poses of two example hands.



Pose for maximum along σ_{human} , minimum along α_{human} .

Pose for maximum along α_{human} .

Pose for minimum along ϵ_{human} , and minimum along σ_{human} .







Pose for maximum along ϵ_{human} .

Figure 4.3: Poses which demonstrate the human hand's kinematics limits along the basis vectors of T . To see the poses in the context of the teleoperation subspace, refer to Figure 4.1.

- Define what the hand motions (finger spread, finger curl, and hand opening) mean in the context of the hand's kinematics, then identify which joints contribute to that motion. This is a winner-take-all approach, so a joint may only contribute to a single motion. We set joints which adduct the fingers to 1 in α_H , joints which open the hand to 1 in σ_H , and joints which curl the fingers to 1 in ϵ_H . We then normalize the vectors to create A . Table 4.1 shows this process for the Schunk SDH hand.

Creating the the empirical mapping is a simple, winner-take-all, three step approach. Despite this simplicity, we show experimentally that these calculations are sufficient to meaningfully project pose space into T in a way that enables teleoperation for novice users.

Table 4.1: Process to empirically define the projection matrix for the teleoperation subspace mapping

Hand Motion	Motion Defined by Hand-Specific Kinematics	Joints which affect the motion	Basis vector
Finger Spread			$\alpha_{schunk} = [1, 0, 0, 0, 0, 0, 0]^T$
Hand Opening			$\sigma_{schunk} = [0, 0.577, 0, 0.577, 0, 0.577, 0]^T$
Finger Curl			$\epsilon_{schunk} = [0, 0, 0.577, 0, 0.577, 0, 0.577]^T$

4.2 State-of-the-Art Comparisons

Here, we review our mapping and the state-of-the-art comparisons which we use in our experiments.

Subspace Mapping: This is our mapping which uses Eq. 3.13 to transform the joint values from the dataglove into joint values for the slave hand using T as an intermediary between the two pose spaces.

Fingertip Mapping: We chose fingertip mapping as a comparison because it is applicable to precision grasps, particularly with smaller objects. The fingertip mapping method was designed as follows: first, we found the cartesian positions of the thumb, index, and ring fingers of the human

Cyberglove Sensor		Robotic Hand Joints	
Joint Label	Name	Joint Label	Name
a	Thumb adduction	1	Thumb proximal flexion
b	Thumb distal flexion	2	Thumb distal flexion
e	Index/Middle adduction	3	Finger 1 adduction
c	Index proximal flexion	4	Finger 1 proximal flexion
d	Index medial flexion	5	Finger 1 distal flexion
e	Index/Middle adduction	6	Finger 2 adduction
f	Middle proximal flexion	7	Finger 2 proximal flexion
g	Middle medial flexion	8	Finger 2 distal flexion



Figure 4.4: Joint mapping scheme between the Cyberglove and the custom robotic hand.

hand with respect to the wrist by attaching trakSTAR sensors to each of the listed fingers and to the back of the wrist. We calculated transforms between the wrist and finger sensors to find the finger position in the hand frame. We multiplied these positions by a scaling factor of 1.5, the ratio between an average human finger and one of the robotic fingers. We rotated the positions from the human hand frame into the robotic hand frame. We translated the coordinates from the robotic hand frame into the finger frame to find the desired robotic fingertip positions. Finally inverse kinematics determined the joint angles which placed the fingertips at these positions. This process is documented elsewhere [20].

Joint Mapping: We chose joint mapping as the second comparison method because it is applicable to power grasps. We also predicted that explicit control over individual joints of the robotic fingers would be intuitive for novice users. Rosell et al. teleoperated the Schunk SDH using joint mapping by assigning the joints of all fingers of the robot hand to correspond the joints of one

finger of the human hand, with finger adduction set to a constant value [114]. We instead chose to map each joint of the robot finger to a separate joint of the human hand so as not to limit the subject’s ability to perform different grasp types.

To implement joint mapping, we assigned each of the robot joints to a corresponding human hand joint. This mapping can be found in Figure 4.4. We rotated the joint angles of the human hand received from the Cyberglove so they aligned with the robotic hand, and set the joints of the robot hand to these rotated angles.

Preliminary tests showed teleoperation is difficult if the robot thumb’s proximal joint maps to the human thumb’s metacarpophalangeal (MCP) joint. We therefore mapped the robot thumb’s proximal joint to the human thumb’s adductor.

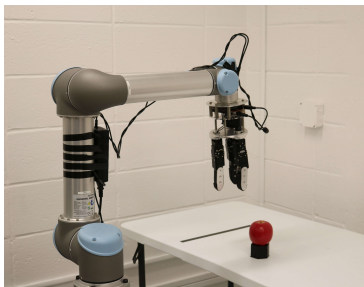
4.3 Experiments

In our experiments, we wish to prove that the teleoperation subspace we defined in Section 3.1 is relevant to teleoperation. We also wish to show that the control we have developed is intuitive for novice users.

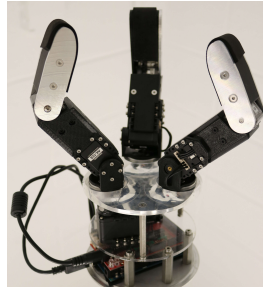
Experimentally, we prove that T is relevant for teleoperation for at least the human hand and a non-anthropomorphic robot hand, similar to the Schunk SDH. We theorize that T is also relevant to teleoperation for other hands.

Experimental Setup To show that the proposed method is intuitive for novice users, experiments were performed with five healthy subjects. Two subjects were female, three subjects were male, all were aged between 23 and 28, and all were novice robot teleoperators. Subjects gave their informed consent and the study was approved by the Columbia University IRB.

We compared our teleoperation method with two state-of-the-art teleoperation techniques - joint mapping and fingertip mapping (Section 4.2). These methods were chosen from the state-of-the-art as being the most applicable to our problem: intuitive for novice users, and able to use different grasping types (power and precision, for example) to grasp objects of various shapes and



(a) Experimental set-up with UR5 arm, robotic hand and table.



(b) Close-up of the custom robotic hand used in experiments.



(c) Objects used in pick and place tasks.

Figure 4.5: Setup for pick and place experiments.

sizes. We used time to complete a pick and place task as a metric for usability and intuitiveness.

The slave hand of our teleoperation system is a custom-built robotic hand with kinematics similar to the Schunk SDH. It has a thumb finger and two opposing fingers; all three fingers have two links. The hand is fully actuated and the two opposing fingers adduct independently.

We attached the slave hand to a Universal Robot (UR5) arm. The UR5 stands in front of a table where the grasping objects are placed one at a time during testing. Figure 4.5 shows our setup, a close-up of the slave hand, and the object set used for the experiments.

The novice teleoperator stands next to the robot and teleoperates based on visual feedback. Attached to the user's hand is an Ascension 3D Guidance trakSTAR™ system, which tracks hand position and orientation. The UR5 follows the position and the orientation of the trakSTAR with a cartesian controller. We control the robotic hand using either joint, fingertip or teleoperation subspace mapping.

Subjects participated in two testing sessions. During the first session, we presented the subjects with the subspace and joint mapping teleoperation methods. During a second session, we presented subjects with the fingertip mapping control. The order in which the joint mapping and subspace mappings were presented to the users was randomized; however, all subjects performed the fingertip mapping last. Nominally, subjects should have been able to use the fingertip mapping faster because they were already familiar with the arm and the hand by the second session.

During the first session, the teleoperator wore a dataglove (a Cyberglove) with a trakSTAR

sensor attached to the back of the hand. The Cyberglove provided the joint angles of the human hand to the mapping control method. The trakSTAR provided hand position and orientation to the UR5 controller.

During the second session, the teleoperator wore a trakSTAR sensor attached to the back of the hand, as well as additional trakSTAR sensors attached to three of the fingers. All four sensors were used for the fingertip mapping and the sensor on the back of the hand again provided hand position and orientation to the UR5 controller.

We placed a series of objects on the table one at a time and asked subjects to pick them up and move them across a line 0.3 meters away. Shorter objects were placed on a stand so as to facilitate easier grasping with the large robotic fingers. The objects for the pick and place tasks were: a box, a ball, a stack of Legos, a roll of tape, a plastic apple and a mesh bag of marbles (Figure 4.5c). We selected these object to illustrate a variety of grasping types. We did not instruct the users how they should grasp the objects. After the subject placed an object in the designated area, we reset the UR5 to a neutral position before the next pick and place task.

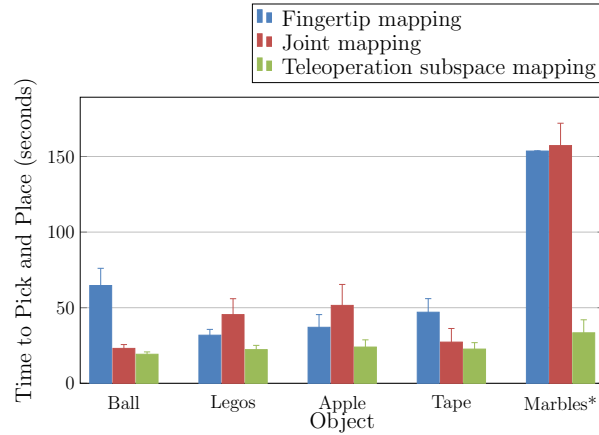
At the beginning of each session, the subject was given three minutes to move the arm, but not the hand. We did not want the user's unfamiliarity with the arm to bias the results towards the second control method during the first session. Furthermore, the first object the user attempted to grasp (the box) was labeled as a training object and not included in the evaluation.

We explained the mapping method to the teleoperator immediately before they were asked to pick and place objects using that control.

4.4 Results

We timed how long it took for the user to pick and place each object. If the user did not complete the task in four minutes, they were considered to be unable to pick up the object.

All five novices picked up the ball, legos, apple, and tape in the allotted time. All of the users picked up the marbles using subspace mapping. However, in the allotted time, only two users picked up the marbles using joint mapping and only one user picked up the marbles using fingertip



Mapping Method	Object					Average
	Ball	Legos	Apple	Tape	Marbles*	
Fingertip	64.67	31.83	37.05	47.02	153.6	62.27
Joint	23.09	45.43	51.53	27.25	157.21	56.66
Subspaces	19.22	22.27	23.97	22.63	33.4	27.52

Figure 4.6: Average time (in seconds) to pick and place objects using different teleoperation controls in (top) graphical and (bottom) numerical format. * denotes that the average was calculated with a smaller sample size for some of the mapping methods of that object.

mapping. We therefore calculated the average time to pick up the marbles from a smaller sample size for the fingertip and joint mappings (denoted by an asterisk in the results figures).

Figure 4.6 shows the average time a novice took to pick and place an object using each of the control methods. We show our results numerically and graphically.

We found that novices using fingertip mapping took 2.75 times longer to pick and place an object than they took using teleoperation subspace mapping. On average, novice teleoperators using joint mapping took 2.51 times longer to complete a task than they took using subspace mapping.

For larger objects, like the ball and the tape, fingertip mapping took on average 2.75 times longer than subspace mapping, whereas joint mapping only took 1.2 times longer. For smaller objects, like the legos and the apple, fingertip mapping took 1.49 times longer than subspace mapping, whereas joint mapping took on average 2.09 times longer than subspace mapping.

Discussion: The results show that teleoperation is possible using T as an intermediary between the pose spaces of two dissimilar hands. This result has several meanings: first, it proves that our projection algorithm and our methods for calculating the projection, while simple, are sufficient to enable teleoperation. Second, it proves that T is relevant to teleoperation for at least the human hand and the custom robotic hand used in our experiments. For these two hands, the teleoperation subspace encapsulates the range of motion needed to teleoperate the slave hand with the human hand.

Our experiments also show that teleoperation subspace mapping allows novice users to complete tasks more quickly than they are able to using either joint mapping or fingertip mapping. A novice user asked to pick up an object will, on average, complete the task 2.75 times slower using fingertip mapping and 2.51 times slower using joint mapping than they would using teleoperation subspace mapping. Using time to task completion as a metric for intuitiveness, these experiments prove that our subspace mapping is more intuitive for novice users than state-of-the-art teleoperation mappings.

Fingertip mapping is applicable to precision grasps and joint mapping is applicable to power grasps [12]. Our experiments confirm this holds true for novice users. For larger objects, like the ball and the tape, which were selected to illustrate power grasps, joint mapping allowed the user to complete the task in half the time of fingertip mapping. For smaller objects, like the stack of Legos and the apple, which were selected to demonstrate precision grasps, the reverse was true - joint mapping took 1.43 times longer than fingertip mapping. Subspace mapping outperformed fingertip mapping and joint mapping for both precision grasps and power grasps, which shows that it is versatile enough to be applicable to different grasp types.

Overall, our experiments showed teleoperation subspace mapping to be faster and more versatile when presented with a variety of objects than state-of-the-art mapping methods.

4.5 Summary

We present an empirical algorithm to generate a mapping for the teleoperation subspace that leverages the user’s intuition to define hand motions for a specific kinematic configuration.

Our experiments show that the proposed teleoperation subspace is indeed relevant to teleoperation and that it can enable online teleoperation of a non-anthropomorphic hand. We also show that our teleoperation subspace allows novice teleoperators to pick and place objects faster than state-of-the-art teleoperation methods.

In this chapter, we assume that the user will be able to use intuition to generate the three variables needed for each hand to create the mapping from joint space to teleoperation subspace. We have provided a clear guideline so the user will be able to calculate these variables, and the process we describe is fairly simple. If the hand is so non-anthropomorphic that even human intuition cannot find a clear mapping into teleoperation subspace, our method will not apply.

In the next chapter, we will propose an algorithmic way to create the mapping for the teleoperation subspace, where the user does not have to provide intuition about hand kinematics. We also provide further validation for the teleoperation subspace and the empirical mapping proposed in this chapter. Experimentally, we show that both the algorithmic and the empirical mappings can provide intuitive teleoperation for novice users for two different robotic hands and for two different manipulation tasks.

Chapter 5: An Algorithmic Mapping for the Teleoperation Subspace

One shortcoming of using an empirical mapping for the teleoperation subspace is its reliance on human intuition: effective teleoperation could be attributed to either the structure of the subspace, or simply to the hand-specific intuition provided by the person creating the mapping. We strive to show that the teleoperation subspace which we propose can also be defined without such kinematic-specific intuition.

Therefore, in this chapter, we propose a second, algorithmic method where we formalize the hand motions used to define the teleoperation subspace described in Chapter 3 using objects. Rather than considering, for example ‘hand opening’ as an intuitive concept to be defined by the person creating the mapping, this paradigm considers hand opening as the hand grasping a series of incrementally larger objects. In this way, we can use a set of objects to provide the same understanding of hand motions as the user provided in the empirical method.

This definition of the subspace is done exclusively through an object set, and is kinematically-independent. It lends itself to the algorithmic creation of a teleoperation mapping for any hand. We introduce a method which uses this algorithmic definition to generate subspace mappings for hands in a fully automated fashion (Section 5.1). We aim to show that this algorithmic mapping also enables effective teleoperation for novices, implying that the value of the teleoperation subspace does not derive exclusively from the kinematic-specific human intuition used to create it.

In Chapter 4, we introduced the concept of a teleoperation subspace defined exclusively via empirical mapping, and validated it with teleoperation experiments on a single robotic hand for a single task. In this chapter, we show that the subspace can be defined in a kinematically-independent fashion by considering variations in the grasped object shape, and introduce a fully automated process for creating mappings into this subspace. We also expand our validation of the subspace considerably with experiments that test both the empirical mapping from the previous chapter, and

the proposed algorithmic mapping on multiple manipulation tasks on two kinematically distinct robotic hands.

Though there has been a significant effort in the literature to map between human and robotic hands, and to use low dimensional spaces in order to achieve this mapping, to our knowledge, we are the first to validate with real-time experiments that the subspace we propose is relevant to teleoperation for multiple robotic hands, as well as the human hand.

The main contributions of this chapter are:

- We provide an algorithmic method for defining the teleoperation subspace from Chapter 3. This definition is independent of hand kinematics and lends itself to an automated algorithm which can create a mapping for hands of various kinematics. We are the first to show that an automated method for generating a teleoperation mapping can enable online teleoperation which is intuitive for novices.
- We show experimentally that the subspace is relevant to teleoperation for two different non-anthropomorphic hands and for two different manipulation tasks. These experiments demonstrate that our mappings created empirically and algorithmically allow novice teleoperators to pick and place objects and perform in-hand manipulation as fast as or faster than state-of-the-art teleoperation mapping methods using robot hands with non-anthropomorphic kinematics. As we discuss in Chapter 2.1.4, in most literature on hand teleoperation, mappings are only validated on one or two expert users and on one robotic system. We validate the teleoperation subspace mappings on ten novice users and two different robotic hands.

5.1 Algorithmically Defining the Subspace Mapping

We encourage the reader to review Chapter 3, where we define the teleoperation subspace and the variables needed to create a mapping from pose space into the subspace T .

In the previous chapter, we rely on a human to look at the hand's kinematics, define each of the motions associated with the different subspace basis vectors, and then determine which joints

contribute to that motion.

We would like to demonstrate that we can define the subspace in a way which is independent of hand kinematics. We also hypothesize that this subspace definition allows us to create a teleoperation subspace mapping for a hand automatically (i.e. an algorithmic mapping). If the algorithmic mapping can enable teleoperation for novices, this would demonstrate that the value of the teleoperation subspace does not derive exclusively from the human intuition used to create it.

To create a subspace mapping algorithmically, we must formalize the notion of a hand motion in a way that does not depend on the hand's kinematics. We do this using objects. Hand opening can be thought of as the hand grasping a series of objects that grow incrementally larger. Spreading the fingers results from the hand grasping a series of objects whose curvature increases incrementally. Finger curl is binary, and can be defined as the difference between a precision grasp and a power grasp for the same object.

Based on this formalized notion of hand movements, we can use object characteristics to predict the location of a grasp in T . We posit that when a hand, regardless of kinematics, is holding an object, we can predict where the resulting grasp will lie in the teleoperation subspace, based on the object's size, shape, and the type of grasp used. This is illustrated in Figure 5.1.

If we can use the object's characteristics to predict where a grasp will lie on the subspace, we can create a set of objects which we predict will result in grasps along the basis vectors of T . Regardless of a hand's kinematics, when the hand holds any of the objects in this set, the resulting grasp will lie along one of the basis vectors of T . The object set we design consists of 8 objects and is described in more detail in Section 5.1.1.

If a hand of any kinematic configuration grasps all the objects in our set, the result will be a set of grasps in the pose space of that hand, but that we predict can be used to find the basis vectors of T . So, given a hand with a specific kinematic configuration, for each of the objects in our object set, we can generate a set of grasps \mathcal{G}_{object} . Each of the grasps \mathbf{g} in \mathcal{G}_{object} shows one possible way for a hand to grasp that object in a stable configuration. Each grasp \mathbf{g} is an N dimensional vector,

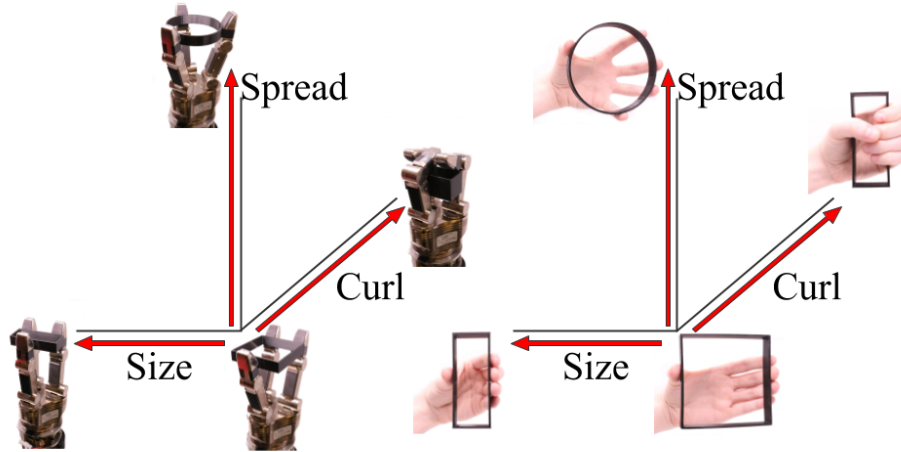


Figure 5.1: We can formalize the hand motions that define the subspace with objects, then use this definition to predict where a grasp for that object will lie in the subspace, regardless of the hand’s kinematics. This figure shows the human hand and the Schunk hand grasping the same set of objects. When an object is held by either hand, the resulting grasp will lie at the same location in the subspace T .

where N is the number of degrees of freedom for the hand. Once we have generated grasps for each of the objects, we can combine these individual sets into one grasp set \mathcal{G} , which encompasses all the objects:

$$\mathcal{G} = \mathcal{G}_{object1} \cup \mathcal{G}_{object2}, \dots, \mathcal{G}_{object8}$$

$$\mathcal{G}_{object1} = \{\mathbf{g}_{object1}^1, \mathbf{g}_{object1}^2, \dots\}, \mathbf{g} \in \mathcal{R}^N.$$

Since \mathcal{G} is a set of grasps in pose space which spans T , we can find a model of T by fitting a subspace to \mathcal{G} . The model for T provides us with the subspace mapping needed to teleoperate the hand. The model of the subspace includes the origin and the directions of the basis vectors, which translate to \mathbf{o} and \mathbf{A} in the teleoperation mapping. We can then find δ with a simple iterative method.

Once the object set has been designed, algorithmically creating a subspace teleoperation mapping requires three steps. For both the master and the slave hand, we need to:

- Generate a set of grasps \mathcal{G} where the hand is grasping each of the objects in the object set.

Table 5.1: Object set
Dimensions (in mm)

Identifier	Object Primitive	x	y	z	Grasp Type	Predicted location of grasp in T
1	Disk	70	70	10	Precision	$\psi = [1, 0.5, 0]$
2	Disk	110	110	10	Precision	$\psi = [1, 1, 0]$
3	Box	45	300	10	Precision	$\psi = [0, 0, 0]$
4	Box	70	300	10	Precision	$\psi = [0, 0.5, 0]$
5	Box	100	300	10	Precision	$\psi = [0, 1, 0]$
6	Disk	70	70	10	Power	$\psi = [1, 0.5, 1]$
7	Box	45	300	10	Power	$\psi = [0, 0, 1]$
8	Box	70	300	10	Power	$\psi = [0, 0.5, 1]$

- Fit a subspace to the grasps. The subspace model provides us with the projection between T and joint space.
- Use an iterative approach to find δ .

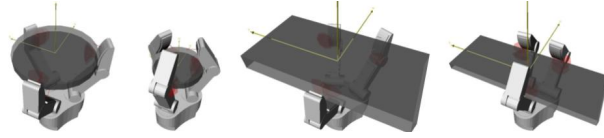
Once the mapping has been generated for a given hand, it does not have to be generated again for a new master slave pairing. For example, once we generate the human mapping, it will work with slave hand mappings that we generate in the same way.

We discuss the design of the object set, and the steps needed to implement teleoperation in the sections below.

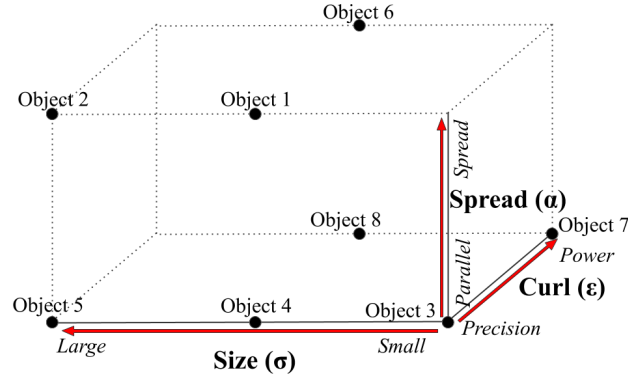
5.1.1 Object Set

We hypothesize that we can design a set of objects to elicit grasps which lie along the basis vectors of T . Table 5.1, and Figure 5.2 show the objects in our set, and where in the subspace we predict hands grasping those objects will lie.

The object set consists of eight objects. We use disks and boxes as our shape primitives. We specify the type of grasp (power or precision) which must be used with each object, in order to guarantee the grasp's location along the curl basis vector of T . In the set, there are objects that have the same dimensions, but are grasped with a different grasp type.



(a) Four example objects from our set, held by the Schunk SDH hand. From left to right: Object 2, Object 6, Object 5, and Object 8.



(b) A visualization of where we predict grasps will lie in the teleoperation subspace when a hand is holding objects in the object set. This image can be used to interpret the last column of Table 5.1, where $\psi = [\alpha, \sigma, \epsilon]$.

Figure 5.2: Object set visualizations.

The approach direction of the hand is along the z axis, and we orient the objects in the same way relative to the hand.

We note that using a different object set would create a different subspace that would not necessarily correspond to T . We have designed this object set specifically to fit our subspace. We selected simple objects to minimize the variance of the grasps that could be selected, both by the human and by the grasp planner. The process of object selection is driven by our intuition, but our experiments show that fitting a subspace to this object set results in a subspace which is relevant to teleoperation (though we do not guarantee that it is isomorphic to the T generated empirically).

5.1.2 Grasp Generation

Once we define our object set, we generate a set of grasps \mathcal{G} , which demonstrate how a hand of a specific kinematic configuration can hold the objects in our set in stable configurations. For

robotic hands, we generate \mathcal{G} using a grasp planner, and for human hands, we use human subjects.

We acknowledge that there are many ways to grasp an object. To compensate, we generate multiple grasps for each object, and use a subspace fitting method which is robust to outliers. In this way, we assume that we have sufficiently sampled grasps for the object set which would fall along the basis vectors of the subspace.

Robot Datasets: To generate the robot grasps, we use a grasp planner provided by the *GraspIt!* simulator [115]. Given a hand and an object, the planner returns grasp configurations in which the hand stably grasps that object, ranked by the epsilon quality metric [116]. This quality metric is a geometric method that determines the total space of possible wrenches, within certain friction constraints, for a given grasp.

For grasp planning we apply a random search: we randomly sample an object pose (3 dimensions, we do not consider object rotation) that lies within the workspace of the hand. We also sample pre-grasp pose joint angles (N dimensions, where N is the number of degrees of freedom of the hand) that lie within the joint limits. We then close the fingers until they make contact with the object and evaluate the resulting grasp. In order to ensure robustness of the resulting grasps, particularly with respect to small deviations in object and pre-grasp pose, we also evaluate the grasps that arise when small perturbations are applied. Specifically we apply both positive and negative disturbances along each coordinate axis of the search space individually. Thus, for the $3 + N$ dimensions from which candidate object and pre-grasp poses are sampled, we evaluate a total of $3(3 + N)$ grasps. We choose the minimum quality encountered across these trials to represent the sampled grasp overall. This process is repeated until an iteration limit is reached and the sampled grasps are stored in a database.

Given a hand and an object, the planner returns up to 1,000 stable grasp configurations for that object. We parse the dataset by removing grasps which are closer than a parsing threshold ξ in Euclidean distance to a higher ranking grasp. ξ starts at 0.0 and is increased in intervals of 0.1. Each time ξ increases, the dataset is re-parsed. This is repeated until each object has fewer than

20 grasps remaining. We note that this means that the final parsed grasp set may have a different number of grasps for each object.

The object set we present is sized to the human hand. However, some robot hands are larger than the human hand. We therefore scale the objects based on hand size. For example, the fingers of the Schunk SDH are approximately 1.5 times the size of the average human finger. We therefore multiply the dimensions of the objects by 1.5 when we plan grasps for the Schunk SDH.

Human Dataset: Our grasp planner does not have a robust model of the human hand, so we instead generate a dataset for the human hand using grasps generated by test subjects.

Subjects were asked to don an instrumented dataglove (a Cyberglove III) and grasp objects in the object set. After the subjects grasp a given object stably, their joint angles are collected from the Cyberglove. We collected grasps from five subjects. The human dataset is not parsed because there are no metrics available which would tell us how well each of the subjects grasped the objects.

5.1.3 Fitting a Subspace to a Grasp Dataset

We hypothesized that grasps created by holding the objects in our set would exist in the pose space of the hand, but lie along the basis vectors of T . If this is true, then we can find a model of T by fitting a subspace to the grasps in \mathcal{G} . We wish for this model of T to explain enough of \mathcal{G} to enable teleoperation.

A model of T would provide us with the information necessary to create a teleoperation mapping for the hand. The model of the subspace consists of an origin and three N -dimensional orthogonal vectors, which describe the bases of the subspace. For a given hand, the origin pose of the subspace provides us with an origin pose \mathbf{o} for T , and the basis vectors provide us with a projection matrix A .

