

**Design of a Multi-Articulating Prosthetic Hand that Enables Continual
Reinforcement Learning Methods**

by

Annette C. Lau

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Rehabilitation Science

Faculty of Rehabilitation Medicine
University of Alberta

© Annette C. Lau, 2026

This work is licensed under a CC Attribution-ShareAlike 4.0 International.

Abstract

Upper-limb loss introduces complex functional and psychological challenges which modern powered prostheses attempt to address. However, prosthetic devices are often limited by high costs and difficulties in user control. Traditional control strategies often rely on rigid, manual switching between functions such as grasp configurations. Furthermore, controlling these devices using muscle signals can lead to slow operation, user frustration, and high abandonment rates.

This thesis addresses the gap between modern prosthetic hands and advanced control systems that use continual machine learning. We first present the design and evaluation of the HANDi (Humanoid Anthropomorphic Naturally Dexterous Intelligent) Hand V2. It is a fully open-source, 3D-printed robotic hand developed with machine learning research in mind. The hand incorporates a myriad of sensors, including internal motor sensors, fingertip force sensors, a wrist-mounted camera, and more. Beyond hardware, and to highlight the research-readiness of the HANDi Hand V2, we demonstrate for the first time how a continual predictive learning strategy called Adaptive Switching can be extended to enhance a user’s selection of prosthetic grasps. Rather than manually navigating a menu of grasp patterns to complete multi-stage tasks, the machine learning system dynamically reorders controllable grasp functions for the user based on context.

The HANDi Hand V2 was evaluated using the Anthropomorphic Hand Assessment Protocol (AHAP) to assess functional grasping ability across common household objects. Achieving a Grasping Ability Score of 79%, the HANDi Hand V2 brings 3D-printed hands closer in performance to commercial devices. In addition, our ex-

periments with adaptive switching demonstrated that combining a highly sensorized robotic hand with predictive machine learning reduced a simple grasp selection task’s completion time by 12 seconds. Furthermore, the number of manual switching interactions required within this task was reduced by a factor of 4.

Taken as a whole, this thesis contributes a new viewpoint on prosthetic control by combining real-time predictive learning for grasp selection with an open-source device made to enable continual machine learning research. It further contributes insight into how open-source releases, such as that of the HANDi Hand V2, can better serve the communities that engage with them. Ultimately, we believe this thesis presents a scalable foundation for future adaptive prostheses, advancing both the hardware and software of human-robot interaction.

Preface

This thesis is an original work by Annette C. Lau (ORCID: 0009-0009-1743-9080). The research project, of which this thesis is a part, received research ethics approval from the University of Alberta Research Ethics Board, Project Name “Evaluation of novel prosthetic hardware and control strategies”, No. 00077893, February 26, 2025. This thesis comprises content from three manuscripts currently published or submitted.

Chapter 2 contains information published as:

Joshua R. Siegel, Jedidiah K. Harwood, **Annette C. Lau**, Dylan J. A. Brenneis, Michael R. Dawson, Patrick M. Pilarski, Jonathon S. Schofield, “A performance evaluation of commercially available and 3D-printable prosthetic hands: a comparison using the anthropomorphic hand assessment protocol”. *BMC Biomedical Engineering* 6(11), December 2024.

Contributions as third author: led the University of Alberta experiments in a collaboration with UC Davis; experimentation for 20% of the hands included in the assessment; data analysis; figure generation; writing, review and editing of draft manuscripts for co-author review; and preparation of manuscript revisions in response to peer reviews.

Chapter 3 contains information published as:

Annette C. Lau, Michael R. Dawson, Patrick M. Pilarski, “Continual Prediction Learning Techniques to Support Users’ Grasp Selection on a

Multi-Articulating Prosthetic Hand.” *Proceedings of the 38th Canadian Conference on Artificial Intelligence*, Calgary, Alberta, May 2025.

Contributions as first author: experimental protocol design; fabrication and modifications of experimental apparatuses; processing and analysis of experimental data; figure generation; writing, review, and editing of draft manuscripts for co-author review; and preparation of manuscript revisions in response to peer reviews.

Chapter 4 contains information published as:

Annette C. Lau, Stevie M. Desmarais, Simon M.-Y. Wong, Dylan J. A. Brenneis, Michael R. Dawson, Patrick M. Pilarski, “The HANDi Hand V2: Enhancing a Multi-articulating Prosthetic Hand for Continual Machine Learning Research” *Submitted to BMC Biomedical Engineering*, August 2026 (under review).

All participants provided informed consent via a study approved by the University of Alberta Research Ethics Board (Pro00077893) in accordance with the Declaration of Helsinki.

Contributions as first author: experimental protocol design; fabrication and modifications of experimental apparatuses; processing and analysis of experimental data; figure generation; writing, review, and editing of draft manuscripts for co-author review.

Dedicated to my past, present, and future self, for having the stubborn (and perhaps foolish) tenacity to do hard things.

Acknowledgements

As I sit at a desk in the co-working space at Amii, I'm finding myself typing the most difficult and also the most meaningful section of my thesis. There is an unexpected poetry in writing the acknowledgements, in the sense that I have to now somehow summarize these remarkable years of human experience and gratitude into a sequential series of thank-yous. In this section, I will attempt to acknowledge everyone in my entire life while someone cuts onions in the background.

First, I'd like to extend infinite thanks to my dream supervisory team. To Rory, thank you for all the silly to serious chats and top-notch mentorship. You have shaped me to be the designer I am today. I will never forget to add extra strain relief to cables ever again. To Patrick, thank you for always encouraging me to step out of my comfort zone. I never believed in my coding skills, but thankfully, you did. I'll always appreciate our nerdy conversations about 3D printing, reinforcement learning, and Pokémon. Next, a huge shoutout to my BLINC Lab family for adopting me. Every single one of you, whether you've graduated or are still chugging along, has made the windowless sterile lab feel like a second home. From the day I started as a shy undergraduate student who was scared to leave her desk for lunch, to someone who plans and executes elaborate pranks on supervisors, I look back fondly on all the memories we share. I also want to thank my fellow HANDi Hand hooligans, Dylan Brenneis, Stevie Desmarais and Simon Wong, for the hard work and contributions to this project. The HANDi Hand would not be where it is today without you! Next, I want to express my appreciation to the BLINC girlies who continuously inspire me as a woman in STEM. I have no doubt that you all will change the world.

I would also like to acknowledge the lovely 2nd floor co-working space at Amii, which coaxed me out of my bed to make the 15-minute trek across Jasper Ave with sweet treats. Writing this thesis required a concerningly high number of caffeinated beverages. I want to especially recognize Chynna for the constant supply of delicious coffee and Silk[®] Vanilla Oat Creamer. I could not have done it without you!

Next, to Angela, the Graduate Studies Coordinator in Rehab Med; thank you for giving me the smoothest introduction to graduate school chaos. I will never regret choosing to study rehabilitation science with the amount of care I've experienced in this faculty. You are a star!

To my eccentric family, which we endearingly refer to as the rat colony; you kept me afloat during times of turbulence. Thank you for making sure I was always well-fed and not afraid to be myself. To Papa Rat, whom I always said that I wouldn't follow in your academic steps; well, here I am, studying machine learning in my Master's degree. Thank you for actually reading my papers and inspiring me to pursue graduate studies. To Mama Rat, thank you for confusedly supporting my studies, even though I still don't think you fully understand what it's about. To the Big Rat, thank you for always (begrudgingly) running the dishwasher and tolerating all my pre-presentation nights where I rehearse out loud in our living-room-turned-bike-room. May we continue making fake teeth and hands for those who need them.

To all the friends I've met along the way and have supported both my academic career and my very amateur sports career that I'm treating professionally, thank you. I'm especially grateful for the sense of community you've provided outside of school and work. It truly does take a village.

Finally, to my patient partner, Jason, thank you for fighting alongside me as I battled all the obstacles life threw at me during this time. You could probably defend my thesis for me at this point. I'm forever grateful for your unwavering love and acceptance of all my info dumps ranging from fish facts to robots. They will never end.

Table of Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives and Contributions	2
1.3	Thesis Outline	3
2	Background	4
2.1	Prosthetic Hands	4
2.1.1	Types of Prosthetic Hands	4
2.1.2	Evaluating Prosthetic Hands	6
2.1.3	Control Methods	9
2.2	Reinforcement Learning	10
2.2.1	Adaptive Switching	12
3	Continual Prediction Learning Techniques to Support Users' Grasp Selection on a Multi-Articulating Prosthetic Hand	14
3.1	Abstract	14
3.2	Introduction	15
3.3	Materials and Methods	17
3.4	Results & Discussion	20
3.5	Conclusion	23
4	The Development of the HANDi Hand V2	25
4.1	Abstract	25
4.2	Introduction	26
4.3	Designing the HANDi Hand V2	27
4.3.1	Design Specifications	27
4.3.2	Development Process	31
4.3.3	Sensorization	33

4.4	Testing the HANDi Hand V2	35
4.4.1	Grasp Force Testing	35
4.4.2	Functional Grasp Assessment	35
4.5	Application of Machine Learning	37
4.5.1	Adaptive Switching	37
4.5.2	Experimental Setup	38
4.5.3	Results and Discussion	40
4.6	Conclusion	43
5	Open-source Release of the HANDi Hand V2	44
5.1	Introduction	44
5.1.1	The History of Open-Source Hardware	45
5.1.2	Motivation and Objectives	45
5.1.3	Assembly Manual Outline	46
5.2	Methods	47
5.2.1	Considerations for User Experience	48
5.2.2	Steps to Design the Assembly Manual	55
5.3	User Validation	57
5.4	Future Work & Conclusions	59
6	Discussion, Future Directions, and Conclusions	60
6.1	Significance and Discussion of Contributions	60
6.2	Limitations of Current Work	62
6.3	Future Work	63
6.4	Conclusion	64
	Bibliography	65
	Appendix A: The HANDi Hand Assembly Manual	73

List of Tables

2.1	AHAP Results for Various Prosthetic Hands (adapted from Siegel et al. [51])	8
4.1	Design Specifications for the HANDi Hand V2	30

List of Figures

2.1	The Anthropomorphic Hand Assessment Protocol (AHAP) Tasks and Grasp Types (adapted from Siegel et al. [51])	7
2.2	The disparity between the number of EMG signals and the number of controllable functions in a myoelectric prosthesis. The user can control only one function at a time.	9
2.3	The RL framework. The agent takes an action after receiving information about the current state of the environment and receives a reward in the next time step [63].	11
3.1	<i>Left:</i> Updated variant of The HANDi Hand used in these experiments (original version released by Brenneis et al. [34]), showing digit joint motors D0–D5; this device includes updates to make the system more robust for longer-term machine learning experiments. <i>Right:</i> an example diagram of how EMG (muscle) signals are mapped to grasp in sequential switching.	17
3.2	GVF predictions could be learned in real time to predict the upcoming movement of different finger digits. (a) Representative example trial of prediction strength during learning shown for each finger joint (digits D0–D5), in terms of the learned predictions (solid lines), the computed return for the data stream (dotted lines), and the cumulant being predicted (shaded regions). (b) Mean absolute prediction error over all 30 trials (solid lines, with shading for the StdErr; 100-step sliding window averaging) shows repeatable and consistent prediction error decreases throughout learning.	21

3.3	GVF predictions at the beginning of the ongoing learning process show clear tracking behaviour; predictions <i>lag</i> the signals of interest they are predicting. After a period of learning, predictions can be seen to now rise <i>in advance</i> of the signals they are predicting; they are truly forecasting future movement based on the present state. This can be seen in the representative examples (a) showing prediction strength at the beginning of learning and (b) at the end of a learning trial period. Predictions are shown for each finger joint (digits D0–D5), in terms of learned predictions (solid lines), the computed return for the data stream (dotted lines), and the cumulant (shading).	22
4.1	The HANDi Hand V2 with palm cover removed, annotating key design improvements over the V1 design. Supplementary video available at https://www.youtube.com/watch?v=PT6Ybdv_29o	28
4.2	The flexion of the finger uses a single zip-tie. One end is attached to the inside of the fingertip, while the other end is secured using a setscrew to a 3D-printed servo spool.	32
4.3	The intermediate phalanx of the thumb is directly mounted to the in-thumb flexion servo horn. Rotation of this servo directly actuates the intermediate phalanx and simultaneously causes the zip-tie to wrap around the servo housing for distal-phalanx flexion. Thumb rotation is accomplished by direct drive from a palm-mounted rotation servo. .	32
4.4	Sensors included in the HANDi Hand V2 redesign.	34
4.5	An example of the wrist-mounted camera’s field of vision, with the target object distant and close to the hand during reaching.	34
4.6	Comparison of total GAS of HANDi Hand V2 (HH) with other 3D-printed hands. H: hook, SG: spherical grip, TP: tripod pinch, EG: extension grip, CG: cylindrical grip, DVG: diagonal volar grip, LP: lateral pinch, PP: pulp pinch, IP: index pointing/pressing, P: platform.	36
4.7	Comparison of partial GAS of redesigned HANDi Hand with other 3D-printed hands. Most 3D-printed hands struggled with certain grasps that had strict posture requirements.	37

4.8	Comparison of sequential and adaptive switching averaged over 15 trials. (Left) Total time and (Right) number of switches shown by iteration of the task. The dotted line shows the minimum number of switches possible using a static switching list. The dashed line shows the best possible switching performance with no user errors and a single switch at each switching event.	40
4.9	Cumulative number of switches over the course of a 3-minute trial. . .	41
5.1	An example of the IKEA manual style. There is little to no use of text [84].	48
5.2	An example of the Gundam manual style. Many “micro-steps” are shown in a single cell [86].	49
5.3	Example of a common mistake during the assembly process mitigated by warnings and colour cues.	50
5.4	Naming convention for the fingers (D1 to D5) and motors (ID10 to ID15).	51
5.5	An example of the printable 1:1 scale template. Users can place both 3D-printed parts and off-the-shelf parts on the paper to identify which ones are needed for the assembly step.	52
5.6	The three printing profiles used in the HANDi Hand.	54
5.7	Example of part orientation on the 3D-printer build plate.	54
5.8	Example of support removal instructions.	54
5.9	Comparison of an assembly step before and after annotation in Adobe Illustrator.	57
5.10	Example of a finished page in the assembly manual.	58

List of Symbols

Latin

N Newton

s Second

Machine Learning & Reinforcement Learning Terms

α Learning Rate: A parameter that determines the magnitude of weight updates during learning.

δ Temporal-Difference Error: The difference between the current estimate of the return and discounted value of the future return.

γ Discount factor: A parameter ($0 \leq \gamma \leq 1$) that defines the window of time of future predictions in a general value function (GVF).

λ Trace Decay Parameter: A decay factor controlling how fast eligibility traces fade over time.

π Policy: The behaviour or control rule used while selecting an action a given state S .

A Action: The command or motor action executed by an agent in the reinforcement learning loop after receiving a state and reward value.

C Cumulant: Any signal of interest used to build predictions as opposed to reward.

e Eligibility Trace: A short-term memory vector used to assign credit to recent states.

G Return: A discounted expectation of how the cumulant will accumulate in the future.

S State: A representation of the environment or sensorimotor space.

w Weight Vector: A learned parameter vector used to approximate general value functions.

$X(S)$ State Vector: A feature vector representation derived from the state (e.g. via tile coding).

Abbreviations

2D Two-dimensional.

3D Three-dimensional.

ADLs/ADL Activities of Daily Living.

AHAP Anthropomorphic Hand Assessment Protocol.

AI Artificial Intelligence.

API Application Programming Interface.

ARA/ARAT Action Research Arm Test.

BOM Bill of Materials.

CAD Computer-Aided Design.

CPCA Clinically Prescribable and Commercially Available.

DoF/DoFs Degree(s) of Freedom.

EDIL Environment-Dependent Incremental Learning.

EMG Electromyography.

FDM Fused Deposition Modeling.

FSR/FSRs Force Sensitive Resistor(s).

GAS Grasping Ability Score.

GVF General Value Function.

IDE Integrated Development Environment.

IMU Inertial Measurement Unit.

ML Machine Learning.

OSH Open-Source Hardware.

OSS Open-Source Software.

PLA Polylactic Acid.

PROS-TLX Prosthesis Task Load Index.

RL Reinforcement Learning.

SDK Software Development Kit.

SHAP Southampton Hand Assessment Procedure.

TD Temporal-Difference learning.

TPU Thermoplastic Polyurethane.

UDP User Datagram Protocol.

UX User Experience.

YCB Yale-CMU-Berkeley.

Glossary of Terms

Adaptive Switching A machine learning control strategy that uses General Value Functions to automatically reorder functions in a user switching list based on context and real-time user intent.

Autonomous Switching A step above adaptive switching where the system automatically switches functions on behalf of the user.

Bento Arm Open-source robotic arm compatible with the HANDi Hand.

Body-Powered Prosthesis Cable-driven prosthetic device operated manually via shoulder or body harness tension.

brachI/Oplexus Open-source software used to connect human control interfaces to robotic arms.

Direct Control Traditional myoelectric control scheme where a pair of antagonist muscle contractions controls a single movement directly.

General Value Function (GVF) A predictive reinforcement learning framework with value functions that store/represent expected future returns for arbitrary scalar signals beyond traditional reward.

HANDi Hand Humanoid Anthropometric Naturally Dexterous Intelligent Hand.

Horde Architecture A real-time learning system made of multiple independent GVF agents predicting different signals simultaneously across a shared state space.

Myoelectric/Powered Prosthesis A motorized prosthesis controlled via surface electromyographic (EMG) signals generated by muscle contractions in the residual limb.

Passive Prosthesis Non-actuated prosthetic limb used for cosmetic appearance, balance, or passive support.

Pattern Recognition Control Statistical machine learning control strategy that classifies multi-site EMG signal patterns into predefined movement classes.

Proportional Control Control strategy where the velocity or force of an actuator scales continuously with muscle contraction amplitude.

Residual Limb The remaining part of an arm or leg after limb loss.

Sequential Switching/Switched Control Control strategy where a user generates specific muscle signals to manually cycle through a fixed list of functions or grasp patterns.

Slicing A process in 3D-printing that turns a digital 3D model into machine instructions that a 3D printer can execute.

Tile Coding A feature-representation technique that segments continuous variables into overlapping grids (tiles) to provide a learnable input space that balances computation and representation.

“Grasping the concept of machine learning was hard. Getting machine learning to grasp anything was harder.”

Chapter 1

Introduction

1.1 Motivation

The human experience is governed by our ability to interact with the world. Our hands are a primary interface through which we express ideas, acquire knowledge, and assert our independence in life [1, 2]. When an individual experiences upper-limb loss, they are suddenly deprived of this sensorimotor pathway, leading to numerous physical and social complications [2–6]. For decades, a major goal of upper-limb prosthetic research has been to restore this lost functionality [7–9]. Furthermore, researchers have explored various methods to enhance the quality of life for prosthesis users, such as conducting patient-centered research, adding functionality, using new techniques in prosthetic design, and exploring novel manufacturing materials [2, 10–13]. In recent years, the field of rehabilitation robotics has also experienced rapid technological acceleration. Driven by advances in additive manufacturing and specialized actuator technology, modern commercial prostheses now feature independently actuated digits that enable dozens of complex grasp patterns (e.g., pinch grip, cylindrical grip, etc.) [14–16]. However, the development of hardware that strives to replicate the dexterity of the human hand exposed a large gap — the control channel between the user and their device remains a bottleneck [17–22].

As a clinical standard, sequential switching is offered to prosthesis users. To tran-

sition between grasp patterns, the user must generate specific muscle contractions to manually cycle through a static menu of grasp patterns. This control scheme is slow and cognitively taxing [23, 24]. Control burden and high cognitive load are primary reasons behind the frustrating reality: despite millions of dollars invested in the advancement of highly dexterous prosthetic hands, device abandonment rates among adults with upper-limb loss range between 23% and 44% [25–28]. A solution to this control problem lies in machine learning, specifically in the field of reinforcement learning. Yet, implementing these solutions reveals a second bottleneck: the lack of research hardware [29, 30].

Commercially available multi-articulating hands are often prohibitively expensive and proprietary, making them poorly suited for experimental research purposes [31, 32]. Conversely, existing open-source 3D-printed hands often lack the rich sensory streams required for continual real-time machine learning [33, 34]. Without such a platform, it is difficult to bridge the gap between these machine learning solutions and modern prosthetic hands. We suggest that developing an accessible, open-source, multi-articulating robotic hand may facilitate further exploration in the use of continual machine learning in the field of upper-limb prosthetics.

1.2 Objectives and Contributions

The objectives in the scope of this thesis are as follows:

1. To design and validate a highly sensorized, open-source robotic prosthesis (the HANDi Hand V2) specifically optimized for the use of real-time, continual machine learning research.
2. To apply reinforcement learning prediction techniques (General Value Functions) to dexterous finger actuation and automated grasp selection.
3. To demonstrate that adaptive switching may reduce the time and manual in-

teractions required for a user to operate a high-DOF device such as a multi-articulating prosthetic hand.

1.3 Thesis Outline

This thesis conforms to the University of Alberta’s Faculty of Graduate & Postdoctoral Studies’ chapter and paper-based formatting requirements, with papers presented verbatim.

Chapter 2: Provides relevant background information to aid in the understanding of the following chapters. It includes an overview of the role of the human hand in bi-manual manipulation and the profound impact of upper-limb loss. It covers the various types of prosthetic hands and previous work on machine learning control strategies. This chapter also lays the foundation for the research motivation of the subsequent chapters.

Chapter 3: Describes the fundamental machine learning techniques used to predict the movement of individual finger joints in a prosthetic hand. We establish the baseline capacity needed for more complex grasping tasks outlined in the next chapter.

Chapter 4: Outlines the technical development of the HANDi Hand V2 and its functional validation. Here, we demonstrate the first application of real-time adaptive grasp switching on a multi-articulating hand.

Chapter 5: Explores the comprehensive development of the HANDi Hand V2 assembly manual. This chapter describes the methodology to used craft highly accessible open-source material.

Chapter 6: Summarizes the key findings of the thesis and provides insights into future work.

Chapter 2

Background

2.1 Prosthetic Hands

The human hand is a remarkably capable system with many degrees of freedom (DoFs) and complex control [1, 2]. Its loss is therefore both physically and psychologically distressing for an individual, causing significant loss of productivity and agency [2–6]. Moreover, the loss of a hand warrants the inability to interact with the world, communicate with others, and act as a sensory receptor [7, 17]. Using robotic technology, such as artificial hands, can help replace or provide new alternatives to the lost functions of a person’s missing limb [9, 26].

2.1.1 Types of Prosthetic Hands

There are three main classes of upper-limb prostheses: body-powered, passive, and powered [12, 27]. Body-powered prostheses are driven by cables and are harnessed to the user’s intact side, such as the shoulder. Specific motion in the shoulder causes cable tension which opens and closes the terminal device [35]. Those who opt for a body-powered prosthesis often prefer the terminal device to be a hook due to its functionality and ease of use [36]. Passive prostheses are not powered and are typically designed to look like an intact arm. They are used when physical appearance, comfort, and cost-effectiveness are priorities for the user [37]. Finally, powered, or myoelectric

prostheses are designed to emulate the natural movements of the anatomical arm and hand with motors and electronic systems. Since these devices are designed to restore a lifelike appearance and a varying degree of function, they are a clinical standard for many users in countries that offer access to all types of upper-limb prostheses [38]. Moreover, the grasp force of these devices is often greater than that of their body-powered counterparts [12, 39]. These devices are typically connected to a user’s residual limb with a custom socket. Within the socket, surface electrodes detect intentional muscle contractions or electromyographic (EMG) signals which, through a controller, instruct the device’s motors to execute the desired actions [18].

Leveraging hardware advancements, modern designs of powered prostheses have evolved beyond simple grippers. State-of-the-art commercial myoelectric hands such as the i-Limb Quantum (Össur; Reykjavík, Iceland), the bebionic3 (Ottobock; Duderstadt, Germany), and the TASKA hand (TASKA Prosthetics; Christchurch, New Zealand) offer up to 36 pre-programmed grasp patterns (e.g. hook grip, pinch grip, etc.). Their individually actuated fingers allow the user to perform anthropomorphic grip actions similar to human hand motions [14]. However, commercially available prostheses are typically seen as a black box and cannot send low-level information from the actuators and sensors to researchers [26]. Additionally, the high cost of these devices can prevent their use in some research settings [31, 32].

To address this issue, the design and manufacturing of 3D-printed myoelectric hands emerged from a surge in the accessibility of 3D printers [12, 15]. A major advantage of 3D-printing for upper-limb prostheses is that their parts can be customized to suit the user’s needs, whether it be in clinical or research settings. Although the majority of 3D-printed hands used in clinical settings are developed for children, there is rapid growth of their benefits in research settings [5, 16]. Examples of 3D-printed prosthetic hands that have been used in research include the HACKberry Hand, the BEAR PAW, and InMoov [40–42]. As a precursor to the work presented in this thesis, researchers at the Bionic Limbs for Improved Natural Control (BLINC) Lab devel-

oped a low-cost, prosthetic hand capable of providing real-time sensory information for machine learning-based myoelectric control: The Humanoid Anthropometric Naturally Dexterous Intelligent (HANDi) Hand. This earlier model could report fingertip forces, angular position for individual finger joints, and visual information via an in-palm camera. The HANDi Hand V1 has 6 degrees of freedom, including individually actuated fingers plus separate thumb rotation and flexion [34, 43].

2.1.2 Evaluating Prosthetic Hands

Evaluating the efficacy of an upper-limb prosthetic hand requires a careful balance between its physical functionality and clinical relevance. Historically, evaluations for these devices has relied heavily on user-in-the-loop assessments, such as the Southampton Hand Assessment Procedure (SHAP) [44, 45], the Action Research Arm Test (ARA or ARAT) [46, 47]. Although these protocols are valuable for assessing the interaction between a user and their device, they introduce significant confounding variables when specifically evaluating hardware performance. For example, a participant’s cognitive ability and learning curve heavily influence the overall assessment. Consequently, an exceptionally skilled user can sometimes compensate for a poorly designed prosthesis. To truly isolate and evaluate the physical functionality of a prosthetic hand, researchers require an objective protocol that removes human variability.

The Anthropomorphic Hand Assessment Protocol (AHAP) was established as a validated, bench-top testing framework [48–50]. It evaluates a robotic hand’s ability to execute 26 distinct tasks across 25 common household objects from the Yale-CMU-Berkeley (YCB) Object and Model Set. These tasks cover the 8 most common grasp types (hook, spherical, tripod pinch, extension, cylindrical, diagonal volar, lateral pinch, and pulp pinch) and 2 non-grasping postures (platform, index pointing/pressing). These objects and grasp types can be seen in Figure 2.1. An advantage of the AHAP framework is its distinction between the grasping and maintenance phases of

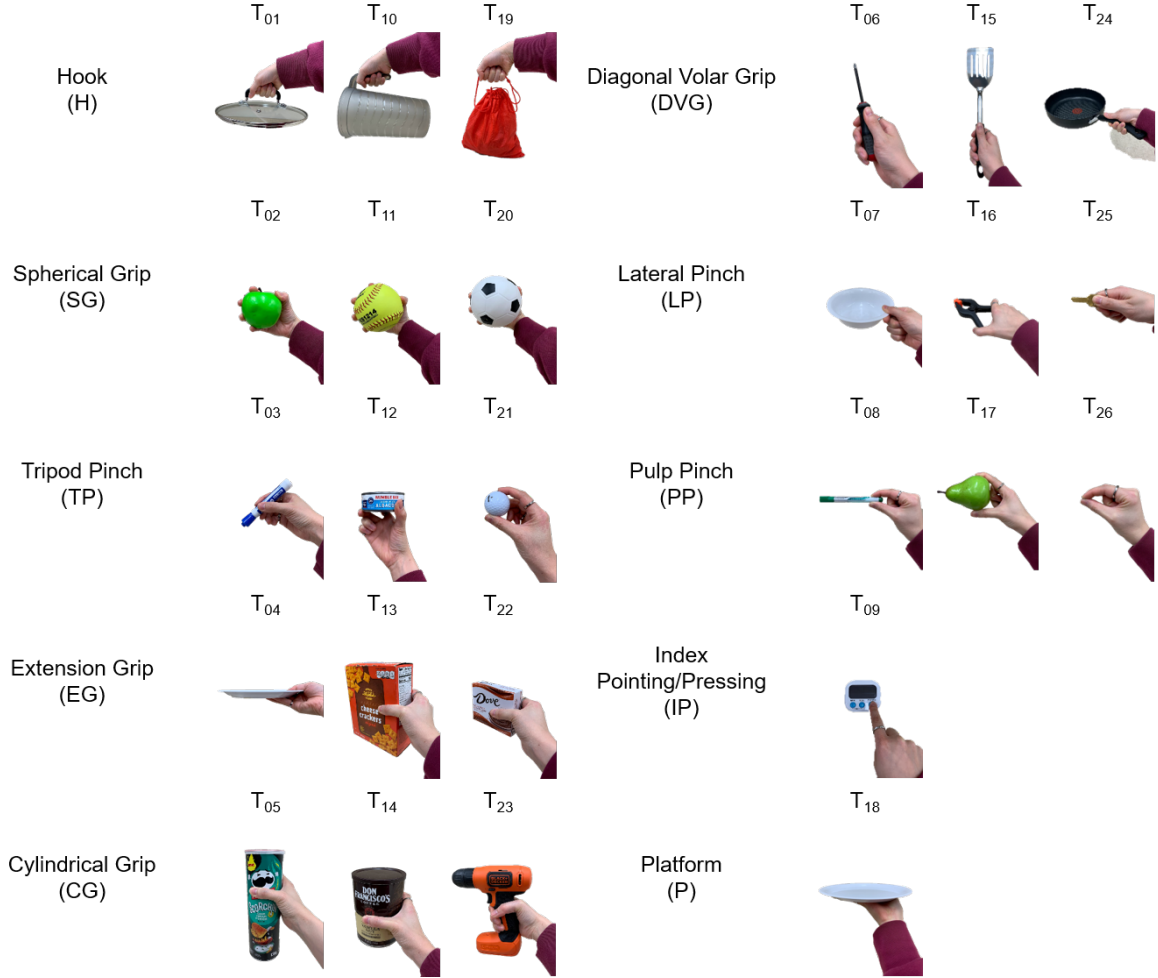


Figure 2.1: The Anthropomorphic Hand Assessment Protocol (AHAP) Tasks and Grasp Types (adapted from Siegel et al. [51])

hand manipulation. The grasping phase focuses on the hand’s ability to close its digits around the object and establish specific contact points with the palm facing up. The maintenance phase requires the hand to flip the object 180 degrees and hold the object for exactly 3 seconds without slipping. By evaluating these phases separately, the AHAP provides a distinct breakdown of the hand’s mechanical design. For instance, failure during the grasping phase can indicate poor finger kinematics while a failure during the maintenance phase can expose insufficient friction on the fingertip pads.

The AHAP has been used to benchmark two classes of anthropomorphic hands:

Clinically Prescribable and Commercially Available (CPCA) hands and open-source 3D-printed hands. Table 2.1 combines the Grasping Ability Scores (GAS) and phase-specific scores across various CPCA and 3D-printed hands [48, 51].

Table 2.1: AHAP Results for Various Prosthetic Hands (adapted from Siegel et al. [51])

Prosthetic Hand	Grasping Phase	Maintaining Phase	Overall GAS
CPCA Hands			
Psyonic Ability Hand	97%	94%	95%
Össur i-Limb Quantum	85%	96%	90%
RSL Steeper BeBionic V3	77%	79%	77%
3D-Printed Hands			
BEAR PAW (Pediatric)	77%	57%	68%
HANDi Hand (V1)	77%	53%	66%
IMMA Hand	77%	37%	57%
HACKberry Hand	64%	36%	53%
Limbitless	63%	37%	50%
InMoov	57%	40%	49%
Dextrus v2.0	61%	34%	48%

These studies reveal a noticeable performance disparity between the CPCA and 3D-printed devices. While 3D-printed hands achieve decent grasping phase scores, their performance drops during the maintaining phase. Key design vulnerabilities causing these challenges include inadequate material friction, motor backlash, and poor physical dimensions of the hands. Despite these limitations, 3D-printed hands remain important for research settings due to their cost-effectiveness. To bridge this functionality gap without sacrificing accessibility, future open-source hands should prioritize low-cost modifications such as adding rubber fingertip coatings, integrating stronger actuators, and optimizing finger/thumb kinematics to better match human grasp patterns.

2.1.3 Control Methods

The development of both commercial and research-specific multi-articulated prosthetic hands begs the question of how these devices are controlled. Control is defined as the link between the user’s intent and the motion of their prosthesis. Often, as shown in Figure 2.2, the controllable functions or DoFs in the myoelectric arm outnumber the available muscle sites, making control a challenge to solve [17, 18, 20]. For instance, an individual with an amputation at the elbow may only have EMG recording sites from the biceps and triceps. The more function the user has lost, the more function must be compensated by their device [8, 18, 20].

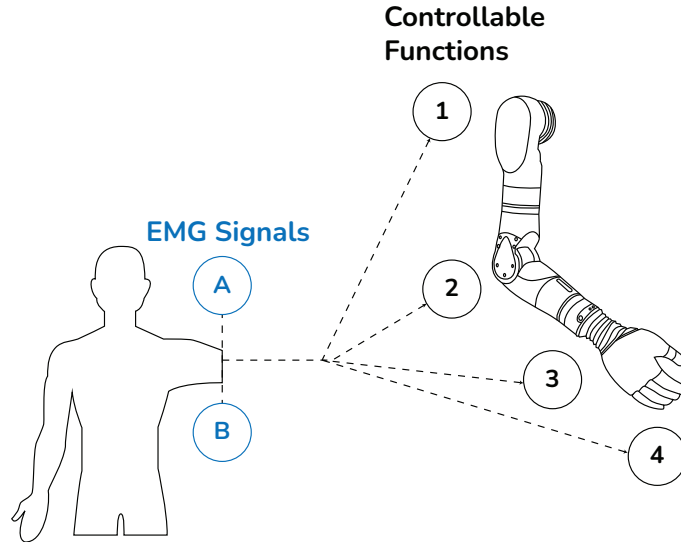


Figure 2.2: The disparity between the number of EMG signals and the number of controllable functions in a myoelectric prosthesis. The user can control only one function at a time.

Conventional control strategies include on/off control and proportional control [23, 52]. Both strategies activate the opening and closing of a hand with EMG signals from two different residual muscles to control opposing actions [24]. This control strategy allows the user to switch through a list of functions, such as grasp patterns. While robust, this method is not intuitive since users must consciously activate specific muscles for each movement. Kannenberg et al. compared different multi-articulated myoelectric prostheses and found that providing a greater number of grasp patterns

did not necessarily translate to a greater ease of use [38]. Challenges arose in the intuitiveness and time taken to switch to different grasp patterns [5]. Davidson et al. also expressed that adding more DOFs to a device could increase its complexity, leading to control challenges [53]. As such, the number of functions available to the user is often reduced to a clinically specified list [54, 55]. Although this solution allows the user to control their myoelectric arm, it sacrifices function for ease of control [18].

Machine learning (ML) control strategies have been implemented to make myoelectric prostheses more intelligent and able to predict the user’s intended movements. EMG pattern-recognition-based control has been developed to control the prosthesis more naturally using statistical machine-learning approaches. It extracts patterns from multiple EMG sites and learns to classify them into movements such as hand open, hand close, and no movement [18, 24, 56]. However, pattern recognition algorithms are limited due to the need for them to be trained offline with previously recorded signals. Myoelectric arms using pattern recognition must be regularly recalibrated to account for degradation in signal recording, change in electrode position, and differences in signal strength between training samples [25, 57, 58]. Although a user may achieve high offline classification accuracy during calibration (e.g. $> 90\%$), it does not necessarily correlate to better online functional performance. Without recalibration, the device may act inappropriately, causing frustration for the user as their intentions no longer produce repeatable movements.

2.2 Reinforcement Learning

Despite these advancements in prosthetic hand control techniques in the past 60 years, the mean rate of rejection of powered upper limb prostheses remains unfavourably high [27, 28]. Users may reject their upper-limb prosthesis for a variety of reasons, one being the lack of reliable control [25, 26, 59, 60]. The limitations of the clinical standard of myoelectric control (static pattern recognition, where a model is trained,

frozen, and deployed) have fueled the exploration of ways that a model could continuously adapt to both the user’s behaviour and the environmental context [61, 62].

Using reinforcement learning (RL) techniques in myoelectric control is a potential solution. RL involves an *agent* (learner) who learns a *policy* (what actions to take in a given state) by interacting with an *environment* to maximize a numerical *reward* signal [63]. The RL framework presented by Sutton and Barto can be seen in Figure 2.3. Within the framework, there is the *reward*, which is the immediate appeal of the current state, which differs from the *value*, which is the expected appeal over the long run, starting from the current state. A *reward function* maps states to rewards given a policy to follow. On the other hand, a *value function* maps states to the expected sum of future rewards when given a policy.

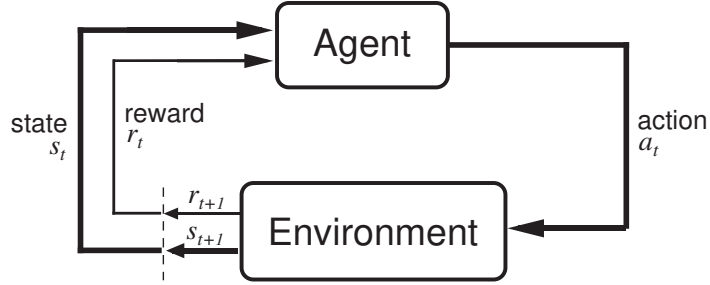


Figure 2.3: The RL framework. The agent takes an action after receiving information about the current state of the environment and receives a reward in the next time step [63].

General Value Functions (GVFs) can be used to build up predictions about a specific signal of interest, not just reward, to inform an agent’s next actions [64, 65]. For example, if the question for the agent is “When will the user move the elbow joint?”, the signal would be the movement of the elbow. As the elbow is moved over time, the prediction of this signal would increase before the elbow moves.

Another component of RL is the learning algorithm. Temporal-difference (TD) learning algorithms allow the agent to learn from experiences as they happen in real-time [63, 66]. TD(λ) is a type of TD learning algorithm which uses a trace decay parameter λ . This parameter allows the algorithm to give credit to previous states

given the reward earned in the current state. Combining $TD(\lambda)$ and GVFs gives the control system a way to predict how the user operates their upper-limb prosthesis in the environment in real-time.

2.2.1 Adaptive Switching

Adaptive Switching is a myoelectric control strategy that uses reinforcement learning techniques. It describes the ability to automatically switch between functions (e.g. shoulder, elbow, wrist, and hand movements) based on the user’s intent or context of the task [55]. In the taxonomy of incremental learning strategies presented by Campbell et al., adaptive switching falls under Environment-Dependent Incremental Learning (EDIL) [58]. EDIL online models have been shown to perform well without initial calibration data, making them suitable for the stochastic nature of using a myoelectric prosthesis [55, 67]. Adaptive switching was first introduced to upper-limb prosthetics by Pilarski, Dawson, et al. [68]. Their work suggests that predicting the next joint a user will control could reduce switching effort and improve control efficiency. To extend this study, Edwards et al. [55, 69] applied adaptive switching to the real-time operation of a robotic arm by a prosthesis user. Adaptive switching uses GVFs and the $TD(\lambda)$ algorithm to continuously rank joints in order of prediction strength. Algorithm 1 describes the algorithm used in adaptive switching.

