Vision-Based Force Planning and Voice-Based Human-Machine Interface of an Assistive Robotic Exoskeleton Glove for Brachial Plexus Injuries

Yunfei Guo

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Pinhas Ben-Tzvi, Chair
Paul E. Plassmann
Creed F. Jones
Steve C. Southward
Ryan M. Gerdes

September 12, 2023
Blacksburg, Virginia

Keywords: Exoskeleton Glove, Exoskeleton Glove Force Control, Exoskeleton Glove Force Planning, Human Machine Interface, Speaker Verification

Copyright 2023, Yunfei Guo

Vision-Based Force Planning and Voice-Based Human-Machine Interface of an Assistive Robotic Exoskeleton Glove for Brachial Plexus Injuries

Yunfei Guo

(ABSTRACT)

This dissertation focuses on improving the capabilities of an assistive robotic exoskeleton glove designed for patients with Brachial Plexus Injuries (BPI). The aim of this research is to develop a force control method, an automatic force planning method, and a Human-Machine Interface (HMI) to refine the grasping functionalities of the exoskeleton glove, thus helping rehabilitation and independent living for individuals with BPI. The exoskeleton glove is a useful tool in post-surgery therapy for patients with BPI, as it helps counteract hand muscle atrophy by allowing controlled and assisted hand movements. This study introduces an assistive exoskeleton glove with rigid side-mounted linkages driven by Series Elastic Actuators (SEAs) to perform five different types of grasps. In the aspect of force control, data-driven SEA fingertip force prediction methods were developed to assist force control with the Linear Series Elastic Actuators (LSEAs). This data-driven force prediction method can provide precise prediction of SEA fingertip force taking into account the deformation and friction force on the exoskeleton glove. In the aspect of force planning, a slip-grasp force planning method with hybrid slip detection is implemented. This method incorporates a vision-based approach to estimate object properties to refine grasp force predictions, thus mimicking human grasping processes and reducing the trial-and-error iterations required for the slip-grasp method, increasing the grasp success rate from 71.9% to 87.5%. In terms of HMI, the Configurable Voice Activation and Speaker Verification (CVASV) system was developed

to control the proposed exoskeleton glove, which was then complemented by an innovative one-shot learning-based alternative, which proved to be more effective than CVASV in terms of training time and connectivity requirements. Clinical trials were conducted successfully in patients with BPI, demonstrating the effectiveness of the exoskeleton glove.

Vision-Based Force Planning and Voice-Based Human-Machine Interface of an Assistive Robotic Exoskeleton Glove for Brachial Plexus Injuries

Yunfei Guo

(GENERAL AUDIENCE ABSTRACT)

This dissertation focuses on improving the capabilities of a robotic exoskeleton glove designed to assist individuals with Brachial Plexus Injuries (BPI). The goal is to enhance the glove's ability to grasp and manipulate objects, which can help in the recovery process and enable patients with BPI to live more independently. The exoskeleton glove is a tool for patients with BPI to used after surgery to prevent the muscles of the hand from weakening due to lack of use. This research introduces an exoskeleton glove that utilizes special mechanisms to perform various types of grasp. The study has three main components. First, it focuses on ensuring that the glove can accurately control its grip strength. This is achieved through a special method that takes into account factors such as how the materials in the glove change when it moves and the amount of friction present. Second, the study works on a method for planning how much force the glove should use to hold objects without letting them slip. This method combines a camera-based object and material detection to estimate the weight and size of the target object, making the glove better at holding things without dropping them. The third part involves designing how people can instruct the glove what to do. The command can be sent to the robot by voice. This study proposed a new method that quickly learns how you talk and recognizes your voice. The exoskeleton glove was tested on patients with BPI and the results showed that it is successful in helping them. This study enhances

assistive technology, especially in the field of assistive exoskeleton glove, making it more effective and beneficial for individuals with hand disabilities.

Dedication

This dissertation is dedicated to

My parents: Zhou Guohong, and Guo hong

Acknowledgments

Firstly, the experimental procedure involving human subjects described in this dissertation received approval from the Carilion Clinic Institutional Review Board (IRB-19-330).

Secondly, the research presented in this dissertation was made possible by the support of the Eunice Kennedy Shriver National Institute of Child Health & Human Development of the National Institutes of Health, under Award Number R21HD095027. The authors hold sole responsibility for the content of this paper, and it does not necessarily reflect the official views of the National Institutes of Health.