To find the model of T , we fit a subspace to the set of grasps \mathcal{G} using random sample consensus (RANSAC) [117, 118]. RANSAC is a consensus based algorithm used to find the model

underlying data with a large number of outliers. The basic algorithm of RANSAC is as follows:

- Generate a model hypothesis using random samples from the dataset. The number of samples selected should be the minimum number needed to define your model.
- Looking at all the points in the dataset, determine how well the hypothesis model explains/supports the data. If it is better than the best hypothesis to date, update the best model to your current hypothesis.

This process is repeated M times, where M is a number high enough to ensure that the probability of finding a model that is better than the current best model is sufficiently low. For our algorithm, we use $M = 2,000,000$. When RANSAC is parallelized, its runtime is 187 minutes on a computer with 24 CPUs.

Since our subspace is three dimensional, our model hypothesis consists of an origin grasp and three basis vectors. We also keep track of which of the three basis vectors corresponds to size, spread, and curl.

To generate a model hypothesis, we select random samples from the dataset. We first select an origin grasp. We specify that the origin must come from the set of grasps where the hand is holding Object 8 (\mathcal{G}_8). Preliminary tests with several hands showed that the performance for this origin was the highest. We hypothesize that this is because the constraints of the enveloping grasps are greater than the constraints of fingertip grasps. This gives the grasp planner (and the human) fewer options in how to grasp the objects, so the grasps have less variability.

Next, we select three additional grasps. We specify that each additional grasp must be selected from an object whose position in the subspace is identical to the origin object, except along a single basis vector. Since we have specified the origin, we randomly select one grasp from the set where the hand is holding Object 7 (\mathcal{G}_7), another grasp from \mathcal{G}_4 , and the last grasp from \mathcal{G}_6 . These objects correspond to the size, curl, and spread directions, respectively.

After we choose four random samples, we generate our model hypothesis. We subtract the three non-origin grasps from the origin and normalize the result to find the three basis vectors of

the subspace. We randomize the order of the three basis vectors, then orthogonalize these three vectors using Gram-Schmidt orthogonalization [119].

We determine the quality of our model hypothesis by how many objects from the object set can be grasped using the hypothesized subspace. To determine how well the hypothesis model explains the grasp data, we find the inliers in \mathcal{G} by calculating the distance from each grasp to the subspace defined by the hypothesis model. The distance d from each grasp \mathbf{g} to the hypothesis subspace model is found by projecting the grasp onto the subspace \mathbf{g}_{proj} and then finding the distance between the true grasp and the projected grasp :

$$\mathbf{P} = \omega_1^\top \cdot \omega_1 + \omega_2^\top \cdot \omega_2 + \omega_3^\top \cdot \omega_3 \quad (5.1)$$

$$\mathbf{g}_{proj} = \mathbf{P} \cdot (\mathbf{g} - \mathbf{o}) + \mathbf{o} \quad (5.2)$$

$$d = \|\mathbf{g} - \mathbf{g}_{proj}\| \quad (5.3)$$

where ω_1, ω_2 , and ω_3 are the basis vectors of the hypothesis model, and \mathbf{o} is the origin of the hypothesis model. Grasps which are closer than ξ (the final threshold used when we parsed the datasets) in Euclidean distance to the subspace are considered inliers:

$$\mathbf{g} = \begin{cases} \text{inlier} & \text{if } |d| < \xi \\ \text{not an inlier} & \text{otherwise} \end{cases} \quad (5.4)$$

Since we did not parse the human grasps, we simply set ξ for the human dataset as 0.1.

Many RANSAC algorithms use the total number of inliers to estimate how well the model explains the data; however, we wish all parts of our subspace to fit equally well. If the grasps for a few objects contain all the inliers and grasps for all other objects are far from the subspace, we do not consider this to be a sufficiently good model, even if it has the highest total number of inliers. We want our model to be able to grasp all the objects in our dataset. So, we use a tiered metric which considers the quality of fit in all parts of the subspace.

Our tiered metric has 4 components, ranked by importance:

1. Minimum number of inliers per object, over all the objects in our set. For example, if each object has at least one inlying grasp, then we consider that model to be better than a model where one or more of the objects have no inliers, because we can grasp all the objects in our object set.
2. Number of objects which have the minimum number of inliers. If the minimum number of inliers is one, if only one object has one inlier and all other objects have more than one inlier, this is preferable to all of the objects only having one inlier.
3. Total number of inliers across all grasps. The higher the number of inliers, the better the model.
4. Sum of the distances (error) between all the grasps and the subspace. The model with the lower error is better.

When two models tie in one or more of the metrics, the subsequent tier is used as a tiebreaker to determine the best model between two hypotheses.

Once we have generated and tested a sufficient number of hypotheses, the hypothesis model which explained the data the best, as defined by our metric, is considered to be the model of our subspace.

We perform one more processing step to find our final model. The same preliminary testing which indicated the best origin for the subspace model was Object 8 also showed that this was not the best origin when we combined the mappings for two hands into a complete teleoperation pipeline. For the final processing step, we choose a grasp from a different object to serve as the origin; empirically, we have found Object 1 to serve best in this role. For the robot hand, we move the origin to the grasp from $\mathcal{G}_{Object1}$ that is closest to the original subspace. For the human hand, we ask the teloperator to grasp a model of Object 1, and use the resulting pose as the subspace origin. Performing this additional step for every teleoperator also calibrates the mapping to the dimensions of their hand.

5.1.4 A Complete Teleoperation Algorithm

Projection Matrix: We use the three basis vectors of the subspace model found by RANSAC as the vectors which make up the projection matrix A . During RANSAC, we keep track of which of the three vectors corresponds to size, spread, and curl. We use this information to determine which vector is σ_H , α_H , and ϵ_H , respectively. We emphasize that, although our objects only fall on two locations along the curl basis vector of the subspace, once the mapping is created, we can move continuously along the curl basis vector.

Origin Pose: For a robot hand, the origin of the subspace model found by RANSAC becomes \mathbf{o} , the origin of the teleoperation mapping. For a human hand, we find the origin by asking the user to perform a calibration pose at the beginning of teleoperation. A standardized pose will not work for humans because user hand size varies.

Scaling Factor: To determine the scaling factors for our mapping, we require poses for the hand at the extremes of the subspace. We could select grasps from the dataset to determine these ranges, but it is faster to use a simple assumption and an iterative solution to find them.

For a robot, we assume that the hand will achieve its minimum and maximum value along each basis vector when the joints relevant to that basis are at some combination of their maximum and minimum values. We are given the maximum and minimum values for each joint from our robot model and the projection matrix tells us which joints are relevant to each subspace basis (if they are non-zero, they are relevant). For each subspace basis, we iterate through all the combinations of the relevant joints at their maximum and minimum values. These become the poses which show the hand's kinematic extrema.

For the human hand, we require the human to perform four calibration poses which will give us the ranges along each basis vector (see Figure 4.3). We require these poses because the differences in user hand size mean ranges which work for one person may not work for another.

For both human and robot hands, we project all the poses into the subspace, using the projection

matrix and the origin of our subspace model. We use the largest and the smallest value for each of the dimensions to calculate the range of that basis, and use Eq. 3.8 and Eq. 3.10 to find δ and δ^* .

Using the Mapping to Teleoperate: Once we have A , σ , δ , and δ^* for both hands, we can use Equation 3.13 to teleoperate the slave hand.

5.2 State-of-the-Art Comparisons

We selected two state-of-the-art teleoperation mappings with which to compare our subspace mappings. Below we describe them for the Schunk SDH and a two fingered gripper, the two hands we use in our experiments:

Fingertip Mapping We use fingertip mapping as a state-of-the-art comparison because it is one of the most common mapping methods and it is applicable to precision grasps, particularly with smaller objects [19]. The fingertip mapping was designed as follows: first, we found the cartesian positions of the thumb, index, and ring fingers of the human hand using the joint values from the Cyberglove and forward kinematics. The kinematic model we used for the human hand is described elsewhere [120]. We multiplied these positions by a scaling factor of 1.5, the ratio between an average human finger and the robot fingers. This ratio is 1.5 for both the Schunk SDH and the two fingered gripper. We assign each human finger a corresponding robot finger (for the two finger gripper, only the thumb and the index fingers are used). We translated the coordinates from the hand frame into the finger frame to find the desired robotic fingertip positions. Finally, inverse kinematics determined the joint angles which placed the robot fingertips at these positions [20]. In the previous chapter, fingertip mapping was controlled using a trakSTAR system. In this chapter, a dataglove is used as input for the fingertip mapping to keep the control more consistent for the users.

Joint Mapping We chose joint mapping as the second state-of-the-art comparison because of its common use in the field, and because we predicted that explicit control over individual joints

of the robotic fingers would be intuitive for novice users [25]. To implement joint mapping, we assigned each of the robot joints to a corresponding human hand joint. This mapping can be found in Table 5.2 for the Schunk SDH and Table 5.3 for the two fingered gripper. Once we received joint angles from the Cyberglove, we set the corresponding joints of the robot hand to the same values. Preliminary tests showed teleoperation is difficult if the robot thumb’s proximal joint maps to the human thumb’s metacarpophalangeal (MCP) joint. We therefore mapped the Schunk thumb’s proximal joint and the left proximal joint of the two finger gripper to the human thumb’s adductor. The joint mapping for the Schunk SDH is the same joint mapping we used in the previous chapter for the custom 3 fingered hand.

We chose not to compare our mapping with a pose or synergy mapping because none of these methods have shown themselves to be effective for a fully actuated, non-anthropomorphic hand, as we stated in Chapter 2.

5.3 Experiments

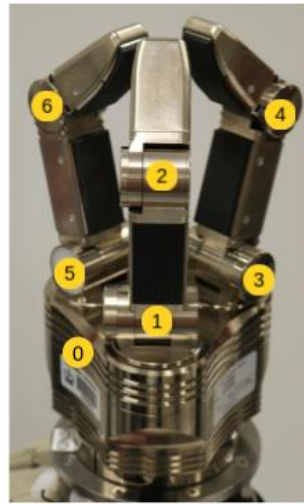
To validate that both the algorithmic and empirical mappings project to a subspace which is relevant to teleoperation, we asked ten novice users to complete manipulation tasks using both our mappings, and two state-of-the-art mappings as baselines. Half of the novices performed pick and place experiments with a Schunk SDH hand, and half of them performed in-hand manipulation tasks with a two fingered gripper.

For both the pick and place and the in-hand manipulation experiments, subjects were presented with the objects in the same order, and completed objects with one control before moving on to another control. We randomized the order in which the subjects used the controls. We also did not tell subjects how each of the control methods worked, but gave them two minutes to play with the hand when they were introduced to a new control method. The subjects gave their informed consent and the study was approved by the Columbia University IRB.

Below, we describe the mappings used in both experiments and the experiments themselves.

Table 5.2: Joint mapping from the Cyberglove to the Schunk SDH

Cyberglove Sensor		Robotic Hand Joints	
Joint Label	Name	Joint Label	Name
e	Index/Middle adduction	0	Finger 1 adduction
a	Thumb adduction	1	Thumb proximal flexion
b	Thumb distal flexion	2	Thumb distal flexion
c	Index proximal flexion	3	Finger 1 proximal flexion
d	Index medial flexion	4	Finger 1 distal flexion
f	Middle proximal flexion	5	Finger 2 proximal flexion
g	Middle medial flexion	6	Finger 2 distal flexion



Subspace Teleoperation Mappings: We generated teleoperation mappings for the human hand, the Schunk SDH and a two fingered gripper. For each hand, we created teleoperation mappings empirically and algorithmically, using the procedures outlined in Section 4.1 and Section 5.1, respectively. Figure 5.3 shows the resulting mappings for all three hands.

Pick and Place Experiments: We asked five novice users to complete pick and place tasks with our mappings and with state-of-the-art mappings.

We asked our novice users to pick and place the ten objects shown in Figure 5.4b using a Schunk SDH [121] mounted on a Sawyer arm [122]. In the previous chapter, we used a custom robotic hand with kinematics similar to the Schunk SDH, and in this chapter, we use the Schunk SDH hand. The Sawyer’s end effector position and orientation are controlled with a cartesian

Table 5.3: Joint mapping from the Cyberglove to the two finger gripper

Cyberglove Sensor		Robotic Hand Joints	
Joint Label	Name	Joint Label	Name
a	Thumb adduction	0	Finger 1 proximal flexion
b	Index distal flexion	1	Finger 1 distal flexion
c	Middle proximal flexion	2	Finger 2 proximal flexion
d	Middle medial flexion	3	Finger 2 distal flexion



controller (completely separate from the hand control) using a magnetic tracker (Ascension 3D Guidance trakSTAR™ [123]) placed on the back of the user's hand. Using the arm, the user could move and orient the hand however they chose. Figure 5.4a shows the experimental setup. Subjects were asked to don a Cyberglove [124], then pick up one object at a time and move the object across a line based on visual feedback.

In-Hand Manipulation Experiments: We asked the other five novice users to perform in-hand manipulation tasks with a two fingered gripper [125]. The gripper is stationary and placed on a table. An object was placed on the table between the distal links of the fingers in a precision grasp. We then asked the subjects to transition the object to a power grasp by moving the object closer to the palm and enveloping it with the robot fingers. For a transition to be counted as successful, the subject had to move an object so that it was in contact with both the proximal and distal links on one finger and at least one link on the other finger. Figure 5.5 shows the experimental setup and

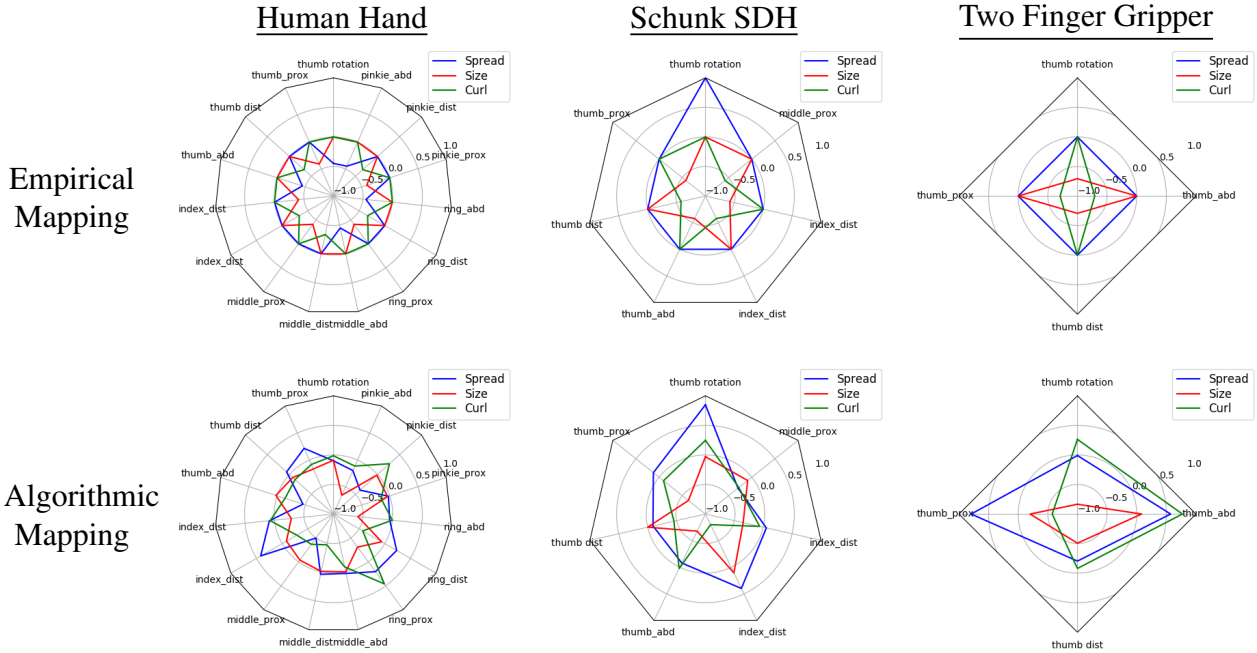
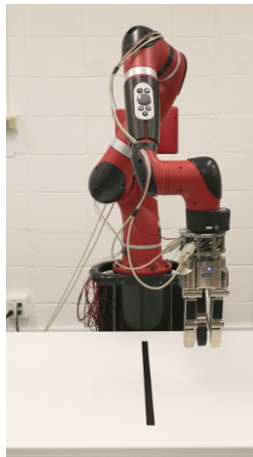


Figure 5.3: Teleoperation mappings generated for the human hand, Schunk SDH, and two finger gripper, both empirically and algorithmically. Each of the spokes represents a degree of freedom for the hand, and the blue (spread), red (size) and green (curl) values along those spokes indicate the values in the α_H , σ_H , and ϵ_H , respectively, at that degree of freedom.

the objects used for the in-hand manipulation experiments.

System Latency: To investigate the latency of our teleoperation system, we divide the problem into three parts: the first is the Cyberglove latency, which is five milliseconds (ms) [126]. The second part is computational latency, the interval between when the Cyberglove input arrives and when the system outputs a command for the robot. For all mappings presented in this study, this latency was measured to be less than 30 ms. The third part is hardware latency, the interval from when a robot command is sent to when the robot initiates the motion. We measured this value five times and took the average. The latency is 686 (± 172) ms and 50 (± 9) ms for the Schunk and the gripper, respectively. Thus, the total latency is approximately 735 ms for the pick-and-place system (dominated by the Schunk SDH firmware latency), and 85 ms for the in-hand manipulation system.



(a) Experimental setup with Sawyer arm, Schunk SDH hand, and table.



(b) Object set for our pick and place experiments.

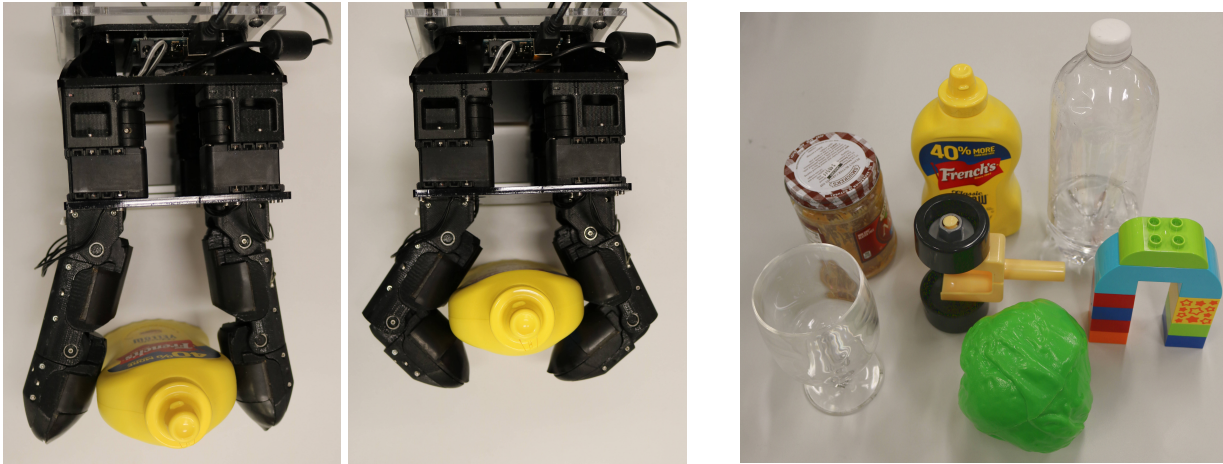
Figure 5.4: Pick and place experimental setup.

5.4 Results

For both experiments, we use three performance metrics: time to complete the task, how many tries the subject needed to complete the task and how many objects for which the task was completed. We describe these metrics below.

Our first metric was time to completion: we timed how long it took for the user to perform the task with each object. If the user did not complete the task in the given time limit (two minutes for pick and place experiments and one minute for in-hand manipulation experiments), they were considered to be unable to pick up the object and their final time was set to the respective time limit. We elected to shorten the time limit for the in-hand manipulation because there is no arm to move, and therefore the task should be completed faster.

Our second metric is the number of tries needed to complete a task. We define a ‘try’ as a completed task, an attempt where the user drops the object, an attempt where the user knocks over the object, or an attempt where the user knocks an object out of the range of the robot hand. In the last two scenarios, the object is reset by the experimenter. If the subject was unable to pick the object, we report the number of tries the user took before the time elapsed.



(a) Experimental setup with the two fingered gripper, and the table. (Left) Top view of experimental set-up with object in a precision grasp, (Right) top view of experimental set-up with object in a power grasp

(b) Object set for our in-hand manipulation experiments.

Figure 5.5: In-hand manipulation experimental setup.

Our final metric is how many objects for which the task was completed: for each mapping we counted the number of objects with which the user was able to successfully complete the assigned task.

Pick and Place Results: We report our results as the average across all subjects. We report averages for all objects, for large objects (the box, ball, wire spool, and water bottle), for small objects (the peg, valve, and marbles), and for irregular objects (the drill, screwdriver, and lego stack). We classify the drill, screwdriver, and legos as irregular objects because their width to length ratios and their irregular shapes allow users to pick up the objects with a wide variety of grasps. Users tended to pick up the large objects and the small objects with consistent grasps.

We report the average time to pick and place across all subjects in Table 5.4a. Across all subjects and all objects, novices using the fingertip mapping took 3 times longer than when using the empirical subspace mapping, and 1.7 times longer than when using the algorithmic subspace mapping. Similarly, joint mapping took 2.5 times longer than the empirical subspace mapping and 1.4 times longer than the algorithmic subspace mapping.

Table 5.4: Pick and place experiment results

Objects	Fingertip	Joint	Empirical	Algorithmic
All	67.8 ± 6.1	59.8 ± 6.2	25.9 ± 2.7	42.5 ± 4.7
Small	85.6 ± 10.9	92.6 ± 10.2	38.3 ± 7.3	51.3 ± 10.3
Large	49.1 ± 9.7	38.4 ± 8.2	18.7 ± 2.1	30.0 ± 5.3
Irregular	75.1 ± 10.8	55.4 ± 11.2	22.9 ± 3.3	50.2 ± 9.2

(a) AVERAGE TIME TO PICK AND PLACE (SECONDS)

Objects	Fingertip	Joint	Empirical	Algorithmic
All	1.9 ± 0.2	1.9 ± 0.2	1.2 ± 0.1	1.7 ± 0.2
Small	1.9 ± 0.4	2.4 ± 0.4	1.3 ± 0.2	2.3 ± 0.4
Large	1.6 ± 0.3	1.6 ± 0.2	1.0 ± 0.0	1.4 ± 0.2
Irregular	2.5 ± 0.4	1.8 ± 0.4	1.2 ± 0.1	1.5 ± 0.4

(b) AVERAGE TRIES TO PICK AND PLACE

Objects	Fingertip	Joint	Empirical	Algorithmic
All	6.8 ± 1.0	7.2 ± 0.7	10.0 ± 0.0	9.0 ± 0.5
Small	1.4 ± 0.5	1.2 ± 0.5	3.0 ± 0.0	2.4 ± 0.4
Large	3.4 ± 0.4	3.8 ± 0.2	4.0 ± 0.0	4.0 ± 0.0
Irregular	2.0 ± 0.5	2.2 ± 0.4	3.0 ± 0.0	2.6 ± 0.2

(c) AVERAGE NUMBER OF OBJECTS PICKED

For the four combinations of objects we look at (all objects, small, large, and irregular), the empirical subspace mapping took the least amount of time, with the algorithmic subspace mapping coming in second in every case. The algorithmic subspace mapping was at least 1.6 times slower than the empirical subspace mapping for all of these object combinations. However, in turn, the state-of-the-art mappings were at least 1.3 times slower than the algorithmic subspace mapping.

We report the average number of tries in Table 5.4b. In all object combinations, users were able to pick and place objects with the fewest amount of tries using the empirical subspace mapping. The algorithmic subspace mapping came in second in all cases except for with the small objects, where fingertip mapping came in second.

Finally, we report the average number of objects the users were able to pick up with each of the mappings in Table 5.4c. For all the objects, the maximum number of objects that can be picked is 10, for the small and irregular objects, the maximum is three, and for the large objects, the maximum is four.

Table 5.5: In-hand manipulation experiment results

Objects	Fingertip	Joint	Empirical	Algorithmic
All	16.6 ± 2.3	8.8 ± 1.7	8.5 ± 1.7	13.1 ± 2.0
Circular	15.4 ± 2.6	7.5 ± 1.5	5.6 ± 0.9	13.6 ± 2.6
Irregular	17.5 ± 3.6	9.7 ± 2.8	10.7 ± 2.9	12.8 ± 2.9

(a) AVERAGE TIME TO TRANSITION FROM A PRECISION GRASP TO A POWER GRASP (SECONDS)

Objects	Fingertip	Joint	Empirical	Algorithmic
All	1.6 ± 0.2	1.2 ± 0.1	1.1 ± 0.0	1.3 ± 0.1
Circular	1.1 ± 0.1	1.1 ± 0.1	1.0 ± 0.0	1.1 ± 0.1
Irregular	2.1 ± 0.3	1.3 ± 0.2	1.1 ± 0.1	1.5 ± 0.2

(b) AVERAGE TRIES TO TRANSITION

Objects	Fingertip	Joint	Empirical	Algorithmic
All	6.6 ± 0.4	7.0 ± 0.0	6.8 ± 0.2	7.0 ± 0.0
Circular	3.0 ± 0.0	4.0 ± 0.0	3.0 ± 0.0	4.0 ± 0.0
Irregular	3.6 ± 0.4	3.0 ± 0.0	3.8 ± 0.2	3.0 ± 0.0

(c) AVERAGE NUMBER OF OBJECTS MANIPULATED

The empirical subspace mapping allowed every novice to pick up every object. With the algorithmic subspace mapping, novices could pick up most objects, and with the state-of-the-art mappings, novices picked up the majority of objects, but still fewer than either of the subspace mapping methods.

In-Hand Manipulation Results: We report our results as the average across all subjects. We report averages for all objects, for circular objects (the bottle, peanut butter container, and goblet), and for irregularly shaped objects (the wheels, legos, lettuce, and mustard).

We report the average time to perform the in-hand manipulation task across all subjects in Table 5.5a. Across all subjects and all objects, novices using the fingertip mapping took 2 times longer than when using the empirical subspace mapping, and 1.3 times longer than when using the algorithmic subspace mapping. Joint mapping performed about the same as the empirical subspace mapping and was 1.5 times faster than the algorithmic subspace mapping.

For the three combinations of objects (all, circular, and irregular), manipulation with the empirical subspace mapping took the least amount of time for all objects and the circular objects, with

joint mapping taking the least amount of time for the irregular objects. In all cases, the algorithmic subspace mapping was third and fingertip mapping took the longest.

We report the average number of tries subjects took to manipulate the objects in Table 5.5b. In all object combinations, users were able to transition the objects with the fewest amount of tries using the empirical subspace mapping.

Finally, we report the average number of objects the users were able to manipulate with each of the mappings in Table 5.5c. For all objects, the maximum number of objects that can be manipulated is 7, for the circular objects, the maximum is three, and for the irregular objects, the maximum is four.

The joint and algorithmic subspace mappings allowed every novice to manipulate every object. For the empirical subspace mapping, one subject was not able to transition one object, and for the fingertip mapping, one subject was not able to manipulate two objects.

Discussion: We begin by discussing the teleoperation mappings generated algorithmically and empirically. In both cases, novice users were able to complete two different types of manipulation tasks using two different non-anthropomorphic robot hands. This shows that the mappings created with both methods rely on a subspace which is relevant to teleoperation and which can encompass the range of motion necessary to manipulate a variety of objects in different ways. Similarly, these experiments show that the subspace is relevant for more than one hand.

The algorithmic subspace mapping, in particular, not only shows that the subspace we propose is relevant to teleoperation, but that the benefit of using such a subspace does not derive exclusively from the human intuition used to create the mapping. Since this mapping is created automatically, without kinematic-specific intuition from the mapping creator, and can still enable teleoperation, we conclude that T is a concept that has value even when there is no human intelligence ‘built into’ the mapping. That being said, we do note that the empirical subspace mapping, which is defined with the benefit of human intuition, outperforms the algorithmic subspace mapping. The use of human intuition to define the basis vectors, while not exclusively defining the value of the

subspace, can make it a more powerful and intuitive control.

We would like to emphasize that we have not designed the algorithmic mapping to replace the empirical mapping. In some cases, an empirical mapping can take significantly less time to create and also outperform the algorithmic mapping. The purpose of the algorithmic mapping is to show that the concept of a teleoperation subspace is relevant and useful for multiple hands, and that this relevance does not come exclusively from the human intuition built into the subspace via an empirical mapping. We also envision the algorithmic mapping to be useful for continuum robots, which do not have traditional finger-like structures, making the mapping difficult for a human to create, but we leave this to future work.