Their studies demonstrate that although the system had an initial period of learning, typically only one switch was required to move to the next desired joint. If made more intuitive, prosthesis users could be encouraged to use more controllable functions in their device rather than having to deliberately reduce their options. Due to the nature of the device continually learning, it removes the need for recalibration by clinical staff, which proves the potential to be clinically viable over time [18, 70]. With time, there will be a mutual adaptation process where the system learns the user’s behaviour, and concurrently, the user adjusts to the machine’s learning patterns [55, 71]. However, it is important to consider that successful co-adaptation requires

Algorithm 1: Adaptive Switching [55]

```
Initialize: 1 GVF per joint, switch_list, active_joint_index, switching
Repeat
  observe  $EMG_C, EMG_S, s$            // Observe EMG control and switch signals, and state
  get predictions  $p$  from GVFs given  $s$  // Compute predictions
  if not switching then
    update switch_list based on magnitude  $p$ 
    // Update switch list
  if  $switch(EMG_S)$  then
    switching  $\leftarrow$  true           // Switching flag set to true
    increment active_joint_index      // Switch to the next joint in the switching list
  move_joint(switch_list[active_joint_index],  $EMG_C$ )
  // Move active joint according to EMG control signals
  if moved then
    switching  $\leftarrow$  false         // Switching flag set to false
  for all  $gvf$  in GVFs do
     $gvf.update(s)$                   // Update GVF and learner using TD learning
```

careful tuning of the agent’s learning rate. Over-adaptation by the machine may force the user to struggle to “chase” the model’s intention, while under-adaptation may force the user to adapt unnatural muscle contractions, accelerating fatigue [58, 72]. A matching adaptation rate between the user and the system results in a user’s reliable expectations of how the system will behave [73]. Additionally, closed-loop sensory feedback should be provided, whether it be visual, auditory, or haptic. Without feedback during the co-adaptation process, the user would not be able to correct their inputs, leading to compounding error [74].

The theoretical foundations outlined in this chapter highlight the use of reinforcement learning techniques, specifically General Value Functions (GVFs) and adaptive switching. While previous implementations focused on larger arm movements [55, 75], the predictions needed for successful grasp selection remain unexplored. To bridge this gap, Chapter 3 investigates how a Horde of GVFs could be trained through real-time temporal-difference learning to predict movement in individual finger joints. This chapter establishes the foundation necessary to extend adaptive switching to grasp selection.

Chapter 3

Continual Prediction Learning Techniques to Support Users’ Grasp Selection on a Multi-Articulating Prosthetic Hand

Chapter 3 presents “Continual Prediction Learning Techniques to Support Users’ Grasp Selection on a Multi-Articulating Prosthetic Hand”, originally published in the Proceedings of the 38th Canadian Conference on Artificial Intelligence, in 2025 [76].

3.1 Abstract

Artificial intelligence methods have shown promise for enhancing human-machine interaction, and also for accelerating progress in the field of robotics. Here, we discuss an application case that brings together these two areas of impact: human interaction with a robotic prosthesis. We present preliminary findings that suggest how continual prediction learning methods from the field of reinforcement learning can be used to streamline user selection of different grasping movements of a dexterous robotic

hand. Using powered prosthetic hands is particularly challenging for people with limb differences (e.g., amputations) due to the disparity between the many possible grasp patterns a device can support and the limited number of control channels provided by the human body of someone who has lost a limb. Previous work introduced a machine learning technique called *adaptive switching* that demonstrated how a robotic prosthesis could learn to predict the modes or functions a user would deploy during their control interactions, and continually optimize the user’s control interface using this learned predictive information. Adaptive switching has, to date, only been tested on a desk-mounted arm prosthesis involving simple open/close hand movement, and has not been extended to the more complicated, contextual case of grasp selection on a prosthetic hand. Our work contributes a first look at how the prediction-learning foundation of adaptive switching can be extended to support grasp selection on a sensorized multi-articulated prosthetic hand with multiple grip patterns. Specifically, we show how a cohort of generalized value functions, learned in real time via processes of temporal-difference learning, can predict the future activity of individual fingers in a robotic hand. These predictions were shown to accurately anticipate a temporal (cascading) pattern of finger movements, which demonstrates the capacity for grasp-related adaptive switching. Our findings suggest the scalability of this approach to predict more complex grasp types, and we recommend moving forward to user-in-the-loop studies of adaptive switching in daily-life reaching and grasping tasks.

3.2 Introduction

Individuals with upper limb loss, for example, through amputation, can experience both physical and psychological distress, causing significant loss of productivity [3, 4]. Robotic technology, such as artificial hands, can help replace the functions of a person’s missing limb [7–9, 25]. As such, robotic upper limb prostheses are designed to emulate the natural movements of the anatomical arm and hand with motors and

electronic systems (e.g., Fig. 3.1, left). These movements are controlled by intentional muscle contractions or electromyographic (EMG) signals [18, 24]—electrical signals of muscle activation that are typically sampled non-invasively at the skin by way of metal electrodes. As a standard clinical approach, with what is called *direct control*, one pair of surface electrodes is placed over two muscle sites on a user’s residual limb to control movement of a single prosthesis joint or function (e.g., hand open/close). In this case, one pair of antagonistic muscles can be used to control one joint’s movement, and what is called *switched control* is often employed to allow a user to toggle this control activity between a pre-determined set of movements—e.g., to sequentially switch between moving their hand, wrist, and elbow according to a fixed switching list (Fig. 3.1, right). A common issue with upper limb prosthetics is that the number of controllable functions in the device often outnumbers the available muscle sites [17–21]. As such, prosthetists typically reduce the number of functions given to a user to only a subset of those available [55]. Sequential control options are practical, but increase cognitive demand and sacrifice device usability [18].

A promising area of innovation in sequentially switched prosthesis control methods is *adaptive and autonomous switching* [55, 68, 77]. Based on continual machine learning methods, it provides the ability to predict and automatically re-order or switch between functions (e.g. shoulder, elbow, wrist, and hand movements) of a prosthesis based on the user’s intent or context of the task [55]. Adaptive switching continuously ranks joints in order of learned, state-dependent prediction strength. These predictions are used to make real-time changes to the order of options in a user’s sequential switching list [55], and in some cases, automatically switch between options in the list on behalf of the user [77]. Continual learning is especially important for long-term deployments such as in prosthetic devices, which can benefit from learning and adapting to user behaviour over time. Prior work on adaptive switching with actual prosthesis users indicated that predicting the next joint a user will control could lower the time and cognitive load of joint switching [55]. These earlier experi-

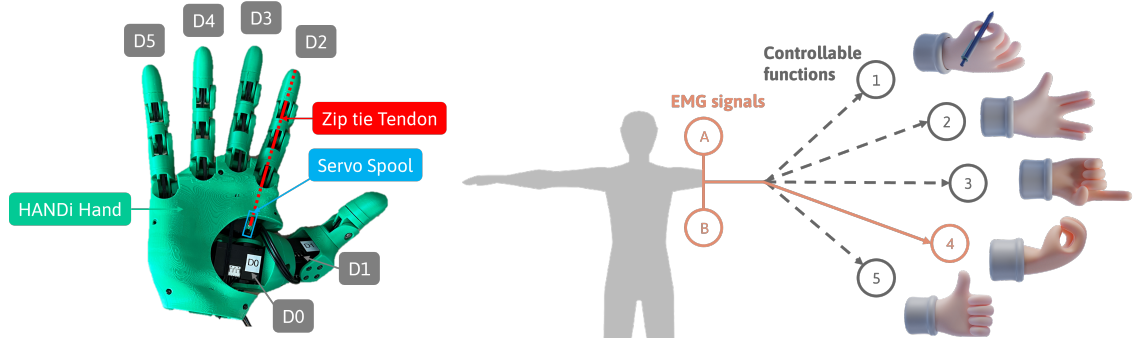


Figure 3.1: *Left:* Updated variant of The HANDi Hand used in these experiments (original version released by Brenneis et al. [34]), showing digit joint motors D0–D5; this device includes updates to make the system more robust for longer-term machine learning experiments. *Right:* an example diagram of how EMG (muscle) signals are mapped to grasp in sequential switching.

ments in fact represent one of the first times continual machine learning was deployed to improve devices in the setting of rehabilitation and assistive medical technology. However, this prior work focused on larger, gross motions of a robotic arm: e.g., a user sequentially controlling robot wrist, hand, arm, and shoulder joints to move objects across a barrier [55]. *In the present work, we aim to understand if adaptive switching might be extended to finer-grain manipulation.* Specifically, we consider switching between the myriad grasping modes now available with modern prosthetic hands that incorporate the coordinated and situation-appropriate movement of multiple robotic fingers.

3.3 Materials and Methods

Adaptive switching leverages foundational concepts from the field of reinforcement learning (RL) [63], specifically *general value functions* (GVFs) [65]. As an advantage over other prediction model classes that deal with time-series data, GVFs have been shown to effectively and incrementally in memory and samples allow an agent (e.g., a robot) to build up predictions about any signal of interest; this extends their use beyond the specific signal of reward as is common in RL problem domains [64, 65, 78]. The GVF’s prediction question is governed by three parameters: the signal of

interest or cumulant (C), the policy (π), which defines the agent’s behaviour while making the prediction, and the discount factor, ($0 \leq \gamma \leq 1$), which determines how far into the future the prediction spans. The return, or sum of future signals, is given by Equation 3.1 and represents an expectation of how signals will accumulate in the future. GVF’s allow us to define a prediction problem by posing a computational question about the environment.

$$G_t = \sum_{k=0}^{\infty} \gamma^k C_{t+k+1} \quad (3.1)$$

Temporal-difference (TD) learning algorithms allow an agent to learn approximate GVF’s from experiences as they happen in real time [65]. We here use TD(λ), a type of TD learning algorithm which uses an eligibility trace decay parameter λ . Three essential equations for the TD(λ) algorithm are the temporal-difference error (δ), eligibility traces ($\mathbf{e}_t$), and weight vector update ($\mathbf{w}_{t+1}$) equations which can be seen in Equations 3.2, 3.3, and 3.4.

$$\delta = C_{t+1} + \gamma \mathbf{w}_t^\top \mathbf{x}(S_{t+1}) - \mathbf{w}_t^\top \mathbf{x}(S_t) \quad (3.2)$$

$$\mathbf{e}_t = \min(\mathbf{e}_{t-1}\lambda + \mathbf{x}(S_t), 1) \quad (3.3)$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha \delta_t \mathbf{e}_t \quad (3.4)$$

In addition to question parameters, there are answer parameters which can change how the agent answers the GVF question. These include the learning rate (α), the trace decay parameter (λ), and the function that we can use to construct the state or environment ($\mathbf{x}(S)$). By combining TD(λ) and GVF’s, a system can, for example, frame and ask questions about a user’s behaviour and learn to answer them in real time [55]. Modern multi-articulating prosthetic hands have many signals from the motors, the user, and sensors that might be targets of predictions. Since GVF’s can only build predictions of one signal of interest, we employ the Horde architecture [65].

This is an RL framework that comprises multiple GVF agents to learn and maintain predictions simultaneously [65]. Each GVF has a cumulant, discount factor, and policy while sharing the same state space. For readers to better understand how the GVF predictions can be used to support the adaptation of a user’s switched control interface, we here present the adaptive switching algorithm of Edwards et al. [55] in Algorithm 2 for easy reference.

Algorithm 2: Adaptive Switching [55]

```

Initialize: 1 GVF per joint, switch_list, active_joint_index, switching
Repeat
  observe  $EMG_C, EMG_S, s$            // Observe EMG control and switch signals, and state
  get predictions  $p$  from GVFs given  $s$    // Compute predictions
  if not switching then
    update switch_list based on magnitude  $p$ 
    // Update switch list
  if switch( $EMG_S$ ) then
    switching  $\leftarrow$  true           // Switching flag set to true
    increment active_joint_index      // Switch to the next joint in the switching list
  move_joint(switch_list[active_joint_index],  $EMG_C$ )
  // Move active joint according to EMG control signals
  if moved then
    switching  $\leftarrow$  false         // Switching flag set to false
  for all gvf in GVFs do
    gvf.update( $s$ )                   // Update GVF and learner using TD learning

```

As an experimental platform to demonstrate real-time learning, we created a custom extension of an open-source multi-articulating prosthetic hand, the HANDi Hand [34] (Fig. 3.1). Our modified hand uses six Dynamixel XL330-M288-T servo motors—one for each finger and two for thumb rotation and flexion. Each motor can send real-time data such as its position, velocity, temperature, and more. Communication between the computer and the motors was established using the Dynamixel Software Development Kit (SDK). Position and velocity of each motor were chosen to represent the system’s state space. This state space was discretized using a simple binned implementation of tile coding [63], reducing computational complexity, which is important in real-time continual learning. Each motor produced a flattened one-hot encoding of length 900 of its position and velocity values, which when concatenated across motors, resulted in a single vector representation $\mathbf{x}(S)$ of length 5400 with six

active bits. Representing the state in this manner clarifies how changes in the motors’ position and velocity influence the predictions, while also preserving the connection between these two properties. We conducted a basic experiment to demonstrate the foundations for adaptive grasp switching, wherein grasping was defined in terms of individual finger movements. A single GVF was assigned to each finger joint motor (D0–D5, Fig. 1), posing the computational question: “how much flexion will a given finger motor demonstrate in the near future?” From these predictions, we aimed to determine which finger would be most likely to move next, a precursor to predicting which grasp type a user may need. The behaviour policy (π) was effected as a sequential movement consisting of each finger closing and opening, one after the other in a cascade, and repeated for 3000 time steps (approximately 380s of real time). This allowed each finger to experience motion approximately 30 times. The cumulant (C) of each GVF was straightforward, being the flexion of a finger motor in terms of its positional encoding, normalized to values between 0 and 1. The discount factor (γ) was set to an empirically selected constant value of 0.9 and the trace decay parameter (λ) was set to 0.9. In total, 30 trials were run.

3.4 Results & Discussion

Figure 3.2a shows a representative plot of each finger motor prediction value (D0–D5) at the end of learning. The solid lines show the normalized prediction strength of each motor, whereas the dotted lines show the normalized true return values, computed post hoc. The GVF’s cumulant, the motor’s position, can be seen in the shaded regions. True returns and predictions are normalized by timescale $(1 - \gamma)$ to allow easier visual comparison between predictions and cumulants. The prediction strength positively correlates with the cumulant, suggesting that the horde of GVFs captures the dynamic behaviour of the hand during movement transitions from finger to finger. This behaviour is consistent over many grasp cycles, and independent trials, indicating

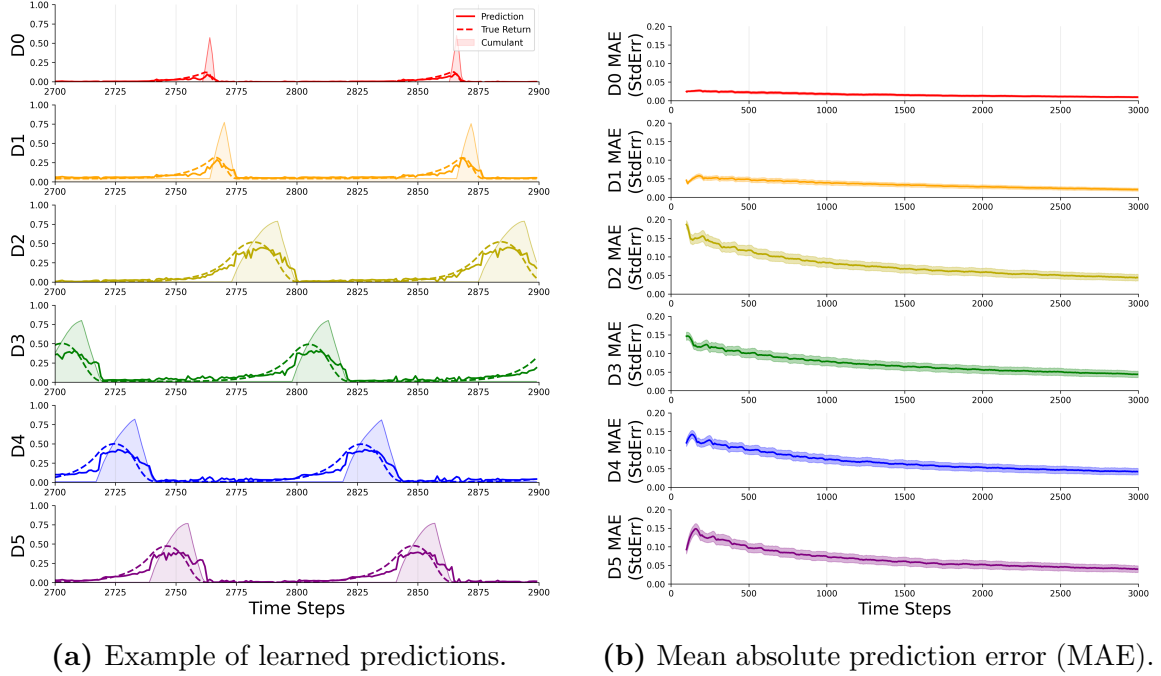
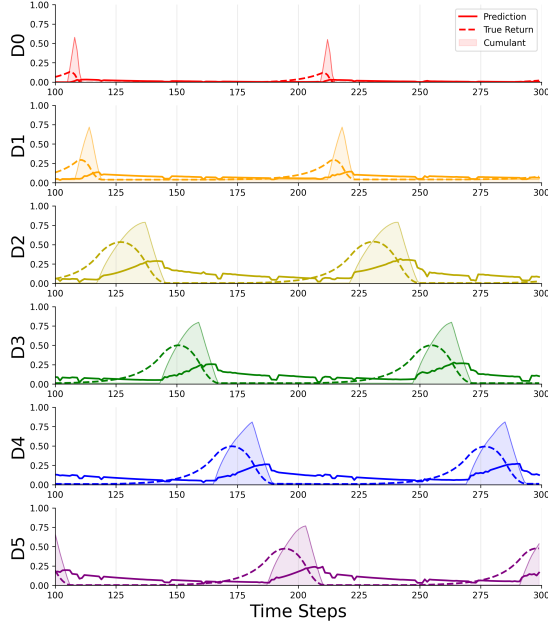


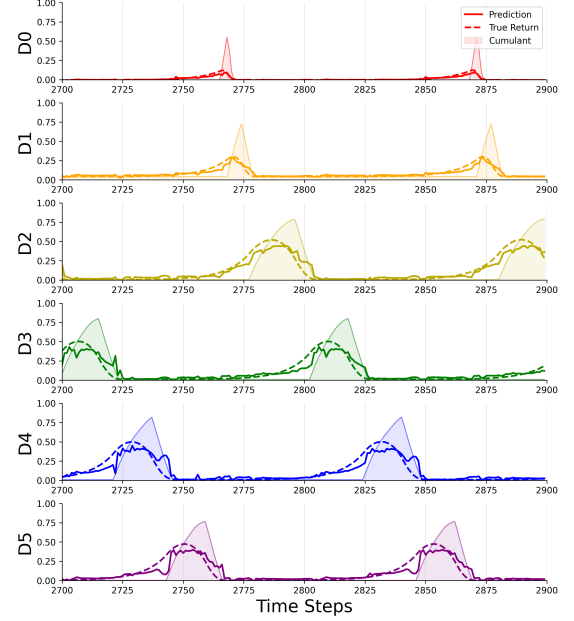
Figure 3.2: GVF predictions could be learned in real time to predict the upcoming movement of different finger digits. (a) Representative example trial of prediction strength during learning shown for each finger joint (digits D0-D5), in terms of the learned predictions (solid lines), the computed return for the data stream (dotted lines), and the cumulant being predicted (shaded regions). (b) Mean absolute prediction error over all 30 trials (solid lines, with shading for the StdErr; 100-step sliding window averaging) shows repeatable and consistent prediction error decreases throughout learning.

that learning is reliable and stable as the hand is used. Finally, the predictions can capture the difference between active and inactive states, confirming that the chosen state representation effectively separates different phases of movement. One key finding was that the system could learn how to anticipate movement before sensing motor activity. Comparing the beginning (Fig. 3.3a) to the end of learning (Fig. 3.3b) in Figure 3.3, predictions accumulated well before the finger moved. This suggests that, with time, the system could be extended to learn patterns of user activity whether it be individual finger movement or grasp patterns.

Figure 3.2b shows the decrease in mean absolute prediction error (MAE) over 30 experimental trials, with shaded regions indicating the standard error of the mean. These findings are important for two reasons. First, these preliminary results indi-



(a) Learned predictions at the beginning of a trial.



(b) Learned predictions at the end of a trial.

Figure 3.3: GVF predictions at the beginning of the ongoing learning process show clear tracking behaviour; predictions *lag* the signals of interest they are predicting. After a period of learning, predictions can be seen to now rise *in advance* of the signals they are predicting; they are truly forecasting future movement based on the present state. This can be seen in the representative examples (a) showing prediction strength at the beginning of learning and (b) at the end of a learning trial period. Predictions are shown for each finger joint (digits D0–D5), in terms of learned predictions (solid lines), the computed return for the data stream (dotted lines), and the cumulant (shading).

cate that adaptive switching can likely be extended from the setting of larger limb movements to predicting the configuration and smaller movements of individual fingers. Imagine, for instance, a user rapidly switching between appropriate finger chord patterns when playing a piano, then voluntarily contacting their muscles to modulate hand shape as they play. Second, our results show that finger movement prediction can approximate the true return when incorporating just motor position and velocity state information. This is promising, as extending the system’s state space with sampled biosignals, inertial measurement sensor data, fingertip force sensors, finger joint angle sensors, and other early indicators of context and use, can likely allow a full transition from desk-mounted adaptive switching to mobile cases where the device is affixed to the human body during tasks of daily living, i.e., real world prosthesis use. Based on the findings herein, our team is now extending the present study to complex grasp patterns that involve simultaneous multi-finger actuation with contextual tasks, advanced state representations, and user-in-the-loop control.

3.5 Conclusion

The use of GVFs to predict motor behaviour in prosthetic hands showcases the potential for using AI techniques in assistive rehabilitation technologies intended to support humans in their day-to-day activities. It is a straightforward, yet concrete, example of aligning AI systems to human behaviour and goals. In this work we contributed new preliminary evidence that continually learned GVFs can form accurate predictions of the sequential movement of fingers of a prosthetic hand. This work serves as a foundation for translating previous adaptive switching algorithms to a dexterous manipulation setting, and eventually to real-world contextual prosthesis use. Further, in sharing early findings on a new variant of the open-source HANDi Hand robotic platform, this work contributes further insight and awareness on environments for real-world AI studies that may be of use to other researchers outside the domain of

biomedical robotics. Taken as a whole, this study and the findings presented here add to the body of evidence on how machine learning methods can be deployed in a continual learning setting during interactions with human users, and advance our ability to create assistive technologies, like prosthetics hands, that can operate less like conventional tools and more like intelligence partners in achieving our goals.

Chapter 4

The Development of the HANDi Hand V2

Chapter 4 presents “The HANDi Hand V2: Enhancing a Multi-articulating Prosthetic Hand for Continual Machine Learning Research”, submitted to and in review in the BMC Biomedical Engineering Journal in 2026.

4.1 Abstract

Background: The advancement of upper-limb prosthetic devices is currently limited by complex control requirements and a lack of accessible, durable research platforms. This work addresses both barriers by presenting the redesign of the Humanoid, Anthropometric, Naturally Dexterous Intelligent (HANDi) Hand, an open-source platform optimized for machine learning research.

Methods: We describe multiple mechanical enhancements to the hand’s design, including the integration of highly sensorized actuators, which improve the platform’s durability for continual learning and expand its sensory feedback capabilities. The functional capabilities of the hand are benchmarked using the Anthropomorphic Hand Assessment Protocol (AHAP). Furthermore, we use the HANDi Hand V2 to demonstrate the first application of machine-learned adaptive switching for grasp selection.

Results: Using the AHAP, the hand yielded an overall Grasping Ability Score of

79%, demonstrating narrowing the performance gap between 3D-printed and commercial prosthetic hands. In the real-time predictive learning experiment, adaptive switching significantly reduced control effort, dropping the total switching duration from a 45-second baseline to approximately 33 seconds. Our findings suggest that, when deployed on a multi-articulating hand, predictive continual learning methods can create a scalable path towards offering users a more diverse selection of grasps without a proportional increase in control effort.

Conclusions: The functional enhancements and the machine learning viability demonstrated in this work show that the HANDi Hand V2 provides a robust, accessible platform for advancing upper-limb prosthetic research. By shifting the switching burden from the user to the device, this work supports an ongoing paradigm shift where rehabilitation technologies can transition from reactive tools into intelligent partners that enhance user agency and reduce device rejection.

4.2 Introduction

State-of-the-art commercial hands such as the i-Limb Quantum (Össur; Reykjavík, Iceland) and the bebionic3 (Ottobock; Duderstadt, Germany) are impressive feats of scientific innovation. Offering up to 36 pre-programmed grasp patterns (e.g. spherical grip, precision pinch), these devices’ extended functionality results largely from individually actuated fingers with movements similar to a biological human hand [14].

Despite their potential, the high cognitive burden from manual grasp selection remains a primary cause of device rejection in clinical use, cited to be between 23–44% for adult users [25–28]. As such, clinicians typically reduce the number of functions given to a user to only a subset of those available [54]. To address these challenges, machine learning (ML) control strategies, such as pattern recognition [17, 18, 24], and continual learning methods like adaptive switching [55, 68, 69, 77], have been developed to enable individualized control solutions.

However, researchers still face limited access to affordable and robust sensorized platforms for testing these novel ML systems. Using clinical-grade, commercial prostheses is also not always an option for researchers due to factors such as costs and global availability. Previous work by Brenneis et al. introduced the Humanoid Anthropomorphic Naturally Dexterous Intelligent (HANDi) Hand, designed as a platform to facilitate research on ML control strategies [34, 43]. Other examples of 3D printed prosthetic hands that have been used in a research setting include the HACKberry Hand, the BEAR PAW, and InMoov [40–42]. Although these open-source projects see continued use, they suffer from frequent hardware obsolescence, such as the discontinuation of the radio-controlled motors in the original HANDi Hand. Furthermore, many existing 3D-printed hands lack both the rigorous benchmarking and rich sensorization necessary for ML-specific research.

We present an extension of the HANDi Hand (V2), featuring advances in motor technology and mechanical design that enhance its durability and performance for continuous-use ML applications (Fig. 4.1). With this redesign, we provide a verified functional comparison against both commercial and open-source alternatives via the Anthropomorphic Hand Assessment Protocol (AHAP) [48]. Finally, we show the first demonstration of the HANDi Hand V2’s capabilities in a continual machine learning context with adaptive switching. The HANDi Hand V2 fills a critical gap in the translational research pipeline as an accessible and highly sensorized platform.

4.3 Designing the HANDi Hand V2

4.3.1 Design Specifications

Table 4.1 summarizes the design specifications for the HANDi Hand V2 and provides a comparison with the original 2017 version. The redesign preserves core anatomical proportions, six degrees of freedom (DOFs), and independent finger actuation. Furthermore, V2 maintains full compatibility with the Bento Arm research platform

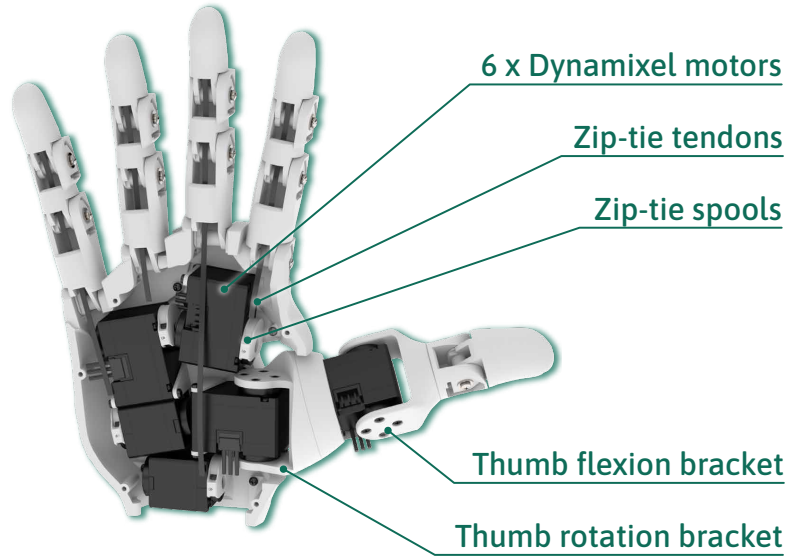


Figure 4.1: The HANDi Hand V2 with palm cover removed, annotating key design improvements over the V1 design. Supplementary video available at https://www.youtube.com/watch?v=PT6Ybdv_29o

[75]. Integrating Dynamixel XL330-M288-T motors facilitates a direct interface with the Bento Arm, which uses the same family of motors. This enables the control of the entire system from one central hub, reducing hardware complexity.

The redesign also addresses functional gaps in the previous design. While the previous motors frequently burned out after reaching temperatures as high as 85°C , the V2 is rated for at least one hour of continuous operation. This one-hour benchmark was selected to mirror standard laboratory continual-learning experiments; within this window, the Dynamixel motors reached a thermal equilibrium of 40°C , well below the maximum operational temperature of 60°C . Additionally, the maximum grip force increased to 6.34 N, extending the device’s ability to grasp a broader range of objects.

Previous work has shown that increasing the sensor space provided to a machine learning algorithm may improve its ability to make correct classifications in various situations [79]. Although the overall mass increased slightly to 275 g, this trade-off

was necessary to accommodate the more durable and sensor-rich actuators.

Finally, to ensure global accessibility, the HANDi Hand V2 continues to be designed entirely around straightforward fabrication techniques and open-source software libraries. Off-the-shelf components are specified in an open-access Bill of Materials, while all 3D-printed design files, including the parametric CAD models (SLDPRT, STEP) and pre-rendered mesh files (STL), are shared on a public repository.

Table 4.1: Design Specifications for the HANDi Hand V2

Item	Design Spec.	HANDi Hand V1	HANDi Hand V2
Size	Full scale, anatomical proportions		✓
Degrees of Freedom	Flexion/extension of each finger plus abduction/adduction of thumb		✓
Degrees of Actuation	Each finger independent; thumb adduction separate from flexion		✓
Ease of manufacturing	Little to no support material (PLA) required		✓
Interface	Compatible with Bento Arm		✓
Modularity	Exists as standalone system		✓
Mass (g)	< 500 g	256	275
Cost (CAD)	< \$2500	\$800	\$700
Open-Close Time (s)	< 1 s	0.43	0.47
Power Grip Force (N)	> 4N	4.20	6.34
Sensing	Joint angle, fingertip force, and visual data	✓	✓
Internal Motor Sensing	Position, velocity, current, voltage, temperature	X	✓
Durability	Runs for at least 1 hr without overheating	X	✓
Ease of assembly	Little to no custom circuitry	X	✓
Accessibility	Release of 3D CAD files	X	✓

4.3.2 Development Process

We modelled all components in the computer-aided design software SOLIDWORKS 2016 and fabricated them from polylactic acid (PLA) using a Bambu Lab X1C FDM printer. PLA remains the industry standard for 3D-printed prosthetics due to its accessibility and printability. To improve grasp stability on smooth surfaces such as PLA, we applied adhesive rubber pads to the fingertips and palm.

The most significant component change between V2 and V1 was the replacement of the servo motors, which necessitated the major redesign of the whole hand. The Dynamixel XL330-M288-T motors are used to actuate the flexion of each finger. These motors interface with either an Arduino-compatible OpenRB-150 microcontroller or a U2D2 PC-interface via the Dynamixel SDK. This configuration eliminates the need for the custom wiring and breadboards required by the legacy Hitec motors. All six actuators reside within the palm.

The HANDi Hand V2 continues to use an under-actuated flexion mechanism in each digit, allowing natural compliance around objects of arbitrary shape. A zip-tie tendon system improves on the original string-and-spring-based system by eliminating the need for tensioning and slack management, and greatly reducing the complexity of finger assembly. The durability of nylon zip-ties serves as an intentional solution to the durability barriers in machine learning research, and the design specifically enables the rapid (less than 5 min) replacement of damaged tendons. Rotation of each motor toward the wrist pulls the zip-tie in the palmar side of the corresponding finger, thereby causing flexion. Because the zip-tie is restricted from buckling by the hand’s internal geometry, motor rotation toward the finger pushes the zip-tie against the bottom of the fingertip, causing extension.

The thumb also underwent a major mechanical redesign to make use of the Dynamixel’s symmetric form factor. Previously, the thumb rotation motor was mounted within the wrist, driving movement via a 3D-printed gear. This presented multiple

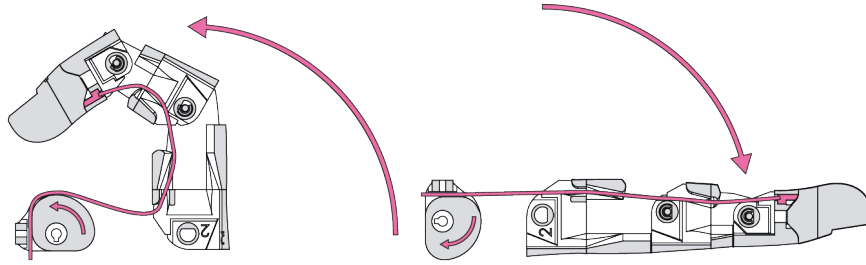


Figure 4.2: The flexion of the finger uses a single zip-tie. One end is attached to the inside of the fingertip, while the other end is secured using a setscrew to a 3D-printed servo spool.

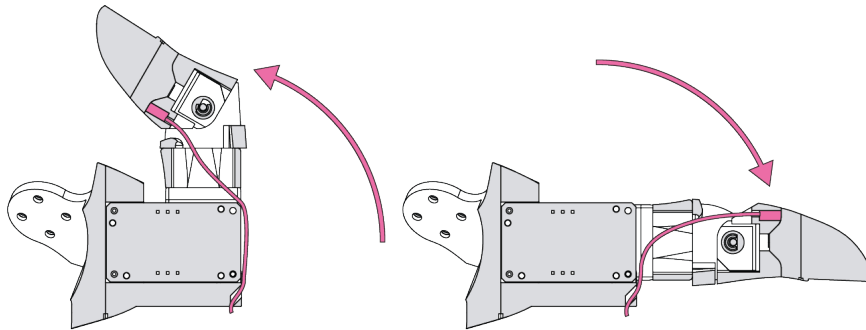


Figure 4.3: The intermediate phalanx of the thumb is directly mounted to the in-thumb flexion servo horn. Rotation of this servo directly actuates the intermediate phalanx and simultaneously causes the zip-tie to wrap around the servo housing for distal-phalanx flexion. Thumb rotation is accomplished by direct drive from a palm-mounted rotation servo.

issues, including weaker grips involving the thumb and gear tooth erosion. Taking advantage of the motor’s symmetrical form factor, a bracket mechanism now drives the thumb rotation and flexion, improving strength and durability. The flexion motor is integrated directly into the thumb’s silhouette, reducing printing material and complex tendon routing.

These key design features can be seen in Figures 4.2 and 4.3.

4.3.3 Sensorization

The primary advantage of the Dynamixel motors is the abundance of internal sensor feedback. Unlike the Hitec motors, which only received position commands, the Dynamixels broadcast real-time position, velocity, current, temperature, and voltage data. This sensory stream is crucial for “on-the-fly” learning required in ML control systems.

The V2 also supports a variety of optional sensors, including joint-angle potentiometers, fingertip force sensors, a wrist-mounted camera, and an inertial measurement unit (IMU). These can be seen in Fig. 4.4, where the IMU could replace the camera or be added on in the back of the wrist.

The HANDi Hand V1 included force sensitive resistors (FSRs) underneath a rigid 3D printed button to act as fingertip force sensors. Due to the mechanics, this version was only capable of sensing forces directly perpendicular to the button and would frequently jam when exposed to angled forces. To address repeatability issues and allow the fingertips to capture a wider variety of forces, the rigid buttons were replaced with flexible finger pads. These finger pads are 3D-printed from thermoplastic polyurethane (TPU), a widely available flexible filament, and slide over the FSRs in the fingertip assembly.

A wrist-mounted camera configuration was selected over the V1’s palm-mounted design to improve the field of view for machine learning tasks. Positioning the camera at the wrist allows it to capture both the approach trajectory of the HANDi Hand toward an object and the hand’s interaction with that object without full occlusion, providing improved visual context for grasp prediction. The camera module is designed to be easily installed in one of multiple mounting positions around the wrist, depending on the requirements of the application. An example of the camera’s field of view is shown in Fig. 4.5.

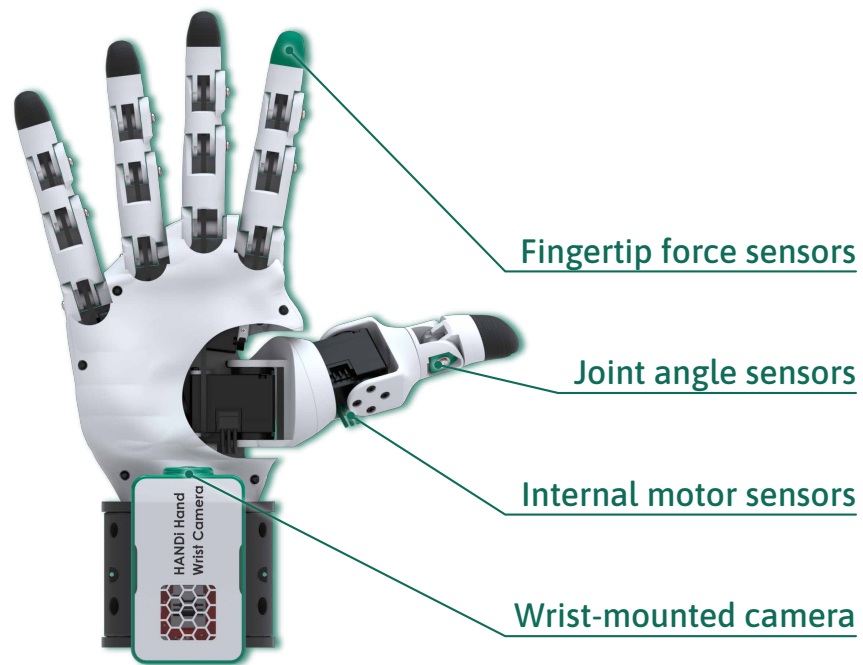


Figure 4.4: Sensors included in the HANDi Hand V2 redesign.

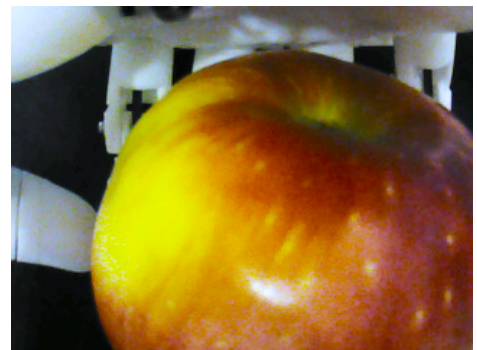
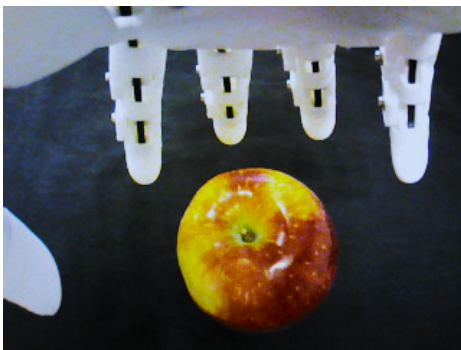


Figure 4.5: An example of the wrist-mounted camera's field of vision, with the target object distant and close to the hand during reaching.

4.4 Testing the HANDi Hand V2

4.4.1 Grasp Force Testing

To evaluate the mechanical performance of the HANDi Hand V2, we measured its grasp force across three of the most frequently used grasp types: cylindrical grip, tripod pinch, and pulp pinch [80]. We used an Omega LCM703-5 load cell (5 kgf capacity, 0.02 N accuracy, 0.003 N resolution) fitted with custom 75 mm wide 3D-printed shells. With cylindrical shells, the hand produced 6.3 N of grasp force — a 2 N increase over the original design. Tripod and pulp pinch forces reached 3.7 N and 2.5 N, respectively.

4.4.2 Functional Grasp Assessment

The Anthropomorphic Hand Assessment Protocol (AHAP) is a validated bench-top test for comparing the functional grasping abilities of 3D-printed and clinically prescribable/commercially available (CPCA) prosthetic hands [48, 49]. The protocol evaluates ten prevalent grasp types and postures across 26 common household objects. Each task consists of two stages: a grasping phase to achieve the required posture and a maintenance phase where the hand must hold the object for three seconds while oriented palm-down.

To conduct the trials, we mounted the HANDi Hand on a custom-built wand with a button control interface. The hand moved at a constant velocity and grip strength to eliminate human error. We assigned scores following the AHAP protocol: 1 for successful grasps or stable maintenance, 0.5 for partial grasp success (e.g., incorrect posture or object movement) and 0 for failure. We also assigned a score of 0 and skipped the task if an object threatened to damage the 3D-printed device (i.e. the skillet).