Lastly, I would like to express my heartfelt gratitude to my advisor, Prof. Pinhas Ben-Tzvi, and my co-worker, Dr. Xu Wenda. Without the invaluable guidance and assistance provided by my advisor and co-worker, this remarkable achievement would not have been possible. I would also like to extend my thanks to my committee members and my family for their unwavering support throughout my Ph.D. journey. Additionally, I want to acknowledge the emotional support provided by my girlfriend, Tingting Song, and my cat, Tudou, during the challenging days.

Contents

List of Figures xiii

List of Tables xxi

1 Introduction 1

1.1 Motivation 1

1.2 Literature Review 3

1.2.1 Robotic Exoskeleton Gloves 3

1.2.2 Actuator Force Control Techniques for Rigid Robotic Exoskeleton Gloves 5

1.2.3 Grasp Force Planning Techniques for Robotic Exoskeleton Gloves . . 11

1.2.4 Human-Machine Interface (HMI) for Robotic Exoskeleton Gloves . . 13

1.3 Problem Statement and Proposed Solution 14

1.3.1 Actuator Force Control 15

1.3.2 Exoskeleton Glove Force Planning 15

1.3.3 Human-Machine Interface (HMI) Design 16

1.4 Dissertation Structure 16

1.5 Summary of Contributions 17

1.6 Selected Publications 18

2 Exoskeleton Glove Hardware Introduction **20**

2.1 Mechanical Design . . . 21

2.1.1 Exoskeleton Glove Linkage Optimization . . . 21

2.1.2 Linear Series Elastic Actuator (LSEA) Design . . . 26

2.1.3 Rotary Series Elastic Actuator (RSEA) Design . . . 27

2.2 Electrical Design . . . 28

3 Exoskeleton Glove Series Elastic Actuator (LSEA) Force Control **31**

3.1 LSEA Force Modeling . . . 31

3.1.1 Rotatory Encoder Setup . . . 33

3.1.2 Angular Potentiometer Setup . . . 34

3.1.3 Force Measurement at LSEA Fingertip Linkage . . . 35

3.1.4 Linkage Force Transformation Ratio . . . 36

3.2 Challenges of LSEA Force Calibration . . . 37

3.2.1 Friction Force Between Actuator Shafts and Housing . . . 38

3.2.2 The Combined Friction Force of the Linkage Joints and the Actuator Shafts . . . 39

3.2.3 Deformation of the Actuator Shafts and Linkages . . . 40

3.3 LSEA Force Calibration using Multi-Layer Perceptron (MLP) Method . . . 41

3.3.1 Network Tuning and Lost Function Selection . . . 42

3.4 LSEA Force Calibration using Surface Fitting (SF) Method . . . 43

3.5 LSEA Force Control using Multi-Layer Perceptron (MLP) Method 43
3.5.1 Low-level Control Method 45
3.5.2 High-Level Multi-Layer Perceptron (MLP) Compression Predictor . . 46
3.6 LSEA Force Control using Surface Fitting (SF) Method 46
3.7 Experimental Results 47
3.7.1 Data Preparation 48
3.7.2 Force Calibration Results 48
3.7.3 Force Control Results 50
3.8 Discussion 54

4 Exoskeleton Glove Force Planning Techniques 56

4.1 Introduction 56
4.2 Slip Grasp Force Planning 57
4.2.1 Hybrid Slip Detection 57
4.2.2 Experimental Results 58
4.3 Vision-Based Force Planning for Exoskeleton Gloves 59
4.3.1 Review of Vision-Based Material Analysis 60
4.3.2 System Overview 60
4.3.3 Target Object Detection 62
4.3.4 Size Estimation for Target Object 64

4.3.5 Material Classification for Target Object 67
4.3.6 Gasp Force Estimation 75
4.3.7 Experimental Results 75
4.3.8 Conclusion 88

5 Voice-Based Human-Machine Interface (HMI) Design 90

5.1 Introduction 90
5.2 Voice-Based HMI with Text-Independent Speaker Verification 90
5.2.1 Review of Configurable Voice Activation and Speaker Verification (CVASV) HMI 91
5.2.2 Experimental Results 98
5.3 Voice-Based HMI with Text-Dependent Speaker Verification 105
5.3.1 Command Detection 106
5.3.2 Text-dependent Speaker Verification 107
5.3.3 One-shot Learning Based Voice HMI System Design 112
5.3.4 Experimental Results 112

