

SCUOLA SUPERIORE SANT'ANNA

DOCTORAL THESIS

---

# Design, Implementation and Control of an Underactuated Hand Exoskeleton

---

*Author:*

Mine SARAÇ STROPPIA

*Supervisor:*

Antonio FRISOLI, Professor

*A thesis submitted in fulfillment of the requirements  
for the degree of Doctor of Philosophy*

*in the*

Perceptual Robotics Laboratory

Institute of Communication, Information and Perception Technologies

Scuola Superiore Sant'Anna

Pisa - Italy

April, 2017

---

# Design, Implementation and Control of an Underactuated Hand Exoskeleton

Mine Saraç Stroppa

Perceptual Robotics, Doctor of Philosophy (PhD), 2017.

Thesis Supervisor: Assoc. Prof. Dr. Antonio Frisoli

Keywords: Robotic Rehabilitation, Hand Exoskeletons, Underactuation, Haptic Rendering.

## Abstract

I present the design, implementation and control of a novel, linkage-based, underactuated hand exoskeleton aimed to assist patients with hand disabilities during grasping tasks for robot assisted physical rehabilitation or robot assisted activities of daily living. Even though the proposed exoskeleton was designed with the purpose of assisting users during interactions with real objects around them, its use can be extended for haptic applications as well.

The design requirements of a generic hand exoskeleton, which can be used for all types of applications, were listed after an intensive literature search. These requirements lead us to have a novel kinematics selection. In particular, a generic hand exoskeleton should be portable, lightweight and easily wearable for allowing patients with hand disabilities to use the device. The hand exoskeleton should assist all fingers of the user independently. Using linkage-based kinematics with intentional misalignment between mechanical and anatomical finger joints allows the device to adopt its operation for different hand sizes automatically. The device assists only 2 finger joints of each finger for flexion/extension using a single actuator. Such an assistance can be achieved by adopting underactuation concept, which can adjust force transmission for finger joints based on physical interaction forces. Doing so, the exoskeleton can assist users grasping objects with different sizes and shapes automatically, with no prior information. Finally, the connection mechanics of the device is designed to exert only perpendicular forces to the finger phalanges to increase the realism of natural interaction forces between the user and objects. Overall, the easiness of the attachment to user's fingers, better comfort and improved security are guaranteed.

I performed pose analysis, differential kinematics analysis, statics analysis and stability of grasp analysis for the proposed kinematics. The lengths of mechanical links for each

finger component are optimized to increase range of motion for finger joints and efficacy of force transmission. An additional potentiometer attached on the system allows the finger pose to be predicted during operation. For the electronic components, a DSP board is selected to run all sensory measurements, control algorithms and motor driver connections.

The proposed underactuated hand exoskeleton can be controlled in various ways. First of all, a strict position control is performed based on a simple position controller. The performance of the PI control is enhanced by temperature filters and backdrive force support. The position controller can be used to perform grasping tasks for assistive or rehabilitation applications, thanks to the automatic adjustment of operation. The same position control can be used with a EMG based trajectory, instead of pre-defined passive trajectory. In particular, patients with disabilities are asked to move their unimpaired hand, on which muscular activity can be measured with EMG sensors, and the exoskeleton can assist their impaired hand to perform the same movement. Involving users in the execution of the tasks to define the trajectory turns the task into an active exercise. The EMG activities can move user's impaired hand in a coupled or independent manner. Finally, a force control algorithm is proposed by equipping an additional force sensor for each finger component to let the user open/close his fingers by applying external forces to it. Thanks to the active backdriveability, the exoskeleton can detect user's intentions and follow his intentions or amplify the movements for assistance.

The force control algorithm was extended for stiffness rendering algorithm to provide force feedback to user based on virtual interactions during a haptic task. Since the underactuation property suffers from the lack of controllability of joints, additional rendering strategies are proposed, which can be generalized for any underactuated device in the literature. Feasibility studies show the efficacy of the proposed strategies to determine the transmitted forces along finger joints, while ensuring the safety of these strategies compared to conventional rendering algorithms.

## *Acknowledgements*

It is a great pleasure to extend my gratitude to my thesis advisor Assoc. Prof. Dr. Antonio Frisoli and Assoc. Prof. Dr. Miguel Angel Otaduy with whom I worked with during my period abroad for their guidance and support. I would also like to thank Dr. Massimiliano Solazzi for his advices and support throughout my doctoral studies, and Assist. Prof. Dr. Eray Baran and Assoc. Prof. Dr. Rocco Vertechy for their time to review my work and help me to improve it.

I am grateful for having the decision of moving to Italy and before anyone else, I would like to thank my dearest husband Fabio Stroppa, who gave me a new family. Thank you for all the coffee you made for me, all your love and your understanding. Without your support, I could never have completed the obligatory credits and my doctoral degree.

I would also like to extend my appreciation to my parents, my dearest aunt, and my sweetest cousins for always supporting and loving me despite of all the distance between us. I would also like to thank my new Italian family for welcoming me with open arms.

It was a pleasure to work with Massimiliano Solazzi and Daniele Leonardis and I am very thankful to them for their help and advices during my studies. I am also grateful to Daniel Lobo for all his help during my survival in Spain and his collaboration with Mickeal Verschoor. Many thanks to my group friends in PERCRO who I spent all my working days with for making the laboratory enjoyable. I would also like to thank all the members of Multimodal Simulation Lab in Universidad Rey Juan Carlos in Madrid, Spain for their friendship and brain storming lunch talks. Furthermore, I would like to thank Bengi Deniz Kırındı Demirkıran and many other friends from Turkey, who gave me all their support online in sadness and happiness.

I would like to express my gratitude for Elisa Zanobini, who work in the International Office of Scuola Superiore Sant'Anna, for all her time and effort during any bureaucratic chaos I was passing through.

This research was funded within the project "WEARHAP – WEARable HAPTics for humans and robots" of the European Union Seventh Framework Programme FP7/2007-2013, grant agreement n. 601165 and project "CENTAURO - Robust Mobility and Dexterous Manipulation in Disaster Response by Fullbody Telepresence in a Centaur-like Robot" of the the European Union's Horizon 2020 Programme, Grant Agreement n. 644839.

# Contents

<b>Acknowledgements</b>	<b>iii</b>
<b>Contents</b>	<b>iv</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Physical Rehabilitation . . . . .	2
1.2 Hand Exoskeletons . . . . .	4
1.3 Objectives . . . . .	5
1.4 Contributions . . . . .	6
<b>2 Background and Literature</b>	<b>9</b>
2.1 Anatomical Properties of Human Hand . . . . .	10
2.2 Possible Applications for Hand Exoskeletons . . . . .	14
2.3 Hand Exoskeleton Literature . . . . .	15
2.3.1 Mobility . . . . .	19
2.3.1.1 Hand mobility . . . . .	19
2.3.1.2 Finger mobility . . . . .	20
2.3.1.3 Number of interactions . . . . .	22
2.3.2 Mechanical Design . . . . .	23
2.3.2.1 Kinematics selection . . . . .	24
2.3.2.2 Strategies for adjusting to different hand sizes . . . . .	26
2.3.2.3 Mechanism Placement . . . . .	27
2.3.3 Actuation . . . . .	29
2.3.3.1 Actuator type . . . . .	29
2.3.3.2 Transmission units . . . . .	30
2.3.3.3 Direction of movement . . . . .	31
2.3.4 Operation strategies . . . . .	31
2.3.4.1 Control . . . . .	32
2.3.4.2 Finger pose estimation . . . . .	33
<b>3 Design of the Hand Exoskeleton</b>	<b>35</b>

3.1	Design Requirements . . . . .	36
3.2	Proposed Hand Exoskeleton . . . . .	42
3.2.1	Underactuation . . . . .	43
3.3	Mechanical Pose Analysis . . . . .	46
3.3.1	Inverse Kinematics . . . . .	48
3.3.2	Forward Kinematics . . . . .	52
3.3.3	Analytical Forward Kinematics . . . . .	57
3.3.4	Differential Kinematics . . . . .	62
3.3.5	Statics Analysis and Stability of Grasp . . . . .	66
3.4	Link Length Optimization . . . . .	68
3.4.1	Sensitivity Analysis . . . . .	69
3.4.2	Pre-optimization Constraints . . . . .	72
3.4.2.1	Linear constraints . . . . .	74
3.4.2.2	Static constraints . . . . .	75
3.4.3	Optimization . . . . .	75
<b>4</b>	<b>Implementation of the Hand Exoskeleton</b>	<b>81</b>
4.1	Control Board . . . . .	82
4.1.1	Actuators . . . . .	83
4.1.2	Control Electronics and Motor Drivers . . . . .	84
4.1.3	Sensors . . . . .	85
4.2	Multi-Finger Exoskeleton . . . . .	87
4.3	Control Algorithms . . . . .	88
4.3.1	Position Control . . . . .	89
4.3.1.1	Temperature filter . . . . .	92
4.3.1.2	Grasping forces . . . . .	94
4.3.1.3	Pose analysis . . . . .	97
4.3.1.4	Calibration for finger sizes . . . . .	100
4.3.2	EMG Control . . . . .	102
4.3.3	Force Control . . . . .	107
<b>5</b>	<b>Haptic Rendering</b>	<b>111</b>
5.1	Haptic Devices . . . . .	113
5.1.1	Underactuated Haptic Devices . . . . .	115
5.2	Virtual Interaction . . . . .	118
5.2.1	Actuator Level Stiffness Rendering . . . . .	118
5.2.2	Joint Level Stiffness Rendering . . . . .	120
5.2.3	Stiffness Rendering for Verification . . . . .	123
5.3	Stiffness Rendering for the Proposed Underactuated Hand Exoskeleton . . . . .	129
5.3.1	Rendering Strategy based on Joint Torques . . . . .	130
5.3.2	Rendering Strategy based on Finger Pose . . . . .	137
5.3.3	Comparison between Rendering Strategies . . . . .	144
5.4	Proxy based Haptic Rendering . . . . .	147
5.4.1	Notation . . . . .	147

---

5.4.2	Review of Proxy-based Haptic Rendering . . . . .	148
5.4.2.1	Standard rendering algorithm . . . . .	149
5.4.2.2	Analysis for underactuated systems . . . . .	149
5.4.2.3	Null-space force optimization . . . . .	150
5.4.3	Rendering for Underactuated Devices . . . . .	151
5.4.3.1	Subspace proxy . . . . .	151
5.4.3.2	Linearized subspace proxy . . . . .	152
5.4.4	Results . . . . .	153
<b>6</b>	<b>Conclusion</b>	<b>158</b>



# List of Figures

2.1	Kinematic model of the human hand [1]. (a) dorsal view of the hand shows the definitions for the thumb (I), the index (II), the middle (III), the ring (IV), and the little fingers (V). The thumb has two phalanges (proximal, and distal), while other fingers have three (proximal, middle, and distal) from the palm to the fingertip. (b) ulnar view of the index finger as an example for finger kinematics. MCP, PIP and DIP joints are shown from the palm to the fingertip. . . . .	11
2.2	Hand exoskeletons can be used to lead, assist or resist user's hand movements for: (a) rehabilitation use, (b) assistive use and (c) haptic use. . . .	14
2.3	Hand exoskeletons should be categorized by design selections that can be categorized under design aspects: mobility, mechanical design, actuation and operational strategies. . . . .	15
2.4	Possible design choices for mobility based on hand mobility, finger mobility and number of interaction points. . . . .	19
2.5	Hand exoskeletons can be designed with different numbers of interaction points between the device and user's fingers. Multiple interaction points improve grasping stability, user's safety and perception of touch but are harder to be worn. . . . .	21
2.6	Possible design choices for mechanical design based on kinematics selection, mechanical placement and adjustment strategies for hand sizes. . . .	23
2.7	Types of kinematics selections as hand exoskeletons: the black circles show the mechanical joints, while the red stars represent the actuated ones. Glove-based devices can track finger pose easily and efficiently but are hard to be worn. Linkage-based devices are lightweight, portable and easily wearable. Linkage-based devices can be categorized based on the finger mobility choices detailed in Section 2.3.1.2. . . . .	24
2.8	Hand exoskeletons can be designed to be placed on different locations with respect of fingers. . . . .	28
2.9	Possible design choices for actuation technologies based on actuator selection, transmission system and direction of movement. . . . .	29
2.10	Possible design choices for operational strategies based on control and finger pose tracking strategies. . . . .	32
3.1	Kinematic selection for a finger component of the proposed hand exoskeleton under given assumptions drawn in CAD. . . . .	42

3.2	Underactuation concept of the proposed hand exoskeleton during a grasping task: (a) initial pose with all finger joints extended, (b) actuator force moves MCP joint until the first finger phalange gets in touch with the object, (c) actuator force is transmitted to move PIP joint when the first finger phalange touches the object and (d) finally grasping task is completed when both phalanges touch the object. . . . .	43
3.3	Different poses of finger joints that can be achieved when the actuator displacement is locked due to the extra mobility introduced by the underactuation concept. . . . .	44
3.4	Adaptability of underactuated hand exoskeleton for objects with different sizes and shapes due to the extra mobility. . . . .	45
3.5	Kinematic scheme with point and joint definitions of a finger component for the proposed underactuated hand exoskeleton. . . . .	47
3.6	Independent kinematic loops to solve numerical inverse kinematics. Red arrows show the vectors defined by passive joints and rigid links, while the green arrow shows the active actuator displacement, which can be controlled. . . . .	49
3.7	Finger joint rotations defined as input for the CAD simulation and the kinematics analysis to calculate inverse kinematics. . . . .	52
3.8	Comparison between the simulations of the CAD model and the inverse kinematics analysis through actuator displacement $l_x$ and an additional passive joint $q_B$ . . . . .	53
3.9	First loop for analytical forward kinematics to reach passive joint pose $q_N$ and $q_K$ using actuator displacement $l_x$ and passive joint measurement $q_B$ . . . . .	57
3.10	Second loop to calculate analytical forward kinematics to reach passive joint pose $q_G$ and $q_D$ using passive joint pose values $q_N$ and $q_K$ obtained in the previous loop. . . . .	58
3.11	Third loop to calculate analytical forward kinematics to reach passive slider displacement $c_1$ and MCP joint rotation $q_{o1}$ using previously obtained $q_K$ and sensory measurement $q_B$ . . . . .	59
3.12	Final loop to calculate analytical forward kinematics to reach passive slider displacement $c_2$ and PIP joint rotation $q_{o2}$ using previously obtained $q_D$ , $q_K$ and sensory measurement $q_B$ . . . . .	60
3.13	Simulation setup to compare numerical and analytical forward kinematic analyses using the same measurement set, which is computed by inverse kinematics for a set of finger joint rotations. . . . .	61
3.14	Comparison between numerical and analytical forward kinematics in simulation: (a) comparison for MCP joint rotation and (b) comparison for PIP joint rotation. . . . .	62
3.15	Simulation setup to compare differential kinematics to simple derivative of a set of finger joint rotation. . . . .	65
3.16	Comparison between differential kinematics and simple derivative in simulation: (a) comparison for MCP joint rotation and (b) comparison for PIP joint rotation. . . . .	66
3.17	Computation of sensitivity index for first passive slider movement $c_1$ and second passive slider movement $c_2$ independently. . . . .	71

---

3.18	Computation of performance index $p$ for different sets of link lengths, which satisfy the physical linear and statistical constraints, for index finger component optimization. . . . .	76
3.19	Comparison between anatomical finger workspace and workspace under the assistance of the proposed hand exoskeleton using only the rotation of MCP and PIP joints. . . . .	78
3.20	Maximum flexion and extension pose for index finger component of the proposed hand exoskeleton. . . . .	79
3.21	Maximum flexion and extension pose for the proposed hand exoskeleton with index, middle, ring and little finger components. . . . .	80
4.1	General scheme of electronics components chosen for the actuation and control of the proposed hand exoskeleton: a DSP microcontroller has a control algorithm creating a PWM data to send to the motor driver. This motor driver converts the PWM signal to the current to be given to the linear actuator. As the linear actuator moves, the linear potentiometer inside the actuator and the additional potentiometer measures the movements on the exoskeleton. Furthermore, the force sensor attached on the exoskeleton measures how much force the user applies to open/close his fingers. All of these measurements are converted to digital to be used as feedback by the control algorithm. . . . .	82
4.2	Sample time and data type definitions for different operations performed in DSP control board or communication network with the host computer. . . . .	85
4.3	Implementation of hand exoskeleton with 5 finger components, including the thumb, which is the product of an individual research study. . . . .	88
4.4	Observed actuator displacement with respect to the PWM percentage input to observe the impact of the high backdrive force of the actuator. . . . .	89
4.5	Adjusted position control algorithm with additional PWM supply in the same direction of control algorithm to overcome internal actuator friction. . . . .	90
4.6	Simple PI control results with the step and the ramp input references. . . . .	91
4.7	Simulation plot of how the temperature filter works: (a) adjusted limitation set for the input PWM signal based on the PWM signal values over time, and (b) output PWM signal to be supplied for actuator compared to the PWM limitation. . . . .	92
4.8	Real time implementation of the temperature filter being applied to the PWM signals generated during a PI control for the chosen actuator. . . . .	93
4.9	Real time implementation of grasping tasks of objects with different sizes and shapes while the proposed hand exoskeleton assists two subjects with different hand sizes. . . . .	94
4.10	Placement of the FSR sensors to measure the interaction forces between user's finger phalanges and cylindrical objects during grasping: (a) a water bottle and (b) a mug. . . . .	95
4.11	Interaction forces during grasping between user's finger phalanges and FSR sensors attached on the object in Figure 4.10(a) and object in Figure 4.10(b). . . . .	96

4.12	An image of the rotational potentiometer for an additional sensory measurement along one of the passive mechanical joints to predict user's pose during operation. . . . .	97
4.13	Results of pose estimation for different tasks over the actuator displacement $l_x$ : (a) moving PIP joint with stable MCP joint, (b) moving MCP joint with stable PIP joint, (c) moving both MCP and PIP joints. . . . .	98
4.14	Hand mannequin model to visualize finger joints, which are calculated using pose estimation, simultaneously during operation. . . . .	99
4.15	A representative scheme shows the pose estimation based calibration to estimate the length of the first finger phalange at a pre-defined pose of user's finger. . . . .	100
4.16	Pre-defined pose needed to compute the length of first finger phalange during calibration. . . . .	101
4.17	Myo bracelet provides the muscle activity of hand gestures in a compact manner [2]. . . . .	102
4.18	Distribution of 8 channel EMG sensors around the Myo bracelet. . . . .	103
4.19	A representative scheme to use the EMG measurements as a reference to control fingers independently using a low pass filter and an observation based finger classification process. . . . .	103
4.20	Classifying the most effective EMG channels measured by Myo bracelet corresponding to flexion of index, middle and ring fingers independently. . . . .	104
4.21	The actuator references depicted for independent finger movement through classified EMG activity. . . . .	105
4.22	The experimental setup for EMG-controlled underactuated hand exoskeleton with MYO bracelet. . . . .	106
4.23	Utilization of 1 – $DoF$ force sensors for each finger component independently to measure user's intention to open/close his finger and move the actuator with high transparency to achieve backdriveability over control. . . . .	107
4.24	A representative scheme of a generic force control algorithm that can be used for active backdriveability when $F_d = 0$ or haptic rendering when $F_d$ is adjusted based on the virtual interactions. . . . .	108
4.25	Adjusted force control algorithm with additional PWM supply in the same direction of control algorithm to overcome internal actuator friction. . . . .	108
4.26	Real time implementation of an active backdriveability task using the proposed index finger component of the exoskeleton: (a) measured forces through additional force sensor, (b) PWM input for the given control algorithm in Eqn. (4.1) and (c) obtained displacement of the actuator to open/close the finger. . . . .	110
5.1	Actuator level stiffness rendering while the displacement after the virtual contact are different due to the various stiffness values. . . . .	120
5.2	Underactuation concept through a grasping simulation through a 2 $DoF$ gripper with a single rotational actuator. . . . .	121
5.3	Computing desired pose $\mathbf{q}_d$ to be used for stiffness rendering algorithm for given actual pose $\mathbf{q}$ using Algorithm 3. . . . .	122

5.4	Finger motion during the experiment: (a) the user aligns his MCP joint with the mechanical joint of the object to rotate his MCP joint, while his PIP joint remains with minimum movement. (b) the user aligns his PIP joint with the mechanical joint of the object to rotate his PIP joint, while his MCP joint remains with minimum movement. . . . .	124
5.5	FSR Interface to measure forces on the finger phalanges: the finger contact with a mechanical joint allows the user to grasp and squeeze the interface. The FSR sensor measures the pressure applied by the user's finger. The silicon interface aims to imperfection the diversities of how the forces are applied to the FSR sensor. The spring allows the user to squeeze the object, imitating a soft object during haptic rendering task. . . . .	125
5.6	Experimental results for force applied by the exoskeleton vs. force measured by FSR and the pose estimation for: (a) stiffness rendering task around the MCP joint; (b) stiffness rendering task around the PIP joint. . . . .	126
5.7	Sensed impedance calculations in order to study the stability: (a-a) actual impedance value over time for experiment Figure 5.4 (a), (a-b) rotation around active joint MCP and its virtual limit, (b-a) actual impedance value over time for experiment Figure 5.4 (b), (b-b) rotation around active joint PIP and its virtual limit. . . . .	127
5.8	Simulation plot for the desired torque optimization based on minimizing the error between the real and the proxy torque sets. . . . .	132
5.9	Desired torque optimization for different stiffness assumptions: (a) $K1_{cont} = [5 \ 0; \ 0 \ 10]$ ; (b) $K2_{cont} = [1 \ 0; \ 0 \ 0.2]$ . . . . .	133
5.10	Desired torque optimization based on minimizing the error between the real and the proxy torque sets: (a) trajectory of the real time task performed by the user $\mathbf{q}$ with the virtual limits $\mathbf{q}_{lim}$ defined for the joints, (b) desired torque values $\boldsymbol{\tau}$ calculated for user's trajectory, (c) optimized proxy set of torque values $\boldsymbol{\tau}^*$ calculated for user's trajectory and (d) corresponding actuator force for the proxy torque values $F_a^*$ . . . . .	134
5.11	How additional FSR sensors are inserted between the hand exoskeleton and the user. . . . .	135
5.12	Comparison between the calculated proxy set of desired torques and measured torques through FSR measurements. . . . .	136
5.13	Simulation plot for configurational space optimization problem based on minimizing the distance between desired finger pose and proxy finger pose. . . . .	139
5.14	Desired displacement optimization for different previous behavior: (a) only MCP joint was being rotated (ratio = 0.01); (b) only PIP joint was being rotated (ratio = 100). . . . .	140
5.15	Proxy pose optimization based on minimizing the error between the real and the proxy pose sets: (a) trajectory of the real time task performed by the user $\mathbf{q}$ with the virtual limits $\mathbf{q}_{lim}$ defined for the joints, (b) optimized proxy set of finger pose $\mathbf{q}^*$ calculated for user's trajectory, (c) desired finger torques $\boldsymbol{\tau}^*$ calculated using proxy set of finger pose $\mathbf{q}^*$ and (d) corresponding actuator force for the proxy torque values $F_a^*$ . . . . .	141
5.16	Validation of user's behavior estimation. . . . .	142

5.17	Validation of user's behavior estimation after user reaches the virtual limit pose. . . . .	143
5.18	Comparison between desired torques and actual torques measured by FSR sensors during rendering strategy based on proxy joint pose. . . . .	144
5.19	Stiffness rendering algorithms for force space and operational space individually. . . . .	145
5.20	Sensed impedance $S_a = F_a^*/\mathbf{q}$ calculated during the experiment in Figure 5.15 for (a) proxy torque strategy in an offline manner and (b) proxy pose strategy in an online manner. . . . .	146
5.21	Schematic representation of the device configuration space $Q$ (left) and the virtual configuration space $\chi$ (right). $\mathbf{q}$ represents the device state and $f(\mathbf{q})$ its corresponding configuration in the virtual environment, i.e., the haptic probe; $\mathbf{x}$ represents the standard proxy; $\mathbf{q}^*$ represents the subspace proxy and $f(\mathbf{q}^*)$ its corresponding configuration in the virtual environment. The images also represent the linear subspaces of actuated and non-actuated motion in the virtual environment, $\mathbf{J}_a$ and $\mathbf{J}_n$ . . . . .	148
5.22	Contact with a soft hand being rendered through an underactuated exoskeleton. . . . .	154
5.23	Performance comparison of the three rendering methods discussed in the paper (standard, optimized, and subspace). In the test, the proxy is kept still at a zero angle. . . . .	156

# List of Tables

2.1	Average finger phalange sizes for index, middle, ring and little fingers [3].	12
2.2	Average ratios between proximal and middle phalanges over distal phalange for the index, middle, ring, and little fingers [3]. . . . .	12
2.3	Ranges of motion (RoM) for finger joints during flexion/extension: means (standard deviations) in degrees [4]. . . . .	13
2.4	<b>5 fingered hand exoskeletons:</b> Main application (rehabilitation (R) / assistive (A) / haptic (H)), number of independent fingers, number of assisted and independent DoF for each finger, mechanism type, device placement (dorsal (DOR)/ palmar (PAL)/ lateral (LAT)), actuator, control modes (position (POS)/ velocity (VEL)/ backdriveable (BAC)/ admittance (ADM)/ impedance (IMP)/ EMG triggered (EMG)), pose estimation method (encoder (ENC)/ flex sensor (FLE)/ motion tracking (MT)/ additional sensor (SEN)) . . . . .	16
2.5	<b>4, and 3 fingered hand exoskeletons:</b> Main application (rehabilitation (R) / haptic (H) / assistive (A)), number of assisted and independent fingers, number of assisted and independent DoF for each finger, mechanism type, device placement (dorsal (DOR)/ palmar (PAL)/ lateral (LAT)), actuator, control modes (position (POS)/ velocity (VEL)/ backdriveable (BAC)/ admittance (ADM)/ impedance (IMP)/ EMG triggered (EMG)), pose estimation method (encoder (ENC)/ flex sensor (FLE)/ motion tracking (MT)/ additional sensor (SEN)) . . . . .	17
2.6	<b>2, and 1 fingered hand exoskeletons:</b> Main application (rehabilitation (R) / haptic (H) / assistive (A)), number of assisted and independent fingers, number of assisted and independent DoF for each finger, mechanism type, device placement (dorsal (DOR)/ palmar (PAL)/ lateral (LAT)), actuator, control modes (position (POS)/ velocity (VEL)/ backdriveable (BAC)/ admittance (ADM)/ impedance (IMP)/ EMG triggered (EMG)), pose estimation method (encoder (ENC)/ flex sensor (FLE)/ motion tracking (MT)/ additional sensor (SEN)) . . . . .	18
3.1	Variable and constants for the vectors to solve inverse kinematics . . . . .	48
3.2	Input and output parameters used for each loop and unique output parameters for inverse kinematics. . . . .	51
3.3	Variable and constants for the vectors to solve forward kinematics. . . . .	54
3.4	Analyzing input and output parameters used for each loop and unique output parameters for forward kinematics . . . . .	55

---

3.5	Variable and constants for the vectors to solve forward kinematics for calibration. . . . .	56
3.6	Analyzing input and output parameters used for each loop and unique output parameters to solve forward kinematics for calibration . . . . .	56
3.7	Matrix sizes of Jacobian matrix components. . . . .	64
3.8	Results of generic sensitivity index. . . . .	71
3.9	Constant variables for the index finger. . . . .	72
3.10	Constant variables for the middle finger. . . . .	72
3.11	Constant variables for the ring finger. . . . .	72
3.12	Constant variables for the little finger. . . . .	73
3.13	Range of variables for index finger optimization. . . . .	73
3.14	Range of variables for middle finger optimization. . . . .	73
3.15	Range of variables for ring finger optimization. . . . .	74
3.16	Range of variables for little finger optimization. . . . .	74
3.17	Results of optimization for index finger. . . . .	76
3.18	Results of optimization for middle finger. . . . .	77
3.19	Results of optimization for ring finger. . . . .	77
3.20	Results of optimization for little finger. . . . .	77
3.21	Ranges of motion for all finger joints with hand exoskeleton . . . . .	78
4.1	Specifications of Firgelli L16 linear actuator. . . . .	84
4.2	Specifications of the proposed exoskeleton and its actuators. . . . .	87

# Chapter 1

## Introduction

As humans, we can sense the physical environment around us, discriminate the properties of objects we encounter with or perform tasks with high dexterity through our hands. Therefore, the functionality of our hands is highly crucial for our daily lives. Unfortunately, hand injuries are very common, and the complexity of such an important organ makes it harder to be treated in the event of these injuries. This study mainly focuses on developing a portable, efficient, and wearable hand exoskeleton. Besides our main focus, we want to create an exoskeleton that can be used for assistive applications, where the exoskeleton assists patients to complete various tasks during their daily lives, or for haptic applications, where the exoskeleton renders kinesthetic feedback based on teleoperation tasks or user's activity in the virtual environment.

This chapter focuses on the properties and the requirements of physical rehabilitation. The importance and the positive impact of utilizing robotic devices to assist rehabilitation exercises are stated in general and specifically for the hand. Finally, the objectives and contributions of this study, which presents the design and implementation of a novel hand exoskeleton, are listed briefly.

## 1.1 Physical Rehabilitation

The nervous system can be explained as an incredibly complex communication system, that controls and regulates the functions of body. This system is composed of the brain, the spinal cord and an intricate network of nerves. Unfortunately, this extraordinary system is highly vulnerable to diseases or injury [5]. Neurological injuries are the leading cause of serious, long-term disabilities that restrict the daily functions of millions of patients. Stroke is one of the most serious neurological injuries and approximately 9 million people have had a stroke in 2008.

30 million people have previously had a stroke and are still alive [6]. 75% of stroke survivors suffer from the disabilities that affect their daily living routines physically, mentally, and emotionally [7]. When the stroke affects the upper limbs, having hand injuries is inevitable. After these injuries, stroke patients not only experience discomfort and pain, but also get affected significantly during the daily living activities in physical, psychological, social, and financial aspects of their lives [8]. Survivors mostly experience pain and stiffness for up to four years after their injury.

Especially during the initial stage of the injury, patients with hand disabilities cannot perform many activities of daily living (ADLs) without any help or assistance from other people. Not being able to perform even the simplest task during their daily lives might cause them to go through a psychological burden, such as frustration, discouragement and loss of confidence. In the meantime, they might face with stress, the loss of hope, and the fear of getting injured again. All of these emotions might lead to avoidance of many activities. In addition, sustaining hand injuries might have a negative impact on social and financial security, as the individuals will be absent from their works or may lose their employment following the injury.

Physical rehabilitation is an indispensable solution to treat patients, who are dealing with disabilities of neurological injuries, and to help them regain their functional abilities. There are two main approaches of physical therapy to treat hand injuries. The first one focuses on increasing the effective Range of Motion (RoM) for the impaired finger joints. These therapy exercises consist of repetitive implicit joint rotations, without considering any possible reflections in the real life. The second approach simulates tasks and scenarios from real life, and provides a platform for survivors to practice these given scenarios. By engaging with daily activities repeatedly in a clinical setting, these survivors might regain the functionality of their hand.

Recovering from the injuries in many instances requires a long treatment period and often results in a variable but persistent disability. Such a recovery is challenging even if the treatment procedure is initiated in an early stage of disability. The treatment of hand injuries might take even longer time compared to other regions of the body. The rehabilitation therapies were found to be more effective when they are repetitive [9], intense [10], long term [11], and task specific [12].

Conventionally, the physical rehabilitation sessions are designed and performed by a therapist [13]. During the sessions, the therapist decides the needs and physical limitations of the patient, and designs a unique path of rehabilitation based on the physical examination. For every session, the patient meets the therapist in the clinic, as the therapist physically leads the impaired limbs of the patient. The motivation of these sessions is mostly to increase the motion, dexterity and/or strength of the impair hand. To achieve these goals, the rehabilitation exercises to be performed should be motivations, repetitious, interesting, challenging and graded.

The technological developments created robot-assisted rehabilitation devices that can be used during therapy exercises instead of conventional, manual assistance/guidance techniques. Such robotic rehabilitation devices have been shown to be improving the functional independence of patients when integrated to the clinical therapy exercises [14–17]. During the robot-assisted rehabilitation therapies, the patient still needs the therapist to design the therapy pathway, but they can use these devices instead of manually guiding all the tasks for the patients. As a result, they eliminate the physical burden of repetitive physical therapy for the therapists. Furthermore, a clinic can operate with less number of therapists but more health staff, and still treat more patients at a given time. Such a practicality might actually decrease the overall treatment cost, despite the initial cost of purchasing these devices. Robot assisted rehabilitation also increases the reliability and accuracy of desired tasks, ensuring that the same task can be repeated by patients. Providing quantitative measurements might allow the therapist to track patient's progress over time. Finally, various control algorithms can be implemented on the same robotic device to address different therapy scenarios that might improve the treatment for patients with various levels of impairments, while motivating them to endure long, intense therapy sessions.

The changes in the technology and current trends in the rehabilitation create the need of developing better treatment scenarios. Portable, low-cost and user friendly hand devices might even carry the rehabilitation exercises from hospitals to homes.

## 1.2 Hand Exoskeletons

A hand exoskeleton is a wearable, haptic device that provides realistic kinesthetic feedback to user's fingers through active force transmission over a series of mechanical components. Such a device applies forces to the fingers in order to move and to assist them completing a pre-defined task to imitate ADLs in a natural manner. The interaction with users, especially with patients suffering from hand disabilities, the stability of the device, ergonomic and comfortable mechanical design, and efficacy of force transmission become highly crucial. A hand exoskeleton can be used for physical rehabilitation as much as haptic applications based on their mechanical properties.

The human hand has a very complex kinematics. Many researchers tried to analyze and imitate the natural behavior of a human hand in robotic devices to be used for industrial grippers [18–20], prosthesis [21–23] or humanoid robotics [24–27]. Despite of these efforts, we are still nowhere near getting close to copying the whole behavior, so reaching realistic devices is still an up to date challenge. Creating hand exoskeletons that can allow the natural and realistic behavior of a human hand is even more challenging due to the safety measures and ergonomic and easy wearability of the mechanism that should be considered for any mechanical system with human interaction.

A hand exoskeleton should be designed to satisfy generic requirements, such as user's safety, ergonomy and compliance with the natural behavior of the user, and specific requirements, which might depend on the target application, such as rehabilitation, assistance, teletoperation or haptics. Most of the hand exoskeletons existing in the literature are designed for a specific purpose, therefore with limited requirements. However, the diversity of the design requirements of each application type and the complexity of the hand model forces the hand exoskeletons in the literature to be defined mostly for one type of application in an efficient manner. Therefore, the literature has an increased interest of generic hand exoskeletons that can be used for all types of applications in an easy and efficient manner.

### 1.3 Objectives

Our motivation is to develop a hand exoskeleton to be used mainly for rehabilitation purposes, but also for assistive and haptic tasks. Such an exoskeleton must satisfy a few conditions for each application. In the first one, the hand exoskeleton should be assisting patient's fingers along desired tasks repetitively. It is important to create an easily wearable device with no pose requirement since different disability levels of patients might create difficulties and pain during the preparation time. A generic device that can be useful for different disability definitions and levels should be providing passive tasks, where the device leads the user's fingers to a predefined trajectory, or active tasks, where the device measures the user's intentions through force, pressure or muscular activity sensors and allow them to contribute to the tasks based on their performances and abilities. Meanwhile, the exoskeletons provide realistic feedback to patients during serious game scenarios.

For the second application, the hand exoskeleton should be used for is to assist patients with hand disabilities during ADLs. One of the most common daily activities is to grasp different objects around the environment. Grasping real objects under the assistance of the exoskeleton is possible only when the internal part of the fingers is free. With the same motivation, the fingertip should be left free in case the user needs to press objects. The device should be portable in order to allow the user to explore the environment with minimum attachment to a stationary place. Such a hand exoskeleton should assist the user to grasp various objects with any shape and any size in an automatic manner.

For the final application, user's finger movements should be estimated and represented in a virtual environment. Furthermore, the exoskeleton can give kinesthetic feedback to the user based on his interactions in the virtual environment with other objects. Even though the proposed exoskeleton suffers from the lack of controllability, which was chosen for the sake of simplicity and task adjustability during grasping tasks for objects with different sizes and shapes, during virtual tasks, utilizing further control strategies can provide sufficient performance regarding the perception of virtual grasping.

Besides these specific properties, the proposed hand exoskeleton should have a light-weight mechanical design to minimize the fatigue during all types of operations and an efficient force transmission in a natural manner.

## 1.4 Contributions

The contributions of this study to the literature can be summarized as follows:

- A novel kinematics design was developed in the form of a single finger component of the hand exoskeleton, for which the international patent application is pending [28]. The properties of this kinematics can be listed as :
  - achieving 2 *DoFs* mobility to rotate the finger joints thanks to the linkage based kinematic selection,
  - performing independent finger control by actuating each finger component by a single motor,
  - automatically adapting the operation for different hand sizes by completing the kinematics loop only when worn by the user,
  - automatically adapting the operation for different object sizes and shapes thanks to the underactuation concept based on the contact forces, and
  - constraining the applied forces to be always perpendicular to the finger phalanges.
- The initial feasibility study [29] presented various performance analyses and link length optimization. In this work,
  - The inverse kinematics was analyzed for the given mechanism design,
  - The differential kinematics was derived with the assumption of an additional sensory tool, and
  - The link length optimization was performed to find a set of link lengths with the best force transmission performance satisfying the following constraints:
    - \* the required actuator displacement should be less than the stroke of the chosen actuator,
    - \* the displacement of the passive sliders should be less than the length of finger phalanges,
    - \* the ratio between the joint torques should be in a reasonable range for the safety and comfort of the user, and
    - \* all possible combinations of finger pose in the natural range of motion should be reached.

- The interaction forces between the user and the object while grasping various objects were presented to show the adaptability of the device for various grasping tasks.
- The undersensing property of the device was overcome to reach the pose estimation in an online manner as the user moves attached to the exoskeleton [30]. The content of this work can be summarized as:
  - utilizing an additional potentiometer to measure the rotation of one of the of passive joints,
  - reaching a unique solution for the forward kinematics numerically thanks to the additional measurements,
  - presenting the pose estimation results for various tasks, and
  - creating a calibration process to estimate the length of the first finger phalange at a certain pose.
- The proposed hand exoskeleton was also used to perform a haptic grasping task in a virtual environment simulation by proposing an optimized solution to overcome the issues of the underactuation [31]. With this study, we won the Best Student Presentation awards in IEEE World Haptics Conference, 2017. We are recently invited to submit an extended version of this work as a journal for IEEE Transactions on Haptics. In particular,
  - the literature based on the underactuated devices in haptics was investigated,
  - the existing methods for haptic rendering were detailed,
  - a new method was proposed to implement haptic rendering tasks for underactuated devices, and
  - a set of experiment results were used to show the feasibility of this method in which the hand exoskeleton was being worn by a user to perform a simple virtual grasping task in Unreal simulation engine.
- A journal is being prepared for submission regarding the design of specific control algorithms for underactuated systems stating:
  - a force control algorithm to be implemented for the underactuated hand exoskeleton,

- an experimental stage for stiffness rendering task, where the underactuated device is limited by the task to perform only 1 – *DoF* rotation,
- two different optimization algorithms to implement stiffness rendering task to the underactuated exoskeleton with no limitations on the task, by
  - \* optimizing the desired torques to satisfy the underactuation constraint, or
  - \* optimizing the desired finger pose that is predicted to be reachable based on the previous behavior of the user.

The rest of this study goes as follows: Chapter 2 will list the hand exoskeletons existing in the literature for rehabilitative, assistive, or haptic applications. These devices in the literature will be categorized in terms of their: (i) mobility, (ii) number of contact points, (iii) mechanism type, (iv) actuation and (v) control. Chapter 3 will describe the design requirements for a hand exoskeleton to be used for assistive and rehabilitative exercises specifically using the previous categorizations studied for the devices in the literature. These requirements will define the kinematics of a novel mechanism for an exoskeleton. Once the device kinematics is defined, the link lengths of each finger component will be optimized to maximize the Range of Motion (RoM) of finger joints and the efficiency of the force transmission to finger phalanges. Chapter 4 will focus on the implementation of mechanical design and electronic components. Later on, various control algorithms will be designed and tested on a single finger component based on (i) position control to follow a strict reference, (ii) position control to follow an EMG based reference set by the healthy hand of the user and (iii) force control to follow a force reference set by the therapist or user's interactions in the virtual environment. Chapter 5 will detail the implementation of various haptic rendering strategies to improve the performance of underactuated exoskeleton by (i) modeling the user's hand as  $2DoFs$  with stiff bone structure and (ii) modeling the user's avatar as countless  $DoFs$  due to the deformative model definition of user's hand. Finally, Chapter 6 will conclude this study by summarizing the overall findings, achievements and future works that are left unfinished.

## Chapter 2

# Background and Literature

This chapter will first focus on the anatomical varieties and average properties of human hands. Then, it will reveal how hand exoskeletons in the literature, in terms of active mobility, actuation, number of connection points, and control while detailing the advantages and disadvantages for each application individually. In fact, the literature consists of a wide range of hand exoskeletons with many different properties, characteristics and purposes of use. The existing literature surveys [32-34] provide a good beginning. However, further analysis might be useful to create a generic hand exoskeleton, and to keep up with the latest technological development in the literature.

Despite of common characteristics to be fulfilled, the restrictions and requirements of a hand exoskeleton are shaped by the purpose of use. There is not a unique, straightforward design kinematics for a hand exoskeletons due to the variations of these design properties. Therefore, a systematic classification of the existing robotic devices in the literature is useful as a prior study in order to understand the consequences of all these decisions. Investigating the design aspects such as the active and passive degrees of freedom for each finger, portability, wearability, the kinematic structure, actuation and control strategies can be useful before starting the new design of a hand exoskeleton.

## 2.1 Anatomical Properties of Human Hand

A human hand has five fingers; thumb, index, middle, ring and little fingers as in Figure 2.1. From the kinematic point of view, index, middle, ring and little fingers are similar with different phalange lengths, while the thumb is different than the rest of the fingers. Overall, anatomy of a human hand can be modelled with 20 *DoFs* with 15 joints.

The index, middle, ring, and little fingers have 3 phalanges named as proximal, middle and distal from the palm to the fingertip. Each phalange is connected to each other with joints named as metacarpophalangeal (MCP), proximal interphalangeal (PIP) and distalinterphalangeal (DIP). With these 3 joints, a finger can reach 4 degrees of freedom (DoFs) mobility [35], such that PIP and DIP joints perform only flexion/extension, while MCP joint performs both flexion/extension and abduction/adduction movements.

On the other hand, the thumb has 2 phalanges named as proximal and distal from the palm to the fingertip, and 3 joints named as Carpometacarpal (CMC), metacarpophalangeal (MCP) and interphalangeal (IP). With these 3 joints, a finger can reach 4 *DoFs* mobility, such that IP and MCP joints have flexion/extension, whereas CMC joint has both flexion/extension and abduction/adduction *DoFs*. The thumb is crucial to perform meaningful tasks during virtual or real tasks. However, we handle the design of the thumb component as a separate research topic than the rest of the fingers due to the differences in their kinematics.

The literature reports a wide variety of hand sizes within the society [3]. In particular, Buryanov *et al.* analyzed the anatomical variations using right and left hands of 66 adult patients between the ages of 19 and 78 from anterior-posterior X-ray images. As a result, they reported an average size for bone lengths of proximal, medial and distal phalanges as well as metacarpal bone, soft tissue at fingertip and web height from metacarpophalangeal joint. Table 2.1 shows the average size of human fingers for index, middle, ring and little fingers. Since modeling a finger should consider distances between MCP and PIP joints as the length of the first phalange; the size of proximal phalange should be added to web height. Table 2.2 also shows the ratio between the proximal and the middle phalanges over distal phalanges using the data gathered by Buryanov *et al.*.

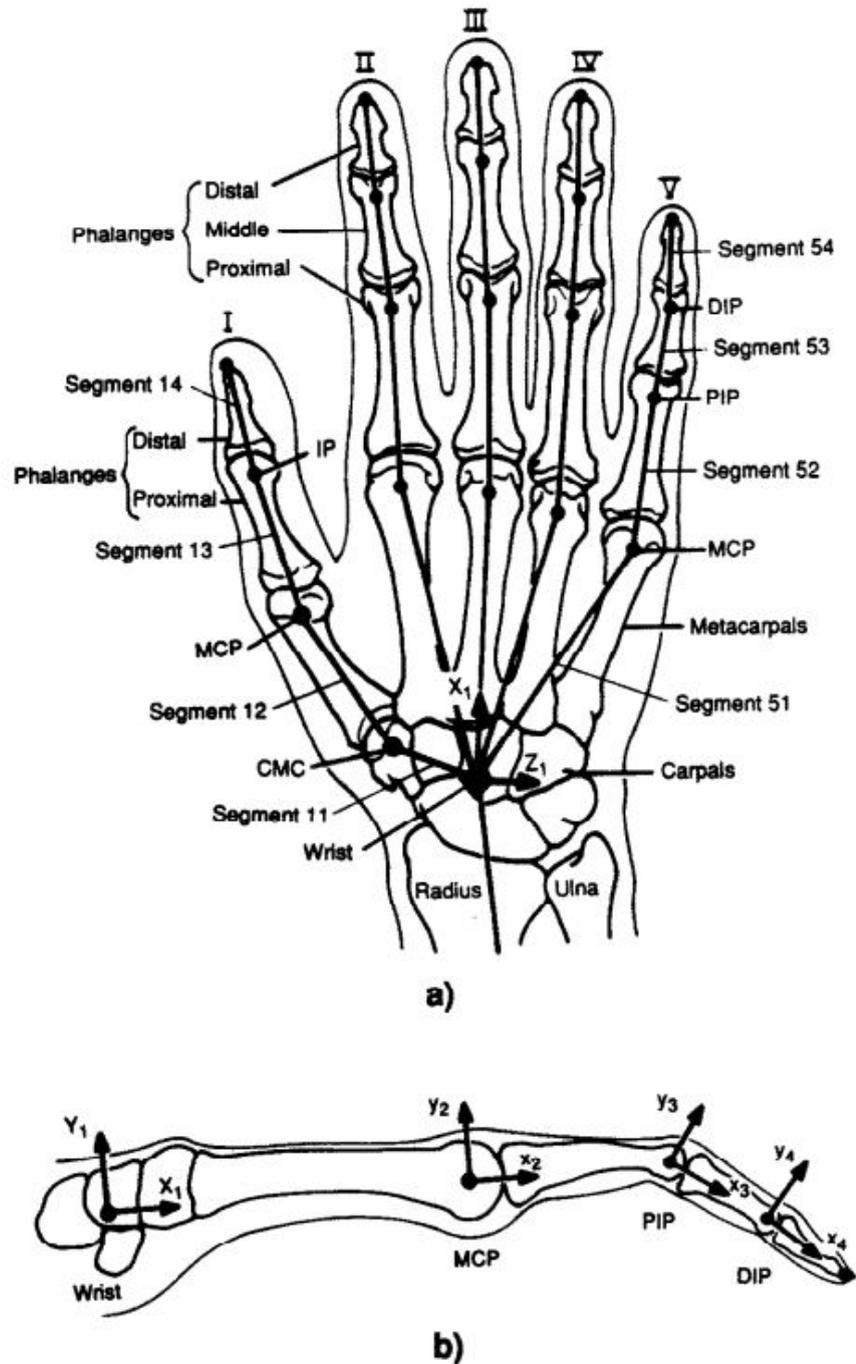


FIGURE 2.1: Kinematic model of the human hand [1]. (a) dorsal view of the hand shows the definitions for the thumb (I), the index (II), the middle (III), the ring (IV), and the little fingers (V). The thumb has two phalanges (proximal, and distal), while other fingers have three (proximal, middle, and distal) from the palm to the fingertip. (b) ulnar view of the index finger as an example for finger kinematics. MCP, PIP and DIP joints are shown from the palm to the fingertip.

TABLE 2.1: Average finger phalange sizes for index, middle, ring and little fingers [3].

Finger	Web Height	Proximal P.	Middle P.	Distal P.
Index Finger	15.52	$39.78 \pm 4.94$	$22.38 \pm 2.51$	$15.82 \pm 2.26$
Middle Finger	15.33	$44.63 \pm 3.81$	$26.33 \pm 3.00$	$17.40 \pm 1.85$
Ring Finger	18.49	$41.37 \pm 3.87$	$25.65 \pm 3.29$	$17.30 \pm 2.22$
Little Finger	24.72	$32.74 \pm 2.77$	$18.11 \pm 2.54$	$15.96 \pm 2.45$

TABLE 2.2: Average ratios between proximal and middle phalanges over distal phalange for the index, middle, ring, and little fingers [3].

Finger	Middle/Distal Phalanges	Proximal/Distal Phalanges
Index Finger	1.4	2.5
Middle Finger	1.5	2.6
Ring Finger	1.5	2.4
Little Finger	1.1	2.1

Similarly, Becker *et al.* [4] studied natural range of motion (RoM) for the flexion/extension movements of MCP, PIP and DIP joints using a video image analyzer. The findings of Becker *et al.* has been summarized in Table 2.3.

TABLE 2.3: Ranges of motion (RoM) for finger joints during flexion/extension: means (standard deviations) in degrees [4].

Finger	MCP	PIP	DIP
Index Finger	70.83 (11.09)	103.87 (7.79)	61.17 (12.71)
Middle Finger	85.30 (9.87)	103.98 (8.98)	73.64 (16.30)
Ring Finger	85.09 (14.46)	107.15 (13.49)	66.96 (15.77)
Little Finger	85.58 (18.09)	98.95 (11.20)	70.79 (15.84)

Even though anatomical studies regarding hand sizes and the RoMs were stated independently, Buchholz *et al.* [1] presented a simulation study where relation between finger sizes and required joint angles while grasping cylindrical object, which shows that required RoM to perform the single task might vary based on hand sizes.

## 2.2 Possible Applications for Hand Exoskeletons

As briefly mentioned previously, a hand exoskeleton can be used for specific desired tasks, user profiles and external factors during different applications, such as rehabilitation, assistive and haptic use, as shown in Figure 2.2.



FIGURE 2.2: Hand exoskeletons can be used to lead, assist or resist user’s hand movements for: (a) rehabilitation use, (b) assistive use and (c) haptic use.

**Rehabilitation exoskeletons** are designed to treat disabilities of patients in a clinical setting (see Figure 2.2(a)). They repeat mimicking the most common activities of daily living (ADL) by opening/closing the fingers. They must be easily wearable not to cause discomfort or pain for the patients during preparation. They should apply high output forces, and monitor finger movements for performance evaluation. They might allow patients to interact with real objects to increase the realism of therapy exercises. Instant adjustability for different tasks also is favorable, even though patients with severe disabilities would not take advantage of the task variety due to the loss of isolated individual finger movement after injury [39]. Their portability is not mandatory especially for clinical devices, but still preferable.

**Assistive exoskeletons** are designed to assist patients with hand disabilities in performing ADLs, such as grasping a cup while drinking coffee, or holding a key while opening the door (see Figure 2.2(b)). Instant adjustability for different tasks, easy wearability and portability are highly important for assistive exoskeletons. They must allow patients to interact with real objects, and apply high output forces. Finger tracking is neither mandatory, nor favorable.

**Haptic exoskeletons** are designed for healthy subjects to interact with objects in a virtual environment (see Figure 2.2(c)). Instant adjustability for different tasks, portability and efficient finger tracking are highly important for haptic exoskeletons. Since the target user profile is assumed to be healthy, the wearability or high output forces are not mandatory, but favorable.

## 2.3 Hand Exoskeleton Literature

The complex anatomy of the human hand and specific requirements for each applications have previously leaded designers to implicitly select a single use and desired task, and to simplify the mechanical design of the exoskeleton. Such differences among the possible target applications increase the need to create a categorization factor while investigating the devices in the literature. State-of-the-art exoskeletons (see Table 2.4, Table 2.5 and Table 2.6) will be investigated based on other properties, such as the number of actuated fingers, number of active and passive joints for each finger, number of connection points, the type of finger pose estimation, the actuation type, the mechanism placement type and the control algorithms, as summarized in Figure 2.3.

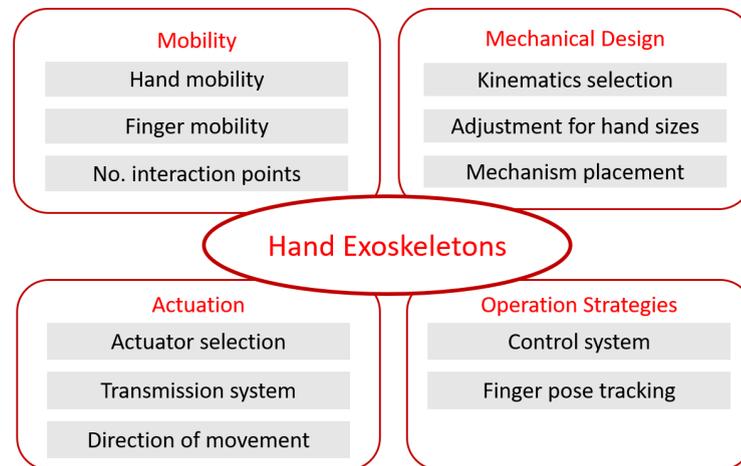


FIGURE 2.3: Hand exoskeletons should be categorized by design selections that can be categorized under design aspects: mobility, mechanical design, actuation and operational strategies.

The very first step of designing a hand exoskeleton from scratch should be listing the design criteria for the focus of application. In particular, these design criteria can be formed only by investigating the advantages and disadvantages of each classification category and find out the most appealing option for the future device. With this motivation, this section combines the hand exoskeleton detailed in the previous section under all possible classification category to study all the possible alternatives.