We compared the HANDi Hand V2’s scores against the original and several other 3D-printed models, including InMoov, Hackberry, and BEAR PAW. The scores for

these additional hands have been previously reported by Siegel et al. [51] and Llop-Harillo et al. [49]. While other 3D-printed hands often exhibit a performance disparity compared to CPCA hands—particularly during the maintenance phases—the V2 achieved a maintenance score of 81%, outperforming its 77% grasping score. This result is attributable to the Dynamixel motors’ active position control, which maintains grip force despite the flexibility of the hand’s zip-tie tendons. Finally, the V2 achieved an overall Grasp Assessment Score (GAS) of 79% (see Fig. 4.6 and 4.7), positioning it as a competitive alternative to other 3D-printed designs.

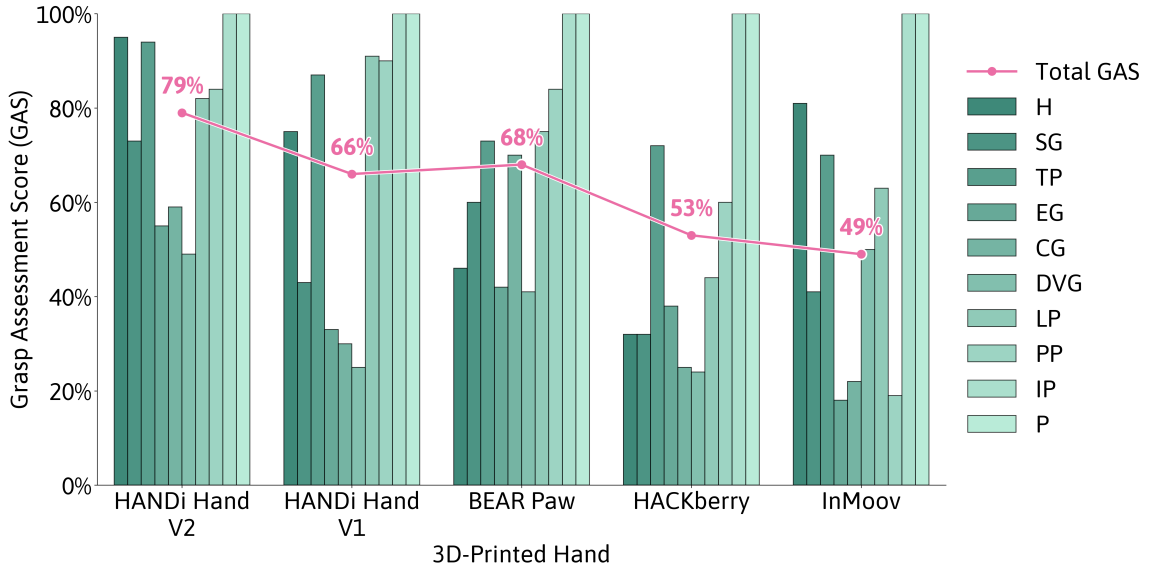


Figure 4.6: Comparison of total GAS of HANDi Hand V2 (HH) with other 3D-printed hands. H: hook, SG: spherical grip, TP: tripod pinch, EG: extension grip, CG: cylindrical grip, DVG: diagonal volar grip, LP: lateral pinch, PP: pulp pinch, IP: index pointing/pressing, P: platform.

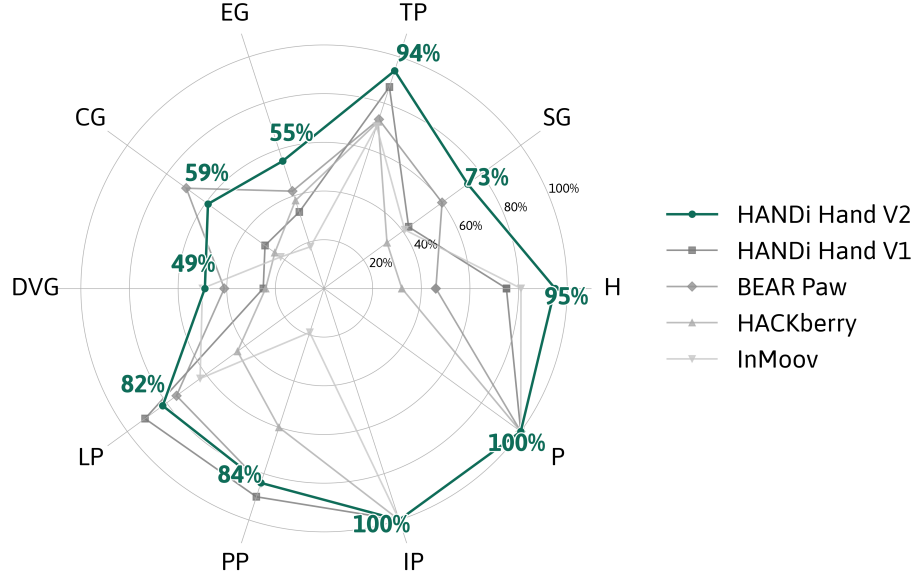


Figure 4.7: Comparison of partial GAS of redesigned HANDi Hand with other 3D-printed hands. Most 3D-printed hands struggled with certain grasps that had strict posture requirements.

4.5 Application of Machine Learning

4.5.1 Adaptive Switching

We now explore the HANDi Hand V2’s suitability as a platform for continual machine learning research. Specifically, we examine how it supports the novel extension of an established continual machine learning approach for macro-level joint control to the more nuanced setting of prosthetic grasp selection. As bionic hands become more complex, the burden on the user to manually toggle between grasp types increases cognitive load.

Here, we deploy a continual machine learning method known as *adaptive switching* [55, 69] to help address this concern by leveraging learned General Value Functions (GVFs, [65]) to automate grasp selection (as suggested by Lau et al. [76]). Unlike static prediction models that require offline training, GVFs use temporal-difference learning to learn and predict concepts such as user intent in real time. This creates a co-adaptive framework where the system learns the user’s behaviour while the user

simultaneously adjusts to the system’s predictions [58, 71].

Pilarski et al. first introduced adaptive switching to upper-limb prosthetics, suggesting that predicting the next joint a user will control could lower the time and cognitive load of joint switching [68]. Edwards et al. extended this by applying adaptive switching to the real-time operation of a robotic arm by a prosthesis user. Their work demonstrated that after an initial learning phase, the system typically requires only a single switch signal to reach the desired function. Adaptive switching may encourage users to use the full functionality of their device rather than restricting themselves to a small subset of grasps [55]. Furthermore, since the system learns continually during use, it could remove the need for frequent recalibration by clinical staff, making it suitable for long-term daily operation [18, 70].

To date, adaptive switching has only been tested by prosthesis users in the context of macro-level joint selection on a desk-mounted arm prosthesis, the Bento Arm [75], with joints including the shoulder, elbow, wrist, and a hand involving a single open/-close movement—not multi-dexterous actuation. Adaptive switching has also been implemented in predicting walking modes in lower-limb exoskeletons [81]. Our preliminary findings show that it should be possible to learn GVF predictions that relate to prosthetic grasping movements [76]. The present experiment utilizes the HANDi Hand V2 to address this research gap by determining 1) whether adaptive switching can be used in grasp selection, and 2) the V2’s suitability to handle multi-modal sensory data to a continual learning architecture.

4.5.2 Experimental Setup

For a detailed algorithmic description of adaptive switching, please see Edwards [69]; for an extended description of GVFs, please see Sutton et al. [65]. In the interest of space, here we only present the details and parameters needed to contextualize and replicate the experiments conducted. We used an Adafruit BNO085 inertial measurement unit (IMU) to establish directional awareness for the learning machine

in the experimental space. For this task, we selected yaw and pitch as the state components and discretized them via tile coding [63] to minimize the computational complexity required for real-time continual learning. Both state components were coded together in a joint 2D space, with each axis having 16 bins. This resulted in a 256 element feature vector with a single active feature.

The IMU signals were acquired from the Adafruit BNO085 via an Arduino Mega 2560 and sent along via a serial interface to our central brachI/Oplexus software [82], which also handled the control of the Dynamixel motors. The left and right buttons on the wand were mapped to open and close velocity movements on the hand, and the central button was mapped as the sequential switching signal to allow for cycling through grasp patterns. The state space and cumulant signals were sent from brachI/Oplexus to an external Python script running the learning code for the GVFs via a User Datagram Protocol (UDP) communication interface. After logging signals and forming predictions, the predictions were sent back to brachI/Oplexus.

The experiment compared sequential and adaptive grasp switching. For the sequential trials, the user was given a fixed list of six grasps in the following order: lateral pinch, pulp pinch, cylindrical grip, tripod pinch, extension grip, and spherical grip. In the adaptive trials, we assigned one prediction (represented via a learned GVF, [55, 76]) to each grasp—i.e., the system learned to predict the expected activity of the different grasping patterns with respect to different orientations. The switching list was then dynamically reordered based on the relative magnitude of the different GVF prediction strengths. When the system received a switch signal, it automatically presented the most probable grasp as the first option. We selected three grasps—extension grip, tripod pinch, and pulp pinch—for a cyclical task. The protocol required the user to perform an extension grip to the left, rotate to the right for a tripod pinch, and tilt upward for a pulp pinch. We selected these grasps to be evenly distributed within the static switching list. An able-bodied participant performed this cycle as many times as possible in a 3-minute trial. For each GVF,

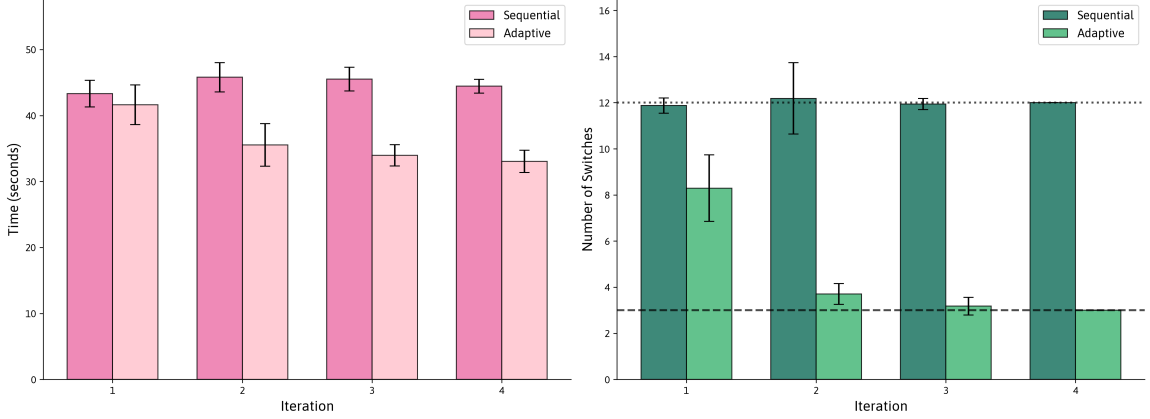


Figure 4.8: Comparison of sequential and adaptive switching averaged over 15 trials. (Left) Total time and (Right) number of switches shown by iteration of the task. The dotted line shows the minimum number of switches possible using a static switching list. The dashed line shows the best possible switching performance with no user errors and a single switch at each switching event.

we set the cumulant (C) to 1 when the participant actively used a grasp (holding the open or close button). We empirically selected a discount factor (γ) of 0.9 and a trace decay parameter (λ) of 0.99. Finally, we conducted 15 trials for each switching condition.

4.5.3 Results and Discussion

The primary objective of this experiment was to provide a demonstration of the HANDi Hand V2’s suitability for use as a research platform in contexts which require rich sensory data streams. Figures 4.8 and 4.9 illustrate the difference between sequential and adaptive switching control methods across 15 trials.

Fig. 4.8 (left) compares the average time required for switching events. We define a switching event as the duration between the initial switch trigger and the initiation of hand movement. While sequential switching maintained a constant total switching time of approximately 45 seconds, the adaptive condition demonstrated a distinct downward trend. Although it started near the sequential baseline, adaptive switching performance significantly improved with successive iterations, settling at about

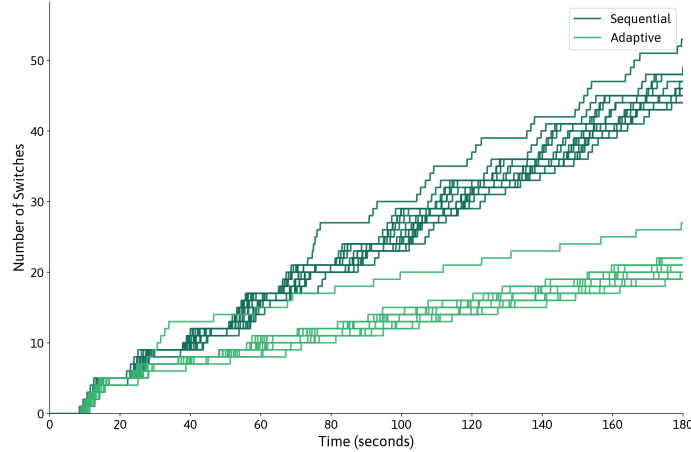


Figure 4.9: Cumulative number of switches over the course of a 3-minute trial.

33 seconds. This reduction in time suggests that the user and system successfully achieved co-adaptation, optimizing the selection process as the predictive model familiarized with the task. As shown in Fig. 4.8 (right), the number of switches per iteration also decreased during adaptive trials. While the sequential method required a minimum number of switches to cycle through the grasp library, the adaptive system learned to present the correct grasp with only a single trigger by the third iteration. The narrowing of error bars in Figure 4.8 indicates that the system and user converged on a stable, efficient task strategy by the fourth iteration. The cumulative switch count in Fig. 4.9 further validates the prediction accuracy of adaptive switching. For the initial 40 seconds, both conditions follow a similar trajectory during the model’s early learning phase. However, after this period, the adaptive switching curve flattened, reaching a near-optimal performance threshold. In this state, the user effectively relied on the system’s predictions. This confirms that adaptive switching is a promising method for managing the complexity of a multi-articulated prosthetic hand like the HANDi Hand V2. A persistent challenge in upper-limb prosthetics is the disparity between the high degrees of freedom in modern hardware and the limited control signals provided by the user. Our results extend prior findings on adaptive switching to demonstrate that adaptive switching can help bridge this gap

by predicting intended grasps based on historical usage patterns and environmental context. This supports a shift in the burden of switching control from the user to the machine learning system. User feedback revealed a significant qualitative difference between the two control modes. During sequential switching, the user reported a need to physically move the hand away from the target object while cycling through the grasp list to avoid accidental collisions. Conversely, adaptive switching eliminated this constraint. By presenting the intended grasp immediately, the system allowed the user to approach the object without hesitation. This suggests that the benefits of predictive control extend beyond quantitative metrics, but could also foster user agency and trust. Current clinical standards often limit CPCA hands to two or three pre-programmed grasps due to the cognitive demand of sequential switching. Since adaptive switching hides the complexity of a large grasp list by presenting only relevant options, it may prove a feasible way to offer users a much wider array of functional patterns without increasing switching latency. This scalability is essential for the next generation of prosthetic devices, closing the gap between device functionality and usability. Our results confirm that the development of HANDi Hand V2 is not only a functional upgrade but also an asset for future machine learning experiments. Traditional 3D-printed hands often struggle to perform under the many cycles required for trial-and-error learning. The mechanical durability of the HANDi Hand V2 allowed us to observe the convergence of adaptive switching over repeated trials—data that would be difficult to capture on less robust devices. For future work to enable adaptive switching for real-world grasp selection, we will focus on developing effective feedback systems to communicate the system’s predictive confidence levels to the user. While this preliminary study intentionally restricted the GVF state space to yaw and pitch, this was critical to validate algorithmic convergence. Future work will leverage this validated output to scale the GVF state space to include additional environmental context, such as camera vision and tactile feedback from the fingertips. This will refine the prediction model for regular use in stochastic,

daily-life environments. By further exploring how adaptive switching can predict a user’s need in real time, we anticipate finding new ways for the field to move closer to a seamless co-adaptation where the human and machine learn to interact with the world in unison [58, 71].

4.6 Conclusion

The development of the HANDi Hand V2 represents a major step in filling the gap between multi-articulating prosthetic hands and machine-learning research. By re-designing the original hand with a focus on accessibility, sensorization, and new actuators, we have created a robust platform for implementing and testing continual learning algorithms. As a first contribution, functional validation via the AHAP demonstrates that the V2 not only matches but exceeds the performance of similar 3D-printed robotic hands, achieving a GAS of 79%. As a second core contribution of this work, we successfully demonstrated adaptive switching for grasp selection in a multi-articulating hand—the first such demonstration in a setting with such broad prosthetic application. Our results indicate that, while conventional sequential control forces users to cycle through an entire available grasp library, adaptive switching allows the user to only have to switch once between tasks. Taken together, our findings suggest that the HANDi Hand V2 is a viable platform to further investigate how prosthetic devices might move from passive tools to truly intelligent, tightly coupled extensions of the human body.

Chapter 5

Open-source Release of the HANDi Hand V2

Chapter 5 presents a comprehensive perspective on the open-source release process of the HANDi Hand V2. It serves three key purposes: to document the design process behind the assembly manual, to explain the choices made during its evolution, and to serve as a guide for future designers who will be adding to the open-source hardware.

5.1 Introduction

The development of robotic hardware is often limited not by the physical design, but by the friction involved in dissemination. The term “open-source” has become increasingly prevalent in the realm of software, enabling collaborative solutions to global problems. The benefits of open-source software (OSS) are plentiful, including the ability for users to adapt the software to their needs and share it with others who may have similar objectives. While OSS has grown exponentially in the past few decades, open-source hardware is in a relatively nascent stage, especially in the field of robotics. For a complex device such as the HANDi Hand V2, providing CAD printing files is rarely sufficient. Without clear and comprehensive assembly instructions, open-source hardware is prone to remain trapped in a laboratory environment, only reproducible by its creators. This chapter details the development and user-centric

design of the HANDi Hand Assembly Manual.

5.1.1 The History of Open-Source Hardware

Open-source hardware (OSH) traces back to the early 2000’s with Bruce Perens, who was the creator of the *Open Source Definition* and co-founder of the *Open Source Initiative*. By the mid 2000’s, the emergence of several major projects and companies such as Arduino, Adafruit, and Sparkfun spurred the rapid rise of OSH worldwide. Furthermore, desktop additive manufacturing (spearheaded by RepRap) made fabrication even more accessible. These innovations proved that physical technology could be decentralized, enabling communities around the world to collaboratively build and improve systems [83].

In modern robotics and especially in upper-limb prosthetics research, OSH is urgently needed to address the field’s prohibitively expensive research platforms. These steep financial barriers for research institutions, students, and independent developers lead to a constant “reinvention of the wheel” conundrum, in which a brand-new prosthetic hand is often developed rather than modifying an existing design. The introduction of well-documented OSH in upper-limb prosthetics research has the potential to greatly accelerate the pace of innovation across the field.

5.1.2 Motivation and Objectives

Previously, the HANDi Hand V1 was released with an assembly manual and limited CAD files. Several challenges were identified with the initial release. First, only STL 3D print files were included, making it difficult to modify the 3D CAD models. Furthermore, several assembly and manufacturing steps were difficult to reproduce without the assistance of the original designer, especially in the sections describing cable wiring and sensor installation. Thus, the primary objective of the assembly manual for the HANDi Hand V2 is to eliminate these barriers by providing a highly visual and pedagogical guide. Specifically, the manual should achieve the following:

1. The end-user can assemble a fully functional HANDi Hand across different manufacturing environments without requiring intervention from the design team.
2. Common challenges such as improper tendon tensioning, incorrect motor programming, and 3D-printing confusion should be eliminated through intuitive design and graphical cues.
3. The original CAD models should be intuitive to modify, catering to different use cases and printer tolerances.

A critical objective of the open-source documentation is to cater to a diverse user base. The target audience of the HANDi Hand spans prosthetics researchers, engineering student groups, independent hobbyists, and machine-learning scientists. For instance, a computer vision researcher may require a desktop-mounted hand with a camera, while a prosthetics developer may require the hand to be mounted to a wearable socket. While users are assumed to have basic technical literacy, the manual should prioritize modularity.

5.1.3 Assembly Manual Outline

The HANDi Hand assembly manual is structured into several chapters that guide the user through the necessary steps to build a complete, working HANDi Hand.

1. **Introduction:** A brief introduction to the assembly manual and how to use it.
2. **Naming & Definitions:** Establishes the universal naming convention used in the manual as well as a colour-coding scheme.
3. **Required Materials:** Outlines the 3D-printed parts, ordered parts, and tools required. Includes the 3D-printing guide.
4. **Material Preparation:** Describes how the 3D-printed parts need to be post-processed and suggested steps for preparing the zip-ties.

5. **Dynamixel Servo Preparation:** Includes detailed instructions for how to program the servo motors.
6. **Before You Assemble:** Illustrates the repeated symbols, colours, and guidance cues used throughout the following three chapters to streamline the assembly process.
7. **Thumb Assembly:** Step-by-step visual instructions on how to assemble the thumb of the HANDi Hand.
8. **Finger Assembly:** Step-by-step visual instructions on how to assemble the fingers of the HANDi Hand.
9. **Palm Assembly:** Step-by-step visual instructions on how to assemble the palm of the HANDi Hand.
10. **After You Assemble:** Outlines the wiring steps required to use the HANDi Hand as well as options for boilerplate software. Includes optional add-ons such as a desktop stand and a joystick module.
11. **3D Printed Template:** A printable guide to help identify 3D-printed parts as well as a 1:1 template to serve as a “placemat” to sort parts for the three main assembly chapters listed above.

5.2 Methods

The methodologies used to create the HANDi Hand assembly manual encompass two domains: the exploration of user experience (UX) and the specific design process. Designing effective documentation requires a careful balance of technical precision and universal clarity.

5.2.1 Considerations for User Experience

Inspiration was drawn from several leading consumer companies. In particular, key design techniques were borrowed from the HANDi Hand V1’s assembly manual, IKEA [84], LEGO [85], and Gundam Gunpla model kits [86]. An analysis of these assembly manuals led to several questions that helped guide my learning and design trajectory. These questions led to their respective insights:

Why do successful manuals rely almost exclusively on visuals?

- *Insight:* IKEA (see Fig. 5.1) in particular minimizes text due to their global target audience. It bypasses language translation barriers. LEGO minimizes text for their younger target audience. A visual-first approach is essential for a target audience with diverse backgrounds and experiences.

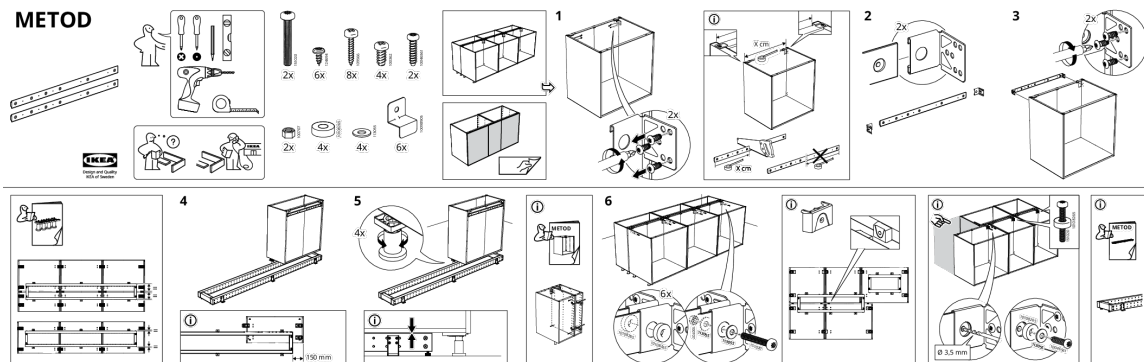


Figure 5.1: An example of the IKEA manual style. There is little to no use of text [84].

How do varying layout densities impact the reader’s experience?

- *Insight:* Gunpla manuals (see Fig. 5.2) pack many micro-steps into single pages to save physical paper since their manuals are packed in the same box as the product. Modern IKEA and LEGO manuals have been less dense as they have also become available in a digital format. Giving steps “room to breathe” could significantly reduce visual fatigue, leading to less user error.

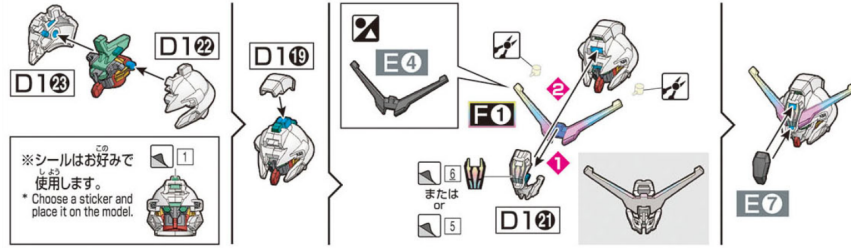


Figure 5.2: An example of the Gundam manual style. Many “micro-steps” are shown in a single cell [86].

Why show only the “action” steps versus showing both “action” and “finished” states?

- *Insight:* Similar to the previous insight, most commercial manuals show only an exploded action view (e.g. an arrow showing where a screw goes) to save printing space. However, this forces the builder to guess if the final state matches the intention of the exploded step.

How should parts be named to maximize clarity?

- *Insight:* IKEA and LEGO use serial numbers which work well for warehouse tracking and ordering replacement parts. However, these do not convey any intuitive meaning to a human builder. Meanwhile, Gunpla manuals use nomenclature based on the plastic runner grids the parts are attached to (e.g. runner A has parts A1 to A12).

After collecting these insights, the ideas were synthesized into concrete decisions for the HANDi Hand assembly manual.

Clear Visuals

- A major design decision was whether to show only the action step or to include the finished state as well. I decided to present both states for all steps. While this increased the length of the document, it would remove ambiguity and reduce assembly errors.

- Inspired by Gunpla manuals, components assembled in previous steps should be rendered in gray while newly introduced parts are rendered in black. This immediately draws the builder’s eye to the parts needed in the current step.
- A variety of arrows should be used to show insertion paths and interactive motion actions (e.g. turning a part to lock in place). Additionally, warning icons and colour cues should be used to highlight crucial steps that are prone to error. An example would be confusing two different screw lengths which can be seen in Figure 5.3.

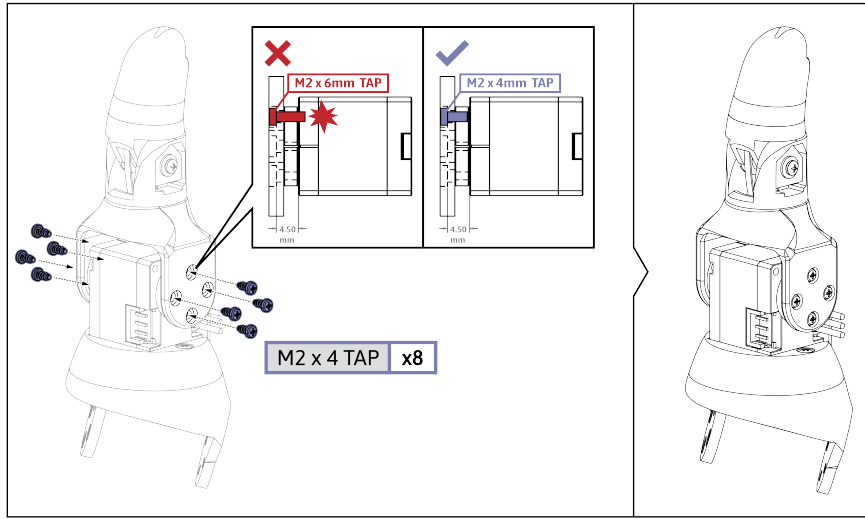


Figure 5.3: Example of a common mistake during the assembly process mitigated by warnings and colour cues.

An Intuitive Naming Convention

- For the HANDi Hand V1, purely anatomical terms were used to identify parts (e.g. IP-P Pivot R for Intermediate phalanx - proximal pivot right hand). This was confusing for builders without a biomechanics background. For a diverse target audience, a human-readable naming convention was needed.
- A structured taxonomy was created that bridged anatomical accuracy with an

intuitive hierarchy:

$$\langle \text{Category} \rangle - \langle \text{Part} \rangle - \langle \text{Subpart} \rangle - \langle \text{Digit} \rangle$$

For example, `finger_proximal_a_d2` informs the builder that the part belongs to a finger, is a proximal (closer to the center of the body) link, is variation A, and belongs to digit 2, the index finger.

- Moreover, the physical fingers should be mapped from D0 and D1 (thumb rotation and flexion, respectively) to D5 (little finger) while matching the motor and software control IDs (ID10-ID15). This can be seen in Figure 5.4

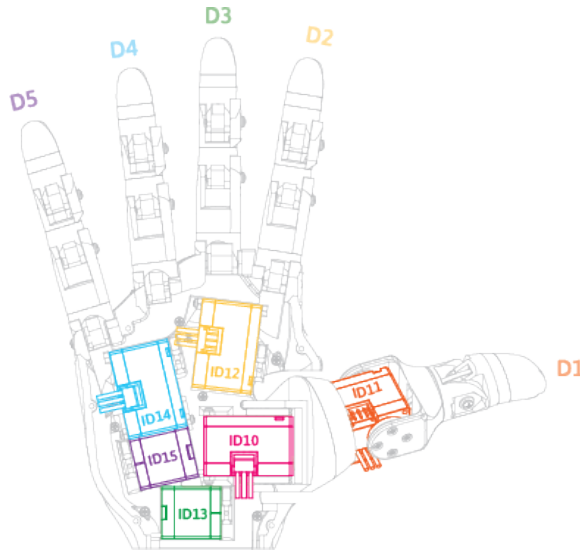


Figure 5.4: Naming convention for the fingers (D1 to D5) and motors (ID10 to ID15).

A Printable 1:1 Scale Template

- To prevent errors such as accidentally replacing a 6 mm screw with an 8 mm screw that would cause damage to components, I decided to include a 1:1 scale printable template. An example of one of the pages can be seen in Figure 5.5. Users can physically lay screws and 3D-printed parts directly on the page to verify their size before assembly.

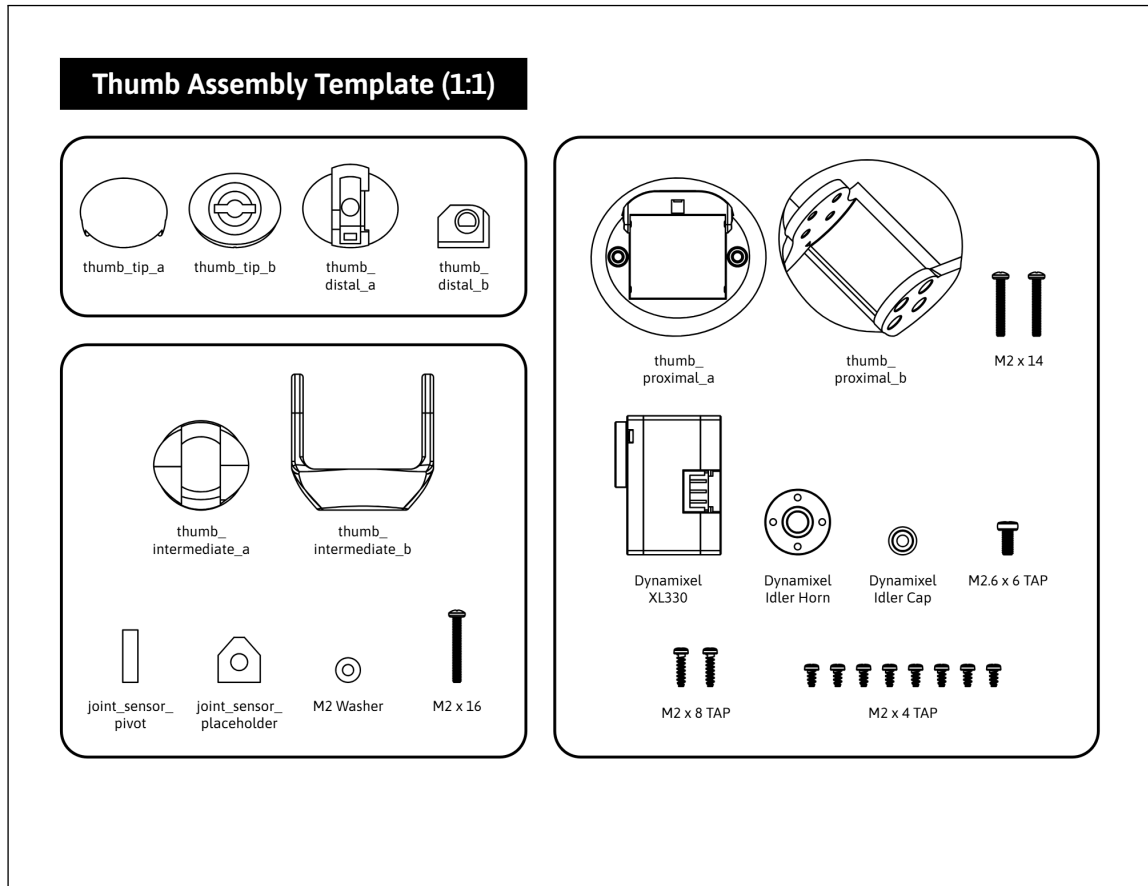


Figure 5.5: An example of the printable 1:1 scale template. Users can place both 3D-printed parts and off-the-shelf parts on the paper to identify which ones are needed for the assembly step.

Unlike companies like IKEA, LEGO, and Gundam that ship pre-made parts in quality-controlled kits, open-source hardware like the HANDi Hand must also address various uncontrolled manufacturing environments. This is a unique problem for the HANDi Hand, and the assembly manual was specifically designed to handle these variables.

Tolerancing in 3D-Printing

3D-printers have a variety of settings that can influence the quality and underlying structure of the print. These settings can vary across printers but most commercial printers share the same core settings. When a user chooses to print a part, they would download the 3D model file (e.g. an .STL file, representing a 3D object's shape) from the open-source repository. After uploading the model to the user's 3D-printer software, a process called *slicing* cuts the 3D model into hundreds of horizontal 2D layers and translates this information into instructions for the machine to print in 3D.

- I chose three slicing profiles or print presets: HANDi Hand (HH) Default, HH Supports, and HH Strong that defined key printing settings including layer height, wall counts, infill pattern, infill percentage, use of supports, and brim width. The specific values can be seen in Figure 5.6. These three slicing profiles represent the default print settings for most parts, for parts that require supports, and for structural parts that undergo higher stresses respectively. The printing settings were chosen as they were common across multiple printer models.
- Since users would have to orient the parts on their printer's print bed, the manual provides visual print-bed layout guides, such as one shown in Figure 5.7, to minimize print failures and necessary support material.
- Clear instructions such as the one in Figure 5.8 are provided on where and how

Print Preset	Layer Height (mm)	Wall Loops	Sparse Infill Pattern	Sparse Infill Density (%)	Supports	Brim
HH Default	0.2	2	Honeycomb	15	None	Outer 5mm
HH Supports	0.2	2	Honeycomb	15	Manual	Outer 5mm
HH Strong	0.2	2	Rectilinear	50	None	Outer 5mm

Figure 5.6: The three printing profiles used in the HANDi Hand.

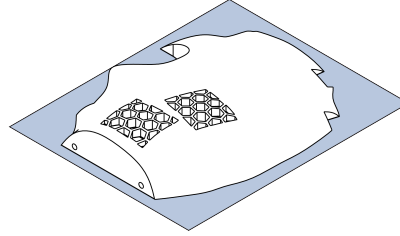


Figure 5.7: Example of part orientation on the 3D-printer build plate.

to sand joint clearances to account for printer-specific extrusion. This ensures smooth mechanical movement across all builds.

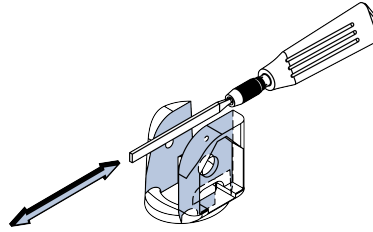


Figure 5.8: Example of support removal instructions.

Mitigating Different Build Environments

- During testing in dry laboratory environments, the nylon zip-ties used as tendons frequently experienced brittle failure. Nylon is hygroscopic, where it readily absorbs moisture from the air to remain ductile.
- To make the performance of the HANDi Hand repeatable across various build environments, a 24-hour soaking protocol was introduced in the manual. Soaking the zip-ties in water increases their long-term flexibility, regardless of local

humidity.

Software Accessibility

- Since users would come from diverse fields ranging from computer scientists, medicine, industrial design, to university student groups, the manual should provide different software starting points:
 1. **Arduino IDE:** Uses pre-configured Arduino sketches and joystick control for simple actuation.
 2. **Python API:** Uses a Python script for low-level motor parameter access for fine control of the HANDi Hand sensor space [87].
- Many errors could occur during software setup, such as configuring the motor position limits before assembly. Explicit safety steps were included to prevent irreversible damage to both the motors and 3D-printed parts.

5.2.2 Steps to Design the Assembly Manual

The design of the assembly manual involved five phases. Rather than writing text and generating figures simultaneously, I followed an iterative workflow that allowed me to collect feedback and refine the manual at every phase.

Phase 1: The First Draft

This phase involved drafting a collaborative document containing the steps required for the three main sections of the hand: the thumb, the fingers, and the palm. Each section included its own Bill of Materials (BOM), listing the necessary number of 3D-printed parts and fasteners. Steps within the section were created based on whether a new part was introduced from the BOM. Starting with a text-only document helped identify missing steps and areas where ambiguous spatial directions such as “attach to the right side” could be clarified with a diagram.

Phase 2: CAD Modelling

Once the written steps were validated, the complete CAD assembly of the HANDi Hand in SolidWorks was used to capture the physical appearance of each step. I created configurations and exploded views for every step. This ensured that the CAD model's state matched the process described in the written text. Screenshots of these steps were added to the first draft to determine whether additional perspective views and close-up pop-outs were needed.

Phase 3: Generating Vector Graphics

To maintain a crisp resolution for all figures, the raw CAD screenshots were replaced with scalable vector line art. Each SolidWorks assembly step configuration was rendered into a SolidWorks Drawing file (.slddraw) with display mode set to "Hidden Lines Removed". The display perspective for all steps was set to isometric to maintain consistency. Finally, the drawings were exported as an Adobe Illustrator (.ai) file for post-processing.

Phase 4: Annotation in Adobe Illustrator

The raw vector exports from SolidWorks contain monochrome linework that is difficult to understand, especially for steps that involved action. Adobe Illustrator was used to add annotations such as arrows, shading, and colour to create functional illustrations. Key features and tools used during this phase included the live paint group tool, which allowed drawings to be "coloured in" to emphasize areas such as gluing surfaces. An example of these annotations is shown in Figure 5.9.

Phase 5: Page Layout in Adobe InDesign

The final stage brought all complete vector graphics, written text, and warnings into Adobe InDesign to create the final layout. InDesign was selected over traditional word processors due to its precise grid alignment system and customizability. A

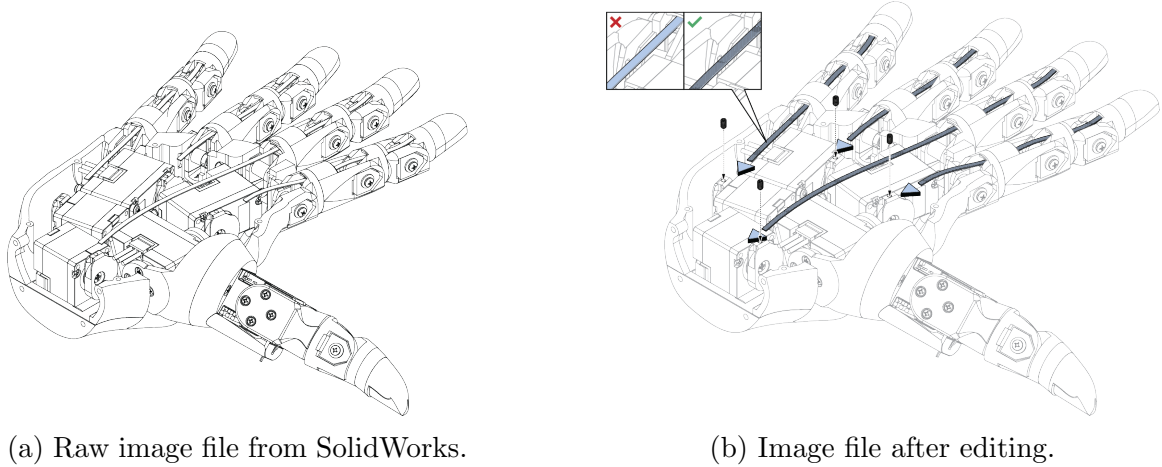


Figure 5.9: Comparison of an assembly step before and after annotation in Adobe Illustrator.

comic-book style was adopted from Gundam to establish a natural reading order of left-to-right, top-to-bottom. Each action step was paired with a finished state panel next to it to provide visual confirmation. The document was formatted to be either printed in a physical booklet or viewed as a digital PDF. An example of a complete page in the assembly manual can be seen in Figure 5.10.