6 Exoskeleton Glove System Integration and Experiments 115

6.1 Control Architecture 115
6.1.1 Low-level Controller 117
6.1.2 High-level Controller 118

6.2 Integration of CVASV HMI 118
6.3 Power Consumption 120
6.4 Human Subject Experiments 120
6.4.1 Experiments on Subjects with Healthy Hand 122
6.4.2 Experiments on Subjects with BPI 123
6.5 Comparison with State-of-the-Art Exoskeleton Gloves 123

7 Conclusion and Future Work **126**
7.1 Summary 126
7.2 Directions for Future Work 127

Bibliography **128**

List of Figures

1.1 (A) Patient with BPI on her right hand. (B) Surgical option to reconstruct the nerve system on the forearm. (C) The patient failed to grasp duct tape. (D) The patient failed to grasp a marker pen. (E) The patient successfully grasped duct tape with the help of an exoskeleton glove. (F) The patient successfully grasped a marker pen with the help of an exoskeleton glove. (G) Top view of the exoskeleton glove used in this clinical experiment. (H) Side view of the exoskeleton glove used in this clinical experiment. 2

1.2 The state-of-the-art wearable robotic exoskeleton glove systems. (A) Exoskeleton glove designed by Ryu et al. [1]. (B) Exoskeleton glove designed by Nycz et al. [2]. (C) Gloreha Lite exoskeleton glove [3]. (D) Hand of Hope [4]. (E) Exoskeleton glove designed by Dragusanu et al. [5]. (F) Exoskeleton glove designed by Ferguson et al. [6]. (G) Graspy Glove [7]. (H) Exoskeleton glove designed by Agarwal et al. [8]. (I) KULEX Hand [9]. 5

1.3 Examples of exoskeleton gloves which used SEA. (A) An index finger exoskeleton with series elastic actuation for rehabilitation [10]. (B) Series elastic transmission hand exoskeletons [11]. (C) A wearable and force-controllable hand exoskeleton system [12]. (D) A general-purpose robotic hand exoskeleton with series elastic actuation [13]. (E) A robotic glove system for patients with Brachial Plexus Injuries [14]. 8

1.4 Human-Machine Interface (HMI) designed by previous researchers. (A) EEG-based HMI designed for neurorehabilitation [15]. (B) EEG-based HMI designed for patients with Quadriplegia [16]. (C) Voice-base HMI designed for patients with hand disabilities [17]. (D) EMG-based HMI designed for patients with hand disabilities [18]. (E) Vision-based HMI designed for Spinal Cord Injuries [19]. 14

2.1 Five different types of grasp can be performed by the proposed exoskeleton glove. (A) Cylinder grasp. (B) Sphere grasp. (C) Tripod grasp. (D) Tip grasp. (E) Lateral grasp. 21

2.2 Exoskeleton glove prototype. (A) Top view. (B) Side view. 22

2.3 Index finger LSEA linkage design. 23

2.4 LSEA linkage optimization results. The red line represents the optimized linkages motion. Multi-color lines represent human finger motion. (A) Ring finger optimization result. (B) Thumb finger optimization result. 25

2.5 Actuator, linkage, and abduction-adduction mechanism design of the LSEA. 27

2.6 RSEA design. (A) Exploded view of RSEA. The red box shows the mechanical limit to prevent SEA from back-drive. (B) RSEA side view. 28

2.7 Electronics design overview. Yellow: microcontroller board. Green: brushed motor controller board. Blue: Power conversion board. Red: 3-in-1 battery package. 29

3.1 The legend for the calculation of the SEA force is as follows: Point A is the endpoint of the output shaft of the LSEA. Point B is the MCP joint. Points C and D are located on the input shaft of the LSEA. Point E is situated on the LSEA housing, which is stationary. Point F is the fingertip endpoint of the LSEA linkage. The 12-CPR magnetic encoder, which is marked in blue, is used to measure the length of DE. The rotary potentiometer, which is marked in red, is used to measure the angle of the MCP joint. 32

3.2 Fingertip linkage force measurement setup for LSEA. (A) Horizontally mounted load cell used for LSEA fingertip linkage force measurement when the actuator is not fully extended. (B) Vertical mounted load cell used for LSEA fingertip linkage force measurement when the actuator is fully extended. . . 36