TABLE 2.4: **5 fingered hand exoskeletons**: Main application (rehabilitation (R) / assistive (A) / haptic (H)), number of independent fingers, number of assisted and independent DoF for each finger, mechanism type, device placement (dorsal (DOR)/ palmar (PAL)/ lateral (LAT)), actuator, control modes (position (POS)/ velocity (VEL)/ backdriveable (BAC)/ admittance (ADM)/ impedance (IMP)/ EMG triggered (EMG)), pose estimation method (encoder (ENC)/ flex sensor (FLE)/ motion tracking (MT)/ additional sensor (SEN))

Device	Applic.	Ind. Fing.	Fing. DoF (act.)	Mechanism	Placement	Actuation	Control	Fing. pose
Tong et al. [36, 40, 41]	R	5	2(1)	Link (coupled)	DOR	DC (Lin)	BAC, EMG	ENC
HEXOSYS [42, 43]	R	5	4(1)	Link (fingertip)	DOR	DC (Rot)	POS	SEN
SAFE [44, 45]	H,R	5	3(1)	Link (fingertip)	DOR	DC (Rot)	ADM	SEN
Fu et al. [46]	R	5	4(1)	Link (coupled)	DOR	DC (Rot)	POS	SEN
HANDEXOS [47, 48]	R	5	4(1)	Link (coupled)	DOR	DC (Rot)	BAC	SEN,MT
Rahman et al. [49]	R	5	3(1)	Link (coupled)	DOR	DC (Lin)	POS	ENC
Jo et al. [50, 51]	H	5	3(1)	Link-glove	DOR	DC (Lin)	VEL	FLE
PneuGlove [52]	H,R	5	3(1)	Glove	DOR, PAL	Pneumatic	POS, BAC	FLE
Fang et al. [53]	H	5	3(1)	Link (fingertip)	DOR	DC (Rot)	ADM	ENC
Delph et al. [54]	R	5	3(1)	Glove	DOR, PAL	Servo	POS, EMG	-
Polygerinos et al. [55]	R	5	3(1)	Link (compliant)	DOR	Pneumatic	POS	-
Cui et al. [56]	R	5	3(1)	Link (coupled)	DOR	DC (Lin)	POS	-
Kim et al. [57]	R	5	3(2)	Link (coupled)	DOR	DC (Rot)	POS	ENC
Decker et al. [58]	R	5	2(1)	Link-glove	DOR	DC (Rot)	BAC	FLE
Jo et al. [59]	R	5	2(1)	Link-glove	DOR	DC (Lin)	POS	ENC
Yap et al. [60, 61]	A,R	5	3(1)	Compliant	DOR	Pneumatic	POS	-
Lu et al. [62]	R	5	2(1)	Link (coupled)	DOR	DC (Rot)	-	ENC
Sarac et al. [29, 30, 63, 64]	R,A,H	5	2(1)	Link (underact.)	DOR	DC (Lin)	POS, ADM	SEN
Hasegawa et al. [65, 66]	A	3	3(3)	Link (indep)	DOR, LAT	DC (Rot)	POS, ADM	ENC
Burton et al. [67]	R	3	3(2)	Link (coupled)	DOR	Pneumatic	-	ENC,MT
Ferguson et al. [68]	R	3	4(2)	Link (coupled)	DOR	DC (Rot)	POS, ADM	-
BiomHED [69, 70]	R	2	4(3)	Link-glove	DOR, PAL	DC (Rot)	ADM	MT
BRAVO Hand [71, 72]	R	2	3(1)	Link (mitten)	DOR	DC (Rot)	POS, EMG	ENC
Mulas et al. [73]	R	2	3(1)	Link-glove	DOR, LAT	Servo	EMG	ENC
HANDSOME [74]	R	2	1	Link (indep)	DOR	Spring	-	ENC
Exophalanx-2 [75]	H	2	3(2)	Link-glove	DOR, LAT	SMA	ADM	ENC
Li et al. [76]	R	1	3(1)	Glove	DOR	Pneumatic	POS	FLE

TABLE 2.5: **4, and 3 fingered hand exoskeletons:** Main application (rehabilitation (R) / haptic (H) / assistive (A)), number of assisted and independent fingers, number of assisted and independent DoF for each finger, mechanism type, device placement (dorsal (DOR) / palmar (PAL) / lateral (LAT)), actuator, control modes (position (POS) / velocity (VEL) / backdriveable (BAC) / admittance (ADM) / impedance (IMP) / EMG triggered (EMG)), pose estimation method (encoder (ENC) / flex sensor (FLE) / motion tracking (MT) / additional sensor (SEN))

Device	Applic.	Ind. Fing.	Fing. DoF (act.)	Mechanism	Placement	Actuation	Control	Fing. pose
<b>Rutgers Master</b> [77]	H	4(4)	4(1)	Link (fingertip)	PAL	Pneumatic	POS	SEN
<b>Popov et al.</b> [78]	A	4(4)	3(1)	Glove	DOR	Pneumatic	POS	FLE
<b>Allotta et al.</b> [79]	A	4(4)	3(1)	Link (coupled)	DOR	Servo	POS	ENC
<b>Wu et al.</b> [80]	R	4(1)	2(2)	Link (mitten)	DOR	Pneumatic	POS	SEN
<b>Weiss et al.</b> [81]	R	4(1)	4(1)	Link (coupled)	DOR,LAT	DC (Rot)	POS	SEN
<b>Wei et al.</b> [82]	R	4(1)	2(1)	Link (mitten)	DOR,LAT	DC (Rot)	-	FLE
<b>Arata et al.</b> [83, 84]	A	4(1)	3(1)	Link (compliant)	DOR	DC (Lin)	POS, BAC	ENC,MT
<b>Gasser et al.</b> [37, 85]	R,A	4(1)	2(1)	Link (mitten)	DOR,LAT	DC (Rot)	POS	ENC
<b>Lince et al.</b> [86]	R,A	4(1)	3(1)	Link (mitten)	DOR	DC (Rot)	EMG	ENC
<b>Wei et al.</b> [87]	R,A	3(3)	3(2)	Link (coupled)	DOR	DC (Rot)	IMP	ENC
<b>Exophalanx</b> [88]	H	3(3)	3(1)	Link-glove	DOR	SMA	ADM	FLE,MT
<b>Ryu et al.</b> [89]	H	3(3)	3(1)	Link (compliant)	DOR	Pneumatic	POS, BAC	FLE
<b>Sarakoglou et al.</b> [90]	H	3(3)	4(1)	Link (fingertip)	DOR	DC (Rot)	ADM	SEN
<b>In et al.</b> [91, 92]	A	3(1)	3(1)	Glove	DOR	DC (Rot)	ADM	ENC
<b>Geo et al.</b> [93]	R	3(1)	3(1)	Link (coupled)	DOR	DC (Rot)	-	ENC

TABLE 2.6: **2, and 1 fingered hand exoskeletons:** Main application (rehabilitation (R) / haptic (H) / assistive (A)), number of assisted and independent fingers, number of assisted and independent DoF for each finger, mechanism type, device placement (dorsal (DOR) / palmar (PAL) / lateral (LAT)), actuator, control modes (position (POS) / velocity (VEL) / backdriveable (BAC) / admittance (ADM) / impedance (IMP) / EMG triggered (EMG)), pose estimation method (encoder (ENC) / flex sensor (FLE) / motion tracking (MT) / additional sensor (SEN))

Device	Applic.	Ind. Fing.	Fing. DoF	Mechanism	Placement	Actuation	Control	Fing. pose
<b>Cempini et al. [94,95]</b>	R	2(2)	4(1)	Link (f.coupled)	DOR,LAT	DC (Rot)	VEL	ENC
<b>FINGER [96]</b>	R	2(2)	2(2)	Link (indep)	DOR	DC (Lin)	POS, BAC	ENC
<b>WHIPFI [97]</b>	H	2(2)	3(1)	Link (fingertip)	DOR	DC (Rot)	ADM	SEN
<b>Fontana et al. [98]</b>	H	2(2)	4(3)	Link (coupled)	DOR	DC (Rot)	-	ENC
<b>iHandRehab [99]</b>	R	2(2)	4(4)	Link (indep)	DOR	DC (Rot)	POS, BAC	ENC
<b>Fiorilla et al. [100]</b>	R	2(2)	1	Link (MCP)	DOR	DC (Rot)	-	SEN
<b>Wang et al. [35]</b>	R	1(1)	4(4)	Link (indep)	DOR	DC (Rot)	POS, BAC	ENC
<b>Yamaura et al. [101]</b>	R	1(1)	3(2)	Link (coupled)	DOR	Servo	POS	ENC
<b>Jones et al. [102,103]</b>	R	1(1)	3(3)	Link (indep)	DOR,LAT	DC (Rot)	POS, BAC	ENC
<b>Polotto et al. [104]</b>	R, A	1(1)	4(4)	Link (indep)	DOR,LAT	DC (Rot)	POS, BAC	ENC
<b>Tang et al. [105,106]</b>	R	1(1)	2(1)	Link (coupled)	DOR	Ultrasonic	POS, BAC	ENC
<b>DiCicco et al. [107]</b>	A	1(1)	3(1)	Link (coupled)	DOR	Pneumatic	EMG	ENC
<b>Wege et al. [108-110]</b>	R	1(1)	4(1)	Link (coupled)	DOR	DC (Rot)	POS, EMG	SEN
<b>Sun et al. [111]</b>	R	1(1)	4(1)	Link (fingertip)	DOR	Pneumatic	POS, ADM	MT
<b>Agarwal et al. [112]</b>	R	1(1)	4(3)	Link (indep)	DOR	Pneumatic	-	SEN
<b>Agarwal2 et al. [113]</b>	R	1(1)	3(2)	Link (coupled)	DOR	Elastic	IMP	ENC
<b>AssistOnFinger [114, 115]</b>	R	1(1)	3(1)	Link (underac.)	DOR	DC (Rot)	POS,BAC,EM	-
<b>Aubin et al. [116]</b>	R	1(1)	2(2)	Link-glove	DOR	Servo	POS, BAC	ENC
<b>Lee et al. [117]</b>	R	1(1)	3(1)	Link (coupled)	DOR	DC (Rot)	-	SEN
<b>Maeder York et al. [118]</b>	R	1(1)	3(1)	Link (compliant)	DOR	Pneumatic	-	ENC

### 2.3.1 Mobility

Mobility assisted by an exoskeleton can be handled in terms of hand mobility, finger mobility and the number of interaction points between the mechanism and user's finger (see Figure 2.4). Both hand and finger mobility can be categorized further based on the number of assisted and independently controlled mobility.

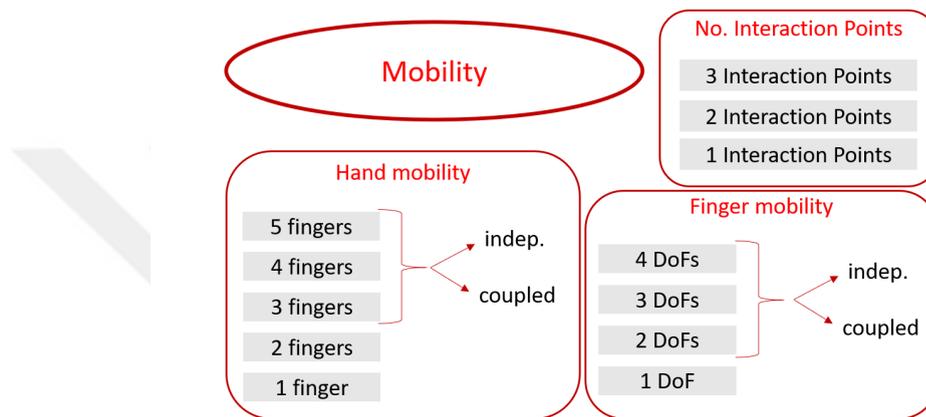


FIGURE 2.4: Possible design choices for mobility based on hand mobility, finger mobility and number of interaction points.

#### 2.3.1.1 Hand mobility

A human hand has 5 fingers, and an exoskeleton can be designed to assist and control various numbers of fingers. Finger exoskeletons [35, 101, 102, 104, 105, 107, 108, 111, 112, 114, 116–118] are designed mostly for the index finger, and are mostly stated as an initial study for a multi-finger exoskeleton. 2-finger exoskeletons control thumb and index finger independently, and support only specific hand movements for rehabilitation or haptics, such as finger tapping or pick-and-place tasks [66, 94–100, 119].

Even though exoskeletons with 1 or 2 fingers are simpler to implement, most of ADLs require at least 3 fingers to be assisted. One approach to design multi-finger exoskeletons is to control each finger component individually. 5-finger exoskeletons control each finger independently, and can be used for all applications with minimum constraints [29, 36, 42, 44, 46, 47, 49, 50, 52–60, 62, 67]. Since the middle, ring and little fingers of a healthy person are highly coupled, 4-finger exoskeletons, which control thumb, index, middle and ring fingers [77, 78], or 3-finger exoskeletons, which control thumb, index and middle

fingers [87–90], can be used all applications. Even though these devices can assist users during all ADLs, the perception of realism would drop as the number of assisted fingers decrease. On the other hand, 4-finger exoskeletons, which control index, middle, ring and little fingers, cannot be effective for grasping or picking tasks during assistive or haptic applications due to the lack of resistive forces acting on the objects through the thumb [79].

Increasing the number of assisted fingers improves the overall mobility while complicating the design. The second approach to design multi-finger exoskeletons is to couple finger movements through mechanical [37, 65, 67–69, 71, 73–75, 80, 82, 86] or differential [76, 81, 83, 85, 91, 93] systems.

Even though we cannot claim that moving finger components together prevents the exoskeleton to be used for certain applications, it limits certain tasks. For instance, a 5-finger exoskeleton with coupled index, middle, ring and little fingers can assist users grasping objects only in certain shapes (e.g. a water bottle) during assistive or haptic applications, but not a key. This is why a generic hand exoskeleton should control 4 or 5 fingers independently.

A generic exoskeleton should assist user’s natural finger movements. The designer can choose to independently control 5 fingers or 4 fingers, while the little finger is either left free or coupled with the ring finger. In particular, the anatomic coupling between the ring and little fingers would allow designers to simplify the mechanical system without sacrificing the natural hand movements.

### 2.3.1.2 Finger mobility

A human finger has 4 *DoF* mobility, and an exoskeleton can be designed to assist and control various numbers of finger joints for each finger. 1 *DoF* mechanisms [74, 100] only flex/extend MCP joint for repetitive rehabilitation exercises and enhanced motor learning. Even though finger components with 1 *DoF* mobility are simpler to implement and easier to be worn, most of ADLs require at least 2 DoF to be assisted for each finger. One approach to design multi-DoF mechanisms is to control each finger joint individually with 2 *DoF* [80, 96, 116], 3 *DoF* [65, 102] or 4 *DoF* [35, 99, 104] mobility.

Increasing the number of assisted joints improves the overall mobility while complicating the design. The second approach to design multi-DoF mechanisms is to couple finger



FIGURE 2.5: Hand exoskeletons can be designed with different numbers of interaction points between the device and user's fingers. Multiple interaction points improve grasping stability, user's safety and perception of touch but are harder to be worn.

joints through mechanical or differential systems. Towards simplifying the finger components, the first step can be leaving the abduction/adduction of MCP joint passive [112], or neglected completely, since most of the ADLs focus on finger opening/closing. Even then, controlling 3 *DoF* flexion/extension independently can be challenging. As the second simplification step, DIP and PIP joints can be coupled with a mechanically adjustable ratio, while MCP joint is controlled independently [57, 67, 68, 75, 87, 88, 95, 98, 101, 113]. Since DIP and PIP joints are anatomically coupled, this simplification does not affect the perception significantly, but coupling them with a constant ratio might limit certain finger synergies.

Finally, a mechanism can be designed with a single actuator to control finger opening/closing through 4 *DoF* [42, 46, 47, 69, 77, 81, 90, 94, 108, 111], 3 *DoF* [44, 49, 50, 52–56, 60, 71,

73,76,78,79,83,86,88,89,91,93,97,107,114,117,118] or 2 *DoF* [29,36,37,58,59,62,82,105] mobility. Such coupling can be set by a constant ratio through mechanical linkages or differential systems, or by adjusting the transmitted forces automatically based on contact forces [121].

Even though we cannot claim that moving finger joints together prevents the exoskeleton from being used for certain applications, it limits certain tasks. For instance, a 3 *DoF* mechanism with constant ratio can assist users grasping objects in certain shapes (e.g. a water bottle) during assistive or haptic applications, but not a phone without having mechanical adjustments. This is why a generic hand exoskeleton should flex/extend 2 or 3 finger joints independently, or coupled based on contact forces. Compared to fully controlled mechanisms, underactuated systems based on contact forces are mechanically simpler and cheaper, but require more complicated operational strategies.

A generic exoskeleton should allow finger joints to flex/extend in different synergies, based on different tasks. The designer can passively abduct/adduct MCP joint, since it does not significantly change the task performance during ADLs. Furthermore, the designer might focus on flexion/extension of MCP and PIP joints only, since the natural coupling between DIP and PIP joints would cause the DIP joints to move accordingly even without assistance. The designer can choose to achieve 2 *DoF* or 3 *DoF* mobility for each finger either by controlling them independently, or by coupling them using strategies to adjust for different tasks.

### 2.3.1.3 Number of interactions

A human finger has 3 phalanges, and an exoskeleton can be designed to interact with various numbers of phalanges to transmit actuator forces and to rotate finger joints. The number of interactions mostly depends on finger mobility. One approach to design finger components is to choose the same number of interaction points as the number of *DoF*. In other words, an exoskeleton can be designed with 4 *DoF* and 3 interaction points [35, 46, 47, 69, 81, 94, 99, 104, 108, 112], 3 *DoF* and 3 interaction points (Figure 2.5(a)) [49, 50, 52, 54, 57, 60, 65, 67, 71, 73, 75, 76, 78, 79, 83, 86–89, 91, 93, 101, 102, 107, 113, 114, 117, 118], 2 *DoF* and 2 interaction points (Figure 2.5(b)) [29, 36, 58, 59, 62, 80, 82, 96, 105, 116] or 1 *DoF* and 1 interaction point (Figure 2.5(c)) [74, 100].

Devices with multiple interactions enhance the grasping stability during assistive and rehabilitation, and improve the haptic perception. Furthermore, they improve patients'

safety by strictly limiting the spasticity. However, they might suffer from the design complexity of choosing high finger mobility. Mechanisms with 2 interaction points can achieve 3 *DoF* [56,95] or 4 *DoF* [98] finger mobility. Alternatively, fingertip devices can achieve 2 *DoF* [37], 3 *DoF* (Figure 2.5(d)) [44,53,55,97] or 4 *DoF* [42,68,77,90,111] mobility. Even though having less number of interaction points simplifies the device mechanically, they might fail to reflect realistic interactions for certain haptic or assistive applications. For instance, a fingertip device can allow users to interact with objects and apply event-based forces, but cannot apply grasping forces on finger phalanges realistically.

Even though having less number of interaction points have simpler design and are easier to be worn, a generic exoskeleton should be designed with the same number of interaction points as the number of *DoF*. The number of interaction points between a generic exoskeleton and user's fingers should be decided according to the number of finger mobility.

### 2.3.2 Mechanical Design

Towards creating a hand exoskeleton, the next step of the designer should be how to achieve the mobility decisions through mechanical design. The mechanical design aspect can be handled based on kinematics selection, mechanical placement, and adjustment strategies for different hand sizes (see Figure 2.6).

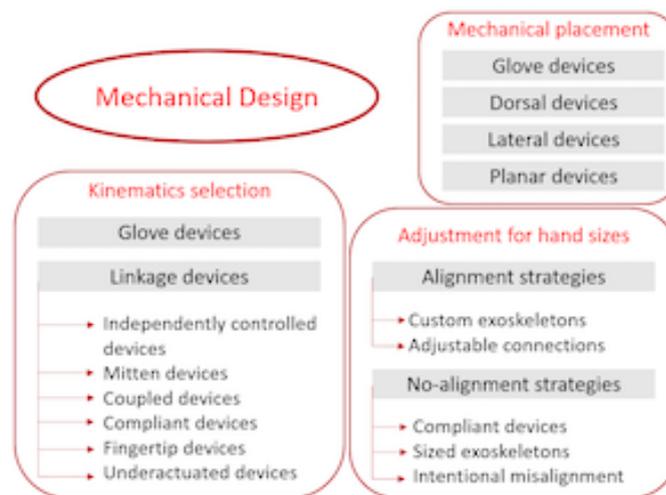


FIGURE 2.6: Possible design choices for mechanical design based on kinematics selection, mechanical placement and adjustment strategies for hand sizes.

### 2.3.2.1 Kinematics selection

The kinematics structure of a hand exoskeleton can be handled as glove-based or linkage-based devices. Figure 2.7 shows examples of exoskeletons with different kinematics selections.

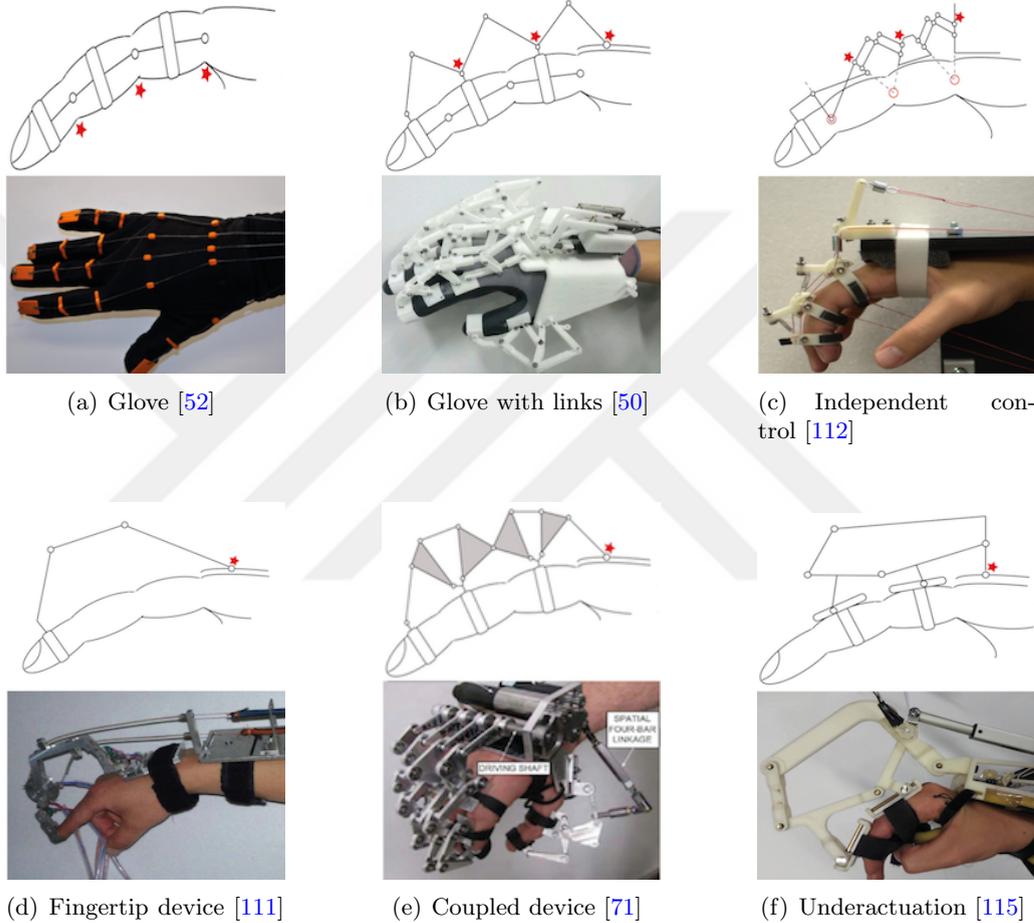


FIGURE 2.7: Types of kinematics selections as hand exoskeletons: the black circles show the mechanical joints, while the red stars represent the actuated ones. Glove-based devices can track finger pose easily and efficiently but are hard to be worn. Linkage-based devices are lightweight, portable and easily wearable. Linkage-based devices can be categorized based on the finger mobility choices detailed in Section 2.3.1.2.

**Glove-based devices** require the user to wear a flexible glove equipped with sensors for motion tracking, and are perfect for haptic applications. They can assist/resist user's activity through cable transmission (Figure 2.7(a)) [52, 54, 76, 78, 91], or linkage transmission (Figure 2.7(b)) [50, 58, 59, 69, 73, 75, 88, 116]. Even though their wearability

can be improved using Velcro connections in the palm [58] or half gloves [59,73], patients still have to reach an initial pose to wear the glove.

**Linkage-based devices** use mechanical links to form the finger components, and can be further categorized with independent joint control, MCP rotation only, full coupling, partial coupling, mitten style, fingertip connection, compliance and contact based underactuation. Devices with independent control have an individual actuator for each assisted finger joint (Figure 2.7(c)) [35, 65, 74, 96, 99, 100, 102, 104, 112]. These actuators are mostly placed remotely and their forces are transmitted through cables. Even though they can achieve full mobility, increasing the number of actuators significantly affects their cost and portability.

Linkage-based devices can be simplified in terms of the number of actuators with different kinematical structures. Mitten devices open/close the hand in a unique, repetitive way by coupling index, middle, ring and little fingers physically [37, 71, 80, 82, 86]. Controlling the hand with 1 or 2 actuators simplifies the design and decreases the overall cost, but limit the mobility and task adjustability.

Coupled devices interact with user's finger from multiple points and move finger joints together with a ratio adjusted by mechanical links or differential system (Figure 2.7(e)). Such mechanisms can control finger movements with 1 actuator [36, 46, 47, 49, 56, 62, 79, 81, 93, 94, 105, 107, 108, 117] or 2 actuators [57, 67, 68, 87, 95, 98, 101, 113]. Compliant devices couple finger joints through compliant elements [83], artificial muscles [89] or soft actuators [55, 60, 118] instead of rigid links. Their coupling ratio is set by the mechanical stiffness of these soft elements. They are low-cost, but suffer from mandatory mechanical adjustments to change the finger synergies.

Unlike coupled devices, fingertip devices interact with user's finger from a single point and control the fingertip position regardless how finger joints move (Figure 2.7(d)) [42, 44, 53, 77, 90, 97, 111]. Each finger component is controlled using a single actuator, so they are low-cost, easily wearable and portable. Not having strict mechanical connections around every finger phalange allows users to adjust tasks within the limits of their abilities. However, they cannot impose strict finger synergies, limit spastic movements for patients with disabilities or convey realistic information about virtual interactions.

Finally, underactuated devices based on contact forces control multiple finger joints with a single actuator by adjusting forces acting on finger phalanges automatically based on interaction forces, thanks to passive elements along the mechanism (Figure 2.7(f)) [29, 114]. Each finger component is controlled using a single actuator, so they are low-cost,

lightweight and portable. Passive elements along the mechanism ensures the device to be worn easily. Even though the actuator does not control the joints implicitly, alternative control strategies can improve the trajectory following tasks because they have multiple interactions for each finger (see Section 2.3.4).

The kinematics of a generic exoskeleton should be consistent with the desired finger mobility. Full finger mobility can be achieved with linkage-based devices with independent control. Alternatively, finger joints can be coupled with underactuated linkage-based devices based on contact forces. Doing so, a single actuator controls a single finger component while adjusting the operation for different tasks automatically.

Kinematics selection should be made based on mobility. The designer can adopt glove-based or linkage-based exoskeletons for independently controlled finger components. Despite their bulky and expensive design, they will achieve high performance for strict trajectory following tasks. Furthermore, mechanical and finger joints must be aligned carefully to ensure user's safety and efficacy of applied forces.

### 2.3.2.2 Strategies for adjusting to different hand sizes

The society has a wide range of hand sizes [3], and a hand exoskeleton should operate correctly and comfortably for all users [122]. Exoskeletons with a single interaction point [90, 97, 100] can neglect such variety, since they control the fingertip pose without imposing strict trajectory for finger joints. For exoskeletons with multiple interaction points, several adjustment strategies can be found in the literature:

**Alignment strategies** require mechanical and finger joints to be aligned, such that the exoskeleton can fit on user's hand accurately, and actuator forces can be mapped into perceived ones directly. The first alignment strategy is to manufacture a custom exoskeleton for each user individually [54, 56, 65, 67, 81]. A custom exoskeleton must be designed with variable link lengths corresponding to user's hand size. Such an exoskeleton must be manufactured individually, so the user must agree to purchase it for personal use. Due to the lack of mass production, the overall cost of the device is expected to be high. Even though this strategy might be suitable for assistive or haptic applications, it is not applicable for clinical use, where a single device is expected to serve for multiple patients in a day.

Alternatively, an exoskeleton can align mechanical and finger joints through adjusted mechanical connections and links [35, 36, 42, 46, 53, 58, 74, 77, 78, 82, 87, 93, 96, 99, 102, 104,

107, 108, 112, 113, 116]. The user wears the device before operation and a technician fixes a slider-screw system for fitting. Even though it requires a crucial preparation process, the exoskeleton can fit all users in the end. The constant need for a technician's presence might make such an exoskeleton suitable for clinical settings more than home therapy.

**No-Alignment Strategies** accept the misalignment between mechanical and finger joints, and address the issue of hand sizes in other ways. Increasing the compliance of the actuator [52, 55, 60, 76, 80, 89, 91] or the mechanical links [50] transmits lower interaction forces, hence minimizes the after effects of misalignment. However, the output forces might be insufficient for certain rehabilitation or assistive applications.

A hand exoskeleton can be designed in small, medium and large sizes, such that the misalignment between mechanical and finger joints can be limited [37, 49, 59, 62, 69, 73, 83, 88, 111]. Even though misalignment are not prevented, they are ensured not to harm users. Finally, a designer can place passive joints along the mechanical structure to turn additional loads, which are caused by misalignment, into motion [29, 44, 47, 57, 68, 71, 79, 86, 94, 95, 98, 101, 105, 115, 117]. Such an exoskeleton adapts its behavior for different hand sizes automatically. Designing sized exoskeletons and passive joints are the best practices for generic exoskeletons, thanks to their usability and preparation time. Furthermore, since they can be mass produced, they can be low-cost.

The designer can couple finger joints using contact based underactuation, such that a single actuator moves multiple finger joints while passive elements adjust the operation based on interaction forces acting on finger phalanges. Thanks to the automatic adjustability, the mechanism can be simplified significantly. Furthermore, passive elements ensure users' safety during operation. However, they require complex control strategies to achieve high tracking performance.

### 2.3.2.3 Mechanism Placement

Finally, the designer should device where to place the finger components with respect to the fingers. This design selection is especially important for linkage-based exoskeletons, such that transmission units can be placed on dorsal, lateral or palmar side of fingers (see Figure 2.8).

**Palmar devices** consist of mechanical or transmission components placed inside the palm of the hand (see Figure 2.8(a)) [52, 54, 69, 77]. Unfortunately, they prevent users to get in touch with real objects for assistive use.

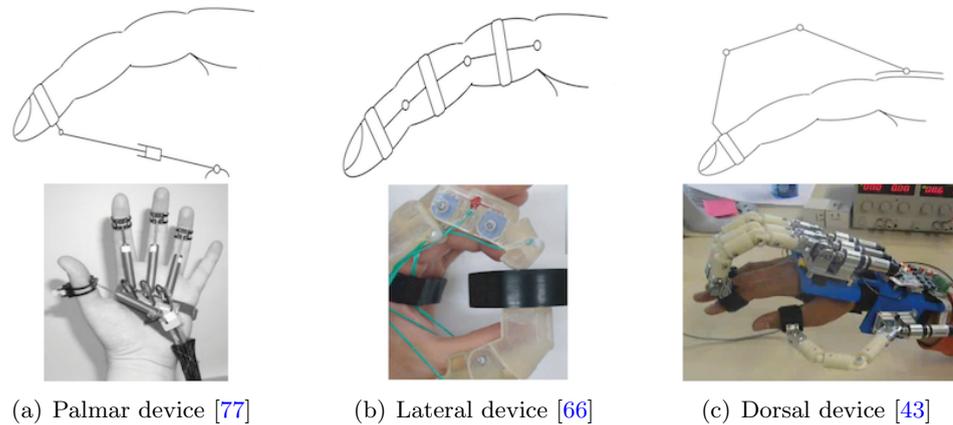


FIGURE 2.8: Hand exoskeletons can be designed to be placed on different locations with respect of fingers.

**Lateral devices** consist of mechanical or transmission components placed on both sides of finger phalanges (see Figure 2.8(b)) [37, 65, 73, 81, 82, 94, 95, 102, 104]. Finger joints can be rotated independently through cable transmission or remote center of motion (RCM). These devices free the palm of the hand for future interactions in the real environment. However, they might suffer from possible collisions for multi-finger implementations, especially when abduction/adduction of MCP is allowed. Compared to other options, lateral devices might be harder to be worn by patients with disabilities, so their use for rehabilitation or assistive should be reconsidered.

**Dorsal devices** consist of mechanical or transmission components placed on top of the finger phalanges (Figure 2.8(c)) [29, 35–37, 42, 44, 46, 47, 49, 50, 52–60, 62, 65, 67–69, 71, 73–76, 78–81, 83, 86–88, 90, 91, 93, 94, 96–102, 104, 105, 107, 108, 111–113, 116–118]. Doing so, the collision between multiple finger components can be minimized while user’s palm is free for future interactions with real objects. They do not possess any strong limitation regarding the number of finger components to be manufactured or the performance, and can be used for all possible target applications.

Regardless the placement of finger components, linkage-based exoskeletons are attached to user’s fingers through rings or flexible attachments. Since there is no recorded impact of mechanism placement on perception during finger opening/closing, we can assume that actuator forces can be distributed around finger phalanges naturally.

### 2.3.3 Actuation

An exoskeleton can assist/resist user's fingers through actuator and transmission technologies. In this section, we will investigate the exoskeletons in the literature based on actuator selection, direction of movement and transmission system from the perspective of achieving generic exoskeletons (see Figure 2.9).

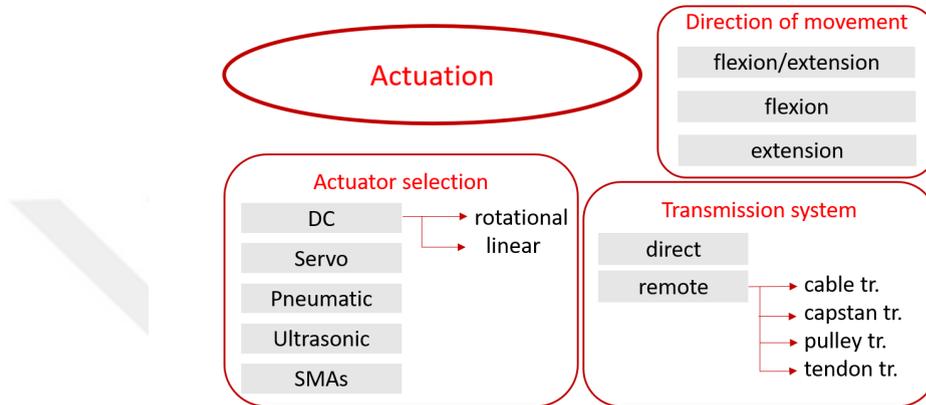


FIGURE 2.9: Possible design choices for actuation technologies based on actuator selection, transmission system and direction of movement.

#### 2.3.3.1 Actuator type

Even though there are some exceptional studies that apply assistance based on wrist activity [123] or resistance using springs [74], most of the exoskeletons move user's fingers through active manipulation. Such a manipulation can be achieved through different actuator types.

**DC motors** are the most popular technology since they are highly available in the market, reliable and easily controllable. Linear movement can be achieved using linear DC motors [29, 36, 49, 50, 56, 59, 83, 96] or rotational DC motors with linear sliders [47, 95]. Then, rotational movement can be achieved using brushed motors [35, 42, 44, 46, 57, 58, 62, 65, 68, 69, 71, 78, 78, 81, 87, 90, 93, 94, 97–100, 102, 104, 108, 114, 117] or brushless motors [37, 53, 82, 86, 91]. Linear motors are simpler to be placed on top of the hand for coupled finger opening/closing, while rotational motors are mostly backdriveable and provide unlimited movement. Furthermore, brushed motors have low-cost, simple

wiring, compact design and easy control but require maintenance, cause vibration and lose torque in high speeds due to friction.

**Servo motors** can be defined as rotational DC motors with a limited workspace [54, 73, 79, 101, 116]. They are fast, and can achieve high output torque and accurate position control; but require a special driving circuit for control and have higher cost compared to DC motors.

**Ultrasonic motors (USMs)** can also be defined as rotational DC motors powered by ultrasonic vibration [105]. They are silent, light weight and efficient in terms of output force, but they suffer from hysteresis and temperature increase over time.

**Pneumatic actuators** control the hydraulic or air flow through compressors, using pneumatic cylinders [67, 77, 107], air balloons [76], hydraulic pump [89], air bladder [52], flexible thermoplastic fabrics [60], soft actuation [55] or pneumatic artificial muscles [80, 111, 112, 118]. They can achieve high, adjustable force and speed at low-cost. The size of the compressor and its storage lead the exoskeletons to be controlled remotely. Even though pneumatic actuators are not necessarily compliant, they consequently increase the overall compliance as mentioned in Section 2.3.2.2.

**Shape memory alloy actuators (SMAs)** use deformation of materials upon heating and cooling at critical temperatures [75, 88]. Even though they have high power-to-weight ratio, their output motion is hysteresis, highly nonlinear and saturated. As a result, their control is challenging [124].

Actuation types do not possess strong limitation about applications or tasks. Therefore, any actuator type can be selected for a generic exoskeleton as long as they are low-cost, easily controllable and effective in terms of output forces.

### 2.3.3.2 Transmission units

The actuators should be connected to the mechanical structure through alternative transmission strategies. The simplest transmission scenario is designing a direct-drive system, such that the actuators are placed on top of the hand or along the mechanism, while the actuator shafts are attached to mechanical components directly [29, 42, 71, 83, 114]. Even though direct-drive is preferable to improve the portability, the chosen actuators should be highly miniaturized and lightweight.

If the chosen actuators are big and heavy, they should be located away from the exoskeleton and their forces should be transmitted remotely through cables [35, 46, 47, 54, 62, 65, 68, 69, 79, 81, 84, 88, 94, 96, 99, 99, 101, 102, 104, 107, 108, 110, 112, 113, 116], capstan systems [58, 98, 115], tendons [37, 78, 82, 91, 95], or pulleys [93]. Even though choosing big actuators can create high output forces for the exoskeleton, remote transmission limit the workspace of users.

Both transmission strategies can be implemented for all application types. The designers should make the selection based on the actuator decision.

### **2.3.3.3 Direction of movement**

Even though the majority of actuators are bidirectional, certain rotational DC motors and pneumatic motors are not. If the chosen actuator is unidirectional, then the designer should decide how to use them for finger movements. One approach is to assist user's fingers in one direction actively, and to leave the other direction passive. The active assistance can be used either to open the finger for rehabilitation [48, 52, 74, 112], or to close the finger for assistive use [86, 91, 107]. Devices with active flexion cannot be used for haptic use due to the lack of resistive forces, while devices with active extension cannot be used for assistive use due to the lack of assistive forces. This is why leaving one direction passive cannot be chosen for a generic exoskeleton, even though they provide simple and effective solutions for specific tasks.

The second approach is to achieve bidirectional movement using multiple actuators and transmission units [35, 37, 46, 81, 94, 99, 101, 102, 111]. Bi-directional movements can be adopted for all target applications with no specific limitations. Achieving bi-directional movements might make exoskeletons bulkier and more expensive due to the increased number of actuators. Even though choosing bidirectional actuators is the best choice for generic exoskeletons, the designer should equip multiple actuators if unidirectional actuators are chosen for a specific purpose.

### **2.3.4 Operation strategies**

The design of a hand exoskeleton is completed once the mechanical structure is equipped with actuators and transmission units. Then the designer should decide how to control and track user's fingers during operation (see Figure 2.10).

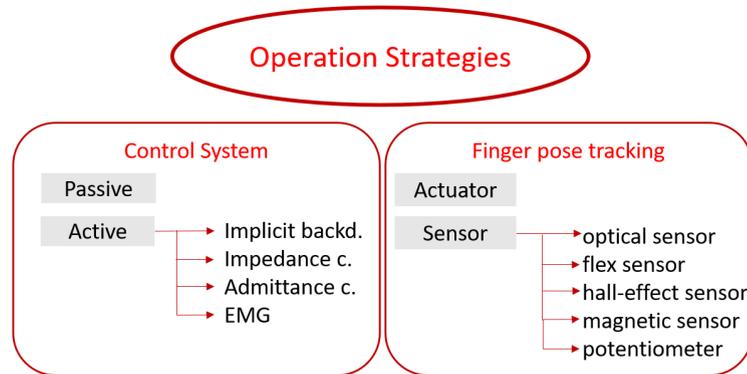


FIGURE 2.10: Possible design choices for operational strategies based on control and finger pose tracking strategies.

### 2.3.4.1 Control

Control strategies for existing hand exoskeletons can be categorized mainly as active and passive, based on how much users participate to the task [125].

**Passive control strategies** control the exoskeleton to follow a strict trajectory or to reach a specific target. As the device leads their fingers, the user is asked to obey the movement. The control strategies can be designed based on position [29, 35, 37, 42, 46, 49, 52, 55–57, 59, 60, 67, 68, 71, 76–81, 83, 89, 101, 108, 111, 119] or velocity [50, 119]. Even though passive exercises can be used to treat disabilities of patients during rehabilitation, they might cause patients to lose interest during long, intense therapy sessions. They can be used for assistive applications as long as they are triggered by an external state, such as an additional sensor or a condition satisfied by an arm exoskeleton. However, they are impractical for haptic use.

**Active control strategies** control the exoskeleton to assist/resist user's fingers based on user's performance as they are in charge of following a trajectory or reaching a target. One way to achieve active control is to adopt implicit backdriveability, which requires actuation, transmission and mechanical units to be chosen accordingly. With implicit backdriveability, the user can move their fingers freely even if the exoskeleton is attached to their fingers with no control [35, 40, 47, 52, 58, 83, 89, 96, 99, 102, 104, 114, 116]. The backdriveable devices can be controlled with passive strategies when the user fails to keep their performance within a predefined range.

Implicit backdriveability cannot be achieved if the mechanical and actuator components of the designed exoskeleton require high backdrive forces or cause high friction. If

so, backdriveability can be achieved actively using force control techniques based on impedance [87, 113] or admittance [44, 50, 63, 68, 75, 88, 90, 92, 97, 111]. These techniques require additional force sensors to be included for the exoskeleton, such that user's intention to move can be measured and be used as a control reference for the exoskeleton. In either case, backdriveability can easily be used by all target applications and improve user's safety during operation.

Furthermore, user's intention can be detected through additional sensors, such as electromyography (EMG) sensors [40, 54, 71, 73, 83, 86, 107, 110] or active bioelectric potential electrodes [66]. Then, these measurements are used to create a control reference for passive control strategies in an online manner. Similarly, bilateral teleoperation tasks can be developed by controlling the device passively to follow the reference set by user's other hand [46, 101]. These assist-as-needed or bilateral control strategies are useful for rehabilitation or assistive applications but their use for haptics is out of context.

A generic hand exoskeleton should be backdriveable with or without control, depending on the actuator selection. Additional control strategies might be used for different target applications.

#### 2.3.4.2 Finger pose estimation

A generic hand exoskeleton must track user's finger movements efficiently during operation. The exoskeletons in the literature adopt various strategies to track finger movements, mostly depending on mechanical and actuation choices.

**Actuator displacements** reveal the finger pose directly with high quality for the exoskeletons with independent finger control [35, 66, 74, 96, 99, 100, 102, 104, 112, 116]. Similarly, coupled exoskeletons with constant joint ratio between joint rotations track finger movements using the actuator displacements and this ratio [36, 37, 49, 53, 57, 59, 62, 67, 71, 73, 79, 83, 86, 87, 91, 93–95, 98, 101, 105, 107]. Using actuator displacements directly result in simple operational strategies and high quality tracking performance.

**Additional sensors** are needed for exoskeletons with other kinematics selections, when the actuated joints are not directly mapped into finger joints. For glove-based exoskeletons, flex sensors are placed along user's finger joints to measure the finger pose directly [50, 52, 58, 76, 78, 82, 88, 89]. Such flex sensors are low-cost, lightweight and of high

quality. Since these flex sensors should be grounded along finger joints to measure the finger pose directly, they require a texture-based interface.

Furthermore, additional sensors can track user's finger movements when inserted along mechanical joints, which are aligned with finger joints. These sensors can be chosen among hall-effect sensors [46, 81, 100, 108] or potentiometers [117]. The alignment between mechanical and finger joints measure the finger pose directly, so the measurements are quick and of high quality, while the sensors are mostly low-cost and lightweight. However, such direct pose tracking can be implemented only for exoskeletons with RCM mechanisms.

Non-contact optical [48, 67, 69, 83, 88] or magnetic sensors [111] require markers to be placed on finger phalanges or finger joints. They can be applied only if the exoskeleton allows for these markers to be placed on user's fingers without optical interface. It is important to note that in case of interference, the continuity of finger pose might be disturbed.

The sensor implementations discussed above require certain kinematics decisions. If a hand exoskeleton does not satisfy any of these properties, forward kinematics computation can be used to estimate the finger pose using additional sensors attached along random mechanical joints. Hall-effect sensors [77, 80, 108], optical encoders [44, 97], magnetic encoders [42, 90, 113] or potentiometers [30] can be used for such measurements. Even though the speed and efficacy of finger pose tracking depend on the quality of sensors and capabilities of control board, they can be implemented basically for all kinematics choices and target applications.

For devices with independent joint control, the actuator displacements measure the finger pose directly. For underactuated devices, additional sensors and forward kinematics are needed to estimate the finger pose.

## Chapter 3

# Design of the Hand Exoskeleton

Detailing the state-of-the-art exoskeletons based on their mobility, mechanical design, actuation and operation strategies helps a designer to understand the advantages and the disadvantages of possible design properties. Designing a new hand exoskeleton should start with highlighting the target application, listing the design requirements to be achieved, and analyzing the advantages/disadvantages of each possible design selection detailed in Section 2.

This chapter presents the design decisions for the proposed generic hand exoskeleton. Once the design requirements of a new device are revealed, the first kinematic sketch for a finger component will be presented. Furthermore, some of the presented design properties, such as the adjustability of the operational performance with respect to different object shapes and sizes are presented through CAD simulations.

After the first kinematics design, various pose analyses are needed to be performed in order to study the mechanism performance and operation. First, the inverse kinematics analysis will be performed to study whether the mechanical limits are satisfied for natural RoM of finger joints. Then, forward kinematics analysis will be presented using numerical and analytical approaches to obtain finger pose estimation thanks to an additional measurement tool. These analyses can be transformed to differential kinematics to form a relation between the finger and actuator velocities, as much as actuator forces and finger joint torques. Finally, an optimization algorithm will be performed to obtain the mechanical link lengths, which allows the user to reach their natural finger RoM, maximizing force transmission from actuator to finger joints.

### 3.1 Design Requirements

In Chapter 2, hand exoskeletons in the literature were detailed to understand common and individual properties based on their actuation, mechanics and working principles. The wide literature search proved that each choice regarding the previously specified classification items ensure different advantages and drawbacks of the design, considering different application scenarios. Yet, the search for the best hand exoskeleton that can be used for all types of applications is not over.

The complexity of hand anatomy, discussed in Section 2.1 prevents a unique list of design features to be achieved for an exoskeleton design. Furthermore, recent findings of clinical studies, technological developments about sensors, actuators and electronic equipment or recent developments of hand exoskeletons in general might affect main requirements and motivations of creating a new device.

First of all, a hand exoskeleton can be worn by users with impaired or unimpaired hand. Such an interaction introduces a set of generic requirements for a hand exoskeleton, without specifying the target application:

- **Effective force transmission:** As all human-interacted devices, the most important and mandatory specification for a hand exoskeleton system is to ensure user's safety throughout the operation. Such safety measures should be defined especially for patients with disabilities since they are more sensitive and vulnerable than healthy users. The overall system should be stable at all times, which might require additional precautions by control algorithms. Furthermore, forces between actuators and user's finger joints should be transmitted in a safe and efficient manner. The mechanism should be designed by strictly respecting the natural RoM of finger joints, such that the mechanism does not impose further reference to finger joints. Rotations along finger joints can be performed by imposing forces to finger phalanges, which should be natural at all times, not to cause any discomfort to users. Finally, while controlling multiple finger joints, torques being applied to finger joints should always have a balanced ratio not to cause finger pain at any orientation during operation.
- **Adjustability for the finger sizes:** Section 2.1 showed that the society has a wide range of variety regarding finger sizes. It is mandatory for a hand exoskeleton to adjust its operation respectively for different hand sizes. Devices, which align mechanical actuated joints with finger joints, have to create adjustable link

lengths to cope with size variations, such that they do not cause pain or unnatural movements to finger joints. Alternatively, such adjustment can be made automatically through misalignments between mechanical and finger joints intentionally and utilizing passive joints along the mechanism to improve the adjustability for different finger sizes.

- **Complexity of the hand movements & diversity of the tasks:** A generic hand exoskeleton should assist user's fingers to perform ADLs as much as basic joint movements. Certain rehabilitation tasks might require simple, repetitive flexion/extension movements of finger joints in order to improve finger mobility and RoM or to strengthen finger muscles. Conversely, other types of rehabilitation exercises might require users to perform ADLs in order to improve the cognitive abilities of patients with disabilities. A hand exoskeleton should be able to apply both assistance or assistance to patients with various needs to accelerate treatment process. Meanwhile, a hand exoskeleton should be able to assist users to perform various tasks in a real environment and to resist users to mimic an interaction during virtual environment tasks. Furthermore, a hand exoskeleton should not be designed only for grasping a certain object with a certain diameter, but the device's operation should be adjustable based on required motion. Such adjustability might involve both the independent mobility of fingers and finger joints for each task.
- **Wearability of the device:** A generic hand exoskeleton should be worn by users with impaired and unimpaired hand, which leads the designer to consider the wearability of such a device based on the capabilities of users with least free mobility. Post-stroke patients might exhibit residual contraction forces to close their fingers and difficulties to open their hands. In order to simplify the wearing and preparation process for those users, the device fitting would require to attach the exoskeleton to subject's fingers with no certain initial pose requirement. Avoiding stiff connections could make the fitting easier and less painful while improving the flexibility. With this motivation, the hand exoskeleton should be designed to be attached along the user's phalanges at any pose of the human hand. Creating an easily wearable device with no restriction for the initial pose not only simplifies the preparation time for patients with disabilities, but also for healthy users.
- **Portability:** The mechanical device should be designed with its actuators and power transmission components in a lightweight and efficient manner, without causing fatigue. There are some stationary devices in the literature to address rehabilitation exercises, but they might suffer from unnatural behavior of therapy

exercises. While designing a hand exoskeleton, it is crucial to find a solution to increase the portability of the device without increasing the fatigue and overall weight of the device.

A hand exoskeleton should satisfy these requirements without defining the purpose of use. Nevertheless, narrowing the purpose of use might define more specific requirements for a new hand exoskeleton. For instance, a hand exoskeleton to be used for physical rehabilitation should provide repetitive and useful physical tasks to patients with physical disabilities. The output forces of the device should be sufficient to overcome the disability level of patients, while the transmission of forces to finger phalanges possess less significance. Another important design criteria for a rehabilitation device is the easy wearability of device not to cause any trouble for users. However, a hand exoskeleton to be used for haptic purposes does not have limitations regarding wearability, but portability and light-weight of the device becomes more crucial so that users can move freely. Since the haptic device is used for force feedback, the transmission of forces to finger phalanges becomes more significant.

All additional properties can be achieved by modifying high-level choices for the kinematic architecture of device. These choices can be organized using classification items that were covered in Chapter 2. The motivation of the hand exoskeleton to be proposed in this thesis can be defined as assistive and rehabilitative applications, without restricting the possibility for haptic implementations. The characteristics of such a hand exoskeleton can be listed based on the previous classification items as following:

- **Hand Mobility:** The proposed hand exoskeleton should provide mobility to all fingers of user in order to complete various kinds of ADLs for haptics and assistance, and physical exercises for rehabilitation. In fact, assisting all fingers provide a natural operation for the user to open/close fingers for all application types. Such a mobility can be given by coupling finger joints together, or by transmitting forces to finger joints using a single actuator. Even though such opening/closing the fingers together might be useful for grasping a certain type of objects, some ADLs require the fingers to be moving independently, such as fingertip grasping or using a key. Allowing independent control of fingers require individual finger components mechanically, which utilize independent actuators, in order to increase the variety of different tasks to be performed by the device. Since index, middle, ring and little fingers are kinematically same, the same mechanical structure can be extended to all these fingers, optimizing link lengths of finger components based on

finger lengths depicted in Table 2.1. Unfortunately, the joint capabilities of thumb for abduction/adduction require a different mechanism to be designed specifically for it. Due to the complexity of thumb component, this study focuses on the development of other four finger components.

- **Finger Mobility:** As mentioned before, a human finger has 4 *DoF* mobility, 3 flexion/extension and 1 abduction/adduction. Since most of the ADLs do not require fingers to perform abduction/adduction movements, a hand exoskeleton can be designed to ignore such mobility or allow users to move them passively. A healthy user cannot move his DIP joint independently from PIP joint due to an anatomical coupling between these two. Such natural coupling allows a hand exoskeleton to be simplified by leaving DIP joint free. Assisting 2 *DoF* flexion/extension, rotating MCP and PIP joints only, improves simplicity and wearability of the mechanism without sacrificing operational performance. The assisted joints need to be consistent with natural RoM stated previously in Table 2.3 to ensure user's safety at all times. Even though each finger component aims to achieve 2 *DoF* mobility, actuating both joints independently might cause the device to be costly, bulky and heavy. Instead of controlling finger joints independently, they can be controlled by a single actuator in order to improve design simplicity, portability and affordability.
- **Number of Connection Points:** A hand exoskeleton can control MCP and PIP joints for flexion/extension rotations in various ways. Even though embracing a 1 *CP* design, where the device is connected to middle phalange of user, can rotate both joints in a coupled manner using a single actuator, it cannot give sufficient support for both finger joints. Alternatively, a 2 *CPs* mechanism can be designed to connect the device from proximal and middle finger phalanges of users. Doing so, the mobility of finger phalanges are being limited by the mechanical device, even though they are not controlled independently. Leaving fingertip free allows the device to be possible merged with a tactile device while improving the perception of touch during virtual tasks. 2 *DoF* finger mobility by an exoskeleton with 2 *CPs* can be controlled either by utilizing differential mechanisms, which define a rotation ratio between finger joints, or further underactuation methods, which adjust finger movements based on interaction forces. Differential mechanisms to transmit actuator forces to finger phalanges define a strict rotational ratio between finger joints, which can be adjusted only by changing the link lengths of the mechanism or transmission system manually. Therefore, the task adjustment cannot

be achieved automatically. For the proposed device, underactuation concept [121] based on contact forces was adopted by relating forces acting on finger phalanges to each other through linkage mechanism and passive joints. This property allows the device to apply stable forces to finger phalanges during grasping tasks using objects with any shape and size. Kinematics selection to achieve underactuation defines mechanical joints implicitly misaligned with finger joints, thanks to linkage mechanism and passive mechanical joints. Such a misalignment considers finger phalanges as a part of kinematic chain [72], which allows the device to adjust for various finger sizes automatically.