Both experiments showed that the empirical and algorithmic subspace mappings were as intuitive as or more intuitive than the state-of-the-art mappings. We measure intuitiveness as the combination of our three metrics: we hypothesize that controls which allow the user to manipulate more objects in less time, with fewer tries are more intuitive. We note that the measure of which method is preferable is a trade-off between intuitiveness for the teleoperator and intuitiveness for the person who must generate the teleoperation mapping. For our real-time experiments, the three metrics we have selected only measure intuitiveness for the teleoperator.

For the pick and place experiments, the empirical subspace mapping was the most intuitive control for novices, and the algorithmic subspace mapping was the second most intuitive control. In all metrics, the empirical and algorithmic subspace mappings outperformed the state-of-the-art. The empirical subspace mapping provides the greatest advantage for small objects, but still has a significant edge for all other object combinations. It is worth noting that the standard error we report for all the metrics is also lowest for the empirical subspace mapping. We hypothesize that this means the novices were able to use the empirical subspace mapping more consistently than the other controls.

For the in-hand manipulation experiments, our empirical subspace mapping proved the most effective in terms of time to perform the experiments, followed by joint mapping and the algorithmic subspace mapping. A similar result was observed for the average number of tries required to suc-

ceed; for total objects manipulated, all three of these mappings showed similar performance, with joint mapping and algorithmic subspace mapping having a very slight advantage. This ranking is thus less definitive than for the pick and place experiments because different mappings performed better for different metrics.

We note that the in-hand manipulation experiments lend themselves particularly well to joint mapping, which allows users to individuate the robot digits, an advantage when performing in-hand manipulation, and something which our subspace mappings do not allow. This individuation provides a particular advantage for irregular objects and is likely why joint mapping outperformed the empirical subspace mapping in time to completion for that particular object category.

These experiments show that our two subspace mapping methods can generalize across different hands and different manipulation tasks.

5.5 Summary

In this chapter, we expand upon an intuitive, low dimensional mapping between the pose spaces of the human hand and non-anthropomorphic robot hands. We propose an algorithmic method to generate the mapping, which is completely automatic. This automated process is made possible by defining the subspace independently of hand kinematics, using objects to define hand motions that span the desired subspace.

We expand our validation of both the empirical and algorithmic subspace mappings with real-time teleoperation experiments with novice users on two kinematically different robotic hands. We found that, for pick and place experiments, our empirical subspace mapping was most intuitive for users, with the algorithmic subspace mapping still performing better than state-of-the-art alternatives. For the in-hand manipulation experiments, we found that our empirical subspace mapping performed as well as joint mapping, one of the state-of-the-art methods, and better than fingertip mapping, the other baseline we employed. For the in-hand manipulation experiments, the algorithmic subspace mapping was generally less intuitive for novices than joint mapping, but more intuitive than fingertip mapping.

The real-time teleoperation experiments show that the subspace we propose is relevant to teleoperation for multiple hands with distinct kinematics and for different manipulation tasks. This is the first time, to our knowledge, that a teleoperation mapping generated without requiring a user's understanding of hand-specific kinematics has been shown to be intuitive for novices for real-time teleoperation. The fact that the algorithmic mapping can enable teleoperation shows that the subspace encodes useful information for teleoperation that does not rely exclusively on human intuition.

The future of this work could take a number of directions. Our experiments in this paper show that the subspace is relevant for at least three different hands, and we would like to continue to show that is relevant for other hands with different kinematic configurations. It would be interesting to add more dimensions to the teleoperation subspace to see if this increases the dexterity of the hand while remaining intuitive for the user. We would like to show that the subspace is useful for more complex tasks, like assembling and disassembling machinery.

In the previous two chapters, we show that a subspace which is low dimensional can encode enough information to allow teleoperation. In the next chapter, we show that we can leverage the same low-dimensional teleoperation subspace to teleoperate robotic hands using other control inputs, like electromyography (EMG).

Chapter 6: Using the Teleoperation Subspace with an EMG Input

In the previous two chapters, we introduced a teleoperation subspace and two methods to define the mapping between joint space and the subspace. Up until now, we have used datagloves as our input devices. Datagloves and vision are common in the field of teleoperation because they are robust. However, datagloves can interfere with a hand's fine manipulation or tactile sensing abilities, and are easily damaged. Vision-based methods often require environments which are well-lit and which do not have many obstacles that occlude the hand. We would like to additionally have a control input that collects information about the teleoperator's hand pose while being *wearable, inexpensive, and unobtrusive*.

One such input is forearm electromyography (EMG). EMG is low profile, and its position on the forearm, rather than the hand, makes it less susceptible to damage, while leaving the hand completely unencumbered. Inexpensive EMG armbands are quickly becoming commodity products.

In this chapter, we use the teleoperation subspace introduced in Chapter 3 as an intermediary between the EMG signal of the human forearm and the robot pose space. The key to our projection between EMG and the teleoperation subspace is our use of a hybrid model, which is a combination of discrete and continuous predictors, along with the teleoperation subspace, which is independent of training. These elements allow users to control a robot using natural hand movements, making the control intuitive.

To our knowledge, we are the first to demonstrate teleoperation of a non-anthropomorphic, multi-degree-of-freedom (multi-DOF) robot hand using forearm EMG as a control input. Other state-of-the-art teleoperation methods have used forearm EMG as a control input but have not demonstrated control of non-anthropomorphic, multi-DOF robot hands. We modified several state-of-the-art teleoperation controls as necessary to make them applicable to these kinds of hands and compare them to our method. Experiments with novice teleoperators prove that our method can

grasp a wide variety of objects, and do so faster than state-of-the-art methods.

6.1 Mapping EMG into Teleoperation Subspace

EMG signals from the forearm are noisy and many of the muscles which control finger movement lie deep under the skin. As a result, it is difficult to project from EMG signals to a high dimensional joint space of a fully actuated robot. It is much more practical to project EMG to a lower dimensional space, and then project from that subspace to the pose space of a robot hand.

We will project EMG into the teleoperation subspace described in Chapter 3 and validated in Chapters 4.1 and 5.1. We have already shown that this subspace is relevant to teleoperation, and it is low dimensional, having three basis vectors. Again, we encourage the reader to review Section 3.1 and Section 3.2 for an overview of the basis vectors of T and the variables required to map from T to pose space, respectively.

In this chapter, we will bypass the pose space of the human hand. We map directly from EMG into the teleoperation subspace, and then we map from the teleoperation subspace to the pose space of the robot hand. We note that to map between T to the slave pose space, in this chapter, we only use empirically generated mappings.

Unlike the mapping between pose space and teleoperation subspace, where we can project both into and out of the subspace, mapping EMG into the teleoperation subspace is a one-directional operation. We only ever wish to project EMG into the subspace, and never the reverse.

We map EMG signals into T using a hybrid method that combines regression (continuous) and classification (discrete). Although we would ideally like to use a regressor to predict the value along all three axes of T , because of the noise inherent in EMG and the distal location of the intrinsic muscles which control finger spread [127], regression does not always provide an accurate estimation of hand pose in T . We therefore offer the user the option of using two discrete hand signals to move the robot in specific ways. The classifier distinguishes between these two gestures and normal motion of the hand (when the user is not making a gesture). Therefore, classifier has three classes:

- Gesture 1 - Finger spread: the user spreads their fingers apart, making the robot fingers spread as well;
- Gesture 2 - Isometric contraction: the user contracts their muscles and the robot hand closes;
- Normal motion: the user moves their hand normally, and regression determines the position of the robot.

The regressor is flexible and intuitive, providing a continuous prediction for the size σ and curl ϵ bases of T , for which EMG gives a clear signal while the user makes natural grasping motions. The classifier provides stability when the EMG signals are too noisy for regression; it uses discrete signals to control finger spread and to maintain stable grasps, cases where EMG tends to be less reliable. As we show experimentally, the combination of continuous and discrete models enables robust and stable teleoperation.

EMG collection and processing: We collect forearm EMG data with the Myo from Thalmic Labs, an armband with 8 EMG sensors. We receive EMG signals $\mathbf{x}_t \in \mathbb{R}^8$ from the forearm at time T and use a lowpass filter with a window size of 0.5 seconds to remove noise from the EMG. The filter has a sampling frequency of 5kHz and a cutoff of 200Hz. We refer to the filtered EMG signals as $\hat{\mathbf{x}}$.

Our goal at every time step is to find ψ_t , the hand's position in T , given $\hat{\mathbf{x}}_t$. To do this, we pass $\hat{\mathbf{x}}_t$ through a classifier and then, depending on the output of the classifier, the signals are either passed to a regressor, or the pose from the previous time step is altered in a predefined way. We explain our method below and illustrate it in Figure 6.1.

Classification and Regression: When the classifier identifies that the user is performing one of the two gestures we chose as having meaning for our control method (spreading their fingers, or performing an isometric contraction), our model modifies the pose in T in a predetermined manner.

When classifier predicts that the teleoperator is spreading their fingers, we freeze the values of σ and ϵ and begin to change the spread value along α . The pose in T begins with α at its

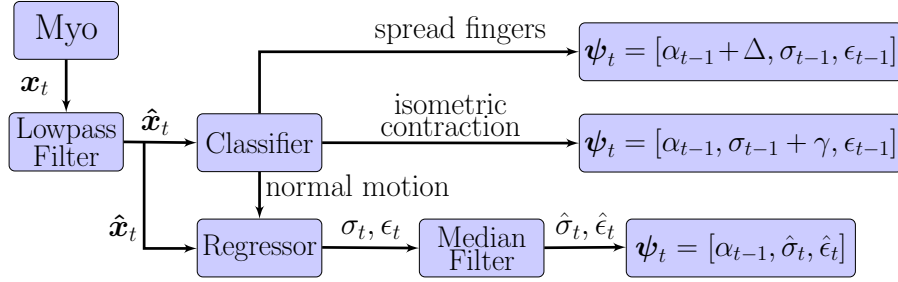


Figure 6.1: Control scheme for the proposed EMG teleoperation.

minimum value. When the classifier identifies that the user has spread their fingers, the value of α begins to change at a predetermined rate Δ until the user stops spreading. α remains at that value until the user spreads their fingers again. Δ begins as a positive rate, causing the fingers to spread further apart. When the fingers of the robot hand reach their maximum spread value, the sign of Δ changes, and the fingers begin to move back towards each other.

The classifier also uses isometric contractions as a discrete signal to help the robot hand close, in order to create more stable grasps. Teleoperators have a natural tendency to perform isometric contractions to ensure the slave robot maintains its grasp. If the model has no special case for this, the muscle contractions can cause a regression model to behave in unexpected ways. When the classifier identifies an isometric contraction, we freeze the values of ϵ and α , the σ value output by the regressor is ignored, and σ starts to increase at a predetermined rate γ . The value of σ increases until the fingers stall or until the user performs another isometric contraction. If the user performs a second isometric contraction, the regressor resumes predicting the value of σ and ϵ , and the classifier resumes controlling the value of α .

Δ and γ are set by the experimenter, depending on how fast we want the fingers to spread and the hand to close, respectively.

If the classifier predicts that the teleoperator is moving normally, i.e. not spreading their fingers or performing isometric contractions, the filtered EMG signals $\hat{\mathbf{x}}_t$ are passed to a regressor. α_t remains equal to α_{t-1} and the regressor outputs new values for σ_t and ϵ_t .

Our prediction still has some noise after the regression, so, to keep the robot fingers steady, we pass the output of the regressor through a median filter with a window size of 200ms to find our

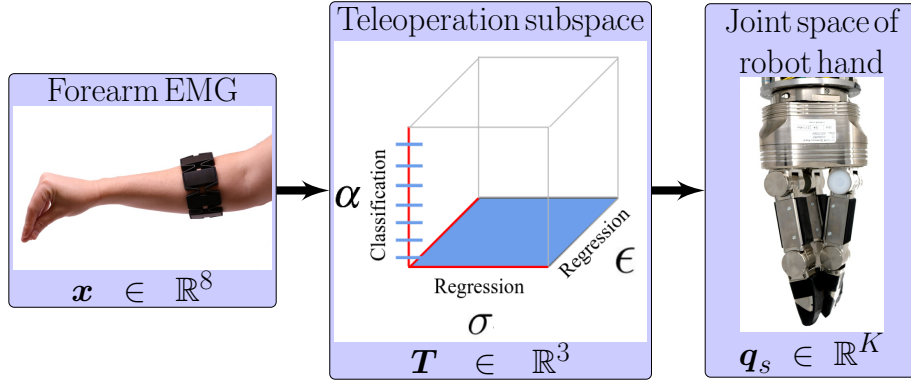


Figure 6.2: Steps to enable real time teleoperation using forearm EMG as a control input.

final $\hat{\sigma}_t$ and $\hat{\epsilon}_t$ in T .

A Complete Teleoperation Algorithm To teleoperate using T with an EMG input, there are two steps (Figure 6.2):

1. Given forearm EMG signals from the master hand, find the equivalent pose ψ in teleoperation subspace T using the projection method outlined in Section 6.1;
2. Given ψ computed above, find the joint values of the slave hand using Equation 3.12, and move the slave hand to these values.

6.2 State-of-the-Art Comparisons

We have identified four unique approaches to EMG teleoperation we wish to evaluate experimentally. The first method is our method, which is described above. The next three methods are state-of-the-art EMG teleoperation controls, created for underactuated or anthropomorphic hands. We have modified them as necessary to control non-anthropomorphic, fully actuated hands. The methods are described below and diagrams of their control structure are shown in Figure 6.3.

Method 1: Regression and Classification using Teleoperation Subspace This is our method as described in Section 6.1. It combines a classifier and a regressor to project EMG signals to T and teleoperation subspace mapping to project from T to the robot pose space.

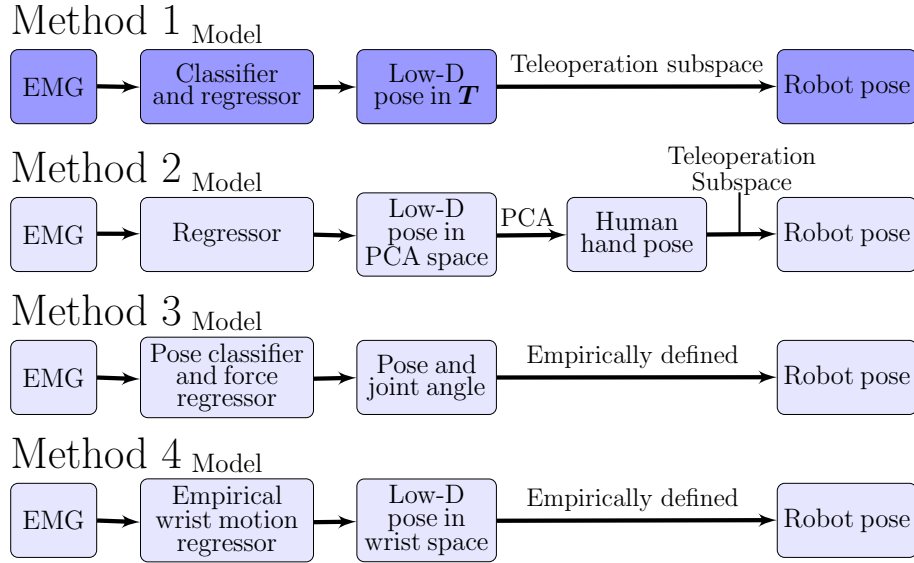


Figure 6.3: Comparison of control methods which map between forearm EMG and a robot pose space.

Method 2: Regression using Low-Dimensional Subspace found with PCA For this method, a PCA-based subspace is found by performing PCA on joint angles collected during training. A regressor takes as input filtered forearm EMG signals and outputs a pose in the PCA-based subspace. The predicted poses are projected into human pose space using the PCA components and then mapped into a robotic pose space using teleoperation subspace mapping.

This PCA-based strategy is used in the literature both for underactuated hands [66], and hand/arm systems [68]. We modified it to work with non-anthropomorphic hands by mapping from the human pose space to the robot pose space with teleoperation subspace mapping.

The basis vectors of the PCA subspace change depending on the training data, so there is no guarantee that the basis vectors will correspond to a given hand motion; therefore, the model cannot combine a classifier and a regressor.

Method 3: Pose Classification and Force Regression This method, created by Yoshikawa et al. [61], uses a gesture classifier and a force regressor to enable teleoperation.

For the classifier, we selected different hand poses than those presented in the original paper, which mostly classified wrist positions, with only an open and close pose for the hand. We selected

three hand poses which represent basic grasp types (power, precision, and pinch). These poses are intuitive because they are distinct from each other and they have a clear corresponding pose in robot pose space.

The force regression is an empirically defined method based on maximum voluntary contraction (MVC) and minimum voluntary contraction for each gesture. The regressor outputs a single joint angle θ , which then must be used to determine hand motion of the robot empirically. We have created our own empirical mapping between θ and the pose of a non-anthropomorphic robot hand. The greater the isometric contraction, the more the robot hand opens. Low force closes the robot hand because isometric contractions are difficult to maintain over a long period of time.

Method 4: Empirically Defined Regression from EMG to a Low-Dimensional Space using Wrist Motion This method uses an empirically defined regressor created by Matrone et al. [67]. The regressor projects EMG data into a 2D space. The basis vectors of the space, $C1$ and $C2$, have been shown to correspond well to wrist flexion/extension and wrist abduction/adduction, respectively. The pose in the 2D space is then mapped to robot hand position.

In the original work [67], EMG sensors are placed on two agonistic/antagonistic muscle pairs, one for wrist extension/flexion and one for wrist abduction/adduction. In this work, we place sensors on the same wrist extension/flexion muscle pair, and wrist adduction muscle. However, we are constrained by using an EMG armband, so we cannot place the wrist abduction sensor on the extensor pollicis longus, which is further down the forearm. Instead we place the wrist abduction EMG sensor on the extensor carpi radialis longus.

6.3 Experiments

In this section, we evaluate all of the above methods for complete teleoperation with novice users. Since many of them rely on classification or regression, we begin by describing the model training and the selection of model algorithms.

Training The training process for each of the methods is as follows:

Method 1: The user generates a training dataset by moving their hand for two minutes while wearing the Myo armband, which provides forearm EMG signals, and a Cyberglove, which provides ground truth joint angles.

We instructed users to move at a moderate pace and to explore the hand’s full range of motion. We also prompted users to perform gestures at 30 second intervals to provide the classifier with training information.

The control was trained as follows: we passed EMG signals through a bandpass filter and projected joint angles into \mathbf{T} to provide a ground truth values for σ and ϵ . We project the joint angles into \mathbf{T} using Eq. 3.11. We trained the regressor on all data where the user was not performing a gesture. The regressor takes as input the filtered EMG signal and outputs values for σ and ϵ . We trained the classifier on all training data. It takes as input the filtered EMG signal and outputs a predicted gesture (or a prediction of no gesture, i.e. normal movement).

Kernel ridge regressors (KRRs) [57], non-negative factorization (NMF) combined with linear regression (LR) [66] and latent space models (LS) [10] have all been used for EMG controls. We trained three regressors to determine which is best suited to the motions and teleoperation subspace which we use here: using the filtered EMG and the projected values in \mathbf{T} , we trained a KRR and a NMF+LS model. For the latent space model, like the original work [10], we performed PCA on the EMG signals and projected them into a low-dimensional space. We used the projected EMG data and the poses in \mathbf{T} to train an LS model.

We tested the regressors on three sets of data generated in the same way as the training data. Table 6.1, shows the normalized root mean squared error (nRMSE) of each regressor as a percentage averaged over the three testing datasets.

We chose to use a KRR for the rest of our experiments because its nRMSE averaged across σ and ϵ was lower than the other two regressors and because KRRs have the ability to update their model with future training data, which can make them robust to drift and donning/doffing, an important issue with EMG controllers [64]. We do not address model updating here, but we plan to explore this in the future.

Table 6.1: Normalized root mean square error of trained regressors and global accuracy of trained classifiers

Regressor	Method 1		Method 2		Classifier	Method 1	Method 3
	σ	ϵ	Basis 1	Basis 2			
KRR	3.7%	15.6%	4.2%	1.9%	SVM	90.4%	91.8%
NMF+LR	23.4%	17.3%	4.8%	3.0%	RF	91.6%	94.1%
LS	3.5%	19.6%	3.3%	2.6%			

The KRR has a radial basis function (RBF) kernel and its alpha and gamma parameters are determined by cross-validation on the data collected to train the model.

We also train two classifiers - a support vector machine (SVM) and a random forest (RF) classifier that can identify the gestures relevant for Method 1 (normal motion, spread, isometric contractions). These have both shown high accuracies when identifying hand gestures [128] [61].

The classifiers are trained by the user performing a predefined series of gestures while giving the system ground truth labels for the gesture. We tested each classifier on three testing datasets. Table 6.1 shows the average accuracy of both classifiers across the three testing datasets. We chose a RF classifier over a SVM because its accuracy is higher and the time it takes to fit a RF is faster.

Method 2: Generating the training dataset for this method is the same process as the training for Method 1, except the user does not perform gestures intermittently.

Once the dataset is gathered, we perform PCA on the joint angles from the Cyberglove and keep the eigenvectors which explain 90% of the variance in the data. The joint angles are projected to the PCA-based space with the eigenvectors.

The EMG is put through the same bandpass filter used in Method 1. The model for this method takes filtered EMG as an input and outputs a pose in the PCA-based space.

As for Method 1, we wished to determine which of the three regression models presented in the literature work best with a PCA-based subspace. We trained KRR, NMF+LR and LS models in the same way as for Method 1, but using low-dimensional poses in the PCA subspace as ground truth.

The Method 2 regressors were also tested on three datasets (Table 6.1). We again chose to use the KRR because of the possibility of updating the model with future data.

Method 3: Method 3 uses a classifier to determine hand pose and an empirically defined force regressor to determine a joint angle θ . The predicted hand pose and θ map to robot hand position. We trained the classifier by asking the user to perform a series of gestures. During these gestures, the user performed several isometric contractions and then relaxed.

The user also had to perform six calibration gestures to train the empirically defined force regressor. They performed each of the three hand poses while they were relaxed, and while they were performing isometric contractions.

As for Method 1, we wished to determine if an SVM or an RF classifier would better distinguish between the classes relevant for Method 3 (pinch, spread, parallel grasp). We trained two more classifiers and tested in the same way as for Method 1. The average accuracy of the two classifiers are shown in Table 6.1. We again chose to use a RF classifier because of its higher accuracy and shorter fit time.

Method 4: Method 4 uses an empirically defined regression which maps EMG signals to a two dimensional space whose basis vectors have been shown to correspond well with certain wrist motions [67]. Because the regression is empirically defined, the training only requires four calibration poses - flexing, extending, abducting and adducting the wrist to provide the MVC for each of the gestures.

In the original work that outlined this method, medical grade sensors were placed by trained experimenters on specific muscle pairs. Our EMG armband is inexpensive and easy to don even by a novice, but also provides constraints: lower quality signal, and the need to use muscles that are all at similar height on the forearm. Within these constraints, our experiments showed that the basis vector $C2$ no longer corresponds to the expected wrist motion (abduction/adduction).

We asked a user to perform three different wrist gestures and projected the collected EMG data into the 2D subspace. To see if the basis vectors corresponded to the expected wrist motions,

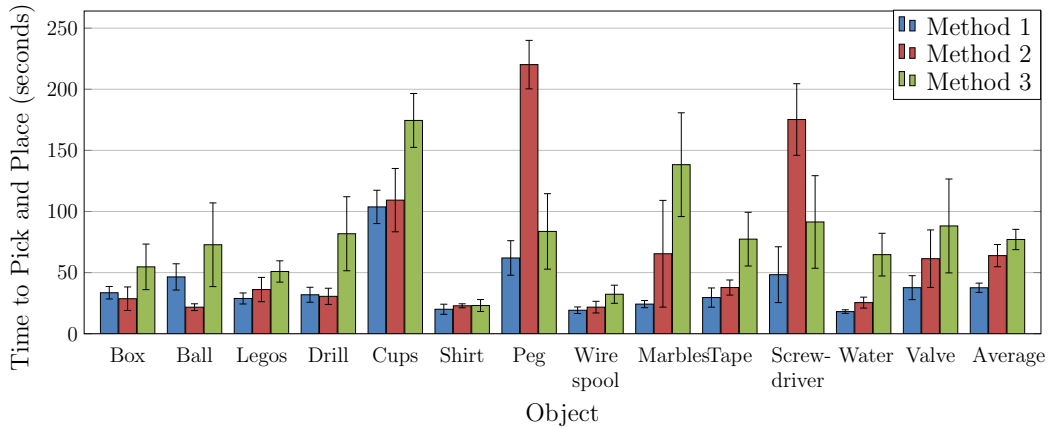
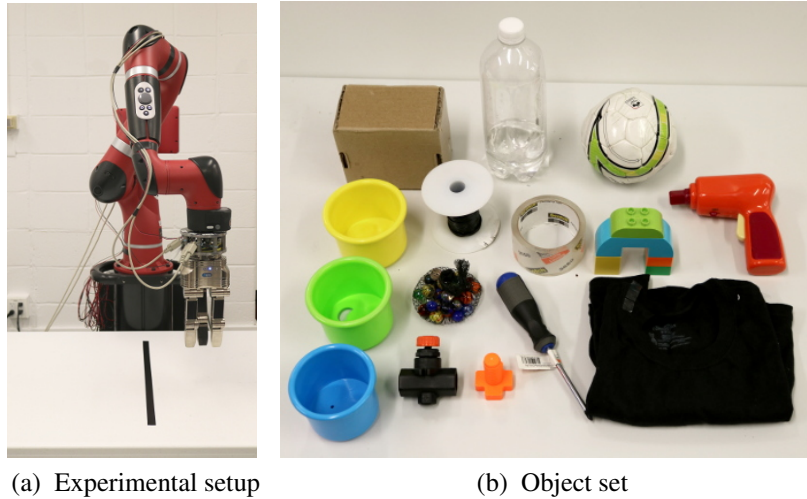
we compared the variance along the basis vectors. The variance along $C1$ is 0.0 when the user holds their wrist still and increases 66-fold when the user performs wrist flexion/extension. On the other hand, the variance along $C2$ is 0.0 when the user holds their wrist still, 0.02 when the user abducts/adducts, and 0.02 when the user flexes/extends. We conclude that, within the constraints of a commercial EMG armband, this method is ineffective because $C2$ does not correspond to the expected wrist motion. We therefore do not use this control method in our experiments.

Experimental Setup: To show that EMG controls enable effective teleoperation of non-anthropomorphic hands, we asked five novices to pick and place a variety of objects using Methods 1, 2 and 3 above (we excluded Method 4 since, as discussed above, it proved not applicable with a commercial EMG armband). The subjects gave their informed consent and the study was approved by the Columbia University IRB.

The hand used for teleoperation was the Schunk SDH hand attached to a Sawyer Robot arm (Figure 6.4a). The Sawyer's end effector position and orientation are controlled with a simple cartesian controller (completely separate from the hand control) using a magnetic tracker (Ascension 3D Guidance trakSTAR™) placed on the back of the user's hand. Users teleoperated based on visual feedback.

The users were asked to complete 13 pick and place tasks (the object set is shown in Figure 6.4b). The user was either asked to pick up the item and move it across a line or, in the case of the three cups, stack the items. The user completed the pick and places for all objects with one control method before moving to the next control. The order in which each subject used the control methods was randomized.

We timed how long it took for the users complete each pick and place. If a user did not complete the task in four minutes, they were considered to be unable to pick up the object and their final time was set to four minutes.



(c) Time to pick and place each object for pick and place experiments

Figure 6.4: Pick and place experiments - setup and results.

6.4 Results

Across all subjects and all objects, with Method 2, it took novices 1.70 times longer to complete a pick and place than when they were using Method 1. Method 3 took 2.05 times longer than Method 1. Figure 6.4c shows the average time to pick and place and Table 6.2 summarizes the average statistics for each method calculated across all subjects and all objects.

Let us consider the pick and place times more granularly. We hypothesized that Method 1 would help most for small, circular objects. If we only consider the valve, the marbles, the orange peg, and the screwdriver, the average time to pick and place using Method 2 was 3.43 times longer

Table 6.2: Pick and place statistics

Method	Time to completion (s)	Tries per object	Successful picks (max 13)
1	37.6 ± 3.8	1.2 ± 0.1	13 ± 0.0
2	63.9 ± 9.1	1.8 ± 0.2	11.6 ± 0.5
3	77.1 ± 8.3	1.6 ± 0.2	12.2 ± 0.6

than Method 1. With Method 3, it was 2.33 times longer than with Method 1. If we consider all objects but the four mentioned above (the larger objects), the average time to pick and place with Method 2 was 1.07 times longer than Method 1 and with Method 3 was 1.95 times longer than Method 1.