3.3 LSEA actuator output shaft to fingertip linkage force transformation ratio. (A) Index finger LSEA linkage force transformation ratio when $\angle ABC$ is between 5 degree and 16 degree. (B) Index finger LSEA linkage force transformation ratio when $\angle ABC$ is between 16 degree and 28 degree. 37

3.4 Difference of the actuator output force and measured force due to friction force. (A) Force output measurement setup. (B) A comparison was made between the calculated (ideal), measured, and calibrated output force. The calibrated value was expected to be the same as the measured value, however, it did not match. 38

3.5 Different index finger linkage angle have different friction force. (A) The measured friction force at 8 degree is 2.1N (measured by 0.51mm spring compression). (B) The measured friction force at 16 degree is is 2.4N (measured by 0.57mm spring compression). (C) The measured friction force at 25 degree is is 4.45N (measured by 1.06mm spring compression). 39

3.6 Force measurement error caused by deformation. (A) Inconsistency in force measurement caused by actuator shaft deformation. (B) Force measurement error caused by linkage deformation. 40

3.7 The Multi-Layer Perceptron (MLP) LSEA fingertip force prediction method. 41

3.8 SEA control software architecture using MLP method. The low-level control is running on a microcontroller. The desired compression is fed into the microcontroller by the high-level MLP SEA force predictor through cloud computing. 44

3.9 SEA control software architecture using SF method. All computation is performed locally on microcontroller. The SF method uses spring compression and filtered potentiometer data to predict the desired spring compression. . . 47

3.10 LSEA fingertip linkage force calibration model built using Surface Fitting (SF) method. 49

3.11 LSEA force calibration simulation performed using data from the validation dataset. 50

3.12 LSEA compression prediction model built using Surface Fitting (SF) method. 51

3.13 LSEA compression setpoint prediction simulation performed using data from the validation dataset. 52

3.14 Index finger LSEA force control experiments. This experiment is performed on the LSEA hardware. The MCP joint begins at an angle of 5 degrees and makes contact with the load cell at 10 degrees. (A) The desired LSEA output shaft force is set to 5N. (B) The desired LSEA output shaft force is set to 10N. (C) The desired LSEA output shaft force is set to 15N. 53

4.1 Hybrid slip detection and slip-grasp force planning method demonstration. (A) Slip detected using the FSR raw reading, the force added within 500ms after a slip is detected. (2) Slip detected using the SEA raw reading, the force added within 500ms after a slip is detected. 58

4.2 Slip detection and slip-grasp force planning method experiment setup. The red box showed the amount of slippage observed on the target object. (A) The user can lift the object without falling. (B) The user shakes the target object causing it to slip. (C) Force is added to stop the target object from slipping. 59

4.3 Overview of vision-based initial grasp force prediction procedure. 61

4.4 Sample images of the exoskeleton grasping environment, object category, and object material. 62

4.5 Illustration of the camera, marker, and pixel coordinates. 66

4.6 Training and inference procedure for vision-based material classification and segmentation. 70

4.7 The coordinate systems for initial force estimation. WCS: world coordinate system. ECS: exoskeleton glove coordinate system. ICS: i-th fingertip coordinate system. 76

4.8 Sample images used in the material classification training. 77

4.9 Performance comparison of different Object detection methods. (A) mean Average Precision (mAP) at 50% Intersection over Union (IoU) of 7 different state-of-the-art neural networks. (B) mAP vs. average inference time of each neural network. 78

4.10 Examples of ground truth object size measurements. 80

4.11 Sample material segmentation results. 81

4.12 Experimental results of grasping daily used objects using vision-based initial grasp force prediction method and slip-grasp method. Blue: number of successful grasps performed using the vision-based initial force estimation with slip-grasp method. Red: number of successful grasps performed using only the slip-grasp method. Yellow: the total number of grasps for each individual method. 85

4.13 Series Elastic Actuators (SEA) are used to apply force on the exoskeleton glove in the grasp experiment. 86

4.14 Demonstration of grasping daily used objects using vision-based initial grasp force prediction method and slip-grasp method. (A) Successfully grasp a 512g water bottle with vision system. (B) Failed to grasp a 512g water bottle using the slip-grasp method due to inadequate thumb torque. (C) and (D) Successfully grasp an 188g wine glass with both the vision system and the slip-grasp method. (E) Successfully grasp a 3g plastic spoon with vision system. (F) Failed to grasp a 3g plastic spoon using the slip-grasp method due to excessive force and torque. (G) and (H) Successfully grasp a 48g metal spoon with both the vision system and the slip-grasp method. 87