- **Finger Pose Estimation:** The choice of embracing underactuation allows 2 finger joints to be controlled using 1 actuator without sacrificing from adjustability of achievable tasks, while suffering from the lack of sufficient sensory measurements. Such a lack of measurements prevent finger joints to be predicted during operation only using measurements about actuator activity. Since the proposed underactuation concept is based on contact forces instead of joint ratio, there is no constant relation between MCP and PIP joints to predict instantaneous finger pose. The lack of glove-based interaction between the user and the device abolishes the possibility to implement bending sensors. Alternatively, an additional encoder or a potentiometer can be inserted on a mechanical passive joint to provide extra sensory measurements to calculate finger joints through pose analysis.
- **Actuation:** Controlling each underactuated finger component requires independent actuators to be used. Even considering a future thumb component, 5 required actuators should be inserted on top of the hand in order to connect the actuator directly to finger component, removing the need for a transmission unit. Placing actuators on top of the hand improves the control accuracy and portability, while limiting possible actuators to be used, such that fitting all actuators in a compact manner requires the use of low-cost, light-weight linear DC motors. There are many compact linear DC motors in the market; some of them reach a linear displacement using the magnetic field within closed case. Despite of their miniature design, their magnetic based working principle requires a large gap between actuators not to cause magnetic interference. Alternatively, linear motors can be supported by mechanical gear boxes to amplify output forces, without affecting the environment. Even though utilizing gear boxes prevents backdriveability of actuators, they can apply sufficient force to open/close user's fingers.

- **Mechanism Placement:** The finger components of the hand exoskeleton was proposed to be placed above fingers in order to keep the hand palm free for real grasping tasks and to minimize the width of mechanism for multi-finger implementations with no mechanical interference. Placing finger components above fingers allow actuators to be placed on top of the hand and the wrist, simplifying force transmission problem and increasing portability. Since the dorsal side of the hand has no physical limitation for the mechanical device, the height of the device can be extended in order to optimize its performance and reach higher RoM of finger joints, as long as it does not cause fatigue to users during operation.
- **Control:** The hand exoskeleton can be controlled passively by a position control, where the actuator displacement is moved to follow its reference. Such control allows fingers to be opened/closed, while the underactuated mechanism flexes/extends finger joints in various modes. The hand exoskeleton can easily be controlled to open/close the fingers through muscular activity of the other hand, through teleoperation. Finally, the device can be controlled by the user in an active manner through force sensors and force control techniques. The mechanical gear boxes of selected actuators limit passive backdriveability, but the overall backdriveability of finger components can be ensured by introducing an individual force sensor for each finger component. The force sensor measures user's intention to open/close the finger and moves the finger component accordingly using force control methods.

The hand exoskeleton presented in this work ensures forces between user's phalanges and device to have a perpendicular direction at all times. This property significantly improves the design and functionality of the fasteners, which attach fingers to the exoskeleton. Since the fasteners are crucially used to transmit the tangential shear forces, there is no need to excessively tighten the finger for transmitting either torques or longitudinal forces to the skin. Doing so, the longitudinal forces and torques are prevented to be applied on finger phalanges by the mechanism. The absence of longitudinal forces allows the finger to be connected to the exoskeleton by simple straps around finger phalanges.

## 3.2 Proposed Hand Exoskeleton

Each one of these properties either answers to a design issue or gets us closer to define the kinematics for a finger component of the proposed hand exoskeleton. Figure 3.1 shows the sketch of the final kinematics.

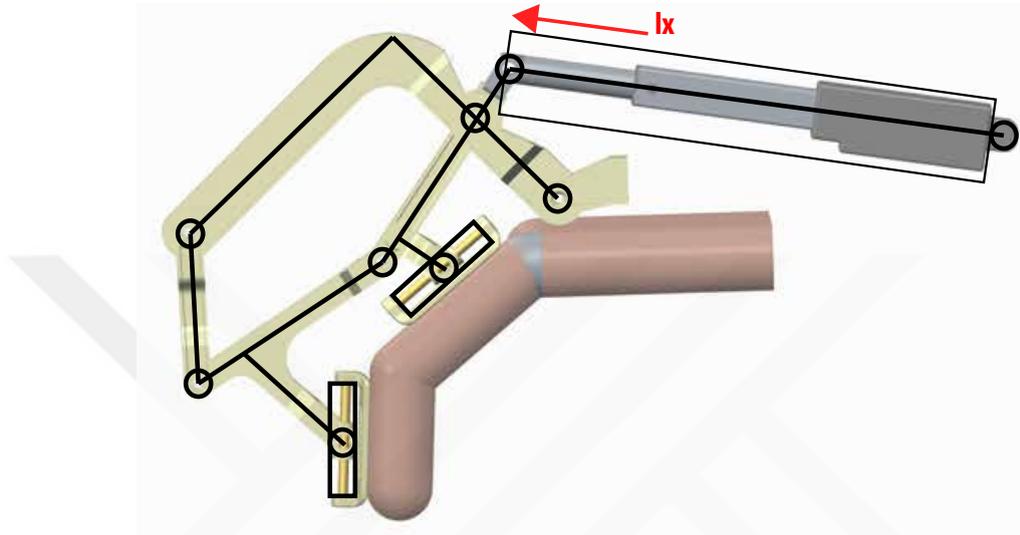


FIGURE 3.1: Kinematic selection for a finger component of the proposed hand exoskeleton under given assumptions drawn in CAD.

The kinematics structure is designed as an underactuation based linkage structure to ensure easy wearability and high adjustability for different users. Let's focus on the details of the underactuation concept in the next subsection. A linear DC motor allows the device to close/open the finger in a direct manner. The chosen actuators are small and compact in size, so that it can be placed on top of the hand, along with the mechanism itself.

Placing the mechanical parts on the dorsal side of the finger allows user to contact with real objects. The whole structure is connected to the user's finger from the proximal and the middle phalanges to control the MCP and PIP joints. Mechanical connection between the device and the finger phalanges are performed through combining a cylindrical joint and a rotational joint with perpendicular axis, such that the lateral forces can be transmitted into motion and the perpendicular forces can be used to move the MCP and PIP joints.

### 3.2.1 Underactuation

Underactuation concept has been implemented on robotic gripper hands to allow a smaller number of actuators without decreasing the number of DoFs during grasping tasks. For mechanical grippers, passive spring-like elements [126] or soft structure [127] can transmit forces to the upcoming joints based on the interaction forces acting on the gripper links. Such a transmission makes the robotic hand to adjust its behavior to the object's size and shape automatically [128, 129].

The idea behind underactuation is simple: the actuation is turned into motion at the first output joint, until the first mechanical link gets in contact with an object. After the first link reaches an object and starts feeling the interaction forces, the spring-like joint coupling between output joints allows the mechanism to transmit the same actuation for the second output joint. Actuator forces are transmitted to output joints until the grasping is completed. Using a single actuator to move multiple joints allows the device to achieve smaller size, less weight and lower cost, since actuators are typically the largest, heaviest and most expensive components of robotic devices.

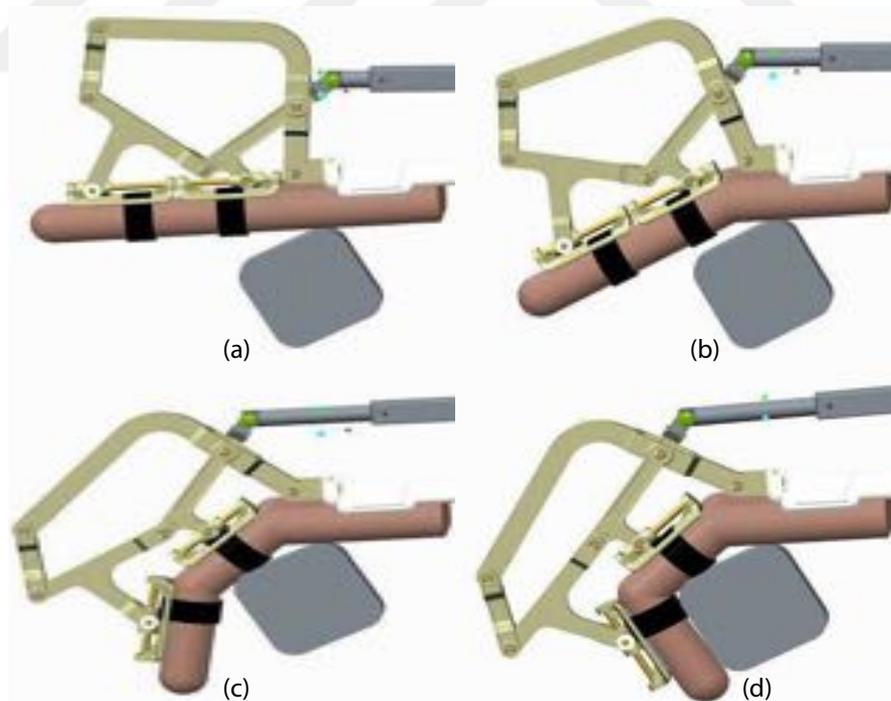


FIGURE 3.2: Underactuation concept of the proposed hand exoskeleton during a grasping task: (a) initial pose with all finger joints extended, (b) actuator force moves MCP joint until the first finger phalange gets in touch with the object, (c) actuator force is transmitted to move PIP joint when the first finger phalange touches the object and (d) finally grasping task is completed when both phalanges touch the object.

The same underactuation concept can be extended to a hand exoskeleton, by considering user's fingers as a part of the mechanism. Anatomical coupling between finger joints define the required spring-like behavior for the underactuation. Then, the efficient force transmission requires the use of additional passive joints, which can assist motion for finger joints without aligning mechanical joints to the finger joints. Such an inherent passive compensation also improves the robustness of the device.

Figure 3.2 represents the operation flow of the underactuated hand exoskeleton. The finger component is connected to the user's finger, and assists the finger to flex until both finger joints are fully extended and the actuator stroke is at its minimum limit. In the beginning, as the actuator starts to move, the exoskeleton transmits forces to proximal phalange and rotates MCP joint. Encountering external forces along the proximal phalange caused by the physical interaction finalizes the rotation along MCP joint. These forces transmit actuation forces to intermediate phalange through the linkage based mechanism with passive joints, rotating PIP joint. Having  $2DoF$  to control, grasping task is completed when PIP joint is rotated enough for intermediate phalange to reach the object. Similarly, the extension of the finger starts by rotating PIP joint first and MCP joint later until the finger is totally extended and finger joints reach their physical limits. As a safety measure, the displacement stroke is designed to reach its minimum limit when both finger joints are extended to their natural limits.

The underactuation concept does not control each finger joint individually, but controls the relation between them based on interaction forces. Fig 3.3 shows that finger joints can perform a constrained movement even when the actuator is fixed. This constraint

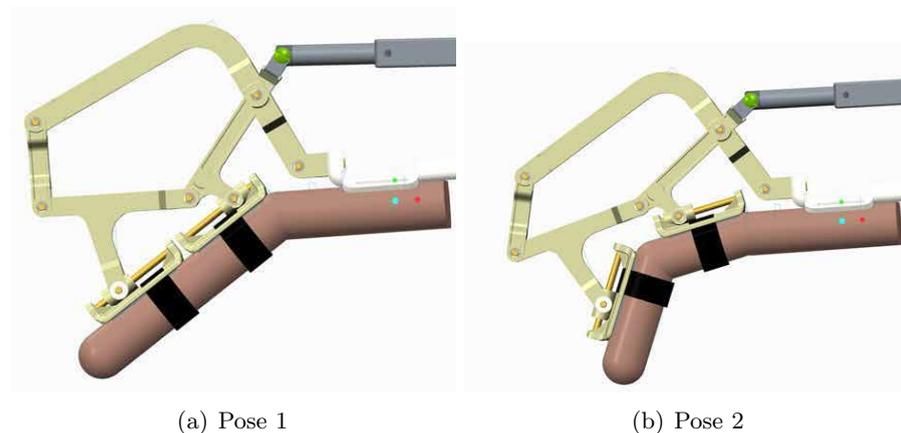


FIGURE 3.3: Different poses of finger joints that can be achieved when the actuator displacement is locked due to the extra mobility introduced by the underactuation concept.

is achieved by introducing additional passive joints to the system in order to increase the mobility of the overall device.

Thanks to this extra mobility, underactuation concept can adjust the operation of a mechanism based on the shape and the size of the grasping object with no prior information about the object or the orientation of the hand respect to the object. Figure 3.4 shows the device grasping objects with different shapes and sizes.

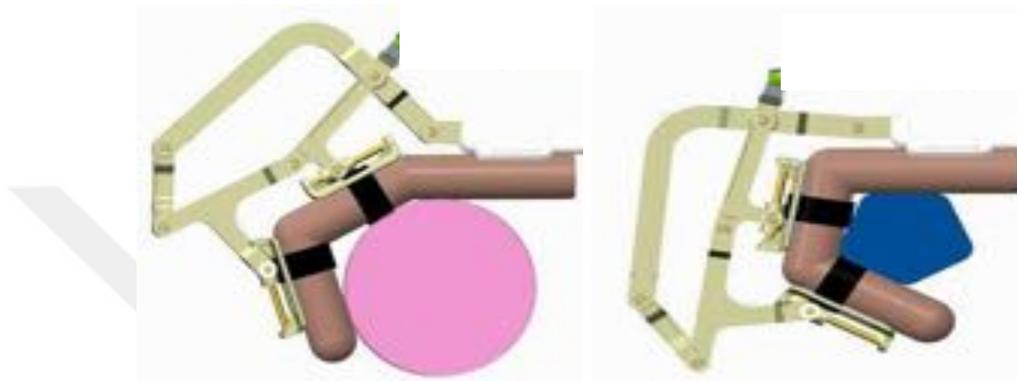


FIGURE 3.4: Adaptability of underactuated hand exoskeleton for objects with different sizes and shapes due to the extra mobility.

### 3.3 Mechanical Pose Analysis

After designing the first sketch of the proposed exoskeleton as Figure 3.1, we need to analyze the kinematics for the following purposes:

- to analyze the behavior of the proposed device during operation,
- to set limits along mechanical joints and transmitted forces to finger phalanges,
- to avoid physical limitations of the mechanism, such that the finger lengths, the stroke limits of the actuator, etc., and
- to optimize the force transmission and RoM for finger joints.

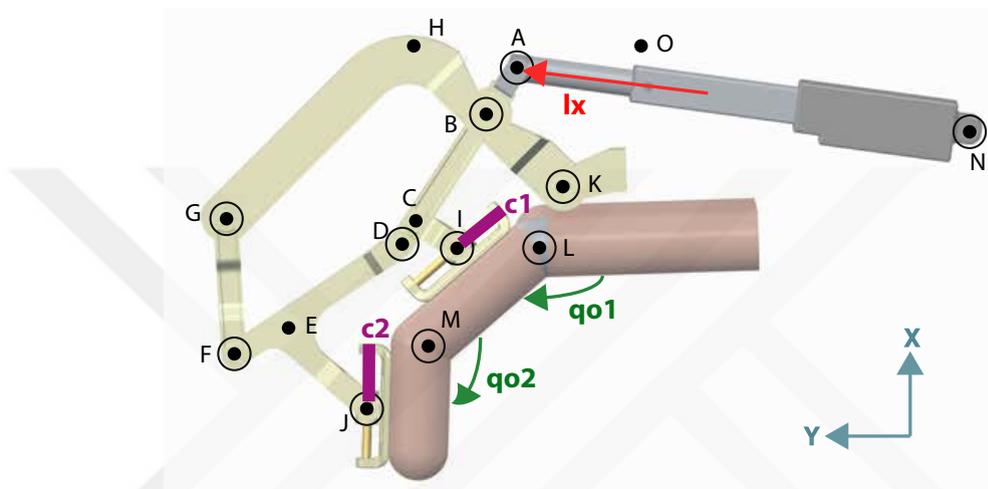
The extra mobility introduced by the underactuation prevents a unique solution for finger joints knowing only actuator displacement. The coupled movement between finger joints for a certain actuator displacement cannot predict finger joints with no further equipment. This is why, the initial mechanical pose analysis will be performed using inverse kinematics in Section 3.3.1, such that the unique configuration for actuator displacement and mechanism configuration will be calculated for the given pose of finger joints.

Inverse kinematics is sufficient to study joint limitations and to optimize the link lengths in order to improve joint RoM and force transmission over finger phalanges. However, using inverse kinematics during real time implementations is almost useless. As discussed before, a unique solution with forward kinematics cannot be reached unless an additional sensory measurement is used within the system. With this motivation, one of the passive joints of the mechanical system is measured with a rotational potentiometer to solve the forward kinematics. We will approach forward kinematics problem numerically in Section 3.3.2, and analytically in Section 3.3.3.

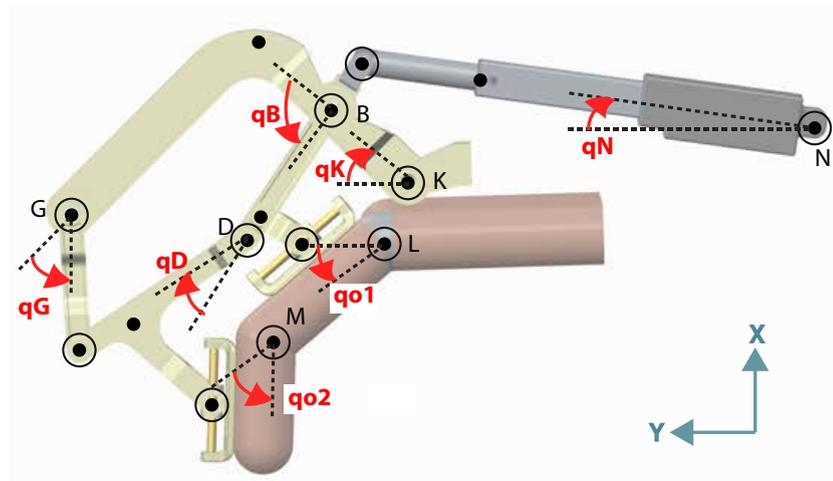
Having 2 sensory measurements and 2 finger joints as output results a square, invertible Jacobian matrix through differential kinematics for velocities (see Section 3.3.4) and for statics (see Section 3.3.5).

Figure 3.5 defines and shows both active and passive joints to solve kinematics. A linear actuator is attached to the point  $A$  with a displacement of  $l_x$ . MCP and PIP joints of user's finger are defined as points  $L$  and  $M$  respectively, with the rotations as  $q_{o1}$  and  $q_{o2}$ . The mechanism consists of 9 passive revolute joints at points  $A, B, D, F, G, I,$

$J$ ,  $K$ , and  $N$  while their rotations are represented as  $q_i$ ,  $i$  being the point name. The passive sliders are shown with the points  $I$  and  $J$ , with the displacement of their passive linear joints  $c_1$  and  $c_2$ . Finger phalanges are connected to the device at these points. The points  $H$ ,  $C$  and  $E$  are additional points that are needed for the calculations, even though they are located on rigid links and do not have any mobility. The point  $N$  shows the point between the actuator and the base. Finally, the point  $O$  shows the initial position of point  $A$  when the actuation stroke  $l_x$  is zero.



(a) Point definitions depicted on CAD model of proposed finger exoskeleton component.



(b) Joint definitions depicted on CAD model of proposed finger exoskeleton component.

FIGURE 3.5: Kinematic scheme with point and joint definitions of a finger component for the proposed underactuated hand exoskeleton.

### 3.3.1 Inverse Kinematics

Inverse kinematics uses finger joint rotations ( $[q_{o1}, q_{o2}]$ ) and finds the desired mechanical joints and actuation displacement ( $[q_B, q_D, q_K, q_G, q_N, c_1, c_2, l_x]$ ) to achieve these desired joints. Since these rotational joints can be used only in the trigonometric relations, the system becomes nonlinear. Such nonlinearity leads the inverse kinematics to be performed through mechanical closed loops of system with a set of vector equations. Table 3.1 describes all unknown parameters, constants and finger pose parameters that will be used to define vectors between two points stated in Figure 3.5. In this table, the notation  $\mathbf{r}_j^i$  is used to indicate vectors connecting point  $i$  to point  $j$ , and  $l_{ij}$  shows a constant length between point  $i$  and point  $j$ . Inverse kinematics problem defines a given finger pose as known parameters, they are specified separately in Table 3.1.

TABLE 3.1: Variable and constants for the vectors to solve inverse kinematics

Vector	Unknown Parameter	Constants	Finger Pose Parameter
$\mathbf{r}_A^O$	$l_x, q_N$		
$\mathbf{r}_N^O$		$l_{act}$	
$\mathbf{r}_D^A$	$q_B$	$l_{AD}$	
$\mathbf{r}_I^C$		$l_{CI}$	
$\mathbf{r}_K^G$	$q_K$	$l_{GK}$	
$\mathbf{r}_N^K$		$l_{KN}, q_{KN}$	
$\mathbf{r}_L^I$	$c_1$		$q_{o1}$
$\mathbf{r}_I^M$		$l_{ML}$	
$\mathbf{r}_K^L$		$l_{LK}, q_{LK}$	
$\mathbf{r}_F^D$	$q_D$	$l_{DF}$	
$\mathbf{r}_J^E$		$l_{EJ}$	
$\mathbf{r}_M^J$	$c_2$		$q_{o2}$
$\mathbf{r}_F^G$	$q_G$	$l_{GF}$	

Table 3.1 shows that the whole mechanism can be defined with vectors using 8 unknown parameters as  $\{l_x, c_1, c_2, q_B, q_D, q_G, q_K, q_N\}$  and 2 known finger pose variables as

$\{q_{o1}, q_{o2}\}$ . Even though there are other passive revolute joints ( $\{q_A, q_F, q_I, q_J\}$ ), their rotations are not required to define the device configuration since they are constrained by other variables. To find a unique solution for these 8 unknowns, 8 independent equations are needed: four independent kinematic loops were identified as presented in Figure 3.6 using vector chains along the mechanical links.

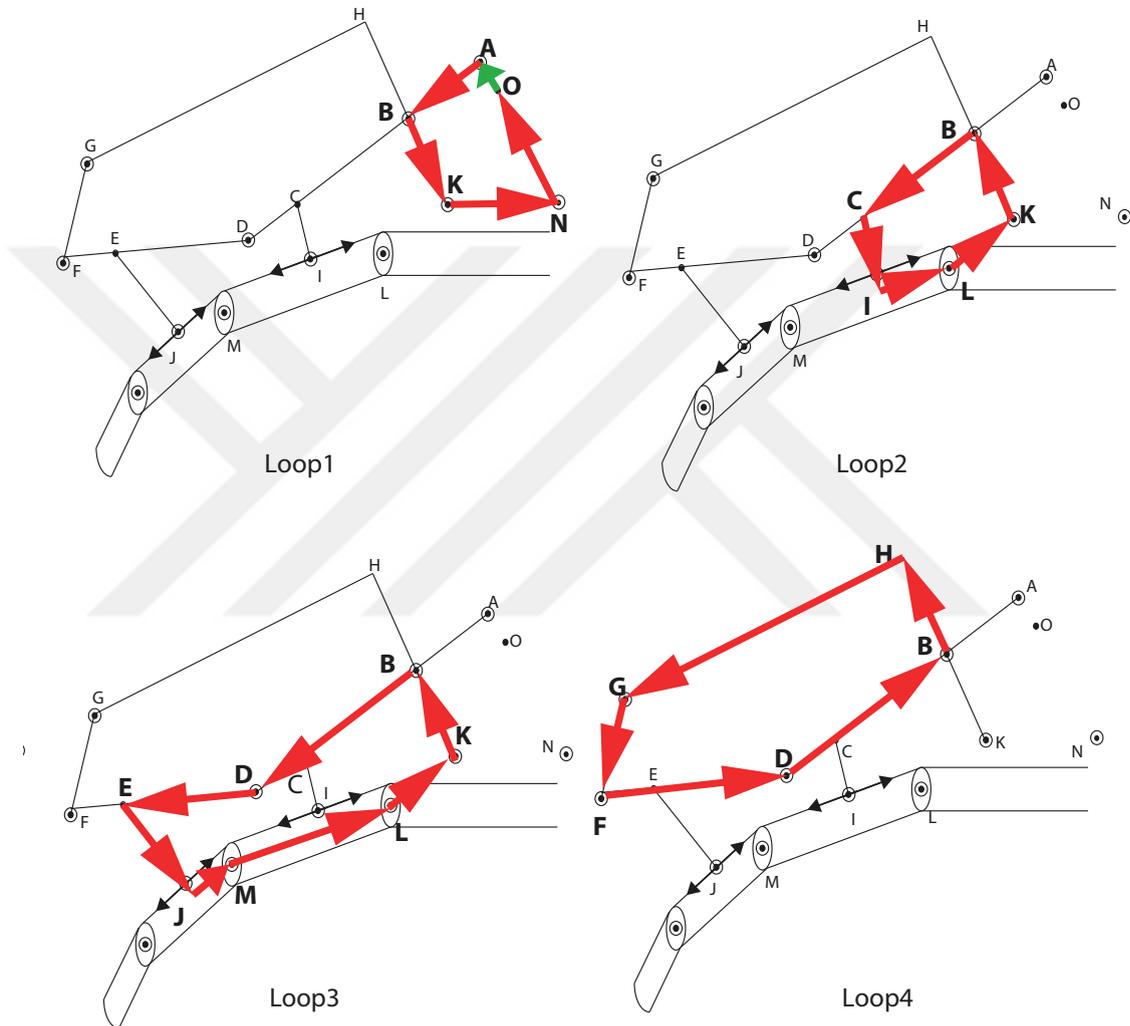


FIGURE 3.6: Independent kinematic loops to solve numerical inverse kinematics. Red arrows show the vectors defined by passive joints and rigid links, while the green arrow shows the active actuator displacement, which can be controlled.

*Loop1* defines a relation between the actuator displacement and the mechanism itself using a vector loop equation and its corresponding exponential expression as:

$$\begin{aligned} \mathbf{r}_A^O + \mathbf{r}_B^A + \mathbf{r}_K^B + \mathbf{r}_N^K + \mathbf{r}_O^N &= \mathbf{0} \\ l_x e^{iq_N} + l_{AB} e^{iq_B} + l_{BK} e^{iq_K} + l_{KN} e^{iq_{KN}} + l_{act} e^{iq_N} &= 0. \end{aligned} \quad (3.1)$$

In this loop, we can solve the unknown parameters  $\{l_x, q_B, q_K, q_N\}$ . *Loop2* defines the corresponding motion around the MCP joint of user's finger using passive joints affected by the actuator:

$$\begin{aligned} \mathbf{r}_B^K + \mathbf{r}_C^B + \mathbf{r}_I^C + \mathbf{r}_L^I + \mathbf{r}_K^L &= \mathbf{0} \\ l_{BK} e^{iq_K} + l_{BC} e^{iq_B} + l_{CI} e^{iq_C} + c_1 e^{iq_{o1}} + l_{LK} e^{iq_{LK}} &= 0. \end{aligned} \quad (3.2)$$

The unknown parameters of *Loop2* can be listed as  $\{q_K, q_B, c_1\}$ , as much as the finger pose parameter  $\{q_{o1}\}$ . Similarly, the PIP joint rotation of user's finger is defined by *Loop3* using passive joints of the mechanism:

$$\begin{aligned} \mathbf{r}_B^K + \mathbf{r}_D^B + \mathbf{r}_E^D + \mathbf{r}_J^E + \mathbf{r}_M^J + \mathbf{r}_L^M + \mathbf{r}_K^L &= \mathbf{0} \\ l_{BK} e^{iq_K} + l_{BD} e^{iq_B} + l_{DE} e^{iq_D} + l_{EJ} e^{iq_D} + c_2 e^{iq_{o2}} + l_{ML} e^{iq_{o1}} + l_{LK} e^{iq_{LK}} &= 0. \end{aligned} \quad (3.3)$$

*Loop3* covers the unknown parameters  $\{q_D, c_2\}$  and and finger pose parameters  $\{q_{o2}, q_{o1}\}$ . Finally, *Loop4* provides a relation between mechanical passive joints alone using the variables  $\{q_K, q_B, q_G, q_D\}$  in order to achieve a unique solution for the pose analysis:

$$\begin{aligned} \mathbf{r}_H^B + \mathbf{r}_G^H + \mathbf{r}_F^G + \mathbf{r}_D^F + \mathbf{r}_B^D &= \mathbf{0} \\ l_{BH} e^{iq_K} + l_{HG} e^{iq_K} + l_{GF} e^{iq_G} + l_{FD} e^{iq_D} + l_{DB} e^{iq_K} &= 0. \end{aligned} \quad (3.4)$$

These loops were chosen such that each loop (*Loop1* - *Loop4*) has at least one parameter, which is not covered by the other loops. In particular, Table 3.2 details the input and the output parameters covered by each loop, and unique output parameters covered by each loop.

TABLE 3.2: Input and output parameters used for each loop and unique output parameters for inverse kinematics.

Loop	Input Parameters	Output Parameters	Unique Output Parameters
<i>Loop1</i>		$l_x, q_N, q_B, q_K$	$l_x, q_N$
<i>Loop2</i>	$q_{o1}$	$q_B, q_K, c_1$	$c_1$
<i>Loop3</i>	$q_{o1}, q_{o2}$	$q_D, c_2, q_K$	$c_2$
<i>Loop4</i>		$q_K, q_G, q_D, q_B$	$q_G$

We can assume that the 4 loops are chosen carefully to be independent from each other, since there is at least one parameter covered only by each loop. Even though the loops above can be chosen in a different manner as well, they were chosen in the simplest way possible.

The vertical and the horizontal ( $X$  and  $Y$  as depicted in Figure 3.5) components of the vector equations in Equations (3.1 - 3.4) achieve 8 nonlinear equations. Due to the nonlinearity of the system, the numerical method finds a unique configuration for actuator displacement  $l_x$  as well as passive joints  $[c_1, c_2, q_B, q_D, q_G, q_K, q_N]$  for given finger pose  $q_{o1}$  and  $q_{o2}$ .

The inverse kinematics analysis outputs were compared to the CAD simulation results for the index finger exoskeleton. We present the index finger results only, since the mechanical design is identical for the other fingers as well (except the thumb).

The finger joint rotations for MCP and PIP joints as shown in Figure 3.7 was supplied to both the mechanical simulation in the CAD software, and the kinematics simulation in MATLAB. Figure 3.8 presents the resulting actuator displacement  $l_x$  and joint rotation  $q_B$ . Firstly, the actuator displacement  $l_x$  was compared between CAD simulation and kinematics analysis, since it is the only controllable state of the system. We chose passive joint  $q_B$  to be an observable state, so  $q_B$  is the second joint compared between the two simulations. Even though any of other passive joint could be chosen for confirmation, joint  $q_B$  will be used for further analysis for reasons to be justified later. The errors between simulation tools for both states prove that inverse kinematics calculations are sufficiently accurate define the mechanical device.

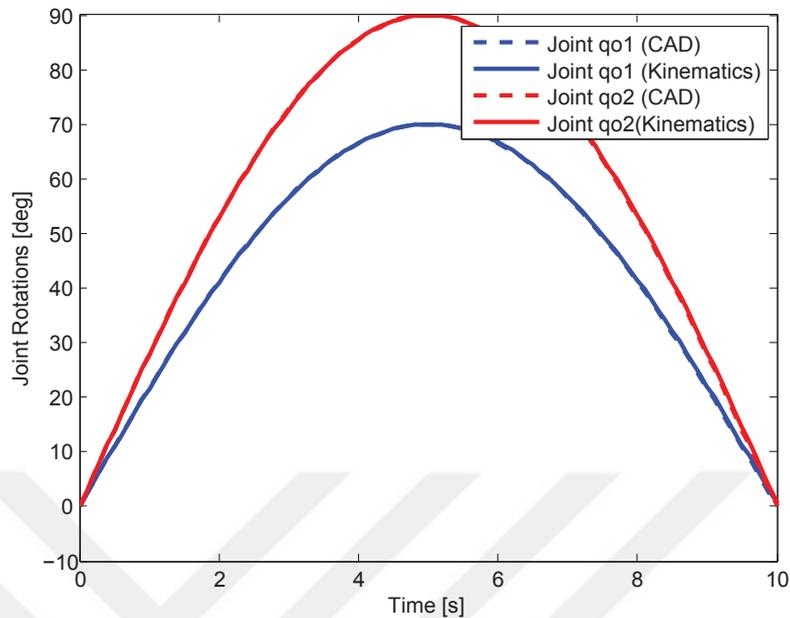


FIGURE 3.7: Finger joint rotations defined as input for the CAD simulation and the kinematics analysis to calculate inverse kinematics.

### 3.3.2 Forward Kinematics

Even though inverse kinematics gives sufficient information about the mechanism behavior, physical limits, actuator requirements and feasible workspace for finger joints; it is useless for real time implementation of the device. The kinematics analysis will be useful to predict finger pose using mechanical pose and measurements. In particular, each finger component of the hand exoskeleton is actuated by a linear actuator and its linear displacement is the only available measurement tool during operation.

A unique solution for the forward kinematics can be reached simply by providing an additional sensor to the mechanism. Such addition does not affect the underactuation since it does not provide any energy to the overall system, but contributes to estimate the coupled mobility between finger joints that is left uncontrolled. The proposed kinematics have multiple passive joints and choosing the appropriate one to measure during operation should consider the following issues:

- It should be away from the passive joints at the connection tools near finger phalanges, not to interfere with user and other mechanical components.
- It should be close to the exoskeleton base to shorten the cables as much as possible.

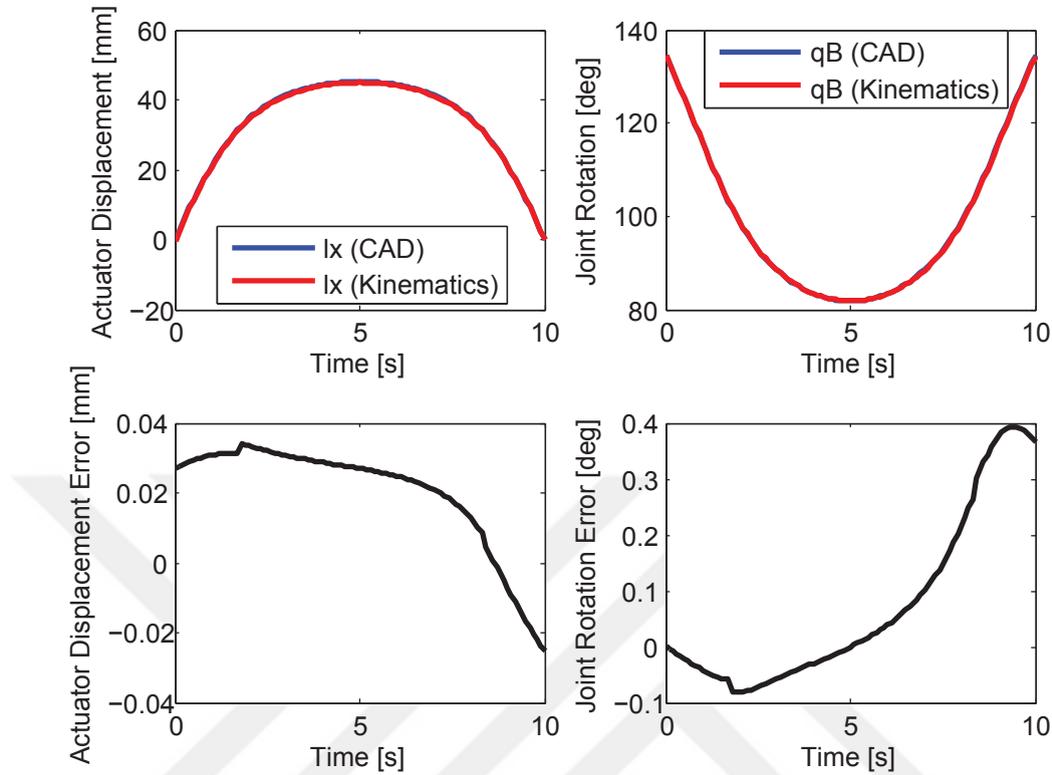


FIGURE 3.8: Comparison between the simulations of the CAD model and the inverse kinematics analysis through actuator displacement  $l_x$  and an additional passive joint  $q_B$ .

- Its displacement during operation should be wide to increase the accuracy of pose estimation.

These considerations highlighted the joint along point  $B$  depicted on Figure 3.5 as a second measurement tool for forward kinematics and pose estimation. The wide workspace of joint  $B$  can easily be obtained in Figure 3.2 and Figure 3.4. The pin connecting two links along joint  $B$  allows a simple assembly of a rotational potentiometer while its place minimizes the complexity of cable transmission.

Using such an additional sensory measurement, forward kinematics of the mechanism can be calculated uniquely. Numerical forward kinematics approach use the same vector definitions in Figure 3.6 using 2 input parameters as  $\{q_B, l_x\}$ , and 8 unknown parameters as  $\{q_{o1}, q_{o2}, c_1, c_2, q_D, q_G, q_K, q_N\}$  as redefined for forward kinematics in Table 3.3. Similar to inverse kinematics, additional passive joints  $\{q_A, q_F, q_I, q_J\}$ , are ignored, since their movements are constrained by other parameters. Similarly, 8 independent

loop equations are obtained using  $X$  and  $Y$  components of 4 loop equations as described in Figure 3.6 in order to find a unique solution for these 8 unknowns. The forward kinematics loop equations are the same as inverse kinematics equations, Equation (3.1) - Equation (3.4) having different input and output variables.

TABLE 3.3: Variable and constants for the vectors to solve forward kinematics.

Vector	Unknown Parameters	Constants	Sensor Parameters
$\mathbf{r}_A^O$	$q_N$		$l_x$
$\mathbf{r}_N^O$		$l_{act}$	
$\mathbf{r}_D^A$		$l_{AD}$	$q_B$
$\mathbf{r}_I^C$		$l_{CI}$	
$\mathbf{r}_K^G$	$q_K$	$l_{GK}$	
$\mathbf{r}_N^K$		$l_{KN}, q_{KN}$	
$\mathbf{r}_L^I$	$c_1, q_{o1}$		
$\mathbf{r}_I^M$		$l_{ML}$	
$\mathbf{r}_K^L$		$l_{LK}, q_{LK}$	
$\mathbf{r}_F^D$	$q_D$	$l_{DF}$	
$\mathbf{r}_J^E$		$l_{EJ}$	
$\mathbf{r}_M^J$	$c_2, q_{o2}$		
$\mathbf{r}_F^G$	$q_G$	$l_{GF}$	

Even though the vectors used to define the mechanical links are the same for the forward and the inverse problems, changing input and output variables for each problem changes the whole analysis. Not only the analysis results, but also the independency of the defined loops vary by changing the variable definitions. Table 3.4 redefines input and output parameters that are covered by each loop and signifies unique output parameters that are not covered by other loops.

Similarly, the existence of unique parameters for each loop signifies the independency of those loops, such that the analysis might be trusted. The accuracy of numerical forward

TABLE 3.4: Analyzing input and output parameters used for each loop and unique output parameters for forward kinematics

Loop	Input Parameters	Output Parameters	Unique Output Parameter
<i>Loop1</i>	$l_x, q_B$	$q_N, q_K$	$q_N$
<i>Loop2</i>	$q_B$	$q_{o1}, c_1, q_K$	$c_1$
<i>Loop3</i>	$q_B$	$q_{o1}, q_{o2}, c_2, q_D, q_K$	$c_2, q_{o2}$
<i>Loop4</i>	$q_B$	$q_K, q_G, q_D$	$q_G$

kinematics will be performed together with analytical approach in Section 3.3.3.

Alternatively, the proposed kinematics structure can be easily adjusted for alternative needs. For instance, a calibration process can easily be created to estimate the length of proximal finger phalange in a quick way instead of measuring it manually for each user. Such a calibration process can be defined simply by changing input and output parameters as  $\{q_B, l_x, c_2\}$ , and unknown parameters as  $\{q_{o1}, q_{o2}, l_{LM}, c_1, q_D, q_G, q_K, q_N\}$ , where finger phalange is depicted as  $l_{LM}$ . For the calibration process, the lack of measurement can be overcome by asking the user to reach a pose in a manner, forcing the linear displacement along middle finger phalange to its maximum limit. Table 3.5 and Table 3.6 summarizes the parameters and uniqueness of these parameters as before.

TABLE 3.5: Variable and constants for the vectors to solve forward kinematics for calibration.

Vector	Unknown Parameters	Constants	Sensor Parameters
$\mathbf{r}_A^O$	$q_N$		$l_x$
$\mathbf{r}_N^O$		$l_{act}$	
$\mathbf{r}_D^A$		$l_{AD}$	$q_B$
$\mathbf{r}_I^C$		$l_{CI}$	
$\mathbf{r}_K^G$	$q_K$	$l_{GK}$	
$\mathbf{r}_N^K$		$l_{KN}, q_{KN}$	
$\mathbf{r}_L^I$	$c_1, q_{o1}$		
$\mathbf{r}_I^M$	$c_1, q_{o1}, l_{LM}$		
$\mathbf{r}_K^L$		$l_{LK}, q_{LK}$	
$\mathbf{r}_F^D$	$q_D$	$l_{DF}$	
$\mathbf{r}_J^E$		$l_{EJ}$	
$\mathbf{r}_M^J$	$q_{o2}$		$c_2$
$\mathbf{r}_F^G$	$q_G$	$l_{GF}$	

TABLE 3.6: Analyzing input and output parameters used for each loop and unique output parameters to solve forward kinematics for calibration

Loop	Input Parameters	Output Parameters	Unique Output Parameter
<i>Loop1</i>	$l_x, q_B$	$q_N, q_K$	$q_N$
<i>Loop2</i>	$q_B$	$q_{o1}, c_1, q_K$	$c_1$
<i>Loop3</i>	$q_B, c_2$	$q_{o1}, q_{o2}, q_D, l_{LM}, q_K$	$l_{LM}, q_{o2}$
<i>Loop4</i>	$q_B$	$q_K, q_G, q_D$	$q_G$

### 3.3.3 Analytical Forward Kinematics

Even though numerical forward kinematics provides an accurate output, the calculational burden of numerical approach might limit the speed of the soon to be implemented control board. For a calibration process, which should be run once for each user and for a short time, such a burden can be ignored. However, pose estimation of finger joints during a complete operation might require a simple, smoother and faster approach. Using the same constants, sensor parameters and unknown parameters as defined in Table 3.3 and independent kinematic loops different than Figure 3.6, forward kinematics can be solved analytically thanks to Cosine and Pythagorean Theorems. The new kinematic loops will be shown independently in Figure 3.9 - Figure 3.12.

Considering that input parameters are sensory measurements as  $q_B$  and  $l_x$ , the first loop can be defined as depicted in Figure 3.9 in order to obtain  $q_N$  and  $q_K$  through Equation (3.5) and Equation (3.6).

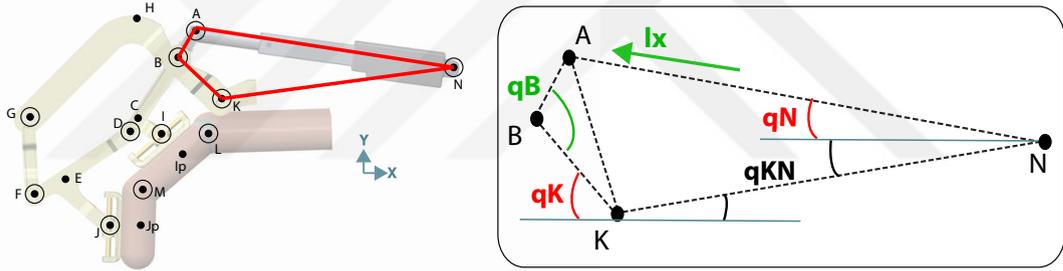


FIGURE 3.9: First loop for analytical forward kinematics to reach passive joint pose  $q_N$  and  $q_K$  using actuator displacement  $l_x$  and passive joint measurement  $q_B$ .

In particular, constant mechanical link lengths  $l_{AB}$ ,  $l_{BK}$  and measured joints  $l_x$  and  $q_B$  allows the distance  $l_{AK}$  to be known for any orientation using Cosine Theorem. Note that actuator distance  $l_x$  is hidden under  $l_{AN}$  expression, such that  $l_{AN} = l_x + l_{act}$ . Once  $l_{AK}$  is defined,  $q_K$  can be obtained using Equation (3.5).

$$q_K = \pi - \left[ \text{acos}\left(\frac{l_{AK}^2 + l_{BK}^2 - l_{AB}^2}{2 l_{AK} l_{BK}}\right) + \text{acos}\left(\frac{l_{AK}^2 + l_{KN}^2 - l_{AN}^2}{2 l_{AK} l_{KN}}\right) + q_{KN} \right] \quad (3.5)$$

Simultaneously, knowing  $l_{AK}$  leads joint rotation  $q_N$  to be calculated using Equation (3.6) based on Cosine Theorem for the right-side triangle of the given loop.

$$q_N = \text{acos}\left(\frac{l_{AN}^2 + l_{KN}^2 - l_{AK}^2}{2 l_{AN} l_{KN}}\right) - q_{KN} \quad (3.6)$$

The rest of analytical kinematics method requires the position of each point to be determined independently using previously obtained parameters, sensory parameters, constant parameters and distances between calculated points. Since these distances depend on the orientation of the device, they cannot be classified as constant variables. Nevertheless, obtaining numerical values of unknown parameters turn these distances from being expressions to numerical values. In fact, obtaining  $q_N$  and  $q_K$  allows the position of points  $G$  and  $D$  to be calculated, which defines distances  $l_{DG}, l_{HD}$  in Figure 3.10

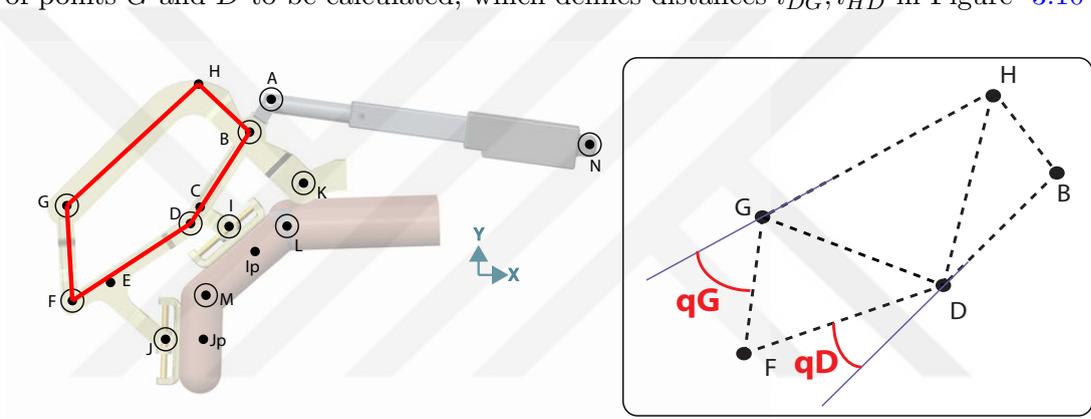


FIGURE 3.10: Second loop to calculate analytical forward kinematics to reach passive joint pose  $q_G$  and  $q_D$  using passive joint pose values  $q_N$  and  $q_K$  obtained in the previous loop.

Achieving passive joint rotation  $q_G$  using Equation (3.7) and  $q_D$  using Equation (3.8) requires the angles within triangles defined by the loop to be calculated using Cosine Theorem.

$$q_G = \pi - \left[ \text{acos}\left(\frac{l_{DG}^2 + l_{GH}^2 - l_{HD}^2}{2 l_{DG} l_{GH}}\right) + \text{acos}\left(\frac{l_{DG}^2 + l_{GF}^2 - l_{DF}^2}{2 l_{DG} l_{GF}}\right) \right] \quad (3.7)$$

$$q_D = \pi - \left[ \text{acos}\left(\frac{l_{HD}^2 + l_{DG}^2 - l_{GH}^2}{2 l_{HD} l_{DG}}\right) + \text{acos}\left(\frac{l_{DG}^2 + l_{DF}^2 - l_{GF}^2}{2 l_{DG} l_{DF}}\right) + \text{acos}\left(\frac{l_{BD}^2 + l_{HD}^2 - l_{BH}^2}{2 l_{BD} l_{HD}}\right) \right] \quad (3.8)$$

So far, analytical kinematic loops are selected similar to *Loop 1* and *Loop 4* depicted in Fig 3.6, which was designed for numerical kinematics approach. Such a similarity is not a surprise, since the previous loops were selected as simple as possible. However, numerical and analytical approaches start to vary in terms of kinematics loops from this point on. Obtaining  $q_K$  using Equation (3.5) allows the position of points  $L$ ,  $I$  and  $I_p$  to be known as well. In fact, point  $I_p$  is defined as the projection of point  $I$  along the plane where point  $L$  is placed and is parallel to finger surface.

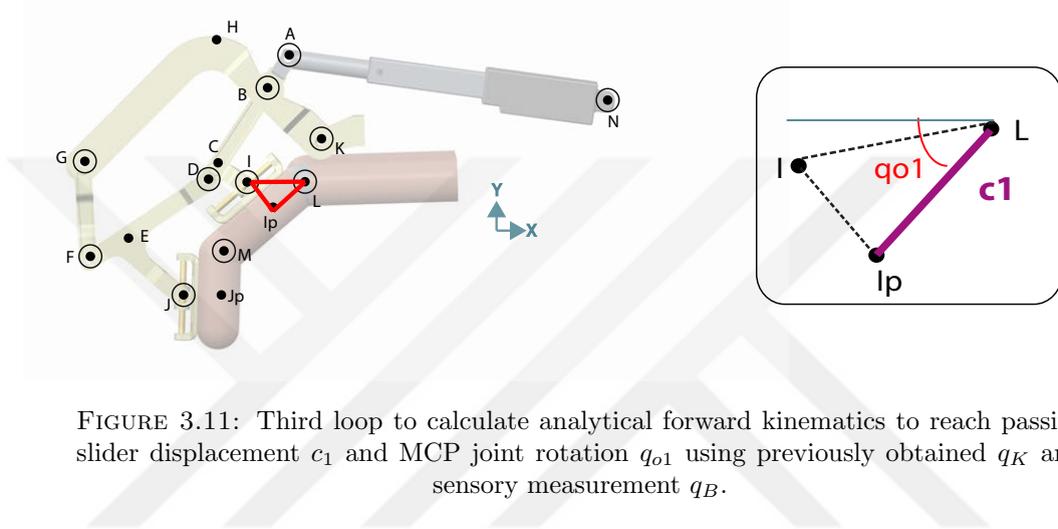


FIGURE 3.11: Third loop to calculate analytical forward kinematics to reach passive slider displacement  $c_1$  and MCP joint rotation  $q_{o1}$  using previously obtained  $q_K$  and sensory measurement  $q_B$ .

Since point  $I_p$  is defined as the projection of point  $I$ , the distance between points  $L$  and  $I_p$  can be calculated directly using Pythagorean Theorem to define passive linear displacement  $c_1$  through Equation (3.9).

$$c_1 = \sqrt{l_{LI}^2 - l_{IIp}^2} \quad (3.9)$$

Simultaneously, the rotation around the MCP joint  $q_{o1}$  can be calculated using Equation (3.10) simply by combining Pythagorean and Cosine Theorems.

$$q_{o1} = \text{acos}\left(\frac{l_{LI}^2 + c_1^2 - l_{IIp}^2}{2 l_{LI} c_1}\right) + \text{atan}\left(\frac{y_L - y_I}{x_L - x_I}\right) \quad (3.10)$$

Finally, obtaining  $q_D$  using Equation (3.8) and  $q_{o1}$  using Equation (3.10) allow the positions of points  $J$ ,  $J_p$  and  $M$  that are shown in Figure 3.12. In particular, point  $J$  is defined thanks to  $q_D$ , while points  $J_p$  and  $M$  are formed by  $q_{o1}$ .

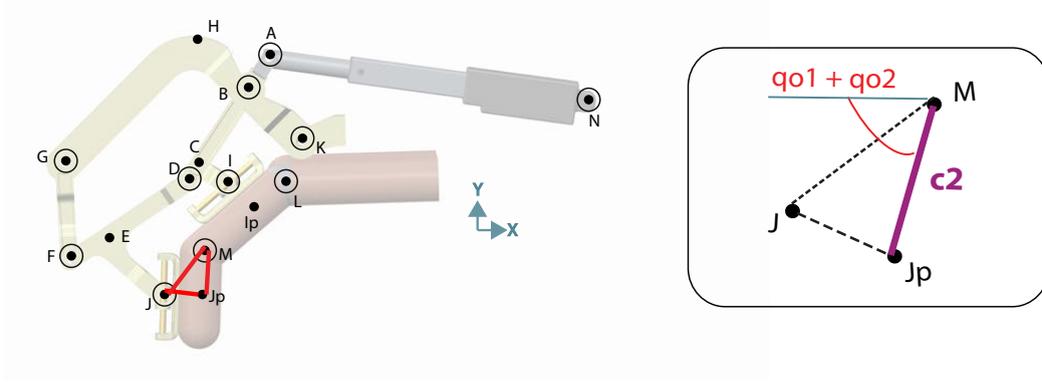


FIGURE 3.12: Final loop to calculate analytical forward kinematics to reach passive slider displacement  $c_2$  and PIP joint rotation  $q_{o2}$  using previously obtained  $q_D$ ,  $q_K$  and sensory measurement  $q_B$ .

Similar to the previous loop, since point  $J_p$  is defined as the projection of point  $J$ , the distance between points  $M$  and  $J_p$  can be calculated directly using Pythagorean Theorem to define passive linear displacement  $c_2$  using Equation (3.11).

$$c_2 = \sqrt{l_{JM}^2 - l_{JJ_p}^2} \quad (3.11)$$

Simultaneously, rotation around PIP joint  $q_{o2}$  can be calculated using Equation (3.10) simply by combining Pythagorean and Cosine Theorems.

$$q_{o2} = \arccos\left(\frac{l_{JM}^2 + c_2^2 - l_{JJ_p}^2}{2 l_{JM} c_2}\right) + \arctan\left(\frac{y_J - y_M}{x_J - x_M}\right) - q_{o1} \quad (3.12)$$

Both numerical and analytical methods to achieve a unique solution for finger pose through forward kinematics should be verified. With this motivation, a simulation setup was designed as can be summarized in Figure 3.13. In particular, inverse kinematics was run to calculate meaningful sensory information of  $l_x$  and  $q_B$  to reach  $70^\circ$  and  $90^\circ$  DoF along MCP and PIP joints respectively. The finger path was chosen the same as the previous experiment shown in Figure 3.7 to achieve compatible simulation results, even though such a verification was performed for many finger curling trajectories. Then, the sensory data obtained from inverse kinematics were used to run numerical and analytical forward kinematics simultaneously to compare their results.

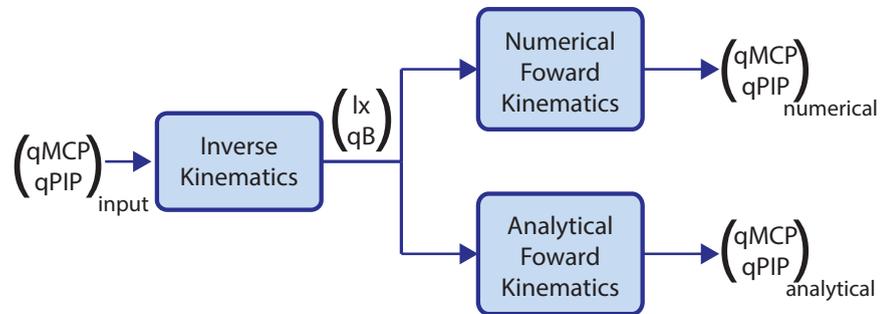


FIGURE 3.13: Simulation setup to compare numerical and analytical forward kinematic analyses using the same measurement set, which is computed by inverse kinematics for a set of finger joint rotations.