Method 1 was the only control method which allowed all the novice users to pick up all 13 objects. With Method 2, the subjects were able to pick up, on average, 11.6 objects, and with Method 3, subjects averaged 12.2 objects.

We counted the number of tries it took for the user to pick up each object. We define a ‘try’ as a completed pick and place task, an attempt where the user drops the object or an attempt where the user knocks over the object. With Method 1, the average number of tries across all subjects and all objects was 1.2. With Method 2, the average number of tries was 1.8 and with Method 3, it was 1.6.

Discussion: Our experiments show that, by leveraging the teleoperation subspace proposed in this thesis, EMG-based teleoperation is possible with a fully actuated, non-anthropomorphic hand. When compared with two state-of-the-art EMG teleoperation methods, novices were able to use Method 1 (ours) to pick up a wider variety of objects, faster and with fewer mistakes, than they could with Methods 2 and 3.

Our method provides the greatest advantage when attempting to pick and place smaller objects. With larger objects, Method 1 and Method 2 enable teleoperation at about the same speed. We hypothesize that our hybrid method provides this advantage because it is difficult for the user to both close their fingers and also spread them. Our hybrid model provides explicit control over the

spread and close motions of the robot hand, thus avoiding awkward hand positions and making stable grasping easier. Our hybrid method works with the teleoperation subspace because the consistent basis vectors allow the use of both discrete and continuous models.

We notice in Figure 6.4c that, for some of the objects, the standard error is high, particularly when the average time to pick and place was over 50 s. For the smaller objects, this is usually because some subjects were able to pick up the object while others were not able to do so in the allotted four minutes. For larger objects, the high error is usually associated with Method 3. Some subjects found it difficult to create hand poses which were easily distinguishable for the classifier and therefore took much longer to pick and place than subjects for whom the pose classifier worked well.

In Section 4.4, we reported the average time to pick and place for novice users with teleoperation subspace mapping controlled by a dataglove was 27.5 s and the average time to pick and place with other state-of-the-art teleoperation mappings was between 62.3 and 56.7 s. When compared to EMG teleoperation, we see that Method 1 enables teleoperation faster than the state-of-the-art methods controlled by a dataglove, and only 10 s slower than teleoperation subspace mapping controlled by a dataglove. Methods 2 and 3 enable teleoperation at about the same speed as state-of-the-art dataglove teleoperation, but much slower than subspace mapping teleoperation. As expected, EMG control is slower than our dataglove control, as the user has to perform additional gestures (spread and isometric contractions) during grasping. The fact that the EMG control is still faster than state-of-the-art methods controlled by a dataglove demonstrates the usefulness of the teleoperation subspace.

6.5 Summary

We introduced here a method for teleoperating a non-anthropomorphic, fully actuated robot with forearm EMG, leveraging the teleoperation subspace introduced in Chapter 3. We use the teleoperation subspace as an intermediary between EMG and robot pose space, and combine continuous and discrete models to map into this subspace. To the best of our knowledge, we show

the first instance of EMG controlled teleoperation of a non-anthropomorphic, fully actuated robot hand.

We compared our control to other state-of-the-art EMG teleoperation methods, which we modified as needed to work with non-anthropomorphic hands. Our method allows users to form stable grasps around a variety of objects faster and with fewer tries than the state-of-the-art methods. We also enable teleoperation faster than state-of-the-art teleoperation controlled with datagloves, and only slightly slower than dataglove-controlled teleoperation subspace mapping.

Although the control we describe here requires a dataglove to train the EMG regressor, the dataglove is removed when our teleoperation method is used in practice. EMG armbands are less expensive and easier to replace than datagloves. Furthermore, their placement on the forearm, rather than on the hand, makes them less susceptible to damage and leaves the hand completely unencumbered as the teleoperator performs collaborative tasks with the robot.

In future work, we would like to validate our EMG control on hands with different kinematic configurations.

Chapter 7: Wearable Orthotics for Stroke Subjects

In the previous four chapters, we propose a subspace relevant to teleoperation, and then develop three methods to leverage this subspace in order to control fully-actuated non-anthropomorphic robots. Our target users are able-bodied subjects working in a human-robot team to complete a task. In this paradigm, we assume that the human and the robot will work as independent agents, unless the robot requires assistance completing its task.

However, there exists a large population of users who do not have full use of their hands. These users would benefit from the robot acting, not as independent agent, but as an assistive device. In these cases, the robot device, when mounted on the user's hand, would allow them to complete tasks they would not otherwise be able to perform.

These types of devices, called orthoses, can be used for rehabilitation, helping the user to regain function of their hand, or to help the users complete Activities of Daily Living (ADLs), functional activities that could improve their quality of life.

Wearable orthoses are an attractive alternative to other robotic rehabilitation therapies that, traditionally, require therapist supervision provided in a clinical setting, and take place in a non-functional context. Therapy is more likely to be effective when training is distributed in smaller but more frequent aliquots [129] and when training includes performing actual ADLs [130]. Wearable orthoses can provide both of these advantages.

When developing controls for wearable orthoses, our end goal is a reliable, intuitive, user-driven control that can be used in functional contexts, and that can be used without therapist supervision. In the following chapters, we develop orthotic controls which represent a step toward user-controlled take-home orthotic devices to help perform functional tasks.

In this dissertation, we focus specifically on developing an orthotic control for stroke subjects. Our control is developed in conjunction with an underactuated exotendon orthosis (more details

about the hardware are provided in the following chapters).

Since the target users in this context do not have full use of their hand, we develop an EMG-based control, using signals from the affected forearm as input. In stroke subjects, EMG signals are still present, even though the physical movement associated with the EMG signals may be impaired. This control input is intuitive, and repeated use can strengthen the muscles of the user's affected arm.

In the previous chapter, we sought to translate subject EMG into a pose in a three dimensional teleoperation subspace. Since the orthosis we use in this dissertation is underactuated, with only one degree of freedom, our orthotic EMG control must only determine whether the orthosis should open the hand or close the hand.

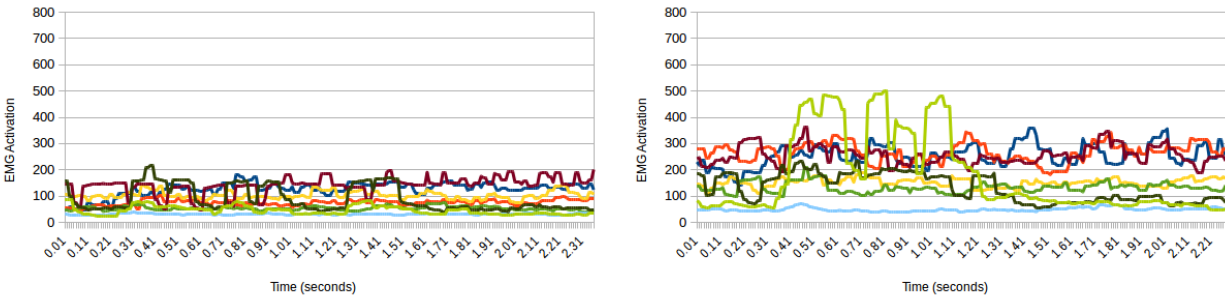
7.1 Challenges

At first glance, it would seem that our end goal of controlling a robotic orthosis with a user's EMG signals is a similar problem to that we solved in the previous chapter, using EMG signal to teleoperate a robot hand using an able-bodied user's EMG signals.

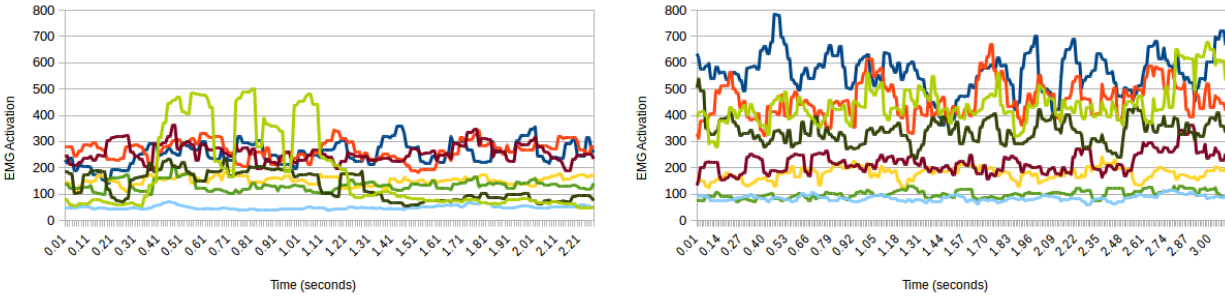
In some cases, we do utilize the same solutions across both applications: to make the control easy to don, we use an encapsulated commodity sensing suite and pattern classification. The sensing suite is an armband which can be slipped on easily, and pattern classification means that we do not require the sensors to be placed on specific muscles. Similarly, in both applications, we tailor the control to each subject's EMG patterns. We collect a training dataset and train the control specifically for that subject.

However, stroke subjects also present a very specific set of challenges for EMG control that must be addressed. The main challenge, and one that is virtually non-existent in the literature for stroke orthotics, is making the control robust to changes in the EMG signal over time, also known as concept drift.

When control algorithms do not account for concept drift, the performance of the control will degrade. Over time, as the signals drift in the feature space, they will become unrecognizable to a



(a) An example of how the abnormal muscle coactivation can affect the subject’s EMG. On the left the subject relaxes their hand with their arm on the table. On the right, the subject relaxes their hand, but raises their arm off the table.



(b) An example of how the orthosis can affect the subject’s EMG. In both photos, the subject relaxes while raising their arm off the table. On the left, the orthosis is not engaged and the subject’s fingers are flexed due to spasticity. On the right, the orthosis is engaged and actively extending the subject’s fingers.

Figure 7.1: Two examples of concept drift in the EMG signals of a stroke patient. Each graph shows eight signals from the sensors of our EMG sensing suite. These signals are produced by a single subject under different conditions.

classifier. EMG based controls, in particular, are susceptible to concept drift, since EMG is not a stationary signal [95].

Concept drift can occur both within a single session (intrasession drift) and across sessions (intersession drift). In healthy subjects, concept drift for EMG based controls can be caused by fatigue, repositioning of the EMG sensors, and even the subject’s sweat, among other factors [93].

In our teleoperation experiments, we did not find that concept drift contributed significantly to the control’s performance. Subjects were generally able to use the controls consistently across the entire session without any special considerations for concept drift.

In stroke subjects, this concept drift is affected by the factors above, and is additionally aggravated by the subject’s abnormal muscle coactivation [131] and by the interaction of the hand and the orthosis. Abnormal muscle coactivation typically affects the control signal when the subject

moves their arm to a different position; we show an example in Figure 7.1a. The interaction between the hand and the orthosis affects the control signal when the orthosis elicits movement in the subject's hand; signals recorded when the orthosis is holding the hand open will be different from signals recorded when the orthosis allows the subject's hand to close, even when the subject's intent is the same. We show an example in Figure 7.1b.

7.2 Summary

In this chapter, we introduce wearable orthoses as a potential area where human-machine interfaces that are accessible to novices could potentially improve the quality of life of a large population. We discuss some of the challenges which are specific to the stroke population, like concept drift.

This dissertation is the first to explore concept drift of forearm EMG in stroke subjects. We propose algorithms and protocols which allow an orthotic control to adapt to concept drift caused by abnormal muscle coactivation or the interaction between the hand and the orthosis.

In the next three chapters, we propose and validate an EMG-based algorithm to control an underactuated orthosis for stroke patients. We seek to make this control easy to don, robust to concept drift, and intuitive. We will outline both a supervised algorithm, whose parameters are determined during an initial training period and do not change, and a semi-supervised algorithm, whose parameters are updated online by incoming, unlabeled test data.

Chapter 8: Supervised EMG Control for Wearable Hand Orthotics

In this chapter, we outline a supervised algorithm to detect a stroke subject's intention to open or close their hand, using EMG as a control input.

Our approach uses pattern classification - identifying patterns in EMG signals from the entire forearm - to determine user intention. We apply EMG pattern classification to determine the user's intention either to open or close the hand, and use this signal to produce physical movement, assisted by the orthosis. While we do not measure cognitive load, the short training sessions that enable the user to operate the device suggest that using signals from the impaired hand muscles is an intuitive control mechanism for an orthosis.

Supervised classifiers, learners whose parameters are determined during an initial training period and do not change, regardless of the concept drift occurring in the incoming test data, must compensate for concept drift using training data. If the training data is insufficient, the performance of the classifier will degrade. We therefore present a training protocol which provides our intent detection algorithm with enough examples to adapt to the concept drift seen during functional tasks.

Previous work from our lab presented an exotendon orthosis - a soft glove with guided tendons driven by linear electric actuators to elicit desired movement patterns [132]. We pair our EMG-based control mechanism with this exotendon device and perform testing as the device elicits functional movements for the subject. Figure 8.1 shows this exotendon device, including the EMG armband which provides the control input.

Overall, we present a complete mechanism comprising hardware and algorithms to:

- Detect a motor impaired user's intention to execute specific hand movements based on forearm EMG.

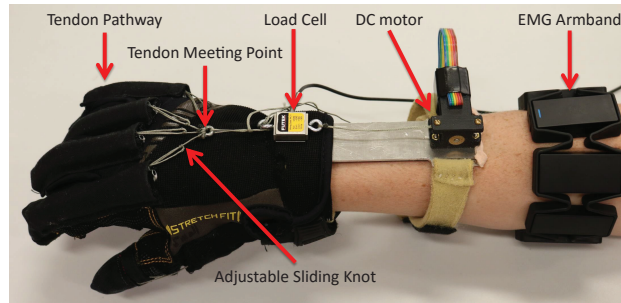


Figure 8.1: Prototype hand orthosis in extension configuration with EMG armband.

- Use an encapsulated commodity EMG sensing suite without needing precise sensor positioning. This is in contrast to medical grade EMG sensors placed on a specific muscle by a trained professional, which do not allow in-home use outside of direct medical supervision.
- Physically elicit the desired movement pattern in stroke patients. We do not study EMG in isolation: we combine the control with a real orthosis and show it enables functional grasping in our target population. The presence of the physical device results in concept drift, altering the EMG data obtained during operation; our approach is designed to cope with this phenomenon.

To the best of our knowledge, no existing hand orthotic system has concurrently demonstrated all of these characteristics. The key to our approach is pattern classification, which enables the use of commodity EMG armbands and a training protocol which makes the control robust to concept drift. Commodity armbands have the potential to allow complete portability as well as user-directed, in-home use. *Pattern classification of functional movements for stroke subjects has so far not been studied in conjunction with a physical orthosis that enables grasping.* Our experience indicates that concept drift makes it important to study these components together, as we aim to progress towards fully user-driven execution of functional tasks.

8.1 Hardware

Our EMG-based control approach is implemented and tested on a complete hand orthosis device, which we briefly describe here. In prior work, Park et al. presented a four fingered orthosis

with two separate 1-degree of freedom (DOF) tendon configurations[132]. Here, we experiment with the configuration assisting with extension, often a difficult task for stroke patients because of the commonly observed impairment pattern of spasticity, which is excessive involuntary flexion. We chose to explore the prototype which enables extension because extension is essential for functional grasping. Extension is achieved by applying extension torques on the fingers through an extendon network pulled by a DC motor.

Mechanical components are split into two modules: a forearm piece and a glove with a tendon network. The modules are connected via eyerings on both sides of the load cell (Futek, FSH00097) to facilitate donning (Figure 8.1). With a therapist, donning takes approximately 5 minutes. The device weighs 135 grams.

The forearm piece is composed of an aluminum splint and a DC motor. The splint constrains wrist movement to efficiently transmit external torques to the fingers. The splint is angled either at 30 degrees, considered functional wrist pose [133], or at 0 degrees for the patients who cannot extend the wrist due to spasticity. A DC motor (Pololu corporation, 210:1 Low-Power Micro Metal Gearmotors) with a 15 mm/s maximum travel speed and a 80N peak force is mounted on the splint. Motor specifications are chosen to prevent dangerous tendon force levels without taking up space to improve wearability. A Proportional-Integral-Derivative (PID) position control is implemented to drive the motor. The motor's range of motion is determined at the beginning of clinical tests, depending on hand size.

The glove has a tendon network guided from the heads of the middle phalanges through raised pathways to a meeting point on the back of the hand. The tendons on each finger are attached to a cloth ring on the middle phalanges rather than on the fingertips to avoid finger hyperextension. The tendons on all digits, except for the thumb, are routed on the dorsal side of the glove. The thumb needs a special routing scheme since it exhibits different movement patterns from the other digits; the thumb tendon is routed from the proximal phalanx head to the metacarpal joint, then wraps clockwise around the wrist to the eyering on the load cell. The tendons on all digits are tied with sliding adjustable knots to allow better fit for different finger lengths.

The DC motor which pulls the exotendon network is driven by the EMG-based control that is the main focus of this chapter. Once the EMG control determines whether the user's intention is to open or close the hand, it sends a command to the DC motor mounted on the splint. The motor then extends or retracts the tendon network to allow the user to open or close the hand.

8.2 Intent Detection

EMG patterns of the hemiparetic forearm are often altered after a stroke event [134]. This study assumes and, through experiments, verifies that these altered EMG patterns can still be used to control a hand orthosis, since control using forearm EMG sensors has a number of compelling characteristics. EMG-based control requires the same type of muscle activation as pre-stroke extension, which makes the control intuitive and place a low cognitive load on the user. Additionally, using ipsilateral EMG control leaves the other hand free to participate in the grasping task or to perform a different task.

Beyond altered signals however, EMG control of an orthosis for a stroke patient is difficult because of additional phenomena, such as spasticity and abnormal coactivation relationships between muscles [134]. As such, many orthoses that enable pick and place collect signals from only two muscles [83, 84, 85], with each muscle controlling a direction of the orthosis, often using a threshold based on the subject's maximum voluntary contraction. In these approaches, the subject must fully extend or close before the orthosis will move in the other direction. We developed an exotendon device that responds immediately to a signal change from the control, throughout the range of motion of the user's hand. The user's ability to end extension allows more natural grasping for smaller objects as well as the option to change grasping tasks mid-motion.

One of the key tenets of our approach is to rely on signals from a multitude of sensors placed around the forearm. Unlike simple intensity thresholding, which is effective for a single sensor precisely located on a specific muscle, pattern classification identifies patterns in the complete set of signals from the sensors. This approach has three main benefits:

1. It enables the use of commodity sensors. Even though the quality of the EMG signal from

commodity sensors is lower than medical grade sensors, we compensate for signal quality with sensor quantity. Pattern classification provides an image of the overall EMG signal in the entire forearm instead of trying to isolate a high quality signal from specific muscles.

2. It eliminates the need to search for specific muscles with exact sensor placement. Pattern recognition examines EMG signals from the entire forearm. Studies have suggested that when electrodes are placed around the entire forearm, targeted and untargeted placement of EMG electrodes result in similar classification accuracies [135]. Throughout our experiments, the only effort to position our EMG sensors was placing one of the sensors on the dorsal side of the arm. Even with this untargeted approach, we were still able to use pattern classification with good accuracy. The flexibility in sensor placement means that donning our control unit does not require a therapist, or even a basic understanding of forearm anatomy. For a device that is designed for take-home use in mind, this is an extremely desirable quality.
3. It allows for the possibility of an orthosis with more DOFs. Current orthoses look at two specific muscles, a flexor and an extensor. The flexor controls the close motion of the orthosis and the the extensor controls the open motion. Pattern classification allows for the recognition of more complex muscle motions, which could control different DOFs of the orthosis [87]. This gives users more motion options, if the orthosis has multiple degrees of freedom.

To acquire the EMG signal, we use the Myo Armband from Thalmic Labs. It has 8 EMG sensors and 8 IMUs, which can indicate the orientation and acceleration of the device. In this study, we only use the EMG sensors; however, the IMU sensors could be useful for future control iterations.

Pattern Classification: Our pattern classification algorithm seeks to take the 8-dimensional raw EMG data from the 8 Myo sensors and identify patterns that correspond to certain desired hand motions. Our current algorithm identifies hand opening and closing, but we plan to incorporate more complex patterns into the classification scheme in future iterations.

We collect raw EMG data from the Myo Armband at a rate of 50Hz. At time t , we collect the EMG signals e_j^t from the sensors and assemble them into a data vector \mathbf{x}_t :

$$\mathbf{x}_t = (e_1^t \dots e_8^t) \quad (8.1)$$

We define the desired hand state at time t as $Intent_t \in \{O, C\}$, where $Intent_t = O$ corresponds to the intent to open the hand and $Intent_t = C$ is the intent to close the hand. While training, ground truth data $Intent_t^g$ is provided by the experimenter who gives the subject verbal commands to open or close the hand. The training period is around 45 seconds - allowing the experimenter to command the user to try to open and close the hand twice. Although this training time is short, we receive a large quantity of data points ($\sim 2,400$) which we use to establish patterns in the EMG with our classifier.

Raw EMG signals are used as the features for the classifier. Although many pattern classifiers require extraction of time-domain features, we receive our data at 50 Hz, so this would be impractical. Our results in Section 8.4 show our classifier is robust enough that it does not need to extract time-domain features to classify intention with high accuracy.

Our first order goal is to predict $Intent_t$ based on \mathbf{x}_t (we will further process this result as explained in the next sections). We use a random forest classifier trained on the ground truth data described above to make this prediction. A random forest classifier is an ensemble machine learning method created from a combination of tree predictors [136]. Because of the random nature of the bootstrap sampling used to create our classifier, the number of decision trees in the forest classifier and the decision trees themselves change with every training iteration. Despite the underlying randomness, our classifiers for all subjects still achieve high accuracy.

We denote the random forest classifier function as:

$$CLAS(\mathbf{x}_t) = p_t^O \in [0, 1] \quad (8.2)$$

where p_t^O is the probability that $Intent_t = O$ (at time t , the user's intention is to open the hand).

The converse probability that the user’s intent is to close the hand is simply $p_t^C = 1 - p_t^O$. We filter and use this result as described in the next section.

Output Processing: We collect raw EMG data \mathbf{x}_t at a rate of 50Hz. However, the time scale for hand opening and closing and for pick and place tasks is much lower frequency than the rate at which data is collected, so classifying individual data points correctly is not as crucial as correctly identifying a hand motion. To identify these motions, we assume hand posture does not change with high frequency, which allows us to filter and process the probabilities returned by the classifier.

While filtering raw EMG signals is a common technique, we chose instead to apply our filter to the results of the classifier. We compute filtered probabilities at time \hat{T} as:

$$\hat{p}_{\hat{T}}^O = \text{MEDIAN}(p_t^O), t \in [\hat{T} - 0.5s, \hat{T}] \quad (8.3)$$

$$\hat{p}_{\hat{T}}^C = \text{MEDIAN}(p_t^C), t \in [\hat{T} - 0.5s, \hat{T}] \quad (8.4)$$

The 0.5s median filter increases transition delays, but helps eliminate spikes and spurious predictions. 0.5s was chosen because shorter filters resulted in spurious classification errors. Despite the delay, our subjects reported no noticeable delay between intention initiation and device movement.

We note that, as a result of filtering, generally $\hat{p}_{\hat{T}}^O + \hat{p}_{\hat{T}}^C \neq 1$.

To produce the final output for our control, we compare $\hat{p}_{\hat{T}}^O$ and $\hat{p}_{\hat{T}}^C$ against two threshold levels, L^O and L^C respectively. If $\hat{p}_{\hat{T}}^O \geq L^O$, then the controller issues an “open” command (retract the tendon). If $\hat{p}_{\hat{T}}^C \geq L^C$, then the controller issues a “close” command (extend the tendon). If neither condition is met, no new command is issued and the orthosis continues executing the command from the previous step. The values of L^O and L^C are set manually by the experimenter for each subject after completing training data collection, then kept constant throughout all tests. The thresholds are set with subject feedback such that the control is responsive, but there are no spurious errors during sustained hand commands.

Device State	Subject Instruction		
	Open	Relax	Close
Tendon extended	O	C	C
Tendon retracting	O		C
Tendon retracted	O	C	C

Table 8.1: Training protocol and assigned labels. For each combination of instruction given to the subject and state of the exotendon device, the table shows the ground truth label $Intent_i^g$ assigned to EMG data. Training begins with the tendon extended and the subject asked to relax (top row, middle column) and proceeds in counter-clockwise fashion

8.3 Training with the Exotendon Device

The most straightforward method for generating training data to use with the classifier described above would be to simply instruct the user to attempt to open or close the hand, and label the resulting data accordingly. However, we quickly found that this simple procedure is flawed for multiple reasons. First, for stroke patients, we found that the default “relaxed” hand state (attempting to neither open nor close) still produces a strong, subject-specific EMG signal. The classifier would displayed a tendency to label this signal as either open or close, unless we provided explicit training data illustrating the difference. Second, we also found that concept drift caused by the physical interaction with the orthosis itself altered the EMG patterns: for the same user intention, signals recorded with the tendon fully retracted (assisting in hand opening) differed from those recorded with the tendon extended.

We address both of these issues through our training protocol and collection of labeled training data. Specifically, we design our training protocol as follows:

- We instruct the subject to attempt three hand poses: open, closed, and relaxed. For data collected during both closed and relaxed intents, we assign a ground truth label $Intent_i^g = C$, corresponding to a closed hand. Since our target population comprises patients with spasticity, this more closely mimics the subjects’ natural state. This means that, for the orthosis to provide assistance, we must be detecting an active attempt by the user to open their hand. Being conservative in when to send a command to retract the tendon (and thus

actively open the hand) reduces the risk of holding the hand open for longer than desired and causing discomfort. We note that one disadvantage is that continuous effort from the subject can lead to muscle fatigue, especially if the subject exerts great strain to provide an open signal.

- For all three user intents (open, close, relaxed) we collect training data in different states of the exotendon device, namely with the tendon fully extended, fully retracted, or moving between states. The training procedure is as follows. We instruct the subject to relax, with the tendon fully extended. We ask the subject to attempt to open the hand, with the tendon still fully extended. As the subject continues trying to open, the experimenter commands the tendon to retract, opening the hand. Once the tendon is fully retracted, we instruct the subject first to relax, then to attempt to close the hand. The experimenter then commands the tendon to extend, allowing the hand to close. Finally, the subject is told to relax. This procedure, and the ground truth labels assigned at every phase, are summarized in Table 8.1.

The result of this training procedure is a labeled ground truth dataset covering combinations of user intent and device state. We use this dataset to train the classifier described above; at run time, the output of the classifier produces a command for the exotendon device as detailed in the previous section.

8.4 Experiments

Testing was performed with 4 stroke survivors, 1 female and 3 male. Subjects showed right side hemiparesis following a stroke event at least 2.5 years prior and had a spasticity level between 1 and 3 on the Modified Ashworth Scale (MAS). Table 8.2 shows clinical scores for all subjects. Testing was approved by the Columbia University Internal Review Board, and performed in a clinical setting under the supervision of Physical and/or Occupational Therapists.

We asked each subject to don the Myo and the exotendon device with the assistance of the supervising therapist. Training the EMG control as described in Section 8.3 was performed for every

Table 8.2: Subject demographic and clinical information

Subject	Age	MAS Extensor Score			MAS Flexor Score		
		Elbow	Wrist	Finger	Elbow	Wrist	Finger
A	60	2	1	1	2	2	2
B	39	1	0	0	2	3	3
C	80	0	0	1	1	0	2
D	66	2	0	0	2	2	1

session. The trained classifier was then used throughout the entire session. In real deployment, we would like the classifier to be robust enough to take the armband off and put it back on. We predict this is possible if the armband orientation on the forearm is consistent; however, this was not explored in this chapter because subjects’ EMG patterns could potentially change as the patient undergoes rehabilitation. We explore reusing a classifier at a later date in the next chapter.

After training, each subject performed 4 experiments:

1. **EMG control without the device operating:** This experiment determined if the EMG signal in the hemiparetic forearm arm was strong enough to indicate the subject’s intention to open or close. Without the device operating, there was little hand movement, but we still were able to determine the user’s intention.

To collect the training set, the subject was asked to try to open and close the hemiparetic hand, with the understanding that the fingers likely would not extend, but that the EMG signal would change as different actions were attempted. (Note that this is the only condition in which we did not use the training protocol described in Section 8.3.) The testing set was collected in the same way as the training set. The subject’s hand did not move, but the classifier was able to predict the subject’s intention by the EMG signals.