5.3 User Validation

To evaluate the manual’s real-world usability, a pilot study was conducted with a cohort of five builders. These individuals represented varying levels of technical experience. During the build session, the cohort was encouraged to “think-aloud” as they navigated the assembly manual. Builders were also prompted to assemble the HANDi Hand using only the draft documentation. By withholding the original designer’s intervention, friction points were noted in real time to evaluate whether the manual could be self-sustaining. Pilot testing revealed several challenges that steered the direction of the final assembly manual. There were frequent mix-ups between the visually similar M2 screw lengths, which led to the subsequent reliance on the 1:1 scale printable templates. Mistakes in motor programming led to a more frequent

For assistance in identifying each 3D-printed part and mechanical part, refer to the 3D-Printed Parts List and Thumb Assembly Template (1:1) print-out in Appendix A.

1 Distal Thumb

thumb_tip_a

thumb_tip_b

thumb_distal_a

thumb_distal_b

use of warning colours and symbols to draw the user’s attention to critical steps. Furthermore, observations of builders’ confusion in the zip-tie installation stage led to the creation of additional 3D perspective figures to provide clarity.

For the official definitive HANDi Hand Assembly Manual, please go to the public repository (<https://github.com/BLINCdev/HANDi-Hand>). The version included in this thesis was visually scaled down to comply with formatting requirements.

5.4 Future Work & Conclusions

While the current assembly manual establishes a foundation for open-source fabrication, future updates should focus on exploring web-based documentation and expanding on the hand’s modular features. To complement the static PDF and print formats, web-based 3D model viewers could provide additional aid. These would allow users to dynamically rotate, explode, and zoom into CAD assemblies to further reduce spatial ambiguity in complex steps such as zip-tie tendon routing. Next, to support even more specialized applications, additional appendices should be made for modular add-ons. These would include dedicated instructions for integrating fingertip sensors and the wrist-mounted camera.

The HANDi Hand Assembly Manual bridges the gap between complex robotics and accessible, repeatable manufacturing. By treating documentation as important as the physical design, this work ensures that the HANDi Hand V2 is attainable to as many users as possible. Beyond the manual, this work also highlights the critical role of scientific communication, demonstrating how thoughtful documentation can make prosthetic technology accessible across a wide range of users.

Chapter 6

Discussion, Future Directions, and Conclusions

6.1 Significance and Discussion of Contributions

Research in continual machine learning methods in upper-limb prosthetics control is uncommon. In part, this is because researchers choose to use either exorbitantly expensive prosthetic research hands to test their algorithms on or design their own hardware, costing time and resources. By developing a durable, sensorized hand with use in continual machine learning research in mind, we have addressed some of the barriers present, both physically and by demonstration.

The first major contribution in this thesis is the HANDi Hand V2. The design houses Dynamixel XL330 actuators that stream live position, velocity, load, current draw, and temperature. In addition to internal motor sensors, the hand accommodates modular additions such as a wrist-mounted camera, individual joint angle sensors, and fingertip force sensors. Direct-drive thumb brackets and durable zip-tie tendons result in the hand delivering a 6.34 N power grip. The functional grasp performance of the HANDi Hand was evaluated under the Anthropomorphic Hand Assessment Protocol (AHAP), where it earned an overall Grasp Assessment Score of 79%. This result places it surprisingly close to commercial-grade prostheses, highlighting the potential of open-source 3D-printed hands.

The development of the HANDi Hand made it possible to demonstrate the use of continual prediction learning on individual finger movements. Using Temporal-Difference learning ($TD(\lambda)$) to acquire and maintain learned values within a Horde of General Value Functions (GVFs), the system predicted upcoming finger movements in real time based solely on motor position and velocity data. These predictions accumulated well before physical movements occurred, proving that reinforcement learning techniques can forecast multi-digit movements without offline calibration.

These predictions were then put into a live control loop through adaptive grasp switching. Using the spatial orientation data from a wrist-mounted IMU to inform live GVF predictions, the hand dynamically reordered its available grasp menu on the fly. In user trials, adaptive switching was compared to sequential switching, where the grasp menu remained in a static sequence. Adaptive switching trials reduced the duration of the task from a baseline of 45 seconds to 33 seconds. By the third round of grasp actions, the algorithm reliably presented the correct grasp on the first switch press, shifting the operational burden from the user to the machine while keeping the user in control. This is the first time adaptive switching has been demonstrated for grasp selection.

The final contribution is a practical assembly manual for the open-source release of the HANDi Hand V2. To ensure that other researchers could build the hand in a wide range of environments and expertise, the assembly guide borrows techniques from world-class commercial examples. The manual used a variety of techniques to make assembly simple, including a visual-first approach, standardized 3D-printing profiles, and 1:1 scale part templates. Independent user testing confirmed that external builders could assemble the hand without the designer’s intervention, establishing the foundation for future open-source hardware dissemination.

6.2 Limitations of Current Work

A few limitations exist in the current design of the HANDi Hand V2. Although the grip force of the hand is well-suited for light to medium ADLs, it would struggle to support heavy-duty tasks or larger objects. Furthermore, the hand was only tested with a handheld demo wand; it was never adapted to be used with a prosthetic socket. The current system lacks an onboard battery management system and a quick-disconnect wrist interface, making body-mounted deployment untested. Although the hand was designed with sensor modularity in mind, the functionality of additional sensors that measure joint angle and fingertip force remains untested. As a result, the base model of the hand can only infer finger position solely from the motor position feedback. This limits the hand’s immediate use for researchers specifically looking to explore joint-level state feedback for machine learning research.

To guarantee clean algorithmic convergence during early live trials, the adaptive switching state space was restricted to two IMU data streams (pitch and yaw). This meant that other sensory information from the camera, fingertip force sensors, and internal motor sensors was excluded due to the simplicity of the task. The experiments were also limited to a single able-bodied participant due to time limitations. Additionally, the reinforcement learning hyperparameters (e.g. learning rates, discount factors, and trace decay parameters) were kept static. Fixed settings risk under- or over-adapting if a user shifts their habits relative to the set update rate. While three commonly used grasp types were included in the experiment, we did not explore the impact of introducing new grasp types during the experiment to determine how the system could dynamically adapt over a longer trial. Furthermore, the feedback system for the adaptive switching experiment only consisted of auditory output. Other techniques such as haptic or visual feedback were not explored. Finally, the long-term stability and user acceptance of adaptive switching remain unmeasured. Since trials were restricted to brief sessions in the laboratory, it is unclear whether dynamic grasp

reordering would remain useful or desirable over time.

Finally, for the open-source release of the HANDi Hand, parametric files were created exclusively in SolidWorks. While the .STEP files were released, which allow for basic viewing and feature recognition, users without proprietary SolidWorks licenses would not be able to modify specific features in the original design file.

6.3 Future Work

Several design features could improve the HANDi Hand while keeping accessibility at its core. At the mechanical level, future iterations could refine the design of the finger assembly. Since the HANDi Hand V2 prioritized the redesign of the palm to house the larger Dynamixel motors, most of the finger geometry was adapted directly from the previous version. This design prioritized the ability to be printed without supports due to the limitations of the commercial 3D printers available in 2017. A dedicated redesign of the finger could optimize the strength and reduce the number of printed parts. Additionally, testing other finger tendon alternatives, such as ones that are 3D-printed in flexible material, could eliminate the need to mitigate zip-tie breakage.

The next logical step would be including more sensory data streams inside the Horde architecture. Using colour and shape detection from the camera, EMG muscle signals, and fingertip force sensors would let the agent predict intended grasps in more complex task environments. Different feedback systems should be explored to reduce hesitation and build mutual trust during use [71].

Moving past bench-top validation would require clinical trials with persons with limb difference using tasks from activities of daily living (ADLs). Evaluating the practicality of deploying adaptive grasp switching in a multi-articulating prosthetic hand would require clinical benchmarks like the Southampton Hand Assessment Procedure (SHAP) [45] and the PROS-TLX cognitive workload survey [88].

Finally, expanding the open-source release with web-based interactive 3D models would lower barriers for more users to build the HANDi Hand V2. Providing native CAD files across multiple formats (e.g. Onshape, Fusion 360) alongside SolidWorks would ensure that users without proprietary software could easily adapt the HANDi Hand to their specific needs.

6.4 Conclusion

Multi-articulating robotic hands are often showcased as the leading edge of modern technology research. However, translating these technologies into the field of prosthetics fails when operating them requires too much conscious effort for completing basic tasks. By pairing a highly sensorized open-source hand with real-time predictive learning, this work offers insight into a future past that barrier. Prostheses do not have to be rigid, passive tools that users have to learn how to use; they can operate as adaptive partners that learn alongside the user, much like how we cooperate with our friends.

Bibliography

- [1] I. A. Kapandji, *The Physiology of the Joints: Upper Limb*, 5th. New York, NY: Churchill Livingstone, 1982, vol. 1, ISBN: 9780443025044.
- [2] F. Cordella *et al.*, “Literature review on needs of upper limb prosthesis users,” *Frontiers in Neuroscience*, vol. 10, p. 209, May 2016. DOI: 10.3389/fnins.2016.00209.
- [3] V. Y. Ma, L. Chan, and K. J. Carruthers, “Incidence, prevalence, costs, and impact on disability of common conditions requiring rehabilitation in the United States: Stroke, spinal cord injury, traumatic brain injury, multiple sclerosis, osteoarthritis, rheumatoid arthritis, limb loss, and back pain,” *Archives of Physical Medicine and Rehabilitation*, vol. 95, no. 5, pp. 986–995, May 2014. DOI: 10.1016/j.apmr.2013.10.032.
- [4] T. P. Sheehan and G. C. Gondo, “Impact of limb loss in the United States,” *Physical Medicine and Rehabilitation Clinics of North America*, Amputee Rehabilitation, vol. 25, no. 1, pp. 9–28, Feb. 2014. DOI: 10.1016/j.pmr.2013.09.007.
- [5] A. Caldas, D. Matos, A. de Eyto, and N. Martins, “Redefining prosthetic needs: Insights from individuals with upper limb loss—a systematic review,” *Sensors*, vol. 26, no. 2, p. 734, 2026. DOI: 10.3390/s26020734.
- [6] C. E. de Putter *et al.*, “Health-related quality of life after upper extremity injuries and predictors for suboptimal outcome,” *Injury*, vol. 45, no. 11, pp. 1752–1758, Sep. 2014. DOI: 10.1016/j.injury.2014.07.016.
- [7] K. J. Zuo and J. L. Olson, “The evolution of functional hand replacement: From iron prostheses to hand transplantation,” *Plastic Surgery*, vol. 22, no. 1, pp. 44–51, 2014. DOI: 10.1177/229255031402200111.
- [8] T. W. Williams III, “Progress on stabilizing and controlling powered upper-limb prostheses,” *Journal of Rehabilitation Research and Development*, vol. 48, no. 6, pp. ix–xix, 2011. DOI: 10.1682/JRRD.2011.04.0078.
- [9] D. S. Childress, “Historical aspects of powered limb prostheses,” *Clinical Prosthetics and Orthotics*, vol. 9, no. 1, pp. 2–13, Jan. 1985.
- [10] L. Trent *et al.*, “A narrative review: Current upper limb prosthetic options and design,” *Disability and Rehabilitation: Assistive Technology*, vol. 15, no. 6, pp. 604–613, 2020. DOI: 10.1080/17483107.2019.1594403.

- [11] N. M. Bajaj, A. J. Spiers, and A. M. Dollar, "State of the art in prosthetic wrists: Commercial and research devices," in *Proceedings of the 2015 International Conference on Rehabilitation Robotics (ICORR)*, 2015, pp. 331–338. DOI: 10.1109/ICORR.2015.7281221.
- [12] R. Brack and E. H. Amalu, "A review of technology, materials and R&D challenges of upper limb prosthesis for improved user suitability," *Journal of Orthopaedics*, vol. 23, pp. 88–96, 2021. DOI: 10.1016/j.jor.2020.12.009.
- [13] L. Resnik, M. Borgia, and M. L. Clark, "Function and quality of life of unilateral major upper limb amputees: Effect of prosthesis use and type," *Archives of Physical Medicine and Rehabilitation*, vol. 101, no. 8, pp. 1396–1406, Aug. 2020. DOI: 10.1016/j.apmr.2020.04.003.
- [14] J. T. Belter, J. L. Segil, A. M. Dollar, and R. F. Weir, "Mechanical design and performance specifications of anthropomorphic prosthetic hands: A review," *Journal of Rehabilitation Research and Development*, vol. 50, no. 5, pp. 599–618, 2013. DOI: 10.1682/jrrd.2011.10.0188.
- [15] K. F. Gretschi, H. D. Lather, K. V. Peddada, C. R. Deeken, L. B. Wall, and C. A. Goldfarb, "Development of novel 3D-printed robotic prosthetic for transradial amputees," *Prosthetics and Orthotics International*, vol. 40, no. 3, pp. 400–403, 2016. DOI: 10.1177/0309364615579317.
- [16] J. Ten Kate, G. Smit, and P. Breedveld, "3D-printed upper limb prostheses: A review," *Disability and Rehabilitation: Assistive Technology*, vol. 12, no. 3, pp. 300–314, 2017. DOI: 10.1080/17483107.2016.1253117.
- [17] S. Micera, J. Carpaneto, and S. Raspopovic, "Control of hand prostheses using peripheral information," *IEEE Reviews in Biomedical Engineering*, vol. 3, pp. 48–68, Oct. 2010. DOI: 10.1109/RBME.2010.2085429.
- [18] E. Scheme and K. Englehart, "Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use," *Journal of Rehabilitation Research and Development*, vol. 48, no. 6, pp. 643–659, 2011. DOI: 10.1682/JRRD.2010.09.0177.
- [19] L. Resnik, S. L. Klinger, K. Etter, and C. Fantini, "Controlling a multi-degree of freedom upper limb prosthesis using foot controls: User experience," *Disability and Rehabilitation: Assistive Technology*, vol. 9, no. 4, pp. 318–329, Jul. 2014. DOI: 10.3109/17483107.2013.822024.
- [20] P. A. Parker, K. B. Englehart, and B. S. Hudgins, "Myoelectric signal processing for control of powered limb prostheses," *Journal of Electromyography and Kinesiology*, vol. 16, no. 6, pp. 541–548, Dec. 2006. DOI: 10.1016/j.jelekin.2006.08.006.
- [21] B. Peerdeman *et al.*, "Myoelectric forearm prostheses: State of the art from a user-centered perspective," *Journal of Rehabilitation Research and Development*, vol. 48, no. 6, pp. 719–737, 2011. DOI: 10.1682/jrrd.2010.08.0161.

- [22] J. J. Kim, J. Kim, J. Lee, and J. Shin, “Influence of lifestyle pattern on preference for prosthetic hands: Understanding the development pathway for 3D-printed prostheses,” *Journal of Cleaner Production*, vol. 379, p. 134 599, 2022. DOI: 10.1016/j.jclepro.2022.134599.
- [23] P. Geethanjali, “Myoelectric control of prosthetic hands: State-of-the-art review,” *Medical Devices: Evidence and Research*, vol. 9, pp. 247–255, Jul. 2016. DOI: 10.2147/MDER.S91102.
- [24] A. W. Shehata, H. E. Williams, J. S. Hebert, and P. M. Pilarski, “Machine learning for the control of prosthetic arms: Using electromyographic signals for improved performance,” *IEEE Signal Processing Magazine*, vol. 38, no. 4, pp. 46–53, Jul. 2021. DOI: 10.1109/MSP.2021.3075931.
- [25] C. Castellini *et al.*, “Proceedings of the first workshop on peripheral machine interfaces: Going beyond traditional surface electromyography,” *Frontiers in Neurorobotics*, vol. 8, no. 22, pp. 1–12, Aug. 2014. DOI: 10.3389/fnbot.2014.00022.
- [26] P. M. Pilarski and J. S. Hebert, “Upper and lower limb robotic prostheses,” in *Robotic Assistive Technologies: Principles and Practice*, P. Encarnação and A. M. Cook, Eds., Boca Raton, FL: CRC Press, 2017, pp. 99–144, ISBN: 9780367875138. DOI: 10.1201/9781315368788-5.
- [27] E. Biddiss and T. Chau, “Upper-limb prosthetics: Critical factors in device abandonment,” *American Journal of Physical Medicine and Rehabilitation*, vol. 86, no. 12, pp. 977–987, Dec. 2007. DOI: 10.1097/PHM.0b013e3181587f6c.
- [28] S. Salminger *et al.*, “Current rates of prosthetic usage in upper-limb amputees – have innovations had an impact on device acceptance?” *Disability and Rehabilitation*, vol. 44, no. 14, pp. 3708–3713, Jul. 2022. DOI: 10.1080/09638288.2020.1866684.
- [29] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone, “Deep reinforcement learning for robotics: A survey of real-world successes,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 8, pp. 153–188, 2025. DOI: 10.1146/annurev-control-030323-022510.
- [30] G. Dulac-Arnold *et al.*, “Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis,” *Machine Learning*, vol. 110, no. 9, pp. 2419–2468, 2021. DOI: 10.1007/s10994-021-05961-4.
- [31] I. Ku, G. K. Lee, C. Y. Park, J. Lee, and E. Jeong, “Clinical outcomes of a low-cost single-channel myoelectric-interface three-dimensional hand prosthesis,” *Archives of Plastic Surgery*, vol. 46, no. 4, pp. 303–310, 2019.
- [32] C. Antfolk, M. D’Alonzo, B. Rosén, G. Lundborg, F. Sebelius, and C. Cipriani, “Sensory feedback in upper limb prosthetics,” *Expert Review of Medical Devices*, vol. 10, no. 1, pp. 45–54, Jan. 2013. DOI: 10.1586/erd.12.68.

- [33] A. Rajeswaran *et al.*, “Learning complex dexterous manipulation with deep reinforcement learning and demonstrations,” *arXiv*, 1709.10087 [cs.LG], 2018. DOI: 10.48550/arXiv.1709.10087.
- [34] D. J. Brenneis, M. R. Dawson, and P. M. Pilarski, “Development of the HANDi Hand: An inexpensive, multi-articulating, sensorized hand for machine learning research in myoelectric control,” in *Proceedings of the 2017 Myoelectric Controls Symposium (MEC’17)*, Fredericton, New Brunswick, Aug. 2017, pp. 102–105.
- [35] L. H. B. Huinink, H. Bouwsema, D. H. Plettenburg, C. K. van der Sluis, and R. M. Bongers, “Learning to use a body-powered prosthesis: Changes in functionality and kinematics,” *Journal of NeuroEngineering and Rehabilitation*, vol. 13, no. 1, p. 90, 2016. DOI: 10.1186/s12984-016-0197-7.
- [36] G. H. Kejlaa, “Consumer concerns and the functional value of prostheses to upper limb amputees,” *Prosthetics and Orthotics International*, vol. 17, no. 3, pp. 157–163, 1993. DOI: 10.3109/03093649309164376.
- [37] B. Maat, G. Smit, D. Plettenburg, and P. Breedveld, “Passive prosthetic hands and tools: A literature review,” *Prosthetics and Orthotics International*, vol. 42, no. 1, pp. 66–74, 2018. DOI: 10.1177/0309364617691622.
- [38] A. Kannenberg, “Active upper-limb prostheses: The international perspective,” *JPO: Journal of Prosthetics and Orthotics*, vol. 29, no. 4S, pp. 57–62, 2017. DOI: 10.1097/JPO.0000000000000158.
- [39] S. L. Carey, D. J. Lura, and M. J. Highsmith, “Differences in myoelectric and body-powered upper-limb prostheses: Systematic literature review,” *Journal of Rehabilitation Research & Development*, vol. 52, no. 3, pp. 247–262, 2015. DOI: 10.1682/JRRD.2014.08.0192.
- [40] exiii Inc., *HACKberry: Open-source 3d-printable bionic hand*, GitHub Repository, 2015. [Online]. Available: <https://github.com/exiii-REVOLVER/HACKberry>.
- [41] M. A. Battraw, P. R. Young, W. M. Joiner, and J. S. Schofield, “A multiarticulate pediatric prosthetic hand for clinical and research applications,” *Frontiers in Robotics and AI*, vol. 9, no. 1000159, Oct. 2022. DOI: 10.3389/frobt.2022.1000159.
- [42] G. Langevin, *InMoov hand and forearm*, G. Langevin, Ed., Open-source Robotics Project, Accessed: April 4, 2024, Mar. 2012. [Online]. Available: <https://inmoov.fr/hand-and-forearm/>.
- [43] BLINC Lab, *HANDi Hand: Open-source multi-articulating robotic prosthetic hand*, GitHub Repository, Dec. 2017. [Online]. Available: <https://github.com/BLINCdev/HANDi-Hand>.
- [44] C. M. Light, P. H. Chappell, and P. J. Kyberd, “Establishing a standardized clinical assessment tool of pathologic and prosthetic hand function: Normative data, reliability, and validity,” *Archives of physical medicine and rehabilitation*, vol. 83, no. 6, Jun. 2002. DOI: 10.1053/apmr.2002.32737.

- [45] L. Resnik, M. Borgia, J. M. Cancio, J. Delikat, and P. Ni, “Psychometric evaluation of the southampton hand assessment procedure (SHAP) in a sample of upper limb prosthesis users,” *Journal of Hand Therapy*, vol. 36, no. 1, pp. 110–120, Jan. 2023. DOI: 10.1016/j.jht.2021.07.003.
- [46] Physiopedia, *Action Research Arm Test (ARAT)*, Jan. 2022. [Online]. Available: [https://www.physio-pedia.com/index.php?title=Action_Research_Arm_Test_\(ARAT\)&oldid=290378](https://www.physio-pedia.com/index.php?title=Action_Research_Arm_Test_(ARAT)&oldid=290378).
- [47] D. Carroll, “A quantitative test of upper extremity function,” *Journal of Chronic Disability*, vol. 18, no. 5, pp. 479–491, May 1965. DOI: 10.1016/0021-9681(65)90030-5.
- [48] I. Llop-Harillo, A. Pérez-González, J. Starke, and T. Asfour, “The Anthropomorphic Hand Assessment Protocol (AHAP),” *Robotics and Autonomous Systems*, vol. 121, no. 103259, Nov. 2019. DOI: 10.1016/j.robot.2019.103259.
- [49] I. Llop-Harillo, A. Pérez-González, and J. Andrés-Esperanza, “Grasping Ability and Motion Synergies in Affordable Tendon-Driven Prosthetic Hands Controlled by Able-Bodied Subjects,” *Frontiers in Neurorobotics*, vol. 14, p. 57, Sep. 2020. DOI: 10.3389/fnbot.2020.00057.
- [50] J. R. Seigel, M. A. Battraw, E. J. Winslow, M. A. James, W. M. Joiner, and J. S. Schofield, “Review and critique of current testing protocols for upper-limb prostheses: A call for standardization amidst rapid technological advancements,” *Frontiers in Robotics and AI*, vol. 10, no. 1292632, Nov. 2023. DOI: 10.3389/frobt.2023.1292632.
- [51] J. R. Siegel *et al.*, “A performance evaluation of commercially available and 3D-printable prosthetic hands: A comparison using the anthropomorphic hand assessment protocol,” *BMC Biomedical Engineering*, vol. 6, no. 1, p. 11, Dec. 2024. DOI: 10.1186/s42490-024-00086-w.
- [52] T. W. Williams III, “Controlling powered upper extremity prostheses now and in the future,” *Proceedings of the Myoelectric Controls Symposium (MEC’05)*, 2005.
- [53] M. Davidson, C. Bodine, and R. F. f. Weir, “User surveys support designing a prosthetic wrist that incorporates the Dart Thrower’s Motion,” *Disability and Rehabilitation: Assistive Technology*, vol. 14, no. 3, pp. 312–315, 2019. DOI: 10.1080/17483107.2018.1447607.
- [54] S. Hubbard, W. Heim, S. Naumann, S. Glasford, G. Montgomery, and S. Ramdial, “Powered upper limb prosthetic practice in paediatrics,” in *Powered Upper Limb Prostheses: Control, Implementation and Clinical Application*, A. Muzumdar, Ed., Berlin, Heidelberg: Springer, 2004, ch. 6, pp. 85–115. DOI: 10.1007/978-3-642-18812-1_6.
- [55] A. L. Edwards *et al.*, “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, Oct. 2016. DOI: 10.1177/0309364615605373.

- [56] K. Englehart and B. Hudgins, “A robust, real-time control scheme for multi-function myoelectric control,” *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 7, pp. 848–854, 2003. DOI: 10.1109/TBME.2003.813539.
- [57] B. A. Lock, A. M. Simon, K. Stubblefield, and L. J. Hargrove, “Prosthesis-guided training for practical use of pattern recognition control of prostheses,” in *Proceedings of the 2011 Myoelectric Controls Symposium (MEC’11)*, Fredericton, NB, Canada, 2011, pp. 61–64.
- [58] E. Campbell *et al.*, “(Un)supervised (Co)adaptation via incremental learning for myoelectric control: Motivation, review, and future directions,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 33, pp. 3565–3582, Aug. 2025. DOI: 10.1109/TNSRE.2025.3602397.
- [59] L. V. McFarland, S. L. H. Winkler, A. W. Heinemann, M. Jones, and A. Esquenazi, “Unilateral upper-limb loss: Satisfaction and prosthetic-device use in veterans and servicemembers from Vietnam and OIF/OEF conflicts,” *Journal of Rehabilitation Research and Development*, vol. 47, no. 4, pp. 299–316, 2010. DOI: 10.1682/jrrd.2009.03.0027.
- [60] D. S. V. Bandara, R. A. R. C. Gopura, K. T. M. U. Hemapala, and K. Kiguchi, “Upper extremity prosthetics: Current status, challenges and future directions,” in *The Seventeenth International Symposium on Artificial Life and Robotics (AROB)*, 2012, pp. 875–880. DOI: 10.5954/icarob.2012.875.
- [61] C. Castellini, “Incremental learning of muscle synergies: From calibration to interaction,” in *Human and Robot Hands*, ser. Springer Series on Touch and Haptic Systems, M. Bianchi and A. Moscatelli, Eds., Cham, Switzerland, 2016, pp. 171–193. DOI: 10.1007/978-3-319-22396-4_9.
- [62] K. Khetarpal, M. Riemer, I. Rish, and D. Precup, “Towards continual reinforcement learning: A review and perspectives,” *Journal of Artificial Intelligence Research*, vol. 75, pp. 1401–1476, 2022. DOI: 10.1613/jair.1.13673.
- [63] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd. Cambridge, Massachusetts: The MIT Press, 2018, ISBN: 9780262039246.
- [64] J. Modayil, A. White, and R. S. Sutton, “Multi-timescale nexting in a reinforcement learning robot,” *Adaptive Behavior*, vol. 22, no. 2, pp. 146–160, Apr. 2014. DOI: 10.1177/1059712313511648.
- [65] R. S. Sutton *et al.*, “Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction,” in *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, vol. 2, Taipei, Taiwan: International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 761–768.
- [66] R. S. Sutton, “Learning to predict by the methods of temporal differences,” *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1988. DOI: 10.1007/BF00115009.

- [67] E. Campbell, U. Côté-Allard, and E. Scheme, “Screen guided training does not capture goal-oriented behaviors: Learning myoelectric control mappings from scratch using context informed incremental learning,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 33, pp. 332–342, 2025. DOI: 10.1109/TNSRE.2024.3516086.
- [68] P. M. Pilarski, M. R. Dawson, T. Degris, J. P. Carey, and R. S. Sutton, “Dynamic switching and real-time machine learning for improved human control of assistive biomedical robots,” in *Proceedings of the 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Rome, Italy, Jun. 2012, pp. 296–302. DOI: 10.1109/BioRob.2012.6290309.
- [69] A. L. Edwards, “Adaptive and autonomous switching: Shared control of powered prosthetic arms using reinforcement learning,” Master’s thesis, University of Alberta, Edmonton, AB, Canada, Nov. 2016. DOI: 10.7939/R35Q4RX49.
- [70] 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, Jun. 2009. DOI: 10.1109/TNSRE.2009.2023282.
- [71] M. R. Dawson *et al.*, “Joint action is a framework for understanding partnerships between humans and upper limb prostheses,” in *Proceedings of the 2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, Heidelberg, Germany: IEEE, Sep. 2024, pp. 605–611. DOI: 10.1109/BioRob60516.2024.10719838.
- [72] J. S. Müller, C. Vidaurre, M. Schreuder, F. C. Meinecke, P. von Büna, and K.-R. Müller, “A mathematical model for the two-learners problem,” *Journal of Neural Engineering*, vol. 14, no. 3, Mar. 2017. DOI: 10.1088/1741-2552/aa620b.
- [73] D. De Santis, “A framework for optimizing co-adaptation in body-machine interfaces,” *Frontiers in Neurorobotics*, vol. 15, no. 662181, 2021. DOI: 10.3389/fnbot.2021.662181.
- [74] E. Campbell, E. Eddy, S. Bateman, U. Côté-Allard, and E. Scheme, “Context-informed incremental learning improves both the performance and resilience of myoelectric control,” *Journal of NeuroEngineering and Rehabilitation*, vol. 21, no. 1, p. 70, 2024. DOI: 10.1186/s12984-024-01362-w.
- [75] M. R. Dawson, C. Sherstan, J. P. Carey, J. S. Hebert, and P. M. Pilarski, “Development of the Bento Arm: An improved robotic arm for myoelectric training and research,” in *Proceedings of the 2014 Myoelectric Controls Symposium (MEC’14)*, Fredericton, New Brunswick, Aug. 2014, pp. 60–64.
- [76] A. C. Lau, M. R. Dawson, and P. M. Pilarski, “Continual prediction learning techniques to support users’ grasp selection on a multi-articulating prosthetic hand,” in *Proceedings of the 38th Canadian Conference on Artificial Intelligence*, ser. Canadian AI, Canadian Artificial Intelligence Association (CAIAC), May 2025, pp. 1–7. DOI: 10.21428/594757db.b3137577.

- [77] A. L. Edwards, J. S. Hebert, and P. M. Pilarski, "Machine learning and un-learning to autonomously switch between the functions of a myoelectric arm," in *Proceedings of the 2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Singapore, Jun. 2016, pp. 514–521. DOI: 10.1109/BIOROB.2016.7523678.
- [78] P. M. Pilarski *et al.*, "Adaptive artificial limbs: A real-time approach to prediction and anticipation," *IEEE Robotics & Automation Magazine*, vol. 20, no. 1, pp. 53–64, Mar. 2013. DOI: 10.1109/MRA.2012.2229948.
- [79] A. Fougner, E. Scheme, A. D. C. Chan, K. Englehart, and Ø. Stavdahl, "Resolving the limb position effect in myoelectric pattern recognition," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 6, pp. 644–651, Dec. 2011. DOI: 10.1109/TNSRE.2011.2163529.
- [80] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic, "The GRASP taxonomy of human grasp types," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 66–77, Feb. 2016. DOI: 10.1109/THMS.2015.2470657.
- [81] P. Faridi *et al.*, "Machine-learned adaptive switching in voluntary lower-limb exoskeleton control: Preliminary results," in *Proceedings of the 2022 International Conference on Rehabilitation Robotics (ICORR)*, Rotterdam, The Netherlands, Jul. 2022, pp. 1–6. DOI: 10.1109/ICORR55369.2022.9896611.
- [82] M. R. Dawson, H. E. Williams, G. Murgatroyd, J. S. Hebert, and P. M. Pilarski, "brachIOplexus: Myoelectric training software for clinical and research applications," in *Proceedings of the Myoelectric Controls Symposium (MEC'20)*, Fredericton, New Brunswick, Canada, Jul. 2020, pp. 87–90. DOI: 10.57922/mec.40.
- [83] A. Gibb, *Building Open Source Hardware: DIY Manufacturing for Hackers and Makers*. Addison-Wesley, 2015, ISBN: 9780321906045.
- [84] IKEA, *Assembly Instructions*, Ingka Holding B.V., Delft, Netherlands. [Online]. Available: <https://www.ikea.com>.
- [85] The LEGO Group, *Building Instructions*, LEGO System A/S, Billund, Denmark. [Online]. Available: <https://www.lego.com/service/buildinginstructions>.
- [86] Bandai Spirits, *Construction Manual: (HGTWFM) X-EX01 Gundam Calibarn*, Kit Code/Jan Code: 4573102653222, Bandai Namco Group, Tokyo, Japan, 2023. [Online]. Available: <https://manual.bandai-hobby.net>.
- [87] ROBOTIS, *DynamixelSDK*, <https://github.com/ROBOTIS-GIT/DynamixelSDK>, 2026.
- [88] J. V. V. Parr *et al.*, "A tool for measuring mental workload during prosthesis use: The Prosthesis Task Load Index (PROS-TLX)," *PLOS ONE*, vol. 18, no. 5, May 2023. DOI: 10.1371/journal.pone.0285382.

Appendix A: The HANDi Hand Assembly Manual

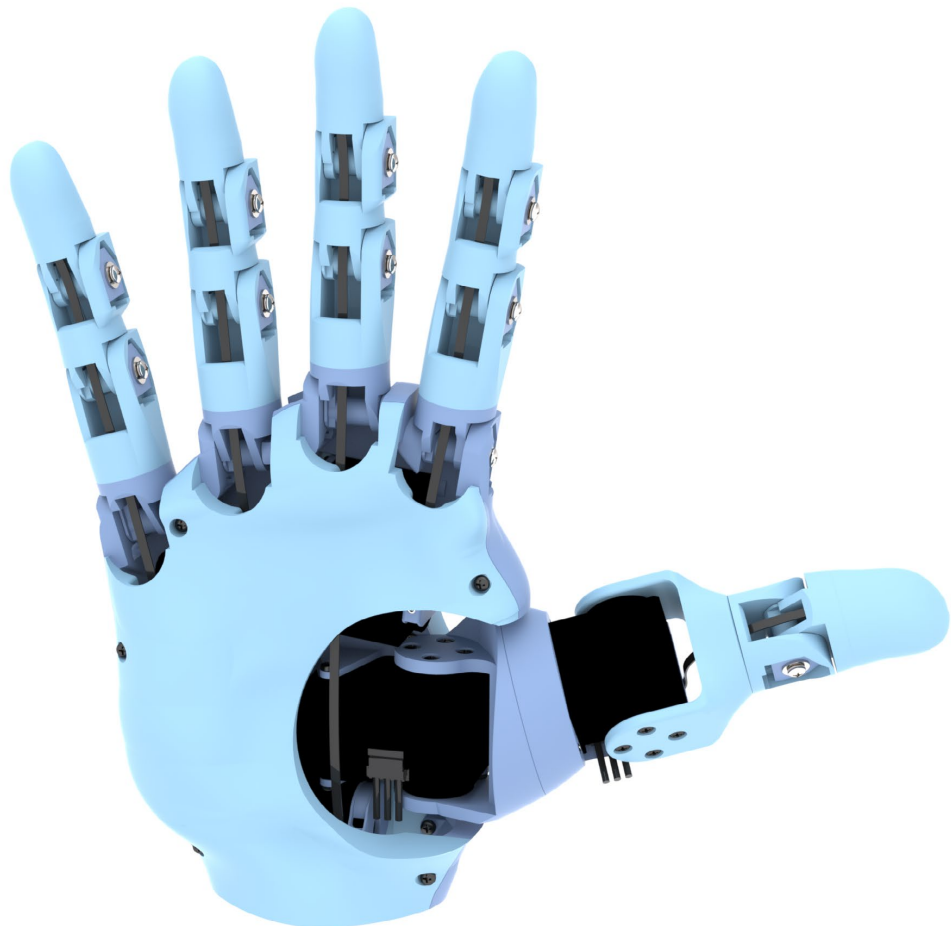


Assembly Guide

Annette C. Lau

Version 1.0

May 15, 2026



Contents

1. Introduction	4
1.1. Background	4
1.2. How to Use this Assembly Manual	4
2. Naming & Definitions	5
3. Required Materials	7
3.1. 3D-Printed Parts	7
3.2. Ordered Parts	14
3.3. Tools	14
4. Material Preparation	15
4.1 3D Printed Parts	15
4.2. Zip Ties	16
5. Dynamixel Servo Preparation	18
5.1. Materials Needed	18
5.2. Instructions	18
6. Before You Assemble	23
6.1. Symbols for Assembly	23
7. Thumb Assembly	24
7.1. Materials Needed	24
7.2. Instructions	24
8. Finger Assembly	29
8.1. Materials Needed	29
8.2. Instructions	29
9. Palm Assembly	32
9.1. Materials Needed	32
9.2. Instructions	32
Palm Grips	42
10. After You Assemble	43

10.1. Wiring	43
10.2. Software	44
10.3. Optional Add-Ons	46
Appendix A: 3D Printed Template	51
A.1. Introduction	51

1. Introduction

1.1. Background

The **Humanoid Anthropometric Naturally Dexterous Intelligent (HANDi) Hand** is an open-source, inexpensive, prosthetic hand capable of providing real-time sensory information for machine learning and prosthetics research. Originally designed in 2017 by Brenneis et al., the newest release of the HANDi Hand includes updates to make the system more robust for longer-term machine learning experiments.

The open-source release provides all the solid modelling files, .stl files, Arduino code, and assembly instructions required to construct a fully functional HANDi Hand. It should also give the maker the flexibility to alter the design as necessary to suit their needs or 3D printer settings. Currently, only the right hand version is available. To contact the designers or receive support for your build, please visit BLINCdev.ca.

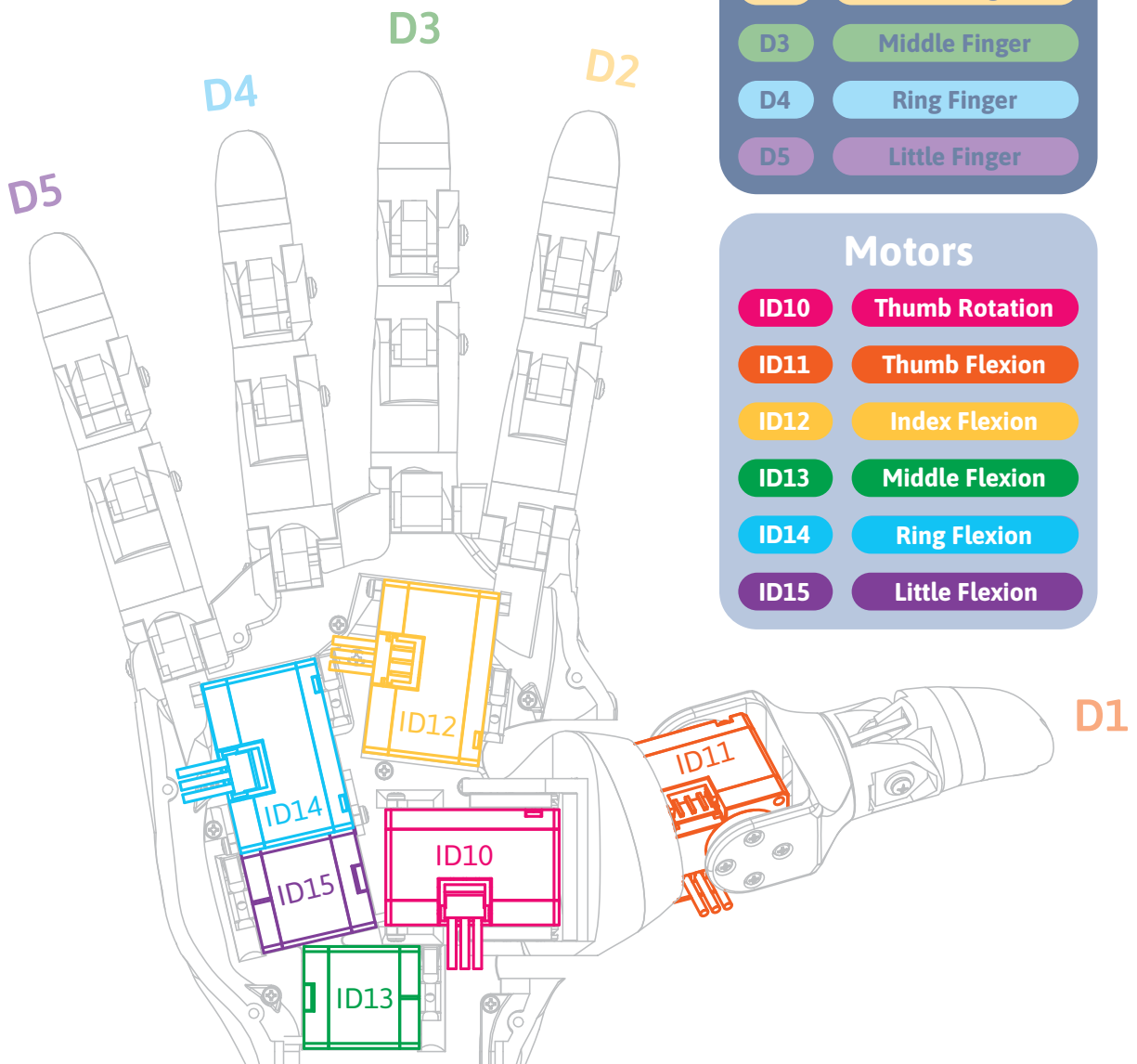
This assembly manual outlines all the information required to print/source parts and assemble the HANDi Hand as designed. The hand takes an estimated 5 hours to assemble (not including 3D-printing time) and costs less than \$400 CAD to build the basic kit without additional sensors.