Figure 3.14 shows the MCP and the PIP joint rotations that are given as input, and are calculated from analytical and numerical forward kinematics methods. The error plot shows the error between numerical and analytical methods only, to focus on the variations caused by running different analyses. The obtained error between the two methods are small enough to be neglected and to assume that these two methods provide the same finger pose. The advantages and disadvantages of these techniques will be discussed later while proposing to estimate finger pose during real time tasks.

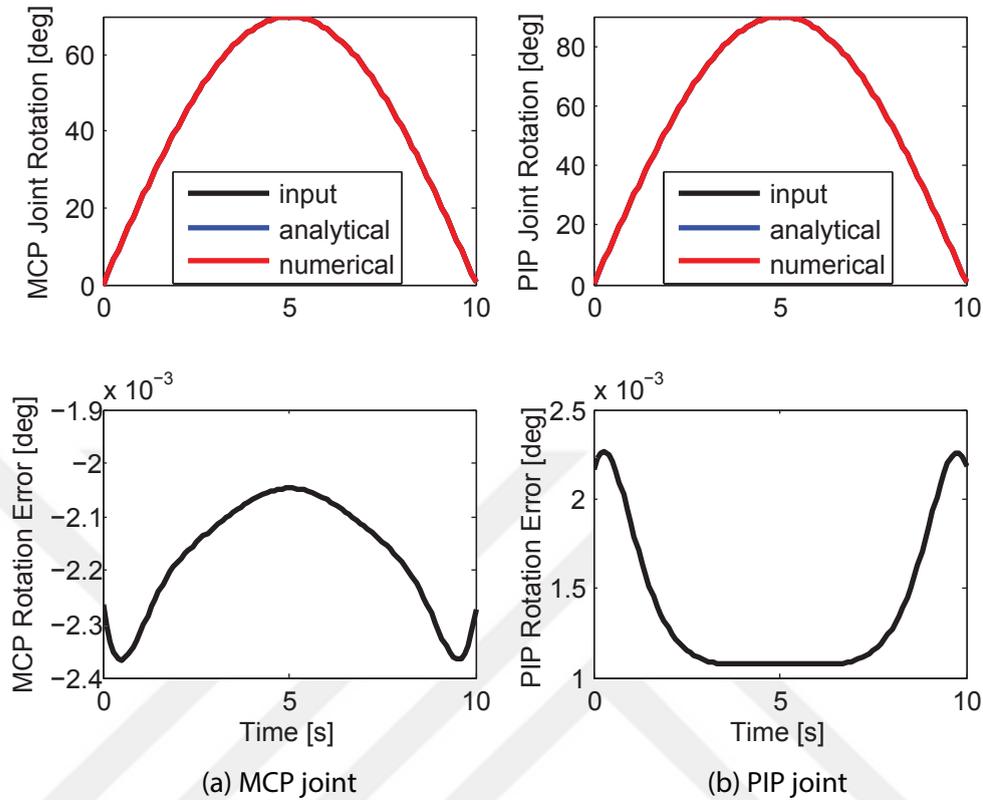


FIGURE 3.14: Comparison between numerical and analytical forward kinematics in simulation: (a) comparison for MCP joint rotation and (b) comparison for PIP joint rotation.

### 3.3.4 Differential Kinematics

Inverse and forward kinematics algorithm provide a relationship between input and output joints in the configuration level, while differential kinematics can be used to set a similar relationship in the velocity level. Such mapping is described by a matrix, based on the instant orientation of the system.

Differential kinematics can be calculated simply using the derivative of previously expressed nonlinear equations in Equation 3.1 - Equation 3.4. For an underactuated system, where actuator force is transmitted to finger joints by designing more passive joints, differential kinematics also focuses on extracting these passive joints in the velocity level. In order to simplify differential kinematics, finger joint velocities ( $\dot{\mathbf{q}}_{\text{fn}}$ ), measured joint velocities ( $\dot{\mathbf{q}}_{\text{m}}$ ) and passive joint velocities ( $\dot{\mathbf{q}}_{\text{p}}$ ) can be defined as presented in Equation (3.13).

$$\dot{\mathbf{q}}_{\text{fn}} = \begin{bmatrix} \dot{q}_{o1} \\ \dot{q}_{o2} \end{bmatrix} \quad \dot{\mathbf{q}}_{\text{m}} = \begin{bmatrix} \dot{l}_x \\ \dot{q}_B \end{bmatrix} \quad \dot{\mathbf{q}}_{\text{p}} = \begin{bmatrix} \dot{q}_K \\ \dot{q}_D \\ \dot{q}_G \\ \dot{q}_N \\ \dot{c}_1 \\ \dot{c}_2 \end{bmatrix} \quad (3.13)$$

Differentiating Equation 3.1) - Equation (3.4) can be categorized in the matrix form based on finger joint velocities, measured joint velocities and passive joint velocities as Equation (3.14).

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \\ \hline O_{31} & O_{32} \\ O_{41} & O_{42} \\ O_{51} & O_{52} \\ O_{61} & O_{62} \\ O_{71} & O_{72} \\ O_{81} & O_{82} \end{bmatrix} \begin{bmatrix} \dot{q}_{o1} \\ \dot{q}_{o2} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & R_{14} & R_{15} & R_{16} & R_{17} & R_{18} \\ R_{21} & R_{22} & R_{23} & R_{24} & R_{25} & R_{26} & R_{27} & R_{28} \\ \hline C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} & C_{37} & C_{38} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} & C_{47} & C_{48} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} & C_{57} & C_{58} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} & C_{67} & C_{68} \\ C_{71} & C_{72} & C_{73} & C_{74} & C_{75} & C_{76} & C_{77} & C_{78} \\ C_{81} & C_{82} & C_{83} & C_{84} & C_{85} & C_{86} & C_{87} & C_{88} \end{bmatrix} \begin{bmatrix} \dot{l}_x \\ \dot{q}_B \\ \dot{q}_K \\ \dot{q}_D \\ \dot{q}_G \\ \dot{q}_N \\ \dot{c}_1 \\ \dot{c}_2 \end{bmatrix}$$

Dealing with each joint velocities individually help Jacobian to achieve its components to form a  $8 \times 8$  matrix and complementary output matrix to form a  $8 \times 2$  shape. Equation (3.14) can be simplified as Equation (3.14) simply by merging finger joint velocities, measured velocities and passive joint velocities as  $\dot{\mathbf{q}}_{\text{m}}$ ,  $\dot{\mathbf{q}}_{\text{fn}}$  and  $\dot{\mathbf{q}}_{\text{p}}$ . The divisions for Jacobian and output matrices are shown with bold lines crossing through Equation (3.14). Table 3.7 represent the sizes of simplified matrices to clarify the definitions.

$$\begin{bmatrix} \mathbf{J}_{O_m} \\ \mathbf{J}_{O_p} \end{bmatrix} \dot{\mathbf{q}}_{\text{fn}} = \begin{bmatrix} \mathbf{J}_{R_m} & \mathbf{J}_{R_p} \\ \mathbf{J}_{C_m} & \mathbf{J}_{C_p} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_{\text{m}} \\ \dot{\mathbf{q}}_{\text{p}} \end{bmatrix} \quad (3.14)$$

The order of equations were chosen such that none of the sub-matrix components are zero matrices, where all the elements are 0, even though they can have 0 value in their elements. In particular, matrix components  $\mathbf{J}_{.m}$  indicate coefficients regarding the measured variables while components  $\mathbf{J}_{.p}$  signify passive variables. Similarly,  $\mathbf{J}_{O.}$ ,

TABLE 3.7: Matrix sizes of Jacobian matrix components.

Matrix Name	Size
$\mathbf{J}_{O_m}$	$2 \times 2$
$\mathbf{J}_{O_p}$	$6 \times 2$
$\mathbf{J}_{R_m}$	$2 \times 2$
$\mathbf{J}_{R_p}$	$2 \times 6$
$\mathbf{J}_{C_m}$	$6 \times 2$
$\mathbf{J}_{C_p}$	$6 \times 6$

$\mathbf{J}_C$  and  $\mathbf{J}_R$  indicate output coefficients, constraint coefficients and real relationship coefficients.

A Jacobian matrix is needed to form a relationship between finger joint velocities  $\dot{q}_{fin}$  and measured joint velocities  $\dot{q}_m$ , while passive joint velocities  $\dot{q}_p$  can be considered as redundant information regarding mechanical behavior. In order to simplify Jacobian for a  $2 \times 2$  matrix between  $\dot{q}_{fin}$  and  $\dot{q}_m$ , by constraining  $\dot{q}_p$  using the bottom row of Equation 3.14:

$$\mathbf{J}_{O_p} \dot{\mathbf{q}}_{fin} = \mathbf{J}_{C_m} \dot{\mathbf{q}}_m + \mathbf{J}_{C_p} \dot{\mathbf{q}}_p, \quad (3.15)$$

and define  $\dot{q}_p$  using  $\dot{q}_{fin}$  and  $\dot{q}_m$  as in:

$$\dot{\mathbf{q}}_p = \mathbf{J}_{C_p}^{-1} [\mathbf{J}_{O_p} \dot{\mathbf{q}}_{fin} - \mathbf{J}_{C_m} \dot{\mathbf{q}}_m] \quad (3.16)$$

The expression of  $\dot{q}_p$  can be replaced on the top row of Equation (3.14):

$$\mathbf{J}_{O_m} \dot{\mathbf{q}}_{fin} = \mathbf{J}_{R_m} \dot{\mathbf{q}}_m + \mathbf{J}_{R_p} \dot{\mathbf{q}}_p$$

$$\mathbf{J}_{O_m} \dot{\mathbf{q}}_{\text{fin}} = \mathbf{J}_{R_m} \dot{\mathbf{q}}_{\text{m}} + \mathbf{J}_{R_p} [\mathbf{J}_{C_p}^{-1} [\mathbf{J}_{O_p} \dot{\mathbf{q}}_{\text{fin}} - \mathbf{J}_{C_m} \dot{\mathbf{q}}_{\text{m}}]], \quad (3.17)$$

which can be simplified into overall Jacobian definition:

$$[\mathbf{J}_{O_m} - \mathbf{J}_{R_p} \mathbf{J}_{C_p}^{-1} \mathbf{J}_{O_p}] \dot{\mathbf{q}}_{\text{fin}} = [\mathbf{J}_{R_m} - \mathbf{J}_{R_p} \mathbf{J}_{C_p}^{-1} \mathbf{J}_{C_m}] \dot{\mathbf{q}}_{\text{m}}.$$

$$\begin{aligned} \dot{\mathbf{q}}_{\text{fin}} &= [\mathbf{J}_{O_m} - \mathbf{J}_{R_p} \mathbf{J}_{C_p}^{-1} \mathbf{J}_{O_p}]^{-1} [\mathbf{J}_{R_m} - \mathbf{J}_{R_p} \mathbf{J}_{C_p}^{-1} \mathbf{J}_{C_m}] \dot{\mathbf{q}}_{\text{m}} \\ &= \mathbf{J}_A \dot{\mathbf{q}}_{\text{m}} \end{aligned} \quad (3.18)$$

The Jacobian obtained by Equation (3.18) provides a relation between velocities of measured data and finger joint velocities. A simulation setup was designed as in Figure 3.15 to compare the outputs of the Jacobian to the derivative of finger joints and verify the efficacy of Jacobian matrix with respect to other analyses.

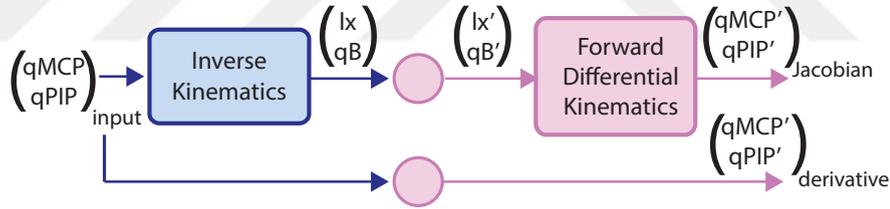


FIGURE 3.15: Simulation setup to compare differential kinematics to simple derivative of a set of finger joint rotation.

In particular, Jacobian calculation was validated by comparing finger joint velocities. To achieve the corresponding sensory measurements  $l_x$  and  $q_B$ , inverse kinematics were run previously. Jacobian matrix can reach numerical values using the orientation of the device, therefore passive joints are extracted from inverse kinematics as well. As a result, finger joint velocities  $(\dot{q}_{o1}, \dot{q}_{o2})$  are compared between Jacobian calculation and derivative of the input. Figure 3.16 shows the comparison between joint velocities obtained through different methods and the error between them show sufficient evidence to assume their unity.

The same Jacobian matrix provides another relation between forces being applied to finger joints and forces being applied by actuators:

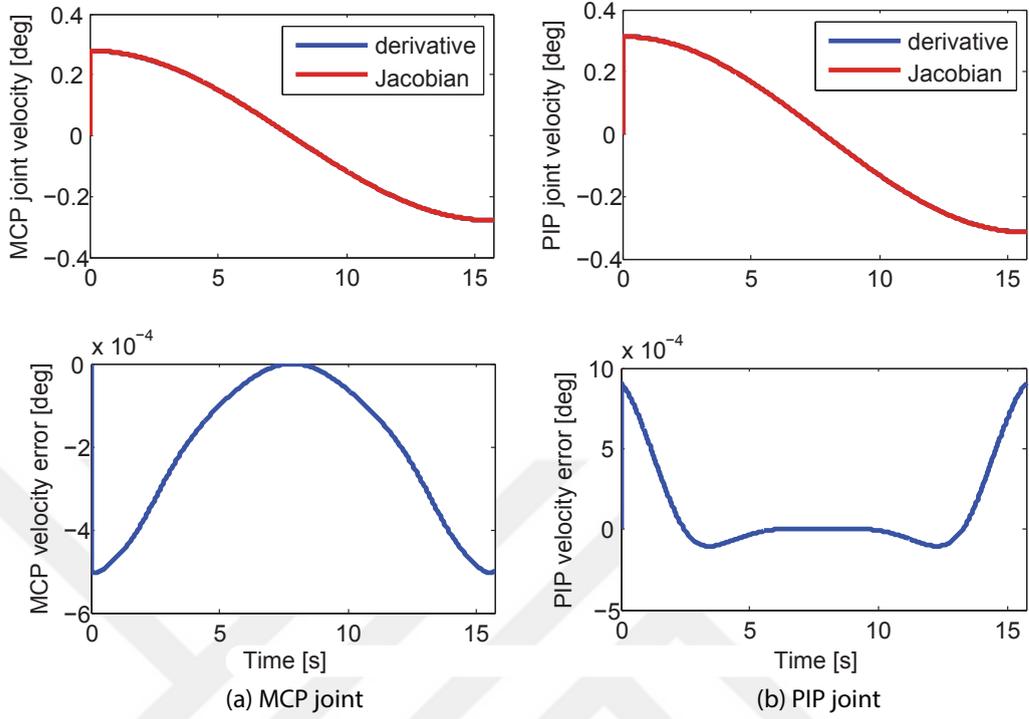


FIGURE 3.16: Comparison between differential kinematics and simple derivative in simulation: (a) comparison for MCP joint rotation and (b) comparison for PIP joint rotation.

$$\begin{bmatrix} F_A \\ \tau_B \end{bmatrix} = \mathbf{J}_A^T \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}. \quad (3.19)$$

where  $\tau_1, \tau_2$  signify torques around the finger joints MCP and PIP correspondingly,  $\tau_B$  signifies output torques around joint  $B$ , and  $F_A$  signifies forces of linear actuator. It is important to know that the device is controlled only by linear actuator while the joint along  $B$  is left passive. The force constraint for the given system by the underactuation concept can mathematically be expressed as  $\tau_B = 0$ . Even though Jacobian matrix might result  $\tau_B$  to reach hypothetical non-zero values, that cannot be provided to the system due to the passivity imposed by underactuation.

### 3.3.5 Statics Analysis and Stability of Grasp

Stability and safety of grasping tasks can be guaranteed only if transmitted forces are applied in the correct direction to provide an interaction between the grasping object

and user's finger. Since the proposed underactuated mechanism does not control forces acting on finger phalanges independently, a static analysis is crucial to ensure stability of grasping tasks at any pose of the mechanism.

The static analysis to ensure stability of grasping forces was already introduced previously for a fully mechanical underactuated gripper [121] through the formulation  $\mathbf{f}_{fin} = \mathbf{J}_T^{-T} \mathbf{J}_A^{-T} \boldsymbol{\tau}_m$ , where  $\mathbf{f}_{fin}$  is the vector of forces between finger phalanges to the grasping object,  $\boldsymbol{\tau}_m$  is the dual force vector of  $\dot{\mathbf{q}}_m$ ,  $\mathbf{J}_T^{-T}$  is the inverse transpose of Jacobian between angular velocities around finger phalanges and linear velocities at the contact points and  $\mathbf{J}_A^{-T}$  is the inverse transpose of Jacobian calculated in Equation (3.18). However, the static analysis in this work aims to control the force transmission on phalanges in terms of finger joint torques  $\boldsymbol{\tau}_{fin} = [\tau_1; \tau_2]$  in order to optimize link lengths of the mechanism. The force transmission is simplified and can be obtained as Equation (3.20) using the inverse Jacobian transpose ( $\mathbf{J}_A^{-T}$ ) in Equation (3.18).

$$\boldsymbol{\tau}_{fin} = \mathbf{J}_A^{-T} \boldsymbol{\tau}_m \quad (3.20)$$

where  $\boldsymbol{\tau}_m = [F_A; \tau_B]$ , and since it can be assumed that  $\tau_B = 0$ , we obtain a direct relationship between actuator force and contact forces. The analysis of sign of  $\tau_1$  and  $\tau_2$  allows us to study stability of grasp assisted by actuator being in contact with an object.

### 3.4 Link Length Optimization

The proposed underactuated hand exoskeleton is designed as a linkage based mechanism and the performance of such mechanisms is known to be affected directly by adjusting their link lengths. Even though the first sketch of the proposed kinematics is drawn with random link lengths, the link lengths of each finger components need to be optimized to improve the overall performance and ensure to satisfy natural RoM of MCP and PIP joints. Yet, the mechanical behavior of the device highlights some physical constraints that need to be satisfied:

- The device is connected to finger phalanges with passive linear sliders. Since these sliders have to be fitted on the finger phalanges, their movements ( $c_1$  and  $c_2$ ) have to be limited by average finger phalange lengths.
- The closing/opening of the hand is performed by transmitting actuator forces to finger joints. To provide a stable grasping, two forces acting on MCP and PIP joints have to be balanced and should be always directed with the same sign, towards opening or closure of the hand.
- The required actuator displacement to satisfy all combinations of natural RoM for finger joints within his physical stroke limitations.

An optimization procedure was conducted by an extensive search procedure to find a set of link lengths that maximizes the following cost function  $p$ :

$$\max p = \sqrt{\tau_1^2 + \tau_2^2}, \text{ such that: } \begin{cases} 0 \leq l_x \leq l_{max} \\ 0 \leq c_1 \leq c_{1max} \\ 0 \leq c_2 \leq c_{2max} \\ 0.25 \leq \tau_1/\tau_2 \leq 7.5 \end{cases}$$

where  $\tau_1$  and  $\tau_2$  are MCP and PIP joint torques,  $c_{1max}$  and  $c_{2max}$  are maximum displacements along passive sliders, which are set as 50 mm and 40 mm, and  $l_{max}$  is the maximum stroke capacity of a linear actuator, which is set as 50 mm due to the actuator choice. The limitation on  $c_{2max}$  might be bigger than the length of middle finger phalange for many users, but the displacement along this joint was found to be crucial

to reach natural RoM for PIP joint. Since there is no mechanical equipment on the third finger phalange to cause a mechanical interference, it does not create a practical issue.

Before the optimization, a set of initial lengths were selected to define a reasonable range of link lengths belonging to the search space, than the following computational steps were followed:

---

**Algorithm 1** Link Length Optimization Algorithm

---

- 1: \* Define a feasible and wide link length range
  - 2: **for**  $do_{q_{o1}} = 0 : 1 : 80^\circ$
  - 3:     **for**  $do_{q_{o2}} = 0 : 1 : 90^\circ$
  - 4:         Calculate the movement on the passive prismatic joints  $c_1$ ,  $c_2$  and actuator  $l_x$ 
    - Control if  $l_x$ ,  $c_1$  and  $c_2$  satisfy the physical limits
    - Go to the next set and start from 2 if not satisfied.
  - 5:         Compute the torques on the finger joints if 1  $N$ . force is applied from the actuator using Jacobian transpose
    - Control if the ratio between the torques of two joints satisfy the predefined limits
    - Go to the next set and start from 2 if not satisfied.
  - 6:         Calculate the optimization objective  $p = \sqrt{\tau_1^2 + \tau_2^2}$
- 

For the sake of simplifying the search space of the optimization, a sensitivity analysis was conducted to identify the link length variables that do not affect the optimization procedure significantly and so reduce the dimension of the search space.

### 3.4.1 Sensitivity Analysis

Although numerical or analytical derivatives of the overall cost function  $p$  with respect to each search parameter would provide an efficient sensitivity analysis approach, the derivatives are not easy to obtain for complex non linear models. The one-at-a-time (OAT) sensitivity analysis is an alternative method for analyzing the effect of a single parameter on a cost function, keeping other parameters fixed [130]. For this purpose, a sensitivity index ( $SI$ ) is expressed in Equation (3.21).

$$SI = \frac{\frac{S_2 - S_1}{S_{av}}}{\frac{E_2 - E_1}{E_{av}}} \quad (3.21)$$

where  $SI$  is the sensitivity index of the model output,  $E_1$  and  $E_2$  are minimum and maximum values of input parameters;  $S_1$  and  $S_2$  are corresponding output values for  $E_1$  and  $E_2$ ;  $S_{av}$  and  $E_{av}$  are the average values of input and output parameters respectively. This index provides a quantitative relation between model outputs and input variables in terms of sensitivity. Negative  $SI$  indicates that the inputs and outputs vary in opposite directions, while positive values signify a change in the same trend. In particular, the aim is to choose the positive, higher than unity  $SI$  values, which show that a change in the parameter creates a higher effect in the output.

While closing/opening human finger, the most crucial output values were stated as displacements along passive linear sliders ( $c_1$  and  $c_2$ ) due to the limitations imposed by finger phalange lengths. Therefore, the impact of each variable on these values was investigated individually. A representative pose of finger was used for this analysis. During the sensitivity analysis, each input variable was changed by  $\pm 10\%$  from the initial value, keeping other variables constant. For each set of lengths, the variations of linear displacements  $c_1$  and  $c_2$  were calculated using the pose analysis discussed in Section 3.3. The sensitivity index was calculated individually for  $c_1$  and  $c_2$  where the analysis limit is set to  $+10\%$  for both cases. In other words, the length parameters, which result in negative or  $\leq 0.1$  SI values, are not considered as effective variables over the performance of such linear movement. Figure 3.17 represents sensitivity analysis results for  $c_1$  and  $c_2$  displacement individually, calculated as  $SI_{c_1}$  and  $SI_{c_2}$ . Note that the bars with values under 0.001, for instance, values for  $L_{AB}$ , cannot be seen in the plot.

It can be easily observed that for some variables, the sign of SI values are different for  $c_1$  and  $c_2$  displacements. When it happens, it means that increasing a link length affects sliders along first and second finger phalanges in different ways. Equation (3.22) combines SI values computed for both sliders.  $SI_g$  computed from Equation (3.22) with positive values indicates similar output effect on both sliders, while the one with negative values indicate different behaviours. Table 3.8 presents output values of  $SI_g$  from Equation (3.22) for each link length variation.

$$SI_g = \text{sign}(SI_{c_1}) \cdot \text{sign}(SI_{c_2}) \sqrt{SI_{c_1}^2 + SI_{c_2}^2} \quad (3.22)$$

From the sensitivity analysis, the most efficient variables are obtained as  $L_{EJ}$ ,  $L_{CI}$ ,  $L_{EF}$ ,  $L_{ED}$ ,  $L_{CD}$  and  $L_{BC}$  as can be seen in Table 3.8. It is important to note that, the negative values represent different impacts on  $c_1$  and  $c_2$  slider displacements as a

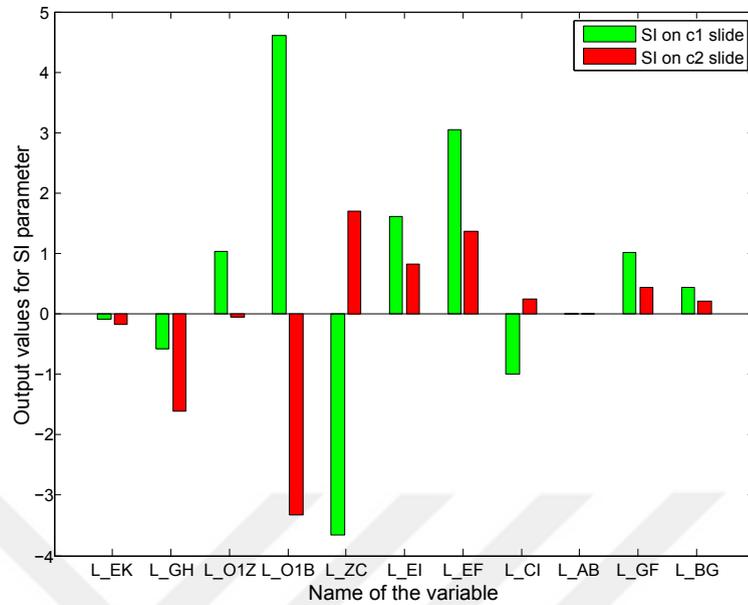


FIGURE 3.17: Computation of sensitivity index for first passive slider movement  $c_1$  and second passive slider movement  $c_2$  independently.

TABLE 3.8: Results of generic sensitivity index.

variable	value [mm]	variable	value [mm]
$SI_{EJ}$	0.1939	$SI_{ED}$	3.3437
$SI_{CI}$	1.7077	$SI_{GF}$	-1.0275
$SI_{KH}$	-1.0298	$SI_{AB}$	-2.1864e-09
$SI_{KB}$	-5.6804	$SI_{CD}$	1.1021
$SI_{GH}$	-4.0324	$SI_{BC}$	0.4877
$SI_{EF}$	1.8146		

result for increased link length. These lengths are excluded from parameter search space since minimizing two of the slide displacements simultaneously is not feasible. These constant values of 6 variables for index, middle, ring and little fingers were set constant as reported in Table 3.9 - Table 3.12 during optimization.

TABLE 3.9: Constant variables for the index finger.

variable	range [mm]	variable	range [mm]
$L_{KH}$	72	$L_{KB}$	35
$L_{GH}$	86	$L_{AB}$	18
$L_{GF}$	46		

TABLE 3.10: Constant variables for the middle finger.

variable	value [mm]	variable	value [mm]
$L_{KH}$	72	$L_{KB}$	35
$L_{GH}$	86	$L_{AB}$	18
$L_{GF}$	46		

TABLE 3.11: Constant variables for the ring finger.

variable	value [mm]	variable	value [mm]
$L_{KH}$	72	$L_{KB}$	35
$L_{GH}$	86	$L_{AB}$	18
$L_{GF}$	46		

### 3.4.2 Pre-optimization Constraints

A preliminary optimization procedure was conducted to select the combinations of link length parameters, satisfying displacement and static constraints among link lengths for index, middle, ring and little fingers as ranged in Table 3.13 - Table 3.16. The variables, which were chosen after the sensitivity analysis, were iterated with a difference of 1 *mm* within the given range throughout the optimization. For the pre-optimization procedure, MCP and PIP joints are moved in different paths for the corresponding iteration set to

TABLE 3.12: Constant variables for the little finger.

variable	value [mm]	variable	value [mm]
$L_{KH}$	72	$L_{KB}$	35
$L_{GH}$	86	$L_{AB}$	18
$L_{GF}$	46		

check whether linear and static constraints are satisfied. The length combinations are eliminated from the optimization performance if the constraints above are not satisfied.

TABLE 3.13: Range of variables for index finger optimization.

variable	range [mm]	variable	range [mm]
$L_{EJ}$	30 - 48	$L_{ED}$	35 - 45
$L_{CI}$	16 - 20	$L_{EF}$	20 - 35
$L_{CD}$	9 - 20	$L_{BC}$	36 - 46

TABLE 3.14: Range of variables for middle finger optimization.

variable	range [mm]	variable	range [mm]
$L_{EJ}$	30 - 48	$L_{ED}$	40 - 55
$L_{CI}$	16 - 20	$L_{EF}$	15 - 30
$L_{CD}$	9 - 20	$L_{BC}$	36 - 46

TABLE 3.15: Range of variables for ring finger optimization.

variable	range [mm]	variable	range [mm]
$L_{EJ}$	30 - 48	$L_{ED}$	35 - 45
$L_{CI}$	16 - 20	$L_{EF}$	20 - 35
$L_{CD}$	9 - 20	$L_{BC}$	36 - 46

TABLE 3.16: Range of variables for little finger optimization.

variable	range [mm]	variable	range [mm]
$L_{EJ}$	30 - 48	$L_{ED}$	30 - 40
$L_{CI}$	16 - 20	$L_{EF}$	20 - 35
$L_{CD}$	9 - 20	$L_{BC}$	36 - 46

### 3.4.2.1 Linear constraints

A set of link lengths should not cause displacements of passive linear sliders,  $c_1$  and  $c_2$ , to exceed limitations set for them based on finger phalange lengths. With this motivation, inverse kinematics in Section 3.3 should calculate variations of  $c_1$  and  $c_2$  as different joint trajectories are given as input to the kinematics analysis. A similar linear constraint should be set based on the required displacement for actuator to reach required MCP and PIP joints. The limitation of maximum displacement is set based on the properties of chosen actuator. An iteration is passed to the next step if both linear constraints are satisfied at all times, but is aborted from the overall optimization procedure and a new iteration is initialized with a new set of link lengths. Within the given range, 65 % of the iteration sets were eliminated by the linear constraints of the index finger.

### 3.4.2.2 Static constraints

If the set of link lengths satisfy previous linear constraints, then the static constraints should be ensured during the iteration. Assuming that actuator applies 1 *N*. linear force, torques around MCP and PIP joints,  $\tau_1$  and  $\tau_2$ , should be computed using Jacobian matrix in different mechanical configurations. A static constraint was set to ensure to have a balanced ratio between torques around MCP and PIP joints  $\tau_1/\tau_2$  in different orientations of finger. Such a ratio was suggested to be limited by a minimum value of 0.25 and a maximum value of 7.5 without considering contact forces acting on the finger from the environment. Similarly, an iteration is passed to the next step if the static constraint is satisfied at all times, but is aborted from the overall optimization procedure and a new iteration is initialized with a new set of link lengths. Setting the statics constraint eliminates 90 % of the remaining sets of variables for index finger.

### 3.4.3 Optimization

To enlarge the efficient workspace of finger joints, previous constraints were controlled repeatedly to explore a workspace up to  $80^\circ$  for MCP joint and  $90^\circ$  for PIP joint. After the pre-optimization selection, where physical constraints were ensured to be satisfied, an optimization by exhaustive search was conducted. With this motivation, each successful iteration, which was not interrupted with unsatisfied constraints, was finalized by calculating the performance cost function  $p$  such that  $p = \sqrt{\tau_1^2 + \tau_2^2}$ . In order to provide a better understanding of the difference between the calculated cost function  $p$  for each parameter set, Figure 3.18 shows  $p$  values over the variable sets during index finger optimization. The  $p$  values were ordered to observe a constant increase, revealing how different the performance index can be calculated. It can be observed that the choice of parameters can increase performance by %50. Once the iterations are finalized within the proposed range of variables, the set of link lengths with the highest performance index  $p$  is selected as the final set to implement the hand exoskeleton.

Table 3.17 - Table 3.20 report the resulting link lengths of the hand exoskeleton components for the index, middle, ring and little fingers independently. Note that the range of variables for the links  $L_{GH}$  and  $L_{GF}$  could not be enlarged more to avoid any possible mechanical interference.

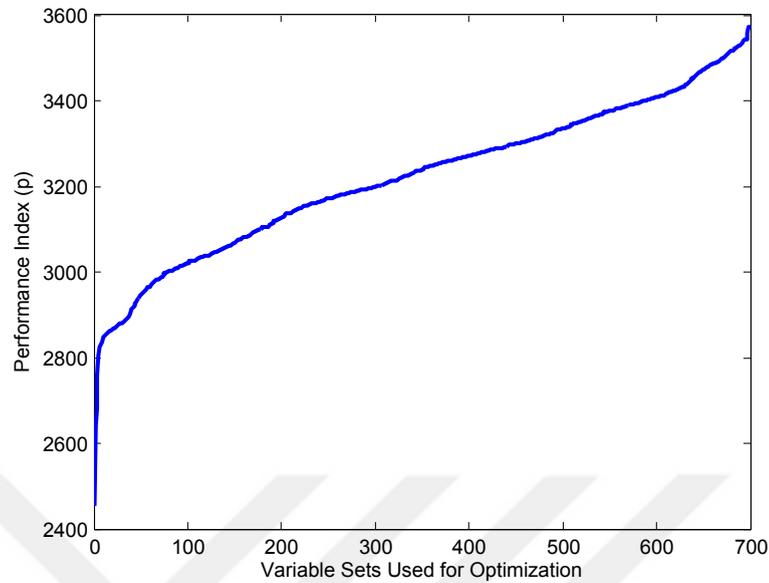


FIGURE 3.18: Computation of performance index  $p$  for different sets of link lengths, which satisfy the physical linear and statistical constraints, for index finger component optimization.

TABLE 3.17: Results of optimization for index finger.

variable	value [mm]	variable	value [mm]
$L_{EJ}$	39	$L_{ED}$	40
$L_{CI}$	16	$L_{EF}$	27
$L_{CD}$	9	$L_{BC}$	43

For a middle-sized hand, the optimized link lengths allow finger joints to reach a workspace stated in Table 3.21 for index, middle, ring and little fingers. In fact, Figure 3.19 represent the index finger workspace of the end of middle phalange to test RoM for MCP and PIP joints only in a natural manner and with physical constraints that are set by the exoskeleton. Even though there are some extreme points that a user cannot reach wearing the exoskeleton, % 95 of the natural workspace of index finger was covered under the exoskeleton assistance.

Figure 3.20 shows the CAD design of index finger component in maximum flexion and

TABLE 3.18: Results of optimization for middle finger.

variable	value [mm]	variable	value [mm]
$L_{EJ}$	39	$L_{ED}$	52
$L_{CI}$	17	$L_{EF}$	21
$L_{CD}$	9	$L_{BC}$	41

TABLE 3.19: Results of optimization for ring finger.

variable	value [mm]	variable	value [mm]
$L_{EJ}$	39	$L_{ED}$	38
$L_{CI}$	16	$L_{EF}$	29
$L_{CD}$	10	$L_{BC}$	42

TABLE 3.20: Results of optimization for little finger.

variable	value [mm]	variable	value [mm]
$L_{EJ}$	42	$L_{ED}$	32
$L_{CI}$	16	$L_{EF}$	23
$L_{CD}$	9	$L_{BC}$	43

extension configurations of a user. Similarly, the overall hand exoskeleton in full flexion and full extension is presented in Figure 3.21 without the thumb. It is useful to highlight that the thumb component for a full hand exoskeleton is left as an independent work due to complexity and diversity of the thumb kinematics compared to rest of the hand.

The device is optimized considering the measurements of a middle-sized hand, where

TABLE 3.21: Ranges of motion for all finger joints with hand exoskeleton

Joint	MCP	PIP	DIP
RoM	0 - 80	0 - 90	-

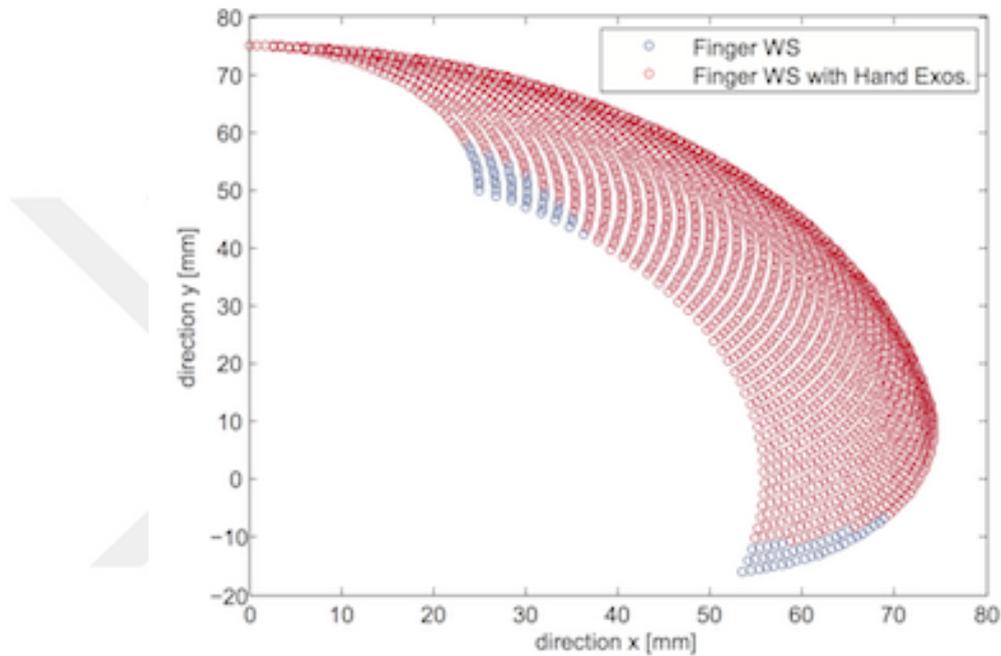
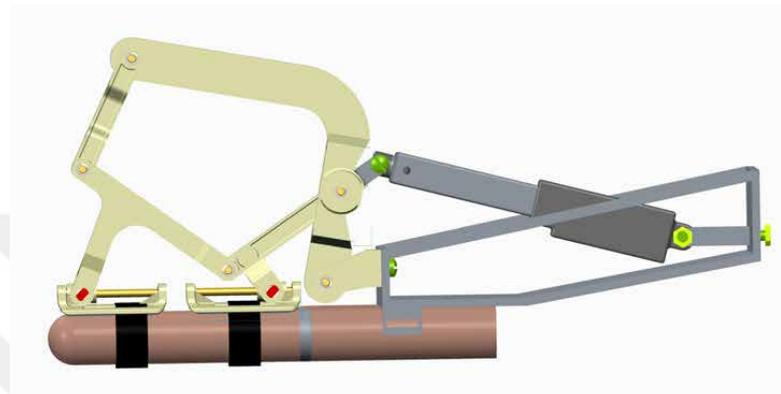
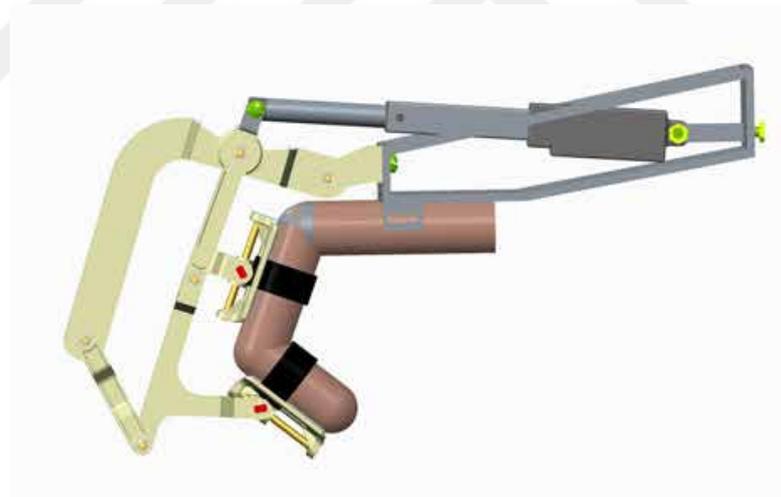


FIGURE 3.19: Comparison between anatomical finger workspace and workspace under the assistance of the proposed hand exoskeleton using only the rotation of MCP and PIP joints.

index finger has the measurements of  $50\text{ mm}$  and  $30\text{ mm}$  for first and second finger phalanges. However, it is observed that the device with selected link lengths satisfy physical constraints and provide an efficient force transmission also for a small-sized hand (index finger of  $45\text{ mm}$ ,  $27\text{ mm}$ ) and big-sized hand (index finger of  $55\text{ mm}$ ,  $34\text{ mm}$ ), which cover a wide range of users in the society. Still, it is important to note that assisted RoM of finger joints stated in Table 3.21 tend to change for different hand sizes.

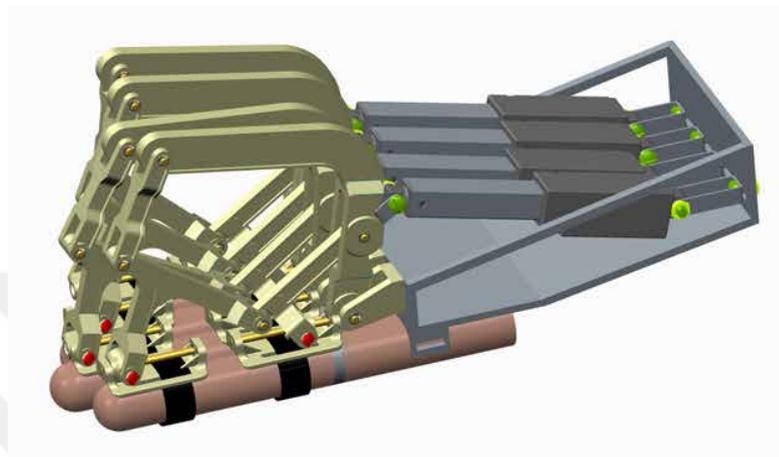


(a) Full flexion

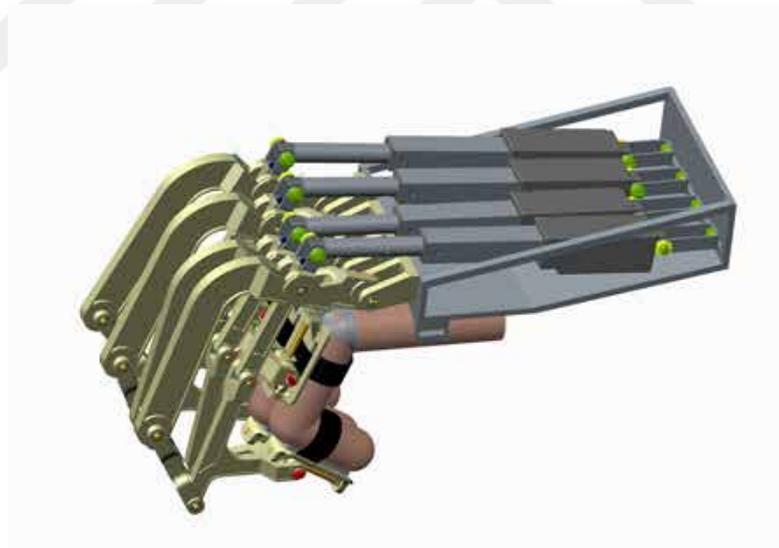


(b) Full extension

FIGURE 3.20: Maximum flexion and extension pose for index finger component of the proposed hand exoskeleton.



(a) Full flexion



(b) Full extension

FIGURE 3.21: Maximum flexion and extension pose for the proposed hand exoskeleton with index, middle, ring and little finger components.

## Chapter 4

# Implementation of the Hand Exoskeleton

After the design, the kinematics calculation and the link length optimization performed in Chapter 3, we will detail the physical implementation and the design of control board and electronic system for the proposed hand exoskeleton. The electronic board mostly focuses on to things: reading sensors (the actuator displacement, additional passive joint rotation, force, pressure, etc.) and drive the actuators based on different control strategies (position control, admittance control, teleoperation, etc.). First, we will detail the design of this electronic board, and then, we will present the 3-D printed hand exoskeleton. Finally, various control algorithms, such as position control with enhanced performance tools, active control through muscular activity using EMG measurements and active control through active backdriveability using additional force sensor will be detailed with feasibility experiment results. The hand exoskeleton for 4 fingers are manufactured using the kinematics and link lengths presented, but most of the control results are performed only using index finger component for simplicity. Control algorithms that are presented here can easily be utilized for rehabilitation exercise scenarios, as much as assistive applications.

## 4.1 Control Board

Designing the hand exoskeleton requires an electronic control hardware to be designed to read sensor measurements coming from position sensors attached to the exoskeleton, to convert analog measurements into digital, to run control algorithms to drive actuators in a stable manner, to use motor drivers to command actuators and to communicate with external environments, such as a host computer, a virtual environment, etc.

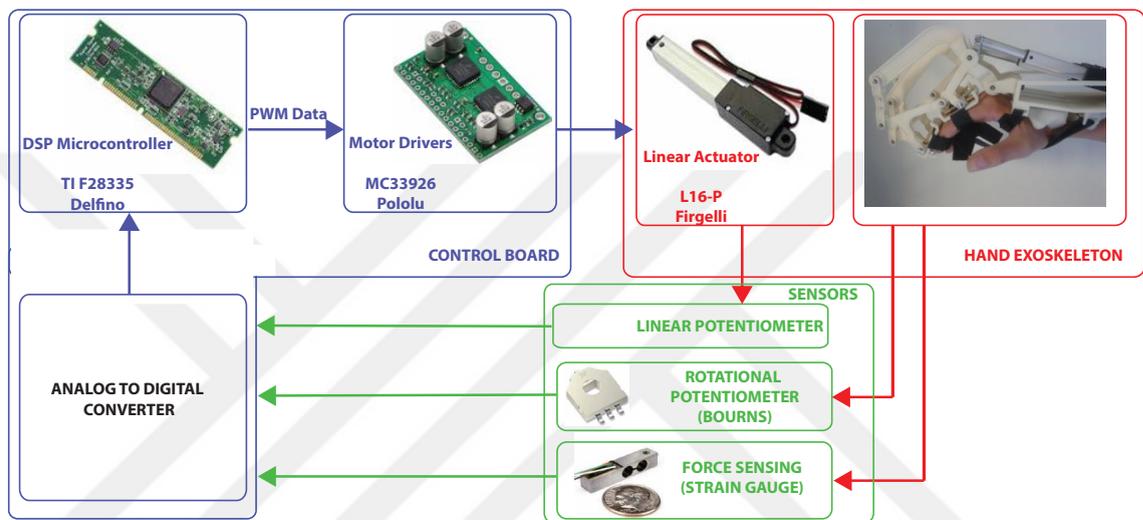


FIGURE 4.1: General scheme of electronics components chosen for the actuation and control of the proposed hand exoskeleton: a DSP microcontroller has a control algorithm creating a PWM data to send to the motor driver. This motor driver converts the PWM signal to the current to be given to the linear actuator. As the linear actuator moves, the linear potentiometer inside the actuator and the additional potentiometer measures the movements on the exoskeleton. Furthermore, the force sensor attached on the exoskeleton measures how much force the user applies to open/close his fingers. All of these measurements are converted to digital to be used as feedback by the control algorithm.

Figure 4.1 summarizes the electronics hardware of the proposed exoskeleton. In particular, a DSP microcontroller board (TI F28335 Delfino) was chosen to track sensors, to turn signals to digital and produce Pulse-width Modulation (PWM) signals to run the motors through the Pololu motor driver carrier. The Pololu motor drivers (MC33926) read PWM signals and create corresponding current to drive selected linear actuators (Firgelli, L16-50-P). As the linear actuator pushes the exoskeleton to open/close fingers, its internal potentiometer measures the movement and sends it to the control board. Alternatively, an additional rotational potentiometer will be put over one of the passive mechanical joints to reach a unique finger pose instantaneously and read by the control

board through analog pins. Finally, a  $1DoF$  force sensor is used for each finger component to measure user's intentions to open/close his finger and run active backdriveability through control. Similarly, force measurements are read by the control board through analog pins. The blocks represented in

#### 4.1.1 Actuators

We know that the actuator directly changes the mechanical design of the proposed exoskeleton. This is why we have to decide the actuator as a first step of the electronics design. Independent finger control can be achieved by implementing a single actuator for each finger component individually. Furthermore, a generic hand exoskeleton should be as portable as possible, such that being attached to a stationary computer through a single cable without turning the overall device into a bulky, heavy design. The portability requires all 4 linear actuators, without the thumb component, to be placed on top of user's hand. By choosing miniature linear actuators, we can place all actuators in a compact manner, while providing sufficient forces for fingers to perform ADLs or rehabilitation exercises.

Even though there are certain linear actuators with similar stroke capabilities and output forces without a screw mechanism to amplify effective forces. Not having a screw mechanism achieves backdriveability, allowing users to manipulate the actuator by applying forces. However, they can reach high forces using magnetic components, which might be problematic if a certain distance between two actuators cannot be satisfied. This distance requirement would prevent the independent finger implementation in a portable manner, so they are found impractical for this study. Instead, we chose Firgelli actuators, since they are low-cost, practical and efficient in terms of output forces to perform grasping tasks assisted by the hand exoskeleton.

Table 4.1 details the specific properties of Firgelli  $P16 - 50 - 22 - 12$  actuator. It has a rotary DC electromagnetic motor actuating a screw mechanism, defining a gear ratio to amplify the output forces. Such a gear box ratio provides effective output forces with minimized actuator box. The screw mechanism requires high backdrive forces such that intrinsic backdriveability is not allowed when actuators are off. Even though the finger components are manufactured with minimum friction, the backdriveability of the overall system is inhibited.

### 4.1.2 Control Electronics and Motor Drivers

The chosen Firgelli actuators are driven through Free scale MC33926 H-Bridge IC, mounted on a Pololu Carrier Board. Each driver board drives two DC actuator, so we need 2 boards to control 4 actuators, which would simplify the electronics design overall. The control board sends PWM signals to the control inputs of the driver board at 25 *KHz* frequency through PWM peripheral.

A microcontroller unit was implemented initially to read actuator displacements and create appropriate PWM signals using control strategies using these measurements. The DSP microcontroller is also used to compute kinematics analyses, communicate with a host PC and read complementary sensors for pose estimation or force control techniques. The Texas Instruments Delfino F28335 DSP microcontroller board was chosen, featuring a core with floating point unit, ADC and PWM peripherals and USB communication with the host PC. It also supports rapid prototyping of the control algorithm through MATLAB Simulink: the Simulink model can access the hardware peripherals and can be compiled, flashed and executed as a stand-alone program on the DSP board.

Figure 4.2 summarizes the step size and the data type that are used within the control board. Serial send/receive blocks set a USB communication between the board and the

TABLE 4.1: Specifications of Firgelli L16 linear actuator.

Property	Value
Actuator Mass	$\cong 56 \text{ g.}$
Actuator Stroke	50 <i>mm.</i>
Motor gear ratio	35 : 1
Max. cont force of the motor	40 <i>N</i>
Backdrive force	31 <i>N</i>
Max. velocity of the output shaft	32 <i>mm/s</i>
Positional Error Max	300 $\mu\text{m}$
Potentiometer Impedance	9 <i>k<math>\Omega</math></i>
Rated Voltage	12 <i>VDC</i>
Maximum Current Consumption	1 <i>A</i>

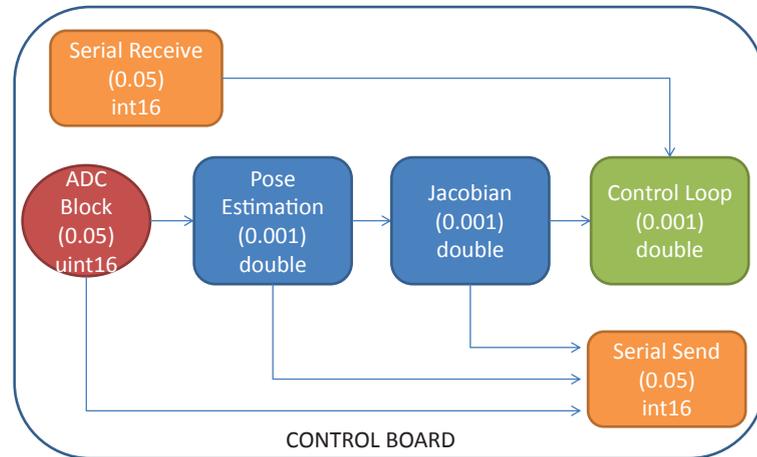


FIGURE 4.2: Sample time and data type definitions for different operations performed in DSP control board or communication network with the host computer.

host model, and we limit this communication speed to 0.05 s. Pose estimation, Jacobian calculation and control algorithms are computed at 0.001 s. Resulting PWM values are fed to actuator with 0.0005 s to achieve smoother movements. The data communication is made as *int16* type and are converted to *double* right after receiving them or right before sending them.

### 4.1.3 Sensors

The most essential sensory measurements of the exoskeleton come from the linear potentiometers attached to Fircelli actuators. These measurements are used for closed-loop control algorithms to form the feedback or to estimate finger pose during operation. Its accuracy and speed are crucial to ensure safety and stability of the overall system. Each actuator potentiometer provides an analog position feedback signal. Minimum and maximum output values of these measurements are set as 0 – 50 mm by mechanical properties of the actuator.

In Section 3.3.2, we mentioned the need of additional sensor from the mechanism and how we can reach a unique forward kinematics solution by implementing an additional potentiometer to measure  $q_B$  (see Figure 3.5). A low-cost, lightweight potentiometer (BOURNS 3382G-1-103G) were chosen to be implemented in a simple and lightweight manner. Each potentiometer is DC powered (5 V) and the cursor voltage is read through

the ADC peripheral of the microcontroller board. The potentiometer was chosen specifically with a hole through property in order to fix it on the mechanism using existing pins. The potentiometer can read measurements between  $0 - 330^\circ$ .

Table 4.1 states the gear box ratio (35 : 1) and high backdrive force (31  $N$ ) about the chosen Firgelli actuator. This internal friction prevents the user to initiate a physical movement manually. Therefore, achieving active tasks can be ensured only through force control techniques. A force sensor was implemented externally for each finger component to measure interaction forces between the device and the user. *1DoF*, miniaturized load cell with strain gauges (*S215LoadCell* by SMD Sensors) is mounted in series between the actuator and the main body of the exoskeleton for each finger component. The SMD load cell has small dimensions ( $26 \times 6 \times 6mm$ ) and high repeatability (0.01%) with a maximum measurable range of force of 18  $N$  (safe overload of 36  $N$ ). Each load cell is connected to a proprietary electronic board (*ZSC31050* by SMD Sensors) integrating a wheatstone bridge, and a signal amplification and conditioning stage. The analog output signal is regulated between  $0 - 5 V$  and then is read through ADC peripheral of the microcontroller board.