2. **EMG control with the device operating:** This experiment determined whether EMG control, in conjunction with our exotendon orthosis, could enable hand extension. With the device on, the Myo sends raw EMG signals to the classifier, which predicts intent and sends a command based on intent to the motor, which retracts or extends the exotendon to move the hand and enable extension. Because this enables extension, it requires the training protocol

described in Section 8.3. The subject's forearm was at rest on the table.

In this experiment, the device operated to extend the hand, so training used the protocol from Section 8.3.

The testing set was collected as the subject was asked to try to open and relax the hemiparetic hand while resting the hand on the table. If the classifier detected the subject was attempting to open, the exotendon device would retract the exotendon and the subject's hand would extend. If the intention to open was absent, the device allowed the hand to close.

3. **EMG control during pick and place:** This experiment determined whether the exotendon device, in conjunction with EMG control, could enable pick and place. The exotendon device enabled hand extension but the forearm was no longer supported by the table.

We did not do additional training for this set, but used the classifier from the previous experiment.

During testing, the subject was asked to operate the exotendon device using EMG control to pick an object up, move it several inches, and then place it back down.

4. **Button control during pick and place:** This experiment provided a baseline control comparison for the EMG control. A push button is attached to the device's motor and can be used to retract and extend the tendon. Pushing down and holding the button opens the glove until the hand is fully extended. Releasing the button at any point of the extend cycle causes the tendon to be released immediately and allows the hand to relax. The subject used the button with the non-affected hand to activate the device and complete pick and place tasks.

Before testing, the subject was instructed how to control the device using their left hand, and allowed to use the control for several minutes before performing pick and place tasks. During testing, the subject performed pick and place on the same object as during the EMG controlled experiment.

Experiments were performed at a pace comfortable for the subject and breaks were given be-

Table 8.3: Results for non-functional motions

Subject	Without the Device Operating		With the Device Operating	
	% Accuracy	Correct Events	% Accuracy	Correct Events
A	85.2%	11/18	93.6%	16/18
B	90.1%	10/16	83.4%	4/16
C	93.6%	12/14	90.9%	9/11
D	82.2%	4/10	N/A	N/A

tween experiments.

8.5 Results

Our results report two metrics: prediction accuracy and correctly predicted events. Prediction accuracy is defined as the percentage of individual data points predicted by the classifier to be the same as ground truth. However, we believe that the more important metric is the ability to correctly execute a complete, meaningful hand motion, such as opening or closing. We attempt to capture this using the number of correctly predicted events. An event is defined as a change in intention signal, and a correctly predicted event means a predicted event which occurs within 850 ms of the ground truth event, with no incorrect classifications until the next event. The 850ms allowed for lag introduced by the median filter and allowed the subject to initiate the action after a verbal command. Success for these metrics was if the EMG control did as well as the baseline button control.

EMG control without the device operating: The classifier for Subject A had an accuracy of 85.2% and correctly predicted 11 of 18 events. The intention for Subject B was predicted with 90.1% accuracy and 10 of 16 events were correctly predicted. The classifier for Subject C had an accuracy of 93.6% and correctly predicted 12 of 14 events. Subject D’s intention was predicted with a 82.2% accuracy and 4 of 10 events were predicted. Figure 8.2 shows ground truth, prediction results and non-thresholded filtered probability vs. time of Subject A, as well as the raw EMG which is classified. See Table 8.3 for a summary of the results.

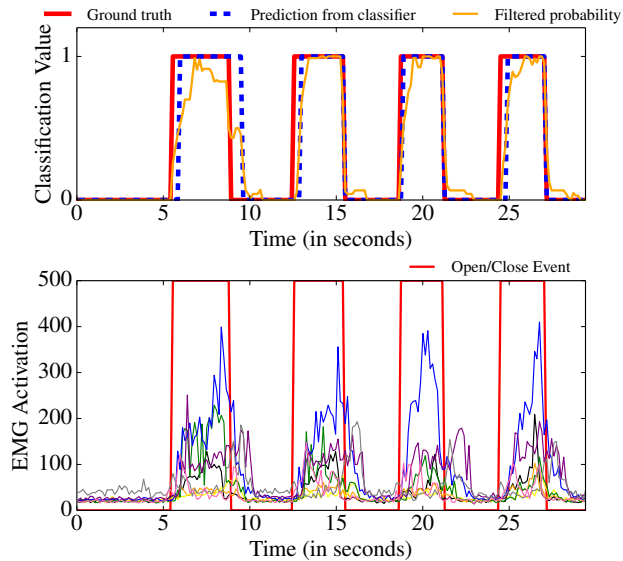


Figure 8.2: **Top:** Prediction of classifier (dotted blue), ground truth (solid red) and filtered probability before thresholding (solid orange) vs. time for Subject A without the device operating. Classification value of 1: open intention, classification value of 0: no open intention. **Bottom:** Raw EMG of Subject A and open close events which correspond to the top graph.

EMG control with the device operating: Subject D was not included in these results because of subject fatigue. The classifier for Subject A had a prediction accuracy of 93.6% and correctly predicted 16 of 18 events. The classifier for Subject B had an accuracy of 83.4% and correctly predicted 4 of 16 events. Subject C had an accuracy of 90.9% and the classifier correctly identified 9 of 11 events. The finger flexor MAS score for Subject B was higher than for Subjects A and C. This could explain why Subject B’s accuracy and correct event prediction are lower. An plot of the ground truth and the prediction results vs. time of Subject A, as well as the raw EMG which is classified, can be found in Figure 8.3. See Table 8.3 for a summary of these results.

EMG control during pick and place: Precise ground truth is difficult to establish when the subject is performing pick and place tasks because an operator instructing the user when to begin and end extension would result in unintuitive grasping. Instead of percent accuracy, we use the number of correctly executed pick and place tasks as a metric for the pick and place experiments (both with EMG control and with button control).

The details of a complete pick and place motion, as well as the exotendon’s role in the action

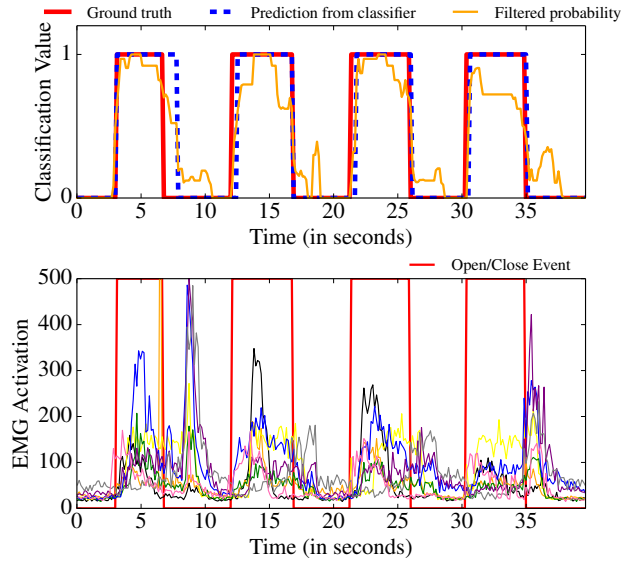


Figure 8.3: **Top:** Prediction of classifier (dotted blue), ground truth (solid red) and filtered probability before thresholding (solid orange) vs. time for Subject A with the device operating. Classification value of 1: open intention, classification value of 0: no open intention. **Bottom:** Raw EMG of Subject A and open close events which correspond to the top graph.

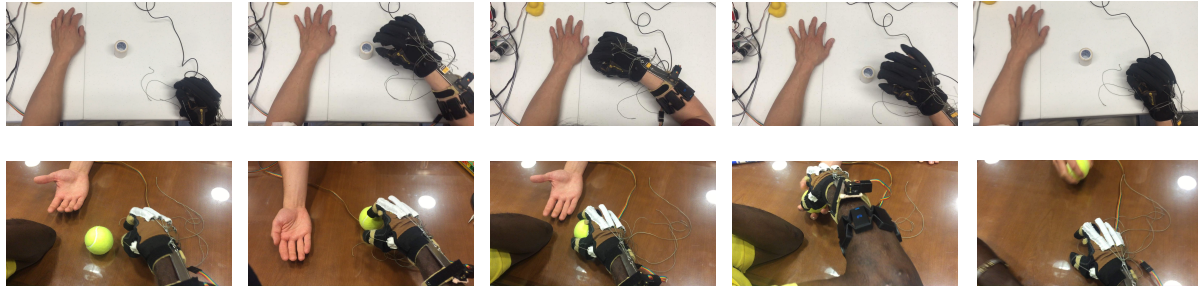
Table 8.4: Results for functional motions (pick and place)

Subject	Correct Events - EMG Control	Correct Events - Button Control
A	6/13	3/3
C	6/6	5/5

are described in Figure 8.4.

Subject B was not included because sizing issues rendered her unable to grasp objects. Due to subject fatigue, Subject D was also not included. Subject A successfully completed 6 of 13 pick and place attempts. Subject C completed 6 of 6 pick and place movements. We note that Subject C was higher functioning than the other subjects and was generally able to complete unassisted hand extension, albeit with significant difficulty. Nevertheless, the subject reported that the device provided assistance in hand opening during pick and place. See Table 8.4 for a summary of the results.

Button control during pick and place: Again, Subjects B and D were not included in these results. Subject A successfully completed 3 of 3 pick and place attempts. Subject C successfully



Subject is at rest.

The tendon begins in fully extended state and the hand is closed due to spasticity. Subject begins grasp. The tendon retracts and the hand opens to enable grasping.

Subject grasps the object. The tendon extends to allow the hand to close over the object.

The subject releases the object. The tendon retracts to open the hand and to allow release of the object.

Grasp is complete and the subject is at rest. The tendon is fully extended.

Figure 8.4: Illustration of pick and place tasks by 2 subjects. Each row shows a complete task execution by one subject, progressing from left to right.

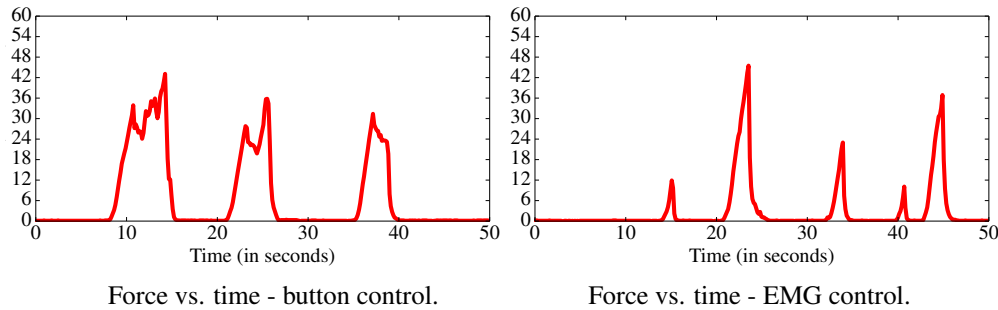


Figure 8.5: Force results for Subject A. Each peak corresponds to one hand extension for grasping.

completed 5 of 5 pick and place attempts. See Table 8.4 for a summary of the results.

Tendon Forces during Pick and Place Experiments: To assess the ability of the control to correctly interpret user intent, as well as the level of discomfort caused by operation, we measured the forces applied on the exotendon network during pick and place experiments. For the same subject, we compared peak forces obtained for EMG control versus button control. Our assumption was that an incorrect interpretation of a “close” signal, where the subject intended to close the hand but the assistive device did not react appropriately, would result in a spike in the force levels as the subject would be effectively fighting against the orthosis.

During the button controlled pick and place, the peak force on the tendon was 53.7N for Subject A and 58.6N for Subject C. During the EMG controlled pick and place, the peak force on the

tendon was 59.6N for Subject A and 77.6N for Subject C. Figure 8.5 shows an example plot of force vs. time during an EMG controlled and a button controlled experiment; each time series contains multiple hand extensions. We measured the forces using a load cell placed in series with the exotendon network. The peak force for the two controls showed little difference for Subject A, but for Subject C, EMG control lead to higher forces. However, there was enough variance between force peaks obtained with the same control mechanism to suggest that the difference could fall within normal operating range. Additional testing will be needed to further study this issue.

We note that, for Subject A, as shown in Figure 8.5, the button control results in more prolonged peak tendon forces. For the EMG controls, the tendon forces peak, and then immediately go down. With the button controls, the tendon forces remain close to the peak value for several seconds before they abate. We hypothesize this indicates that the subjects fight the exotendon device less with the EMG control than they do with the button control.

Discussion: Overall, our results showed effective pattern classification performance, to the level of physically enabling functional hand motion. Still, classification accuracy shows significant room for improvement. In particular, while the percent accuracy metric was consistently above 80% and often above 90%, the same level of performance was not achieved in the number of correctly predicted events. Most of the incorrectly predicted events were the result of the control not correctly recognizing the change in intention within the allowed 850ms window, rather than spikes caused by misclassification in the middle of the event. These delays were caused in part by the median filter, which uses the past 500ms to inform the control, thereby adding lag. Another possible cause was subject spasticity, which made it difficult for the subjects to relax or close after activating their extensor muscles.

Our results were trained and tested on separate data sets, both of which were taken from the same patient during the same session. We would like our trained classifiers to be robust enough to work for the same patient for different sessions. However, we believe it unlikely that pattern classi-

fication would work between different patients as EMG patterns between subjects are substantially different [86].

Pick and place experiments controlled by EMG showed lower accuracy than non-pick and place experiments where the device was operating. The difference between the two types of experiments was that in the former the subject's arm was engaged in the task, while in the latter the forearm was simply resting on the table. We hypothesize that, because of the stroke subjects' abnormal coactivation, a classifier which was trained while the forearm was resting on the table is confused by elbow extension during a grasping motion. In the next chapter, we to compensate for this effect by altering our training protocol to include training data both when the arm is resting and when the arm is extended.

8.6 Summary

In this chapter, we describe a complete algorithm for processing and classifying EMG signals from a stroke subject's forearm. The pattern classification we describe allows the use of commodity devices, which are easy to don, as there is no need to place sensors on specific muscles.

We also describe a training process which helps the classifier generalize to both functional and non-functional movements. Without this training process, concept drift caused by the abnormal coactivation and spasticity commonly found in stroke subjects lowers the classifier's accuracy.

In our experiments, we show that an EMG based pattern classification control of an exotendon device can enable functional movement for a stroke survivor. Our control achieves high accuracy during non-functional open and close hand motions, and can enable functional motions, like pick and place. Our control is intuitive and does not require an extended period of training.

This chapter is presented as a feasibility study. In the next chapter, we present a study conducted over a longer training period. We explore if repeated use of our EMG control, in conjunction with an orthotic device, can provide rehabilitative gains for stroke subjects.

Chapter 9: Rehabilitative and Assistive Performance of a User-Controller

Hand Orthosis

In the previous chapter, we showed that our intent detection algorithm could enable functional motions in conjunction with an orthosis for a stroke patient.

In this chapter, we show that our algorithm, when used in conjunction with the orthosis over an extended training period can provide rehabilitative improvements to stroke subjects. We present clinical outcomes from a 12 session training program, comprising 3 sessions per week for 4 weeks. The original study had 11 subjects; in this dissertation, we will focus on the six subjects who used our EMG based control, but we will draw comparisons between the EMG-based control and the second control from the study. The study in its entirety can be found elsewhere [137].

In order to determine the efficacy of the orthosis as a *rehabilitative device*, we compare clinical outcomes both pre- and post-intervention *without device assistance*. To determine efficacy when used as an *assistive device*, we compare pre-intervention clinical outcomes without assistance with post-intervention performance *while wearing the device*. Our goal is to determine if increased competence using the device from a month-long training program leads to increased performance in tasks requiring grasp, transport, and release of objects while wearing the orthosis.

Since the subjects are required to use the device for longer training periods than in our feasibility study from the previous chapter, we propose modifications to our control algorithm designed to make it more robust to fatigue and to concept drift.

Overall, the main contributions of this chapter are as follows:

- We validate the EMG control proposed in the previous chapter, providing a more extensive study where the subjects are asked to perform functional tasks with a variety of objects of different shapes and sizes.

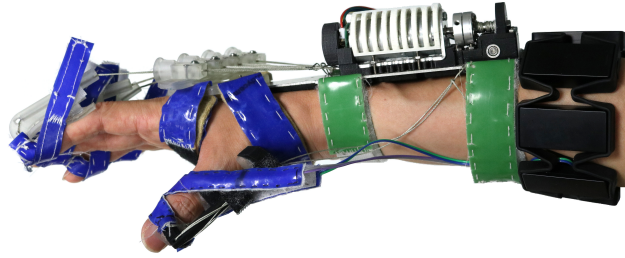


Figure 9.1: Updated hand orthosis prototype in extension configuration.

- Our results suggest that our EMG control, when used in conjunction with an orthosis to enable functional motions over a span of several weeks, can serve as a rehabilitative tool to remediate some functional use in the affected hand.
- To the best of our knowledge, this study is the first time an active wearable hand robot was evaluated in clinical assessments both with and without robotic assistance, following user-driven functional hand exercises over multiple training sessions for chronic stroke patients interacting with various real-world objects.

Our aim is to provide wearable, user-driven assistance for the stroke-affected hand, and to investigate its effects on both rehabilitation and functional performance. These are steps towards our long-term goal to test user-operated active orthoses outside of a clinical setting, for rehabilitation or home-assistance, with the potential to significantly increase the number of motor repetitions and the intensity of training performed by the subject.

9.1 Hardware

In the previous chapter, we described an underactuated exotendon device. In this chapter, we use an updated prototype of the same device. The updated prototype has fingertip components instead of a complete glove for the subject's hand, and splints the thumb instead of actuating it. Figure 9.1 shows the exotendon device. We briefly review the updated hardware and the EMG sensors below.

Exotendon Device Our orthosis is an exotendon-based device which consists of 1) a forearm splint which constrains wrist movement and on which the motor is mounted and 2) 3D-printed fingertip components which are secured to the fingertips using Velcro straps. These two parts are connected through an exotendon network routed through raised pathways. The exotendon network is underactuated, moving all of the subject’s fingers at once. When the tendons are retracted by the motor, the subject’s fingers extend. When the motor releases the tendons, the subject uses their own grip strength to close their hand. The thumb moves differently than the other four fingers, and is therefore splinted into place with two additional tendons which are not attached to the motor.

Forearm EMG We measure the raw EMG signals of the subject’s forearm using a commercial EMG armband (Myo Armband from Thalmic Labs). The armband consists of eight sensors and is placed on the subject’s forearm, proximal to the splint. This device is low profile, and easy to don and doff. EMG is a common orthotic control and can identify hand poses with relatively simple algorithms.

9.2 Intent Detection

Although the intent detection algorithm we describe here is similar to the intent detection in the previous chapter, in this chapter, we now aim to predict three possible user intentions rather than two: to open the hand (*Intent = Open*), to close the hand (*Intent = Closed*), and to relax (*Intent = Relaxed*). The addition of the *Intent = Relaxed* class allows the user to open the hand using the exotendon device, and then relax their hand while they are positioning their arm, for example in order to execute a pick and place task, without having to continue to exert effort to keep the hand open. This additional class helps reduce user fatigue.

In this way, our classifier is now denoted as:

$$CLAS(f_t) = [p_t^R, p_t^O, p_t^C] \quad (9.1)$$

where p_t^R represents the probability that the subject’s intent is to relax, p_t^O represents the probability

that the subject's intent is to open, and p_i^C represents the probability that the subject's intent is to close their hand. We note that $p_i^R, p_i^O, p_i^C \in [0, 1]$ and $p_i^R + p_i^O + p_i^C = 1$.

The output processing in this chapter is very similar to the previous chapter: we still apply a median filter to the output probabilities of the classifier to eliminate spurious predictions, but we use a 0.25 second filter instead of a 0.5 second filter:

$$\hat{p}_{\hat{T}}^R = \text{MEDIAN}(p_i^R), t \in [\hat{T} - 0.25s, \hat{T}] \quad (9.2)$$

$$\hat{p}_{\hat{T}}^O = \text{MEDIAN}(p_i^O), t \in [\hat{T} - 0.25s, \hat{T}] \quad (9.3)$$

$$\hat{p}_{\hat{T}}^C = \text{MEDIAN}(p_i^C), t \in [\hat{T} - 0.25s, \hat{T}] \quad (9.4)$$

We note that generally $\hat{p}_{\hat{T}}^R + \hat{p}_{\hat{T}}^O + \hat{p}_{\hat{T}}^C \neq 1$.

As in the previous chapter, we compare each probability $\hat{p}_{\hat{T}}^R, \hat{p}_{\hat{T}}^O, \hat{p}_{\hat{T}}^C$ to a threshold set by the experimenter L^R, L^O, L^C , respectively. If none of the values exceed the threshold, the $Intent_t$ is set to the same value as the intent from the previous time step $Intent_{t-1}$.

With the addition of $Intent = Relax$, the control issues motor commands to the exotendon device as follows: if the control predicts that the user's intention is $Intent = Open$, the device is commanded to open (retract the tendon, thus extending the fingers). If the user's intention is $Intent = Closed$, the device is commanded to close (extend the tendon, thus allowing the user to flex the fingers). If the predicted user intention is $Intent = Relaxed$, we continue to send the previous motor command to the device.

9.3 Training with the Exotendon Device

In this chapter, we extended our training protocol to include examples where the subject's arm was in different positions (resting on the table and raised off the table) to compensate for concept drift caused by abnormal muscle coactivation. We also modify it to include our new *Relax* intent.

Compared to the previous chapter, when the subjects used the device for longer periods of time, we had to generate a larger training dataset with both the arm and the orthosis in many different

Device State	Subject Instruction		
	Open	Relax	Close
Tendon extended	O	R	C
Tendon retracting	O		C
Tendon retracted	O	R	C

Table 9.1: Training protocol and assigned labels. For each combination of instruction given to the subject and state of the exotendon device, the table shows the ground truth label $Intent_t^g$ assigned to EMG data. Training begins with the tendon extended and the subject asked to relax (top row, middle column) and proceeds in counter-clockwise fashion

states, in the hope that these examples would allow the classifier to generalize to concept drift throughout the training session.

In this chapter, we collect the *Intent* data for all three classes under three conditions: 1) with the arm resting on a table and the orthosis motor off, 2) with the arm raised above the table and the orthosis motor off, and 3) with the arm raised above the table and the orthosis motor on, actively moving the hand. Specifically for the third case, we show in Table 9.1 the commands given to the subject and the ground truth collected while the motor retracts and extends the tendons. This is similar to the training protocol in Section 8.3, but we have added an additional *Intent* class.

Our experience has shown that a classifier trained on data collected during this extended protocol can typically generalize to concept drift and correctly classify subject intent when the subject performs functional motions for an entire testing session.

9.4 Experiments

Eleven individuals with chronic stroke volunteered to participate in the study. All subjects gave informed written consent to participate and the protocol was approved by the Columbia University Medical Center Institutional Review Board. All sessions were performed under the supervision of an occupational or physical therapist.

We had two controls which the subjects could use during this study: a EMG control described in Section 9.2, and a shoulder harness control. The shoulder harness control uses contralateral

shoulder motions to control the orthosis: shoulder depression causes the orthosis to extend the fingers, and shoulder elevation (a shrug motion) allows the hand to close.

Each subject was screened for our EMG control. During the control screening, a classifier was trained with the subject's EMG signals, and the user was instructed on how to use the control. The user was asked to open, relax, and close their hand three times, once with the forearm on the table, and once with it raised above the table. If our EMG method was able to classify the user intention correctly under all conditions, and the user could maintain each signal for at least 2 seconds on every attempt, the user was assigned to the EMG group. Otherwise, the participant was assigned to the shoulder harness based control.

Only six of our eleven subjects qualified for our EMG control. The shoulder harness control is more robust than the EMG control, since it relies exclusively on the unimpaired side. However, it uses a control motion which is unrelated to hand opening, and therefore has more limited rehabilitative value. We believe that the potential advantages of ipsilateral EMG control (more intuitive motor commands, closing the loop on the affected side) outweigh the shoulder harness control's advantage of robustness, as long as EMG control is applicable.

In order to evaluate our active hand orthosis, we performed a clinical study aiming to quantify its performance as either a rehabilitative or assistive device. At the outset of the study, we collected baseline measurements where the subject performed a battery of tests, including the Fugl Meyer Upper Limb (FM) assessment [138], Action Research Arm Test (ARAT) [139], and Box and Block Test (BBT) [140].

During the study, each of our subjects underwent 12 hour long training sessions. In each training session, the subject participated in 30 minutes of active training, during which the subjects were educated on the device operation, demonstrated proficiency in the control without any objects, and then advanced to repetitive drills, which involved grasping an array of objects of various shapes, sizes, and densities (Fig. 9.2).

After the twelve training sessions, the subjects underwent two more testing sessions: the subject repeated the battery of tests unassisted (not wearing the orthosis) and they repeated the battery



Figure 9.2: Real world objects used for treatment.

of tests (with the exception of the FM) assisted, while wearing the orthosis as it provided active assistance.

The FM, ARAT, and BBT were administered in a randomized order to limit order effect. The FM was only performed at post-testing without robotic assistance because the FM assesses capacity of the arm primarily through gross motor tasks. We thus presumed that robotic assistance would have minimal influence on FM scores.

We compare the baseline to our post testing sessions in two experiments:

- **Rehabilitative Experiments:** In these experiments, we compare the baseline assessment to the unassisted tests performed after training. The subject does not wear the orthosis to complete any of the assessments during these experiments. This experiment is designed to assess motor recovery after robot-assisted training. We want to determine if prolonged training with the device and control results in functional improvement for the subject.
- **Assistive Experiments:** In these experiments, we compare the baseline assessment to the assisted tests performed after training. The subject does not wear the orthosis for the baseline tests, but does wear the orthosis to complete the post-training assessments. This experiment is designed to assess assistive capability of the control and the orthosis. We want to determine if wearing the orthosis can help the user complete functional tasks. The clinical assessments

are performed after 12 training sessions to give the subject time to learn how to use the device.

9.5 Results

For the FM, ARAT, and BBT tests, we report the mean gains from the post testing, compared to the baseline testing. FM and ARAT can be sub-scaled. The FM has two sub-scales: FM-proximal evaluates the shoulder and elbow, while FM-distal evaluates the wrist and hand. The ARAT has 4 sub-scales: grasp, grip, pinch, and gross movement. In addition to total scores, we were interested in evaluating grasp components, as those are our targeted areas of intervention.

The results we report will focus primarily on the EMG subjects, but we draw comparisons between the EMG-based control and the shoulder harness control in the discussion portion of this section.

Rehabilitative Experiments: The results from the rehabilitative experiments are shown in Table 9.2a. We report the mean gains from the unassisted post-testing, compared to the baseline testing.

In the FM assessment, the subjects showed a mean gain of 2.67 points but was within the range of standard error. To further understand the implications of the results, we analyze the FM into two subtests, FM-distal and FM-proximal. FM-distal, improved significantly with a mean gain of 3 points while significant improvement was not achieved in FM-proximal. In the ARAT, the EMG-control showed positive mean gains of 1.33 points but was within the range of standard error. BBT scores decreased in the EMG group.

Assistive Experiments: The results from the assistive experiments are shown in Table 9.2b. We report the mean gains from the assisted post-testing, compared to the baseline testing.

In the ARAT, subjects performed worse by 2.5 points, though there was some improvement in the Grasp subscale. BBT scores also decreased by 4.3 points.

Assessment	EMG group (n=6)
FM-Distal	3.00 ± 0.82
FM-Proximal	-0.33 ± 1.48
FM-Total	2.67 ± 1.94
ARAT-Grasp	0.67 ± 0.56
ARAT-Grip	0.33 ± 0.33
ARAT-Pinch	-0.17 ± 0.31
ARAT-Gross Movement	0.5 ± 0.5
ARAT-Total	1.33 ± 0.95
BBT	-1.83 ± 1.76

(a) Rehabilitative results: post-training unassisted (without the device) compared to pre-training assessments

Assessment	EMG group (n=6)
ARAT-Grasp	1.17 ± 0.60
ARAT-Grip	-1.83 ± 0.75
ARAT-Pinch	-1.83 ± 0.79
ARAT-Gross Movement	0.0 ± 0.36
ARAT-Total	-2.50 ± 1.54
BBT	-4.33 ± 3.86

(b) Assistive results: post-training assisted (with the device) compared to pre-training assessments

Table 9.2: Rehabilitative and assistive results, showing mean gains and standard error between pre- and post-intervention. Bold results indicate gains (or losses) outside the range of standard error

Discussion: From a rehabilitation perspective, we discuss results obtained post-intervention without using the device, compared to baseline performance. Positive gains noted on the FM suggest that training with the device can serve as a rehabilitative tool to remediate some functional use in the affected hand, especially for participants with some degree of baseline functionality. The gains were attained in the FM-distal subtest, indicating that our functional robotic treatment was beneficial in improving hand functionality, as intended.