1.2. How to Use this Assembly Manual

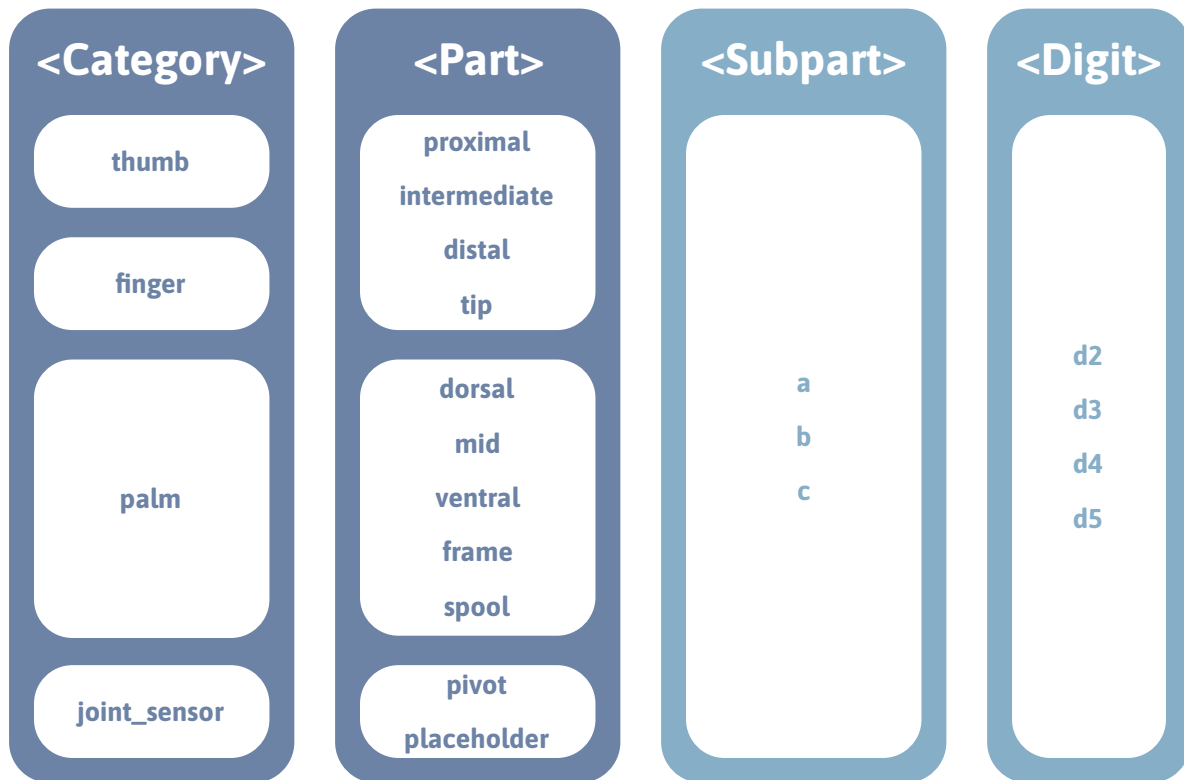
This guide recommends the maker has access to a computer with Windows 10 or newer. It is recommended to follow this assembly manual from beginning to end rather than jumping around between different sections.

2. Naming & Definitions

The figure below displays the different parts of the HANDi Hand that have a naming convention. The finger digits range from D1 to D5 while the motor IDs range from ID10 to ID15.

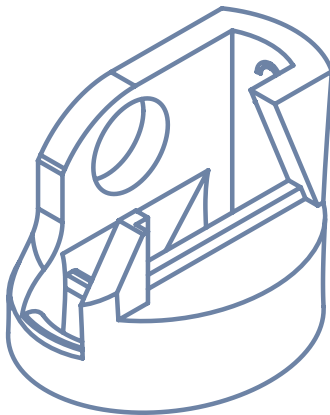


The figure below outlines the naming convention for the 3D-printed parts of the hand. **<Category>** and **<Part>** are listed by default while **<Subpart>** and **<Digit>** are used optionally to specify part function even further.



For example, the following part would be labeled:

finger_proximal_a_d2



3. Required Materials

3.1. 3D-Printed Parts

3.1.1. Introduction

The 3D-printed parts are designed to be printed in PLA. All parts are designed for the print tolerance of a Bambu X1C printer. However, these tolerances, such as PLA shrinkage, can be altered in the [common_dimensions](#) file.

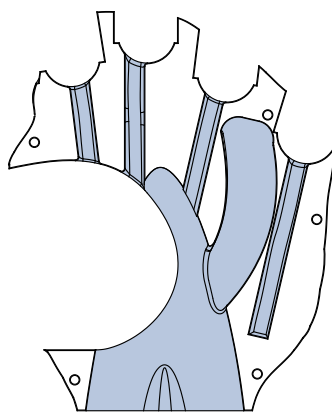
All files required for 3D printing can be accessed via [GitHub](#). This includes the stl, 3mf, and gcode files. The source SolidWorks parts and assembly files are also included for those who may want to customize the parts to their needs. The suggested print specifications for each part are found in the table on page 9. This table lists all the part sets that must be printed for a complete hand. If all parts are printed using Bambu Lab's PLA Basic, it should cost about \$50 CAD to print. If none of the parts are being altered or customized, use the gcode files to quickly get started.

3.1.2. Print Settings

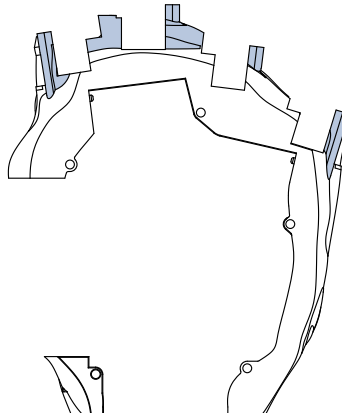
The table below specifies the print settings for each preset used in the HANDi Hand.

Print Preset	Layer Height (mm)	Wall Loops	Sparse Infill Pattern	Sparse Infill Density (%)	Supports	Brim
HH Default	0.2	2	Honeycomb	15	None	Outer 5mm
HH Supports	0.2	2	Honeycomb	15	Manual	Outer 5mm
HH Strong	0.2	2	Rectilinear	50	None	Outer 5mm

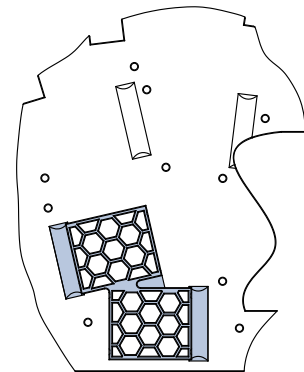
The parts that need manual supports require supports in specific areas. Below is a diagram of areas that need support. These are all views from the perspective of the build bed.



palm_ventral



palm_mid



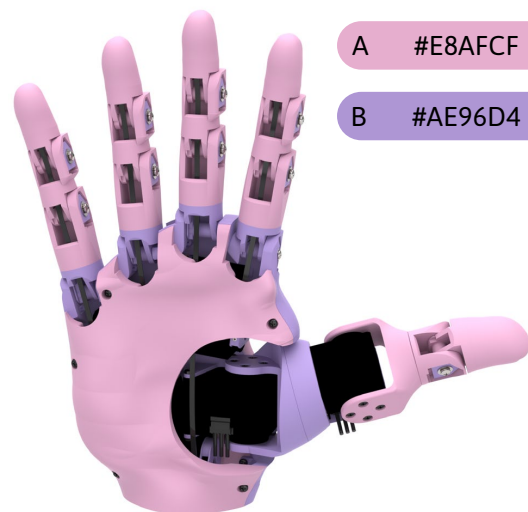
palm_dorsal

3.1.3. Colourways

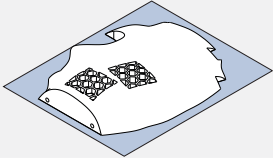
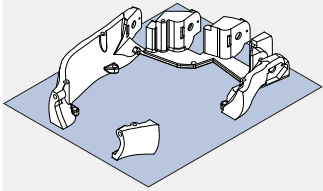
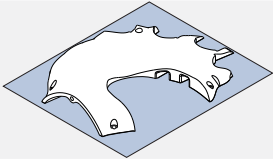
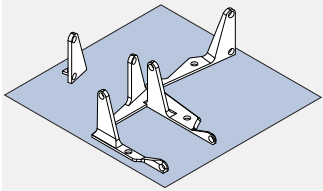
There are many different colourways to print the HANDi Hand in. The parts have been split into print groups according to the following colour schemes (A and B in the print guide). The standard colourway uses white PLA.

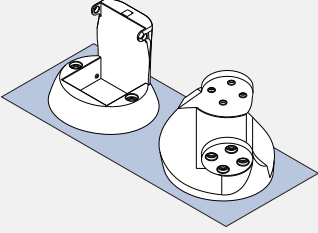
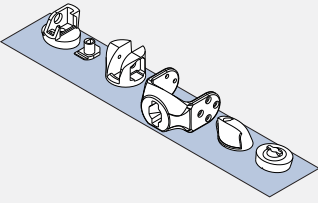


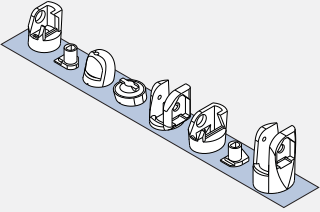
Additional colourway options are shown below as inspiration. However, these colours can be changed to the maker's liking.

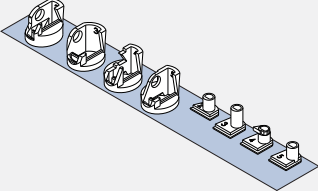


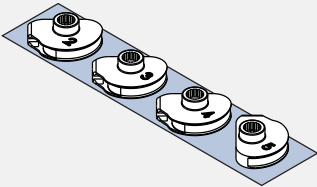
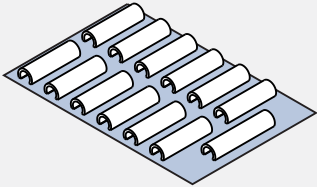
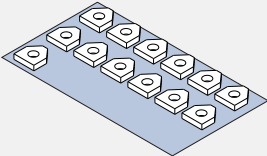
3.1.4. Print Guide

Set Name	Parts in Set	Build Orientation	Colour	Print Specs	Time (min)	Material (g)
hh_palm_dorsal	palm_dorsal		A	HH Supports	82	33.47
hh_palm_mid	palm_mid_a		B	HH Supports	57	19.54
	palm_mid_b					
hh_palm_ventral	palm_ventral		A	HH Supports	91	28.90
hh_palm_frame	palm_frame_a		B	HH Strong	20	3.09
	palm_frame_b					

Set Name	Parts in Set	Build Orientation	Colour	Print Specs	Time (min)	Material (g)
hh_thumb_proximal	thumb_proximal_a		B	HH Default	42	14.70
	thumb_proximal_b					
hh_thumb	thumb_distal_a		A	HH Default	38	12.88
	thumb_distal_b					
	thumb_intermediate_a					
	thumb_intermediate_b					
	thumb_tip_a					
	thumb_tip_b					

Set Name	Parts in Set	Build Orientation	Colour	Print Specs	Time (min)	Material (g)
hh_finger (x4)	finger_proximal_c		A	HH Default	136	36.16
	finger_intermediate_a					
	finger_intermediate_b					
	finger_intermediate_c					
	finger_distal_a					
	finger_distal_b					
	finger_tip_a					
	finger_tip_b					

Set Name	Parts in Set	Build Orientation	Colour	Print Specs	Time (min)	Material (g)
hh_finger_proximal	finger_proximal_a_d2		B	HH Default	28	7.03
	finger_proximal_b_d2					
	finger_proximal_a_d3					
	finger_proximal_b_d3					
	finger_proximal_a_d4					
	finger_proximal_b_d4					
	finger_proximal_a_d5					
	finger_proximal_b_d5					

Set Name	Parts in Set	Build Orientation	Colour	Print Specs	Time (min)	Material (g)
hh_palm_spool	palm_spool_d2		B	HH Strong	19	4.17
	palm_spool_d3					
	palm_spool_d4					
	palm_spool_d5					
hh_joint_sensor_pivot	joint_sensor_pivot (QTY:12)		B	HH Default	12	1.47
	joint_sensor_pivot_d3 (QTY:1)					
hh_joint_sensor_placeholder	joint_sensor_placeholder (QTY: 13)		B	HH Default	18	3.64
Total:					9 h 3 m	165.05 g

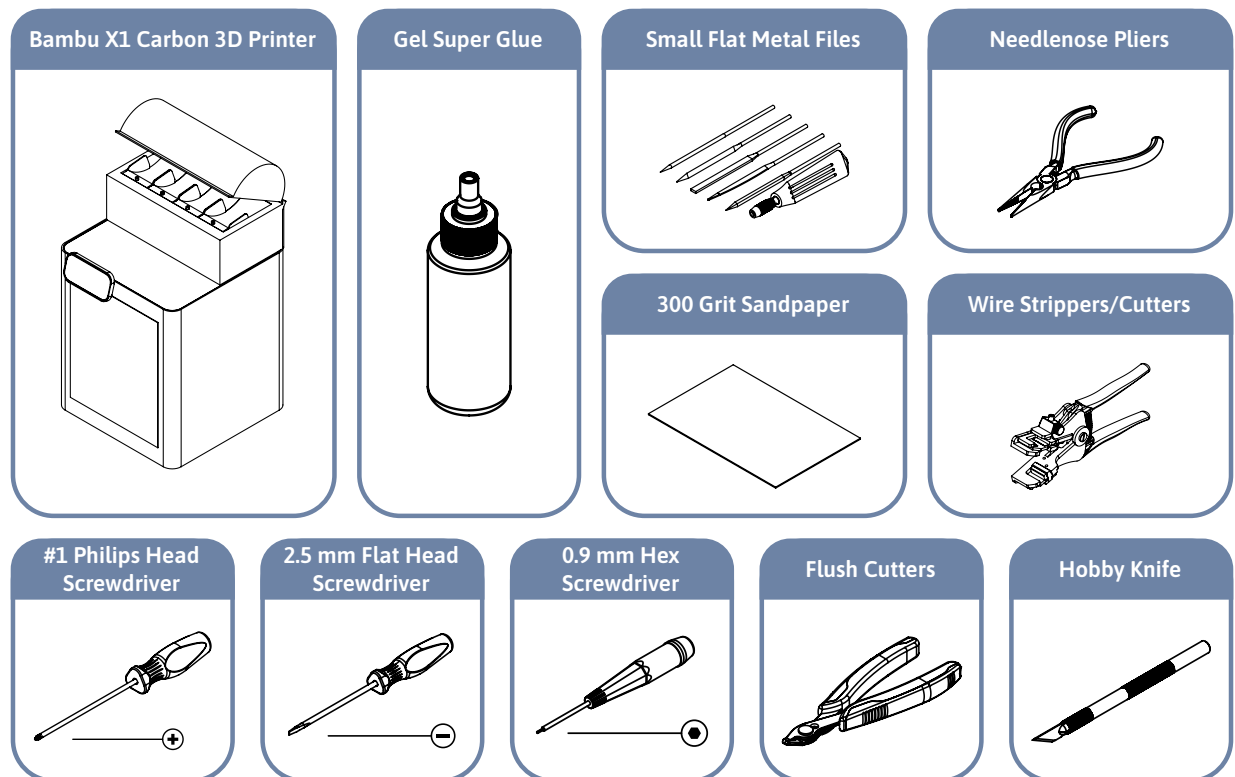
3.2. Ordered Parts

The [link here](#) leads to the public Google Sheets file that contains all the items (excluding tools) required for building a complete HANDi Hand. Makers may create a copy of this spreadsheet for their own convenience.

Select the kit using the dropdown menu depending on the additional sensors desired. If you've selected any kit other than the base kit, additional instructions are included at the end of the assembly guide.

3.3. Tools

Below are the tools required for building the HANDi Hand.



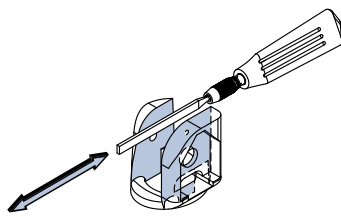
4. Material Preparation

4.1 3D Printed Parts

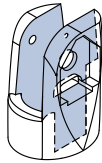
Prints the parts as specified in the 3D-Printed Parts section. Although all parts are designed to involve as little post-processing as possible, some parts should be prepared further to result in the best finished product.

4.1.1. Fingers & Thumb

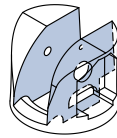
The following finger and thumb pieces need to be filed to ensure smooth flexion. Use a small metal file to buff the blue regions in the figures below until the faces are smooth to the touch. Additionally, the locking wings on all **finger_tip_b** parts should be filed slightly. The goal is to have them twist into a locked position with the **finger_distal_a** parts. They should be filed such that the twisting motion is possible, but sufficient friction still exists to keep the fingertips in the locked position.



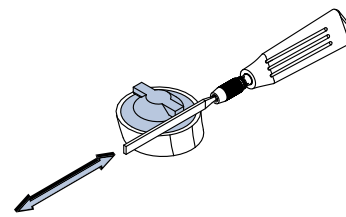
finger_
intermediate_c



finger_
proximal_c



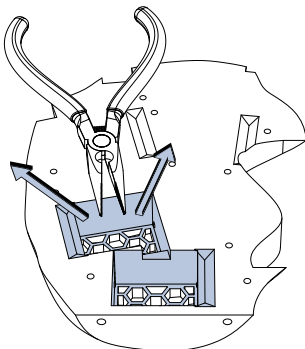
thumb_
intermediate_a



finger_tip_b

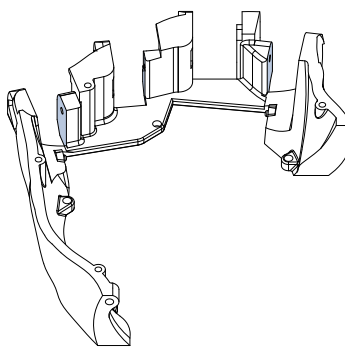
4.1.2. Dorsal Palm

Remove the print supports using needlenose pliers, then sand smooth using a small metal file if needed.



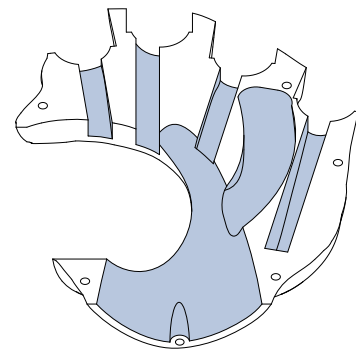
4.1.3. Mid Palm

Remove the print supports using needlenose pliers, then sand smooth using a small metal file if needed. Use the file to buff the blue regions similar to the previous finger/thumb parts.



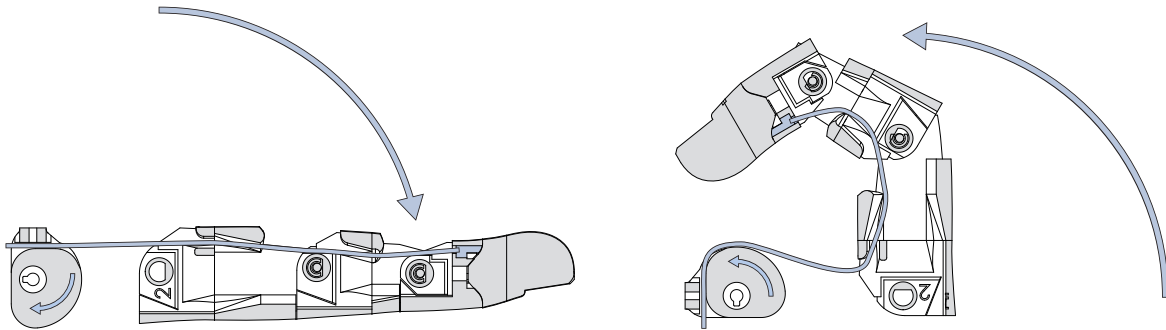
4.1.4. Ventral Palm

Remove the print supports using needlenose pliers, then sand smooth using a small metal file or sandpaper if needed. File the four zip tie channels to reduce friction.

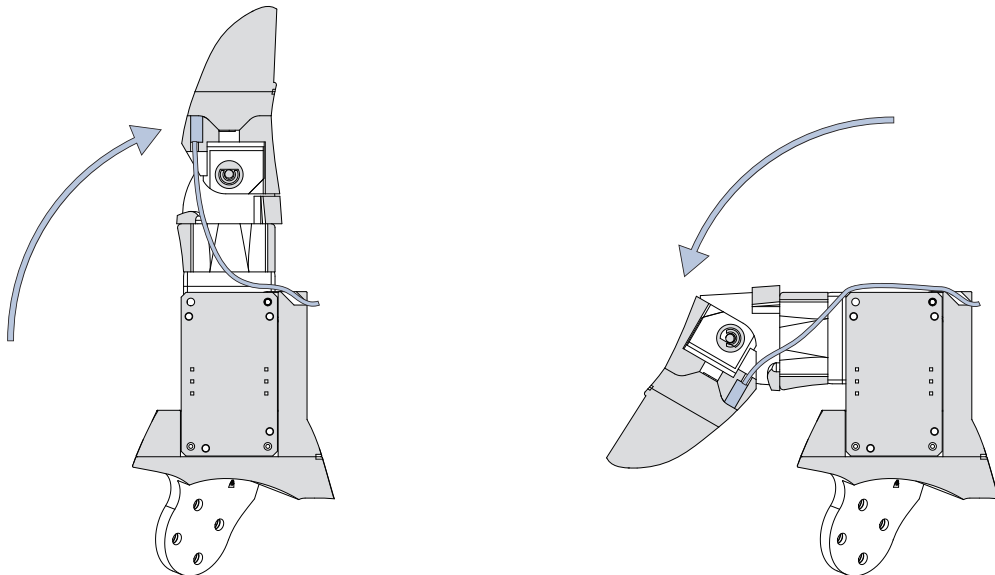


4.2. Zip Ties

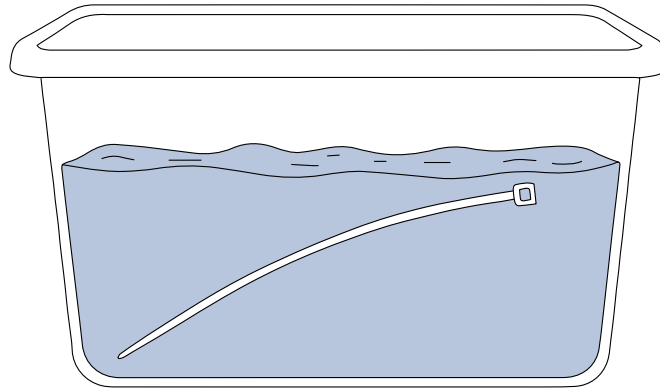
The flexion of each finger is actuated by a zip tie “tendon”. One end of the zip tie is secured at the finger tip (**finger_distal_a**). The other end is secured to the round spool (**palm_spool_dx**) connected to the Dynamixel motor. When the motor rotates, the zip tie tugs on the finger tip, flexing the entire finger. The diagram below shows a cutaway view of the finger actuation mechanism where the zip tie is shown in blue.



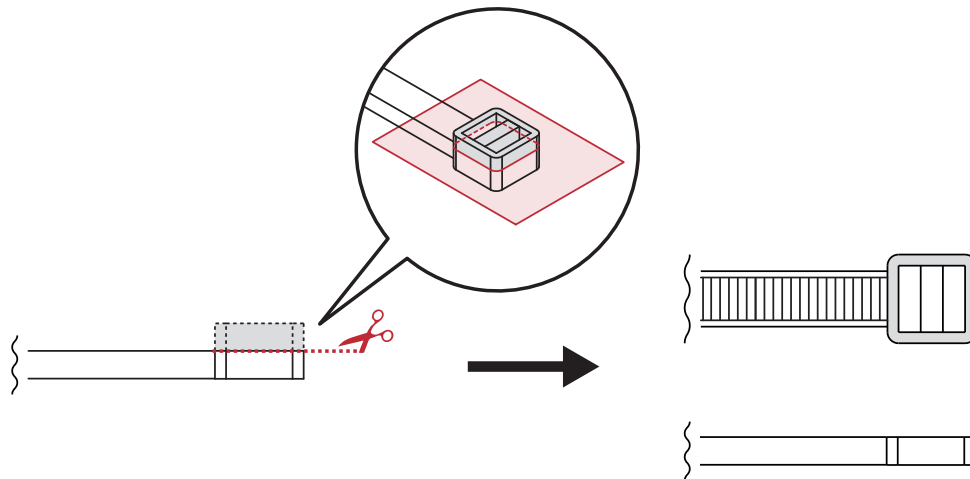
The thumb operates similarly to the finger. One end of the zip tie is secured at the thumb tip (**thumb_distal_a**) while the other end is secured to the back of the thumb (**thumb_proximal_a**). When the motor rotates, the zip tie tugs on the thumb tip, wraps around the motor, and flexes the entire thumb. The diagram below shows a cutaway view of the thumb actuation mechanism where the zip tie is shown in blue.



Since the zip ties are made of nylon 6/6, they are prone to becoming brittle due to a lack of moisture. To prevent the zip ties from snapping during regular use, place the zip ties into a tub of water and ensure they are fully submerged. Leave the zip ties in the water for 24 hours or more before installing them in the HANDi Hand.

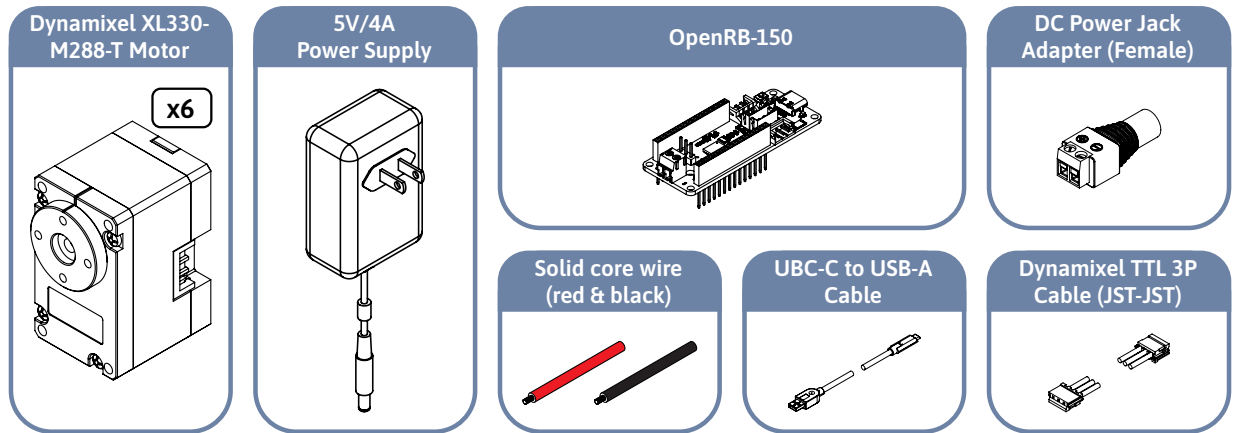


To prepare the zip ties to fit in the finger and thumb tip, cut a zip tie's receiving end so that the thickness is constant along the entire zip tie. Flush cutters work best for this operation and fine adjustments can be made using a hobby knife. This cut is illustrated below. This end of the zip tie will serve as the stopper at the tip of the finger or thumb. Repeat this cut for a total of five zipties.



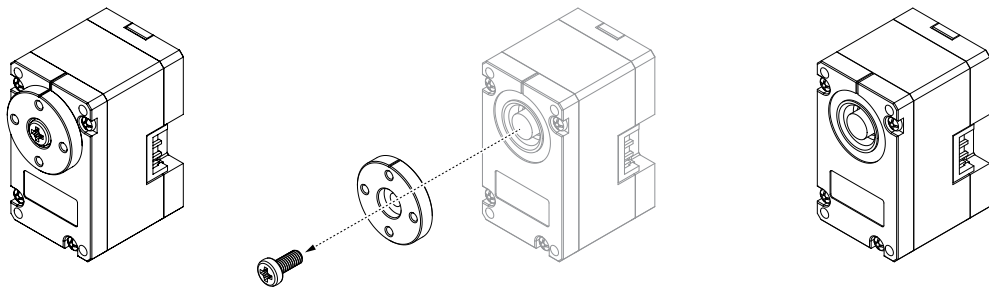
5. Dynamixel Servo Preparation

5.1. Materials Needed

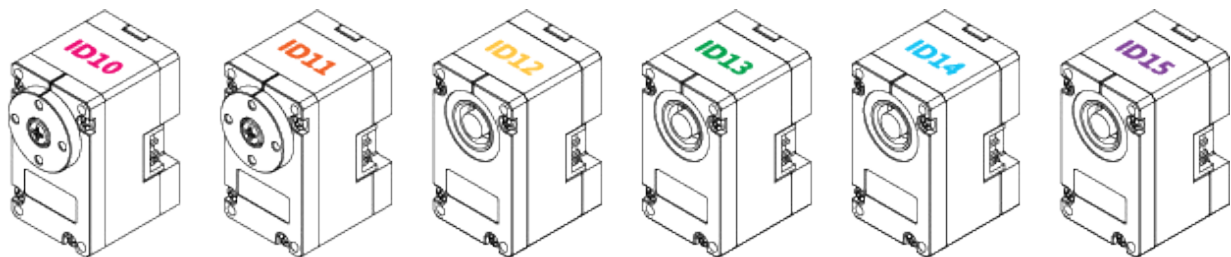


5.2. Instructions

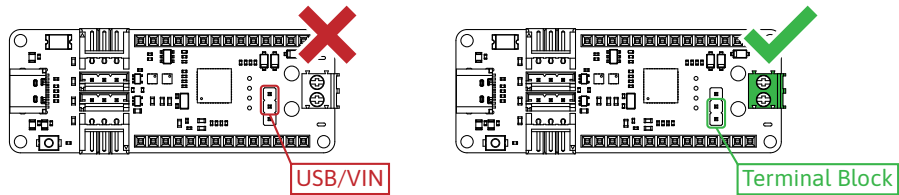
1. Set aside two Dynamixel motors as is. Use a #1 Philips Head screwdriver to remove the default servo horns for the other four motors. Hold the default servo horn still while removing it.



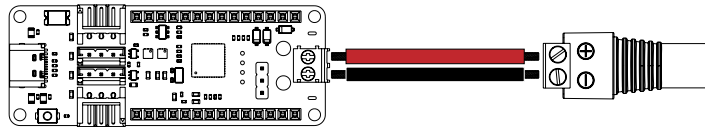
2. Use the label maker to label the 2 motors you've set aside ID10 and ID11. Label the remaining four motors ID12 - ID15.



3. Ensure the power source jumper on the OpenRB-150 controller is set to VIN(DXL). When in this mode, the board and connected servomotors will be powered via the green terminal block.



4. Connect the DC Barrel Jack (female) to the green terminal block on the OpenRB-150 using the red and black wires. Make sure that the red wire goes into the side labelled “+” and the black wire goes into the side labelled “-”. Once the wires are in place, use a 2.5 mm flat head screw driver to tighten the terminal blocks until the wires are secure.



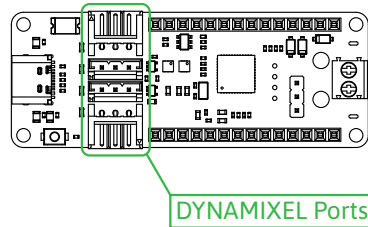
5. Install the Arduino IDE and set it up to work with the OpenRB-150a.
 - a. Download the legacy version of the [Arduino IDE 1.8.19](#) from the Arduino website and double-click on “[arduino-1.8.19-windows.exe](#)” to install it.
 - i. The newer versions of the Arduino IDE may also work, but we have not had a chance to test them yet
 - b. Open the Arduino IDE and go to **File > Preferences**. Copy the following address into the “**Additional Boards Manager URLs**” textbox. **Double-check for extra spaces in the URL:**

`https://raw.githubusercontent.com/ROBOTIS-GIT/OpenRB-150/master/package\_openrb\_index.json`
 - c. Install the following board managers by going to **Tools > Board > Boards Manager**.
 - i. Search for “**Arduino SAMD**” and install the latest version.
 - ii. Search for “**OpenRB**” and install the latest version.
 - d. Install the following libraries by going to **Sketch > Include Library > Manage Libraries**.
 - i. Search for “**Dynamixel2Arduino**” and install the latest version.
 - ii. Search for “**DynamixelShield**” and install the latest version.
6. Power on the OpenRB-150 by connecting the 5V/4A power supply (plugged into an outlet) to the DC Barrel Jack. The green LED will illuminate (board power), followed shortly by the red LED (Dynamixel bus initialized and powered).
7. Connect the OpenRB-150 to the computer using the USB-C to USB-A cable.

8. Upload the “**usb_to_dynamixel**” sketch in the Arduino IDE in order to configure and manage the motors with DYNAMIXEL Wizard 2.0.
 - a. Go to **Tools > Board > OpenRB-150** and select **OpenRB-150**.
 - b. Go to **Tools > Port** and select the option that has “(OpenRB-150)” next to it.
 - c. Go to **File > Examples > OpenRB-150** and select the “**usb_to_dynamixel**” sketch. A new window should open in the Arduino IDE that is titled “**usb_to_dynamixel**”.
 - d. Click “**upload**” in the “**usb_to_dynamixel**” sketch to upload the sketch to the OpenRB-150.
 - e. Close the Arduino IDE.
9. If it is not already installed on your computer, install the **Visual C++ 2015-2019 Distributable (x86)** from the following link:
 - a. https://www.dropbox.com/s/3e5horhq73hm677/VC_redist.x86.exe?dl=0
10. Download and install the latest version of the “**Dynamixel Wizard 2.0**” Software:
 - a. http://en.robotis.com/service/downloadpage.php?ca_id=10
11. Once installed, open **Dynamixel Wizard 2.0** and click “**Options**”. Select the following:
 - a. Select the port to scan: Select the one labelled “**USB Serial Device**” (should be the same as Step 8b).
 - b. Select the baudrate to scan: **57600 bps** and **1000000 bps**. These are the most common baud rates that the servos are set to by default.
 - c. Click the “**OK**” button to save these options.
12. Servos must have unique IDs to operate on the Dynamixel bus. Since the default ID for all servos is ID=1, you must perform the following procedure once for **each servo** to change its ID. The settings you will change are defined below. **Read the next steps before proceeding.**

Setting	ID	Baud Rate	Return Delay Time	Drive Mode	PWM Limit	Max Pos Limit	Min Pos Limit
Address	7	8	9	10	36	48	52
Thumb Rotation	10	3 (1 Mbps)	10 (20 usec)	Bit 0	354	2896	2221
Thumb Flexion	11	3 (1 Mbps)	10 (20 usec)	Bit 0	354	3146	2048
Index Flexion	12	3 (1 Mbps)	10 (20 usec)	-	354	4095	2048
Middle Flexion	13	3 (1 Mbps)	10 (20 usec)	-	354	4095	2048
Ring Flexion	14	3 (1 Mbps)	10 (20 usec)	-	354	4095	2048
Little Flexion	15	3 (1 Mbps)	10 (20 usec)	Bit 0	354	3886	2048

13. Unplug the 5V/4A power supply from the barrel jack on the OpenRB-150.
14. Use the black, 3-wire cable to connect one servomotor to any of the four connection ports on the OpenRB-150.



15. Reconnect the power as described in Step 6.
16. Open Dynamixel Wizard 2.0 and click **“Scan”**. When your servo appears in the list, click **“Skip”**.
17. You must adjust parameters for each servo motor separately. We will configure the Thumb Rotation servo first as an example.
18. Change the ID to 10, baud rate to 3 (1 Mbps), and return delay time to 10 (20usec), Min Position Limit to 2221, and Max Position Limit to 2896, and PWM Limit to 354. Since this motor has reversed motion due to how it is physically installed in the hand, change the Drive Mode to Bit 0 (reverse mode) from unselected to selected. You will need to select the register to change the value and then click the ‘save’ button to make these changes.



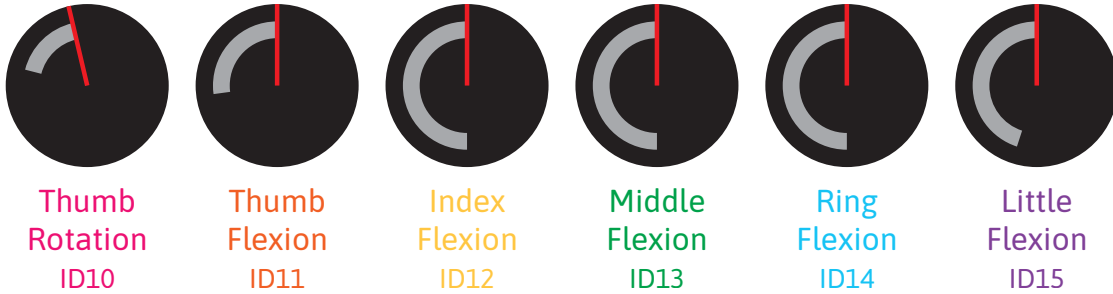
You must set the Min and Max Position Limits before connecting the servos to other software. Failing to do so can damage the servos, brackets, or cables if the motor moves beyond its safe range.



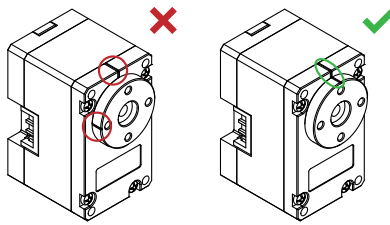
Be extremely careful not to set the Baud Rate higher than 1 Mbps. A higher setting will prevent the actuator from communicating with the OpenRB-150, requiring a separate U2D2 to reset the value.

19. You can now test the servo’s movement in the Dynamixel Wizard 2.0 software.
 - a. Set the **“Profile Velocity”** register (Address 112) to a reasonably low testing speed, such as 40 (9 RPM). You must set the moving speed first! The servo defaults to maximum speed, which increases the chance of causing damage if the settings are incorrect.
 - b. Set the **“Torque Enable”** register (Address 64) to 1 to turn on power to the motor.
 - c. After enabling torque, click the **“Goal Position”** register (Address 116) to change the position using the built-in rotational control.

20. Set the Goal Position to the Min Position Limit value. This aligns the servo horn correctly for assembly. Refer to the diagram to ensure your motor is reset to the position indicated by the red line:



All motors should have their default servo horns as follows (Motor ID10 will be slightly off centered):



21. Click **“Disconnect”** in Dynamixel Wizard 2.0. Repeat Steps 11-17 for the next servo, following the parameters in the settings table, until all servos are complete.

6. Before You Assemble

Please refer to the instructions and pay attention to the parts' orientation/direction during assembly. Some parts may be difficult to distinguish from one another. Check the part names and follow the assembly order. [Appendix A](#) has a printable 1:1 template to identify each part by assembly order.

6.1. Symbols for Assembly

The assembly instructions use several important conventions. The two primary color codes differentiate between 3D-printed parts and ordered parts. If multiple of these are required, a multiplier symbol will appear beside it. If the assembly section requires a previously built component, it will refer to the previous step number.