Finally, additional force sensing resistors (FSRs) are occasionally used to measure interaction forces, either between user's fingers and grasped objects, or between device connectors and user's fingers. These sensors help us understand the force transmission between finger phalanges as much as the comparison between resulting forces along finger phalanges and actuator force. FSR 402 (Interlink Electronics Inc.) sensors have been used, with a circular active area measuring 12,5  $mm$  in diameter and a sensitivity range from less 1 – 1000  $N$ , depending on the electrical setup. The sensors were coupled to a fixed resistor of 2.2  $k\Omega$  according to a voltage divider set up. This configuration allows to measure forces up to 12  $N$ .

The FSR sensor was calibrated using precision weights for the chosen fixed resistance values. The output of FSR is dependent on the surface area, so we need to design the contact area carefully. An additional interface should be placed on top of the FSR sensor to minimize the contact dependency, to obtain comparable results within different experimental tasks, and to increase repeatability of applied forces. Within this study, these additional FSR sensors were used in various configurations, so that their placement will be covered in the following sections whenever utilized. Since these sensors are not essential for each task and they are not required for the control loops, their readings are measured through an additional ARDUINO board with serial USB connection to the host computer.

## 4.2 Multi-Finger Exoskeleton

The first prototype of the underactuated hand exoskeleton was implemented with rapid prototyping parts, allowing the device to be low-cost and light-weight. Table 4.2 presents mechanical specifications of the device, actuated by Firgelli *L16* linear actuator. The link lengths for finger components were found using the optimization algorithm in Section 3.4.

TABLE 4.2: Specifications of the proposed exoskeleton and its actuators.

Property	Value
Number of Assisted Fingers	4
Device Mass	$\cong 400$ g.
RoM for MCP	$80^\circ$
RoM for PIP	$90^\circ$
Max. torque on MCP	1485 Nmm
Max. torque on PIP	434 Nmm

The device is attached to the user from their palm and their wrist, to stabilize the base and from finger phalanges, in order to stabilize finger components. Connecting the device using simple Velcro straps allows the device to be worn in about a minute, without any initial pose requirement of fingers. Actuators are attached on the base of the device in a compact manner. The control board is placed on top of a table, near a host computer, but there is no limitation regarding cable sizes. Figure 4.3 shows the hand exoskeleton that is connected to a user through index, middle, ring and little finger components, whose kinematics and development is detailed in this study and thumb component, whose development is done individually, even though the idea behind its mechanical design is kept the same.



FIGURE 4.3: Implementation of hand exoskeleton with 5 finger components, including the thumb, which is the product of an individual research study.

### 4.3 Control Algorithms

The rest of this section details passive and active control algorithms implemented for the assistive and the rehabilitative experiment scenarios. First, a simple position control will be implemented to open/close fingers individually. Underactuation property of the device allows users to perform rehabilitation exercises and real grasping tasks for assistive exercises by adjusting its behavior automatically based on interaction forces, even using a simple, strict position control. The passive position control can be combined with active sensors to measure user's intention. The first active task is to estimate user's intention through EMG sensors, which detect the muscular activity from the healthy or unassisted arm of the user, and open/close fingers based on such estimation as a position reference for the exoskeleton. The second approach is to run a closed-loop force control, by measuring the interaction forces between the user's finger and the device's base, and by driving actuators to make these measurements follow a reference. Such a force control algorithm can be used only as backdriveable task for rehabilitation or assistive tasks, or as stiffness rendering task for rehabilitation and haptic applications.

### 4.3.1 Position Control

The mechanical gearbox inside the linear actuator increases the internal friction and requires a backdrive force to overcome before starting the motion. Such a backdrive force not only prevents the actuator to be moved by applying forces to the shaft manually, but also affects the control performance adversely by creating a delay between the PWM signal and the corresponding movement. Figure 4.4 presents the actuator displacement over the input PWM signal and shows that the PWM input can produce movement after 37.2 %, which might be the corresponding PWM input for the backdrive force.

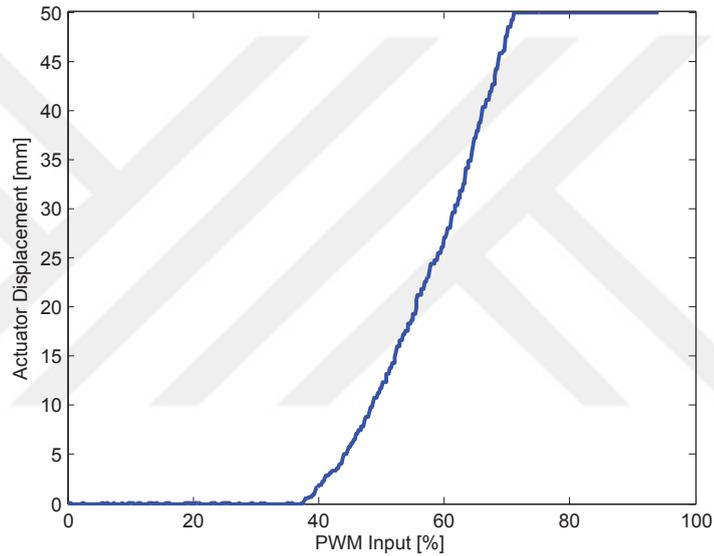


FIGURE 4.4: Observed actuator displacement with respect to the PWM percentage input to observe the impact of the high backdrive force of the actuator.

Actuator displacement cannot be achieved for the PWM input values smaller than 37.2 %. From the control point of view, such a limitation can be defined as a dead spot between  $\pm 37.2$  %, which can be compensated by supplying  $\pm 40$  % to the motor drivers besides the output of the controller algorithm. The sign of such a compensation is adjusted with respect to the sign of the displacement error ( $l_{x_{er}} = l_{x_{ref}} - l_{x_{act}}$ ). PWM input to determine the actuator forces can be calculated as  $F_{thr} = \text{sign}(l_{x_{er}})(40)$  allowing the control loop to have a better sensitivity in a stable manner. Meanwhile, the PWM input can be calculated by the controller strategy as  $F_{cont} = K_P l_{x_{er}} + K_I \int l_{x_{er}}$ , where  $K_P$  and  $K_I$  are proportional and integral control gains. In the end, the PWM signals being supplied to the motor drivers are calculated as the sum of both components

as  $F_{PWM} = F_{thr} + F_{cont}$ . Figure 4.5 presents the algorithm to compensate the friction PWM and the simple PI Control scheme.

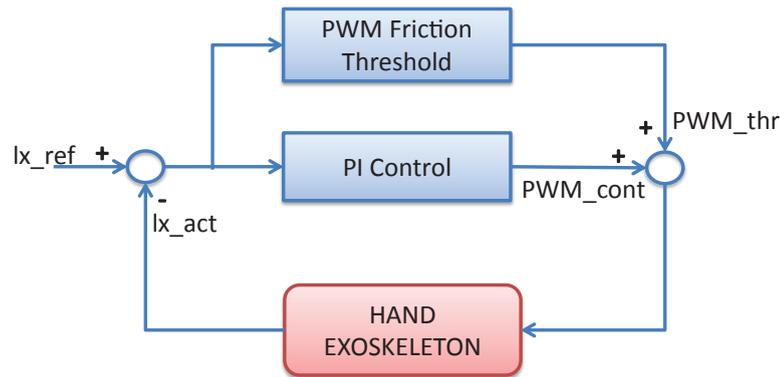
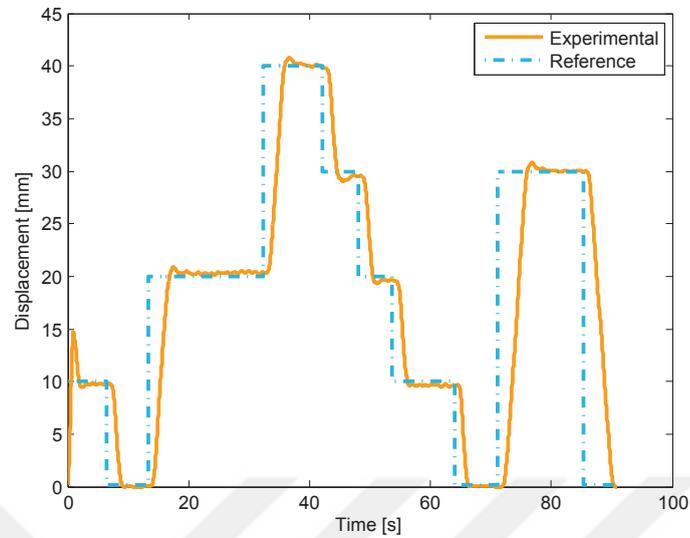


FIGURE 4.5: Adjusted position control algorithm with additional PWM supply in the same direction of control algorithm to overcome internal actuator friction.

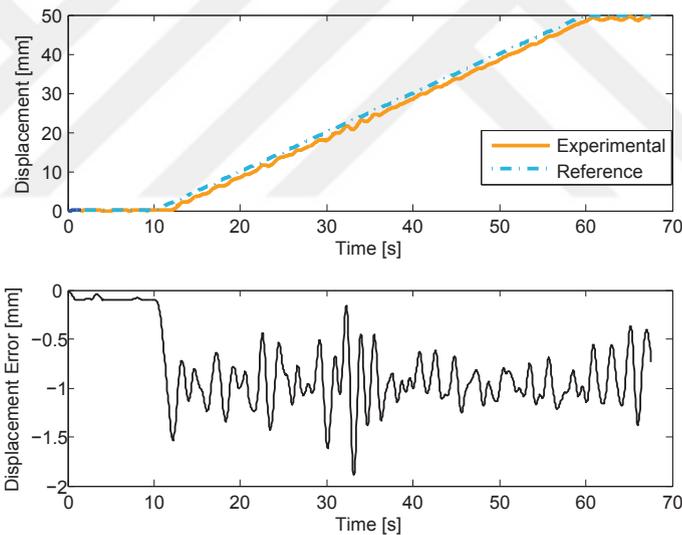
The exoskeleton is controlled by a simple PI control algorithm to minimize the steady state error without calculating the derivative of real time measured sensors. The lack of derivative component allows the overall mechanical system to use simpler and cheaper encoders. The integral component allows the controller to eliminate the steady state error, while affecting the response time of the overall system. However, since the required system speed is much slower than the controller speed itself, response time drawback is not a critical issue.

Figure 4.6 shows the results of position control tasks. Figure 4.6(a) shows the experiment result, where the displacement reference was given randomly and discretely to observe the settling time of the hand exoskeleton with the control scheme presented in Figure 4.5. From the figure, it can be observed that the finger component of the exoskeleton can open/close fingers in a few seconds easily. Both the response time and the steady state error can be observed to allow the control algorithm to run in an efficient and stable manner.

On the other hand, a ramp reference can show the following abilities of the chosen control gains. Figure 4.6(b) shows that the selected controller gains allow the actuator to follow a given continuous trajectory successfully. The displacement error seems to be within 2 mm for this experiment, This error range is quite acceptable for assistive or rehabilitative exercises, because they focus on the overall grasping tasks, and do not need precise control.



(a) Position control with step input.



(b) Position control with ramp input.

FIGURE 4.6: Simple PI control results with the step and the ramp input references.

Besides simple PI control, further control algorithms can be used to open/close fingers in a passive manner. Nevertheless, having adjustable mechanism for various tasks in the environment based on interaction forces does not necessarily require further, more complex controllers to be developed, as long as stability of the controller and safety of the user are guaranteed.

### 4.3.1.1 Temperature filter

The underactuated hand exoskeleton is designed predicting all the possible physical interference during operation. Mechanical precautions are performed to eliminate such interference, but due to different hand sizes and utilization of Velcro straps, we might still need further measures to minimize their negative impacts. For instance, Velcro straps, which are used to connect the device with user's fingers, might create some clutter and get stuck with each other. Especially while controlling finger components individually, such clutter might create a contradiction between the actuator movement and its reference. On such an occasion, insistent current fed to the actuators might harm the mechanical components, specifically the actuator itself. Such an incident can be prevented by limiting the current supply to the actuator through the PWM values at the controller, simply by lowering the PWM values based on time, preventing actuators to heat and giving enough time for the actuators to cool down.

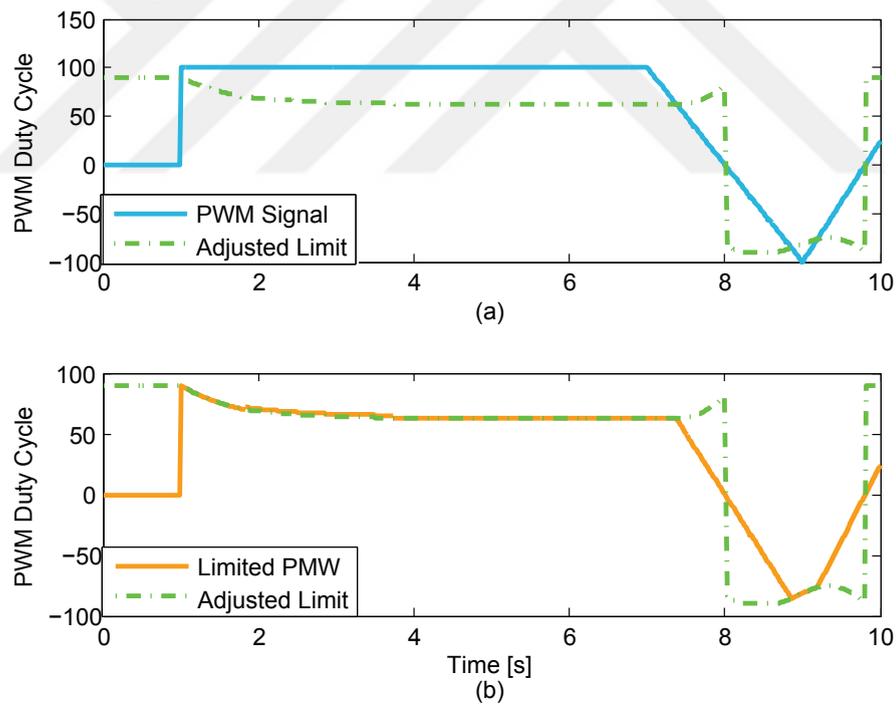


FIGURE 4.7: Simulation plot of how the temperature filter works: (a) adjusted limitation set for the input PWM signal based on the PWM signal values over time, and (b) output PWM signal to be supplied for actuator compared to the PWM limitation.

Figure 4.7 shows the implementation of the proposed filter. A maximum and minimum values of the PWM duty cycle was defined, such that the controller can deliver PWM signals within those limits to the motor drivers. When the original PWM value is

below the minimum limit, the PWM values are directly sent to the motor drivers to run the motors, while the adjusted limit is fixed to the maximum limit. As the PWM value reaches the maximum limit, and continue exceeding the maximum value; the adjusted limit starts to get decreased over time until reaching the minimum limit, as the PWM values supplied to the motor drivers are saturated through these adjusted limits. Whenever the original PWM values start to reach lower values than the minimum limit, the adjusted limit starts to increase until the maximum limit. In the meantime, the motor drivers are supplied the original PWM values. Such a temperature limit should be performed in a bidirectional manner. The minimum and the maximum PWM limits for this study were selected experimentally as 90 % and 60 %, since the PWM values lower than 60 % slows down the speed significantly. Doing so, even if the actuator gets stuck, the control unit of the hand exoskeleton will force actuators to perform the desired task using 60 % of duty cycle. This might extend the life span of the actuators. Similarly, Figure 4.8 shows the implementation of the proposed temperature filter for the hand exoskeleton with a simple PI control.

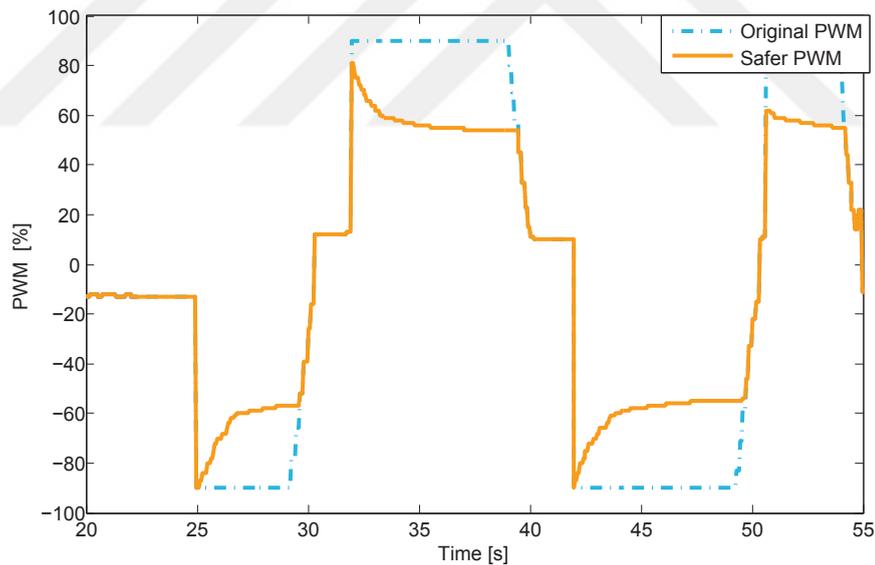


FIGURE 4.8: Real time implementation of the temperature filter being applied to the PWM signals generated during a PI control for the chosen actuator.

Implementing a temperature filter prevents the actuators to be exposed for continuous, excessive amounts of current and be burnt out during operation. However, they limit the operational speed significantly. Such a drawback might be insignificant for grasping tasks or repetitive rehabilitation tasks, since the actuator speed can be limited and modified intentionally, with no restriction. However, the speed limit cannot be tolerated

for active tasks, where the device transparency is crucial. Even though properties of the filter can be adjusted to improve transparency, it should be eliminated for most of the haptic tasks. Such an elimination does not threaten the safety of the electronics components of the system, because allowing each finger component freely over control reduces the possibility for a mechanical clutter significantly.

#### 4.3.1.2 Grasping forces

The proposed hand exoskeleton is designed to embrace the underactuation concept, so that the device can adjust its operation mechanically. This adjustability needs to be tested as the exoskeleton assists a user to grasp various objects with different sizes and shapes. The mechanism was controlled by the simple PI control that was detailed previously. Figure 4.9 shows that the user can grasp objects with different sizes and shapes with no preliminary knowledge of the object. Note that the first two tasks have been completed by a male user and the last two tasks have been completed by a female user. Adaptation of the device for two different hand sizes automatically can be observed.

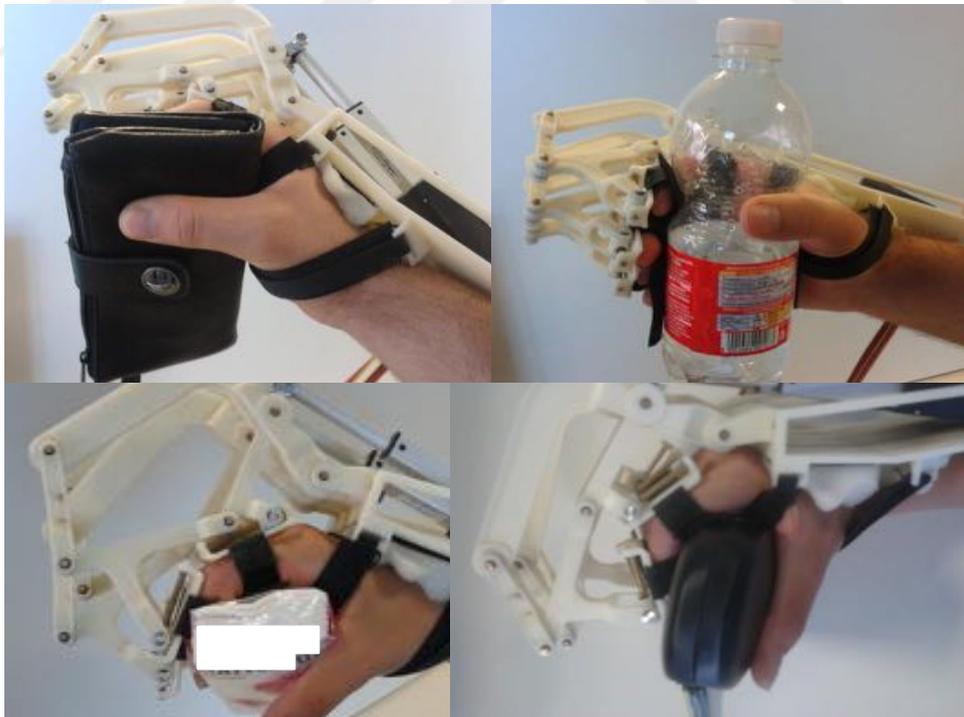
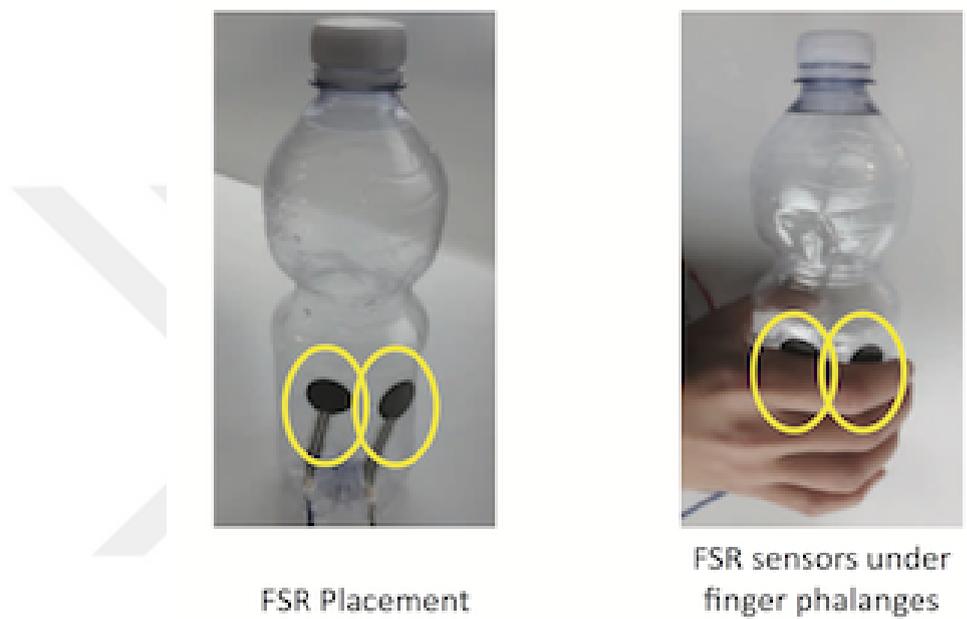
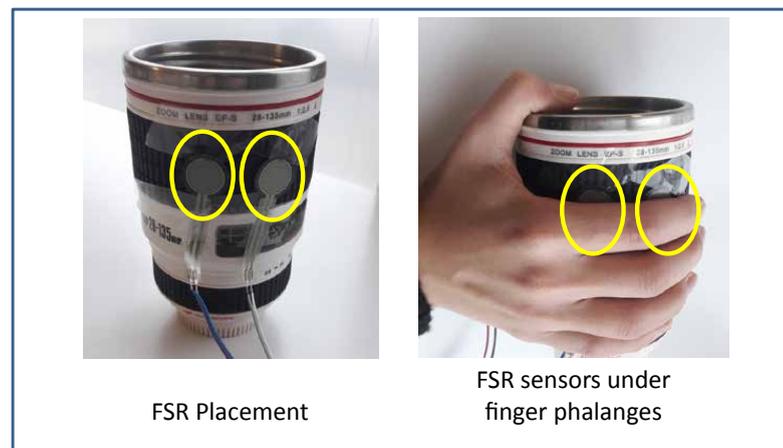


FIGURE 4.9: Real time implementation of grasping tasks of objects with different sizes and shapes while the proposed hand exoskeleton assists two subjects with different hand sizes.

The feasibility of the grasping tasks should be validated by measuring the interaction forces between user's finger phalanges and objects. Two different objects with different dimensions were selected to compare the interaction forces along finger phalanges. To receive efficient measurements during the interaction with the object, the FSR sensors (see Subsection 4.1.3) were inserted on the object as in Figure 4.10. Doing so, the FSR sensors can interact with the proximal and the intermediate phalanges of the index



(a) FSR sensors attached on a water bottle.



(b) FSR sensors attached on a mug.

FIGURE 4.10: Placement of the FSR sensors to measure the interaction forces between user's finger phalanges and cylindrical objects during grasping: (a) a water bottle and (b) a mug.

finger when the object is grasped. For this verification, we selected cylindrical objects to mount the FSR sensors easily on them.

Figure 4.11 shows the grasping forces over the actuator displacements while the user is grasping two cylindrical objects as in Figure 4.10. The hand exoskeleton is controlled with a simple PI control using a ramp reference as detailed in Subsection 4.3.1.

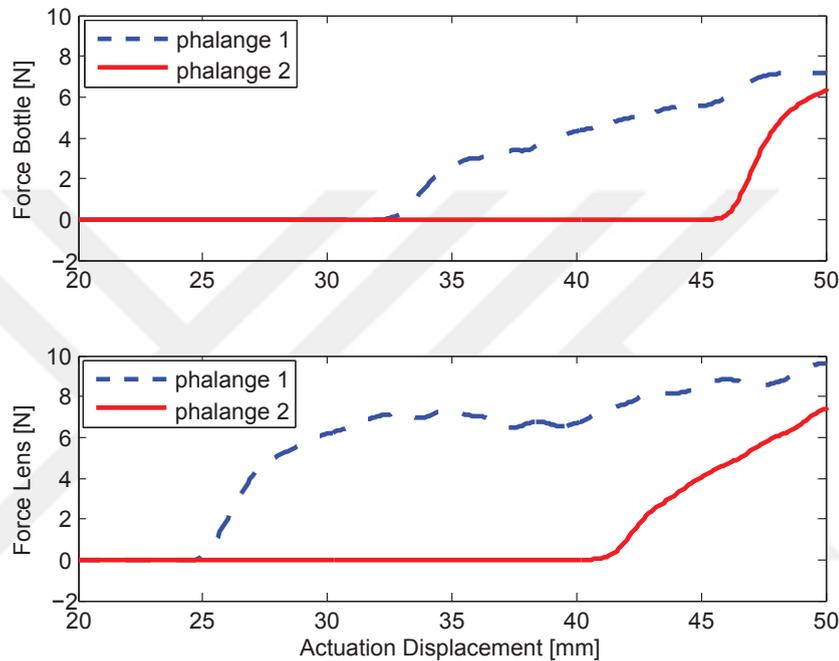


FIGURE 4.11: Interaction forces during grasping between user’s finger phalanges and FSR sensors attached on the object in Figure 4.10(a) and object in Figure 4.10(b).

In the first plot, the actuator activation starts to move the MCP joint of the user’s finger, until her proximal phalange reaches the object. Then, the interaction forces start to increase, as the PIP joint starts to be rotated. The PIP joint keeps rotating until the intermediate phalange reaches the object, and the interaction forces for phalange 2 start to increase as well. The actuator poses when the interactions occur and the force measurements are shown to be different for each object, emphasizing different behaviours for grasping. Such behavioral difference can be used to prove that the device was adjusted automatically by the kinematics. Yet, the stable grasping can be achieved for both cases. Even though the grasping forces are shown only for the index finger phalanges, other fingers have the same behavior, since they use the same kinematics.

### 4.3.1.3 Pose analysis

As detailed in Subsection 3.3.2, a unique solution for the finger pose can be achieved by placing an additional sensor along one of the passive joints of the mechanical system, and by calculating the forward kinematics of the system. A simple rotational potentiometer can be placed along the mechanical system as represented in Figure 4.12. The miniaturized sensor size does not cause the system to be bulky and heavy.

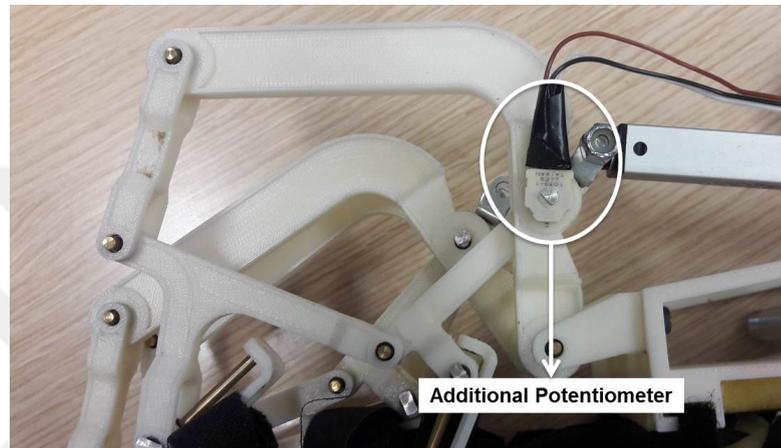


FIGURE 4.12: An image of the rotational potentiometer for an additional sensory measurement along one of the passive mechanical joints to predict user's pose during operation.

Section 3.3.2 and Section 3.3.3 solves the forward kinematics problem using numerical and analytical approaches. Even though they give same results in simulations, they have different characteristics in real life. Numerical solution highly depends on initial values for finger pose and mechanical joints, which might cause the finger pose prediction to converge for an unrealistic value.

Another difference between these two approaches is their calculational burden. The analytic solution is composed of single lines of equations, which can be calculated in a fast manner, while the numerical approach might require a couple of iterations to take place. How much time each iteration takes depends on the configuration and initial estimation. For the cases where finger pose is used only for recordings, such a calculational burden can be easily ignored. However, we need the kinematics to be calculated fast and accurately to use the finger pose as a feedback for the control algorithm.

A final issue is the physical capabilities of the utilized control board. In the proposed electronics design, a single DSP board is used to read all sensory measurements, run

kinematics calculations and run a controller to supply motor drivers for 5 finger components simultaneously. The simplest kinematics approach should be utilized within the control board to calculate the whole kinematics within a single time step, without interruption.

All these issues address a single solution, where analytical forward kinematics approach should be used to calculate finger pose estimations simultaneously. The finger pose estimation was performed while the index component was controlled by a simple PI controller with a ramp function reference (see Subsection 4.3.1). During the experiment, the user was asked to move her MCP and PIP joints in different manners:

- (a) keeping the MCP joint stable, and moving only the PIP joint,
- (b) keeping the PIP joint stable, and moving only the MCP joint, and
- (c) moving both the MCP and the PIP joints together freely.

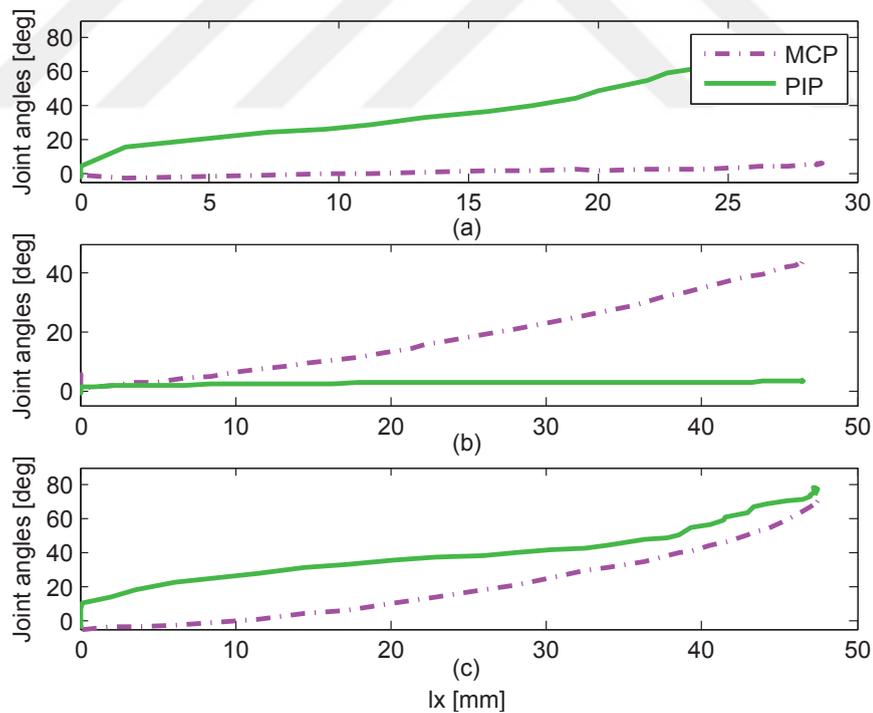


FIGURE 4.13: Results of pose estimation for different tasks over the actuator displacement  $l_x$ : (a) moving PIP joint with stable MCP joint, (b) moving MCP joint with stable PIP joint, (c) moving both MCP and PIP joints.

Physical objects were used to constrain all joint movements during the tasks (a) and (b). Pose estimation results during these three tasks were recorded by the host computer and presented over actuator displacements in Figure 4.13. This figure also shows that the same actuator motion can cause the finger move differently, so it presents how the underactuation works and why the additional sensor is necessary for the unique pose estimation.

The actual finger pose can be used for future serious game scenarios, where the exoskeleton and the application tasks are integrated in the virtual environment. A simple snapshot of such a simulation can be observed in Figure 4.14. For such an application, the MCP and PIP joints of user's hand can even be calculated by the host computer independently from the control algorithm, and these joint rotations are mapped to the hand mannequin. The DIP joint is rotated equally as the PIP joint.

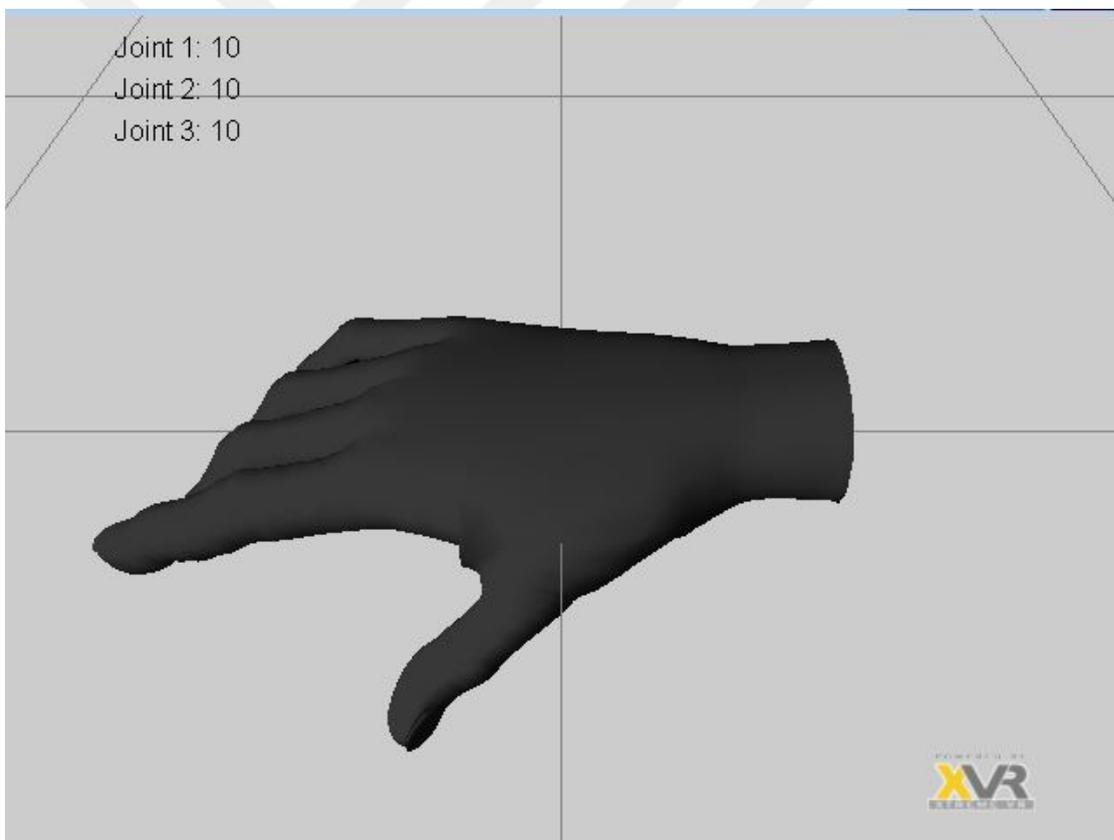


FIGURE 4.14: Hand mannequin model to visualize finger joints, which are calculated using pose estimation, simultaneously during operation.

#### 4.3.1.4 Calibration for finger sizes

Since the finger phalanges are accepted as a part of the mechanism, the lengths of finger phalanges are considered as a significant parameter for the operation performance as well as the active workspace of finger joints. Even though measuring the lengths of proximal phalanges manually might be sufficient, running a calibration block based on forward kinematics might be more accurate, practical and professional compared to manual methods.

Forward kinematics algorithm can also be calculated, under a certain pose of the user, where passive linear sliders around the middle finger phalange reaches its maximum stroke (Section 3.3.2). Such a calibration process is needed for each user or patient only once, and this is why it is sufficient to calculate it in the host computer. This actually prevents the necessity to change the built-in code to be used for the actual tasks for each user, increasing the preparation time. Even though the pose estimation was previously calculated through the analytic approach, the calibration can be performed through the numerical kinematics easily.

When the user reaches a certain pose, a possible singularity might occur when both the MCP and the PIP joints are fully extended and a third measurement estimation can be used without utilizing an additional sensor to run calibration process. Figure 4.15 summarizes the calibration process, where  $c_2$  is a constant with its maximum range,  $l_{LM}$  represents the length of proximal phalange,  $l_x$  and  $q_B$  are the sensory measurements.

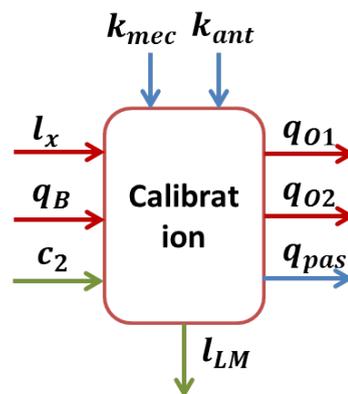


FIGURE 4.15: A representative scheme shows the pose estimation based calibration to estimate the length of the first finger phalange at a pre-defined pose of user's finger.

Such a calibration method would work only if user reaches a previously determined pose, to reach a constant value not only at the passive slider, but also at the actuator (42 mm).

The stable finger pose for the calibration can be observed in Figure 4.16. With this pose, all controllable joints are known, and the uncontrollable joint  $q_B$  is measured. In the end, the numerical forward kinematics can calculate the finger phalange length.

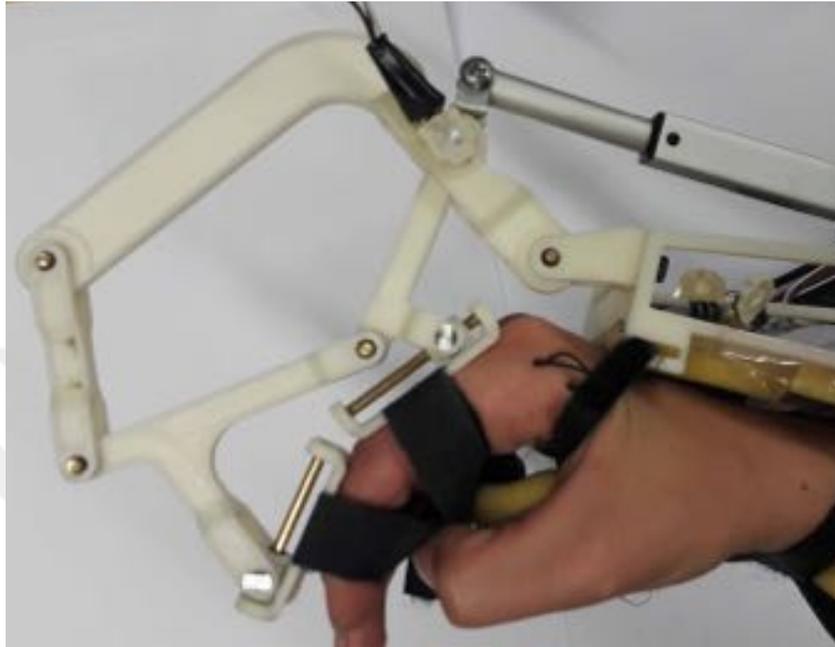


FIGURE 4.16: Pre-defined pose needed to compute the length of first finger phalange during calibration.

The calibration process was tested on a single user for the index finger only and her finger length was calculated as  $49.19\text{ mm}$ , while manual measurement was found as  $48.32\text{ mm}$  with an error rate of  $1.77\%$ . It is important to note that the initial guess and the finger pose should be close to each other. The initial guesses for passive joints should be determined carefully to increase the analysis performance.

### 4.3.2 EMG Control

For rehabilitation applications, the active participation of patients with disabilities might increase the efficacy of treatment process as much as their interest to the tasks, compared to the passive tasks. One common way of achieving such an active participation is to control the hand exoskeleton, which is worn by the impaired hand, using the muscular activity of the unimpaired hand, which can be measured through Electromyography (EMG) sensors.

In the literature, most of the studies use the filtered EMG signals to move multiple fingers worn by the exoskeleton together. Wege *et al.* [109] created this system, where the EMG signals are used to control fingers independently by choosing the specific muscles to be measured implicitly. In their study the control of little finger component is coupled with ring finger activity, since they share similar muscles to function.

Usually, the EMG recordings during rehabilitation tasks are collected by attaching individual EMG sensors on each muscle. Even though this attachment approach provides the therapist to reach required muscle activities in a precise manner, placing these sensors requires a longer preparation time. Furthermore, the exact position of each sensor is highly important to obtain meaningful measurements. Figure 4.17 shows a compact and wearable Myo bracelet [2], which can obtain EMG measurements, and address these issues easily.

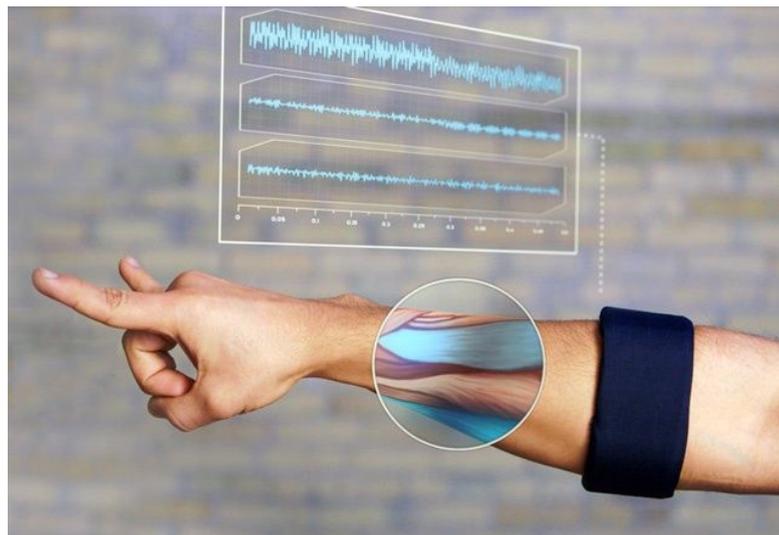


FIGURE 4.17: Myo bracelet provides the muscle activity of hand gestures in a compact manner [2].

Myo bracelet is a low-cost, compact device mostly developed for gesture tracking, measuring 12 channels of inertial measurement unit (IMU) and 8 channels of EMG activities. In this work, only EMG measurements are collected from Myo, while IMU recordings are neglected. 8 channels of EMG measurements are taken from 8 EMG sensors placed along the bracelet. Figure 4.18 shows the distribution and placement of these EMG sensors, which allow the muscle activity to be captured all around the arm. Since the quality of gesture estimation depends on the placement of these sensors around the arm, how the device is worn is highly important.



FIGURE 4.18: Distribution of 8 channel EMG sensors around the Myo bracelet.

The EMG measurements can easily be extracted from its gesture control software and be used for alternative applications. The device can send these EMG measurements to the host computer through Bluetooth. Its sensitivity about finger movements, low cost and easy wearability makes Myo sufficiently suitable for teleoperation tasks of the hand exoskeleton.

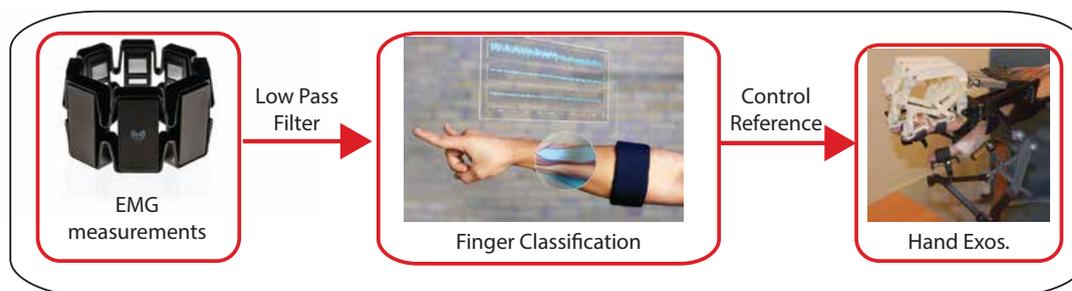


FIGURE 4.19: A representative scheme to use the EMG measurements as a reference to control fingers independently using a low pass filter and an observation based finger classification process.

Figure 4.19 shows the sequence of the required actions to control the hand exoskeleton based on the EMG measurements from the Myo bracelet. The first process filters the raw measurements. These signals are classified to detect individual finger activities through the intensity of muscle activity using an observation method. The finger activities are normalized to have values between 0 and 1, while these normalized values are used as a position reference for actuator displacements between 0 and 50.

The EMG signals extracted from the Myo bracelet are modified to reach more meaningful and useful shape by the classification phase. Since the raw EMG signals are so noisy, a low-pass filter with the frequency of 10 *Hz* is needed to achieve clear signals to be processed.

The filtered EMG signals are observed for each finger movement independently until various weights for the 8 EMG channels of Myo were found to obtain independent finger



(a) Index finger calibration

(b) Middle finger calibration



(c) Ring finger calibration

FIGURE 4.20: Classifying the most effective EMG channels measured by Myo bracelet corresponding to flexion of index, middle and ring fingers independently.

movements. We focus on the index, the middle and the ring finger activities, while the little finger is coupled with the ring finger. The ring finger can be moved the same as the ring finger, because they are controlled by the same muscles on the arm. Figure 4.20 shows the training process to observe the EMG activity based on independent finger movements.

The training process for the finger classification starts by relaxing all fingers to ensure Myo does not detect any activity. Then, each single finger is flexed separately as in Figure 4.20 while 8 EMG activities are observed to detect at least 2 channels being affected by the movement the most. When these activities are combined, the finger activities are obtained and normalized between 0 and 1. Finally, these normalized values are multiplied with the maximum actuator displacement to create the position reference for each actuator. Figure 4.21 shows the actuator references for each finger component using the EMG measurements, while moving the index, the middle and the ring fingers separately.

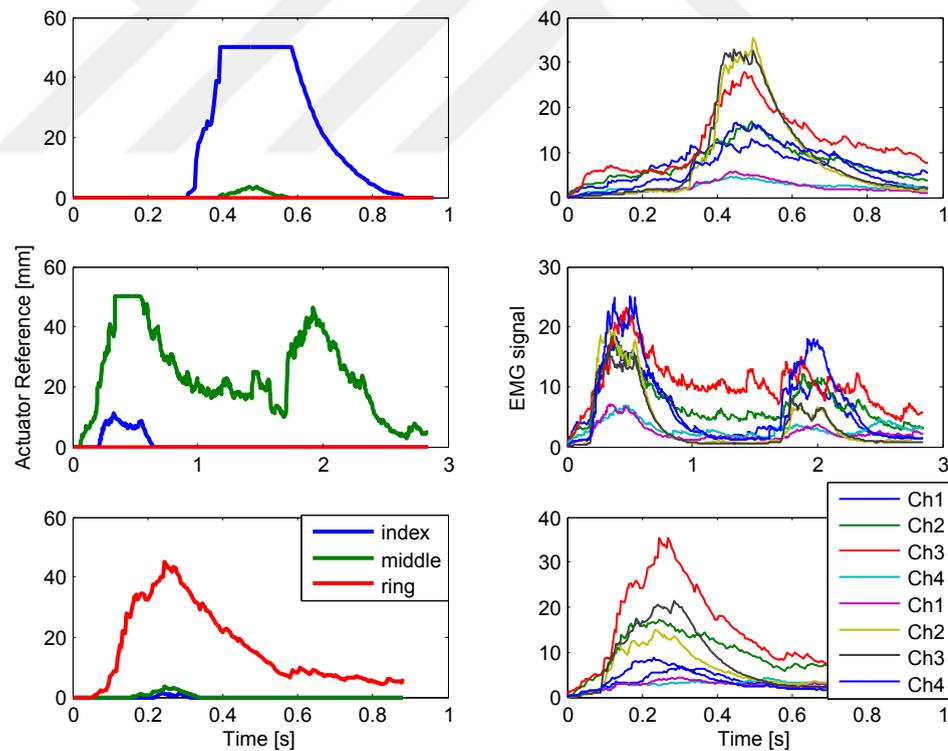


FIGURE 4.21: The actuator references depicted for independent finger movement through classified EMG activity.

The Myo bracelet can be used for teleoperation tasks with individual finger control. Even though we can reach independent finger control, we will not design a task with the

hand exoskeleton, where we expect users to open/close their fingers to their anatomic limits. We know that we, humans, have fingers, with certain coupling between them and in our daily lives, we tend to move them together in different strategies while performing different tasks. It is extremely useful to be able to detect independent finger movement and control the exoskeleton accordingly, but it will be used only to achieve task adjustability.



FIGURE 4.22: The experimental setup for EMG-controlled underactuated hand exoskeleton with MYO bracelet.

Figure 4.22 shows an example experiment with a Myo bracelet and the hand exoskeleton. The EMG activity occurs only when the fingers are flexed strongly, and disappears when they are relaxed. For such a teleoperation scenario, the left arm with the Myo bracelet is designated as the master, which detects the reference finger movements, while the right arm with the exoskeleton is designated as the slave, which mimics the finger movement of the left arm. The user is asked to

To obtain tasks that are similar to ADLs, user was asked to push her fingers that are attached to Myo bracelet against an object, a full water bottle in this case, to control her other fingers individually. Using a full water bottle allows EMG activity to be stronger, which would result with a more accurate classification and control performance.

The position reference obtained by classifier in Figure 4.21 are directly given to position control algorithm as detailed previously in Subsection 4.3.1. Since performance of the position control algorithm was discussed previously, controller performance is not presented again in order to avoid repetitions.

### 4.3.3 Force Control

Firgelli *L16* actuators were chosen to control the underactuated hand exoskeleton due to its compact and portable design, despite the fact that they are nonbackdriveability. Even though these actuators prevent the proposed hand exoskeleton to reach backdriveability when the actuators are off, it can be achieved actively, by measuring the forces coming from the user and move the actuators accordingly. Figure 4.23 shows the assembly of single 1 – *DoF* miniaturized load cell with strain gauges, which are used as force sensors, to the hand exoskeleton.

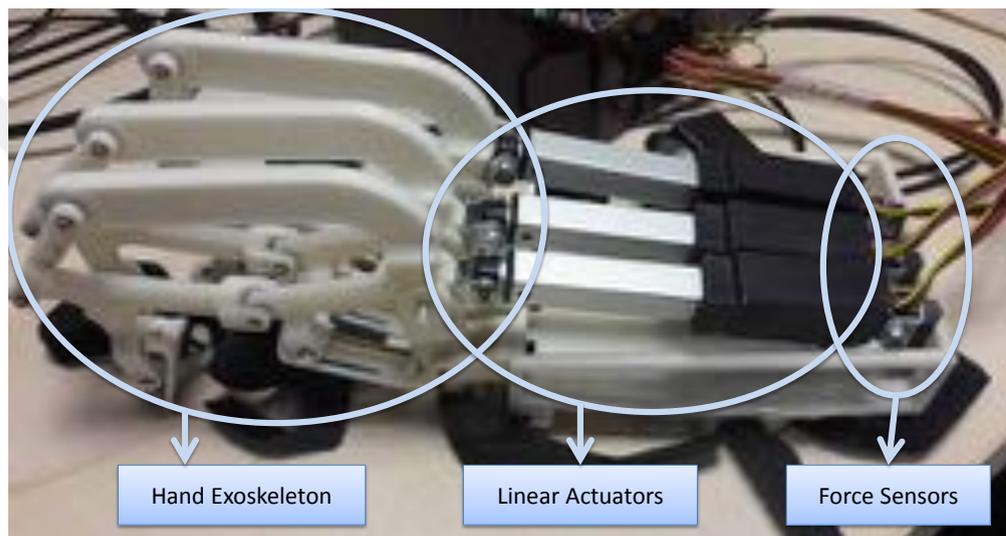


FIGURE 4.23: Utilization of 1 – *DoF* force sensors for each finger component independently to measure user’s intention to open/close his finger and move the actuator with high transparency to achieve backdriveability over control.

When the user, who is wearing the exoskeleton, wants to move his fingers to open/close them, the force sensor captures these forces, and a force control algorithm in Figure 4.24 calculates the required PWM supply to drive motors accordingly. As a result, the hand exoskeleton moves the actuators of the finger components, until the user reaches the pose he desires. Depending on the frequency of the control loop, this movement can be achieved almost simultaneously and transparently. The speed of the motion is also adjusted based on the measured forces.

For active backdriveability, the measured forces  $F_m$  created by the user should be the only the reference for the control scheme, so the  $F_d$  is set to 0. Equation (4.1) explains the mathematical expression of the force control block represented in Figure 4.24. Actuators are controlled using PWM signals  $F_{PWM}$  directly related to the measured forces  $F_m$ ,

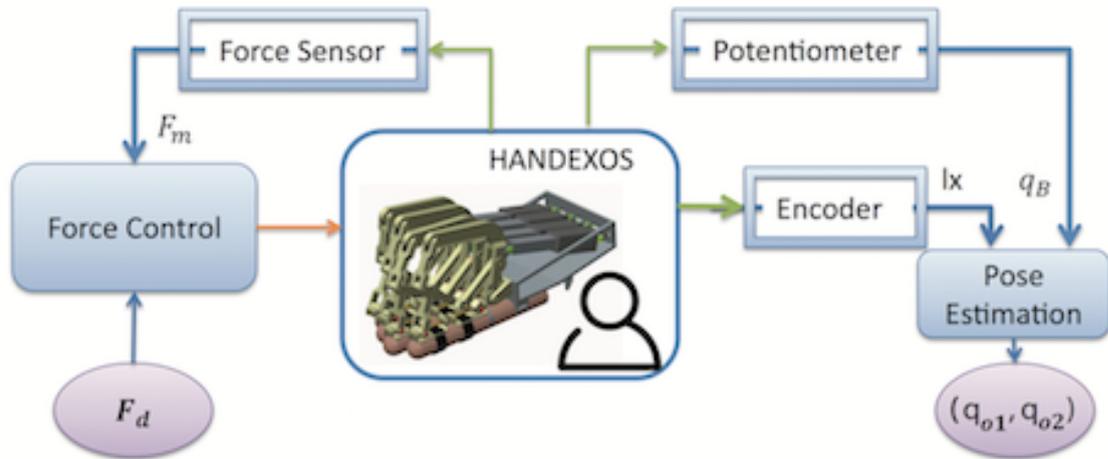


FIGURE 4.24: A representative scheme of a generic force control algorithm that can be used for active backdriveability when  $F_d = 0$  or haptic rendering when  $F_d$  is adjusted based on the virtual interactions.

where  $K_P$  and  $K_I$  are proportional and integral gains of the force controller, since desired force is kept 0.

$$F_{PWM} = K_P F_m + K_I \int F_m. \quad (4.1)$$

Very similar to the Figure 4.5, the performance of the force control will suffer from the delay between the controller command and the initialization of the actuator movement. Therefore, an additional PWM support is added to the controller output before sending PWM signals to motor drivers. Figure 4.25 shows the representative support to overcome backdrive forces during control.