Based on the observation of a positive trend on the ARAT, we posit that increasing the intensity and duration of the intervention in future studies may lead to increased gains quantified using this measure. For example, small gains captured on the FM (e.g. ability to actively flex or extend the fingers) may not be captured on the ARAT because the improvement in range of motion was not sufficient to translate into increased functional ability (e.g. ability to pick up a small object).

From an assistive perspective, our results do not show that our EMG control is effective as an assistive device. We hypothesize that the negative mean gains seen in under these experimental conditions result from two factors: first, EMG participants tend to have residual functionality at baseline, often employing compensatory patterns to grasp and/or pinch objects. When wearing the robotic device, the bulky finger components may have impaired their performance by interfering

with these compensatory patterns. Second, in both the BBT and the ARAT, subjects are required to lift their arm above their shoulder. Neither the training sessions nor the training protocol contained an action that involved lifting an object this high. Concept drift triggered by the subject's abnormal muscle coactivation likely degraded the performance of the classifier in these cases.

Compared to the shoulder harness subjects, the EMG subjects had higher baseline performance. The shoulder harness subjects also regained some functional use in the affected hand - their FM-proximal scores increased by 1.40 points (± 0.51). This is only half of the improvement gained by the EMG subjects, suggesting that EMG-based controls have more rehabilitative value. We hypothesize this is because EMG-based controls require the users to exercise the muscles from the affected limb to engage the orthosis. However, the shoulder harness proved to be a better assistive control. During assisted testing, the EMG control resulted in significant decreases in score for the BBT, and for the total ARAT score. On the other hand, the shoulder harness resulted in significant improvement in the total ARAT score (it added 2.20 ± 1.20 points to the baseline). This suggests that for our EMG control to be effective as an assistive device, it needs to be as robust as our shoulder harness control.

Overall, our results indicate that our control can be used in conjunction with an orthosis to provide effective rehabilitation which improves distal motor function in chronic stroke subjects. In its current form, our control does not work well as an assistive device. We hypothesize that part of the reason for this is concept drift, and will need to do further studies to see if our control can overcome this.

9.6 Summary

The study in this chapter provides further validation for the method we described in the previous chapter. We propose an updated intent detection with an additional *Intent* class, and an updated training protocol which provides examples of the arm in more positions, designed to make the control more robust to subject fatigue and to concept drift. We compare clinical outcomes before and after the subject undergoes a 12 session training period.

Our results suggest that the EMG control we describe, when paired with an exotendon device, and used over a period of 12 training sessions, can provide rehabilitative assistance for chronic stroke patients. However, we found that the control does not currently enable the device for assistive purposes. The results of this study should be cautiously interpreted because of our limited sample size. In future work, we are planning to further explore if the device can be used for assistive purposes if we make our control more robust to concept drift and reduce the profile of the fingertip components.

We believe that this work highlights the potential benefits of wearable and user-driven robotic hand orthoses. These devices can enable hand rehabilitation during daily activities (as opposed to isolated hand exercises with limited upper limb engagement) over extended periods of time. In the long term, we would like to enable rehabilitation in a home environment as well.

However, this study also highlighted limitations of our EMG control. In particular, compared to the shoulder harness control, the EMG control did not perform well as an assistive device. We hypothesize that a lack of robustness to concept drift prevented our device from performing well in this context. It also meant that we had to train the subject at the beginning of each of the 12 testing sessions, which placed a large burden on the user. In the next chapter, we propose semi-supervised learning algorithms we hypothesize will make our EMG control more robust to concept drift.

Chapter 10: Semi-Supervised EMG Control for Wearable Hand Orthotics

The EMG control we propose in Chapter 8 can be used in conjunction with an underactuated orthosis device to enable functional movement in stroke subjects and even provide functional gains to this population. However, our experience in the clinical study from Chapter 9 also highlighted limitations of this control: our control cannot act as an assistive device because it is not sufficiently robust to concept drift, and the process of training our intent detection places a substantial burden on the user, primarily because of the training protocol we develop to compensate for concept drift.

Concept drift can occur both within a single session (intrasession drift) and across sessions (intersession drift). We encourage the reader to review Chapter 7 for a more complete overview of concept drift and its causes.

In the previous chapter, we compensated for intrasession drift caused by the hand's physical interaction with the orthosis with our training protocol, during which we provided the classifier with labeled data for all possible tendon states of the orthosis and from different arm positions. We had to generate a large training dataset with the arm and the orthosis in many different states; these examples allow the classifier to generalize to concept drift throughout the training session. We also ignored intersession concept drift. Retraining the subject at the beginning of every testing session removed the need to compensate for the drift caused by repositioning the EMG sensors.

Even with these measures, our experiments showed our control needed to be more robust to concept drift to be effective as an assistive device. Additionally, although performing the entire training protocol at the beginning of every session takes less than five minutes, when the entire session is an hour, this five minutes is a meaningful period during which the subject could otherwise be working on their rehabilitation. Reducing the training burden on the user is important for stroke subjects because they fatigue quickly [141]. We would prefer to spend more time before they fatigue on rehabilitation, rather than on training the control. If this control was used in conjunction

with a take-home orthosis, then this repeated and multi-step training protocol would be burdensome to the user.

There are two ways in which we could reduce the training burden for the user, while simultaneously making the control robust to concept drift. First, we could use a shorter training protocol. In this case, our control would have to compensate for intrasession concept drift. Second, we could use the same classifier across different days and different training sessions. In this case, our control would have to compensate for intersession concept drift.

Semi-supervised learning provides a potential solution to concept drift, and therefore to the training burden placed on the subjects by our EMG control from the previous chapter. Semi-supervised learning uses a small labeled data set and then exploits additional unlabeled data in order to improve classifier performance. These algorithms have been shown to improve classification accuracy when concept drift occurs [95, 96, 97], and we hypothesize they will require less training data from the subject than supervised controls, while simultaneously maintaining high accuracies.

In this chapter, we propose two semi-supervised learning algorithms to help our orthosis control adapt to concept drift. We propose a disagreement-based algorithm for intrasession drift (i.e. when the subject changes their arm position), and a K-means clustering algorithm for intersession drift where the data has shifted radically in the feature space (i.e. when the subject has doffed and donned the device).

In offline experiments, we show that a disagreement-based semi-supervised intent detection allows the subject to use an abbreviated version of our training protocol, while maintaining high classification accuracies. We also show that a K-means semi-supervised intent detection allows us to re-use a classifier trained in a previous training session at a later date.

To our knowledge, *we are the first to propose a semi-supervised control for a hand orthosis.* We introduce a novel K-means semi-supervised algorithm and apply disagreement-based semi-supervision to a novel application. We show that semi-supervised controls can reduce the burden placed on the subject during training for ipsilateral hand controls.

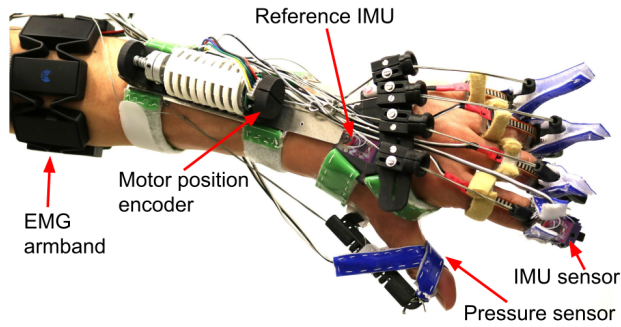


Figure 10.1: Updated hand orthosis prototype in extension configuration with multimodal sensing suite.

This study is exploratory in nature. It is intended to inform future directions of orthotic controls. We do not always achieve the same accuracies as we could with a more extended training procedure or if we were to retrain the subject at the beginning of every testing session; however, we show that semi-supervised, adaptive controls can reduce the training burden placed upon the subjects.

The 2020 pandemic meant that we were not able to bring in more subjects for testing. Despite our small sample size, our experiments indicate that semi-supervised algorithms merit further investigation as potential controls for wearable orthoses.

10.1 Hardware

In the previous chapter, we described an updated prototype of our underactuated exotendon device. In this chapter, we use the same orthosis and EMG sensing, and add three new sensing modalities. Figure 10.1 shows the exotendon device and the multimodal sensing suite. We briefly review the updated hardware and the multimodal sensing suite below.

Exotendon Device This is the same exotendon orthosis used in the previous chapter.

Forearm EMG We use the same eight sensor Myo Armband from the previous chapter.

Motor Position A motor encoder provides position feedback, allowing us to both control the actuator and determine the state of the orthosis. Because the tendons underactuate the hand, this

does not provide information about individual fingers, but about the state of the hand as a whole.

Finger Joint Angles Some subjects retain partial voluntary movement of their fingers, even if they do not have full functional extension. We use two commercially available inertial measurement units (IMUs) (Zio 9DOF IMU BNO080) to measure the joint angles of one of the subject's fingers. One IMU is placed on a fingertip component of the device (usually the pointer, as the subject often has the most voluntary use of this finger), and another reference IMU is placed on the splint of the exotendon device. The difference in orientation between the IMU on the hand and the fingertip IMU measures the aggregate joint angle of the subject's metacarpophalangeal joint (MCP) and proximal interphalangeal joint (PIP). We use the time derivative of the finger's joint angles to determine when the subject initiates voluntary motion. We note that this sensor will also measure when the motor retracts or extends the exotendon network.

Fingertip Pressure The exotendon device is also fitted with a pressure sensor. This pressure sensor allows us to measure both the level of force between the device and the hand, and the level of force between the hand and an object during grasping. This sensing modality is provided by a force sensitive resistor (FSR) mounted inside the Velcro strap of the thumb fingertip component. The thumb is used in all gross grasping, which is why we choose to locate the sensor here. Like for the joint angles, we use the time derivative of the pressure sensor.

10.2 Intent Detection

Although the intent detection algorithm we describe here is similar to the intent detection in Chapter 9, in this chapter, there are three important differences:

- We use linear discriminant analysis (LDA) classifiers instead of RF classifiers. LDAs have similar classification accuracies to RF classifiers, but they are faster to train and are much simpler to update online.
- We allow for the possibility of multiple base learners in our classifier. In the previous chapter,

our classifier had a single base learner. In this chapter, we formulate our intent detection so that the classifier could have one or more base learners. When a classifier has more than one base learner, it is known as an ensemble classifier.

- We allow for the possibility that our classifier features will include more sensing modalities than EMG.

Given these differences, we will briefly reformulate our intent detection algorithm below.

We collect the raw EMG data ($e_1 \dots e_8$), the motor position pos , and the time derivatives of the joint Δj and pressure $\Delta pres$ data at a rate of 100Hz. At time t , we collect the sensor data into a data vector \mathbf{x}_t :

$$\mathbf{x}_t = (e_1^t \dots e_8^t, pos_t, \Delta j_t, \Delta pres_t). \quad (10.1)$$

In this chapter, \mathbf{x} includes EMG and all other sensing modalities.

Our intent detection is an ensemble classifier with η base learners:

$$CLAS^{ens}(\mathbf{x}_t) = \{CLAS^1(\mathbf{f}_t^1), CLAS^2(\mathbf{f}_t^2), \dots, CLAS^\eta(\mathbf{f}_t^\eta)\} \quad (10.2)$$

where \mathbf{f}_t^i represents the features for the base classifier i . \mathbf{f}_t^i may equal \mathbf{x}_t , or it may be a subset of features in \mathbf{x}_t . Each base classifier i outputs a set of probabilities:

$$CLAS^i(\mathbf{f}_t^i) = \Phi_t^i \quad (10.3)$$

$$\Phi_t^i = [p_t^{R_i}, p_t^{O_i}, p_t^{C_i}] \quad (10.4)$$

where p_t^R represents the probability that the subject's intent is to relax, p_t^O represents the probability that the subject's intent is to open, and p_t^C represents the probability that the subject's intent is to close their hand. We note that $p_t^R, p_t^O, p_t^C \in [0, 1]$ and $p_t^R + p_t^O + p_t^C = 1$.

As such, $CLAS^{ens}(\mathbf{x}_t)$ will output a set of probabilities Φ_t^{ens} which consist of the probabilities

predicted by each of the base learners in the ensemble.

$$\Phi_t^{ens} = \{\Phi_t^1, \Phi_t^2, \dots, \Phi_t^\eta\} \quad (10.5)$$

The output processing in this chapter is very similar to the previous chapter - we apply a median filter for each Φ^i in Φ^{ens} :

$$\hat{\Phi}_t^i = \text{MEDIAN}(\Phi_t^i), t \in [\hat{T} - 0.25s, \hat{T}]. \quad (10.6)$$

We then average the filtered $\hat{\Phi}$ values:

$$\bar{\Phi}_{\hat{T}}^{ens} = \frac{1}{\eta} * \sum_{i=1}^{\eta} \hat{\Phi}_{\hat{T}}^i. \quad (10.7)$$

Preliminary experiments showed that performing the median filter before averaging the Φ_t^i values provided higher classification accuracies than averaging before filtering.

As in the previous chapter, we compare each probability in $\bar{\Phi}_{\hat{T}}^{ens}$ to a threshold set by the experimenter L^R, L^O, L^C , respectively. If none of the values exceed the threshold, the $Intent_t$ is set to the same value as the intent from the previous time step $Intent_{t-1}$.

The motor commands are issued based on the subject $Intent$ in the same manner as for the previous chapter.

We note that in this formulation, the EMG control described in the previous two chapters would be a classifier with $\eta = 1$ and $f^1(t) = (e_t^1, \dots, e_t^8)$.

10.3 Semi-Supervised Learning for Intent Detection

Now that we have reformulated our intent detection, we discuss how to use a semi-supervised learning algorithm to improve our classification accuracies.

Semi-supervised learning exploits unlabeled data to modify our intent detection classifier. As data arrives during a testing session, semi-supervised learning adapts the classifier parameters to

fit the new data. Since the classifier adapts as the data arrives, semi-supervised learning can make our intent detection more robust to concept drift caused by fatigue, subject movement, and donning/doffing of the orthosis.

For any semi-supervised learning algorithm, there are two components: an oracle, and an updater. The oracle looks at incoming unlabeled sensor data, determines which datapoints should be used to update the classifier and then labels the data. The data points which are labeled and will be used to update the classifier are assembled into a training dataset $X_{training}$. In our case, since we are using an ensemble classifier, we will generate a training dataset $X_{training}^i$ for every base learner i in the ensemble. The updater uses the training data and labels from the oracle to update the classifier parameters.

We note that semi-supervised learning is not a stable algorithm. If the classifier is updated with incorrectly labeled data, then the performance of the control will degrade. We therefore tend to be conservative about what update data we provide to the updater.

We would like our semi-supervised learning algorithm to address two kinds of concept drift - intrasession and intersession concept drift. Intrasession concept drift tends to be a gradual shift in the data over time, whereas intersession concept drift is a sudden, often large, redistribution of the data in feature space (principally caused by donning and doffing the device). As such, the semi-supervised learning algorithms which work well for intrasession concept drift may not be appropriate for intersession concept drift, and visa versa.

We propose two semi-supervised learning algorithms: a disagreement based semi-supervision, which we hypothesize will adapt well to intrasession concept drift, and a K-means based semi-supervision, which we hypothesize will allow the classifier to adapt to intersession concept drift.

10.3.1 Disagreement Based Oracle

The first semi-supervised learning algorithm we present uses a disagreement based oracle to label data from the sensors. We hypothesize this oracle will be useful at adapting to intrasession concept drift.

When concept drift occurs gradually, as for intrasession testing, we can leverage the disagreement between multiple learners to supervise our data. The different base learners in our ensemble collaborate to label incoming data from the sensors. Because the change is gradual, we assume that when concept drift begins, at least some of the base learners will be robust to the small changes in the data and will remain confident in their intent prediction. Other learners which are not robust to the drift can be corrected by the confident learners.

Our proposed disagreement based semi-supervision is enacted as follows: at time t , we calculate the entropy E of each base classifier i . We use entropy as a measure of the classifier's confidence:

$$E^i = -\log_k(\Phi^i \cdot \Phi^{i\top}) \quad (10.8)$$

where k is the number of possible *Intent* classes for the classifier. Lower entropy indicates that the predictor is more confident.

Base learners which are confident in their prediction are used to correct and label data for less confident predictors. Specifically, we define confident predictors as base learners i where $E^i < 0.2$. Learners which are not confident have $E^i > 0.8$. We select 0.2 and 0.8 as our confidence thresholds empirically.

If all confident classifiers agree on the subject's intent, i.e. if $\operatorname{argmax}(\Phi^i)$ is the same for all the confident predictors, then we use the data from time point t to update the unconfident predictors. For each unconfident predictor i , from the sensor data \mathbf{x}_t , we select the subset of data that corresponds to the features of base learner f_t^i . This data is added to the base learner's training dataset $X_{training}^i$, along with the *Intent* label which was agreed upon by the confident predictors.

We note two important details about this algorithm: first, it does not matter what *Intent* prediction the unconfident predictors generate. Even if they generated the same prediction as the confident predictors, the training data will still be added to $X_{training}^i$ to reinforce their prediction. Second, the correction data will not be evenly distributed across all base learners. If a learner is robust to concept drift and therefore always confident, it may never be updated.

Once the combined number of training points across all $X_{training}^i$ datasets is a sufficiently large

value (we chose this value to be 200 datapoints), the datasets and the accompanying labels are given to the LDA updater, all $X_{training}^i$ datasets are reset, and the supervision process restarts. The number of datapoints required before an update is defined empirically; we set it to a relatively low number, since we wanted the updater to continuously adapt to concept drift.

Oracles, though primarily used to label datapoints for the updater, can also correct the classifier predictions (before any modifications have been made to the parameters of the base learners). In the case of disagreement based supervision, we only want confident predictors to contribute to the final output of the ensemble - if we are going to correct an unconfident base learner, then we do not want it to contribute to our overall prediction. However, in this case, only including confident predictors where $E_i < 0.2$ is too restrictive. We therefore have an additional entropy threshold for the correction, which we set to 0.6. We calculate the final prediction probability for the ensemble as an average of all the probabilities from base learners whose entropy is less than 0.6:

$$\bar{\Phi}_{\hat{T}}^{ens} = \frac{1}{\eta} * \sum_{\substack{i=1 \\ E_i < 0.6}}^n \hat{\Phi}_{\hat{T}}^i \quad (10.9)$$

If there are no base learners where $E_i < 0.6$ then $\bar{\Phi}_{\hat{T}}^{ens}$ is calculated using all of the base learners in the ensemble.

This disagreement based supervision never generates any additional training data if there is only one base learner in our intent detection (when $\eta = 1$). It works best if the base learners include multiple independent views [106], or a large number of base learners [142]. Therefore, in our experiments that use disagreement based semi-supervision, we use ensemble learners with at least five base learners whose features are sampled from all the sensors in the multimodal sensing suite.

10.3.2 K-means Oracle

The second semi-supervised learning algorithm we propose uses K-means clustering to label the incoming data. With intersession concept drift, i.e. when the orthosis is taken off and then put

back on for a new training session, the sensor data will shift suddenly in the feature space. In this case, a disagreement oracle may not be sufficient because we can no longer count on a majority, or even any of the ensemble base learners being correct. We therefore propose a K-means semi-supervision algorithm to compensate for intersession concept drift.

K-means is an unsupervised clustering algorithm, so we do not require the classifier from the previous session to classify any of the new data points correctly. If we can cluster the data to find the different *Intent* classes and then assign the correct intent to the correct classes, we can use semi-supervised learning to adapt the classifier to the shift in the data.

Despite the sensor’s data sudden shift in feature space, we still assume that 1) the three different *Intent* classes are still differentiable and 2) that the subject will perform all three intent motions at least once in the first 40 seconds of a new training session. The first assumption is a requirement for a LDA classifier to work and we can instruct the subject to fulfill the second assumption.

The algorithm for our K-means supervision is as follows: for each base learner i in the ensemble, we collect the first 40 seconds of sensor data during the testing session $X_{training}^i = \mathbf{f}_t^i, t \in [0s, 40s]$. We then use K-means to cluster each $X_{training}^i$ dataset. We specify that the K-means algorithm should look for three clusters, the same number of *Intent* classes that our LDA classifier has.

For each base learner, K-means computes three cluster centers $\mathbf{c}_1^i, \mathbf{c}_2^i, \mathbf{c}_3^i$ and predicts which cluster each sample in $X_{training}^i$ belongs to.

We assume that each of the clusters found by K-means corresponds to an *Intent* in the classifier; however, we do not know which cluster corresponds to which *Intent*. To assign each of the clusters to an *Intent*, we calculate the distance from each cluster center to each class mean in the base classifier i :

$$d_{j,k} = \|\mathbf{c}_j^i - \boldsymbol{\mu}_k^i\| \quad (10.10)$$

where j is the cluster index and k is the *Intent* class. We take the minimum $d_{j,k}$ and assign cluster j to *Intent* k . We then take the next smallest $d_{\hat{j},\hat{k}}$ where \hat{j} and \hat{k} do not belong to the cluster or *Intent* which have already been paired together. Finally, the remaining cluster and *Intent* are

paired together.

This process of assigning each cluster to an *Intent* assumes that even when the data shifts in the feature space between sessions, the data belonging to an *Intent* class in the new session will still be closer to data from the same *Intent* in the previous session than to data from any of the other two *Intent* classes from the previous session. If this assumption holds, K-means can improve the performance of our classifier.

We note that since the base learners will have different features, the features f_t^i which come from a single time point may not be assigned the same *Intent*, despite being subsets of the same x_t . Despite this, our experiments show that K-means semi-supervision can improve the performance of our control.

Since the K-means algorithm already assigned each sample in $X_{training}^i$ to a cluster, we now have sensor data and labels with which we can update our intent detection classifier. Our K-means algorithm only updates the base learners once, unlike the disagreement based algorithm, which updates the base learners throughout the testing session.

10.3.3 LDA Updater

Since all of our base classifiers are LDA learners, this updater can be used with either oracle described above.

Once the oracle has collected a training dataset $X_{training}^i$ and labeled the data for each base learner i , we need to update each base learner in the ensemble with this newly labeled data.

We have chosen LDA classifiers as our base learners because we do not need to past training data when updating an LDA. An LDA base learner i with k classes has $k + 1$ main parameters: a mean vector for each class μ_k^i and a covariance matrix Σ^i . Therefore, an LDA ensemble has $\eta(k + 1)$ parameters.

To update our ensemble intent detection classifier, we will update each of the base learners i independently using the training dataset $X_{training}^i$ which was collected specifically for that LDA.

To update the parameters of an LDA base learner i , let z_k be a sample from $X_{training}^i$ which is

labeled as class k . The updated mean vector for class k $\tilde{\boldsymbol{\mu}}_k$ and the updated covariance matrix $\tilde{\boldsymbol{\Sigma}}$ for the LDA are calculated as follows:

$$\tilde{\boldsymbol{\mu}}_k^i = \frac{n_k^i * \boldsymbol{\mu}_k^i + \mathbf{z}_k}{n_k^i + 1} \quad (10.11)$$

$$\tilde{\boldsymbol{\Sigma}}^i = \frac{N^i}{N^i + 1} \boldsymbol{\Sigma}^i + \frac{1}{N^i + 1} * \frac{n_k^i}{n_k^i + 1} (\mathbf{z}_k - \boldsymbol{\mu}_k^i)(\mathbf{z}_k - \boldsymbol{\mu}_k^i)^\top \quad (10.12)$$

where N^i is the number of training data points for the base classifier i to date, and n_k^i is the number of training datapoints for the base classifier labeled as class k to date.

10.4 Experiments

Our experiments were performed with two chronic stroke survivors with a spasticity level between 0 and 1 on the Modified Ashworth Scale (MAS). Testing was approved by the Columbia University Internal Review Board, and performed in a clinical setting under the supervision of Physical and/or Occupational Therapists.

In our experiments we wish to investigate two questions:

1. Can semi-supervised controls compensate for intersession concept drift? This would reduce the time required to train the classifier since the subject would not have to provide training examples for all possible arm positions and all possible states of the orthosis.
2. Can semi-supervised controls compensate for intrasession concept drift? This would allow us to reuse a classifier trained at an earlier date and the subject could start rehabilitation immediately at the start of a session.

We hypothesize that semi-supervised controls will allow us to compensate for both intersession and intrasession concept drift while maintaining high accuracies.

Since this is a preliminary study to determine the feasibility of semi-supervised controls for orthoses, in each experiment, we perform offline analysis of sensor data collected during training

sessions.

Intersession Experiments In our intersession experiments, we test whether a classifier trained on ipsilateral hand sensor data can be used at a later date without additional labeled training data.

We collect subject data on two separate occasions. The first data collection occurred over a year before the second data collection. During the first session, we collect a single set of data using the training protocol from Section 9.3. During the second session, we collect four sets of data using the same training protocol - one will be used to train an ‘optimal’ control, and the other three will be used as testing datasets.

We train four classifiers:

- **A Supervised EMG control:** this classifier is trained on data from the first data collection session. The ensemble parameters are $\eta = 1$ and $f^1 = (e^1, \dots, e^8)$. Once trained, the parameters of this classifier do not change.
- **A K-means Semi-Supervised EMG control:** initially, this classifier is the same as the supervised EMG control. However, as sensor data from the testing dataset arrives, the parameters of this classifier will be updated using a K-means oracle.
- **A Supervised Same Day EMG control:** this is the only classifier trained on data from the second session. Our experience shows that classifiers trained on the same day have high accuracy, so we use this classifier as an ‘optimal’ example. The ensemble parameters are $\eta = 1$ and $f^1 = (e^1, \dots, e^8)$. Once trained, the parameters of this classifier do not change.

We then use each of the classifiers from above to predict the subjects’ *Intent* for each of the three testing datasets from the second session. We compare the predicted *Intent* to the ground truth *Intent* collected by the experimenter during the session.

Intrasession Experiments In our intrasession experiments, we test whether a classifier trained on ipsilateral hand sensor data can be trained using less data and still achieve high accuracies during *Intent* detection.

In these experiments, all the data is collected during the same session. We collect one dataset using an abbreviated collection protocol. For the abbreviated training protocol, we only collect data using the third condition from our training protocol in Section 9.3: with the arm raised above the table and the orthosis motor on, actively moving the hand. We collect four other datasets using the entire collection protocol - one will be used to train an ‘optimal’ control, and the other three will be used as testing datasets.

We train four classifiers:

- **A Supervised EMG control:** this classifier is trained on data from the abbreviated dataset. The ensemble parameters are $\eta = 1$ and $f^1 = (e^1, \dots, e^8)$. Once trained, the parameters of this classifier do not change.
- **A Supervised Multimodal control:** this classifier is trained on data from the abbreviated dataset. The η is selected as a random number between 5 and 10 and the features for each base learners are selected randomly from \mathbf{x} . Once trained, the parameters of this classifier do not change. We include this control as an additional baseline to explore if improvement in performance is caused by the multimodal nature of the control or by the semi-supervised learning.
- **A Disagreement Semi-Supervised Multimodal control:** initially, this classifier is the same as the supervised multimodal control. However, as sensor data from the testing dataset arrives, the parameters of this classifier will be updated using our disagreement based semi-supervised learning algorithm. We use a multimodal control for this kind of semi-supervision because disagreement base algorithms work best when the data has multiple independent views. Preliminary experiments indicated that using an ensemble classifier with just EMG signals as features did not perform as well as a multimodal ensemble.
- **A Supervised Entire Protocol EMG control:** this is the only classifier trained on data from the entire collection protocol. Our experience shows that classifiers trained under the same conditions have high accuracies, so we include this as an example of ‘optimal’ performance.

The ensemble parameters are $\eta = 1$ and $f^1 = (e^1, \dots, e^8)$. Once trained, the parameters of this classifier do not change.

We use each of the classifiers from above to predict the subjects' *Intent* for each of the three testing datasets collected with the entire protocol. We compare the predicted *Intent* to the ground truth *Intent* collected by the experimenter during the session.

We briefly note that, although the multimodal classifier has features which are selected randomly, in our experience, this does not significantly affect ensemble performance. When we tested three different ensembles whose features were generated randomly, their motor command accuracy only differed by, on average, 2.5%.