3D-Printed Part	Ordered Part
3D-Printed Part x2	1 Built Step



Pay attention to the length of screw used in assembly. Use of the wrong length of screw can lead to irreversible damage to the servo motors! The three lengths of Dynamixel tapping screws have been differentiated with colours. We recommend sorting the screws before assembly.

M2x4 TAP Screw	M2x6 TAP Screw	M2x8 TAP Screw
----------------	----------------	----------------

The labeled motors that were prepared in the last section will also be colourcoded. Please use the correct motor during the assembly process.

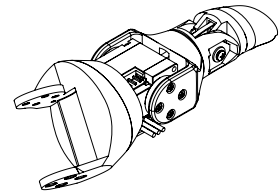
ID10	ID11	ID12	ID13	ID14	ID15
------	------	------	------	------	------

Other symbols you will notice during assembly are as follows:

Notice! Pay attention to the detail

 Lightly sand and spread superglue on this surface	 Rotate the entire part/ assembly
 Turn or twist highlighted section	 Build specified number of parts

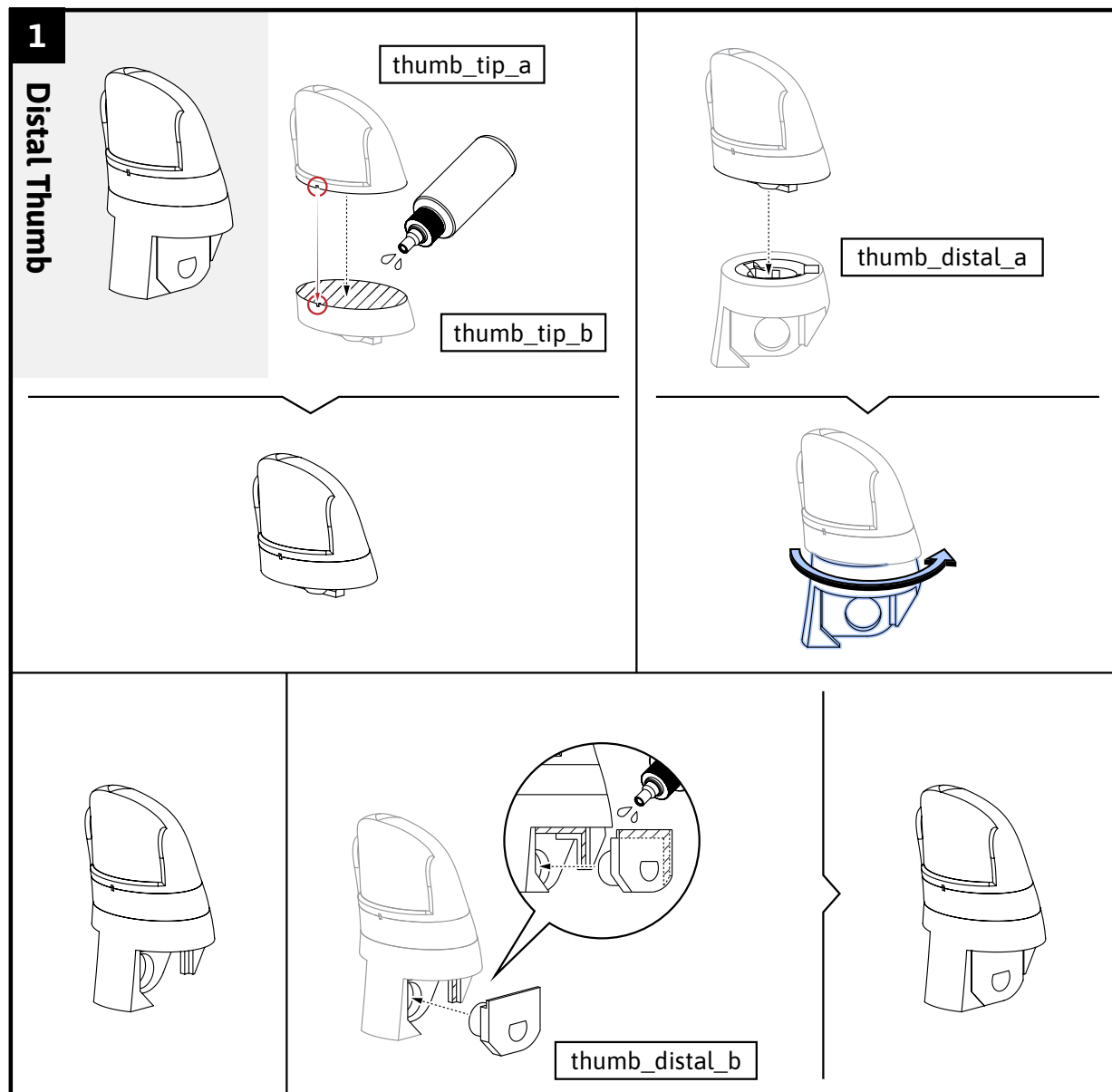
7. Thumb Assembly



7.1. Materials Needed

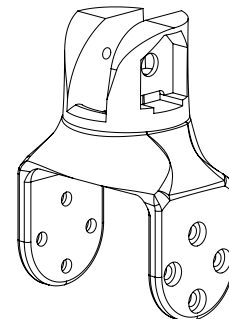
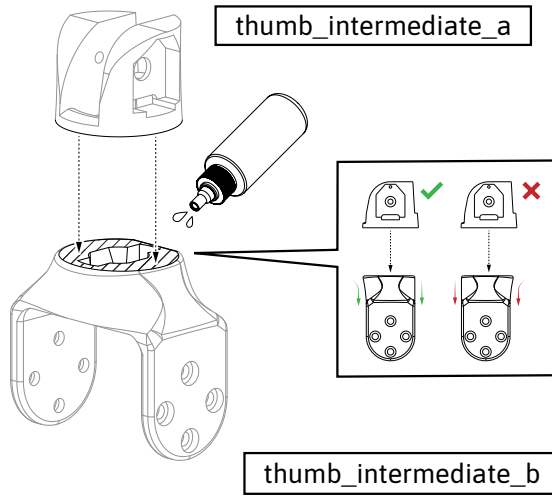
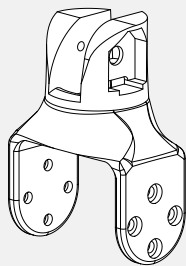
For assistance in identifying each 3D-printed part and mechanical part, refer to the 3D-Printed Parts List and Thumb Assembly Template (1:1) print-out in Appendix A.

7.2. Instructions



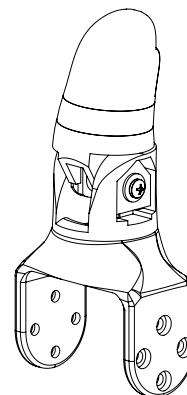
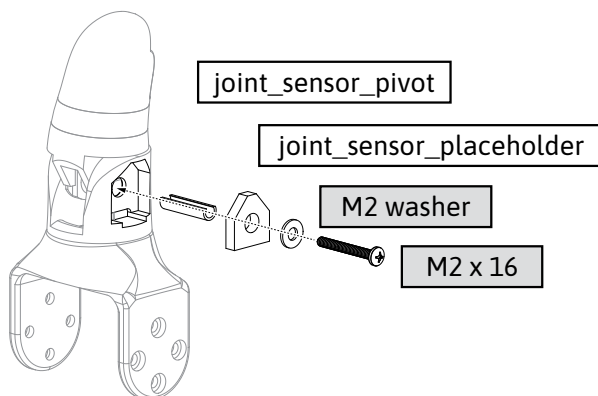
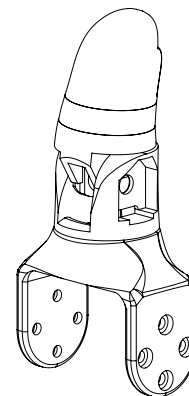
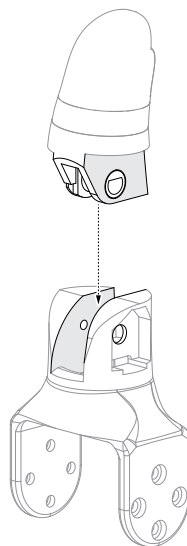
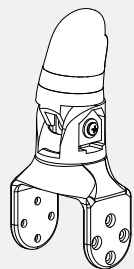
2

Intermediate Thumb



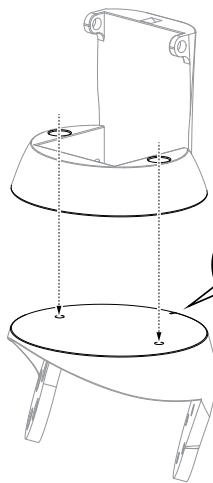
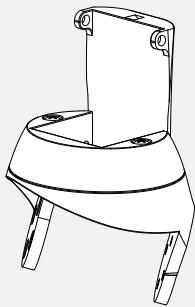
3

Joint Sensor



4

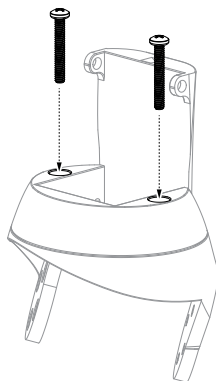
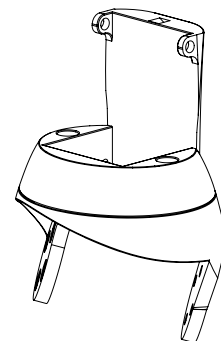
Proximal Thumb



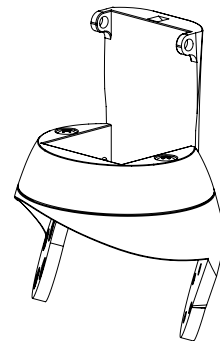
thumb_proximal_a

Notice the alignment!

thumb_proximal_b

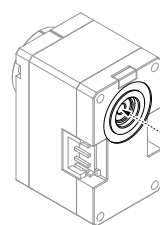
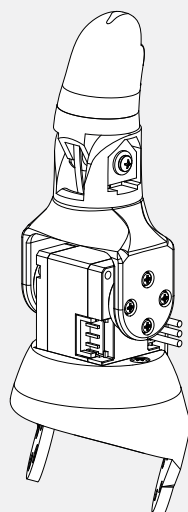


M2 x 14 x2



5

Flexion Motor

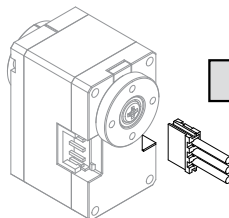
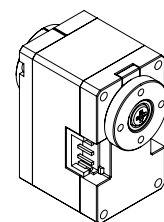


ID11

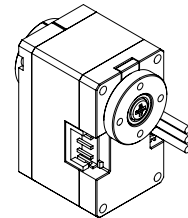
Idler Horn

Idler Cap

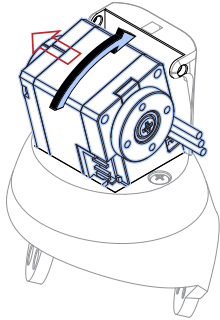
M2.6 x 6 TAP



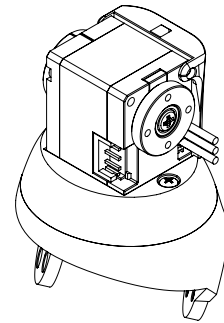
TTL 3P Cable (JST-JST)



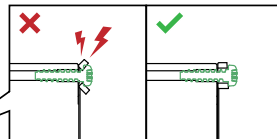
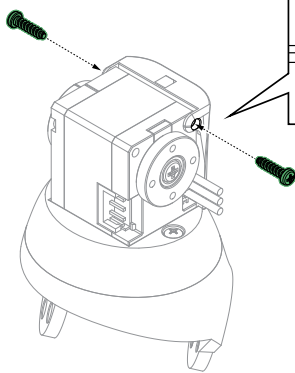
Notice
orientation
of motor!



4 Proximal Thumb

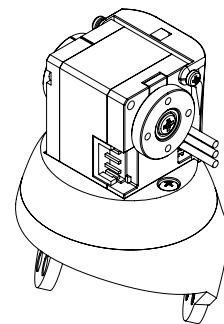


Avoid overtightening!

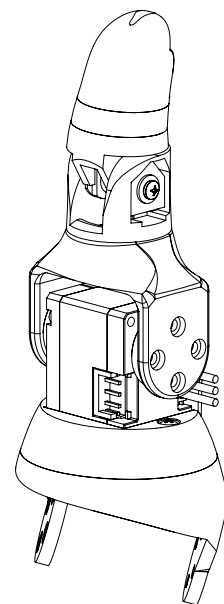
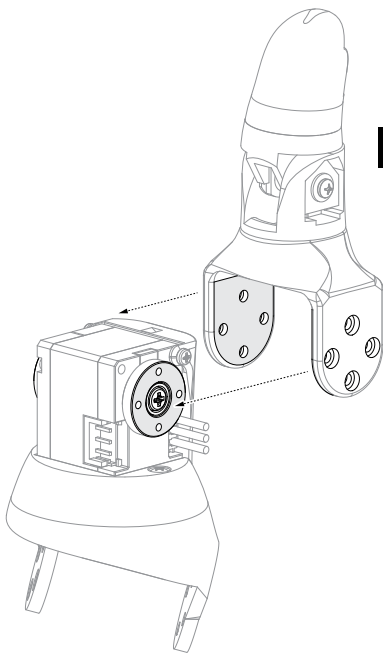


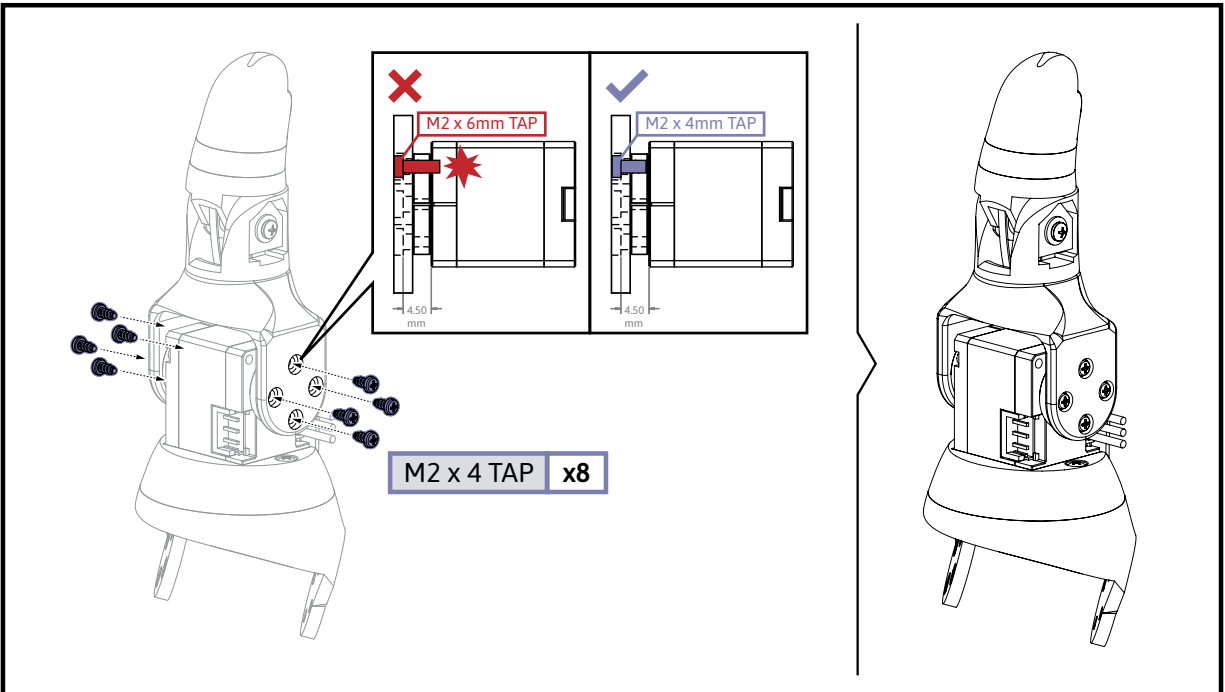
M2 x 8 TAP x2

Notice screw
length!



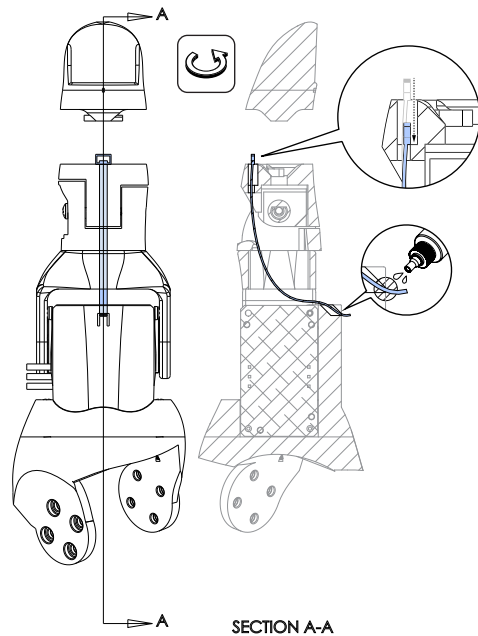
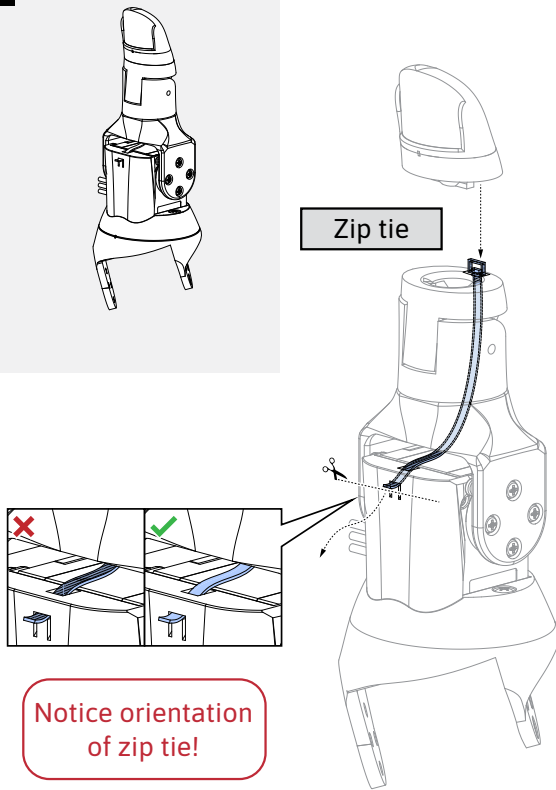
3 Joint Sensor





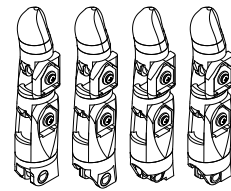
6

Zip tie Installation



A Thumb

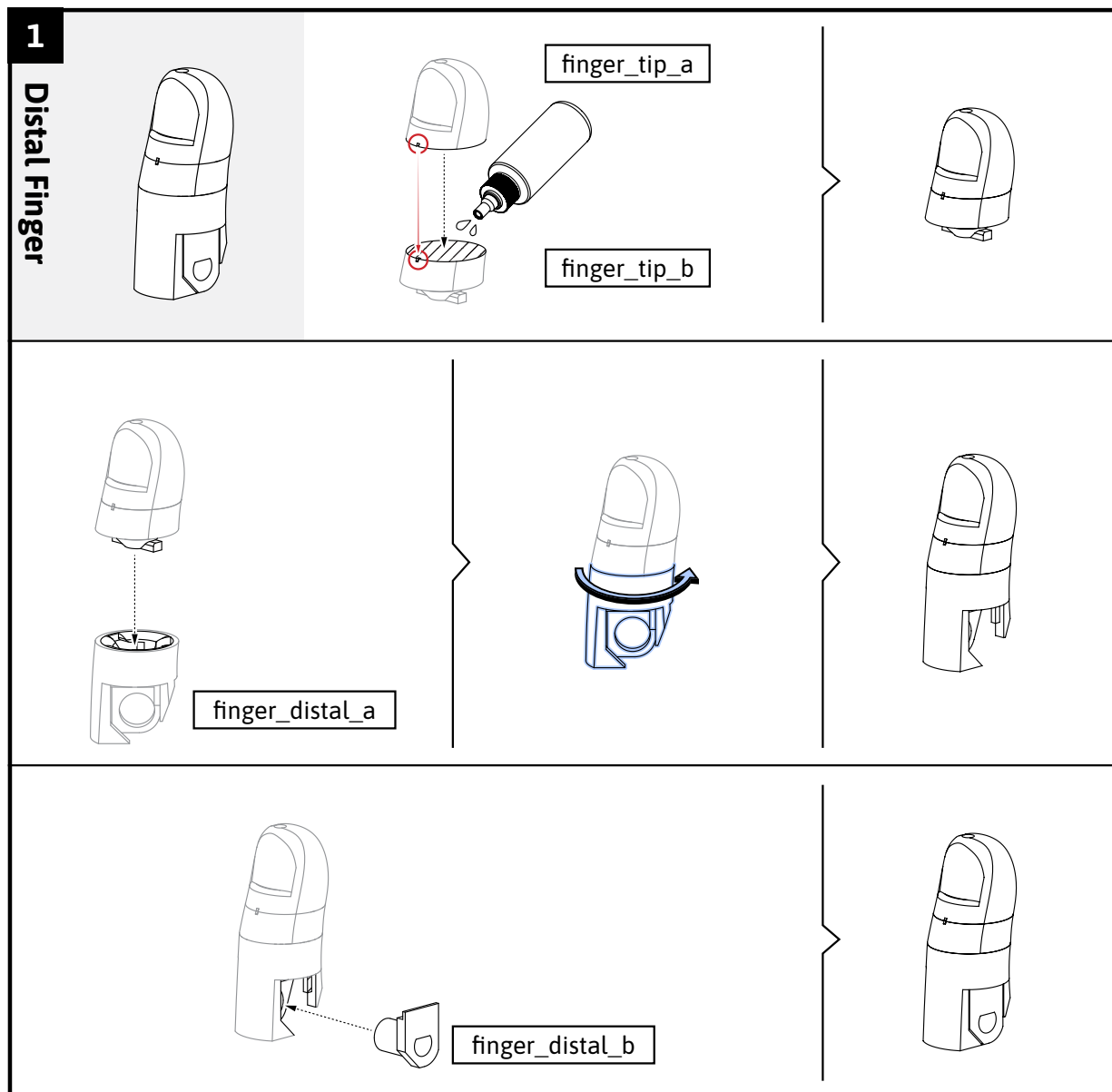
8. Finger Assembly



8.1. Materials Needed

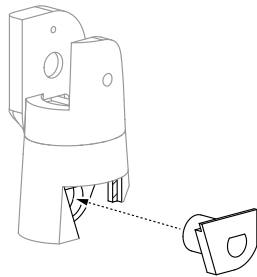
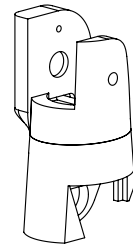
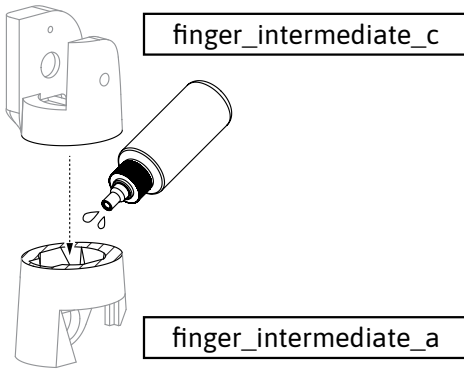
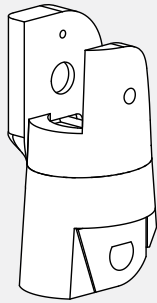
For assistance in identifying each 3D-printed part and mechanical part, refer to the 3D-Printed Parts List and Finger Assembly Template (1:1) print-out in Appendix A.

8.2. Instructions

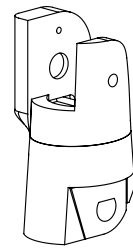


2

Intermediate Finger

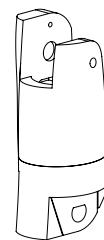
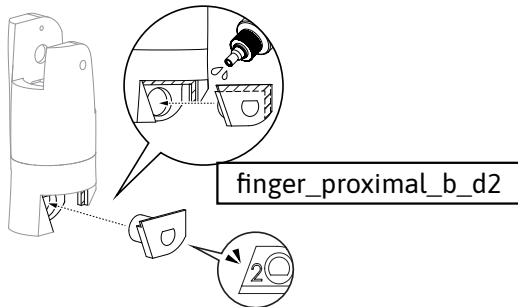
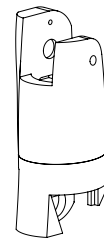
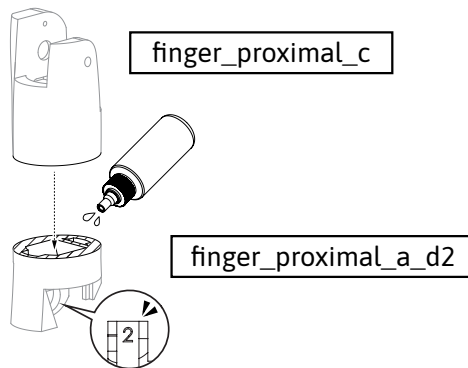
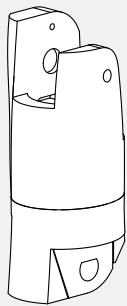


finger_intermediate_b



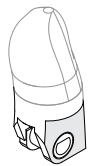
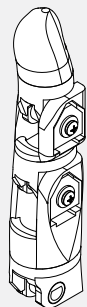
3

Proximal Finger

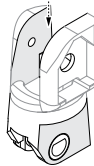


4

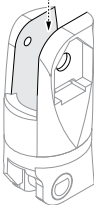
Joint Sensors



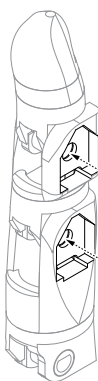
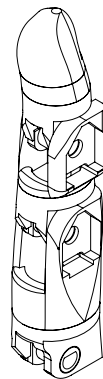
1 Distal Finger



2 Intermediate Finger



3 Proximal Finger

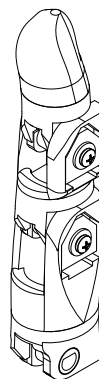


joint_sensor_pivot x2

joint_sensor_placeholder x2

M2 Washer x2

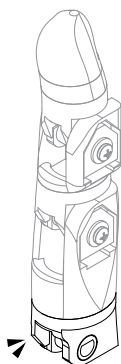
M2 x 16 x2



B D2

5

x4



finger_proximal_a_dX

finger_proximal_b_dX



B D3

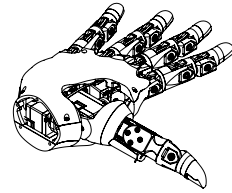


B D4



B D5

9. Palm Assembly

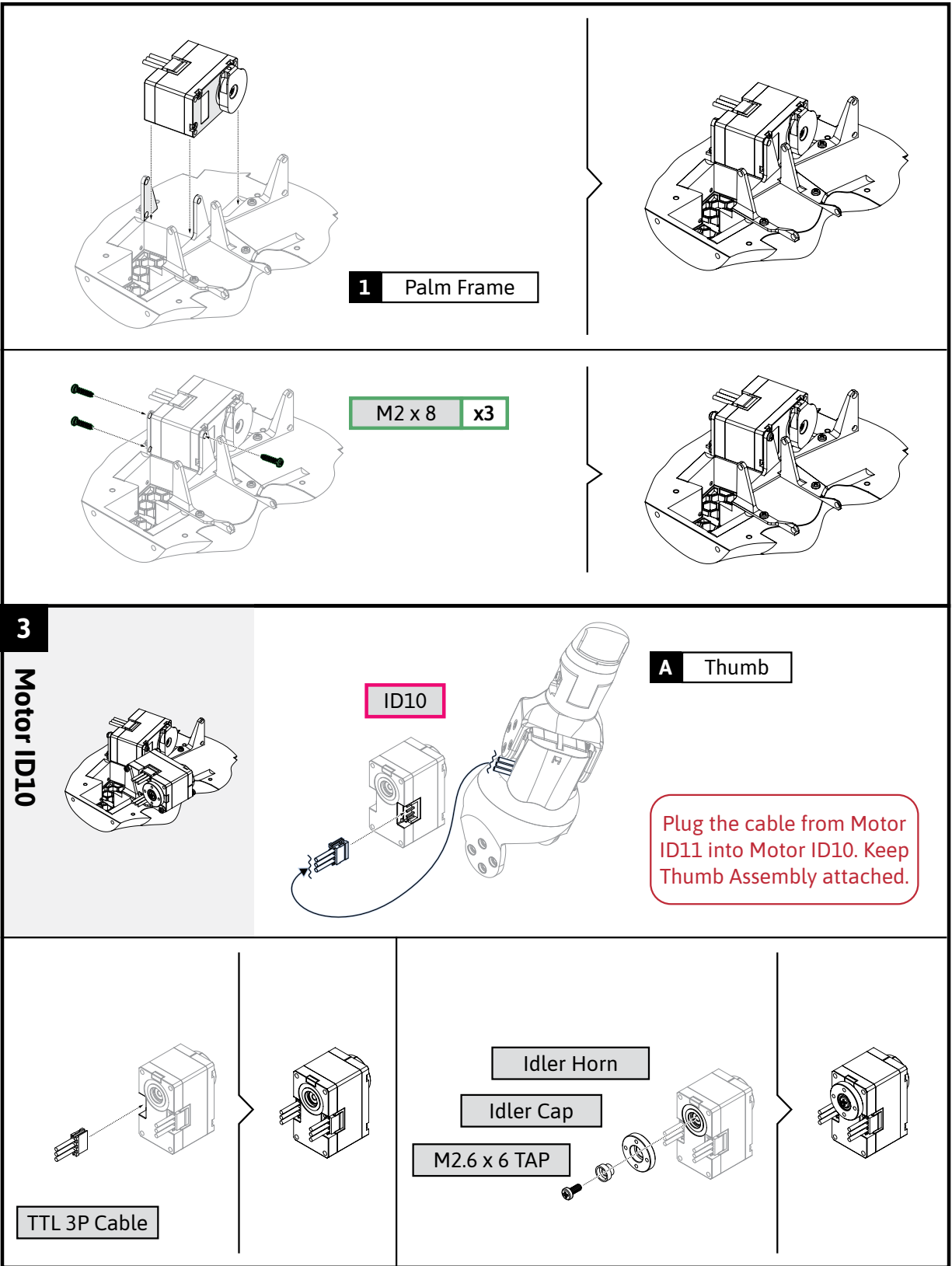


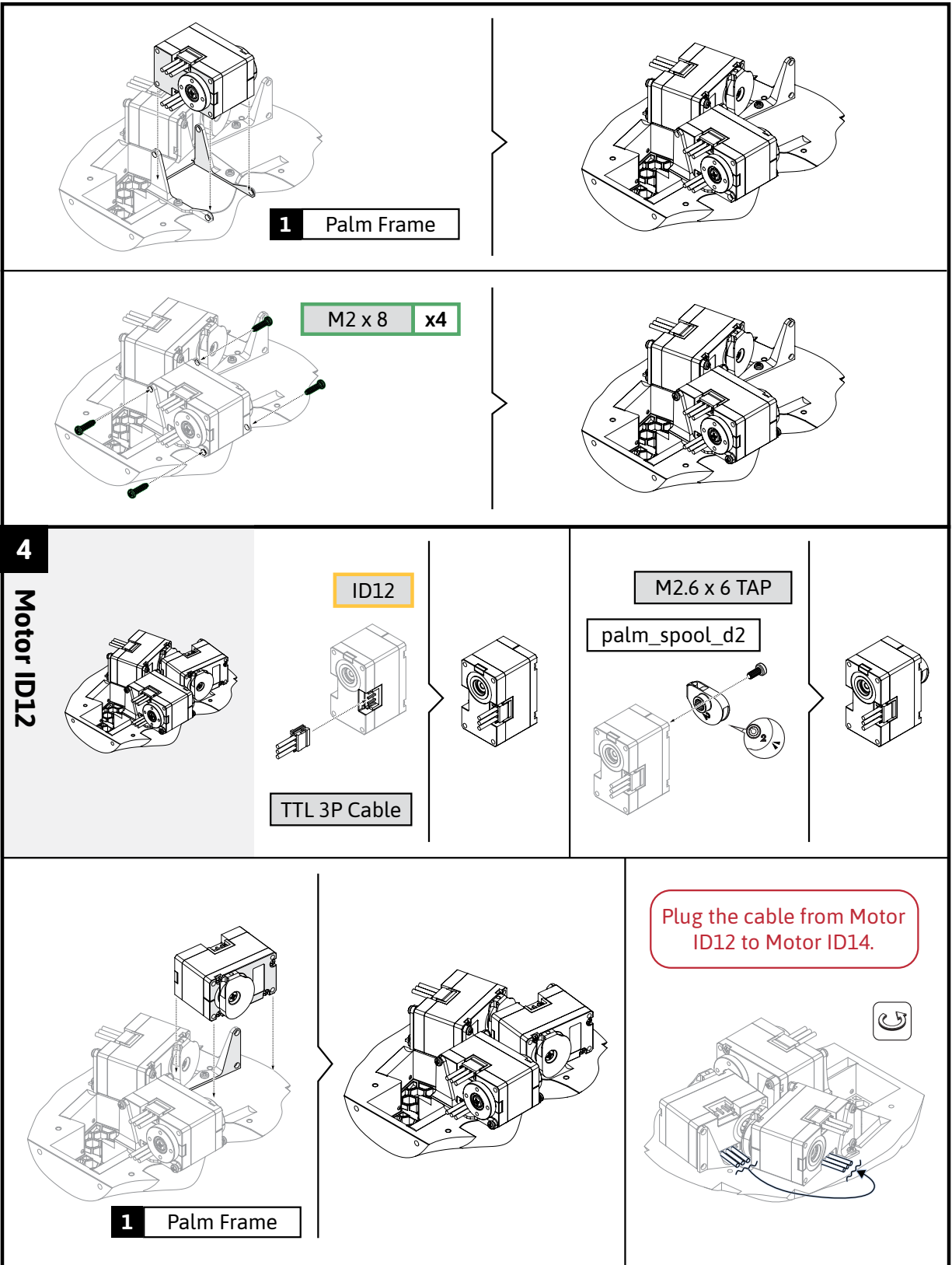
9.1. Materials Needed

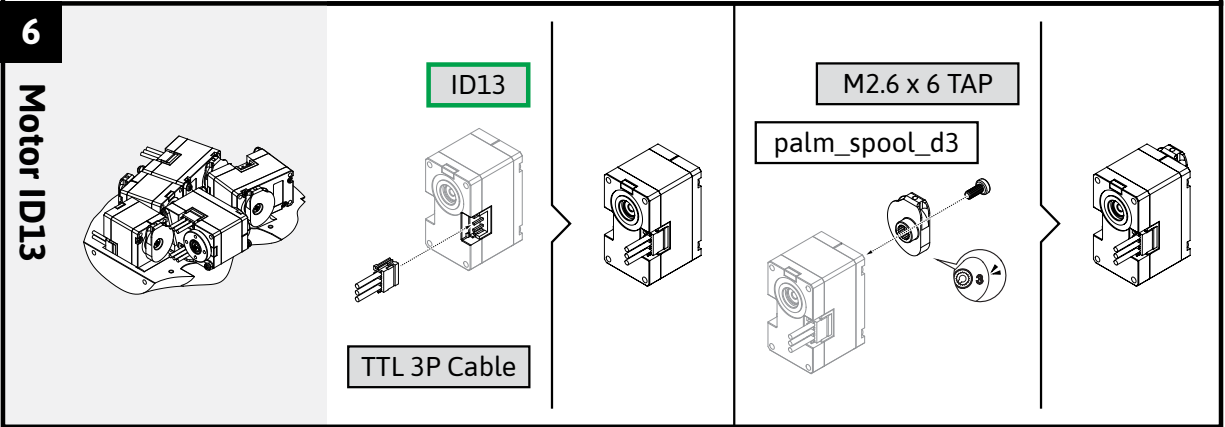
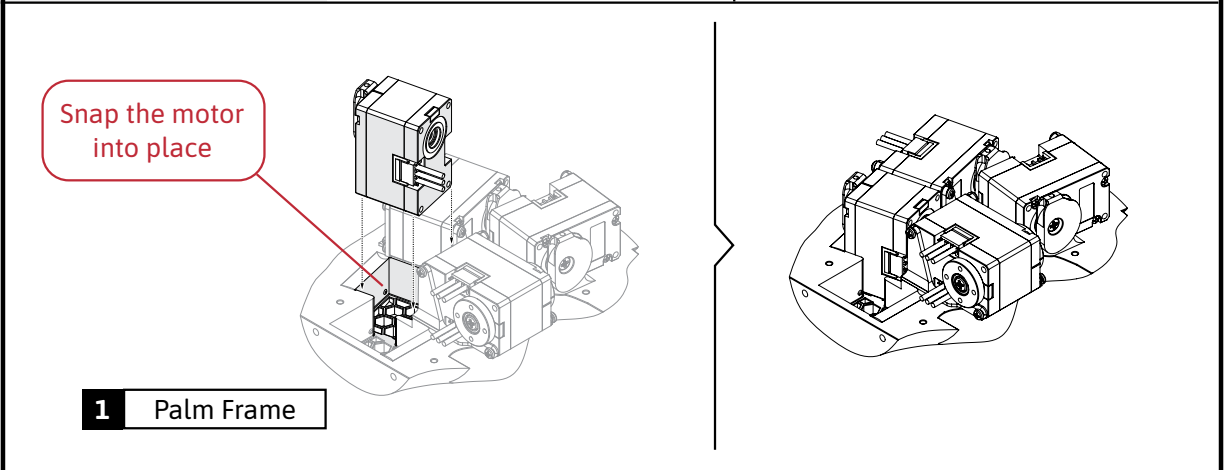
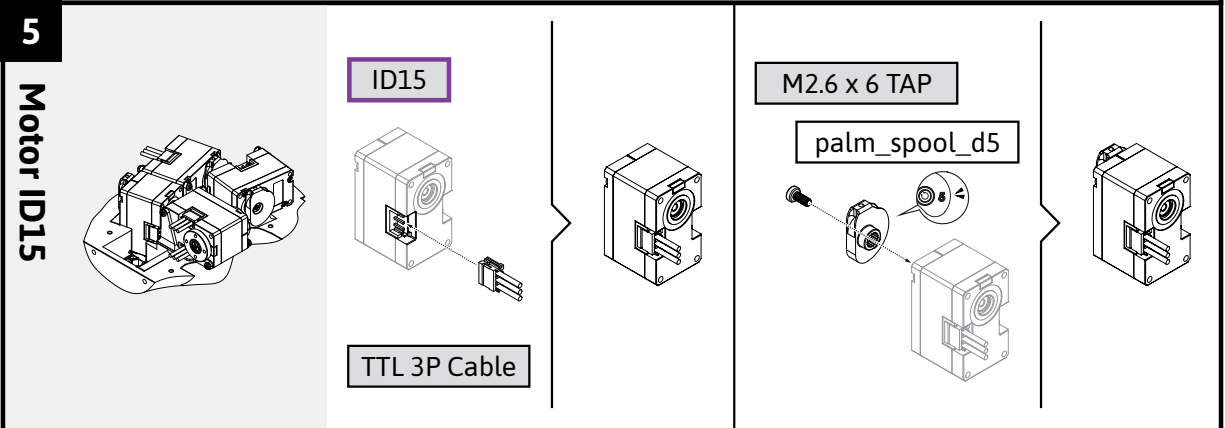
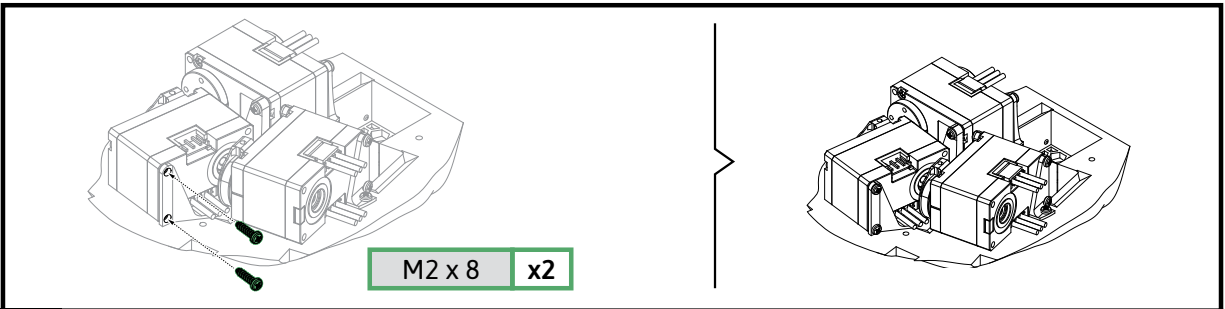
For assistance in identifying each 3D-printed part and mechanical part, refer to the 3D-Printed Parts List and Palm Assembly Template (1:1) print-out in Appendix A.