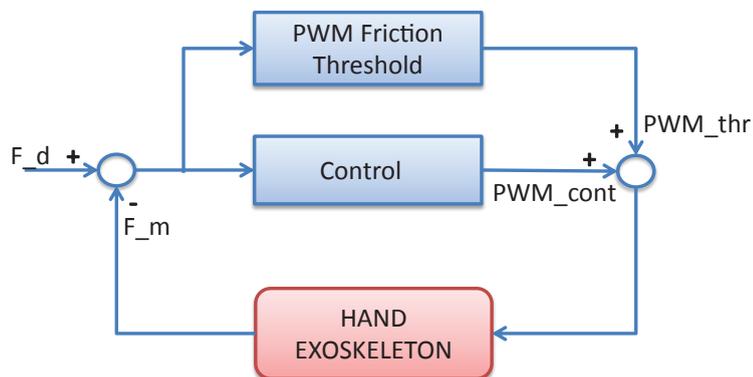


FIGURE 4.25: Adjusted force control algorithm with additional PWM supply in the same direction of control algorithm to overcome internal actuator friction.

The PWM signal strictly adjusts the actuator velocity through input current, so that the user can control the movement speed simply by increasing the applied forces. Figure 4.26 shows the active backdriveability task results. The first plot shows the applied forces by the user, the second plot shows the produced PWM output, and the third plot shows the corresponding actuator displacement. Since the PWM signal controls the actuator velocity, the actuator displacement increases when the PWM value is positive and decreases as the PWM value is negative. Backdrive forces shown in Figure 4.4 introduces a delay between the second and the third plots. Nevertheless, the relation between the applied forces and the actuator displacement shows that the control algorithm provides sufficient transparency to the device for haptic applications.



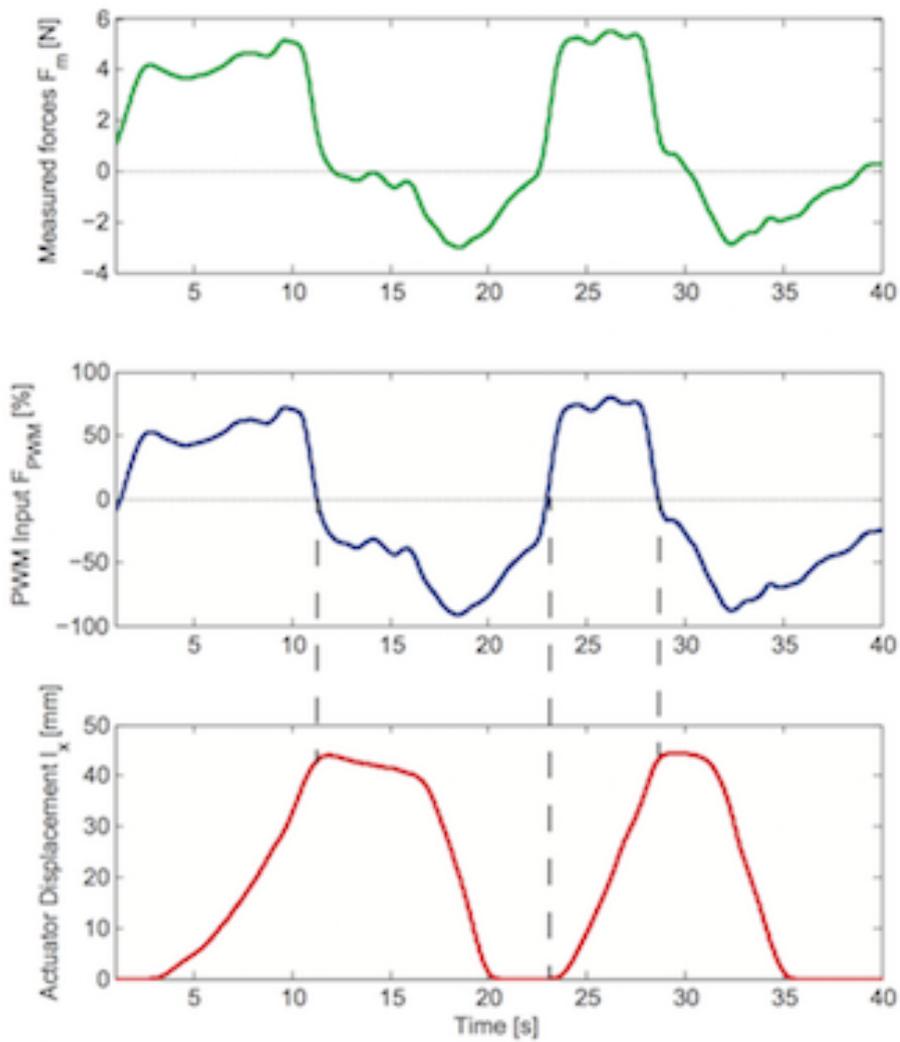


FIGURE 4.26: Real time implementation of an active backdriveability task using the proposed index finger component of the exoskeleton: (a) measured forces through additional force sensor, (b) PWM input for the given control algorithm in Eqn. (4.1) and (c) obtained displacement of the actuator to open/close the finger.

## Chapter 5

# Haptic Rendering

Haptic rendering focuses on the delivery of virtual object properties to a user through sense of touch using robotic devices. In particular, haptics make it possible to access places and tasks that are too dangerous and too expensive to explore, such as surgical training, video games, simulators, CAD systems for engineers, physical training exercises, etc. With improvements in haptics technology. Haptic tools for these applications have the potential to generate realistic scenarios economically and to create the perception of touch based on the properties of virtual objects, such as shape, size, weight and stiffness.

The perception of touch can be divided into two groups: tactile and kinesthetic. The tactile sensation consists of the perception of pressure, warmth, cold, pain, vibration, etc. The tactile devices attempt to mimic the characteristics of virtual object and local stiffness of the surface, targeting the mechanoreceptors on our skin to detect these properties, without moving the muscles of the user. On the other hand, the kinesthetic system refers to the feedback regarding muscles, tendons and joints for the perception of motion and exchanged forces.

In this chapter, a brief literature survey will be presented for the haptic devices and the approaches to utilize underactuated devices for haptic rendering. We will mostly focus on the virtual grasping tasks, since we are searching for ways to implement haptic scenarios for a hand exoskeleton designed in the previous chapters. The feasibility of force transmission and stiffness rendering tasks will be shown with a simple actuator level rendering task and by simplifying the exoskeleton mobility to a single  $DoF$ . Once the force transmission of the mechanism and Jacobian is proven, various rendering strategies

will be developed based on transmitted forces to fingers and desired pose to reach. Even though these strategy ideas are initially developed to overcome the lack of controllability during haptic tasks, they can still improve the overall performance, even for simple trajectory tracking tasks. These strategies were developed under the assumption of 2 *DoF* mobility guided by anatomical joints and stiff bone structure. Finally, a similar proxy based haptic rendering algorithm will be developed for a haptic simulation of soft finger tissue simply without simplifying the finger mobility to 2 *DoF*. Nevertheless, both strategies are generated sharing similar principles.



## 5.1 Haptic Devices

Haptic devices share some characteristics to provide sufficient force feedback through virtual interactions, such as low backdriveable inertia and friction, or minimal motion constraints regarding the device to increase the finger workspace. High control quality can be ensured only by improving the range, the resolution and the bandwidth of position sensing, and force reflection requirements.

Haptic devices in the literature can be classified based on the number of DoF for motion, which signifies the number of dimensions in an actuated or non-actuated manner to modify the possible movements/forces exchanged between the device and the operator [131]. The impedance-type devices in the literature can be listed as 1 – DoF, 2 – DoF, 3 – DoF, 6 – DoF and  $> 6$  – DoF devices.

- **1 DoF devices:** Most of the 1 DoF devices provide tactile feedback perpendicular to user's fingertip. A set of external sensors are equipped to track the position and the orientation of the fingertip. Once a contact takes place in the virtual environment between the user's avatar and the object, the device applies force and/or vibration to imply the texture, the stiffness or other properties of touch. Depending on the application, these devices can be designed in form of a haptic knob [132, 133], a haptic scissor [134], or a force reflecting gripper [135].
- **2 DoF devices:** Tactile devices can be designed to impose 2 DoF feedback in perpendicular and linear tangential directions to the fingertip, instead of 1 – DoF during virtual grasping tasks [136–139].  
Alternatively, kinesthetic devices can constrain user's movement to a single plane such that a virtual contact can occur only in 2 – D environment. Doing so, the perception of contact can be delivered to the user in a sufficient manner by utilizing devices in the form of a simple pantograph [140] or a force feedback mouse (Logitech, Wingman).
- **3 DoF devices:** Tactile devices can also impose 3 DoF feedback in perpendicular and planar tangential directions to the fingertip [141–143]. Doing so, all interaction forces in the virtual contact can be given to the user as in real life, even though the interaction torques are ignored.

From the kinesthetic point of view, haptic devices can support contact appearance in 3 –  $D$  environment. OMEGA [144] provides 3  $DoF$  feedback from its end-effector, where users move its end-effector and grasp virtual objects. Phantom [145] on the other hand provides an active 3  $DoF$  feedback to user’s finger, while a passive thimble worn at the fingertip provides an additional 3  $DoF$  mobility. In particular, passive  $DoF$  allow the fingertip orientation to be known, such that the pure force transmission to the fingertip can be used for 6  $DoF$  virtual tasks. Nevertheless, contact torques between the user’s avatar and the virtual object are ignored.

- **6  $DoF$  devices:** Tactile devices can also be extended to assist 3  $DoF$  contact torques together with contact forces.

The same idea goes for the kinesthetic feedback in terms of upgrading 3  $DoF$  devices to support also the interaction torques. In particular, 6  $DoF$  PHANTOM [146], 6  $DoF$  DELTA [144], and Freedom 6 [147] can measure user’s movements and provide the feedback based on the virtual interactions in 6  $DoF$ .

- **> 6  $DOF$  devices:** A 6  $DoF$  manipulation device can be integrated with a single  $DoF$  grasping mobility to complete the task [148].

Furthermore, a haptic device can be designed in the joint space rather than the Cartesian space to assist hand or arm movements during virtual operations in the shape of exoskeleton. The device can provide force feedback to the anatomical joints of the user, achieving more than 6  $DoF$  overall.

Kinesthetic feedback during haptic applications can be given either by controlling the fingertips or the finger joints implicitly. Hand exoskeletons provide force feedback to the finger phalanges to stimulate full grasping tasks [50, 87, 89, 98, 149–151]. Actuating each finger joint independently can be achieved either by aligning actuated mechanical joints with finger joints or through RCM mechanisms. Despite of mechanism choice, utilizing independent actuators for each finger joint increases the overall cost and the complexity. One way to overcome this complexity is to control only user’s fingertip by achieving multiple  $DoF$  for each finger [38, 44, 97, 152–156]. Fingertip devices are attached only to user’s fingertip such that the fingertip position and orientation can be measured and resistive kinesthetic feedback is given to the user when any virtual interaction occurs. These devices have simple design and are easy to be worn by user,

however the kinesthetic feedback is given only to the fingertip, such that finger phalanges are left free.

During haptic rendering tasks, the desired forces are calculated based on the stiffness value of virtual object, and the distance between the virtual contact and the user's configuration. These forces are mostly delivered to the user by utilizing fully controlled haptic devices, where all mechanical and virtual DoF are controllable individually. Using these devices, all mobility that users perform during virtual tasks can be both tracked and controlled independently. Therefore, the standard haptic rendering algorithm makes the assumption for actuated DoF of device to match the subset of proxy DoF. Unfortunately, this assumption does not hold for many interesting haptic systems. On one hand, computational models and resources grow steadily, and we can currently afford interactive simulation of rich and complex haptic probes [157–159]. On the other hand, novel designs of exoskeletons exploit underactuation to maximize simplicity, wearability, etc. [160]. In such haptic systems, actuated DoF may not map to a subset of the DoF of the haptic probe. As we show in the paper, naive application of proxy-based rendering to underactuated devices produces undesired results. A formalization and generalization of the approach can be found in [161].

The proposed underactuated hand exoskeleton was designed mechanically not only for rehabilitation purposes, but also to respond the requirements of haptic applications. The mechanism was designed in a compact, wearable and portable manner. Then, the properties of pose estimation and force control algorithm allow the device to track user's configuration and provide kinesthetic feedback based on user's activity in the virtual environment. However the underactuation property, which provides many advantages to the device, requires an extension of the conventional haptic rendering algorithm.

### 5.1.1 Underactuated Haptic Devices

Imposing additional mobility through underactuation might create uncertainties to a haptic system. Nevertheless, their advantages based on portability, affordability and simplicity increase for underactuated devices in the haptic applications as well. The most common way to embrace underactuation in haptic applications is to create haptic devices with 6 *DoF* sensing input (translation and rotation) and 3 *DoF* actuation output (only forces, but not torques). Alternatively, underactuation can be made with 2 *DoF* or 1 *DoF* actuation output and 3 – *DoF* sensory input such that the tangential forces

are limited to 1 direction only. The literature involves multiple studies on developing various strategies to handle the underactuation property.

Verner and Okamura [162] performed experimental analysis of the performance gain provided by torque feedback over force-only feedback. Lee [69] designed a modified penalty-based method that ensures admissible rendered stiffness while correctly balancing directional forces. And Kadlecěk *et al.* [163] used sensory substitution and pseudo-haptic feedback to simulate torque feedback. Massimino *et al.* [164] studied the capabilities of sensory substitution for force feedback through the tactile and auditory senses, with and without a time delay. In particular, the efficacy of the sensory substitution during the haptic performance has been found to be dependent on the quality of the visual feedback, the characteristics of the force feedback and the complexity of the sensory substitution. Lecuyer *et al.* [139] uses an underactuated haptic device, where the device provides 3 *DoF* sensory information and 1 *DoF* actuation. In order to cope with such an underactuation, a new technique is proposed to improve the contact perception during point-based haptic exploration. In particular, the virtual scene is rotated when a contact occurs in order to align the contact normal to the haptic device. However, such technique can be useful when the actuated and the sensed DoF are independent from each other in the end-effector.

Barbagli and Salisbury [135] analyzed underactuated haptic devices in terms of controllability and observability of a virtual environment, and consequently provided guidelines and considerations for the design of a device. However, they did not make an effort at trying to optimize haptic rendering settings for a given haptic device. Verner and Okamura [165] analyzed passivity in asymmetric devices (i.e., those with a different number of controlled and observed DoF).

The works of Luecke [166] and Meli and Prattichizzo [167] are probably the closest to our situation with the underactuated hand exoskeleton presented in this thesis. Luecke considered a haptic device whose end-effector maps exactly to a virtual object, but it is underactuated. Haptic rendering is computed by first defining forces for the virtual object, and then finding optimal control parameters that maximize the similarity between those forces and the ones actually displayed by the device. The solution does not consider cases where the virtual object has a higher dimensionality than the end-effector or cases where the haptic device includes non-actuated DoF. Meli and Prattichizzo studied methods to render contact forces through an underactuated device, while maximizing task performance. For each task, they defined an optimality criterion in the selection of

underactuated feedback. In contrast to theirs, our work stands on proxy-based rendering, but it can accommodate task-dependent metrics in the computation of the proxy.

Even though there is sufficient effort to improve the use of underactuated devices for haptic rendering, the focused underactuated devices were mostly chosen among Cartesian devices. Independency between DoF of underactuated Cartesian devices might not exist for underactuated exoskeletons, such that some of the proposed techniques cannot be implemented properly. Furthermore, it was observed that most of the optimization methods were shown to be useful under certain assumptions on the designed tasks. Utilizing proxy-based haptic rendering that focuses on optimization algorithms might be useful to generalize the use of underactuation for virtual tasks.



## 5.2 Virtual Interaction

Haptic applications require users to discover a defined virtual environment and make contacts with the virtual objects within this environment. During a virtual grasping task, a haptic device should follow user's intentions with high transparency while providing resistance to user's fingers when encountered with the virtual object. Such an interaction can be designed to be performed in many cases, such as exploring with fingertips, grasping, pushing, etc. The hand exoskeleton designed in this study was designed to assist grasping tasks, therefore haptic exercises in this chapter will focus on virtual grasping tasks. Nevertheless, the issues caused by underactuation concept or rendering solutions can be used for any interaction task.

In terms of perceiving force feedback, any virtual task can be divided into two groups: before contact and during contact. Before the contact, the user can move freely without perceiving any force feedback. Once the contact is established between user and virtual object, user starts to perceive resistive forces from the point of connection that signifies the behavior of the task itself. In fact, resistive forces are formed based on the stiffness of the object and forces applied by user to object throughout the task.

### 5.2.1 Actuator Level Stiffness Rendering

The behavior of force feedback during grasping can easily be imitated by a robotic device. For an underactuated device, the easiest way to implement such a control algorithm is to define a limit for the displacement of the actuator and form the desired force  $F_d$  to be used in Figure 4.24 accordingly. Since we are dealing with an underactuated system, the desired force  $F_d$  might consist of active and passive components, and to be consistent with the rest of the Chapter, we will call the active desired force  $F_a$ . Even though defining the stiffness rendering task based on the actuator displacement neglects the pose and orientation of finger joints, it simplifies the control algorithm by ruling out the underactuation.

As mentioned previously, the stiffness rendering task should give the perception of touch only, while allowing the user to explore the environment freely. In order to eliminate forces that might assist the user to reach desired values before getting into the contact, the positive values of the desired force  $F_a$  should be kept 0, while only the negative values should be given to the user as a force feedback related to their touch. In order to achieve such a saturation, the desired displacement  $l_{x_d}$  is designed to be equal to actual

displacement before reaching the virtual limit and to the virtual limit after exceeding it, as summarized in Algorithm 2.

---

**Algorithm 2** Setting desired actuator displacement based on virtual limits.

---

- 1: **input:**  $l_x$ , the actuator displacement
  - 2: **input:**  $l_{x_{lim}}$ , the virtual displacement limit
  - 3: **output:**  $l_{x_d}$ , the desired actuator displacement
  - 4: **if**  $l_x \leq l_{x_{lim}}$  **then**
  - 5:  $l_{x_d} = l_x$
  - 6: **else**
  - 7:  $l_x \leq l_{x_{lim}}$
- 

Equation (5.1) presents the mathematical expression of the desired force  $F_d$  based on the instantaneous actuator displacement  $l_x$ , the desired actuator displacement  $l_{x_d}$ , which can be set as in using virtual limit for actuator displacement  $l_{x_{lim}}$ , and the stiffness value of the contact felt by the actuator  $K_{ac}$ .

$$F_a = K_{ac}(l_{x_d} - l_x). \quad (5.1)$$

Once the desired force  $F_a$  is calculated, the control algorithm presented in Figure 4.24 calculates the corresponding PWM signal using Equation (5.2) and runs the actuator. In Equation (5.2),  $K_w$  signifies the weight acting on the desired actuator force  $F_a$ ,  $F_e$  defines the difference between the desired and measured force values ( $F_e = F_a - F_m$ ), and  $K_P$  and  $K_I$  represent the control coefficients of a simple force PI control.

$$F_{PWM} = F_a K_w + K_P F_e + K_I \int F_e. \quad (5.2)$$

How additional FSR sensors are inserted between the hand exoskeleton and the user. In theory, increasing the stiffness value  $K_{ac}$  in Equation (5.1) leads to a higher desired force value, which keeps the error between instantaneous displacement and virtual limit to minimum. Conversely, lower stiffness value allows the user to keep up his movement after reaching the contact limit as the force feedback increases, just like squeezing a soft object. Figure 5.1 shows the effect of stiffness values in terms of allowed displacement after the contact limit. Keep in mind that a kinesthetic device, which can control the muscular activity of the user, actually can prevent the user to move after a certain force level is passed unlike tactile devices. In particular, the lower stiffness value allows the user to exceed the given limit more than the high stiffness values, while the user can

apply more force to the device due to such ability to move. On the other hand, when the contact is stiffer, the user perceives the existence of the stiff object applying less force to the device.

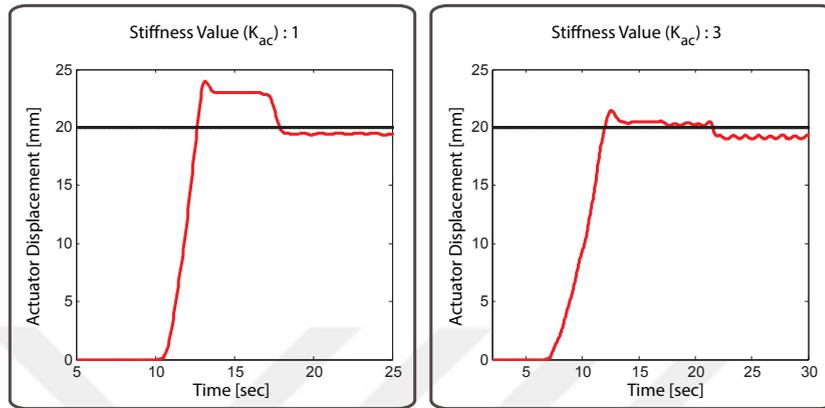


FIGURE 5.1: Actuator level stiffness rendering while the displacement after the virtual contact are different due to the various stiffness values.

Performing stiffness rendering algorithm at the actuator level of an underactuated hand exoskeleton is the simplest approach to provide resistive forces to finger opening/closing based on virtual interactions. However, this approach ignores finger joint activities and cannot estimate the actual perception of user, since the resistive actuator forces are delivered to multiple finger joints of the device. Even though for some simple rehabilitation tasks it could be useful to complete the realism of the tasks, defining the contact limit or the stiffness value for the actuator during the haptic applications would be impractical.

### 5.2.2 Joint Level Stiffness Rendering

Unlike actuator level stiffness rendering, joint level rendering algorithm calculates the interaction forces based on the activity of each joint independently. For a hand exoskeleton designed to perform virtual grasping tasks in haptic applications, interaction forces should be calculated at finger phalanges, where the interaction occurs. Figure 5.2 shows the interaction forces at phalanges during a virtual grasping task while the underactuated hand exoskeleton assists user's fingers. Whenever there is a contact between the virtual object and the user, the actuator should render resistive forces acting on virtual phalanges to allow the perception of touch. Fully controlled exoskeletons impose a kinesthetic feedback by computing the desired torques corresponding to the finger

joints, while the proposed underactuated exoskeleton imposes the feedback by computing the desired actuator force to help the user perceive the interaction at both finger phalanges. It is important to emphasize that actuator forces are computed in joint level method based on interaction forces at phalanges, while previous method did not focus on phalange forces at all.

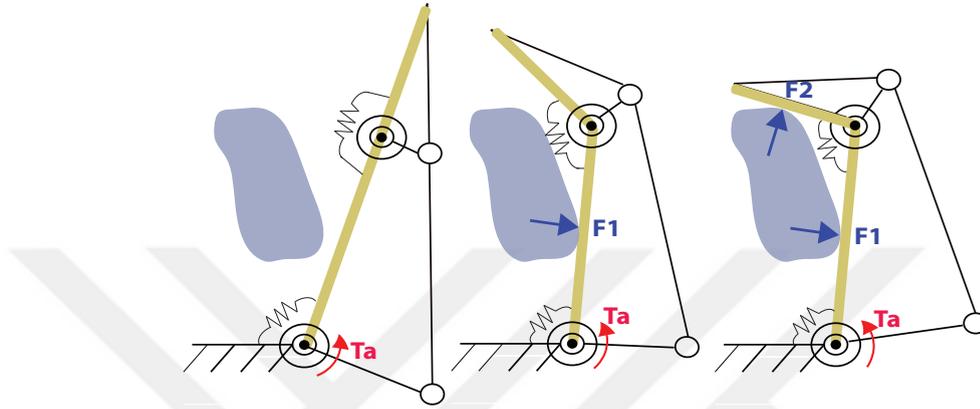


FIGURE 5.2: Underactuation concept through a grasping simulation through a 2 DoF gripper with a single rotational actuator.

The exoskeleton allows the user to move freely with high transparency until he reaches for the virtual object. When encountered with a virtual object, the interaction forces between the object and the user's phalanges are calculated using stiffness rendering algorithm stated as Equation (5.3) where  $\mathbf{q}$  and  $\mathbf{q}_d$  represent the instantaneous finger pose and desired pose. The desired pose  $\mathbf{q}_d$  elements are calculated as in Algorithm 3 based on joint limits for each finger joint.

---

**Algorithm 3** Setting desired joint pose based on virtual limits.

---

- 1: **input:**  $q$ , the joint pose
  - 2: **input:**  $q_{lim}$ , the virtual pose limit
  - 3: **output:**  $q_d$ , the desired joint pose
  - 4: **if**  $q \leq q_{lim}$  **then**
  - 5:      $q_d = q$
  - 6: **else**
  - 7:      $q_d = q_{lim}$
- 

For a real virtual grasping task, the finger pose limit is determined as the finger pose that the interaction occurs. However, we would like to test the capabilities of the hand exoskeleton before complicating the overall haptic system by introducing a virtual environment. For this reason, we will assume a virtual interaction to occur at a previously defined pose, let's say when both MCP and PIP joints are rotated around  $30^\circ$  each. So.

the finger pose limit will be assumed to be  $\mathbf{q}_{lim} = (30, 30)^T$ , while the execution of the desired finger pose accordingly can be shown in Figure 5.3.

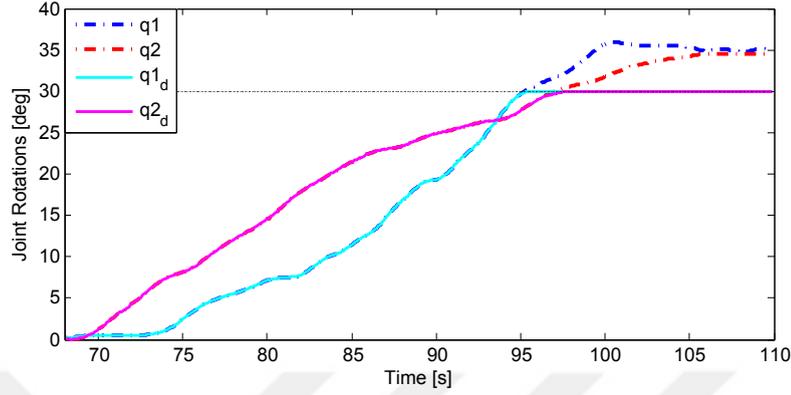


FIGURE 5.3: Computing desired pose  $\mathbf{q}_d$  to be used for stiffness rendering algorithm for given actual pose  $\mathbf{q}$  using Algorithm 3.

Figure 5.3 shows how the user move ( $q1$  and  $q2$ ) and how the desired finger pose ( $q1_d$  and  $q2_d$ ) change when the user reaches the finger pose limit accordingly. Keep in mind that the desired forces are calculated using the difference between the actual and the desired poses, so the desired forces are zero before the limit. Doing so, any assistance provided by the device until the user reaches the pose limit is prevented, and the device only renders resistive forces once the limit is forces.  $\mathbf{K}_{cont}$  is the diagonal, controllable stiffness matrix, which defines the sensed stiffness regarding the virtual object's properties.

$$\boldsymbol{\tau} = \mathbf{K}_{cont} (\mathbf{q}_d - \mathbf{q}) \quad (5.3)$$

For a fully controlled scenario, the actuator forces can be calculated using Jacobian transpose as  $\mathbf{F} = \mathbf{J}^T \boldsymbol{\tau}$ . Having all components in  $\mathbf{F}$  simply by designing a fully controllable device, the possibility of joint torques  $\boldsymbol{\tau}$  to be transmitted to finger phalanges can be achieved by implicit joint control. On the other hand, underactuated devices have non-actuated (passive) joints as well as actuated (active) joints to form a square, invertible Jacobian, which can be divided into active and passive components ( $\mathbf{J} = (\mathbf{J}_a, \mathbf{J}_p)$ ). Similar to fully controlled devices, the divided Jacobian can be used to relate actuator forces and desired joint torques using Equation (5.4)

$$\begin{bmatrix} \mathbf{F}_a \\ \mathbf{F}_p \end{bmatrix} = \begin{bmatrix} \mathbf{J}_a^T \\ \mathbf{J}_p^T \end{bmatrix} \boldsymbol{\tau} \quad (5.4)$$

where  $\boldsymbol{\tau}$  represents the array of end-effector output torques calculated in Equation (5.3),  $\mathbf{F}_a$  and  $\mathbf{F}_p$  define the arrays for actuated (active) and non-actuated (passive) forces. Passive joints are included for Jacobian formulation since they contribute for motion analysis, even though they do not provide any energy to the system. Such passivity to achieve an underactuation design can easily be transformed into a mathematical constraint in Equation (5.4) as  $\mathbf{F}_p = \mathbf{0}$ .

As mentioned previously, the underactuated concept performs force transmission based on contact forces simultaneously. In fact, the relationship between joint forces acting by the device to phalanges can be defined simply by detailing the underactuated constraint as:

$$\mathbf{0} = \mathbf{J}_p^T \boldsymbol{\tau}. \quad (5.5)$$

Since this constraint is implicitly defined by mechanical properties, it cannot be overruled. However, a set of desired forces  $\boldsymbol{\tau}$  can result the desired passive forces  $\mathbf{F}_p$  to have a non-zero value in order to be transmitted to finger joints, even though mechanically it cannot be performed. The conflict between desired and actual values of passive force  $\mathbf{F}_p$  complicates the stiffness rendering for underactuated devices and need to be solved.

### 5.2.3 Stiffness Rendering for Verification

Before getting into the details of underactuation, a simple set of experiments can help us to visualize whether the force transmission through the proposed hand exoskeleton is efficient. In order to do so, a user was asked to perform tasks that require only 1 *DoF* rotation. In fact, the user was asked to rotate only a single joint at a time, while other joint was kept constant, as can be visualized in Figure 5.4. It is safe to assume that ensuring the stable joint torque to be constant 0 throughout the task simplifies the underactuated hand exoskeleton to a simple 1 *DoF* device. The desired torque around active joint should be calculated simply by Equation (5.3) while Equation (5.4) shows how to obtain the corresponding desired actuator force  $F_a$ .

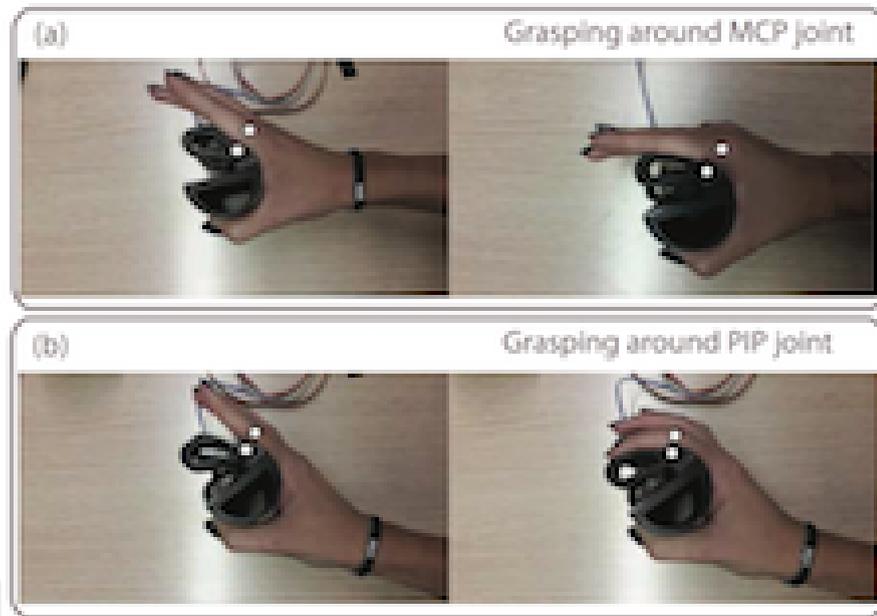


FIGURE 5.4: Finger motion during the experiment: (a) the user aligns his MCP joint with the mechanical joint of the object to rotate his MCP joint, while his PIP joint remains with minimum movement. (b) the user aligns his PIP joint with the mechanical joint of the object to rotate his PIP joint, while his MCP joint remains with minimum movement.

Despite of the inconsistency caused by passivity constraint of passive joint, creating a 1 *DoF* movement would provide a more promising approach to begin studying the mechanical behavior with. The feasibility tests on stiffness rendering algorithm and force transmission of the exoskeleton can be shown simply by comparing desired actuator force  $F_a$  value based on the active joint torque to external measurements from a customized object and a FSR sensor. In order to minimize the dependency on surface area of the contact with FSR and to obtain comparable, repeatable results during different tasks, a custom object was manufactured as in Figure 5.5. The FSR sensor was chosen to provide quick, simple measurements. A spring was attached in front of the FSR to allow user to squeeze the object for force measurements, whose stiffness value is used for the stiffness rendering task  $\mathbf{K}_{cont}$ . Finally, a silicon interface was introduced between the spring and the FSR in order to increase repeatability and efficacy of this pressure sensing tool for the setup.

The experiment starts with a calibration process, where the user was asked to align her active joint to mechanical joint and simply rotate her joint until her active phalange reaches the object, as depicted by Figure 5.4. Meanwhile, the finger pose of the user was monitored in order to determine the desired joint limits to form  $\mathbf{q}_d$  as in Algorithm 3 for

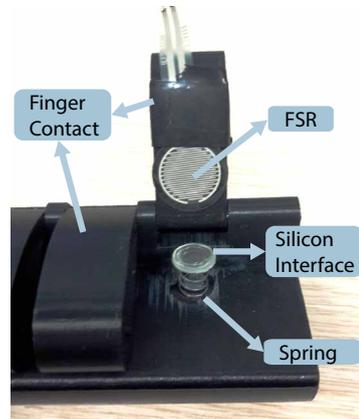


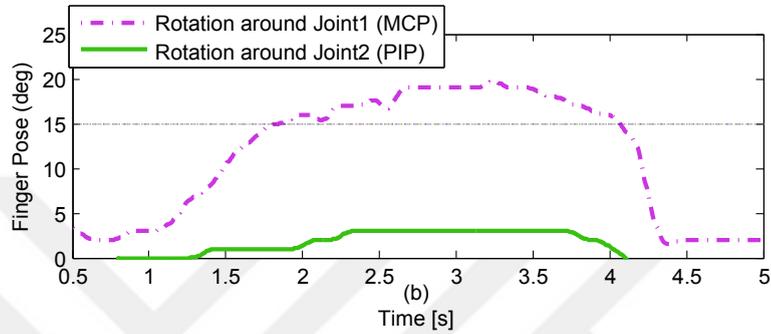
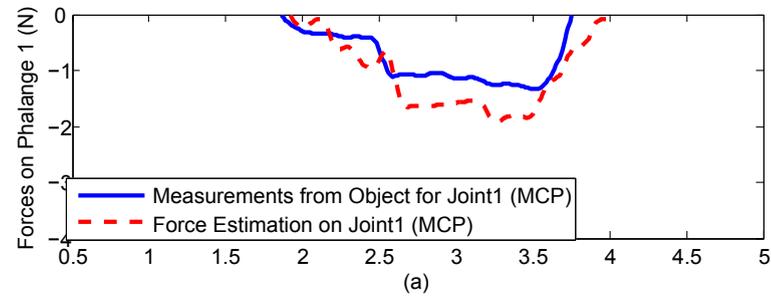
FIGURE 5.5: FSR Interface to measure forces on the finger phalanges: the finger contact with a mechanical joint allows the user to grasp and squeeze the interface. The FSR sensor measures the pressure applied by the user's finger. The silicon interface aims to imperfect the diversities of how the forces are applied to the FSR sensor. The spring allows the user to squeeze the object, imitating a soft object during haptic rendering task.

stiffness rendering in Equation (5.2). The limits for MCP and PIP joints ( $q_{lim_1}$ ,  $q_{lim_2}$ ) were detected as  $15^\circ$  and  $35^\circ$  by the calibration.

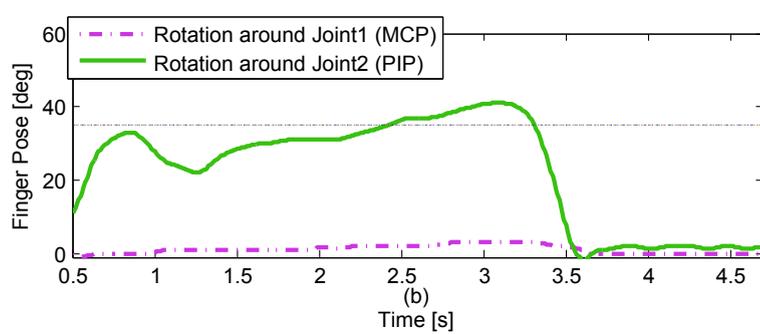
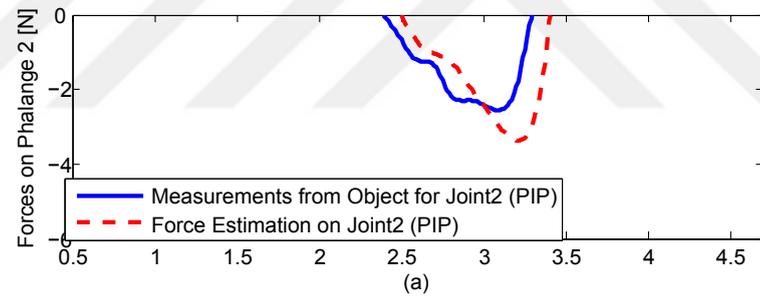
Aligning finger joints with mechanical joints not only increases repeatability of the task, but also allows desired torque values to be converted to forces acting on the contact point, which is the center of the spring shown in Figure 5.5. In fact, such conversion can be made easily using the distance between center of FSR to joint of the custom object ( $m$ ) as in  $F_i = \tau_i * m$ .

Meanwhile, stability of the hand exoskeleton can be analyzed simply through displayed impedance,  $\Delta F_a / \Delta q$ . Since the resistive forces cause the desired impedance to be negative, the negative definition of the impedance values in Figure 5.7 signify the stability. Having a single active finger joint at a time allows the sensed impedance factor to be considered only for the movable finger joint.

Figure 5.6 shows the comparison between the FSR measurements and the desired contact forces at the center of spring for rotating MCP and PIP joints individually. The root mean square error between the actual and desired contact forces were calculated to be 0.623 and 1.74 for MCP and PIP joints respectively, which can be acceptable by some justifications. First of all, the passive joint is assumed to have 0 torques around it, however the given setup in Figure 5.5 does not collect any measurement from the passive finger joint. Furthermore, the FSR sensor was not designed to be suitable for accurate sensing. Nevertheless, observing a similar behavior by the measured and the



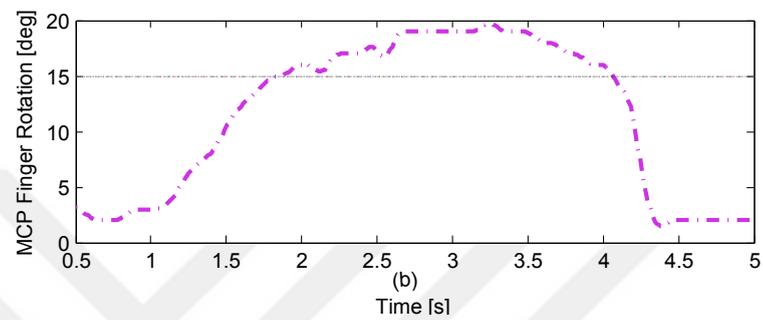
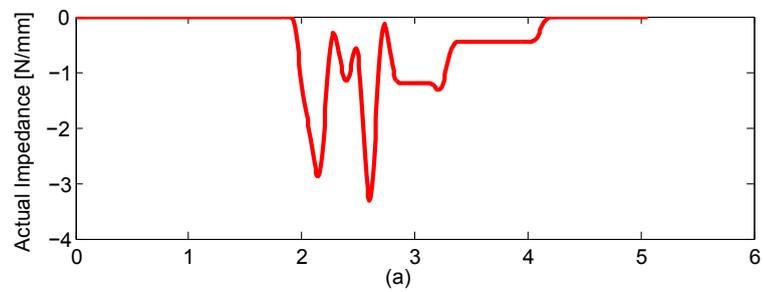
(a) Stiffness rendering task around MCP joint



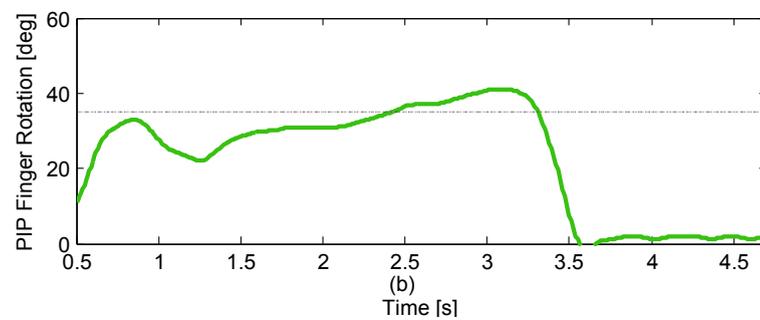
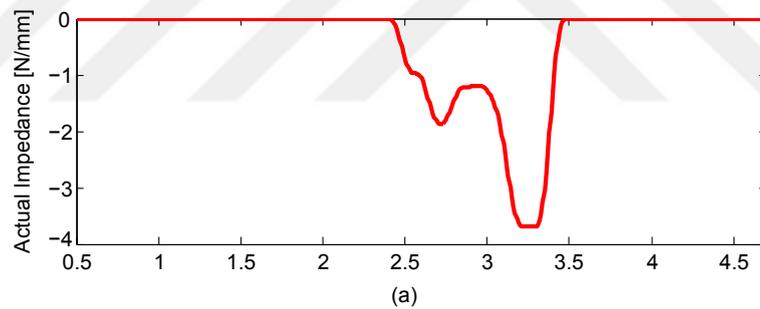
(b) Stiffness rendering task around PIP joint

FIGURE 5.6: Experimental results for force applied by the exoskeleton vs. force measured by FSR and the pose estimation for: (a) stiffness rendering task around the MCP joint; (b) stiffness rendering task around the PIP joint.

desired contact forces show sufficient evidence to develop future strategies to generalize



(a) Actual impedance for joint MCP



(b) Actual impedance for joint PIP

FIGURE 5.7: Sensed impedance calculations in order to study the stability: (a-a) actual impedance value over time for experiment Figure 5.4 (a), (a-b) rotation around active joint MCP and its virtual limit, (b-a) actual impedance value over time for experiment Figure 5.4 (b), (b-b) rotation around active joint PIP and its virtual limit.

the stiffness rendering task with no pre condition about the task or the object. Then, ensuring stability at all times during the experiments as shown in Figure 5.7 proves the usability of proposed stiffness rendering algorithm for the proposed underactuated hand exoskeleton in a general task.



### 5.3 Stiffness Rendering for the Proposed Underactuated Hand Exoskeleton

Using Jacobian matrix as it is to calculate desired forces corresponding to desired joint torques calculated in Equation (5.3) creates uncertainties for passive joint forces. The lack of controllability of the passive joint creates uncertainties when a set of desired joint torques require the passive joint to put energy to the system even though its passivity introduces an underactuated constraint as  $\mathbf{F}_p = \mathbf{0}$ .

Such a constraint introduced by underactuation concept prevents all possible sets of finger torques  $\boldsymbol{\tau}$  obtained by Equation (5.3) and transmits another set of torques instead of the desired set. It is acceptable not to achieve all possible sets with underactuation due to other advantages of such a lack of controllability, but not knowing the transmitted forces is not very appealing, especially for a device with human interaction. Instead of using the desired set of joint torques and hope for the device transmission to be performed in the best manner, a proxy set that satisfies the underactuation constraint can be found with minimum distance to the desired set. Doing so, even though the original desired set of finger torques cannot be transmitted to finger joints, an optimized set will be ensured to be applied to user through mechanism, solving the issue of uncertainty.

The stiffness rendering algorithm can be improved by finding a proxy set of desired torques or a proxy set of desired pose satisfying this property, such that the underactuated device can turn the desired values to the actual values through force transmission. Implementing such an extra effort based on desired torques might seem more straightforward, since the constraint itself is defined in the force space. However, stiffness rendering tasks are designed on the motion space, simply by forming resistance forces based on user's interactions in the virtual environment. Therefore, another rendering strategy should be defined to merge the same underactuated constrain with joint rotations. All these rendering strategies will be detailed in this section as much as experimental results performed by the underactuated hand exoskeleton. The feasibility of the force transmission will be shown by comparing the theoretical desired force values to the real measurements collected by simple FSR sensors. Then, the stability of the controller will be discussed based on the actual impedance calculations.

Each strategy will be detailed and given theoretical examples to highlight the idea behind it. Then, it is implemented on the hand exoskeleton control algorithm to compare the actual and desired torques acting on the finger joints. To do so, a FSR sensor has been

inserted between each finger phalange and passive slider that provides the connection between the user. Furthermore, the stability of the device is investigated by calculating the actual impedance. Finally both strategies are compared to each other in terms of defining the motivations why to implement them.

### 5.3.1 Rendering Strategy based on Joint Torques

Instead of suffering from such an incompatibility between the desired and actual torques around the finger joints, the constraint imposed by the passivity of sensory joint ( $F_p = 0$ ) can be used to create an applicable ratio between the joint torque values  $\tau_1$  and  $\tau_2$ . As discussed in Chapter 4, the passive joint is defined as joint  $B$ , where the additional potentiometer was placed to have an additional sensory measurement for pose estimation. The ratio between the applicable torques can be used to achieve a proxy set of desired torques that the underactuated exoskeleton can actually transmit to the finger joints. Keeping the distance between the proxy set and the desired set of joint torques might provide an efficient stiffness rendering perception while allowing the transmitted torques to be known during operation.

With this motivation, the passivity constraint  $F_p = 0$  can be used to achieve a relation, which bounds the desired torques acting on set of MCP and PIP joints  $\boldsymbol{\tau}$ , as in Equation (5.5). In fact, Equation (5.5) defines a relation between torque values around the finger joints for a given orientation. In particular, for the proposed exoskeleton with 2 *DoF* and a single actuator provides a ratio between joint torques at a given instant by the underactuated exoskeleton and the exoskeleton will fail to transmit a set of desired torques if they do not satisfy this ratio.

When a set of desired joint torques ( $\boldsymbol{\tau}$ ) is calculated from the stiffness rendering in Equation (5.3) that do not satisfy the constraint in Equation (5.5), a proxy set of desired finger torques ( $\boldsymbol{\tau}^* = (\tau_1^*, \tau_2^*)^T$ ), which is ensured to satisfy the passivity constraint, can be calculated with minimum distance to the desired set  $\boldsymbol{\tau}$ . Instead of  $\boldsymbol{\tau}$ , the proxy set  $\boldsymbol{\tau}^*$  will be used to generate the force command for control algorithm in Equation (5.2) to execute the stiffness rendering task. Satisfying the given constraint in Equation (5.5) ensures the underactuated device to transmit the proxy set of torques for user's joints in real life. Even though the finger joints would not achieve the exact force feedback calculated by the stiffness rendering, being able to monitor and control the actual torque values that are being transmitted by the exoskeleton will be ensured.

The stiffness rendering algorithm by finding the proxy set of applicable finger torques  $\boldsymbol{\tau}^*$  with a minimum error from the set of desired torque calculated by rendering algorithm ( $\boldsymbol{\tau}$ ) can be expressed mathematically as in Equation (5.6).

$$\begin{aligned} F_a^* &= \mathbf{J}_a^T \boldsymbol{\tau}^*, \text{ with} \\ \boldsymbol{\tau}^* &= \underset{\boldsymbol{\tau}}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{\tau} - \boldsymbol{\tau}^*\|^2, \text{ s.t. } \mathbf{J}_p^T \boldsymbol{\tau}^* = 0, \end{aligned} \quad (5.6)$$

The minimization problem with the underactuated constraint can also be expressed using the pseudo-inverse as in Equation (5.7).

$$\boldsymbol{\tau}^* = (\mathbf{I} - \mathbf{J}_p (\mathbf{J}_p^T \mathbf{J}_p)^{-1} \mathbf{J}_p^T) \boldsymbol{\tau} \quad (5.7)$$

Using the optimized forces acting on phalanges as in Equation (5.7), the actuator forces can easily be obtained as:

$$F_a^* = \mathbf{J}_a^T (\mathbf{I} - \mathbf{J}_p (\mathbf{J}_p^T \mathbf{J}_p)^{-1} \mathbf{J}_p^T) \mathbf{K}_{cont} \Delta \mathbf{q} \quad (5.8)$$

where  $\Delta \mathbf{q} = (\mathbf{q}_d - \mathbf{q})$ . The sensed impedance of the given strategy  $F_a/\Delta \mathbf{q}$  can be calculated simply as:

$$\mathbf{S}_a = -\mathbf{J}_a^T (\mathbf{I} - \mathbf{J}_p (\mathbf{J}_p^T \mathbf{J}_p)^{-1} \mathbf{J}_p^T) \mathbf{K}_{cont}. \quad (5.9)$$

The idea behind the given strategy can be shown with a simple simulation setup before the real time implementation. The finger joints were simulated to perform a closing movement for this simulation, while Jacobian matrix ( $\mathbf{J}$ ), actuator displacements ( $l_x$ ), finger pose estimation ( $\mathbf{q}$ ) and rotation around the passive joint ( $q_B$ ) were recorded. Through these recordings, a random finger pose was selected as  $\mathbf{q} = (51^\circ, 51^\circ)^T$  while a desired pose was assumed to be  $\mathbf{q}_d = (30^\circ, 40^\circ)^T$ . The stiffness matrix around the finger joints based on the virtual object was set as  $\mathbf{K}_{cont} = (1 \ 0; 0 \ 1)$ . Under

the given assumptions, the desired torque for the given pose was calculated as  $\boldsymbol{\tau} = (-0.36652, -0.19199)^T$  using Equation (5.3). Figure 5.8 provides the scheme of the given example in terms of the desired torques and how to obtain an optimized set of desired torques that the actuator can actually provide.

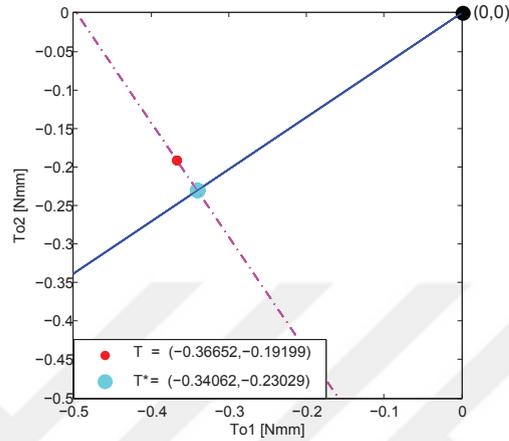


FIGURE 5.8: Simulation plot for the desired torque optimization based on minimizing the error between the real and the proxy torque sets.

The proposed exoskeleton was designed with 2 *DoF* mobility with a single actuator, therefore expressing the optimization process with a 2 – *D* sketch gives a clear idea of the strategy. The same design property forms a ratio between joint torques ( $\tau_2 / \tau_1$ ) using the underactuated constraint defined previously in Equation (5.5). This ratio was shown in Figure 5.8 as the blue constant line, passing through the origin to represent the direction of applicable torques to finger joints for the given instant based on the actuator force values. For the chosen finger pose, the Jacobian matrix caused this ratio as 0.6761, as the slope of the blue line. The red point in Figure 5.8 shows the set of desired torques  $\boldsymbol{\tau}$  calculated using stiffness rendering algorithm in Equation (5.3). Since  $\boldsymbol{\tau}$  does not lay on the blue line, which represents the applicable sets of joint torques, a proxy set of torques  $\boldsymbol{\tau}^*$  needs to be calculated as shown in Equation (5.7), which was found as  $\boldsymbol{\tau}^* = (-0.34062, -0.23029)^T$ .

The fact that proxy set of torques  $\boldsymbol{\tau}^*$  lay on the blue line unlike desired set of torques  $\boldsymbol{\tau}$  promises these torque values to be transmitted to finger joints. The verification of minimum distance between  $\boldsymbol{\tau}^*$  and  $\boldsymbol{\tau}$  was shown simply by drawing a secondary direction line, which was shown as pink dashed line, perpendicular to the blue line. Since both points are connected by the perpendicular line, the minimization issue can be proven. The calculated proxy set of desired torques are mapped to the actuator forces using the Jacobian transpose as  $\mathbf{F}_a^* = \mathbf{J}_a^T * \boldsymbol{\tau}^*$ .

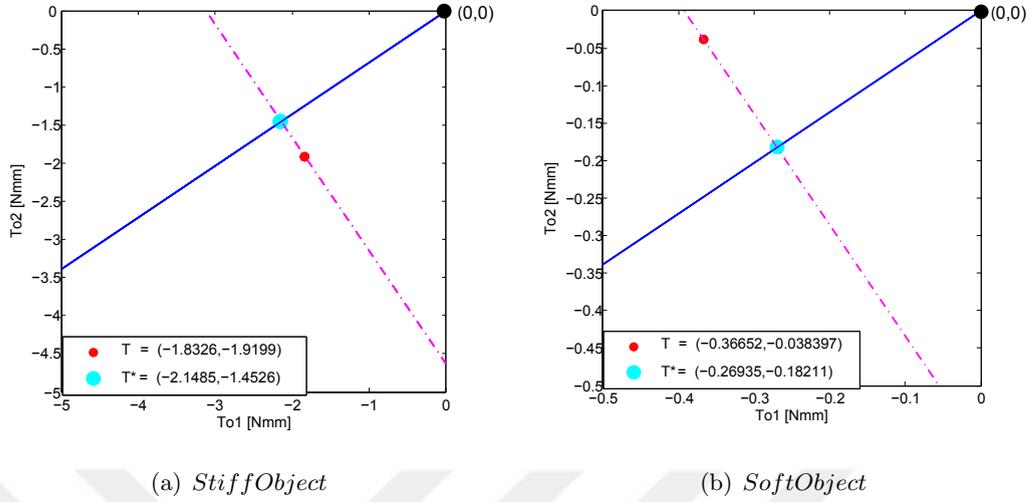


FIGURE 5.9: Desired torque optimization for different stiffness assumptions: (a)  $K1_{cont} = [5 \ 0; \ 0 \ 10]$ ; (b)  $K2_{cont} = [1 \ 0; \ 0 \ 0.2]$ .