10.5 Results

In each set of experiments, we report the results as an average across the 3 testing datasets. At each time point, the control can correctly predict an open (true positive), correctly predict a close (true negative), incorrectly predict an open (false positive) or incorrectly predict a close (false negative) for the motor command. We report the motor command accuracy, the positive predictive value (PPV), and the negative predictive value (NPV) for our classifiers [143]. Motor command accuracy is the number time points at which the command to the motor was correct, divided by the number of total time points. We report motor command accuracy instead of global accuracy (a measure of how often the classifier gets each class correct) because we care less about whether the classifier has predicted the subject *Intent* as 'relaxed' or 'open', and more about whether the motor is moving the hand as the subject intends. PPV is the number of true positives divided by the number of all positive predictions (whether true or false). NPV is the number of true negatives divided by the number of all negative predictions.

We use motor command accuracy as our primary performance metric, since this measures what percent of the time the control operates the orthosis as the subject intended.

Table 10.1: Intersession results

Control		Motor		
		Command Accuracy	NPV	PPV
Subject 1	<i>Supervised Same Day EMG</i>	72.8 ± 8.1	69.9 ± 7.6	89.0 ± 1.8
	Supervised EMG	57.3 ± 5.6	58.0 ± 5.0	61.0 ± 31.3
	K-means Semi-Supervised EMG	53.1 ± 17.4	60.3 ± 19.2	41.8 ± 18.2
Subject 2	<i>Supervised Same Day EMG</i>	83.6 ± 6.5	91.0 ± 5.5	77.4 ± 7.6
	Supervised EMG	54.5 ± 1.7	54.2 ± 1.8	27.2 ± 47.1
	K-means Semi-Supervised EMG	62.9 ± 6.8	63.1 ± 6.9	69.8 ± 13.7

Table 10.2: Intrasession results

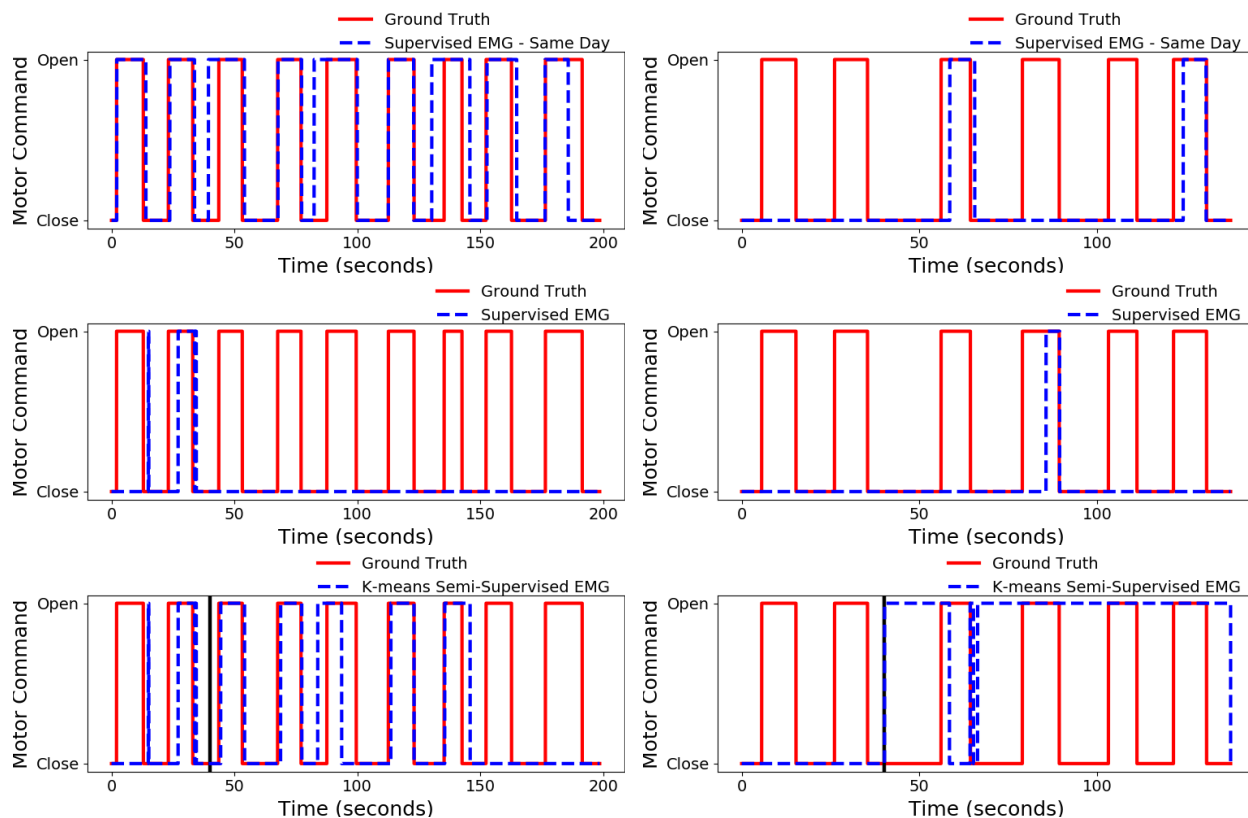
Control		Motor		
		Command Accuracy	NPV	PPV
Subject 1	<i>Supervised Entire Protocol EMG</i>	72.8 ± 8.1	69.9 ± 7.6	89.0 ± 1.8
	Supervised EMG	72.1 ± 4.6	72.0 ± 6.5	76.2 ± 10.2
	Supervised Multimodal	72.7 ± 5.7	69.4 ± 4.6	87.2 ± 2.6
	Disagreement Semi-Supervised Multimodal	79.2 ± 4.4	76.4 ± 4.2	86.8 ± 4.8
Subject 2	<i>Supervised Entire Protocol EMG</i>	85.6 ± 4.6	91.3 ± 5.0	80.5 ± 4.7
	Supervised EMG	81.9 ± 10.3	91.1 ± 4.1	75.6 ± 13.0
	Supervised Multimodal	80.3 ± 5.4	75.6 ± 6.1	91.1 ± 1.2
	Disagreement Semi-Supervised Multimodal	85.8 ± 4.0	87.0 ± 1.9	85.1 ± 9.7

Intersession Experiments Table 10.1 shows the average motor command accuracy, PPV, and NPV across the three testing datasets for each subject. We italicize the output of the ‘optimal’ classifier, and bold the best performing, ‘non-optimal’ classifier for each metric.

For Subject 1, K-means semi-supervision improved the NPV of the supervised EMG classifier, however it lowered the motor command accuracy and the PPV. The motor command accuracy decreased, on average, by 4.2%. Neither the K-means semi-supervised classifier, nor the supervised EMG classifier performed as well as the ‘optimal’ same day supervised EMG classifier.

For Subject 2, K-means semi-supervision improved the motor command accuracy, the NPV, and the PPV of the EMG classifier. The motor command accuracy increased, on average, by 8.4%. Even though it outperformed the supervised EMG classifier, the K-means semi-supervised classifier did not perform as well as the ‘optimal’ same day supervised EMG classifier.

Figure 10.2a shows the output from the three different classifiers for a single example test



(a) Stable Semi-Supervised Learning

(b) Unstable Semi-Supervised Learning

Figure 10.2: Example of both stable and unstable intersession intent prediction. The motor command predicted by the classifier is blue and the ground truth motor command is red. Plot labels indicate the type of classifier used, as defined in Section 10.4. A vertical black line denotes the classifier was updated at that time step.

dataset. Since we have stated that semi-supervised learning is not a stable algorithm, and since this is a preliminary experiment designed to inform the future of orthotic controls, we also provide Figure 10.2b, an example where the performance of the classifier was made worse by our K-means semi-supervised algorithm.

Intrasession Experiments Table 10.2 shows the average motor command accuracy, PPV, and NPV across the three testing datasets for each subject. We italicize the output of the ‘optimal’ classifier, and bold the best performing, ‘non-optimal’ classifier for each metric.

For Subject 1, the disagreement semi-supervised classifier outperformed all three of the other classifiers, including the ‘optimal’ classifier by at least 6.4% for the motor command and 4.4% for

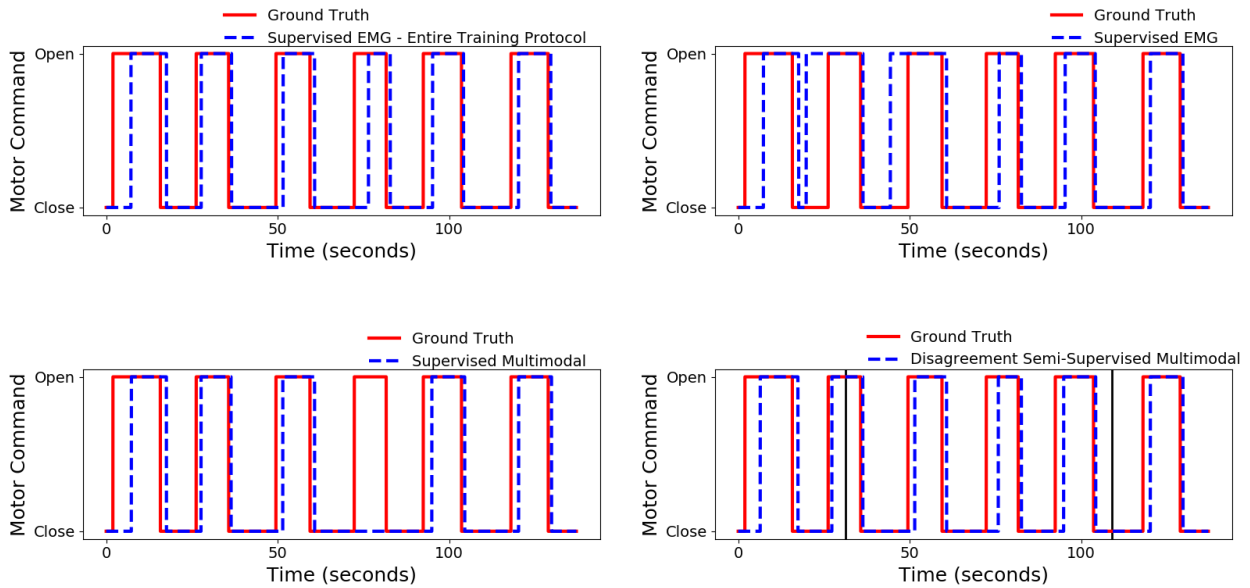


Figure 10.3: Example of stable intrasession intent prediction. The motor command predicted by the classifier is blue and the ground truth motor command is red. Plot labels indicate the type of classifier used, as defined in Section 10.4. A vertical black line denotes the classifier was updated at that time step.

the NPV. It had lower PPV than the ‘optimal’ supervised classifier and the supervised multimodal classifier.

For Subject 2, the disagreement semi-supervised multimodal classifier had the highest motor command accuracy of all the classifiers, including the ‘optimal’ classifier. It also outperformed the ‘optimal’ classifier in PPV, though it did not outperform the supervised multimodal classifier. The disagreement semi-supervised multimodal classifier had lower NPV than the ‘optimal’ classifier and the supervised EMG classifier.

Figure 10.3 shows the output from the four different classifiers for a single example test dataset.

Discussion We begin by discussing the intersession results. In this experiment, we provide an ‘optimal’ classifier trained on data from the same session as the testing data, and a baseline classifier trained on data from a previous session.

The intersession experiments show that K-means semi-supervision improves classifier performance for one subject, but not for the other. It is unclear why our method only worked for one of

our two subjects. Further testing is needed with more subjects to see whether we can predict based on the subject's characteristics when K-means semi-supervised learning will work and when it will not.

The intersession experiments also show that, although it can improve the performance of a classifier trained on a different day, K-means semi-supervision does not perform as well as a classifier trained on data from the same session. These lower accuracies may be due to the fact that our K-means algorithm only updates the classifier once during the session. In future work, we would like to combine K-means with another supervision technique to continue updating the classifier throughout the session. Preliminary experiments combining K-means and our disagreement based oracle showed that this actually degraded performance, so more research must be done to explore the interaction between these algorithms.

For the example of unstable semi-supervised learning, which we provide in Figure 10.2b, we note that, empirically, the semi-supervised classifier does not lower performance. K-means supervision altered the intent detection to almost always detect an open signal, unlike the supervised EMG classifier which almost always predicted a close signal. Neither control is useful for the subject, even though the supervised EMG control nominally has higher accuracies.

Next, we discuss the intrasession experiments. In these experiments, we provide two baseline classifiers: a supervised EMG control and a supervised multimodal control, both trained on data collected using our abbreviated collection protocol. We also provide a classifier trained on data collected with the entire collection protocol to show the performance of an 'optimal' classifier.

The intrasession experiments show that, for both subjects, our disagreement-based semi-supervised multimodal classifier outperforms the 'optimal' classifier in motor command accuracy, despite being trained on less data. This indicates that semi-supervised learning algorithms can make our control more robust to concept drift, despite being trained on less data.

We compare our semi-supervised multimodal classifier to our supervised multimodal classifier, to see if our advantage over the 'optimal' classifier comes from the semi-supervised nature of the control or from the multimodal ensemble. For both subjects, we find that the semi-supervised

learner outperforms the static multimodal classifier in motor command accuracy and NPV, but not in PPV. You can also see in Figure 10.3 that the semi-supervised multimodal classifier identifies an open-close motion that the supervised multimodal classifier misses. For this reason, we conclude that it is the disagreement-based semi-supervision which provides the advantage of this classifier, not the fact that it leverages a multimodal ensemble.

For each subject, our disagreement-based semi-supervised learning outperform the ‘optimal’ classifier and the supervised multimodal classifier in either NPV or PPV, but not both. In future work, we would like to make our disagreement based classifier perform more consistently in these two metrics, while keeping the advantage it already gives in motor command accuracy.

Interestingly, in the intrasession experiments, the ‘optimal’ classifier results in approximately the same motor command accuracies as the supervised classifiers. This result does not match our experience from functional tests with subjects. In our experience, the ‘optimal’ classifier should outperform classifiers trained on more limited data. We hypothesize that the advantage of the ‘optimal’ classifier is less pronounced in these experiments because of the nature of our data collection. When we collected our datasets, subjects were told to raise their arm off the table, but could position the arm however they wanted, as long as it was raised. Subjects likely positioned their arm in a way which was least likely to fatigue them and which did not activate their abnormal muscle coactivation. In the future, we would like to evaluate our semi-supervised learning algorithms as subjects perform functional movements.

Overall, the experiments we perform in this chapter show that semi-supervised learning has the potential to reduce the burden placed on users by controls which require a labeled dataset to detect the subject *Intent*. Both of our proposed algorithms provide some improvement in motor command accuracy when the classifier is trained on a ‘non-optimal’ dataset. The K-means semi-supervised control worked better for Subject 2 than it did for Subject 1, which indicates that different supervision methods could help some subjects more than others. The disagreement based supervision worked for both subjects, and even outperformed the ‘optimal’ classifier in motor command accuracy.

10.6 Summary

In this chapter, we propose two semi-supervised learning algorithms. We hypothesize that semi-supervised learning will help our controls adapt to concept drift in the data, and could be used to reduce the training burden placed upon the user for orthotic EMG-based controls.

We propose a K-means semi-supervision algorithm and a disagreement based semi-supervised learning algorithm. In offline experiments, we show that K-means semi-supervision can compensate for intersession concept drift, though it does not outperform a classifier trained on data from the same session. We show that disagreement based semi-supervision can compensate for intrasession concept drift, and outperforms a classifier trained on more data.

To our knowledge, this is the first time a semi-supervised learning algorithm has been proposed for a hand orthosis. We intend this as an exploratory proof-of-concept study to evaluate semi-supervised learning as a potential avenue for orthotic controls. We believe these experiments show that semi-supervised learning is a promising avenue of exploration for orthotic hand controls. The results of the experiments also indicate areas of future research, emphasizing that these tests should be repeated in the context of functional movements, and indicating that we should try to predict when different semi-supervised learning algorithms will improve performance and when they will degrade performance, based on subject characteristics and incoming sensor data.

We would have liked to collect more data using stroke subjects for further validate both semi-supervision methods, particularly the K-means semi-supervision. Unfortunately, due to the pandemic, we were unable to bring in any more subjects. In the future we would like to do testing on more patients.

Our orthosis is eventually intended to be a take-home device. Minimizing the training burden placed on the user increases the probability that patients will use the device daily in their homes. Semi-supervised learning is one avenue to achieve this goal. In the future, we would also like to explore minimizing the role of the experimenter, who, in this thesis, is required to provide a labeled set of data to the classifier. We would like to automate the role of the experimenter, using

visual cues instead of verbal commands to instruct the patient when to try to open and close, and programmed motor actions to cycle through the different orthosis states.

We hope that our control, and future iterations, will inform the development of a wearable orthosis that can be used outside of clinical settings.

Chapter 11: Conclusion

Robots, once things of science fiction, now are beginning to permeate our homes and workplaces. And while virtual assistant robots are becoming common, robots which act on our physical environments are still largely kept separate from humans. These kinds of robots live in cages, operated primarily by experts, and remain too cost prohibitive for the general population.

We must remove these barriers if we want the next push in the field of robotics to more thoroughly integrate robotics into our everyday lives. This process has already begun in the form of human-safe, impedance controlled, and soft robotics.

Along with these new hardware paradigms, we will need human-machine interfaces which not only allow robots to move into human spaces, but which will allow humans and robots to communicate with each other. These interfaces need to be seamless, intuitive, and accessible to novices, while not interfering with human tasks or workspaces. In this thesis, we have presented human machine interfaces for this next generation of robots that are designed for and tested on novice users.

11.1 Contributions

Overall, the work in this thesis shows that human machine interfaces can be accessible to novice users, even when the interface controls a complex, non-anthropomorphic robot. These controls can be intuitive and require a relatively small amount of training. We show that this is true a variety of robotic applications and robot kinematics. We hope that this work encourages the field to design robotic controls with novices in mind.

We show that even when human-machine interfaces are designed for different applications and different populations, lessons learned in one field can inform controls for other fields. In both the

fields of teleoperation and rehabilitation robotics, using commodity devices paired with pattern classification algorithms made our controls easy to don because they do not require placement on specific muscles.

That being said, this work also emphasizes that some populations will not be able to produce a consistent control input when the robotic hardware interferes with the user's workspace or if it physically moves the user. When the hardware itself causes concept drift in the control input, it is important to develop the control in conjunction with the hardware. For our orthosis application, where the robot interacted with the human hand, we show that it is possible, and necessary, to compensate for the concept drift caused in this population both by the robotic hardware and by the task itself.

At an application specific level, in the field of teleoperation, we present a complete teleoperation pipeline to teleoperate a non-anthropomorphic, fully actuated robot in the context of collaborative manipulation. We are the first to show that a low dimensional subspace can enable intuitive teleoperation for novices operating hands of different kinematic configurations. We consider two control inputs - a dataglove and forearm EMG. Using a dataglove as a control input, we describe a mapping which is generated empirically and a mapping that is generated algorithmically. This algorithmic mapping is the first time that an automated method for generating a teleoperation has been shown to enable online teleoperation which is intuitive for novices. When using EMG as a control input, we are the first to show EMG controlled teleoperation of a non-anthropomorphic, fully actuated robot hand. Taken as a whole, we believe the work in this dissertation has advanced the field of collaborative manipulation by providing a control option which is more versatile in the number of control inputs and in the types of robotic kinematics it can accommodate than any other control to date. It is rare to test teleoperation controls with novices and we have provided the most thorough validation of a teleoperation control's effectiveness for novices to date. We have shown that our control method can be used for collaborative manipulation, and we hypothesize that it could also be used for learning from demonstration and even prosthetic applications.

In the field of rehabilitation robotics, we present a pattern classification algorithm which identi-

fies subject intent based on signals from the hand and forearm. Our target users are stroke subjects wearing an orthosis that assists finger extension. We work with a commodity device which is easy to don and does not require specific sensor placement. When the subject performs functional tasks with the orthosis on, the interaction between the hand and the orthosis causes concept drift, making this an important area of study; but one which has been largely ignored in the context of hand orthotics for stroke subjects. We demonstrate that concept drift is a significant problem for this population and this application and develop different algorithms to make our control robust to concept drift. We show that our control, when used in conjunction with an orthosis, can provide functional gains to chronic stroke subjects, a population for whom these gains is thought to be relatively rare [144, 145]. However, we also show that much work remains to be done to make this control effective as an assistive device which is robust to concept drift. Taken as a whole, this body of work builds toward the goal of a wearable orthotic device which works, not just in isolation, but in conjunction with an orthosis while enabling functional tasks.

11.2 Challenges

One challenge we encountered when designing human machine interfaces is that it is difficult to conduct user studies with large sample sizes. There are very few studies in the relevant literature which contains user studies with a large sample size. Novice users are unpredictable and can be careless. Robots are expensive and most of them are delicate. User studies which require novices to operate the robot with little or no training are a daunting prospect. In our case, three robot hands were harmed in the making of this dissertation. As a result, most research, this dissertation included, tries to draw conclusions from a small sample size, but a control's true efficacy and intuitiveness will likely not be fully known until it is deployed in real-world situations on a larger scale. We attempted to combat this by performing teleoperation user studies with robots of different kinematics, and by making our stroke subjects perform rehabilitation that involved functional Activities of Daily Living, but we need to deploy these controls in real-world settings to fully validate our methods.

Subject expectations present another challenge when designing human machine interfaces. If the robot or orthosis did not do exactly what the subject wanted exactly when and how the subject thought it should be done, they often thought the control was broken or required verbal encouragement to continue the task. This was true even if the robot was successfully executing the task. This challenge is particularly relevant in the field of rehabilitation robotics, where at-home usage correlates to perceived performance [69]. Our stroke subjects often reported dissatisfaction with our device because, although the orthosis improved their performance on many tasks, it did not return their hand to pre-stroke performance. When designing human-machine interfaces for the home or the workplace, it is likely that control training will need to include an education component designed to temper unrealistic expectations, even with effective and intuitive interfaces.

At an application specific level, one unexpected challenge we encountered for our orthotic human-machine interface was concept drift. For the stroke population, concept drift is a significant factor in the control's performance, even though it is not a large factor for healthy subjects. Concept drift has been studied for EMG controls for healthy patients, and even a few studies have looked at concept drift for amputees, but no one has looked at how concept drift affects stroke subjects. We found that designing the hardware and software in isolation and then combining them will result in changes in your control input that your interface will not be able to generalize to. We suggest several techniques to deal with concept drift; however, this is still a challenge we are actively working to solve.

11.3 Future Work

The work presented in this thesis highlights several areas for immediate follow up. We would like to validate our EMG control for the teleoperation subspace on hands of different kinematic configurations. We would like to use our algorithmic method to create a teleoperation method for a hand with highly non-anthropomorphic kinematics, like a continuum robot, and then test it in a user experiment. It would be interesting to add more dimensions to the teleoperation subspace to see if this increases the dexterity of the hand while remaining intuitive for the user. We would like

to show that all three of our teleoperation subspace mappings can be used for more complicated tasks, like to assemble and disassemble machinery. We would like to test our semi-supervised learning algorithms for an orthosis while the subject is performing functional tasks. We like to test if subjects could don and train our control without the help of a therapist. We would also like to see if training the subjects how to create more distinguishable EMG signals consistently could improve their performance with the device. We hypothesize that this would also allow us to expand the pool of subjects who could use our orthotic control.

In the longer term, human machine interfaces should be so seamless and intuitive that novices can not only use robotic devices, but do so without conscious effort. In the next decade, the field should focus on bringing robotic devices into our homes and workplaces and on making sure that these devices can be controlled by novices. In the field of rehabilitation robotics, rehabilitation needs to be broken out of a clinical setting. Currently, people do not use rehabilitation devices at home because they are expensive, hard to put on, hard to use, and do not assist with functional tasks. The field needs to be focused on solving these problems. People should be able to perform rehabilitation at home, unsupervised by therapists. This will allow them to perform therapy more frequently and perform therapy which includes functional tasks. Both of these factors make therapy more likely to be effective [129, 130]. In the field of collaborative manipulation, we would ultimately like the robot to operate autonomously, taking cues from the human in order to complete the task collaboratively. Since much of our environment and our tools are designed for human hands, using guidance from the human teleoperators to improve our autonomous programs should be a focus in the field in the next decade. Currently, the human is asked to complete a task when the autonomous program cannot, and then the autonomous program takes over again. It would be a significant advancement in the field if the autonomous program could learn from the guidance the user provides through teleoperation.

This dissertation shows that it is possible to make robotic devices accessible to novices, even when the robot is high dimensional, as is the case for robotic hands. We believe this is a first step into a world where robots are as common as dishwashers, and as easy to control.

References

- [1] R. Bogue, “Growth in e-commerce boosts innovation in the warehouse robot market,” *Industrial Robot: An International Journal*, vol. 43, no. 6, pp. 583–587, 2016.
- [2] K. Doelling, J. Shin, and D. O. Popa, “Service robotics for the home: A state of the art review,” in *Proceedings of the 7th International Conference on Pervasive Technologies Related to Assistive Environments*, ACM, 2014, p. 35.
- [3] N. Savela, T. Turja, and A. Oksanen, “Social acceptance of robots in different occupational fields: A systematic literature review,” *International Journal of Social Robotics*, vol. 10, no. 4, pp. 493–502, 2018.
- [4] M. Ferre, R. Aracil, C. Balaguer, M. Buss, and C. Melchiorri, *Advances in telerobotics*. Springer, 2007, vol. 31.
- [5] A. Bicchi, “Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity,” *IEEE Transactions on robotics and automation*, vol. 16, no. 6, pp. 652–662, 2000.
- [6] R. Matarneh, S. Maksymova, V Lyashenko, and N Belova, “Speech recognition systems: A comparative review,” 2017.
- [7] A. K. Orphanides and C. S. Nam, “Touchscreen interfaces in context: A systematic review of research into touchscreens across settings, populations, and implementations,” *Applied Ergonomics*, vol. 61, pp. 116–143, 2017.
- [8] M. Simão, N. Mendes, O. Gibaru, and P. Neto, “A review on electromyography decoding and pattern recognition for human-machine interaction,” *IEEE Access*, vol. 7, pp. 39 564–39 582, 2019.
- [9] P. Kumari, L. Mathew, and P. Syal, “Increasing trend of wearables and multimodal interface for human activity monitoring: A review,” *Biosensors and Bioelectronics*, vol. 90, pp. 298–307, 2017.
- [10] M. V. Liarokapis, P. Artemiadis, C. Bechlioulis, and K. Kyriakopoulos, “Directions, methods and metrics for mapping human to robot motion with functional anthropomorphism: A review,” *School of Mechanical Engineering, National Technical University of Athens, Tech. Rep*, 2013.

- [11] K. J. Kyriakopoulos, J. Van Riper, A. Zink, and H. E. Stephanou, “Kinematic analysis and position/force control of the anthropot dextrous hand,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 27, no. 1, pp. 95–104, 1997.
- [12] R. Chattaraj, B. Bepari, and S. Bhaumik, “Grasp mapping for dextrous robot hand: A hybrid approach,” in *Contemporary Computing (IC3), International Conference on*, IEEE, 2014, pp. 242–247.
- [13] A. Peer, S. Einkenkel, and M. Buss, “Multi-fingered telemanipulation-mapping of a human hand to a three finger gripper,” in *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*, IEEE, 2008, pp. 465–470.
- [14] W. B. Griffin, R. P. Findley, M. L. Turner, and M. R. Cutkosky, “Calibration and mapping of a human hand for dextrous telemanipulation,” in *ASME IMECE 2000 Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 2000, pp. 1–8.
- [15] H. Wang, K. H. Low, M. Y. Wang, and F. Gong, “A mapping method for telemanipulation of the non-anthropomorphic robotic hands with initial experimental validation,” in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, IEEE, 2005, pp. 4218–4223.
- [16] G. Gioioso, G. Salvietti, M. Malvezzi, and D. Prattichizzo, “Mapping synergies from human to robotic hands with dissimilar kinematics: An approach in the object domain,” *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 825–837, 2013.
- [17] D. G. Kamper, E. G. Cruz, and M. P. Siegel, “Stereotypical fingertip trajectories during grasp,” *Journal of neurophysiology*, vol. 90, no. 6, pp. 3702–3710, 2003.
- [18] L. Colasanto, R. Suárez, and J. Rosell, “Hybrid mapping for the assistance of teleoperated grasping tasks,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 2, pp. 390–401, 2013.
- [19] R. N. Rohling, J. M. Hollerbach, and S. C. Jacobsen, “Optimized fingertip mapping: A general algorithm for robotic hand teleoperation,” *Presence: Teleoperators & Virtual Environments*, vol. 2, no. 3, pp. 203–220, 1993.
- [20] T. Geng, M. Lee, and M. Hülse, “Transferring human grasping synergies to a robot,” *Mechatronics*, vol. 21, no. 1, pp. 272–284, 2011.
- [21] S. Ekvall and D. Kragic, “Interactive grasp learning based on human demonstration,” in *Robotics and Automation. Proceedings. IEEE International Conference on*, IEEE, vol. 4, 2004, pp. 3519–3524.
- [22] T. Wojtara and K. Nonami, “Hand posture detection by neural network and grasp mapping for a master slave hand system,” in *Intelligent Robots and Systems, 2004.(IROS 2004)*.