9.2. Instructions

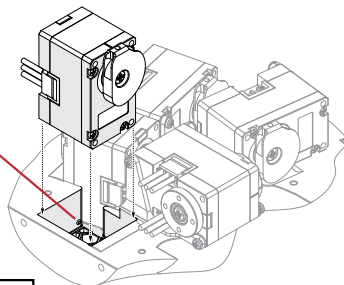
1 Palm Frame			
2 Motor ID14			



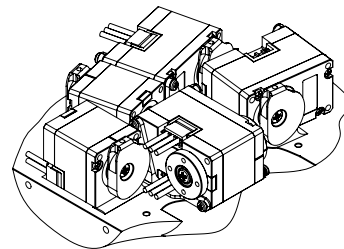




Snap the motor into place

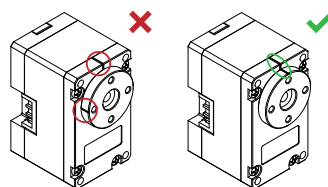
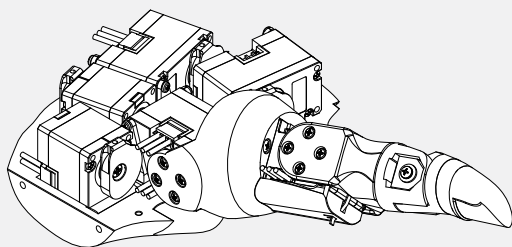


1 Palm Frame

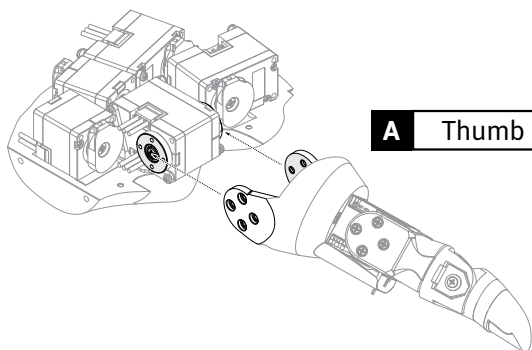


7

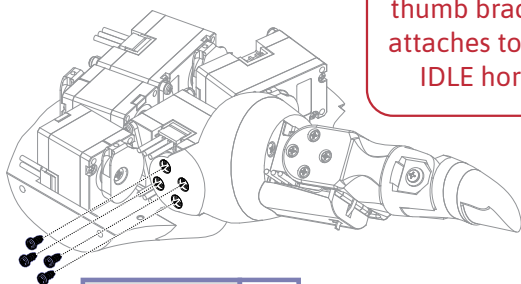
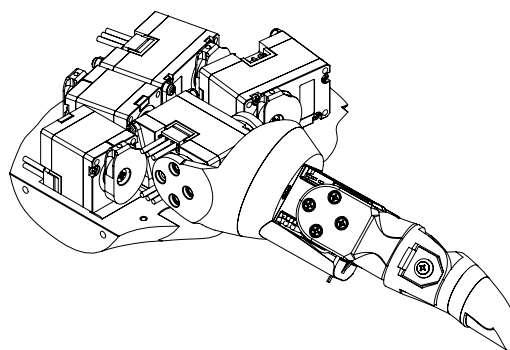
Installing Thumb



Ensure Motor ID10 does not get twisted while installing the thumb.

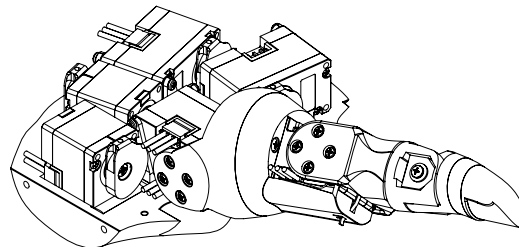


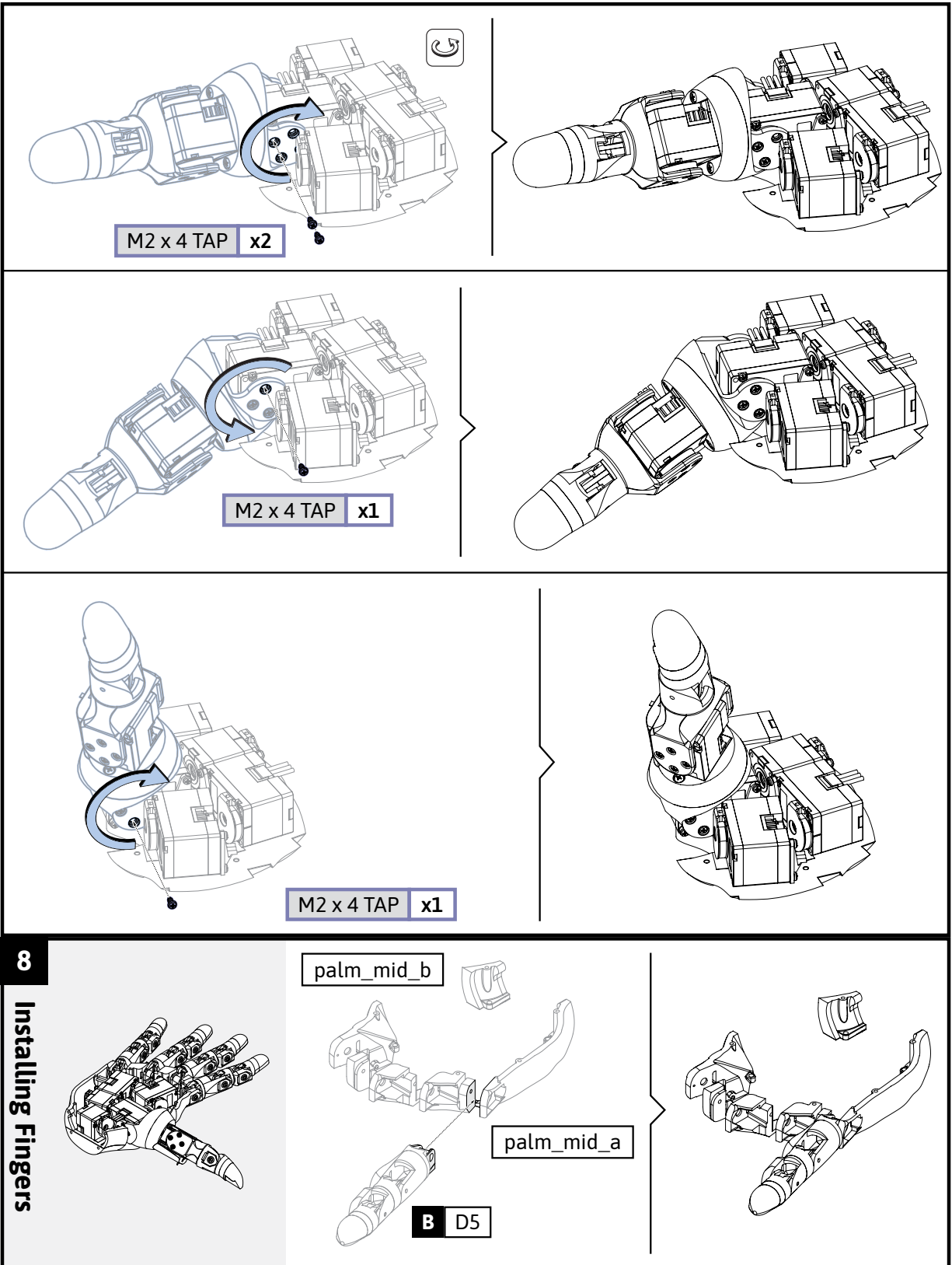
A Thumb

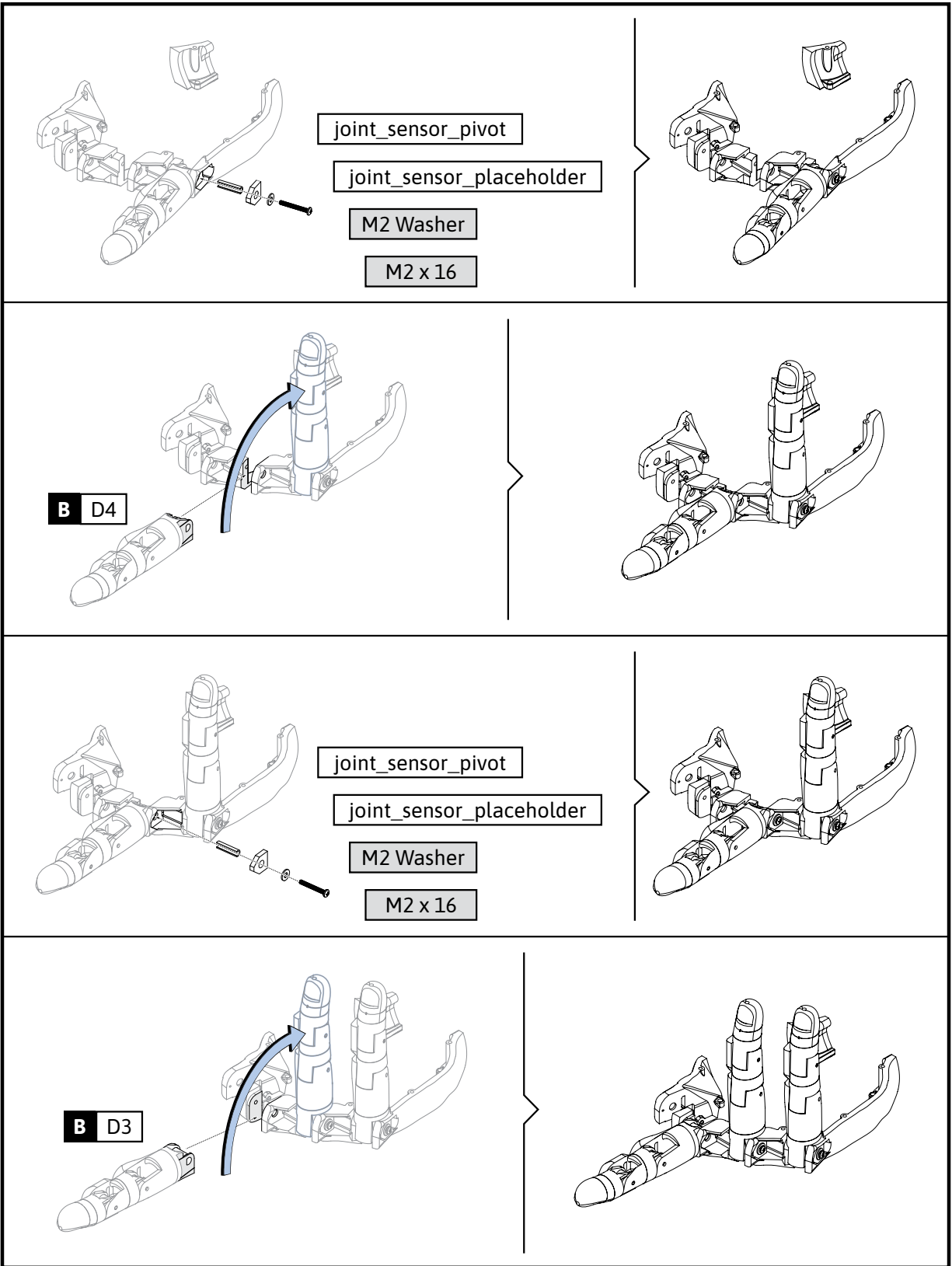


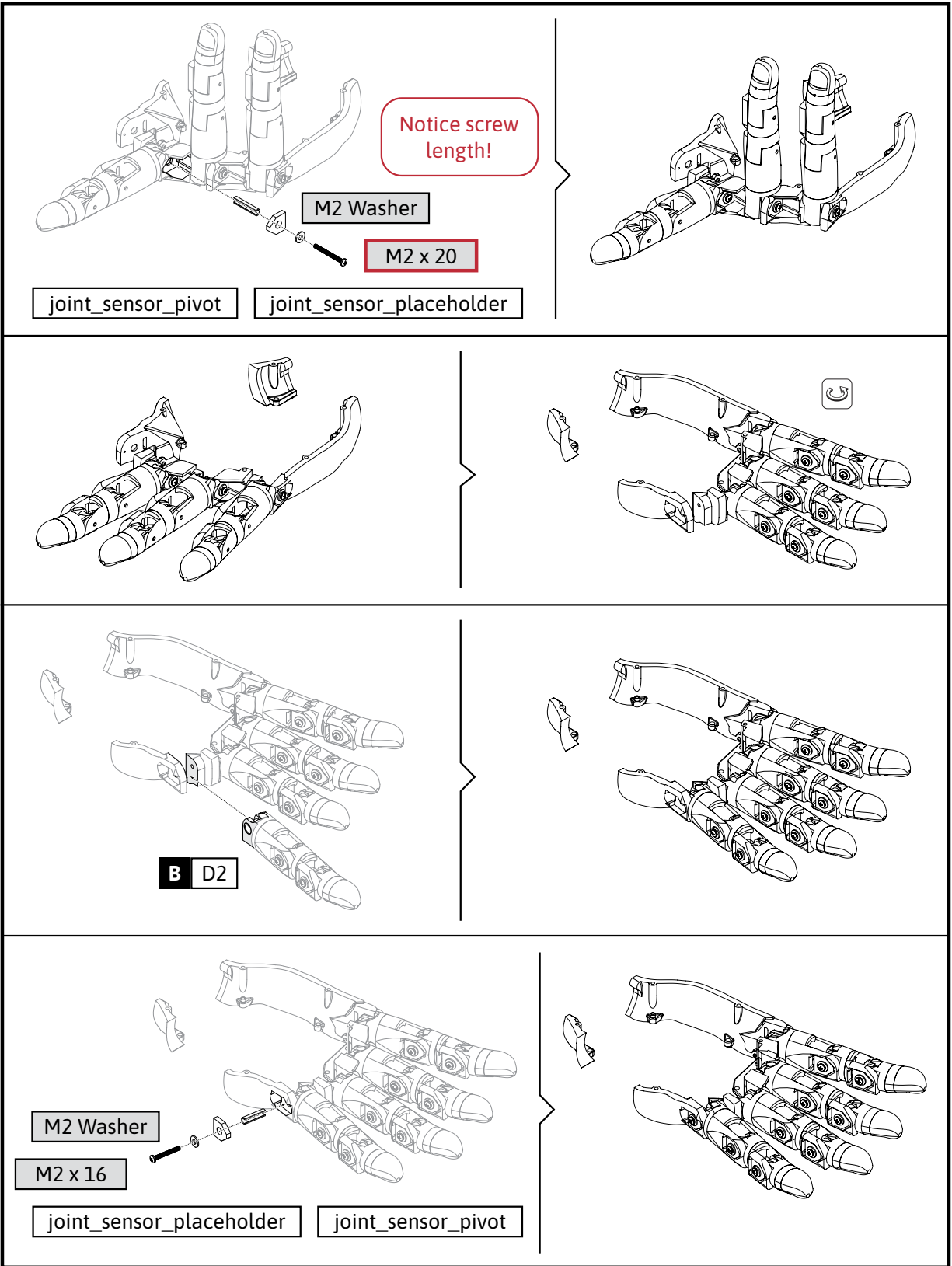
This side of the thumb bracket attaches to the IDLE horn

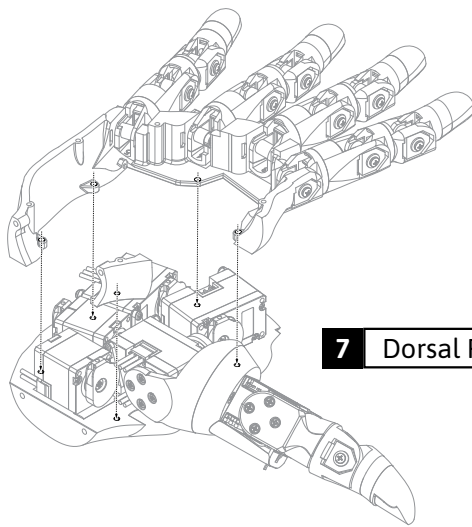
M2 x 4 TAP x8



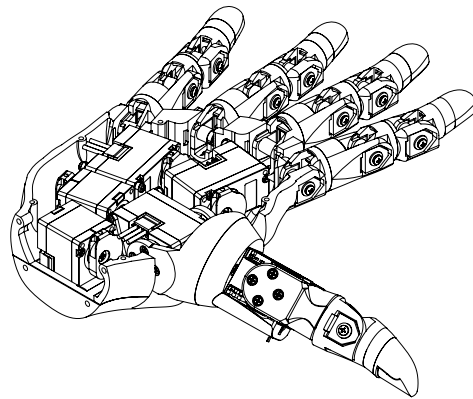




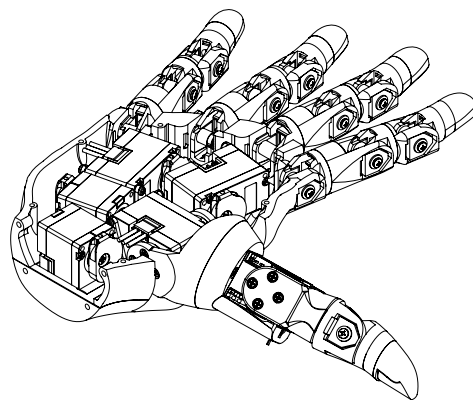
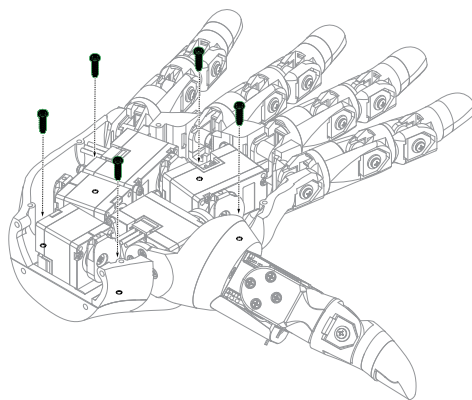




7 Dorsal Palm

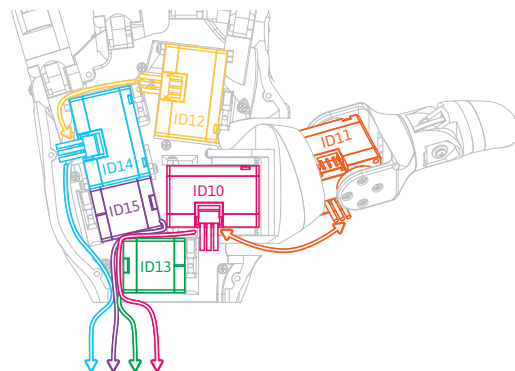


M2 x 8 TAP x5



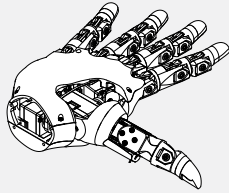
Position the motor wires in their respective position according to the diagram. Label the wires either with a label maker or by writing on the white connectors in permanent marker.

TIP: Use a thin blunt object to fit wires such as the ones coming from motors ID10 and ID15 in place.

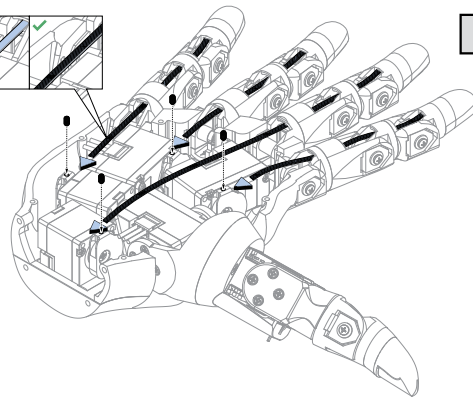
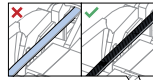


9

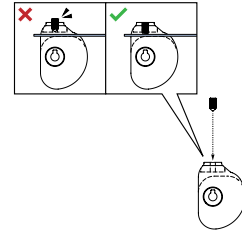
Installing Zipties



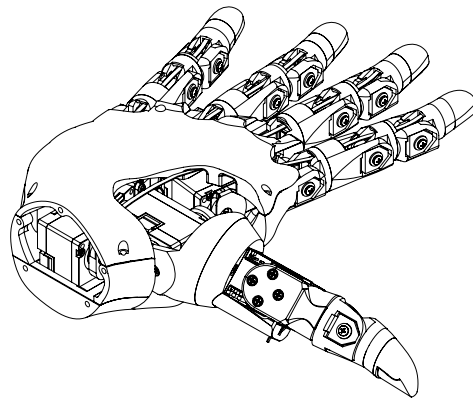
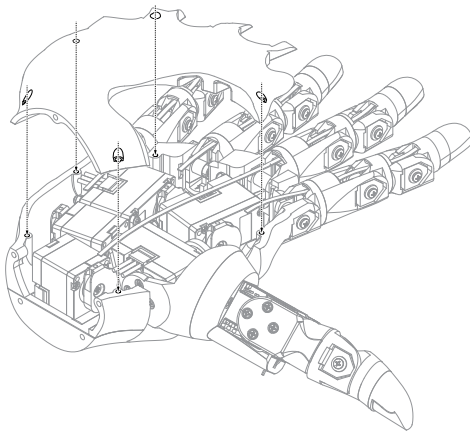
Zip Tie x4



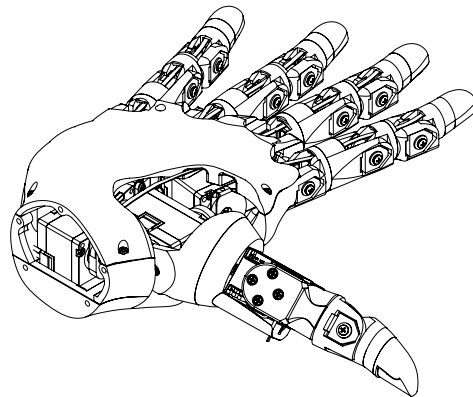
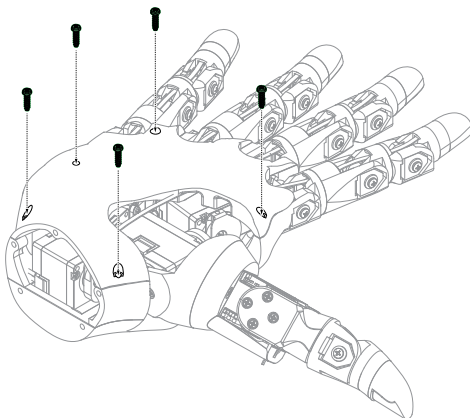
Cone-point Set Screw x4



palm_ventral

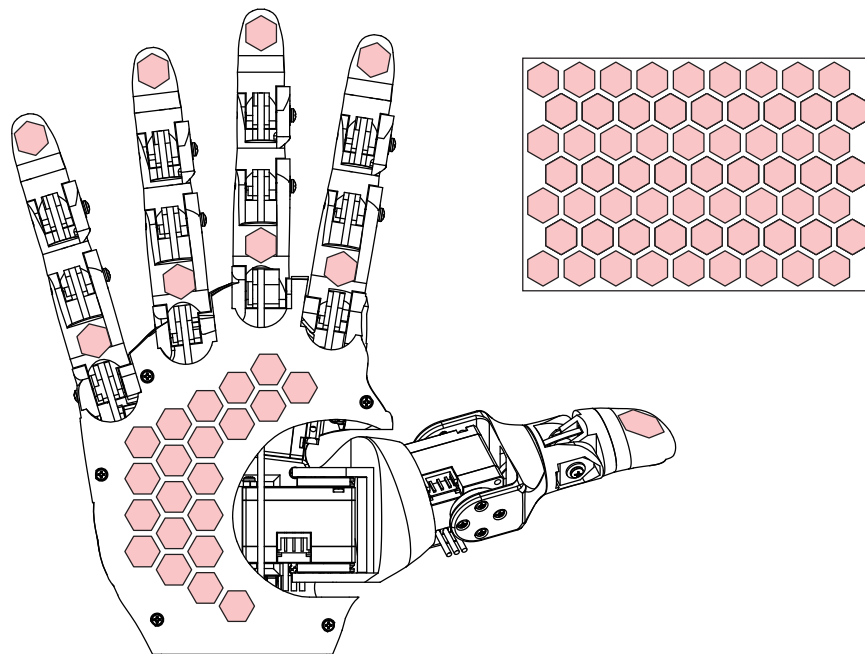
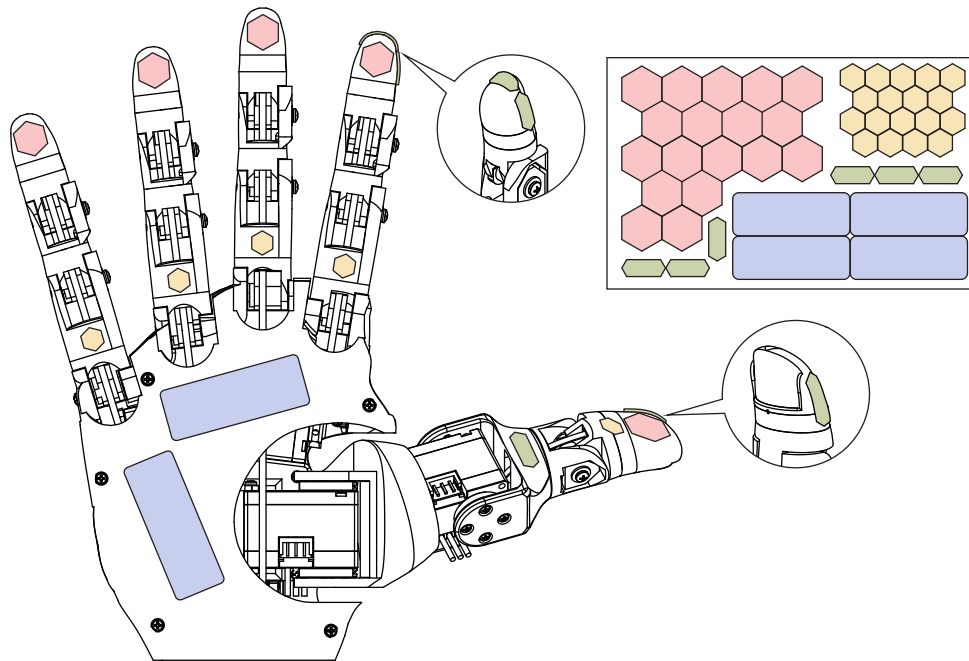


M2 x 8 TAP x5



Palm Grips

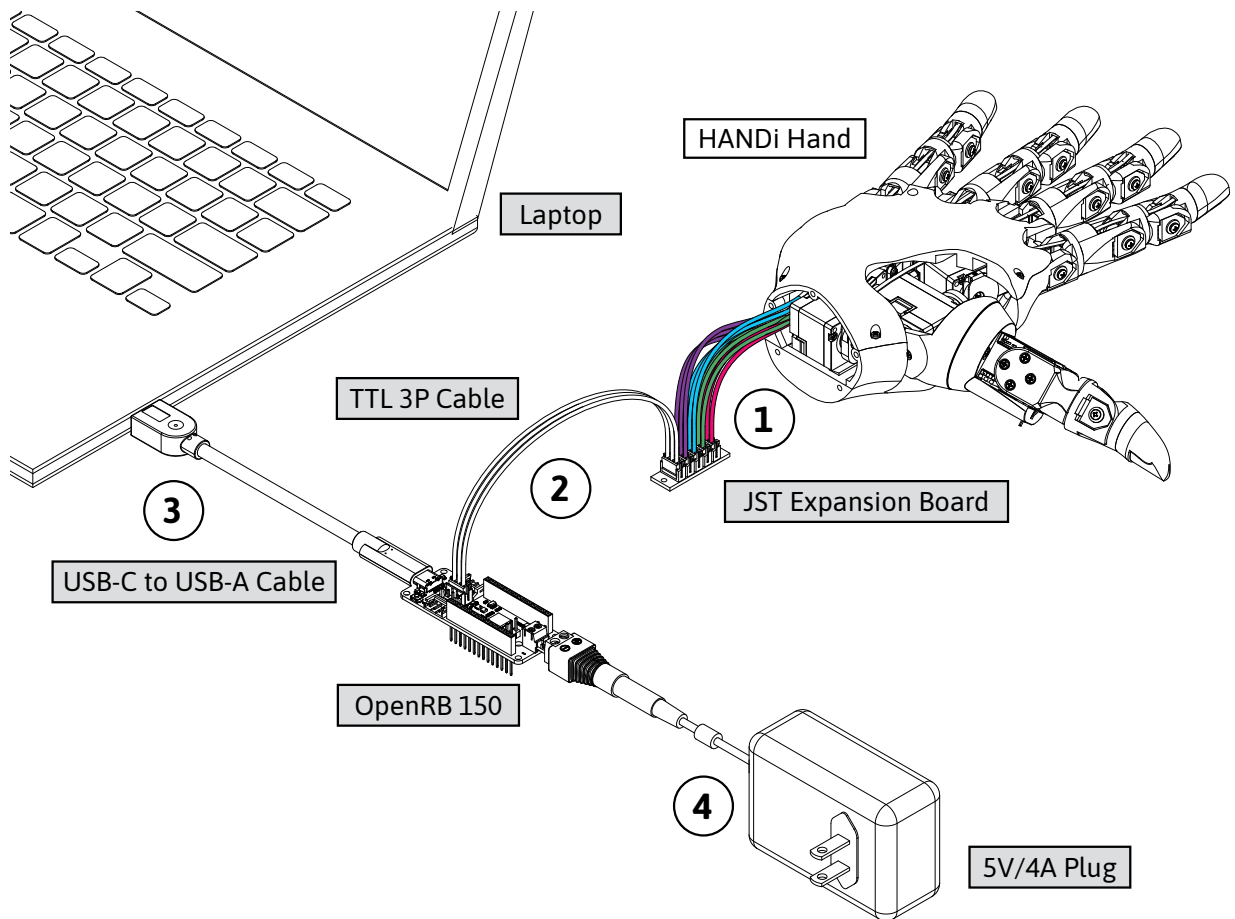
The adhesive grip tape can be placed on various sections of the HANDi Hand to optimize its grasping capabilities. While makers can have creative liberty to place the grip tape where they please, below are two diagrams displaying the suggested grip tape placement depending on what shape of grip tape was purchased.



10. After You Assemble

10.1. Wiring

Follow the wiring diagram below to begin using the HANDi Hand. There is an important order of connections. **If not done in the right order, it can damage the electronic components.**



- 1 Plug all 4 TTL 3P Cables from the HANDi Hand into the JST Expansion Board or directly into the OpenRB 150.
- 2 If using a JST Expansion Board, use a TTL 3P Cable to connect the JST Expansion Board to the OpenRB 150.
- 3 Use a USB-C to USB-A Cable to connect the OpenRB 150 to a laptop.
- 4 Use the 5V/4A Plug to connect the OpenRB 150 to a wall outlet. The LEDs on the motors should flash briefly once powered.

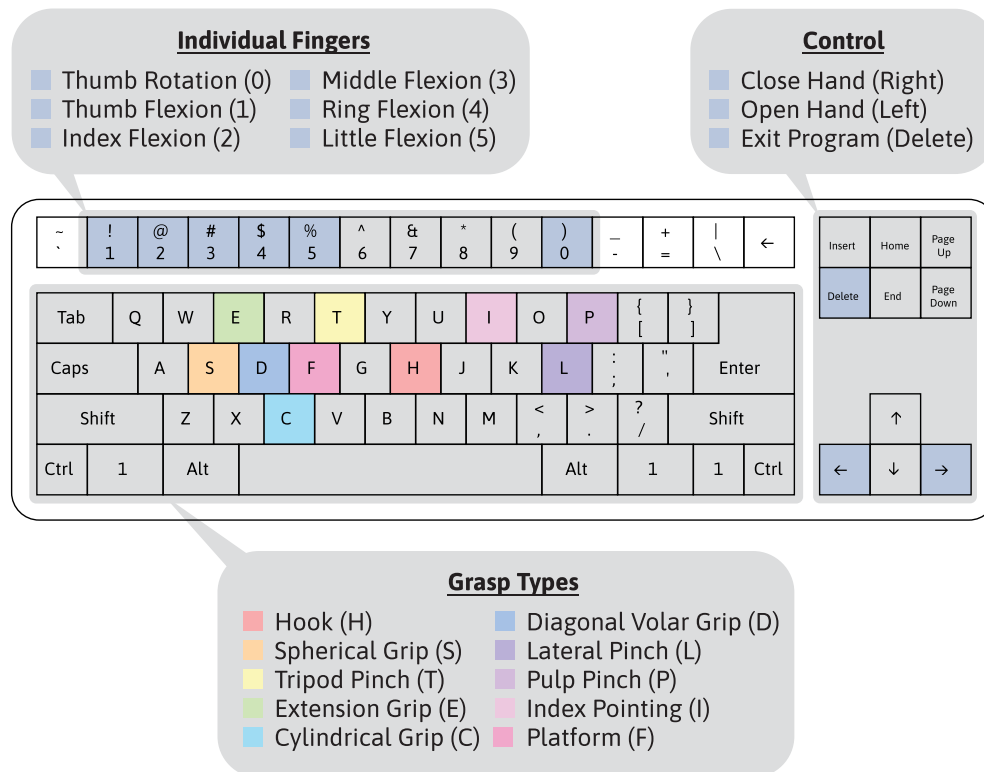
10.2. Software

10.2.1. Python

For basic control of the HANDi Hand, users may download and run the provided Python script provided via [GitHub](#). There is one script that can be run locally on a computer.

hh_keyboard_grasp.py: Uses the keyboard to switch between grasps and move the fingers.

1. Power on the OpenRB-150 by connecting the 5V/4A power supply (plugged into an outlet) to the DC Barrel Jack. The green LED will illuminate (board power), followed shortly by the red LED (Dynamixel bus initialized and powered).
2. Connect the OpenRB-150 to the computer using the USB-C to USB-A cable.
3. Check which COM port the HANDi Hand is on.
 - a. Click on the **Start** menu and type **Device Manager**
 - b. Expand **Ports (COM & LPT)**
 - c. Look for the HANDi Hand, followed by a COM port number in parentheses (e.g., USB Serial Device (COM3)).
4. Open **hh_keyboard_grasp.py** using your preferred method.
5. Change the **DEVICENAME = 'COMX'** to the COM port of the HANDi Hand on your device.
6. Run the script.
7. Control the hand using the following keys:



10.2.2. Arduino

For basic control of the HANDi Hand, users may use the provided Arduino scripts provided via [GitHub](#). There are two scripts that can be uploaded to your OpenRB-150. These can only be used with the additional joystick module (see next section).

hh_joystick_finger.ino: Switches through individual fingers using the joystick button.

hh_joystick_grasp.ino: Switches through grasp patterns using the joystick button.

1. If you have not already, install the Arduino IDE and set it up to work with the OpenRB-150a. If you have already done this, skip to Step 4.
 - a. Download the legacy version of the [Arduino IDE 1.8.19](#) from the Arduino website and double-click on “[arduino-1.8.19-windows.exe](#)” to install it.
 - i. The newer versions of the Arduino IDE may also work, but we have not had a chance to test them yet
 - b. Open the Arduino IDE and go to **File > Preferences**. Copy the following address into the “**Additional Boards Manager URLs**” textbox. **Double-check for extra spaces in the URL:**

`https://raw.githubusercontent.com/ROBOTIS-GIT/OpenRB-150/master/package\_openrb\_index.json`
 - c. Install the following board managers by going to **Tools > Board > Boards Manager**.
 - i. Search for “**Arduino SAMD**” and install the latest version.
 - ii. Search for “**OpenRB**” and install the latest version.
 - d. Install the following libraries by going to **Sketch > Include Library > Manage Libraries**.
 - i. Search for “**Dynamixel2Arduino**” and install the latest version.
 - ii. Search for “**DynamixelShield**” and install the latest version.
2. Power on the OpenRB-150 by connecting the 5V/4A power supply (plugged into an outlet) to the DC Barrel Jack. The green LED will illuminate (board power), followed shortly by the red LED (Dynamixel bus initialized and powered).
3. Connect the OpenRB-150 to the computer using the USB-C to USB-A cable.
4. Upload either the “**hh_joystick_finger.ino**” or “**hh_joystick_grasp.ino**” sketch in the Arduino IDE.
 - a. Go to **Tools > Board > OpenRB-150** and select **OpenRB-150**.
 - b. Go to **Tools > Port** and select the option that has “**(OpenRB-150)**” next to it.
 - c. Click “**upload**” to upload the sketch to the OpenRB-150.
 - d. Close the Arduino IDE.
5. Now there is a sketch uploaded to the OpenRB-150, you may unplug the USB-C to USB-A cable from both the OpenRB-150 and the computer. The hand will now function based on whichever sketch was uploaded to it as long as the 5V/4A power supply is connected.
6. If you want to change the uploaded sketch, follow Step 4 with the new sketch.

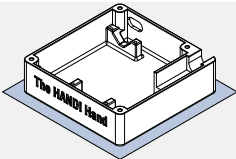
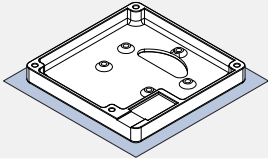
10.3. Optional Add-Ons

10.3.1. STANDi Stand

The STANDi Stand is a simple box that houses the electronics and holds the hand in an upright position. Parts for the STANDi Stand are designed to be printed in PLA. The 3D files including stl, 3mf, and gcode files for the stand can be accessed via [GitHub](#). The source SolidWorks parts and assembly files are also included for those who may want to customize the parts to their needs.

The suggested print specifications for each part are found in the table below. This table lists all the part sets that must be printed for a complete stand. If all parts are printed using Bambu Lab's PLA Basic, it should cost about \$2 CAD to print. If none of the parts are being altered or customized, use the gcode files to quickly get started.