The impact of stiffness matrix and stiffness ratio between joints can easily be observed by running the same simulation again. Two random stiffness matrices were chosen with different ratios as  $K1_{cont} = [5 \ 0; \ 0 \ 10]$  (Figure 5.9(a)) and  $K2_{cont} = [1 \ 0; \ 0 \ 0.2]$  (Figure 5.9(b)). Changing the stiffness values change the desired torque  $\tau$  with respect to the blue line depicted by Equation (5.5), which stays constant despite of the stiffness change. Naturally, the proxy sets of desired torques  $\tau^*$  are found to be different for each experiment, even though the idea remains the same as in Figure 5.8.

Once the proposed rendering strategy was developed for the simulation, it was implemented on the hand exoskeleton for a real time experiment. For a feasibility experiment, the user was asked to reach an imaginary virtual object with joint limitations randomly selected as  $\mathbf{q}_{lim} = (30^\circ, 30^\circ)^T$  and gets in contact with the object using both finger joints. The stiffness of the virtual object was selected in a soft behavior such that the user was allowed to exceed the joint limits after the first contact to complete the second joint contact as well. The desired pose  $\mathbf{q}_d$  was computed during operation simultaneously based on Algorithm 3. Using the instantaneous pose estimation, the desired torques were calculated using Equation (5.3), where the stiffness matrix was selected intuitively as  $K_{cont} = [100 \ 00; \ 0 \ 100]$ .

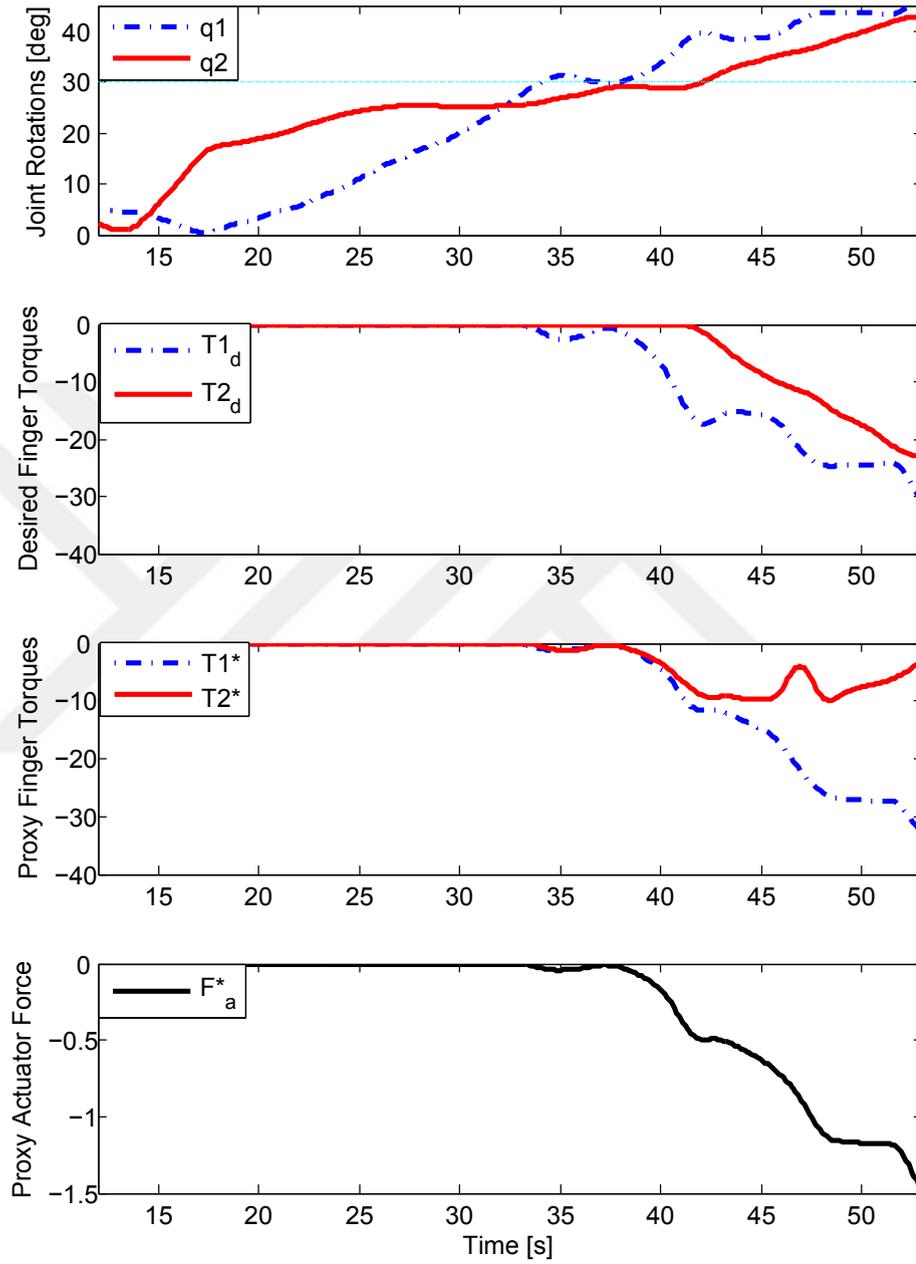


FIGURE 5.10: Desired torque optimization based on minimizing the error between the real and the proxy torque sets: (a) trajectory of the real time task performed by the user  $\mathbf{q}$  with the virtual limits  $\mathbf{q}_{lim}$  defined for the joints, (b) desired torque values  $\boldsymbol{\tau}$  calculated for user's trajectory, (c) optimized proxy set of torque values  $\boldsymbol{\tau}^*$  calculated for user's trajectory and (d) corresponding actuator force for the proxy torque values  $F_a^*$ .

Figure 5.9 (a) shows the movement of finger joints within the defined limits regarding the virtual contact with the grasping object. The corresponding desired torques  $\boldsymbol{\tau}$  are shown in Figure 5.9 (b). Meanwhile, the algorithm depicted in Equation (5.6) was being performed to observe the proxy set of desired torques  $\boldsymbol{\tau}^*$  that can be reached by the hand exoskeleton transmission, which can be observed in Figure 5.9 (c). Once the proxy set of desired torques are obtained, the desired forces for the actuator  $F_a$  can be calculated through Jacobian transpose through  $F_a = \mathbf{J}_a^T \boldsymbol{\tau}^*$  to be used for the control algorithm presented in Figure 4.24.

The efficacy of force transmission through proposed underactuated hand exoskeleton was measured by inserting FSR sensors between user's phalanges and passive sliders of the exoskeleton, as shown in Figure 5.11. As stated before, a silicon interface was inserted between FSRs and device's sliders to decrease the sensibility of sensors with respect to the form of physical interaction. Even though increasing the distance between user's finger and mechanism would change the kinematics and Jacobian analyses, FSR sensors provide a rough understanding of force transmission compared to desired ones. In order to make a comparison based on finger torques, FSR measurements were converted to joint torques simply as  $\tau_{FSR} = F_{FSR}/c_i$ , where  $c_i$  represents the linear displacement of passive linear slider attached to the corresponding finger phalange.



FIGURE 5.11: How additional FSR sensors are inserted between the hand exoskeleton and the user.

Figure 5.12 show sufficient evidence that optimized proxy set of desired torques were applied to user's joints through the device transmission, even though lack of accurate measurements of FSR sensors might create some imperfections during operation.

Comparing actual and desired joint torques as in Figure 5.12, one can say that obtaining proxy desired torques can overcome the uncertainties of underactuation concept. Even

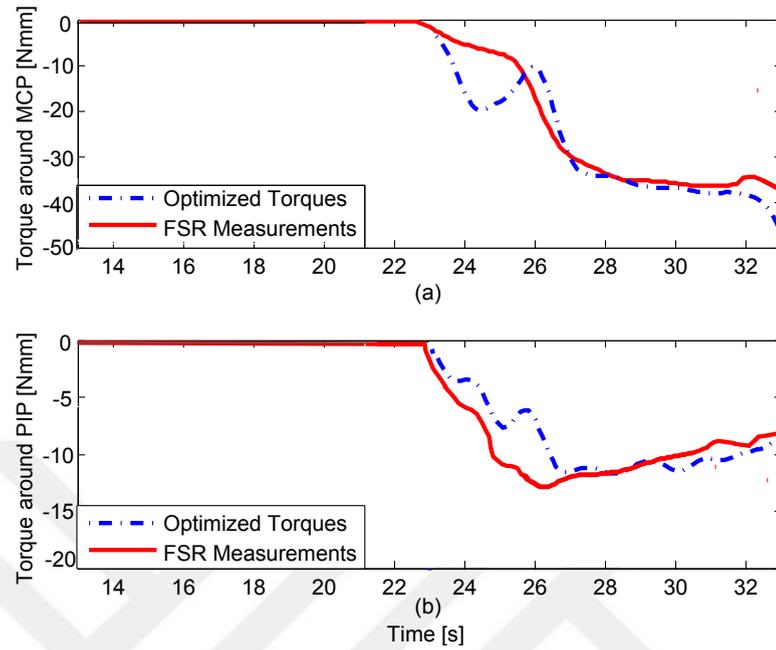


FIGURE 5.12: Comparison between the calculated proxy set of desired torques and measured torques through FSR measurements.

though proxy set of joint torques cannot assist user's fingers to reach the desired finger pose  $\mathbf{q}_d$ , torques being applied to finger joints can be tracked and controlled with this approach for the sake of user's safety and controllability of the device.

### 5.3.2 Rendering Strategy based on Finger Pose

Even though optimizing force transmission based on the underactuation constraints by finding a proxy set of applicable torque values, it might be not very efficient in terms of haptics, since the finger pose is not controlled or tracked by the previous strategy. Alternatively, a rendering strategy can be developed to find a proxy set of finger pose  $\mathbf{q}^*$  to replace the desired pose  $\mathbf{q}_d$ , which satisfies the underactuated constraint for the instantaneous finger pose  $\mathbf{q}$ .

The underactuated grippers were designed simply by introducing more soft or elastic passive joints to perform the force transmission along joints. A similar idea was adopted to create the proposed underactuated hand exoskeleton based on the spring like behavior of user's finger joints anatomically. In particular, anatomical behavior of user's finger joint rotations are modified based on instantaneous stiffness behavior based on user's intention. Using the joint stiffness estimation based on the previous behavior of finger joints, future finger pose might be predicted that will be covered by the finger, assuming the user will continue to move at the same mode. Such a prediction regarding future reachable path by user allows a proxy set of reachable finger pose  $\mathbf{q}^*$  with minimum distance to desired set of joints  $\mathbf{q}_d$  to be found. In practice, running a control board with 1 kHz allows these stiffness estimations to be made as fast as the user changes the mode to move. Each stiffness joint value  $K_j$  can be calculated as

$$K_j = \frac{\tau_j}{\dot{q}_j} \text{ s.t. } j = 1, 2 \quad (5.10)$$

where  $K_j$  values are gathered diagonal to form the experimental stiffness matrix  $\mathbf{K}_{stiff}$ . For an estimated stiffness matrix of joint stiffness  $\mathbf{K}_{stiff}$ , set of joint torques can be calculated using Equation (5.11).

$$\boldsymbol{\tau} = \mathbf{K}_{stiff} \Delta \mathbf{q} \quad (5.11)$$

where  $\Delta \mathbf{q}$  is the difference between a reachable pose and the initial configuration of finger pose ( $\partial \mathbf{q} = \mathbf{q}^* - \mathbf{q}_o$ ) and is the estimation matrix values. The output forces should calculate to reach actuator forces  $F_a$  and passive forces  $F_p$  through Equation (??). Then, the minimization problem is defined as

$$\begin{aligned}
F_a^* &= \mathbf{J}_a^T \mathbf{K}_{stiff} (\mathbf{q}^* - \mathbf{q}_o), \text{ with} & (5.12) \\
\mathbf{q}^* &= \underset{\mathbf{q}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{q}_d - \mathbf{q}\|^2, \text{ s.t.} \\
\mathbf{J}_p^T \mathbf{K}_{stiff} (\mathbf{q}^* - \mathbf{q}_o) &= \mathbf{0},
\end{aligned}$$

Lagrangian optimization method allows the given problem to be expressed in a closed-loop form in Equation (5.13), where  $\mathbf{K} = \mathbf{K}_{stiff} \mathbf{J}_p$ .

$$\mathbf{q}^* = \mathbf{q}_d - \mathbf{K} (\mathbf{K}^T \mathbf{K})^{-1} \mathbf{K}^T (\mathbf{q}_d - \mathbf{q}_o) \quad (5.13)$$

Once the proxy set of joints are achieved through Equation (5.13), desired actuator force to reach the given pose can be calculated easily as:

$$F_a^* = \mathbf{J}_a^T \mathbf{K}_{cont} (\mathbf{I} - \mathbf{K} (\mathbf{K}^T \mathbf{K})^{-1} \mathbf{K}^T) \Delta \mathbf{q} \quad (5.14)$$

where  $\Delta \mathbf{q} = (\mathbf{q}_d - \mathbf{q}_o)$ . The sensed impedance  $F_a / \Delta \mathbf{q}$  of the overall system for configurational space can be depicted as:

$$\mathbf{S}_a = -\mathbf{J}_a^T \mathbf{K}_{cont} (\mathbf{I} - \mathbf{K} (\mathbf{K}^T \mathbf{K})^{-1} \mathbf{K}^T). \quad (5.15)$$

The joint space based rendering strategy aims to estimate the future joint pose based on the previous behavior of the user, since the underactuation imposes no constraint in the joint space. Due to the lack of underactuated constraint in the joint space, the estimation of reachable finger pose proxy was put into torque form in order to take advantage from the same constraint.

The feasibility of the optimization algorithm based on desired pose in Equation (5.14) can be shown with the same example as before, where the current finger pose is selected as  $\mathbf{q} = (51^\circ, 51^\circ)^T$  while the desired displacement is assumed to be  $\mathbf{q}_d = (30^\circ, 40^\circ)^T$ . Since the finger joints are moved by a simulation together with accurate relation, the

joint stiffness elements were calculated as 1, which forms the stiffness matrix  $\mathbf{K}_{stiff} = [1 \ 0; 0 \ 1]$ . The proxy finger pose  $\mathbf{q}^*$  to replicate desired pose  $\mathbf{q}_d$  was found simply using Equation (5.13). Such a proxy pose search was shown in Figure 5.13 in a similar way to Figure 5.8 in the operational space, rather than force space.

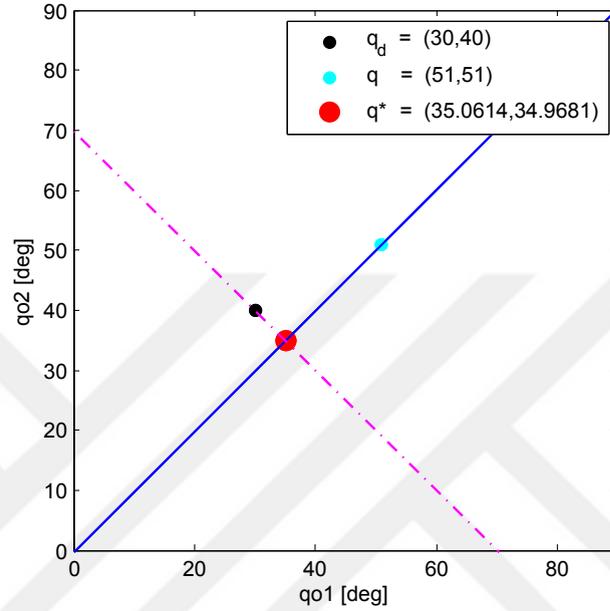


FIGURE 5.13: Simulation plot for configurational space optimization problem based on minimizing the distance between desired finger pose and proxy finger pose.

The constant blue line in Figure 5.13 possible reachable sets of finger pose by the assistance of hand exoskeleton in the next time step for given finger pose  $\mathbf{q}$  using the stiffness matrix  $\mathbf{K}_{stiff}$ . Light blue dot shows the actual finger pose  $\mathbf{q}$ , while the black dot shows the desired pose  $\mathbf{q}_d$ . Since the desired pose  $\mathbf{q}_d$  does not lay on the blue line, it cannot be reached only by actuating the exoskeleton unless the user changes his behavior. That is why a proxy set of finger pose  $\mathbf{q}^*$  is needed to be calculated through optimization in Equation (5.13), which was found as  $(35.0614^\circ, 34.9681^\circ)^T$ . The verification of reaching the minimum distance between  $\mathbf{q}^*$  and  $\mathbf{q}_d$  was performed simply by plotting an additional line that is perpendicular to the blue line and passes from  $\mathbf{q}_d$ . Since the proxy pose  $\mathbf{q}^*$  occurs at the interaction point of both lines, the minimization problem can be verified.

This optimization approach that is based on capturing user's previous behavior can be interpreted with extreme circumstances, as displayed previously in Figure 5.4. In particular, changing the finger joint trajectories in the simulation to obtain the ratio between joint stiffness values in  $\mathbf{K}_{stiff}$  matrix as approximately 0.01 and 100 define

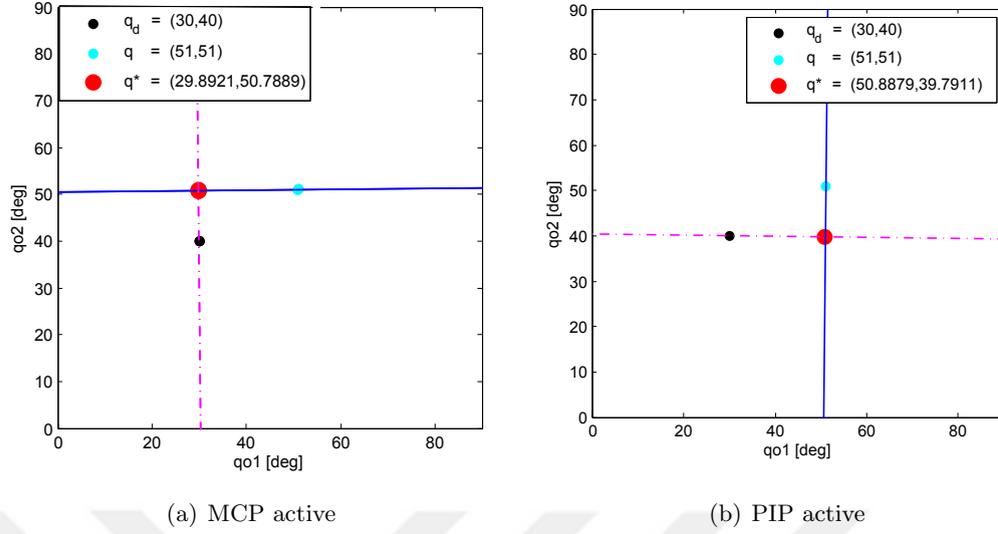


FIGURE 5.14: Desired displacement optimization for different previous behavior: (a) only MCP joint was being rotated (ratio = 0.01); (b) only PIP joint was being rotated (ratio = 100).

scenarios where only MCP or PIP joint rotates while the other joint remains passive. The same optimization scenario was applied to both simulations and the results are represented in Figure 5.14. It can be observed that a proxy for desired finger pose can be observed even in extreme situations.

The proposed optimization algorithm to find a proxy set of desired finger joints  $\mathbf{q}^*$  was tested using the hand exoskeleton. During the real time experiment, the desired finger pose is randomly selected as  $\mathbf{q}_d = (30^\circ, 30^\circ)^T$ . The stability of the control algorithm is ensured by running the control algorithm with  $1kHz$  frequency within the control board. Figure 5.15 (a) shows the movement of finger joints  $\mathbf{q}$  along with virtual joint limits  $\mathbf{q}_{lim}$ . For given instant, Equation (5.13) calculates a proxy set of finger joints that can be reached based on joint stiffness approximation as Figure 5.15 (b). Figure 5.15 (c) and Figure 5.15 (d) show the corresponding joint torques  $\boldsymbol{\tau}^*$  and actuator force  $F_a^*$  for optimized proxy pose  $\mathbf{q}^*$ .

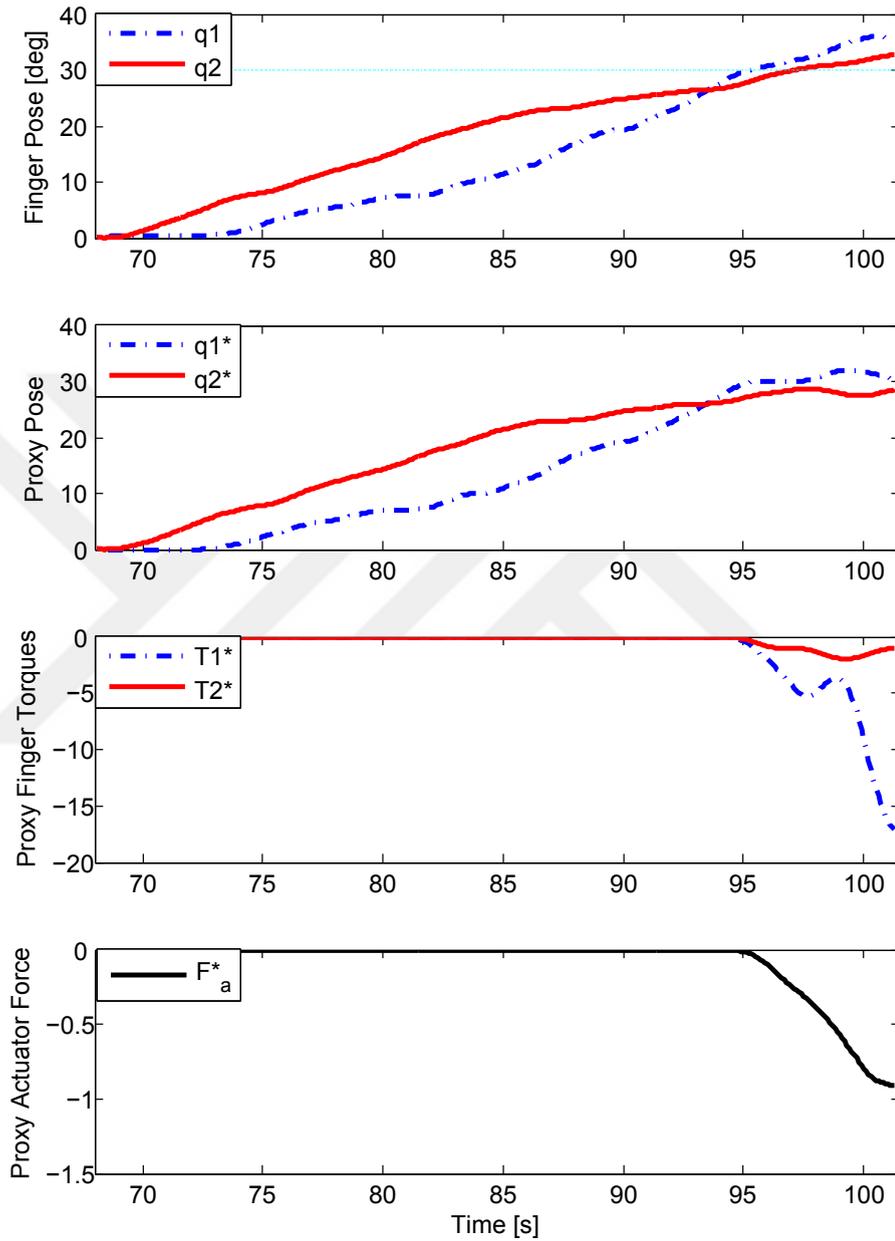


FIGURE 5.15: Proxy pose optimization based on minimizing the error between the real and the proxy pose sets: (a) trajectory of the real time task performed by the user  $\mathbf{q}$  with the virtual limits  $\mathbf{q}_{lim}$  defined for the joints, (b) optimized proxy set of finger pose  $\mathbf{q}^*$  calculated for user's trajectory, (c) desired finger torques  $\boldsymbol{\tau}^*$  calculated using proxy set of finger pose  $\mathbf{q}^*$  and (d) corresponding actuator force for the proxy torque values  $F_a^*$ .

The validity of rendering strategy in the operational space needs to be shown by studying the validation of user's behavior. In particular, the efficacy of the control algorithm under operational space strategy can be studied simply by comparing the ratio between MCP and PIP joints using:

$$\left( \frac{q^*(i) - q(i)}{\|q^*(i) - q(i)\|} \right)^T \frac{q(i+1) - q(i)}{\|q(i+1) - q(i)\|}. \quad (5.16)$$

Doing so, the controller will be revealed to move in the estimated behavior and calculated proxy pose using user's previous estimation. Such a multiplication of ratios can be calculated to be similar when the output of Equation (5.16), as depicted in Figure 5.16.

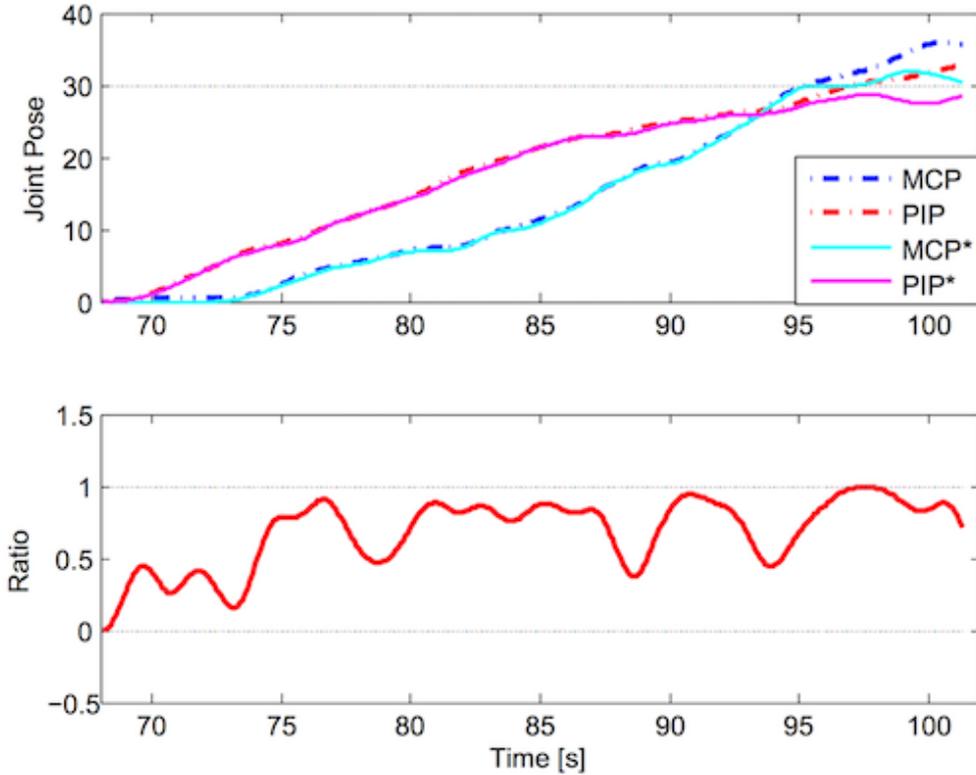


FIGURE 5.16: Validation of user's behavior estimation.

Since the stiffness rendering task, which applies forces to the user in order to impose the proxy finger pose as the reference, applies forces only after user reaches the virtual pose limit, the same ratio between the direction of proxy pose and user's future movement should be focused only after the virtual limit, as in Figure 5.17.

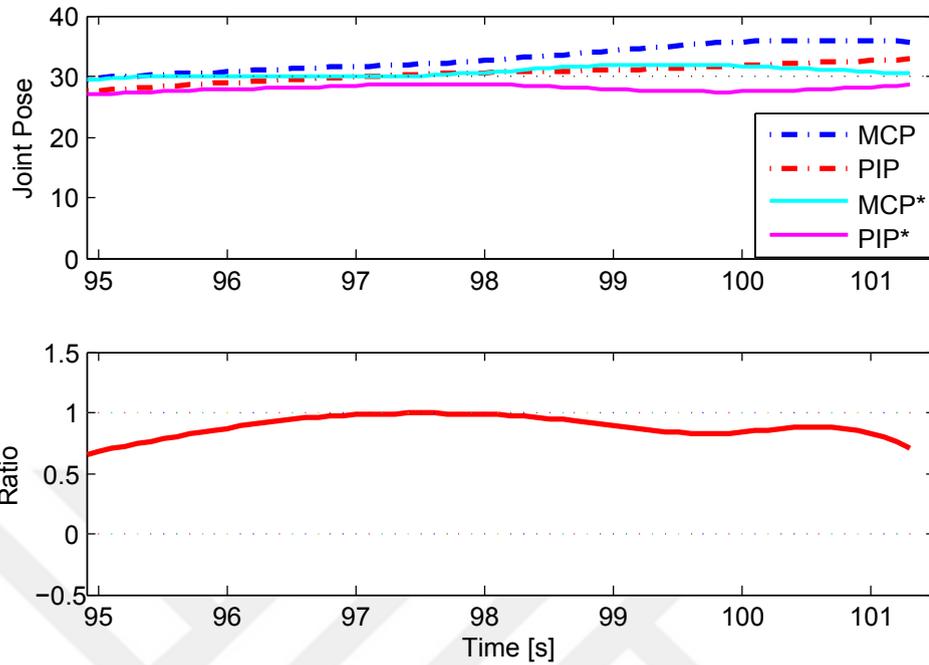


FIGURE 5.17: Validation of user's behavior estimation after user reaches the virtual limit pose.

Having ratio in Equation (5.16) close to value 1 signifies the closeness between the differences with proxy and actual finger pose and future and actual finger pose. Therefore, the efficacy of the control algorithm can be issued.

Similar to the previous strategy, the efficacy of force transmission of the given rendering strategy based on proxy pose should be proven using FSR measurements as depicted in Figure 5.11. Figure 5.18 shows the torques around MCP and PIP joints using FSR measurements and desired torques calculated by the control algorithm during the experiment shown in Figure 5.15.

As expected, despite of the imperfections caused by FSR sensors, Figure 5.18 shows that measured finger torques are highly correlated with desired ones. Therefore, one can say that even by running an optimization to find an achievable set of desired pose  $\mathbf{q}^*$  the torques that are being applied to finger joints can be estimated. Furthermore, unlike the optimization based on desired torques, the rendering strategy based on proxy finger pose still performs the control algorithm in the configuration space.

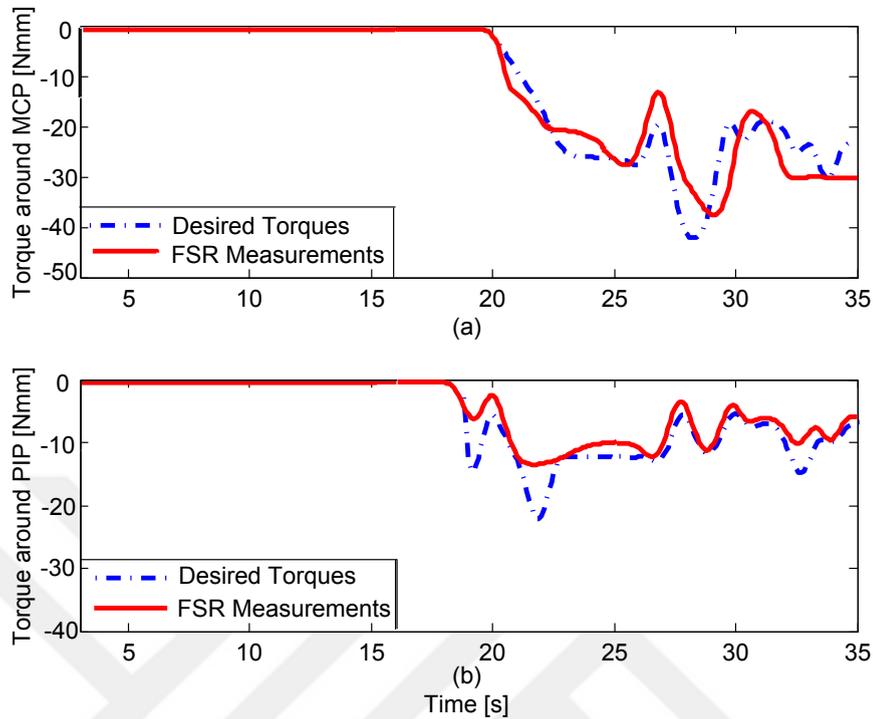


FIGURE 5.18: Comparison between desired torques and actual torques measured by FSR sensors during rendering strategy based on proxy joint pose.

### 5.3.3 Comparison between Rendering Strategies

Previously two different rendering strategies were proposed to overcome the uncertainties of stiffness rendering tasks for underactuated devices. These two methods were developed simply by computing proxy desired sets that might satisfy the underactuated constraints. The first method aims to find an applicable proxy set in the force space, while the second method reaches a reachable proxy set in the configuration space, as shown in Figure 5.19.

In particular, the optimization algorithm based on desired torques calculate the set of desired torques  $\tau_d$  to be rendered based on adjustable stiffness values, which are set based on the properties of the virtual object, and calculates a proxy set of torques  $\tau_d^*$  that can be applied by the underactuated exoskeleton. Besides the fact that desired finger pose  $\mathbf{q}_d$  cannot be achieved due to the lack of controllability of underactuation concept, resulting finger pose cannot be estimated due to closing the control loop based on proxy torque set. Since the stiffness rendering task is theoretically defined based on joint displacements, the lack of joint estimation to be reached might require further analysis. Nevertheless, for implementations, where the accuracy of rendered forces to

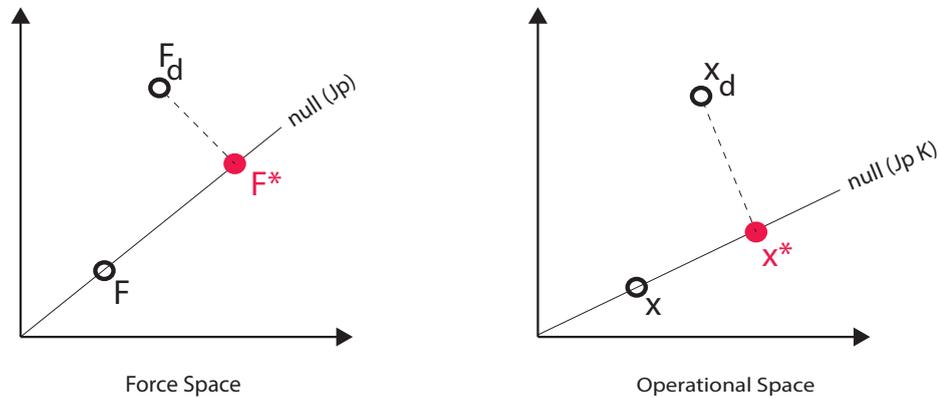


FIGURE 5.19: Stiffness rendering algorithms for force space and operational space individually.

finger joints is more crucial rather than the accuracy of imposed finger pose, rendering strategies based on force space might be more useful.

On the other hand, the optimization algorithm based on configuration space calculates the behavior of user's movement and estimates the future trajectory for finger joints. Once the future trajectory is estimated, desired finger pose is optimized to a proxy finger pose  $\mathbf{q}^*$ , which satisfies the estimated trajectory and the underactuation constraint. The optimized proxy finger pose is used to calculate corresponding joint torques using the force relation of a spring with estimated stiffness values. The given torques are applied the underactuation constraint, then the rendering stiffness is applied for the calculated torques. Such an optimization method defines the stiffness rendering task assuming the finger joints as imaginary springs. Running the optimization problem based on minimizing the applicable finger pose might improve the performance of underactuated device in the configuration space better than proxy torque strategy.

Passivity of the underactuated exoskeleton throughout the operation can be studied for system stability. For both algorithms, the sensed impedance  $F_a^*/\mathbf{q}$  was expressed previously using Equation (5.9) and Equation (5.15). The impedance of both strategies can be compared using the same experiment, where the user perceives force feedback produced by the proxy pose strategy, while sensed impedance for proxy torque strategy was computed in an offline manner. The motivation of such a choice is that force feedback is highly crucial in order to obtain meaningful joint stiffness prediction to form the matrix  $\mathbf{K}_{stiff}$  for proxy pose strategy. The experimental expressions of sensed impedance for proxy torque and proxy pose strategies can be seen in Figure 5.20, for the experiment shown in Figure 5.15.

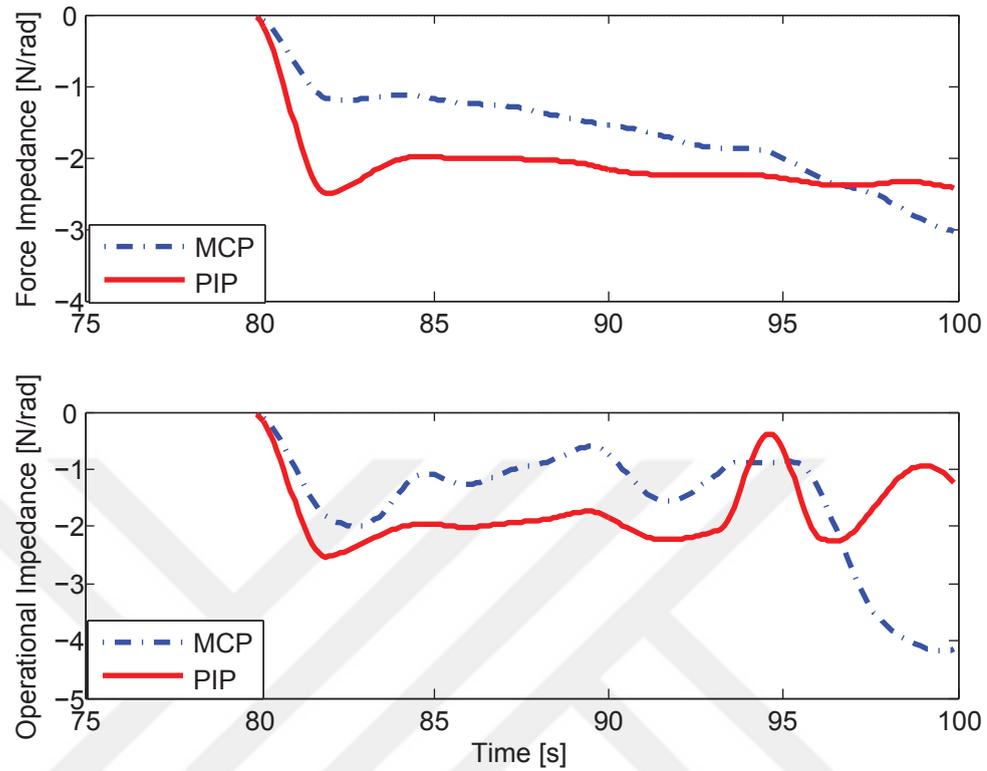


FIGURE 5.20: Sensed impedance  $S_a = F_a^*/\mathbf{q}$  calculated during the experiment in Figure 5.15 for (a) proxy torque strategy in an offline manner and (b) proxy pose strategy in an online manner.

Composing Equation (5.15) is different than Equation (5.9) only because the estimation of joint stiffness values are considered based on previous movement of the user. Since the user cannot ensure a smooth behavior while opening/closing fingers in a free environment, such an estimation can cause the impedance plots to be more chattering. Obtaining all negative values for joint stiffness values of both methods signify stability of the system.

## 5.4 Proxy based Haptic Rendering

Once the possibility of developing stiffness rendering strategies for underactuated systems by real time experiments performed by the proposed underactuated hand exoskeleton, the similar idea can be generalized and extended to haptic rendering. In this scenario, the user will be asked to control an avatar in a virtual environment simulation, where the hand exoskeleton will monitor the user's movements and provide force feedback in case of a contact. Even though a different notation will be introduced, that is more favorable for the haptic rendering point of view, the idea behind the soon to be introduced proxy based haptic rendering is the same as the rendering strategy based on finger pose.

The novelty of the proposed proxy-based haptic rendering is to provide fully simulated and constrained by the virtual environment, even with soft objects and to be constrained to the subspace defined by the actuated DoF of the device. Thanks to the subspace proxy, we can compute feedback haptic commands that are optimally constrained to actuated DoF, and we can do this using regular controllers as in the standard rendering method. It admits haptic and virtual configuration spaces of different dimensionality, that are not defined using the physical constraints of fingers as we did previously, and connected through nonlinear mappings.

### 5.4.1 Notation

Since the following rendering strategies will be proposed for general haptic simulations, countless configurational contact points with virtual objects will be used to compute actuator forces for feedback. With this motivation, a different, generic notation will be used as  $Q = Q_a \times Q_n$  the configuration space of the haptic device, with  $Q_a$  the actuated configuration space (resulting from actuated DoF), and  $Q_n$  the non-actuated configuration space (resulting from non-actuated DoF).  $\mathbf{q} = (\mathbf{q}_a; \mathbf{q}_n)^T \in Q$  represents a device state, with  $\mathbf{q}_a \in Q_a$  the state of actuated DoF, and  $\mathbf{q}_n \in Q_n$  the stated of non-actuated DoF. Similarly, we denote as  $\chi$  the (unconstrained) configuration space of the virtual object.  $\mathbf{x} \in \chi$  represents a virtual state.

There is a kinematic mapping  $f : Q \rightarrow \chi$  from the configuration space of the device to the configuration space of the virtual object. Then, we can compute a virtual state corresponding to a device state, i.e.,  $\mathbf{x} = f(\mathbf{q})$ . Figure 5.21 shows a conceptual representation of haptic and virtual configuration spaces, very similar to the one in Figure 5.19,

together with their mapping. We denote as  $\mathbf{J} = \partial x / \partial q$  the Jacobian of the mapping  $f$ , which allows a linearization of the typically nonlinear mapping  $f$ . We also split the Jacobian into  $\mathbf{J} = (\mathbf{J}_a; \mathbf{J}_n)$ , with  $\mathbf{J}_a = \partial x / \partial q_a$  and  $\mathbf{J}_n = \partial x / \partial q_n$  with respect to the actuated and non-actuated DoF respectively. Only when the number of DoF of  $\chi$  and  $Q$  match,  $\mathbf{J}$  is square, and the mapping  $f$  may be invertible.

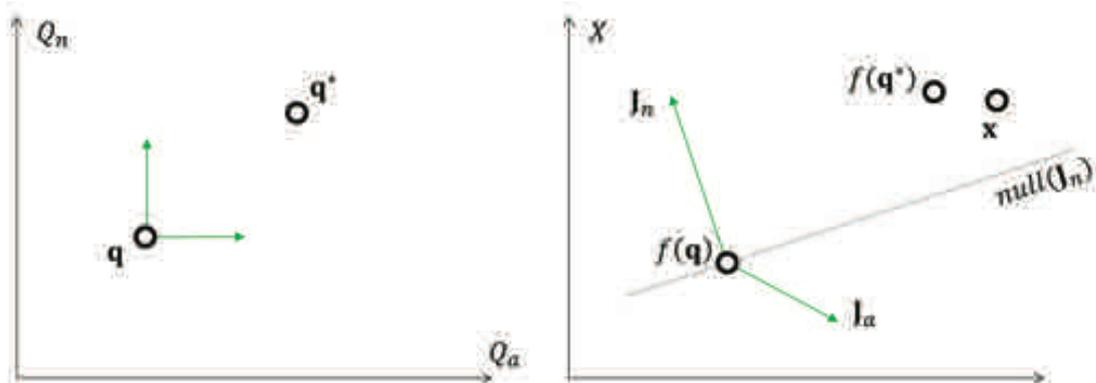


FIGURE 5.21: Schematic representation of the device configuration space  $Q$  (left) and the virtual configuration space  $\chi$  (right).  $\mathbf{q}$  represents the device state and  $f(\mathbf{q})$  its corresponding configuration in the virtual environment, i.e., the haptic probe;  $\mathbf{x}$  represents the standard proxy;  $\mathbf{q}^*$  represents the subspace proxy and  $f(\mathbf{q}^*)$  its corresponding configuration in the virtual environment. The images also represent the linear subspaces of actuated and non-actuated motion in the virtual environment,  $\mathbf{J}_a$  and  $\mathbf{J}_n$ .

Finally, we denote as  $\tau$  a device force vector, and as  $\mathbf{f}$  a generalized virtual force. Since the proposed hand exoskeleton is an impedance-mode rendering, such assumption will be made to proceed the optimization procedure. Nevertheless, all methods could be applied in admittance-mode rendering as well, simply by exchanging force commands with position commands.

### 5.4.2 Review of Proxy-based Haptic Rendering

This section provides a formal description of the standard proxy-based haptic rendering method as much the discussions regarding the major assumptions and the problems induced when the method is applied to underactuated devices. Furthermore the variation of proxy-based rendering to compute optimal actuator forces will be investigated, within the underactuation constraints. Even though such method typically satisfies passivity, it yields a configuration-dependent rendering impedance.

### 5.4.2.1 Standard rendering algorithm

Assume  $\mathbf{q}$  and  $f(\mathbf{q})$  define the state of the device and the corresponding position of the haptic probe in the virtual environment respectively. Proxy-based rendering computes a proxy  $\mathbf{x}$  that minimizes the distance to the probe according to some metric, subject to environment constraints. Then, the method computes a force  $\mathbf{f} = \mathbf{Z}_x \Delta \mathbf{x}$  in the virtual environment, based on some mechanical impedance  $\mathbf{Z}_x$  and the displacement from the probe to the proxy  $\Delta \mathbf{x} = \mathbf{x} - f(\mathbf{q})$ . Next, the force is transformed to the configuration space of the device using the Jacobian transpose approach:  $\boldsymbol{\tau} = \mathbf{J}^T \mathbf{f}$ . Finally, forces are displayed.

The standard proxy-based rendering algorithm makes important assumptions about the configuration spaces  $Q$  and  $X$ : their dimensionality is the same, the Jacobian  $\mathbf{J}$  is hence square, and the mapping  $f$  is invertible. This is the case, for example, in typical 3-DoF and 6-DoF haptic rendering systems. In haptic rendering of deformable objects using stylus devices, it is easy to extract a rigid subspace of the full deformable configuration space (using, e.g., rigid modes or a rigid handle), and define this rigid subspace as  $X$  for the purpose of applying the proxy-based rendering algorithm.

### 5.4.2.2 Analysis for underactuated systems

With underactuated devices, even if a full force vector  $\boldsymbol{\tau}$  is computed, force can obviously be rendered only on actuated DoF. The effective force resulting from the application of the proxy-based rendering algorithm to an underactuated device is then:

$$\begin{aligned} \boldsymbol{\tau}_a &= \mathbf{J}_a^T \mathbf{f} \\ &= \mathbf{J}_a^T \mathbf{Z}_x \Delta \mathbf{x}, \end{aligned} \tag{5.17}$$

Simple projection of the forces to the actuated DoF fails to reproduce target forces that lie in the null-space of the actuated DoF. In some popular types of underactuated haptic systems, it is easy to map the actuated DoF to a well defined subspace of the full virtual configuration space, e.g., in systems with 6-DoF input (translation and rotation) and 3-DoF output (force only). In these cases, the Jacobian  $\mathbf{J}$  is block diagonal, and the forces of virtual DoF that map to actuated DoF are matched exactly, while the forces of other virtual DoF are simply zero.

The rendering algorithm can formally be analyzed in terms of the displayed impedance  $\partial\tau/\partial\mathbf{q}$ . The rendering method is passive if the displayed impedance is negative definite, i.e., all its eigenvalues are negative. At this point, two main assumptions have been made regarding the following analysis: the local change of the proxy position due to the motion of the device is ignored ( $\partial\mathbf{x}/\partial\mathbf{q}$ ) and also the local change of the Jacobian is ignored ( $\partial\mathbf{J}/\partial\mathbf{q}$ ).

Combining the definition  $\tau = \mathbf{S}^T \tau_a$  with Equation (5.17), the expression shown in Equation (5.18) is obtained. It is important to emphasize that this expression does not give any guarantee that the displayed impedance is negative definite.

$$\frac{\partial\tau}{\partial\mathbf{q}} = -\mathbf{S}^T \mathbf{J}_a^T \mathbf{Z}_x \mathbf{J} \quad (5.18)$$

#### 5.4.2.3 Null-space force optimization

Alternatively, a variation of proxy-based haptic rendering can be studied, which accounts for underactuation prior to transforming the target force  $\mathbf{f}$  to the configuration space of the device. In essence, the method transforms a different force  $\mathbf{f}^*$ , as close as possible to the target force, but which yields no forces on non-actuated DoF. The approach can be formulated as a constrained optimization problem:

$$\begin{aligned} \tau_a &= \mathbf{J}_a^T \mathbf{f}^* \text{ with} \\ \mathbf{f}^* &= \operatorname{argmin} \|\mathbf{f}^* - \mathbf{f}\|^2, \text{ s.t. } \mathbf{J}_n^T \mathbf{f}^* = 0, \end{aligned} \quad (5.19)$$

which was expressed previously as Equation (5.6) in Section 5.3.1. The proposed optimization problem can be expressed in a closed-form solution, just as Equation (5.8) with the adjusted notation as:

$$\tau_a = \mathbf{J}_a^T (\mathbf{I} - \mathbf{J}_n (\mathbf{J}_n^T \mathbf{J}_n)^{-1} \mathbf{J}_n^T) \mathbf{Z}_x \Delta\mathbf{x}. \quad (5.20)$$

The interpretation of the equation above is that the method projects the target forces  $\mathbf{f}$  to the null-space of the nonactuated DoF prior to applying the Jacobian transpose. In this case, instead of Equation (5.9), the displayed impedance is:

$$\frac{\partial \tau}{\partial \mathbf{q}} = -\mathbf{S}^T \mathbf{J}_a^T (\mathbf{I} - \mathbf{J}_n (\mathbf{J}_n^T \mathbf{J}_n)^{-1} \mathbf{J}_n^T) \mathbf{Z}_x \mathbf{J}. \quad (5.21)$$

In a simple case where  $\mathbf{Z}_x$  is a uniform stiffness for all DoF of the virtual object, i.e.,  $\mathbf{Z}_x = k\mathbf{I}$ , then  $\partial \tau_a / \partial \mathbf{q}_a = 0$  and all eigenvalues of  $\frac{\partial \tau_a}{\partial \mathbf{q}_a}$  are negative, hence passivity is guaranteed. But this is not necessarily the case if the impedance  $\mathbf{Z}_x$  is more complex.

In addition, to ensure stability of the rendering, the stiffness of the displayed impedance must be bounded as a function of the sampling rate. As  $\frac{\partial \tau}{\partial \mathbf{q}}$  depends on the actuated and non-actuated Jacobian matrices  $\mathbf{J}_a$  and  $\mathbf{J}_n$ , stability imposes complex nonlinear conditions on the impedance  $\mathbf{Z}_x$ . That being said, one can conclude that, with null-space force optimization, maximization of rendering transparency depends in a complex nonlinear way on the mapping from device configuration space to virtual configuration space.

Based on these conclusions, instead of just optimizing rendered forces of the standard proxy-based method, this study seeks a novel rendering method that addresses the challenge of underactuation while remaining passive, and also simplifies maximizing transparency.

### 5.4.3 Rendering for Underactuated Devices

In this section, a novel haptic rendering method for underactuated devices is proposed. Firstly a nonlinear formulation of a subspace proxy constrained to actuated DoF is presented, then the problem is linearized to yield an efficient rendering method. Finally, the benefits of the method are analyzed compared to the two methods discussed in the previous section.

#### 5.4.3.1 Subspace proxy

The core of the proposed method is simple. The main motivation is to exploit all benefits of the proxy-based rendering method, namely: (i) accurate visual simulation of the virtual object, (ii) simple rendering of forces based on deviations between proxy and probe, and (iii) simple tuning of the rendering impedance.

To achieve this, and to circumvent the dimensionality difference of  $Q$  and  $\chi$ , two different proxies are defined. The classical proxy,  $x \in \chi$ , is a virtual object that is simulated by minimizing the distance to the haptic probe subject to environment constraints; and a *subspace proxy*,  $\mathbf{q}^* \in Q$ , is a proxy constrained to the actuated configuration space of the device  $Q$ . Thanks to the classical proxy  $x$ , the visual accuracy of the simulation is retained. Thanks to the subspace proxy  $\mathbf{q}^*$ , the forces can be computed directly based on the deviation  $\Delta \mathbf{q}_a = \mathbf{q}_a^* - \mathbf{q}_a$  in the actuated state of the device. And as a corollary, transparency is easily maximized by tuning the display impedance directly on the actuated DoF.

Given a haptic device state  $\mathbf{q}$ , the subspace proxy  $\mathbf{q}^*$  is proposed to be computed by finding the corresponding virtual configuration  $f(\mathbf{q}^*)$  that minimizes the distance to the proxy  $x$ . In practice, this is done by solving an optimization problem, as in Section 5.3.2.

Once the subspace proxy is computed, we can also compute the device force (on the actuated DoF)  $\tau_a$  based on a rendering impedance  $\mathbf{Z}_q$  and the deviation  $\Delta \mathbf{q}_a = \mathbf{S} \Delta \mathbf{q}$  on the actuated DoF alone. Formally, we define our subspace proxy-based haptic rendering as follows:

$$\begin{aligned} \tau_a &= \mathbf{Z}_q \mathbf{S} (\mathbf{q}^* - \mathbf{q}), \text{ with} \\ \mathbf{q}^* &= \operatorname{argmin} \|x - f(\mathbf{q}^*)\|^2. \end{aligned} \quad (5.22)$$

The main challenge of this formulation is that the mapping  $f$  is nonlinear, and finding  $\mathbf{q}^*$  requires solving a nonlinear optimization. Therefore, the next step would be to relax this challenge.

#### 5.4.3.2 Linearized subspace proxy

By linearizing the mapping  $f$  at the current device state  $\mathbf{q}$ , the optimization problem in Equation (5.22) turns into a simple quadratic optimization:

$$\begin{aligned} \tau_a &= \mathbf{Z}_q \mathbf{S} \Delta \mathbf{q}, \text{ with} \\ \mathbf{q} &= \operatorname{argmin} \|\Delta x - \mathbf{J} \Delta \mathbf{q}\|^2, \end{aligned} \quad (5.23)$$

with the following closed-form solution:

$$\tau_a = \mathbf{Z}_q \mathbf{S} (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \Delta x. \quad (5.24)$$

Please note that the proposed method in Equation (5.23) and Equation (5.24) is the same method in Section 5.3.2 presented as Equation(5.12) and Equation(5.14). In the previous method, simplifying the configurational space into stiff structure of human finger instead of defining a avatar simulation with limitless points gave us the ability to study the previous behavior of user to improve the future estimations. However, such an estimation was eliminated for this analysis.