Proceedings. 2004 IEEE/RSJ International Conference on, IEEE, vol. 1, 2004, pp. 866–871.

- [23] L. Pao and T. H. Speeter, “Transformation of human hand positions for robotic hand control,” in *Robotics and Automation. Proceedings., IEEE International Conference on*, IEEE, 1989, pp. 1758–1763.
- [24] S. Li, X. Ma, H. Liang, M. Görner, P. Ruppel, B. Fang, F. Sun, and J. Zhang, “Vision-based teleoperation of shadow dexterous hand using end-to-end deep neural network,” in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 416–422.
- [25] I. Cerulo, F. Ficuciello, V. Lippiello, and B. Siciliano, “Teleoperation of the schunk s5fh under-actuated anthropomorphic hand using human hand motion tracking,” *Robotics and Autonomous Systems*, vol. 89, pp. 75–84, 2017.
- [26] G. Du, P. Zhang, J. Mai, and Z. Li, “Markerless kinect-based hand tracking for robot teleoperation,” *International Journal of Advanced Robotic Systems*, vol. 9, no. 2, p. 36, 2012.
- [27] J. Romero, H. Kjellström, and D. Kragic, “Monocular real-time 3d articulated hand pose estimation,” in *2009 9th IEEE-RAS International Conference on Humanoid Robots*, IEEE, 2009, pp. 87–92.
- [28] M. Santello, M. Flanders, and J. F. Soechting, “Postural hand synergies for tool use,” *J. Neurosci.*, vol. 18, no. 23, pp. 10 105–10 115, 1998.
- [29] T. Feix, R. Pawlik, H.-B. Schmiedmayer, J. Romero, and D. Kragic, “A comprehensive grasp taxonomy,” in *Robotics, science and systems: workshop on understanding the human hand for advancing robotic manipulation*, 2009, pp. 2–3.
- [30] F. Ficuciello, G. Palli, C. Melchiorri, and B. Siciliano, “Postural synergies and neural network for autonomous grasping: A tool for dextrous prosthetic and robotic hands,” in *Converging Clinical and Engineering Research on Neurorehabilitation*, Springer, 2013, pp. 467–480.
- [31] S. Kim, M. Kim, J. Lee, and J. Park, “Robot hand synergy mapping using multi-factor model and emg signal,” in *Experimental robotics*, Springer, 2016, pp. 671–683.
- [32] F. Ficuciello, G. Palli, C. Melchiorri, and B. Siciliano, “Planning and control during reach to grasp using the three predominant ub hand iv postural synergies,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, IEEE, 2012, pp. 2255–2260.
- [33] M. T. Ciocarlie and P. K. Allen, “Hand posture subspaces for dexterous robotic grasping,” *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 851–867, 2009.

- [34] C. Della Santina, C. Piazza, G. Grioli, M. G. Catalano, and A. Bicchi, “Toward dexterous manipulation with augmented adaptive synergies: The pisa/iit soft hand 2,” *IEEE Transactions on Robotics*, no. 99, pp. 1–16, 2018.
- [35] O. C. Jenkins, “Sparse control for high-dof assistive robots,” *Intelligent Service Robotics*, vol. 1, no. 2, pp. 135–141, 2008.
- [36] A. Brygo, I. Sarakoglou, G. Grioli, and N. Tsagarakis, “Synergy-based bilateral port: A universal control module for tele-manipulation frameworks using asymmetric master–slave systems,” *Frontiers in bioengineering and biotechnology*, vol. 5, 2017.
- [37] G. Palli, C. Melchiorri, G. Vassura, U Scarcia, L. Moriello, G. Berselli, A. Cavallo, G. De Maria, C. Natale, S. Pirozzi, *et al.*, “The dexmart hand: Mechatronic design and experimental evaluation of synergy-based control for human-like grasping,” *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 799–824, 2014.
- [38] K. Yamane, Y. Ariki, and J. Hodgins, “Animating non-humanoid characters with human motion data,” in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 2010, pp. 169–178.
- [39] A. Shon, K. Grochow, A. Hertzmann, and R. P. Rao, “Learning shared latent structure for image synthesis and robotic imitation,” in *Advances in neural information processing systems*, 2006, pp. 1233–1240.
- [40] B. Delhaisse, D. Esteban, L. Rozo, and D. Caldwell, “Transfer learning of shared latent spaces between robots with similar kinematic structure,” in *Neural Networks (IJCNN), 2017 International Joint Conference on*, IEEE, 2017, pp. 4142–4149.
- [41] A. Kheddar, C Tzafestas, and P. Coiffet, “The hidden robot concept-high level abstraction teleoperation,” in *Intelligent Robots and Systems, 1997. IROS’97., Proceedings of the 1997 IEEE/RSJ International Conference on*, IEEE, vol. 3, 1997, pp. 1818–1825.
- [42] A. Kheddar, “Teleoperation based on the hidden robot concept,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 31, no. 1, pp. 1–13, 2001.
- [43] S. Chernova and A. L. Thomaz, “Robot learning from human teachers,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 8, no. 3, pp. 1–121, 2014.
- [44] S. B. Kang and K. Ikeuchi, “Toward automatic robot instruction from perception-mapping human grasps to manipulator grasps,” *IEEE transactions on robotics and automation*, vol. 13, no. 1, pp. 81–95, 1997.

- [45] G Gioioso, G Salvietti, M Malvezzi, and D Prattichizzo, “An object-based approach to map human hand synergies onto robotic hands with dissimilar kinematics,” *Robotics: Science and Systems VIII*, pp. 97–104, 2013.
- [46] G. Gioioso, G. Salvietti, M. Malvezzi, and D. Prattichizzo, “Mapping synergies from human to robotic hands with dissimilar kinematics: An approach in the object domain,” *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 825–837, 2013.
- [47] G. Salvietti, M. Malvezzi, G. Gioioso, and D. Prattichizzo, “On the use of homogeneous transformations to map human hand movements onto robotic hands,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 5352–5357.
- [48] G. Salvietti, L. Meli, G. Gioioso, M. Malvezzi, and D. Prattichizzo, “Object-based bilateral telemanipulation between dissimilar kinematic structures,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2013, pp. 5451–5456.
- [49] —, “Multicontact bilateral telemanipulation with kinematic asymmetries,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 445–456, 2016.
- [50] A. Herrera, A. Bernal, D. Isaza, and M. Adjouadi, “Design of an electrical prosthetic gripper using emg and linear motion approach,” in *Proceedings from the 17th Florida Conference on the Recent Advances in Robotics (FCRAR), Florida, USA, 2004*.
- [51] R. B. Gillespie, J. L. Contreras-Vidal, P. A. Shewokis, M. K. O’Malley, J. D. Brown, H. Agashe, R. Gentili, and A. Davis, “Toward improved sensorimotor integration and learning using upper-limb prosthetic devices,” in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual Int Conf on*, IEEE, 2010, pp. 5077–5080.
- [52] G. Vasan and P. M. Pilarski, “Learning from demonstration: Teaching a myoelectric prosthesis with an intact limb via reinforcement learning,” in *Rehabilitation Robotics (ICORR), 2017 Int Conf on.*, IEEE, 2017, pp. 1457–1464.
- [53] S. K. Choudhary, D. Chakraborty, N. M. Kakoty, and S. M. Hazarika, “Development of cost effective emg controlled three fingered robotic hand,” in *Computer and Communication Technology (ICCCT), 2012 Third International Conference on*, IEEE, 2012, pp. 104–109.
- [54] S. Fani, M. Bianchi, S. Jain, J. S. Pimenta Neto, S. Boege, G. Grioli, A. Bicchi, and M. Santello, “Assessment of myoelectric controller performance and kinematic behavior of a novel soft synergy-inspired robotic hand for prosthetic applications,” *Frontiers in neuro-robotics*, vol. 10, p. 11, 2016.
- [55] N. Malešević, D. Marković, G. Kanitz, M. Controzzi, C. Cipriani, and C. Antfolk, “Decoding of individual finger movements from surface emg signals using vector autoregres-

- sive hierarchical hidden markov models (varhhmm),” in *Rehabilitation Robotics (ICORR), 2017 Int Conf on.*, 2017, pp. 1518–1523.
- [56] R. J. Smith, F. Tenore, D. Huberdeau, R. Etienne-Cummings, and N. V. Thakor, “Continuous decoding of finger position from surface emg signals for the control of powered prostheses,” in *Engineering in Medicine and Biology Society (EMBC), 2008 Annual Int Conf on*, IEEE, 2008, pp. 197–200.
- [57] J. M. Hahne, F. Biessmann, N. Jiang, H. Rehbaum, D. Farina, F. Meinecke, K.-R. Müller, and L. Parra, “Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 22, pp. 269–279, 2014.
- [58] N. Jiang, K. B. Englehart, and P. A. Parker, “Extracting simultaneous and proportional neural control information for multiple-dof prostheses from the surface electromyographic signal,” *IEEE Trans Biomed Eng*, vol. 56, no. 4, pp. 1070–1080, 2009.
- [59] N. Jiang, H. Rehbaum, I. Vujaklija, B. Graimann, and D. Farina, “Intuitive, online, simultaneous, and proportional myoelectric control over two degrees-of-freedom in upper limb amputees,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 22, no. 3, pp. 501–510, 2014.
- [60] C. Lin, B. Wang, N. Jiang, and D. Farina, “Robust extraction of basis functions for simultaneous and proportional myoelectric control via sparse non-negative matrix factorization,” *J Neural Eng*, 2017.
- [61] M. Yoshikawa, M. Mikawa, and K. Tanaka, “Hand pose estimation using emg signals,” in *Engineering in Medicine and Biology Society (EMBC), 2007 Annual Int Conf on*, IEEE, 2007, pp. 4830–4833.
- [62] Y. Yamanoi, S. Morishita, R. Kato, and H. Yokoi, “Development of myoelectric hand that determines hand posture and estimates grip force simultaneously,” *Biomedical Signal Processing and Control*, vol. 38, pp. 312–321, 2017.
- [63] C. Castellini and P. van der Smagt, “Surface emg in advanced hand prosthetics,” *Biological cybernetics*, vol. 100, no. 1, pp. 35–47, 2009.
- [64] A. Gijssberts, R. Bohra, D. Sierra González, A. Werner, M. Nowak, B. Caputo, M. A. Roa, and C. Castellini, “Stable myoelectric control of a hand prosthesis using non-linear incremental learning,” *Frontiers in neurorobotics*, vol. 8, p. 8, 2014.
- [65] M. Santello, M. Bianchi, M. Gabiccini, E. Ricciardi, G. Salvietti, D. Prattichizzo, M. Ernst, A. Moscatelli, H. Jörntell, A. M. Kappers, *et al.*, “Hand synergies: Integration of robotics and neuroscience for understanding the control of biological and artificial hands,” *Physics of life reviews*, vol. 17, pp. 1–23, 2016.

- [66] M. Rossi, C. Della Santina, C. Piazza, G. Grioli, M. Catalano, and A. Biechi, "Preliminary results toward a naturally controlled multi-synergistic prosthetic hand," in *Rehabilitation Robotics (ICORR), 2017 Int Conf on.*, IEEE, 2017, pp. 1356–1363.
- [67] G. C. Matrone, C. Cipriani, M. C. Carrozza, and G. Magenes, "Real-time myoelectric control of a multi-fingered hand prosthesis using principal components analysis," *J Neuroeng Rehabil*, vol. 9, no. 1, p. 40, 2012.
- [68] P. K. Artemiadis and K. J. Kyriakopoulos, "A switching regime model for the emg-based control of a robot arm," *IEEE Trans Syst Man Cybern B Cybern*, vol. 41, no. 1, pp. 53–63, 2011.
- [69] B. Phillips and H. Zhao, "Predictors of assistive technology abandonment," *Assistive technology*, vol. 5, no. 1, pp. 36–45, 1993.
- [70] L. C. Smail, C. Neal, C. Wilkins, and T. L. Packham, "Comfort and function remain key factors in upper limb prosthetic abandonment: Findings of a scoping review," *Disability and Rehabilitation: Assistive Technology*, pp. 1–10, 2020.
- [71] E. Buch, C. Weber, L. G. Cohen, C. Braun, M. A. Dimyan, T. Ard, J. Mellinger, A. Caria, S. Soekadar, A. Fourkas, *et al.*, "Think to move: A neuromagnetic brain-computer interface (bci) system for chronic stroke," *Stroke*, vol. 39, no. 3, pp. 910–917, 2008.
- [72] H. Kawasaki, S. Ito, Y. Ishigure, Y. Nishimoto, T. Aoki, T. Mouri, H. Sakaeda, and M. Abe, "Development of a Hand Motion Assist Robot for Rehabilitation Therapy by Patient Self-Motion Control," in *IEEE International Conference on Rehabilitation Robotics*, 2007.
- [73] C. D. Takahashi, L. Der-Yeghiaian, V. Le, R. R. Motiwala, and S. C. Cramer, "Robot-based hand motor therapy after stroke," *Brain*, vol. 131, no. 2, pp. 425–437, 2008.
- [74] W. Hassani, S. Mohammed, and Y. Amirat, "Real-time emg driven lower limb actuated orthosis for assistance as needed movement strategy.," in *Robotics: Science and Systems*, Citeseer, 2013.
- [75] L. M. Vaca Benitez, M. Tabie, N. Will, S. Schmidt, M. Jordan, and E. A. Kirchner, "Exoskeleton technology in rehabilitation: Towards an emg-based orthosis system for upper limb neuromotor rehabilitation," *Journal of Robotics*, vol. 2013, 2013.
- [76] Z. O. Khokhar, Z. G. Xiao, and C. Menon, "Surface emg pattern recognition for real-time control of a wrist exoskeleton," *Biomedical engineering online*, vol. 9, no. 1, p. 41, 2010.
- [77] L. Lucas, M. DiCicco, and Y. Matsuoka, "An emg-controlled hand exoskeleton for natural pinching," *JRM*, vol. 16, pp. 482–488, 2004.

- [78] M. DiCicco, L. Lucas, and Y. Matsuoka, "Comparison of control strategies for an emg controlled orthotic exoskeleton for the hand," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, IEEE, vol. 2, 2004, pp. 1622–1627.
- [79] P. Maciejasz, J. Eschweiler, K. Gerlach-Hahn, A. Jansen-Troy, and S. Leonhardt, "A survey on robotic devices for upper limb rehabilitation," *J of neuroeng and rehabil*, vol. 11, no. 1, p. 1, 2014.
- [80] M. Mulas, M. Folgheraiter, and G. Gini, "An emg-controlled exoskeleton for hand rehabilitation," in *9th International Conference on Rehabilitation Robotics ICORR*, IEEE, 2005, pp. 371–374.
- [81] S Kavya, R Sushma, G. Kumar, *et al.*, "Controlling the hand and forearm movements of an orthotic arm using surface emg signals," in *2015 Annual IEEE India Conference*, IEEE, 2015, pp. 1–6.
- [82] M Chen, S. Ho, H. F. Zhou, P. Pang, X. Hu, D. Ng, and K. Tong, "Interactive rehabilitation robot for hand function training," in *IEEE Int. Conf. on Rehabilitation Robotics*, 2009, pp. 777–780.
- [83] K. Tong, S. Ho, P. Pang, X. Hu, W. Tam, K. Fung, X. Wei, P. Chen, and M Chen, "An intention driven hand functions task training robotic system," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, IEEE, 2010, pp. 3406–3409.
- [84] N. Ho, K. Tong, X. Hu, K. Fung, X. Wei, W Rong, and E. Susanto, "An emg-driven exoskeleton hand robotic training device on chronic stroke subjects: Task training system for stroke rehabilitation," in *IEEE international conference on rehabilitation robotics*, 2011, pp. 1–5.
- [85] J. M. Ochoa, D. G. Kamper, M. Listenberger, and S. W. Lee, "Use of an electromyographically driven hand orthosis for training after stroke," in *IEEE Int. Conf. on Rehabilitation Robotics*, 2011, pp. 1–5.
- [86] S. W. Lee, K. M. Wilson, B. A. Lock, and D. G. Kamper, "Subject-specific myoelectric pattern classification of functional hand movements for stroke survivors," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 5, pp. 558–566, 2011.
- [87] M. A. Powell and N. V. Thakor, "A training strategy for learning pattern recognition control for myoelectric prostheses," *Journal of prosthetics and orthotics: JPO*, vol. 25, no. 1, p. 30, 2013.
- [88] A. Fougner, Ø. Stavadahl, P. J. Kyberd, Y. G. Losier, and P. A. Parker, "Control of upper limb prostheses: Terminology and proportional myoelectric control—a review," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 20, no. 5, pp. 663–677, 2012.

- [89] A. Ajiboye and R. Weir, "A Heuristic Fuzzy Logic Approach to EMG Pattern Recognition for Multifunctional Prosthesis Control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 3, pp. 280–291, 2005.
- [90] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE Transactions on Bio-Medical Engineering*, vol. 50, no. 7, pp. 848–54, 2003.
- [91] C. Cipriani, F. Zaccone, S. Micera, and M. Carrozza, "On the Shared Control of an EMG-Controlled Prosthetic Hand: Analysis of User–Prosthesis Interaction," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 170–184, 2008.
- [92] J. He, D. Zhang, and X. Zhu, "Adaptive pattern recognition of myoelectric signal towards practical multifunctional prosthesis control," in *International Conference on Intelligent Robotics and Applications*, Springer, 2012, pp. 518–525.
- [93] S. Jain, G. Singhal, R. J. Smith, R. Kaliki, and N. Thakor, "Improving long term myoelectric decoding, using an adaptive classifier with label correction," in *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, IEEE, 2012, pp. 532–537.
- [94] S. Amsüss, P. M. Goebel, N. Jiang, B. Graimann, L. Paredes, and D. Farina, "Self-correcting pattern recognition system of surface emg signals for upper limb prosthesis control," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 4, pp. 1167–1176, 2013.
- [95] X. Chen, D. Zhang, and X. Zhu, "Application of a self-enhancing classification method to electromyography pattern recognition for multifunctional prosthesis control," *Journal of neuroengineering and rehabilitation*, vol. 10, no. 1, p. 44, 2013.
- [96] X. Zhai, B. Jelfs, R. H. Chan, and C. Tin, "Self-recalibrating surface emg pattern recognition for neuroprosthesis control based on convolutional neural network," *Frontiers in neuroscience*, vol. 11, p. 379, 2017.
- [97] J. W. Sensinger, B. A. Lock, and T. A. Kuiken, "Adaptive pattern recognition of myoelectric signals: Exploration of conceptual framework and practical algorithms," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 17, no. 3, pp. 270–278, 2009.
- [98] H. Zhang, Y. Zhao, F. Yao, L. Xu, P. Shang, and G. Li, "An adaptation strategy of using lda classifier for emg pattern recognition," in *2013 35th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, IEEE, 2013, pp. 4267–4270.
- [99] J. Liu, X. Sheng, D. Zhang, N. Jiang, and X. Zhu, "Towards zero retraining for myoelectric control based on common model component analysis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 4, pp. 444–454, 2015.

- [100] A. L. Edwards, M. R. Dawson, J. S. Hebert, C. Sherstan, R. S. Sutton, K. M. Chan, and P. M. Pilarski, “Application of real-time machine learning to myoelectric prosthesis control: A case series in adaptive switching,” *Prosthetics and orthotics international*, vol. 40, no. 5, pp. 573–581, 2016.
- [101] M. Kawakita and J. Takeuchi, “Safe semi-supervised learning based on weighted likelihood,” *Neural Networks*, vol. 53, pp. 146–164, 2014.
- [102] N. Sokolovska, O. Cappé, and F. Yvon, “The asymptotics of semi-supervised learning in discriminative probabilistic models,” in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 984–991.
- [103] Y.-F. Li and Z.-H. Zhou, “Towards making unlabeled data never hurt,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 1, pp. 175–188, 2014.
- [104] Y. Wang, Y. Meng, Z. Fu, and H. Xue, “Towards safe semi-supervised classification: Adjusted cluster assumption via clustering,” *Neural Processing Letters*, vol. 46, no. 3, pp. 1031–1042, 2017.
- [105] Z.-H. Zhou and M. Li, “Semi-supervised learning by disagreement,” *Knowledge and Information Systems*, vol. 24, no. 3, pp. 415–439, 2010.
- [106] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proceedings of the eleventh annual conference on Computational learning theory*, 1998, pp. 92–100.
- [107] W. Wang and Z.-H. Zhou, “Analyzing co-training style algorithms,” in *European conference on machine learning*, Springer, 2007, pp. 454–465.
- [108] S. Goldman and Y. Zhou, “Enhancing supervised learning with unlabeled data,” in *ICML*, Citeseer, 2000, pp. 327–334.
- [109] M.-F. Balcan, A. Blum, and K. Yang, “Co-training and expansion: Towards bridging theory and practice,” in *Advances in neural information processing systems*, 2005, pp. 89–96.
- [110] J. Wang and S.-w. Luo, “Exploiting ensemble method in semi-supervised learning,” in *2006 International Conference on Machine Learning and Cybernetics*, IEEE, 2006, pp. 1104–1107.
- [111] O. Javed, S. Ali, and M. Shah, “Online detection and classification of moving objects using progressively improving detectors,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, IEEE, vol. 1, 2005, pp. 696–701.

- [112] J. Z. Kolter and M. A. Maloof, “Dynamic weighted majority: An ensemble method for drifting concepts,” *Journal of Machine Learning Research*, vol. 8, no. Dec, pp. 2755–2790, 2007.
- [113] Z. Jiang and S. Zhang, “A semi-supervised ensemble learning algorithm,” in *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, IEEE, vol. 2, 2012, pp. 913–918.
- [114] J. Rosell, R. Suárez, and A. Pérez, “Safe teleoperation of a dual hand-arm robotic system,” in *ROBOT2013: First Iberian Robotics Conference*, Springer, 2014, pp. 615–630.
- [115] A. T. Miller and P. K. Allen, “Graspit! a versatile simulator for robotic grasping,” *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.
- [116] C. Ferrari and J. Canny, “Planning optimal grasps,” in *Robotics and Automation, 1992. Proceedings., 1992 IEEE Int’l Conf. on.*
- [117] K. G. Derpanis, “Overview of the ransac algorithm,” *Image Rochester NY*, vol. 4, no. 1, pp. 2–3, 2010.
- [118] R. Raguram, J.-M. Frahm, and M. Pollefeys, “A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus,” in *European Conf. on Computer Vision*, 2008, pp. 500–513.
- [119] Å. Björck, “Numerics of gram-schmidt orthogonalization,” *Linear Algebra and Its Applications*, vol. 197, pp. 297–316, 1994.
- [120] S. Cobos, M. Ferre, M. S. Uran, J. Ortego, and C. Pena, “Efficient human hand kinematics for manipulation tasks,” in *2008 IEEE/RSJ Int’l Conf. on Intelligent Robots and Systems*, 2008, pp. 2246–2251.
- [121] C. Parlitz, “Hardware for industrial gripping at schunk gmbh & co. kg,” in *Grasping in Robotics*, Springer, 2013, pp. 363–384.
- [122] (). Accessed: 2020-07, Rethink Robotics.
- [123] (). Accessed: 2020-07, Ascension Technology Corporation.
- [124] G. D. Kessler, L. F. Hodges, and N. Walker, “Evaluation of the cyberglove as a whole-hand input device,” *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 2, no. 4, pp. 263–283, 1995.
- [125] T. Chen and M. Ciocarlie, “Proprioception-based grasping for unknown objects using a series-elastic-actuated gripper,” in *2018 IEEE/RSJ Int’l Conf. on Intelligent Robots and Syst. (IROS)*, 2018, pp. 6675–6681.

- [126] P. Fuchs, G. Moreau, and P. Guitton, *Virtual reality: concepts and technologies*. CRC Press, 2011.
- [127] R. J. Schwarz and C. Taylor, “The anatomy and mechanics of the human hand,” *Artificial limbs*, vol. 2, no. 2, pp. 22–35, 1955.
- [128] M. V. Liarokapis, P. K. Artemiadis, P. T. Katsiaris, K. J. Kyriakopoulos, and E. S. Manolakos, “Learning human reach-to-grasp strategies: Towards emg-based control of robotic arm-hand systems,” in *Robotics and Automation (ICRA), 2012 IEEE Int Conf on*, 2012, pp. 2287–2292.
- [129] C. H. Shea and R. M. Kohl, “Composition of practice: Influence on the retention of motor skills,” *Research quarterly for exercise and sport*, vol. 62, no. 2, pp. 187–195, 1991.
- [130] J. W. Krakauer, “Motor learning: Its relevance to stroke recovery and neurorehabilitation,” *Current opinion in neurology*, vol. 19, no. 1, pp. 84–90, 2006.
- [131] L. C. Miller and J. P. Dewald, “Involuntary paretic wrist/finger flexion forces and emg increase with shoulder abduction load in individuals with chronic stroke,” *Clinical Neurophysiology*, vol. 123, no. 6, pp. 1216–1225, 2012.
- [132] S. Park, L. Bishop, T. Post, Y. Xiao, J. Stein, and M. Ciocarlie, “On the feasibility of wearable exotendon networks for whole-hand movement patterns in stroke patients,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [133] N. A. Lannin, S. A. Horsley, R. Herbert, A. McCluskey, and A. Cusick, “Splinting the hand in the functional position after brain impairment: A randomized, controlled trial,” *Archives of Physical Medicine and Rehabilitation*, vol. 84, no. 2, pp. 297–302, 2003.
- [134] J. P. Dewald, P. S. Pope, J. D. Given, T. S. Buchanan, and W. Z. Rymer, “Abnormal muscle coactivation patterns during isometric torque generation at the elbow and shoulder in hemiparetic subjects,” *Brain*, vol. 118, no. 2, pp. 495–510, 1995.
- [135] T. R. Farrell *et al.*, “A comparison of the effects of electrode implantation and targeting on pattern classification accuracy for prosthesis control,” *IEEE Trans. Biomed. Eng.*, vol. 55, no. 9, 2008.
- [136] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, 2001.
- [137] S. Park, M. Fraser, L. M. Weber, C. Meeker, L. Bishop, D. Geller, J. Stein, and M. Ciocarlie, “User-driven functional movement training with a wearable hand robot after stroke,” *arXiv preprint arXiv:1911.08003*, 2019.

- [138] A. R. Fugl-Meyer, L. Jääskö, I. Leyman, S. Olsson, and S. Steglind, "The post-stroke hemiplegic patient. 1. a method for evaluation of physical performance.," *Scandinavian journal of rehabilitation medicine*, vol. 7, no. 1, pp. 13–31, 1975.
- [139] R. C. Lyle, "A performance test for assessment of upper limb function in physical rehabilitation treatment and research," *International journal of rehabilitation research*, vol. 4, no. 4, pp. 483–492, 1981.
- [140] V. Mathiowetz, G. Volland, N. Kashman, and K. Weber, "Adult norms for the box and block test of manual dexterity," *American Journal of Occupational Therapy*, vol. 39, no. 6, pp. 386–391, 1985.
- [141] N. Riley and M. Bilodeau, "Changes in upper limb joint torque patterns and emg signals with fatigue following a stroke," *Disability and rehabilitation*, vol. 24, no. 18, pp. 961–969, 2002.
- [142] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.
- [143] M. Ortiz-Catalan, F. Rouhani, R. Brånemark, and B. Håkansson, "Offline accuracy: A potentially misleading metric in myoelectric pattern recognition for prosthetic control," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2015, pp. 1140–1143.
- [144] P. W. Duncan, L. B. Goldstein, D. Matchar, G. W. Divine, and J. Feussner, "Measurement of motor recovery after stroke. outcome assessment and sample size requirements.," *Stroke*, vol. 23, no. 8, 1992.
- [145] H. Y. Jung, J. S. Yoon, and B. S. Park, "Recovery of proximal and distal arm weakness in the ipsilateral upper limb after stroke," *NeuroRehabilitation*, vol. 17, no. 2, pp. 153–159, 2002.