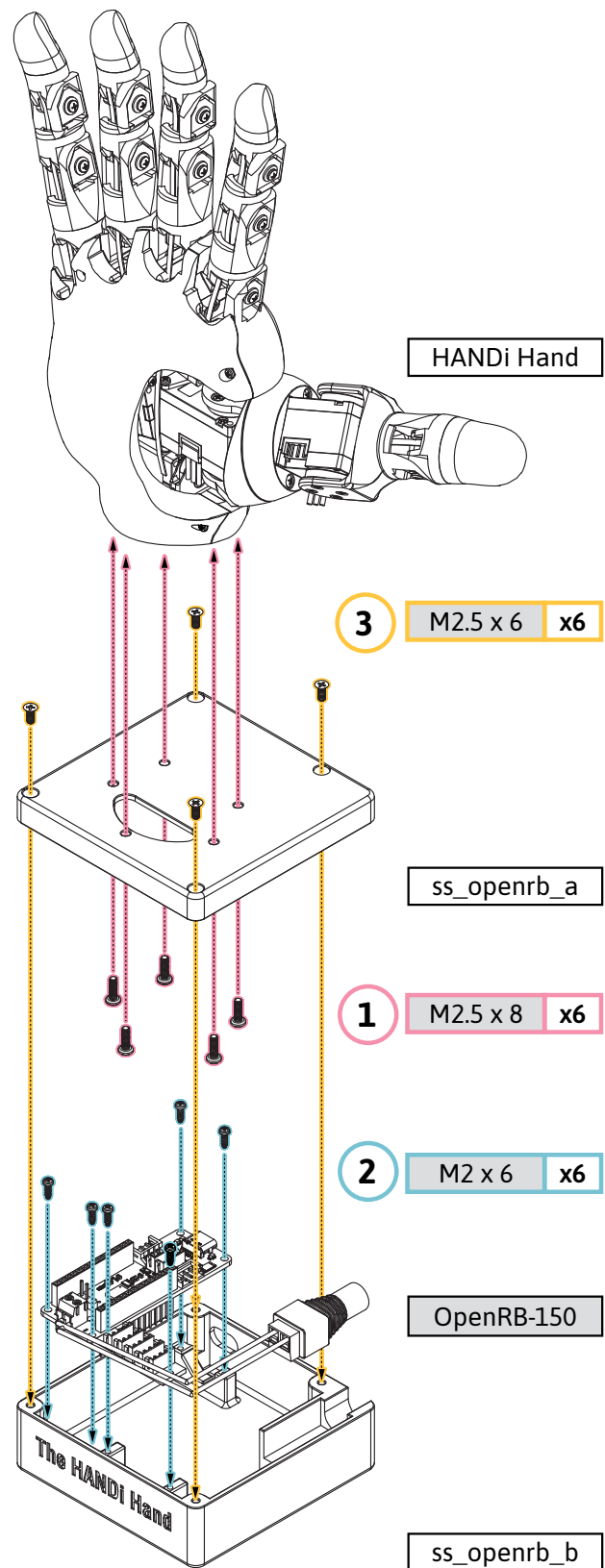
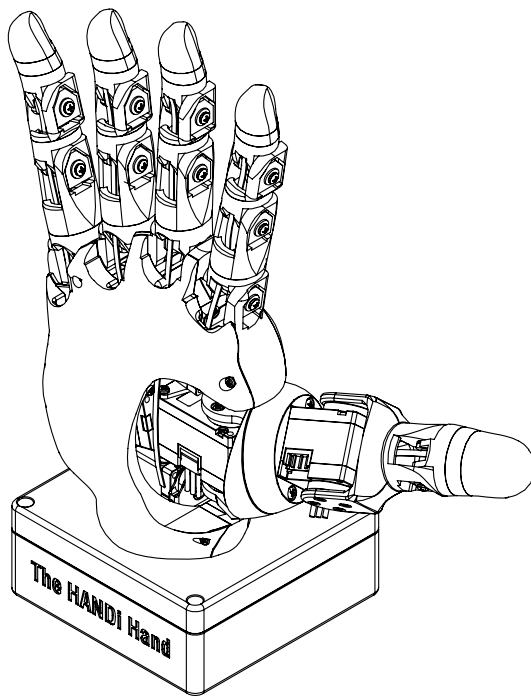
Print Preset	Layer Height (mm)	Wall Loops	Sparse Infill Pattern	Sparse Infill Density (%)	Supports	Brim
SS Default	0.2	2	Rectilinear	20	None	None
SS Brim	0.2	2	Rectilinear	20	None	Outer 5mm

Set Name	Parts in Set	Build Orientation	Colour	Print Specs	Time (min)	Material (g)
ss_openrb_a	ss_openrb_a		A	SS Default	68	40.31
ss_openrb_b	ss_openrb_b		A	SS Default	43	24.05
Total:					1 h 51 m	64.36 g

The [link here](#) leads to the public Google Sheets file that contains all the additional items (excluding tools) required for building the STANDi Stand. Makers may create a copy of this spreadsheet for their own convenience. Select "STANDi Stand" using the kit dropdown menu.

Follow the diagram below to assemble the STANDi Stand:

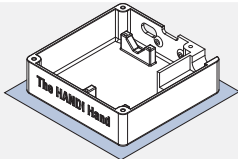
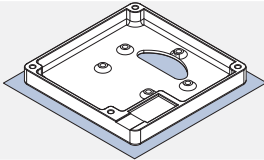
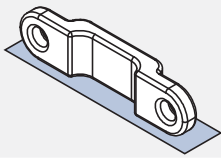
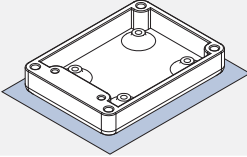
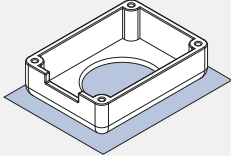
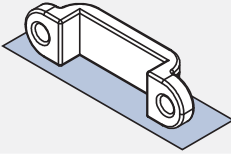
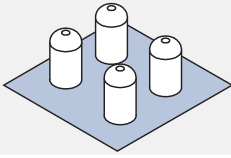
- 1 Attach the HANDi Hand to the lid of the STANDi Stand. Thread the motor wires through the semi-circular hole in the lid.
- 2 Secure the OpenRB-150 and the optional JST Expansion Board to the bottom of the STANDi Stand.
- 3 Close the STANDi Stand and maneuver the wires neatly in place to prevent them from being caught under the lid.



10.3.2. Joystick Module

An add-on to the STANDi Stand is the joystick module. If you have opted to use Arduino for controlling the hand, the joystick module is required.

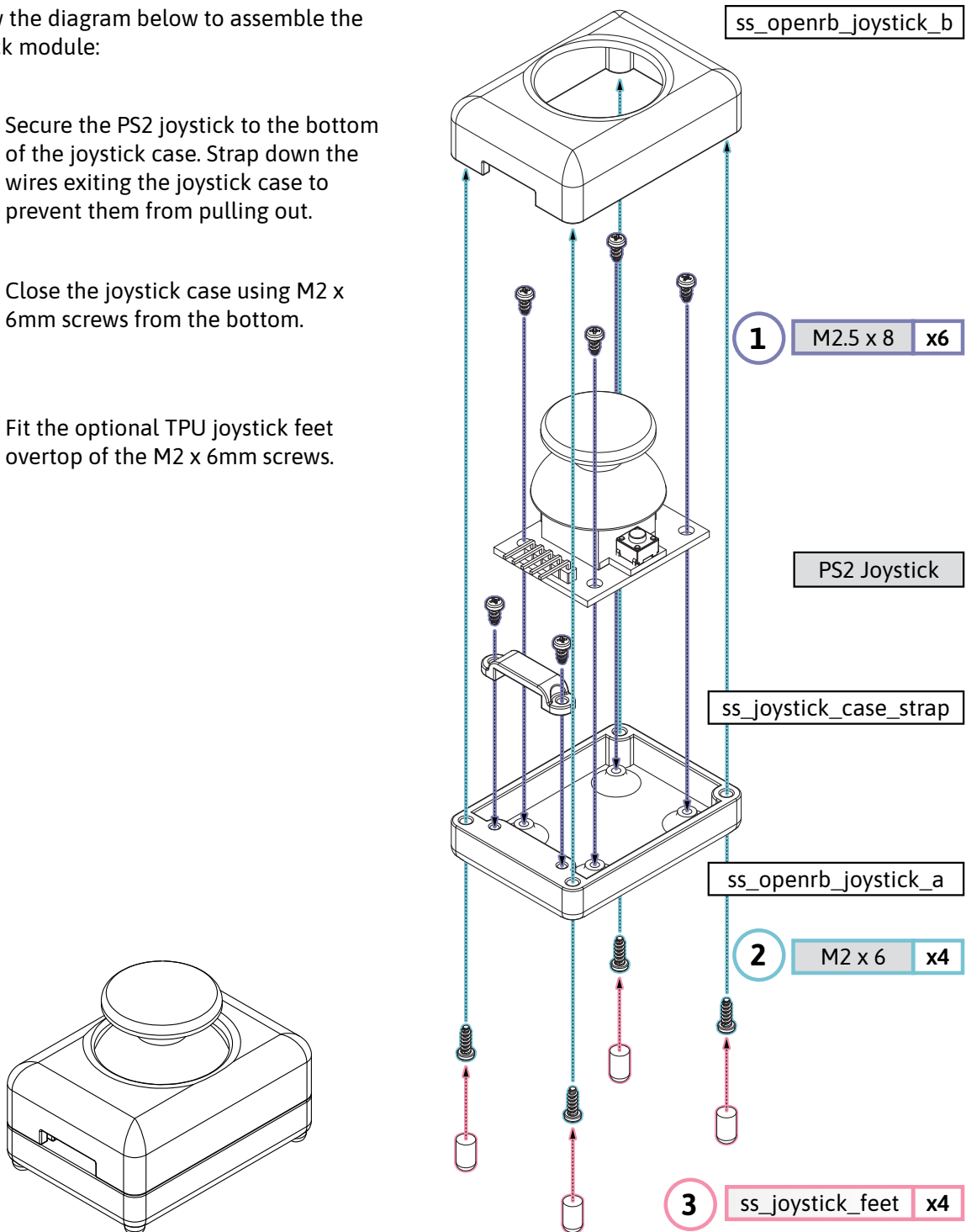
The suggested print specifications for each part of the joystick module are found in the table below. If all parts are printed using Bambu Lab's PLA Basic and TPU 95A HF, it should cost about \$2.50 CAD to print. If none of the parts are being altered or customized, use the gcode files to quickly get started.

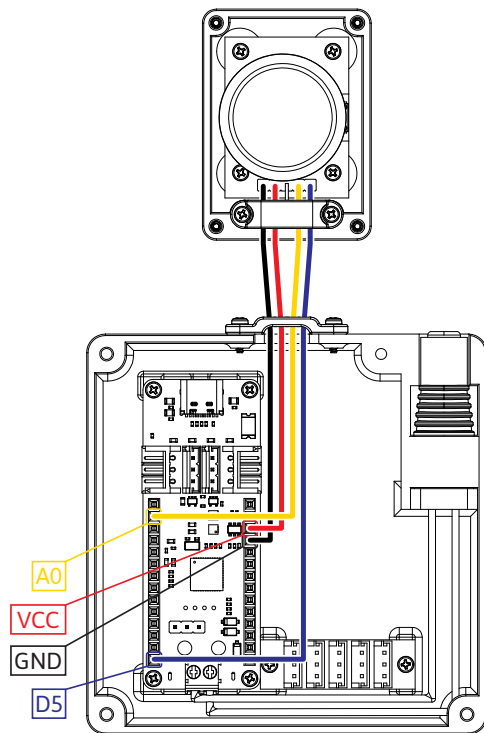
Set Name	Parts in Set	Build Orientation	Colour	Print Specs	Time (min)	Material (g)
ss_openrb_joystick_a	ss_openrb_joystick_a		A	SS Default	68	40.19
ss_openrb_joystick_b	ss_openrb_joystick_b		A	SS Default	43	24.05
ss_strap	ss_strap		A	SS Brim	10	0.70
ss_joystick_case	ss_joystick_case_a		A	SS Default	21	7.18
	ss_joystick_case_b		A	SS Default	21	7.07
	ss_joystick_case_strap		A	SS Brim	10	0.75
ss_joystick_feet (Optional)	ss_joystick_feet		A	SS Default (TPU)	12	0.46
Total:					3 h 5 m	80.40 g

The [link here](#) leads to the public Google Sheets file that contains all the additional items (excluding tools) required for including the joystick module. Makers may create a copy of this spreadsheet for their own convenience. Select “Joystick” using the kit dropdown menu.

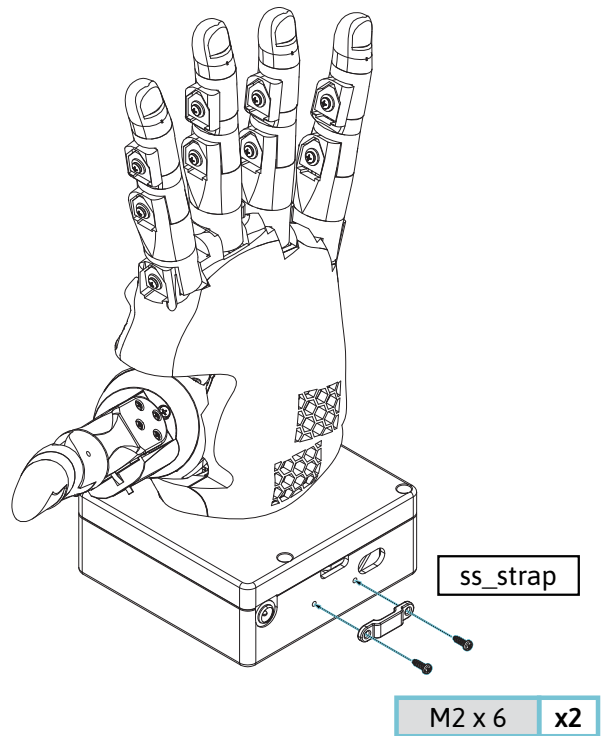
Follow the diagram below to assemble the joystick module:

- 1 Secure the PS2 joystick to the bottom of the joystick case. Strap down the wires exiting the joystick case to prevent them from pulling out.
- 2 Close the joystick case using M2 x 6mm screws from the bottom.
- 3 Fit the optional TPU joystick feet overtop of the M2 x 6mm screws.





- 4** With the power supply and USB cable unplugged, connect the wires between the joystick and OpenRB-150.



- 5** Secure the wires exiting the STANDi Stand to prevent the wires from being pulled out.

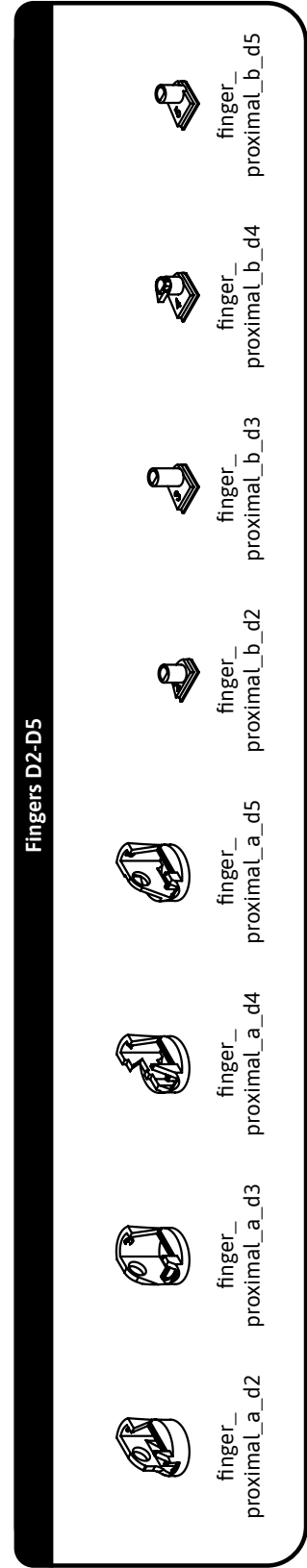
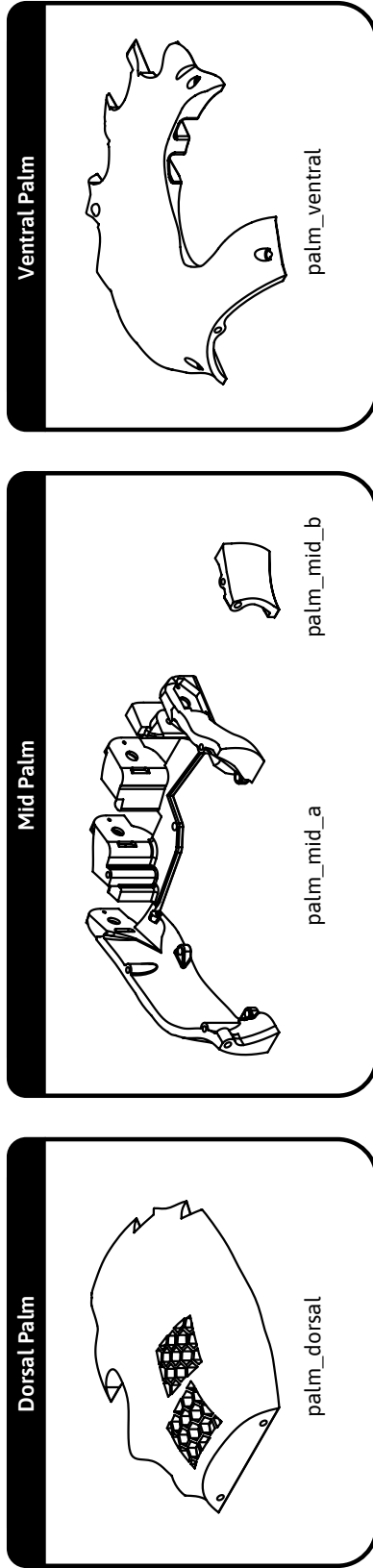
10.3.3. Sensors

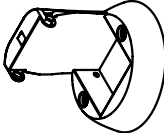
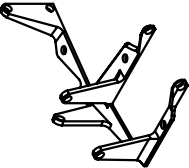
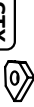
Support for additional sensors such as the TPU fingertips with force sensitive resistors (FSRs) and the wrist-mounted camera is currently on the way. To keep updated on future open-source releases, please regularly check back on BLINCdev.ca.

Appendix A: 3D Printed Template

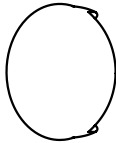
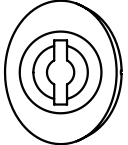
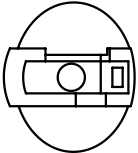
A.1. Introduction

The following eight pages are meant to be printed out to help with the assembly process. The 3D-printed parts of the HANDi Hand can sometimes be difficult to differentiate so a 1:1 template was included so makers can place each part on top of the guide to identify them. It is recommended to print them on 8.5" x 11" letter size paper with the scale set to "actual size".



Thumb		Thumb Motor Skirt	
	thumb_tip_a		thumb_proximal_a
	thumb_tip_b		
	thumb_distal_a		
	thumb_distal_b		
	thumb_intermediate_a		
	thumb_intermediate_b		
Servo Spools		Pot Activators	
	palm_spool_d2		joint_sensor_pivot
	palm_spool_d3		joint_sensor_pivot_d3
	palm_spool_d4		
	palm_spool_d5		
Servo Sled		Pot Placeholder	
	palm_frame_a		joint_sensor_placeholder
	palm_frame_b		

Thumb Assembly Template (1:1)

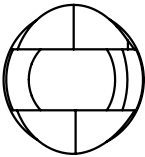
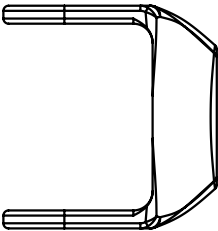





thumb_tip_a

thumb_tip_b

thumb_distal_a

thumb_distal_b

thumb_intermediate_a

thumb_intermediate_b



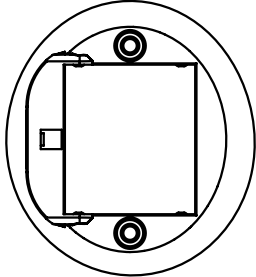
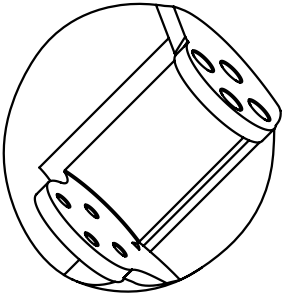



joint_sensor_pivot

joint_sensor_placeholder

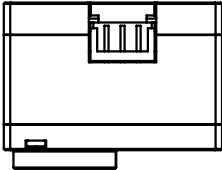
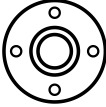
M2 Washer

M2 x 16

thumb_proximal_a

thumb_proximal_b



Dynamixel XL330

Dynamixel Idler Horn

Dynamixel Idler Cap

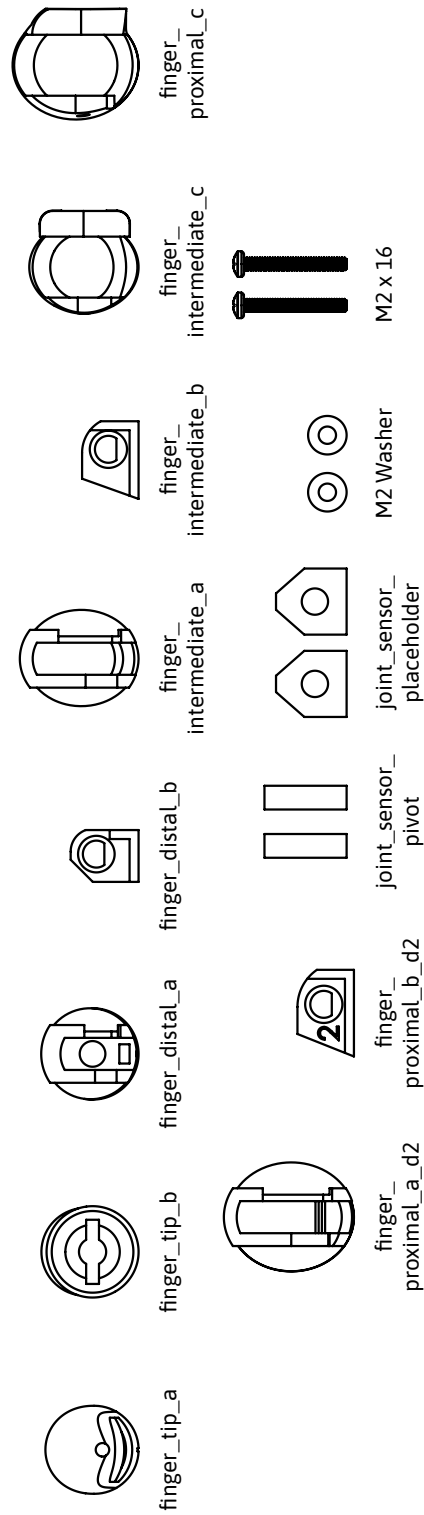
M2.6 x 6 TAP



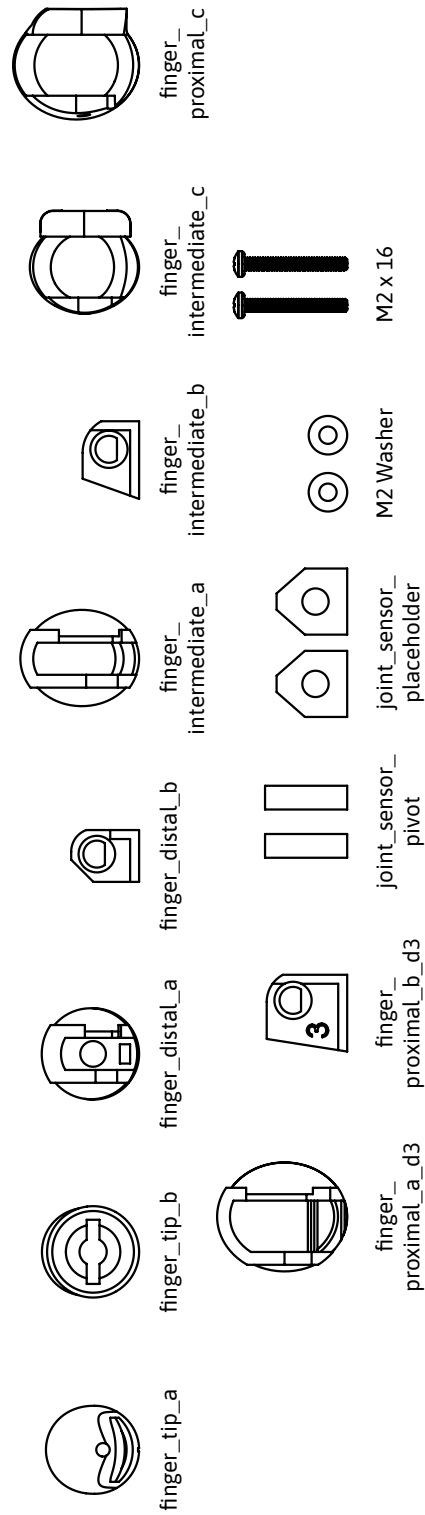

M2 x 8 TAP

M2 x 4 TAP

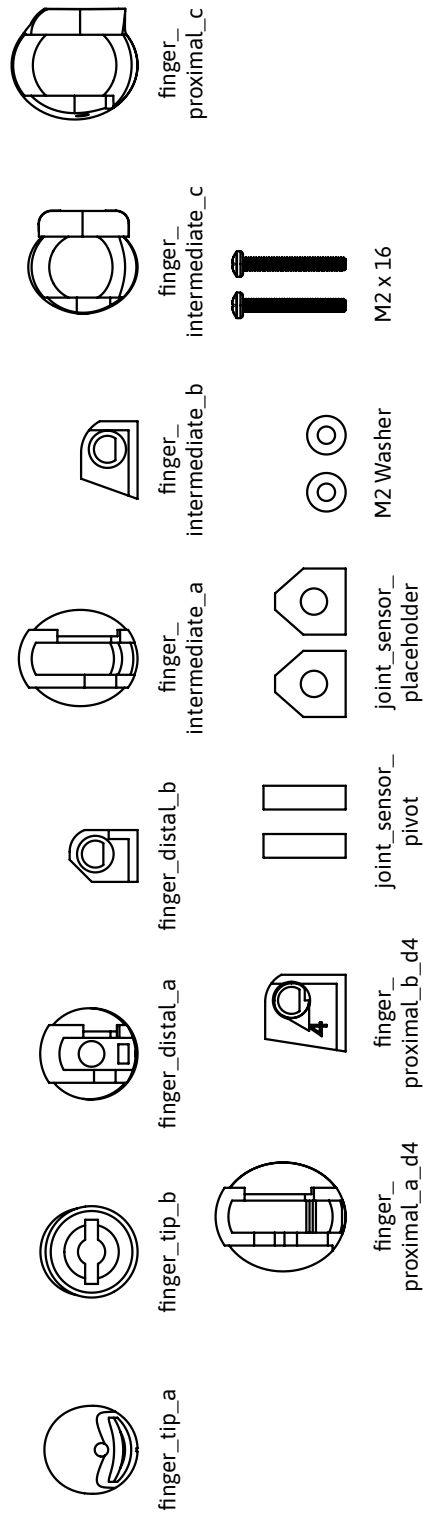
Finger D2



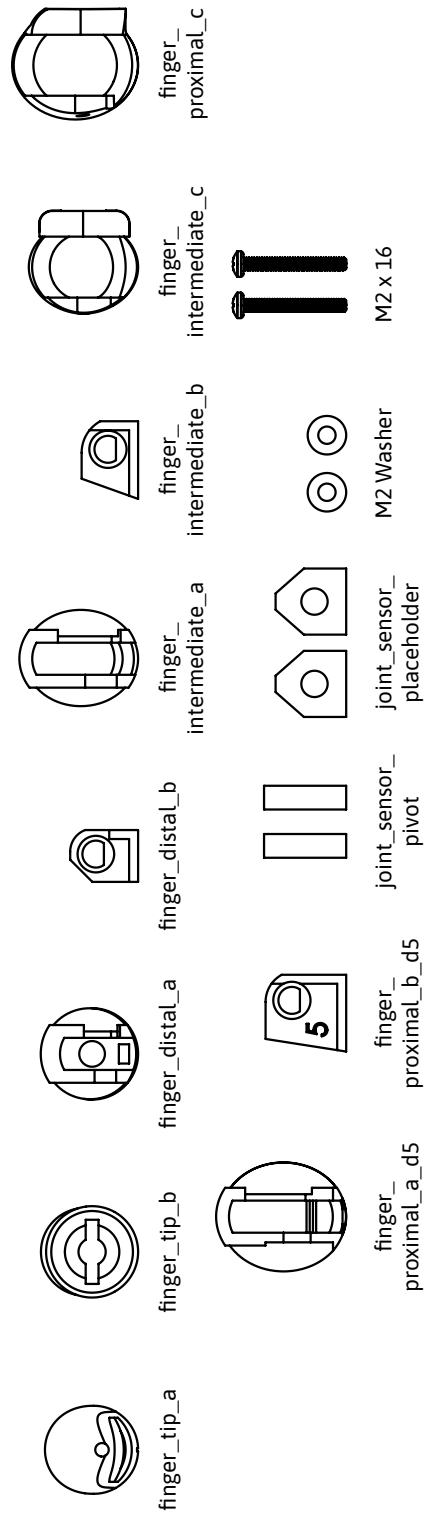
Finger D3

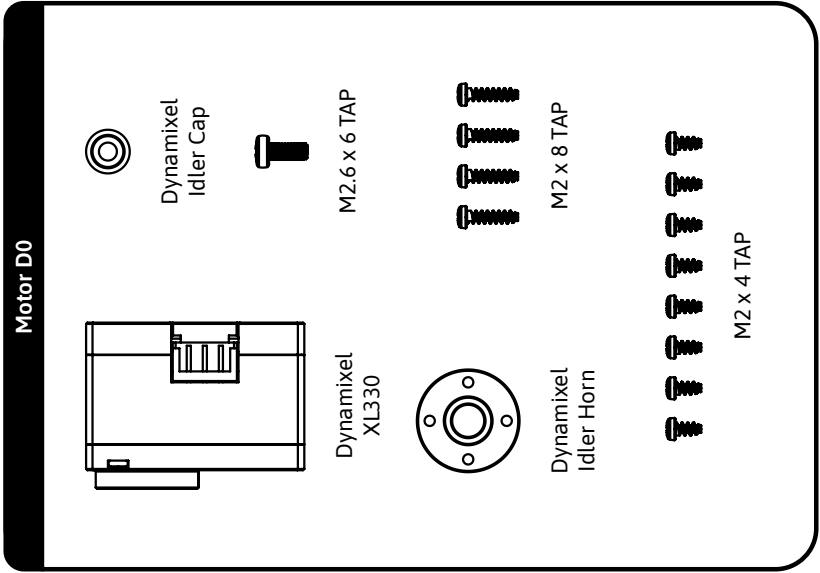
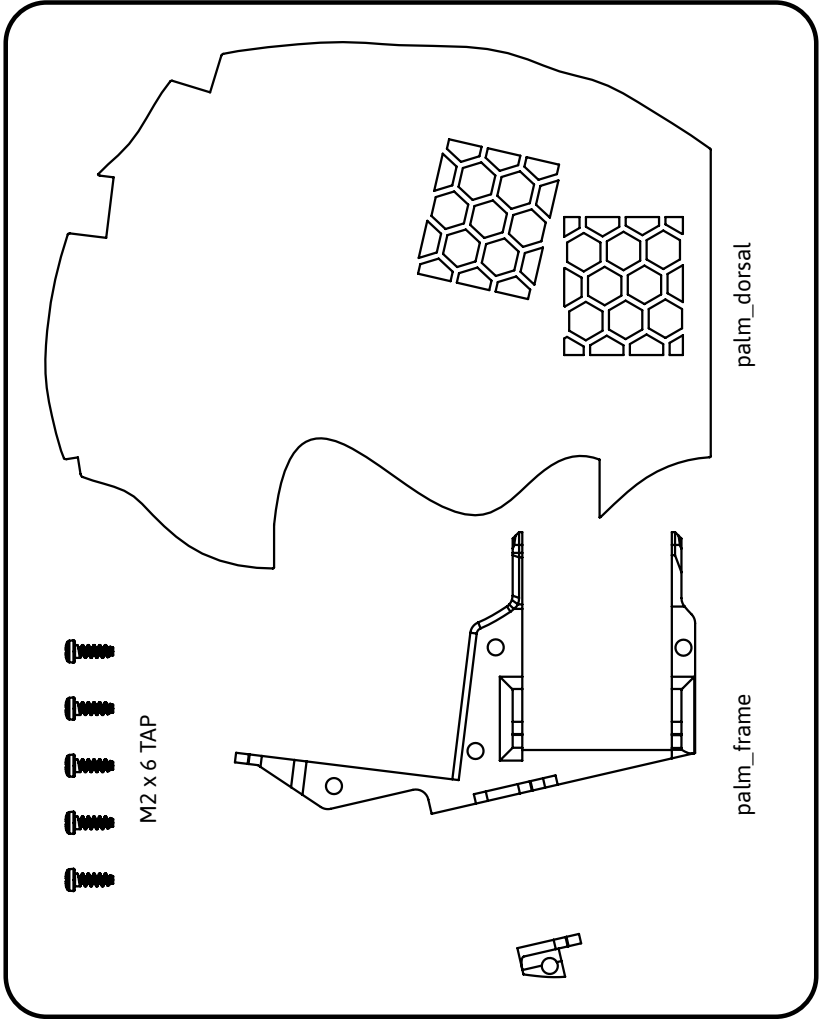


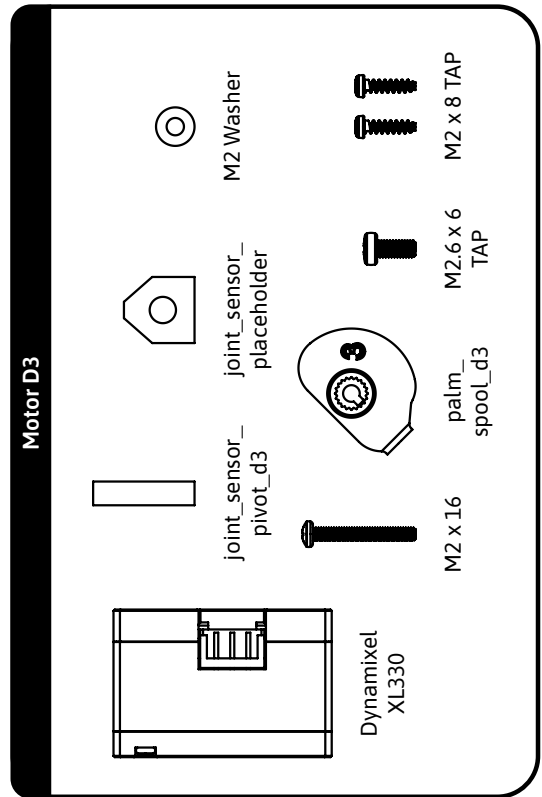
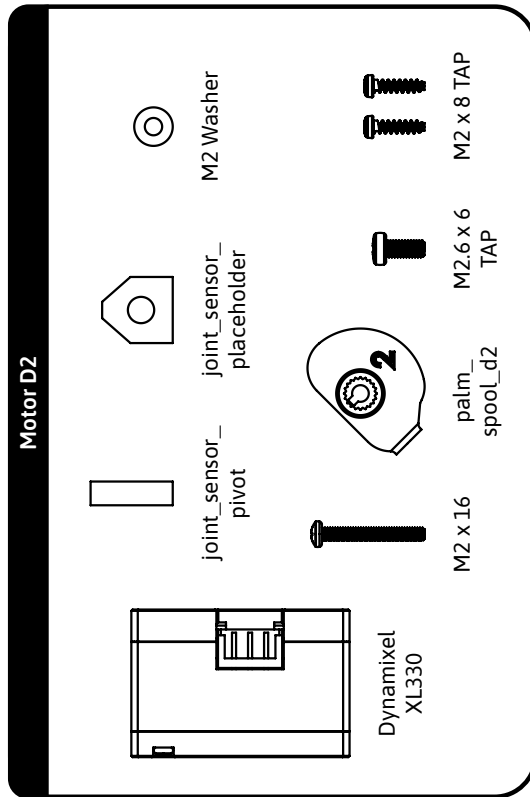
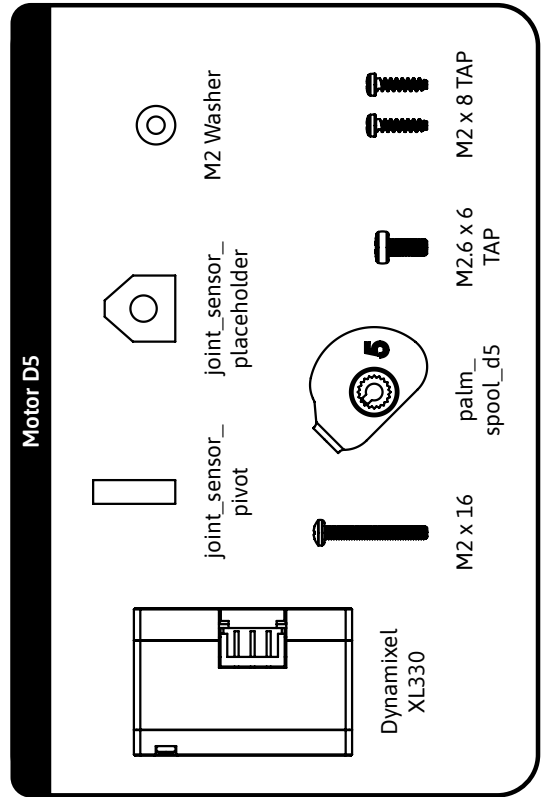
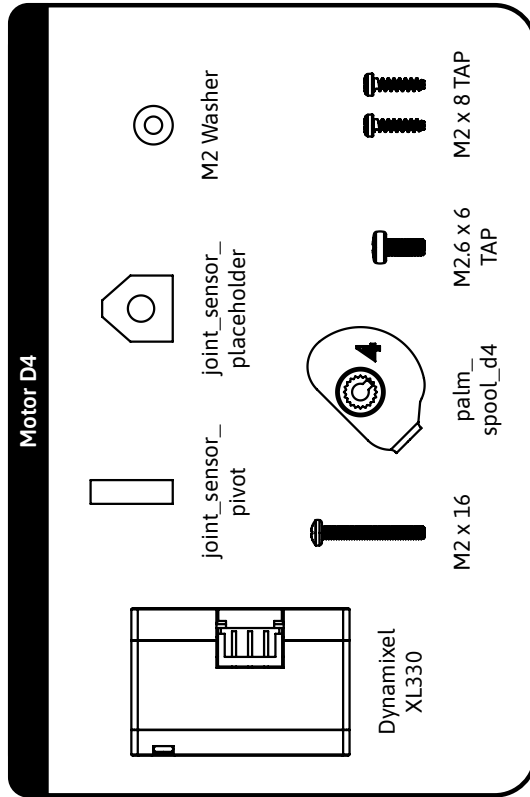
Finger D4

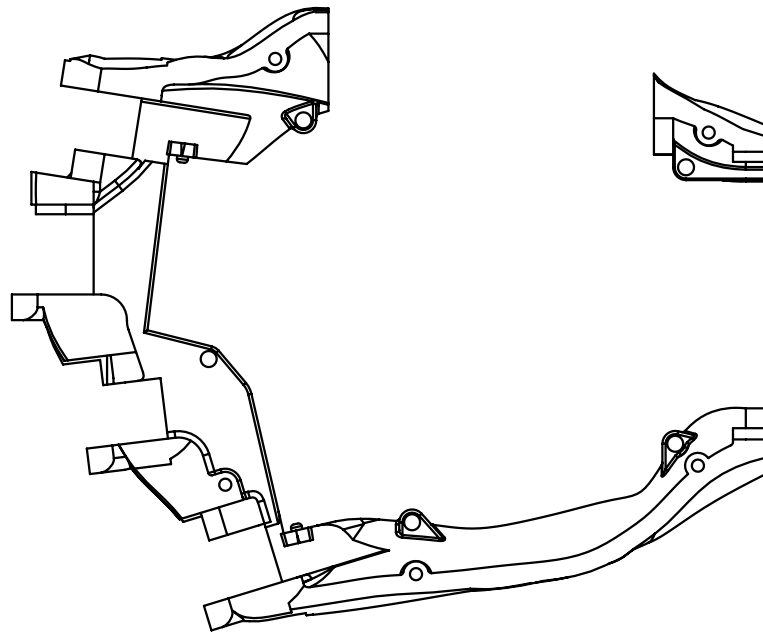


Finger D5

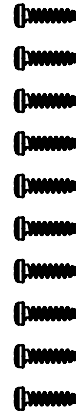




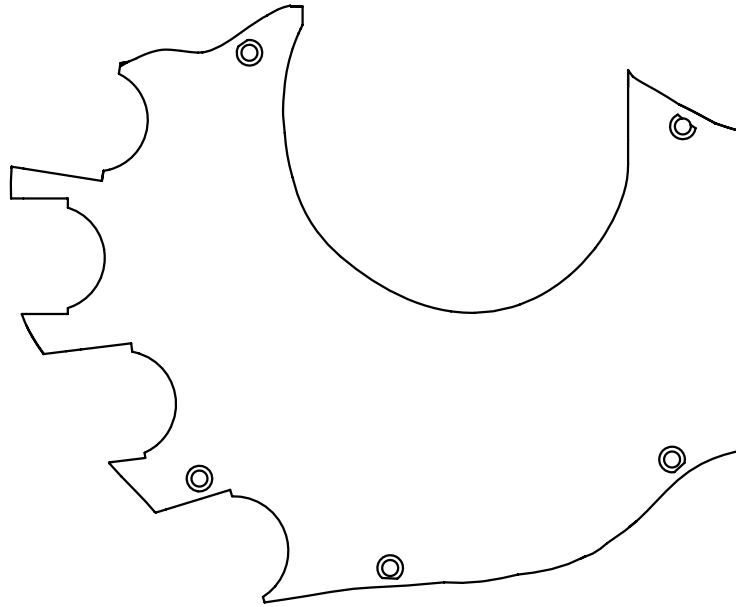




palm_mid



M2 x 8 TAP



palm_ventral



M2 Cone-point Set Screw