The interpretation of the equation above is that the method projects the proxy deviation  $\Delta x$  onto the device DoF to compute a subspace proxy deviation prior to the force computation. This implies an important conceptual difference with respect to the standard proxy-based rendering method. The standard method transforms forces from the virtual environment to the device, whereas our method transforms displacements and keeps force computation at the device level.

The computational cost of the method is negligible. It requires the solution of a linear system whose size is given by the number of DoF of the device.

Similar to Section 5.4.2, the displayed impedance is analyzed, which in this case is:

$$\frac{\partial \tau}{\partial \mathbf{q}} = -\mathbf{S}^T \mathbf{Z}_q \mathbf{S}. \quad (5.25)$$

This impedance yields two notable results. First, passivity is easily enforced, simply by ensuring that the rendering impedance  $\mathbf{Z}_q$  is positive definite. Second, transparency is easily maximized, simply by setting the stiffness terms in  $\mathbf{Z}_q$  to the maximum allowed by stability constraints. Unlike the previous methods, the displayed impedance is not affected by configuration-dependent scaling factors.

#### 5.4.4 Results

The feasibility of the proposed rendering algorithm has been expressed in this study through the given optimization formulation. This approach enables simple and elegant formulation and interpretation, and it leaves the mathematical and implementation details to each particular type of device, virtual probe, and objective function. In this

section, a particular implementation for a single finger component of the proposed underactuated hand exoskeleton will be described for haptic rendering of soft grasping interactions, as shown in Figure 5.22.

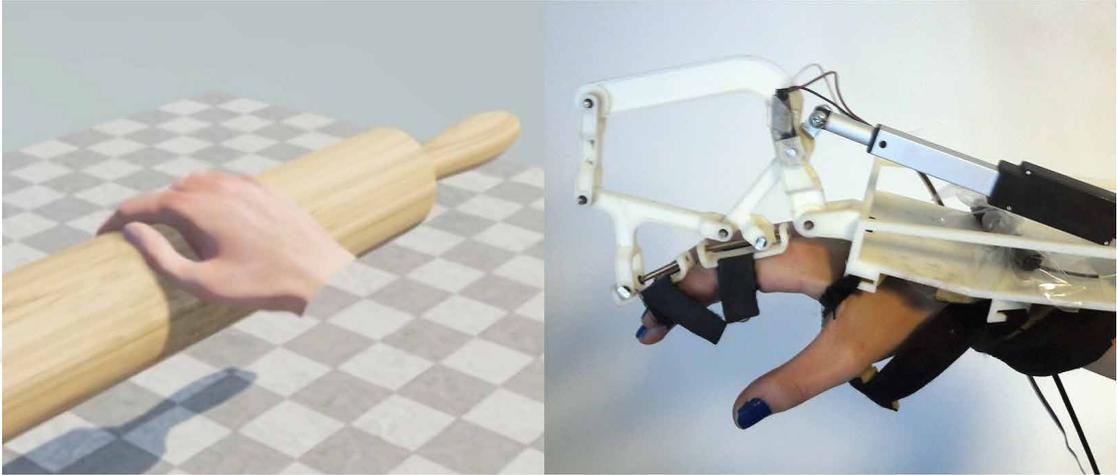


FIGURE 5.22: Contact with a soft hand being rendered through an underactuated exoskeleton.

The virtual interactions are rendered in the virtual environment between a soft finger model [168] and other objects, as shown in Figure 5.22. In particular, the motion of the palm is tracked through a Leap Motion device while the finger tracking is performed by the exoskeleton (see Section 4.3.1.3). This combined tracking sets the configuration  $\mathbf{q}$  of the device, which we transform into the proxy representation of the phalanges  $f(\mathbf{q})$  using the two joints mentioned above. The contact between the soft finger and the grasping objects are modelled by constraining the proxy phalanges. Finally, the proposed rendering algorithm is applied to compute the force command for the finger exoskeleton. The given force command is utilized as a reference for the force control algorithm discussed in Section 4.3.3.

Several finger trajectories were recorded as well as their associated rendering computations. For the experiments, we have used as impedance a normalized stiffness to factor out the average scale in  $\mathbf{J}_a$ , i.e.,  $\mathbf{Z}_q = 1$  and

$$\mathbf{Z}_x = \frac{1}{\text{avg}(\|\mathbf{J}_a\|)^2} \mathbf{I}. \quad (5.26)$$

In order to compare the proposed subspace rendering method to the standard and null-space methods, a force computation and impedance analysis have been carried out in a controlled setting. Given a recorded finger trajectory, the proxy is fixed at  $\mathbf{x} = 0$ , and

the output force for the linear actuator as well as the displayed impedance  $d\tau_a/d\mathbf{q}_a$  are computed. Such impedance is computed (a) following the theoretical formulations in Equation (5.18), Equation (5.21) and Equation (5.25) respectively for the three rendering methods, and (b) through finite differences of the applied force and the device motion between frames, i.e.,  $\Delta\tau_a/\Delta\mathbf{q}_a$ , as performed in the previous sections.

Figure 5.23 shows the results for a sample finger trajectory. Our subspace-proxy-based method is always passive in the experiment, according to the theoretical result but also in practice. The standard and null-space methods, on the other hand, are not always passive in practice. This contradicts the theoretical results, due to the missing  $\partial\mathbf{J}/\partial\mathbf{q}$  term.



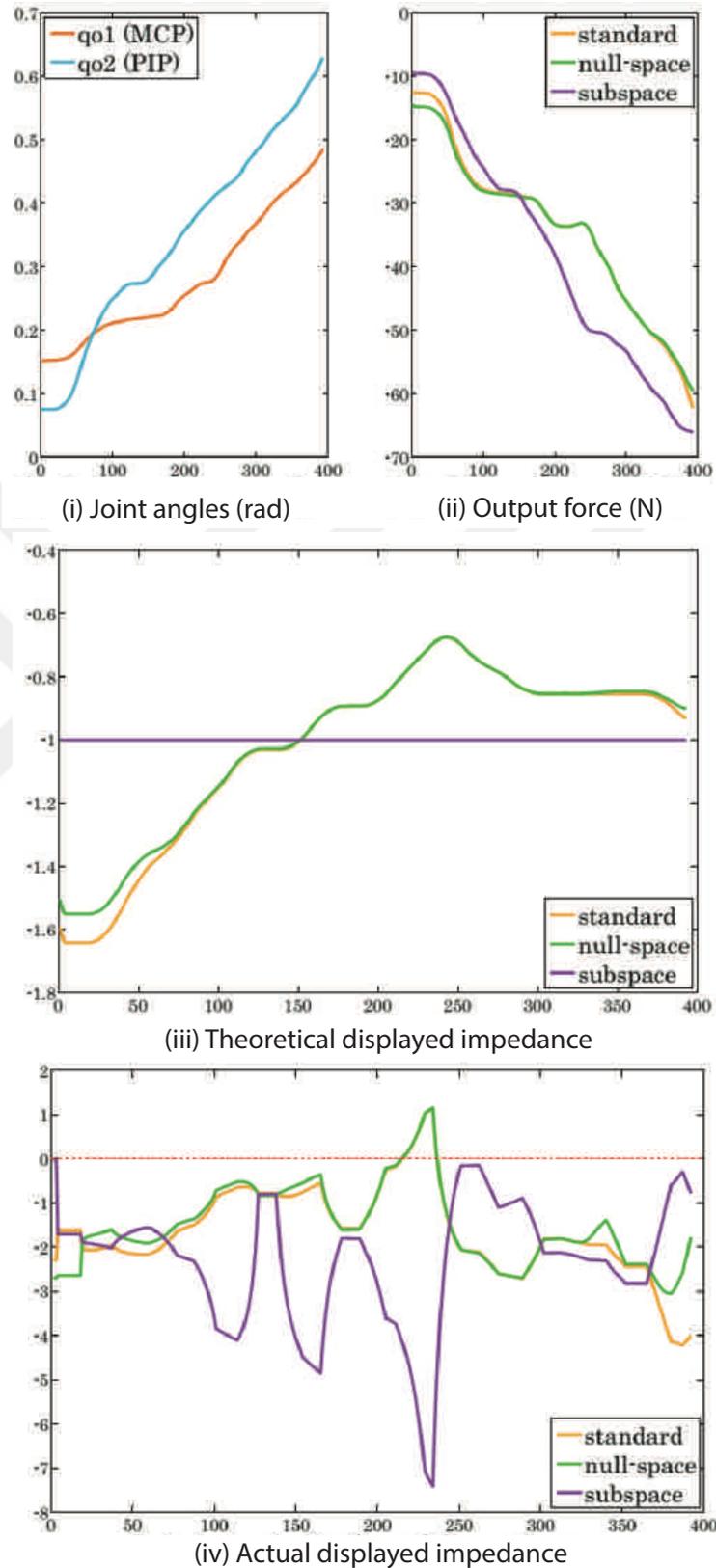


FIGURE 5.23: Performance comparison of the three rendering methods discussed in the paper (standard, optimized, and subspace). In the test, the proxy is kept still at a zero angle.

Such a novel rendering method opens up multiple avenues for further investigation. First, the proposed rendering algorithm linearizes the mapping from device to virtual workspace, which works well when the deviation between device state and subspace proxy is small. A full nonlinear solve would be more robust under large proxy deviations, but it requires efficient solution methods. Second, in the impedance analysis two important approximations have been made, namely that the proxy remains still and that the Jacobian of the mapping from device to virtual workspace is constant. A passivity controller could be needed to enforce passivity in all cases. And third and most important, the overall quality of haptic rendering can be optimized in a task-specific and device-specific manner by tuning the objective functions that guide the computation of the proxy and the subspace proxy.



## Chapter 6

# Conclusion

In this dissertation, a novel underactuated hand exoskeleton was proposed mainly to offer help for post-stroke patients with hand disabilities during physical rehabilitation exercises and daily activities with robotic assistance. Its use can also be extended for haptic applications.

This study began by stating the importance and complexity of the hand in terms of kinematics and crucial functionality during the activities of daily living (ADLs), as much as the importance of providing an efficient treatment process to overcome these physical injuries. The robotic devices can prove efficient, repetitive and accurate physical exercises, therefore the motor recovery and the output of rehabilitation therapy can be improved simply by creating robot assisted rehabilitation scenarios. The kinematics model of the hand was analyzed to understand how the assistance should be provided to the user and possible simplifications due to the focus of motion. A wide range of hand exoskeletons, which focus on recovering the hand functionality, were investigated in details to analyze their characteristics and weaknesses in order to create a generic hand exoskeleton. Furthermore, the devices were categorized based on (i) their active and passive mobility for the hand and for each finger, (ii) their number of connection points between the hand exoskeleton and each finger, (iii) their pose estimation technique throughout the operation, (iv) their actuation type, (v) their placement of each finger component and (vi) their control algorithm. Each category was studied to give kinematics decisions to meet the design requirements of the proposed hand exoskeleton.

The design criteria of the proposed hand exoskeleton were listed based on basic safety precautions for human interaction and specific requirements based on rehabilitation or

assistive applications. Using these requirements, the design properties for the proposed hand exoskeleton were listed as:

- assisting only for 2 *DoFs* finger rotation of each finger for flexion/extension only;
- reaching independent control of fingers, implementing single actuators for each finger component;
- achieving 2 *DoFs* flexion/extension through a single actuator;
- adjusting the grasping tasks for various object with different sizes and shapes based on the contact forces simply by embracing the underactuation concept;
- adjusting the operation for different hand sizes by completing the linkage based mechanism kinematics around user's fingers;
- placing the finger components and actuators on the dorsal side of the hand and
- constraining only perpendicular forces to be transmitted to finger phalanges.

The underactuation concept simplifies the mechanical design to control multiple finger joints using a single actuator and aligns the overall operation automatically to different sizes and shapes of the grasping object. Placing the mechanical finger component and its actuator on the dorsal side of the hand increased the portability of the device. The use of miniaturized linear actuators made possible to extend the finger components for the implementation of a complete hand exoskeleton with independent finger control. The chosen linear actuator were selected with internal gearbox to amplify output forces, which are sufficient for power grasping tasks, with minimum magnetic interference with the given placement. However, the overall backdriveability of the finger components were lost due to high backdrive force caused by such a mechanical gearbox. Embracing linkage based mechanism allowed the actuator forces to be distributed to finger phalanges in an efficient and practical manner, improving the wearability of the device considering the disabilities of stroke patients. As a novelty, the proposed hand exoskeleton was designed with passive linear sliders for connecting the mechanism and user's finger phalanges. Doing so, the tangential forces are turned into motion and perpendicular forces are transmitted to user in a more realistic manner.

Using the aforementioned design criteria, the first sketch of the proposed finger component mechanism was drawn. Further analyses were required to analyze the mechanism

behavior, the range of motion covered by the exoskeleton and the force transmission performance during operation. The performed analyses can be listed as:

- numerical inverse kinematics analysis to calculate the required actuator displacement for given finger pose;
- numerical forward kinematics analysis to estimate the finger pose during operation using the actuator displacement and an additional sensory measurements attached on one of the passive revolute joints;
- a numerical forward kinematics analysis to be used for calibration, where the lengths of proximal finger phalanges are estimated using the actuator displacement, additional potentiometer measurements and a passive slider displacement known thanks to a pre-defined finger pose;
- analytical forward kinematics analysis as an alternative to the numerical forward kinematics due to the calculational burden of numerical approach and
- differential kinematics analysis based on numerical kinematics approach to form a square matrix relating the measurement velocities to finger joint velocities.

Inverse kinematics and differential kinematics were analyzed in order to perform a link length optimization to find a set of link lengths that satisfies the physical limits and natural range of motion for finger joints and reaches the highest force transmission along finger phalanges for unitary actuator force. The physical constraints were defined as the desired actuator displacement, the displacements along the passive sliders at the connection with user's finger phalanges and the ratio between forces acting on finger phalanges. All these constraints were ensured to be satisfied for natural range of motion of finger joints. For a set of link lengths that satisfied all these physical constraints, the torque values along finger joints for unitary actuator force were calculated. The set of link lengths with highest force transmission to finger joints were chosen as the optimized set. Once the link lengths were optimized, the hand exoskeleton was manufactured using 3D printer thanks to its light weight and low cost. A control board was designed to control the exoskeleton simply by reading sensory measurements, computing kinematics analysis, running the controller and driving the motor drivers.

Integrating the mechanical device to electronic board, the feasibility of the proposed hand exoskeleton is shown with various control algorithms to define various physical rehabilitation scenarios. In particular:

- simple position control was implemented with improved performance to provide strict finger opening/closing for users during repetitive rehabilitation exercises;
- simple position control with improved performance was extended to assist real grasping tasks for objects with different sizes and shapes as the interaction forces are measured between the object and the user's finger phalanges;
- simple position control was performed with active trajectory adjustment using muscular activity measured from the healthy arm through EMG sensors for tele-operation or active rehabilitation exercises and
- simple force control algorithm was implemented based on force measurements through 1-DoF strain gauges to achieve backdriveability over control or haptic rendering tasks.

Achieving active backdriveability allowed the proposed hand exoskeleton to be used for haptic applications to provide force feedback to users regarding their interactions in the virtual environment. The lack of controllability of the proposed hand exoskeleton and the lack of physical interaction forces acting on finger phalanges created complications for the device to be controlled virtually. Even if the lack of controllability cannot be overcome, such complications were proposed to be estimated simply by:

- degrading the device's mobility to  $1 - DoF$  by constraining one finger joint at a time to simplify the grasping task;
- optimizing desired torque values, which are calculated based on stiffness rendering algorithm, to satisfy the passivity of additional sensory measurement in the force space, which is the underactuation constraint and
- optimizing the desired finger pose to estimate the reachable pose for finger joints by modelling finger joints as spring and to satisfy the passivity of additional sensory measurement in the force space.

In particular, two different rendering strategies were proposed to implement underactuated devices for haptic tasks: (i) based on the force space and (ii) based on the configurational space. The feasibility of both methods were shown through real time experiments with the proposed hand exoskeleton. Firstly, the behavior of the rendering strategies were presented based on finger pose and corresponding actuator forces.

Then, the comparison between actual transmitted forces for finger joints and desired finger joints were performed using additional FSR sensors attached between the device and user's finger. These two methods were compared to each other by computing the sensed impedance acting on finger joints during operations using the same task. Even though the feasibility of both strategies were shown in this work, an overall comparison to emphasize their advantages and disadvantages was left as a future work.

Once the optimization based rendering strategies were shown effective for user's finger with 2 *DoFs* stiff finger model, they were extended and generalized for proxy-based haptic rendering using Unreal simulation engine. The generalization of the optimization problem led us to run haptic rendering tasks while grasping soft objects. In these tasks, the user's hand was not modeled as simple 2 *DoFs* chain, but all virtual points consisting user's avatar was considered as an independent *DoF*. The experiment results showed the positive effect of the optimization method compared to the standard one.

The experiments and the analyses that were presented in this thesis show sufficient proof for the feasibility of the proposed hand exoskeleton to be used for physical rehabilitation exercises with post-stroke patients. In particular, obtaining active and passive tasks with improved control techniques in terms of perception is promising for the motor learning of patients with disabilities during grasping tasks. Despite of all that was done so far, there are still some works to be done in the future. These future works can be listed as:

- improving the ergonomic and comfort of users, especially while wearing the device as preparation period,
- integrating the hand exoskeleton as a master part for teleoperation tasks to control a slave robotic hand,
- integrating the hand exoskeleton with a wrist exoskeleton and an arm exoskeleton, which were designed in the same environment, to run some clinical tests with stroke patients,
- performing a detailed user study to study the perceptual feedback of the haptic rendering simulation integrated with the hand exoskeleton, and
- comparing both proposed stiffness rendering strategies to each other in a more detailed manner.

# Bibliography

- [1] B. Buchholz and T. J. Armstrong. A kinematic model of the human hand to evaluate its prehensile capabilities. *Journal of Biomechanics*, 25:149–162, 1992.
- [2] *Official Website of the MYO*. <http://www.myo.com>.
- [3] A. Buryanov and V. Kotiuk. Proportions of hand segments. *International Journal of Morphology*, 28:755 – 758, 2010.
- [4] J. C. Becker and N. V. Thakor. A study of the range of motion of human fingers with application to anthropomorphic designs. *IEEE Transactions on Biomedical Engineering*, 35:110 – 118, 1988.
- [5] Neurological injury. [http://www.oshmanlaw.com/personal\\_injury/neurological-injury-lawyer.html](http://www.oshmanlaw.com/personal_injury/neurological-injury-lawyer.html).
- [6] Global burden of disease. [http://www.who.int/healthinfo/global\\_burden\\_disease/GBD\\_report\\_2004update\\_full.pdf](http://www.who.int/healthinfo/global_burden_disease/GBD_report_2004update_full.pdf).
- [7] C. C. Edward, C. Jeffrey, S. Sergio, and R. Robert. *Stroke - the American Psychiatric Press Textbook of Geriatric Neuropsychiatry (Second ed.)*. Washington DC: American Psychiatric Press, 2000.
- [8] A. Z. C. Daud, M. K. Yau, F. Barnett, J. Judd, R. E. Jones, and R. F. M. Nawawi. Integration of occupation based intervention in hand injury rehabilitation: A randomized controlled trial. *Journal of Hand Therapy*, 29(1):30 – 40, 2016.
- [9] C. Bütefisch, H. Hummelsheim, P. Denzler, and K.-H. Mauritz. Repetitive training of isolated movements improves the outcome of motor rehabilitation of the centrally paretic hand. *Journal of the Neurological Sciences*, 130(1):59 – 68, 1995.

- [10] G. Kwakkel, R. C. Wagenaar, J. W. R. Twisk, G. J. Lankhorst, and J. C. Koetsier. Intensity of leg and arm training after primary middle-cerebral-artery stroke: A randomised trial. *The Lancet*, 354(9174):191 – 196, 1999.
- [11] A. Sunderland, D. J. Tinson, E. L. Bradley, D. Fletcher, R. Langton Hewer, and D. T. Wade. Enhanced physical therapy improves recovery of arm function after stroke. A randomised controlled trial. *Journal of Neurology, Neurosurgery and Psychiatry*, 55(7):530–535, 1992.
- [12] Nestor A Bayona, Jamie Bitensky, Katherine Salter, and Robert Teasell. The role of task-specific training in rehabilitation therapies. *Topics in stroke rehabilitation*, 12(3):58–65, 2005.
- [13] Y. Levanon. The advantages and disadvantages of using high technology in hand rehabilitation. *Journal of Hand Therapy*, 26(2):179 – 183, 2013.
- [14] G. Kwakkel, B. J. Kollen, and H. I. Krebs. Effects of robot-assisted therapy on upper limb recovery after stroke: A systematic review. *Neurorehab. and Neural Repair*, 22(2):111–121, 2008.
- [15] J. Mehrholz, T. Platz, J. Kugler, and M. Pohl. Electromechanical and robot-assisted arm training for improving arm function and activities of daily living after stroke. *Stroke*, 40, 2009.
- [16] Gerdienke B. Prange, Michiel J. A. Jannink, Catharina G. M. Groothuis-Oudshoorn, Hermie Hermens, and Maarten J. IJzerman. Systematic review of the effect of robot-aided therapy on recovery of the hemiparetic arm after stroke. *Journal of rehabilitation research and development*, 43 2:171–184, 2006.
- [17] K. Nykanen. The effectiveness of robot-aided upper limb therapy in stroke rehabilitation: A systematic review of randomized controlled studies. Master’s thesis, University of Jyväskylä, Institute of Health Sciences, Physiotherapy, 2010.
- [18] H. Qiao, M. Wang, J. Su, S. Jia, and R. Li. The concept of attractive region in environment and its application in high-precision tasks with low-precision systems. *IEEE/ASME Transactions on Mechatronics*, 20(5):2311–2327, 2015.
- [19] Y. Hashimoto, D. Yatou, T. Yamada, and H. Yamamoto. Fundamental study on robotic assembly of modular fixture parts by a low cost 4-finger 12 DoF hand. In *IEEE/SICE International Symposium on System Integration (SII)*, pages 930–935, 2015.

- [20] Y. Cheung, Y. T. Huang, and J. J. J. Lien. Visual guided adaptive robotic interceptions with occluded target motion estimations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6067–6072, 2015.
- [21] A. Altamirano-Altamirano, A. Vera-Hernández, L. Leija-Salas, and D. Wolf. Feedback system based on current measurements to improve a hand prosthesis. In *Pan American Health Care Exchanges (PAHCE)*, pages 1–4, 2014.
- [22] G. Matrone, C. Cipriani, M. C. Carrozza, and G. Magenes. Two-channel real-time EMG control of a dexterous hand prosthesis. In *International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 554–557, 2011.
- [23] D. A. Bennett, S. A. Dalley, D. Truex, and M. Goldfarb. A multigrasp hand prosthesis for providing precision and conformal grasps. *IEEE/ASME Transactions on Mechatronics*, 20(4):1697–1704, 2015.
- [24] H. Mnyusiwalla, P. Vulliez, J. P. Gazeau, and S. Zeghloul. A new dexterous hand based on bio-inspired finger design for inside-hand manipulation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(6):809–817, 2016.
- [25] N. Fukaya, T. Asfour, R. Dillmann, and S. Toyama. Development of a five-finger dexterous hand without feedback control: The tuat/karlsruhe humanoid hand. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4533–4540, 2013.
- [26] T. Mouri, H. Kawasaki, and S. Ueki. Teleoperated humanoid hand robot using force feedback. In *IEEE/SICE International Symposium on System Integration (SII)*, pages 942–947, 2015.
- [27] Y. Kamogawa, A. Takanishi, Y. Kuwahara, and H. o. Lim. Development of tendon-driven robotic hand for biped humanoid robot. In *International Symposium on Soft Computing and Intelligent Systems (SCIS), Joint International Conference on and Advanced Intelligent Systems (ISIS)*, pages 914–917, 2014.
- [28] M. Sarac Stroppa, M. Solazzi, E. Sotgiu, M. Bergamasco, A. Frisoli, and M. Gabardi. Exoskeleton device for the hand.
- [29] M. Sarac, M. Solazzi, E. Sotgiu, M. Bergamasco, and A. Frisoli. Design and kinematic optimization of a novel underactuated robotic hand exoskeleton. *Meccanica*, pages 1–13, 2016.

- [30] M. Sarac, M. Solazzi, D. Leonardis, E. Sotgiu, M. Bergamasco, and A. Frisoli. *Design of an Underactuated Hand Exoskeleton with Joint Estimation*, pages 97–105. Springer International Publishing, 2017.
- [31] D. Lobo, M. Sarac, M. Verschoor, M. Solazzi, A. Frisoli, and M. A. Otaduy. Proxy-based haptic rendering for underactuated haptic devices. In *IEEE World Haptics Conference (WHC)*, pages 1–6, 2017.
- [32] I. Sarakoglou, S. Kousidou, N. G. Tsagarakis, and D. G. Caldwell. Exoskeleton-based exercisers for the disabilities of the upper arm and hand. *Rehabilitation Robotics*, page 648, 2007.
- [33] M. M. Foumashi, M. Troncossi, and V. P. Castelli. State-of-the-art of hand exoskeleton systems. 2011.
- [34] P. Heo, G. M. Gu, S-J Lee, K. Rhee, and J. Kim. Current hand exoskeleton technologies for rehabilitation and assistive engineering. *International Journal of Precision Engineering and Manufacturing*, 13(5):807–824, 2012.
- [35] J. Wang, J. Li, Y. Zhang, and S. Wang. Design of an exoskeleton for index finger rehabilitation. In *Annual International Conference in Medicine and Biology Society (EMBC)*, pages 5957–5960, 2009.
- [36] K. Y. Tong, S. K. Ho, P. M. K. Pang, X. L. Hu, W. K. Tam, K. L. Fung, X. J. Wei, P. N. Chen, and M. Chen. An intention driven hand functions task training robotic system. In *IEEE International Conference of the Engineering in Medicine and Biology Society (EMBC)*, pages 3406–3409, 2010.
- [37] B. W. Gasser and M. Goldfarb. Design and performance characterization of a hand orthosis prototype to aid activities of daily living in a post-stroke population. In *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3877–3880, 2015.
- [38] I. Choi, E. W. Hawkes, D. L. Christensen, C. J. Ploch, and S. Follmer. Wolverine: A wearable haptic interface for grasping in virtual reality. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 986–993, 2016.
- [39] Anna-Karin Welmer, Lotta Widén Holmqvist, and Disa K Sommerfeld. Limited fine hand use after stroke and its association with other disabilities. *Journal of Rehabilitation Medicine*, 40(8):603–608, 2008.

- [40] X.L. Hu, K.Y. Tong, X.J. Wei, W. Rong, E.A. Susanto, and S.K. Ho. The effects of post-stroke upper-limb training with an electromyography (EMG)-driven hand robot. *Journal of Electromyography and Kinesiology*, 23(5):1065 – 1074, 2013.
- [41] E. A. Susanto, R. K. Y. Tong, C. Ockenfeld, and N. S. K. Ho. Efficacy of robot-assisted fingers training in chronic stroke survivors: A pilot randomized-controlled trial. *Journal of NeuroEngineering and Rehabilitation*, 12(1):1–9, 2015.
- [42] J. Iqbal, N.G. Tsagarakis, and D.G. Caldwell. A multi-DoF robotic exoskeleton interface for hand motion assistance. In *IEEE International Conference of the Engineering in Medicine and Biology Society*, pages 1575–1578, 2011.
- [43] Jamshed Iqbal and Khelifa Baizid. Stroke rehabilitation using exoskeleton-based robotic exercisers: Mini review. *Biomedical Research*, 2015.
- [44] P. Ben-Tzvi and Z. Ma. Sensing and force-feedback exoskeleton (SAFE) robotic glove. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 23(6):992–1002, 2015.
- [45] Z. Ma, P. Ben-Tzvi, and J. Danoff. Hand rehabilitation learning system with an exoskeleton robotic glove. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(12):1323–1332, 2016.
- [46] Y. Fu, P. Wang, S. Wang, H. Liu, and F. Zhang. Design and development of a portable exoskeleton based CPM machine for rehabilitation of hand injuries. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1476–1481, 2007.
- [47] A. Chiri, F. Giovacchini, N. Vitiello, E. Cattin, S. Roccella, F. Vecchi, and M. C. Carrozza. HANDEXOS: Towards an exoskeleton device for the rehabilitation of the hand. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1106–1111, 2009.
- [48] A. Chiri, N. Vitiello, F. Giovacchini, S. Roccella, F. Vecchi, and M. C. Carrozza. Mechatronic design and characterization of the index finger module of a hand exoskeleton for post-stroke rehabilitation. *IEEE/ASME Transactions on Mechatronics*, 17(5):884–894, 2012.
- [49] Md A. Rahman and A. Al-Jumaily. Design and development of a hand exoskeleton for rehabilitation following stroke. *International Symposium on Robotics and Intelligent Sensors (IRIS)*, 41:1028 – 1034, 2012.

- [50] I. Jo and J. Bae. A force-controllable compact actuator module for a wearable hand exoskeleton. *IFAC Proceedings Volumes*, 47(3):4453 – 4458, 2014.
- [51] I. Jo and J. Bae. Design and control of a wearable hand exoskeleton with force-controllable and compact actuator modules. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5596–5601, 2015.
- [52] L. Connelly, Y. Jia, M. L. Toro, M. E. Stoykov, R. V. Kenyon, and D. G. Kamper. A pneumatic glove and immersive virtual reality environment for hand rehabilitative training after stroke. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(5):551–559, 2010.
- [53] H. Fang, Z. Xie, and H. Liu. An exoskeleton master hand for controlling DLR/HIT hand. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3703–3708, 2009.
- [54] M. A. Delph, S. A. Fischer, P. W. Gauthier, C. H. M. Luna, E. A. Clancy, and G. S. Fischer. A soft robotic exomusculature glove with integrated sEMG sensing for hand rehabilitation. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 1–7, 2013.
- [55] P. Polygerinos, Z. Wang, K. C. Galloway, R. J. Wood, and C. J. Walsh. Soft robotic glove for combined assistance and at-home rehabilitation. *Robotics and Autonomous Systems (RAS) Special Issue on Wearable Robotics*, 73:135–143, 2015.
- [56] L. Cui, A. Phan, and G. Allison. Design and fabrication of a three dimensional printable non-assembly articulated hand exoskeleton for rehabilitation. In *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4627–4630, 2015.
- [57] S. Kim, J. Lee, W. Park, and J. Bae. Quantitative evaluation of hand functions using a wearable hand exoskeleton system. In *International Conference on Rehabilitation Robotics (ICORR)*, pages 1488–1493, 2017.
- [58] M. Decker and Y. Kim. A hand exoskeleton device for robot assisted sensory-motor training after stroke. In *IEEE World Haptics Conference (WHC)*, pages 436–441, 2017.
- [59] I. Jo, J. Lee, Y. Park, and J. Bae. Design of a wearable hand exoskeleton for exercising flexion/extension of the fingers. In *International Conference on Rehabilitation Robotics (ICORR)*, pages 1615–1620, 2017.

- [60] Hong Kai. Yap, Jeong Hoon. Lim, Fatima. Nasrallah, James C. H. Goh, and Raye C. H. Yeow. A soft exoskeleton for hand assistive and rehabilitation application using pneumatic actuators with variable stiffness. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4967 – 4972, 2015.
- [61] H. K. Yap, P. M. Khin, T. H. Koh, Y. Sun, X. Liang, J. H. Lim, and C. H. Yeow. A fully fabric-based bidirectional soft robotic glove for assistance and rehabilitation of hand impaired patients. *IEEE Robotics and Automation Letters*, 2(3):1383–1390, 2017.
- [62] X. Lu, Z. Yang, Y. Chen, and J. Wang. Structure design of a wearable device for hand rehabilitation. In *9th International Symposium on Computational Intelligence and Design (ISCID)*, volume 1, pages 93–96, 2016.
- [63] Mine Sarac, Massimiliano Solazzi, Miguel A Otaduy, and Antonio Frisoli. Rendering strategies for underactuated hand exoskeletons. *IEEE Robotics and Automation Letters*, 3(3):2087–2092, 2018.
- [64] M. Gabardi, M. Solazzi, D. Leonardis, and A. Frisoli. Design and evaluation of a novel 5-DoF underactuated thumb-exoskeleton. *IEEE Robotics and Automation Letters*, 3(3):2322–2329, 2018.
- [65] Y. Hasegawa, Y. Mikami, K. Watanabe, and Y. Sankai. Five-fingered assistive hand with mechanical compliance of human finger. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 718–724, 2008.
- [66] Y. Hasegawa, J. Tokita, K. Kamibayashi, and Y. Sankai. Evaluation of fingertip force accuracy in different support conditions of exoskeleton. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 680–685, 2011.
- [67] T. M. W. Burton, R. Vaidyanathan, S. C. Burgess, A. J. Turton, and C. Melhuish. Development of a parametric kinematic model of the human hand and a novel robotic exoskeleton. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 1–7, 2011.
- [68] Peter Ferguson, Brando Dimapasoc, Yang Shen, and Jacob Rosen. Design of a hand exoskeleton for use with upper limb exoskeletons. In *International Symposium on Wearable Robotics, WeRob2018*, 2018.
- [69] S. W. Lee, K. A. Landers, and H-S Park. Biomimetic hand extendon device (BiomHED) for functional hand rehabilitation in stroke. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 1–4, 2013.

- [70] T. C. Lee. Rendering 6-DoF object-to-object interaction with 3-DoF haptic interfaces. Master's thesis, TU Delft, the Netherlands, 2014.
- [71] M. Troncossi, M. M. Fomashi, M. Carricato, and V. P. Castelli. Feasibility study of a hand exoskeleton for rehabilitation of post-stroke patients. In *Proceedings of the ASME, Biennial Conference on Engineering Systems Design and Analysis (ESDA)*, pages 1–10, 2012.
- [72] D. Leonardis, M. Barsotti, C. Loconsole, M. Solazzi, M. Troncossi, C. Mazzotti, V. Parenti Castelli, C. Procopio, G. Lamola, C. Chisari, et al. An EMG-controlled robotic hand exoskeleton for bilateral rehabilitation. In *IEEE*, 2015.
- [73] M. Mulas, M. Folgheraiter, and G. Gini. An EMG-controlled exoskeleton for hand rehabilitation. In *International Conference on Rehabilitation Robotics (ICORR)*, pages 371–374, 2005.
- [74] E. B. Brokaw, I. Black, R. J. Holley, and P. S. Lum. Hand spring operated movement enhancer (HandSOME): A portable, passive hand exoskeleton for stroke rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(4):391–399, 2011.
- [75] K. Fujimoto, F. Kobayashi, H. Nakamoto, and F. Kojima. Development of haptic device for five-fingered robot hand teleoperation. In *Proceedings of the IEEE/SICE International Symposium on System Integration*, pages 820–825, 2013.
- [76] H. Li and L. Cheng. Preliminary study on the design and control of a pneumatically-actuated hand rehabilitation device. In *Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 860–865, 2017.
- [77] M. Bouzit, G. Burdea, G. Popescu, and R. Boian. The rutgers master II- New design force-feedback glove. *IEEE/ASME Transactions on Mechatronics*, 7(2):256–263, 2002.
- [78] D. Popov, I. Gaponov, and J. H. Ryu. Portable exoskeleton glove with soft structure for hand assistance in activities of daily living. *IEEE/ASME Transactions on Mechatronics*, 22(2):865–875, 2017.
- [79] B. Allotta, R. Conti, L. Governi, E. Meli, A. Ridolfi, and Y. Volpe. Development and experimental testing of a portable hand exoskeleton. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5339–5344, 2015.

- [80] J. Wu, J. Huang, Y. Wang, and K. Xing. A wearable rehabilitation robotic hand driven by PM-TS actuators. In *Proceedings on Intelligent Robotics and Applications, Part II: Third International Conference (ICIRA)*. Springer Berlin Heidelberg, 2010.
- [81] P. Weiss, L. Heyer, T. F. Münte, M. Heldmann, A. Schweikard, and E. Maehle. Towards a parameterizable exoskeleton for training of hand function after stroke. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 1–6, 2013.
- [82] W. Wei, S. Guo, F. Zhang, J. Guo, Y. Ji, and Y. Wang. A novel upper limb rehabilitation system with hand exoskeleton mechanism. In *IEEE International Conference on Mechatronics and Automation*, pages 285–290, 2013.
- [83] J. Arata, K. Ohmoto, R. Gassert, O. Lamercy, H. Fujimoto, and I. Wada. A new hand exoskeleton device for rehabilitation using a three-layered sliding spring mechanism. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3902–3907, 2013.
- [84] C. J. Nycz, T. Bützer, O. Lamercy, J. Arata, G. S. Fischer, and R. Gassert. Design and characterization of a lightweight and fully portable remote actuation system for use with a hand exoskeleton. *IEEE Robotics and Automation Letters*, 1(2):976–983, 2016.
- [85] B. W. Gasser, D. A. Bennett, C. M. Durrough, and M. Goldfarb. Design and preliminary assessment of vanderbilt hand exoskeleton. In *2017 International Conference on Rehabilitation Robotics (ICORR)*, pages 1537–1542, 2017.
- [86] A. Lince, N. Celadon, A. Battezzato, A. Favetto, S. Appendino, P. Ariano, and M. Paleari. Design and testing of an under-actuated surface EMG-driven hand exoskeleton. In *2017 International Conference on Rehabilitation Robotics (ICORR)*, pages 670–675, 2017.
- [87] Q-X Wei, C-J Yang, Q. Bi, and W. Yang. *A Grasp Strategy with Flexible Contacting for Multi-fingered Hand Rehabilitation Exoskeleton*, pages 225–237. Springer Singapore, 2017.
- [88] F. Kobayashi, G. Ikai, W. Fukui, H. Nakamoto, and F. Kojima. Multipoint haptic device for robot hand teleoperation. In *International Symposium on Micro-Nano Mechatronics and Human Science (MHS)*, pages 304–309, 2012.

- [89] D. Ryu, K-W Moon, H. Nam, Y. Lee, C. Chun, S. Kang, and J-B Song. Micro hydraulic system using slim artificial muscles for a wearable haptic glove. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3028–3033, 2008.
- [90] I. Sarakoglou, A. Brygo, D. Mazzanti, N. G. Hernandez, D. G. Caldwell, and N. G. Tsagarakis. HEXOTRAC : A highly underactuated hand exoskeleton for finger tracking and force feedback. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1033–1040, 2016.
- [91] H. In, D. Lee, and K. J. Cho. Investigation of friction characteristics of a tendon driven wearable robotic hand. In *International Conference on Control Automation and Systems (ICCAS)*, pages 568–573, 2010.
- [92] H. In, B. B. Kang, M. Sin, and K. J. Cho. Exo-Glove: A wearable robot for the hand with a soft tendon routing system. *IEEE Robotics Automation Magazine*, 22(1):97–105, 2015.
- [93] S. Guo, W. Zhang, J. Guo, J. Gao, and Y. Hu. Design and kinematic simulation of a novel exoskeleton rehabilitation hand robot. In *IEEE International Conference on Mechatronics and Automation*, pages 1125–1130, 2016.
- [94] M. Cempini, S. M. M. De Rossi, T. Lenzi, M. Cortese, F. Giovacchini, N. Vitiello, and M. C. Carrozza. Kinematics and design of a portable and wearable exoskeleton for hand rehabilitation. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 1–6, 2013.
- [95] M. Cortese, M. Cempini, P. R. de Almeida Ribeiro, S. R. Soekadar, M. C. Carrozza, and N. Vitiello. A mechatronic system for robot mediated hand telerehabilitation. *IEEE/ASME Transactions on Mechatronics*, 20(4):1753–1764, 2015.
- [96] H. Taheri, J. B. Rowe, D. Gardner, V. Chan, K. Gray, C. Bower, D. J. Reinkensmeyer, and E. T. Wolbrecht. Design and preliminary evaluation of the FINGER rehabilitation robot: Controlling challenge and quantifying finger individuation during musical computer game play. *Journal of NeuroEngineering and Rehabilitation*, 11(10):1–17, 2014.
- [97] F. Gosselin, T. Jouan, J. Brisset, and C. Andriot. Design of a wearable haptic interface for precise finger interactions in large virtual environments. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual*

- Environment and Teleoperator Systems. World Haptics Conference*, pages 202–207, 2005.
- [98] M. Fontana, A. Dettori, F. Salsedo, and M. Bergamasco. Mechanical design of a novel hand exoskeleton for accurate force displaying. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1704–1709, 2009.
- [99] J. Li, R. Zheng, Y. Zhang, and J. Yao. iHandRehab: An interactive hand exoskeleton for active and passive rehabilitation. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 1 – 6, 2011.
- [100] F. Nori A. E. Fiorilla, N. G. Tsagarakis and G. Sandini. Design of a 2-finger hand exoskeleton for finger stiffness measurements. *Applied Bionics and Biomechanics*, 6:217 – 228, 2009.
- [101] H. Yamaura, K. Matsushita, R. Kato, and H. Yokoi. Development of hand rehabilitation system for paralysis patient - universal design using wire-driven mechanism. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 7122–7125, 2009.
- [102] C. L. Jones, F. Wang, C. Osswald, X. Kang, N. Sarkar, and D. G. Kamper. Control and kinematic performance analysis of an actuated finger exoskeleton for hand rehabilitation following stroke. In *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 282–287. IEEE, 2010.
- [103] C. L. Jones, F. Wang, R. Morrison, N. Sarkar, and D. G. Kamper. Design and development of the cable actuated finger exoskeleton for hand rehabilitation following stroke. *IEEE/ASME Transactions on Mechatronics*, 19(1):131–140, 2014.
- [104] A. Polotto, F. Modulo, F. Flumian, Z. G. Xiao, P. Boscariol, and C. Menon. Index finger rehabilitation/assistive device. In *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 1518–1523, 2012.
- [105] Z. J. Tang, S. Sugano, and H. Iwata. A novel, MRI compatible hand exoskeleton for finger rehabilitation. In *IEEE/SICE International Symposium on System Integration (SII)*, pages 118–123, 2011.
- [106] Z. Tang, S. Sugano, and H. Iwata. A finger exoskeleton for rehabilitation and brain image study. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 1–6, 2013.

- [107] M. DiCicco, L. Lucas, and Y. Matsuoka. Comparison of control strategies for an EMG controlled orthotic exoskeleton for the hand. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1622–1627, 2004.
- [108] A. Wege, K. Kondak, and G. Hommel. *Development and Control of a Hand Exoskeleton for Rehabilitation*, pages 149–157. Springer Netherlands, 2006.
- [109] A. Wege, K. Kondak, and G. Hommel. Force control strategy for a hand exoskeleton based on sliding mode position control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4615–4620, 2006.
- [110] A. Wege and A. Zimmermann. Electromyography sensor based control for a hand exoskeleton. In *IEEE International Conference on Robotics and Biomimetics (RO-BIO)*, pages 1470–1475, 2007.
- [111] Z. Sun, X. Miao, and X. Li. Design of a bidirectional force feedback dataglove based on pneumatic artificial muscles. In *International Conference on Mechatronics and Automation (ICMA)*, pages 1767–1771, 2009.
- [112] P. Agarwal, P. H. Kuo, R. R. Neptune, and A. D. Deshpande. A novel framework for virtual prototyping of rehabilitation exoskeletons. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 1–6, 2013.
- [113] P. Agarwal and A. D. Deshpande. Impedance and force-field control of the index finger module of a hand exoskeleton for rehabilitation. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 85–90, 2015.
- [114] I. H. Ertas, E. Hocaoglu, D. E. Barkana, and V. Patoglu. Finger exoskeleton for treatment of tendon injuries. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 194–201, 2009.
- [115] I. H. Ertas, E. Hocaoglu, and V. Patoglu. AssistOn-Finger: An under-actuated finger exoskeleton for robot-assisted tendon therapy. *Robotica*, 2014.
- [116] P. M. Aubin, H. Sallum, C. Walsh, L. Stirling, and A. Correia. A pediatric robotic thumb exoskeleton for at-home rehabilitation: The isolated orthosis for thumb actuation (IOTA). In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 1–6, 2013.
- [117] J. Lee and J. Bae. Design of a hand exoskeleton for biomechanical analysis of the stroke hand. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 484–489, 2015.

- [118] P. Maeder-York, T. Clites, E. Boggs, R. Neff, P. Polygerinos, D. Holland, L. Stirling, K. C. Galloway, C. Wee, and C. J. Walsh. Biologically inspired soft robot for thumb rehabilitation. In *ASME Design of Medical Devices Conference*, Minneapolis, MN, 8-10 Apr. 2014.
- [119] M. Cempini, M. Cortese, and N. Vitiello. A powered finger - thumb wearable hand exoskeleton with self-aligning joint axes. *IEEE/ASME Transactions on Mechatronics*, 20(2):705–716, 2015.
- [120] Pyung Hun Chang, Seung-Hee Lee, Kwang-Min Koo, Seung-Hyun Lee and Sang Hyun Jin, Sang Seok Yeo, Jeong Pyo Seo, and Sung Ho Jang. The cortical activation pattern by a rehabilitation robotic hand : A functional NIRS study. *Frontiers in Human Neuroscience*, 8(49):1 – 25, 2014.
- [121] T. Laliberte, L. Birglen, and C. Gosselin. Underactuation in robotic grasping hands. *Machine Intelligence & Robotic Control*, 4(3):1–11, 2002.
- [122] N. Jarrasse and G. Morel. Connecting a human limb to an exoskeleton. *IEEE Transactions on Robotics*, 28(3):697–709, 2012.
- [123] R. Bortoletto, A. N. Mello, and D. Piovesan. A springs actuated finger exoskeleton: From mechanical design to spring variables evaluation. In *International Conference on Rehabilitation Robotics (ICORR)*, pages 1319–1325, 2017.
- [124] P.K. Kumar and D.C. Lagoudas. Introduction to shape memory alloys. In *Shape Memory Alloys: Modeling and Engineering Applications*, pages 1–51, 2008.
- [125] U. Jeong, H-K In, and K-J Cho. Implementation of various control algorithms for hand rehabilitation exercise using wearable robotic hand. *Intelligent Service Robotics*, 6(4):181–189, 2013.
- [126] M. Rakic. Multifingered robot hand with selfadaptability. *Robotics and Computer-Integrated Manufacturing*, 5(2/3):269–276, 1989.
- [127] S. Hirose and Y. Umetani. The development of soft gripper for the versatile robot hand. *Mechanism and machine theory*, 13(3):351–359, 1978.
- [128] M. Chiang, C. Chang, M. Chen, Y. Ho, T. Laliberte, and C. M. Gosselin. Simulation and design of underactuated mechanical hands. *Mechanism and Machine Theory*, 33(1):39–57, 1998.

- [129] L. Birglen and C. M. Gosselin. Optimal design of 2-phalanx underactuated fingers. In *Proceedings of 2004 International Conference on Intelligent Manipulation and Grasping*. Citeseer.
- [130] A. Mouida and N. Alaa. Sensitivity analysis of TSEB model by One-Factor-At-A-Time in irrigated olive orchard. *IJCSI International Journal of Computer Science Issues*, 8(1):369–378, 2011.
- [131] K. Salisbury, F. Conti, and F. Barbagli. Haptic rendering: introductory concepts. *IEEE Computer Graphics and Applications*, 24(2):24–32, 2004.
- [132] S. S. Snibbe, K. E. MacLean, R. Shaw, J. Roderick, W. L. Verplank, and M. Scheeff. Haptic techniques for media control. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, volume 3 of *UIST '01*, pages 199–208, 2001.
- [133] N. Beni, M. Grottoli, F. Ferrise, and M. Bordegoni. Rapid prototyping of low cost 1 DoF haptic interfaces. In *IEEE Haptics Symposium (HAPTICS)*, pages 479–483, 2014.
- [134] A. M. Okamura, R. J. Webster, J. T. Nolin, K. W. Johnson, and H. Jafry. The haptic scissors: Cutting in virtual environments. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 828–833, 2003.
- [135] F. Barbagli and K. Salisbury. The effect of sensor/actuator asymmetries in haptic interfaces. In *Proceedings on Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS.*, pages 140–147, 2003.
- [136] K. Minamizawa, H. Kajimoto, N. Kawakami, and S. Tachi. A wearable haptic display to present the gravity sensation - preliminary observations and device design. In *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*, pages 133–138, 2007.
- [137] K. Minamizawa, S. Fukamachi, N. Kawakami, and S. Tachi. Interactive representation of virtual object in hand-held box by finger-worn haptic display. In *2008 Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 367–368, 2008.
- [138] M. Solazzi, W. R. Provancher, A. Frisoli, and M. Bergamasco. Design of a SMA actuated 2-DoF tactile device for displaying tangential skin displacement. In *2011 IEEE World Haptics Conference*, pages 31–36, 2011.

- [139] A. Lecuyer, J. M. Burkhardt, J. Le Biller, and M. Congedo. "a4": a technique to improve perception of contacts with under-actuated haptic devices in virtual reality. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*, pages 316–322, 2005.
- [140] C. Ramstein and V. Hayward. The pantograph: A large workspace haptic device for multimodal human computer interaction. In *Conference companion on Human factors in computing systems (CHI)*, pages 57–58. ACM/SIGCHI Companion, 1994.
- [141] M. Solazzi, A. Frisoli, and M. Bergamasco. Design of a cutaneous fingertip display for improving haptic exploration of virtual objects. In *International Symposium in Robot and Human Interactive Communication*, pages 1–6, 2010.
- [142] F. Chinello, M. Malvezzi, C. Pacchierotti, and D. Prattichizzo. A three DoFs wearable tactile display for exploration and manipulation of virtual objects. In *IEEE Haptics Symposium (HAPTICS)*, pages 71–76, 2012.
- [143] H. Benko, C. Holz, M. Sinclair, and E. Ofek. Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 717–728. ACM, 2016.
- [144] S. Grange, F. Conti, P. Helmer, P. Rouiller, and C. Baur. Overview of the delta haptic device. In *Eurohaptics*, 2001.
- [145] T. H. Massie and J. K. Salisbury. The PHANToM haptic interface: A device for probing virtual objects. In *ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems In Dynamic Systems and Control*, volume 1, pages 295–301, 1994.
- [146] A. Cohen and E. Chen. Six degree-of-freedom haptic system as a desktop virtual prototyping interface. In *ASME Winter Annual Meeting, Dynamics Systems and Control, DSC*, volume 67, pages 401–402, 1999.
- [147] V. Hayward, P. Gregorio, O. Astley, S. Greenish, M. Doyon, L. Lessard, J. McDougall, I. Sinclair, S. Boelen, X. Chen, J.-G. Demers, J. Poulin, I. Benguigui, N. Almey, B. Makuc, and X. Zhang. *Freedom-7: A high fidelity seven axis haptic device with application to surgical training*, pages 443–456. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

- [148] P. Lambert and J. Herder. A novel parallel haptic device with 7 degrees of freedom. In *2015 IEEE World Haptics Conference (WHC)*, pages 183–188, 2015.
- [149] M. Aiple and A. Schiele. Pushing the limits of the CyberGrasp for haptic rendering. In *IEEE International Conference on Robotics and Automation*, pages 3541–3546, 2013.
- [150] C. S. Tzafestas. Whole-hand kinesthetic feedback and haptic perception in dextrous virtual manipulation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 33(1):100–113, 2003.
- [151] M. Feng, J. Li, and R. Zheng. Haptic rendering of virtual hand with force smoothing. In *2011 International Conference on Virtual Reality and Visualization*, pages 271–274, 2011.
- [152] M. Bergamasco, C. A. Avizzano, A. Frisoli, E. Ruffaldi, and S. Marcheschi. Design and validation of a complete haptic system for manipulative tasks. *Advanced Robotics*, 20(3):367–389, 2006.
- [153] P. Stergiopoulos, G. Moreau, M. Ammi, and P. Fuchs. A framework for the haptic rendering of the human hand. In *Proceedings. 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS*, pages 340–347. IEEE, 2003.
- [154] R. Fesharakifard, M. Khalili, L. Leroy, A. Paljic, and P. Fuchs. Integrating a grasp exoskeleton into a string-based interface for human-scale interactions. *Journal of Computing and Information Science in Engineering*, 8:041008–1 – 041008–9, 2008.
- [155] X. Gu, Y. Zhang, W. Sun, Y. Bian, D. Zhou, and P. O. Kristensson. Dexmo: An inexpensive and lightweight mechanical exoskeleton for motion capture and force feedback in VR. In *Proceedings of CHI Conference on Human Factors in Computing Systems, CHI '16*, pages 1991–1995. ACM, 2016.
- [156] O. Halabi and H. Kawasaki. Five fingers haptic interface robot HIRO: Design, rendering, and applications. 2010.
- [157] A. G. Perez, G. Cirio, D. Lobo, F. Chinello, D. Prattichizzo, and M. A. Otaduy. Efficient nonlinear skin simulation for multi-finger tactile rendering. In *2016 IEEE Haptics Symposium (HAPTICS)*, pages 155–160, 2016.

- [158] D. Wang, Y. Shi, S. Liu, Y. Zhang, and J. Xiao. Haptic simulation of organ deformation and hybrid contacts in dental operations. *IEEE Transactions on Haptics*, 7(1):48–60, 2014.
- [159] H. Xu and J. Barbič. Adaptive 6-DoF haptic contact stiffness using the gauss map. *IEEE Transactions on Haptics*, 9(3):323–332, 2016.
- [160] T. Laliberté, L. Birglen, and C. Gosselin. Underactuation in robotic grasping hands. *Machine Intelligence and Robotic Control*, 3:1–11, 2002.
- [161] M. A. Otaduy, C. Garre, and M. C. Lin. Representations and algorithms for force-feedback display. *Proceedings of the IEEE*, 101(9):2068–2080, 2013.
- [162] L. N. Verner and A. M. Okamura. Force and torque feedback vs force only feedback. In *World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 406–410, 2009.
- [163] P. Kadlecěk, P. Kmoch, and J. Křivánek. *Haptic Rendering for Under-Actuated 6/3-DoF Haptic Devices*, pages 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
- [164] M. J. Massimino and T. B. Sheridan. Sensory substitution for force feedback in teleoperation. *Presence: Teleoperators & Virtual Environments*, 2(4):344–352, 1993.
- [165] L. N. Verner and A. M. Okamura. Sensor/Actuator asymmetries in telemanipulators: Implications of partial force feedback. In *14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 309–314, 2006.
- [166] G. R. Luecke. Haptic interactions using virtual manipulator coupling with applications to underactuated systems. *IEEE Transactions on Robotics*, 27(4):730–740, 2011.
- [167] L. Meli and D. Prattichizzo. *Task-Oriented Approach to Simulate a Grasping Action Through Underactuated Haptic Devices*, pages 249–257. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
- [168] A. G. Perez, G. Cirio, F. Hernandez, C. Garre, and M. A. Otaduy. Strain limiting for soft finger contact simulation. In *2013 World Haptics Conference (WHC)*, pages 79–84, 2013.