5.1 CVASV HMI system overview. 92

5.2 Flowchart of the command recognition stage in the CVASV HMI. Demonstrating bandpass filter and command aggregation logic. 95

5.3 Speaker verification model training and inference procedure using MobileNetV1. (A) Train the MobileNetV1 feature extractor by performing speaker classification on the VoxCeleb1 dataset. (B) The last dense layer of the trained speaker classification model is removed. The model is used as speaker utterance feature extractor. The extracted features of the enrolled and tested speakers are compared to verify the enrolled speaker. 97

5.4 ROC curve of the proposed speaker verification method tested on VoxCeleb1 test dataset. 99

5.5 Confusion matrix of CVASV HMI validation experimental results on the collected dataset at cosine distance threshold = 0.30. Label A-J represents speaker A-J. The amount of true positive commands are marked in blue and the amount of false positive commands are marked in red. 102

5.6 HMI operation success rate for 140 voice commands from 2 subjects. Each keyword is tested 20 times, 10 of which should be accepted, and the other ten should be rejected. 103

5.7 The neural network structure of the one-shot-based text-dependent speaker verification method. 109

5.8 The structure of the modified voice-based HMI using the one-shot learning method. 113

6.1 Microcontroller software architecture. Hardware is marked in yellow. Controller is marked in green. The LSEA fingertip force prediction is running locally using SF method. The microcontroller sends sensor data to the computer and receives force command from the computer. 116

6.2 Computer software architecture. The speaker verification and slip-grasp force planning is performed on the computer. 119

6.3 Examples of successful grasps. Each object is grasped five times by a single participant. (A) bottle, (B) jar, (C) screwdriver, (D) ball, (E) tape, (F) small box, (G) pen, (H) small ball, (I) flat box, (J) bowl. 122

6.4 Grasp success rate for 150 grasping trails. This experiment is performed on 3 subjects with healthy hand. 123

6.5 Comparison between grasping objects with and without the proposed exoskeleton glove. This experiment is performed on patients with BPI. 124

List of Tables

1.1 Summary of hand exoskeleton grasp assistance systems 6

3.1 SEA fingertip force calibration model hyper-parameter tuning results 49

3.2 SEA compression prediction model hyper-parameter tuning results 50

3.3 Force control error and settling time when experimenting on LSEA hardware 53

3.4 Force calibration performance comparison on the validation datset 54

3.5 SEA force control performance comparison on the validation datset 55

4.1 Results of training on MINC-2500 and testing on the collected dataset . . . 69

4.2 Results of transfer ImageNet and MINC-2500 weight on the collected dataset 69

4.3 Object size estimation experimental results 80

4.4 Object weight estimation experimental results 83

4.5 Comparison between vision-based force estimation and slip-grasp force planning 84

4.6 Inference speed of processing one 1280x760 pixel image using the vision-based grasp force planning method 88

5.1 Performance comparison between modified MobileNetV1 and the baseline VGG-M method 98

5.2 Performance comparison between the modified MobileNetV1 and the state-of-the-art speaker verification methods on VoxCeleb1 dataset 100

5.3 CVASV HMI performance tested on the collected dataset at cosine distance threshold = 0.30 . . . 101

5.4 Time cost to process one audio command using CVASV HMI . . . 104

5.5 Power consumption of CVASV HMI . . . 104

5.6 Comparison between CVASV HMI and other state-of-the-art voice-based HMI system . . . 105

5.7 Comparison between different prepossessing techniques and comparator networks used for one-shot learning speaker verification . . . 110

5.8 Performance comparison between CNN-cosine speaker verification method and one-shot speaker verification method . . . 111

5.9 Speaker verification system training and inference speed comparison . . . 112

5.10 HMI system processing speed comparison . . . 114

5.11 Comparison between the proposed HMI systems and other state-of-the-art voice-based HMI systems . . . 114

6.1 Power consumption of the proposed exoskeleton glove . . . 121

6.2 Target objects selection . . . 121

6.3 Summary of hand exoskeleton grasp assistance systems . . . 125

List of Abbreviations

API: Application Programming Interface

BPI: Brachial Plexus Injuries

CNN: Convolution Neural Network

COCO: Common Objects in Context Database

CPR: Count Per Revolution

CRF: Conditional Random Field

CVASV: Configurable Voice Activation and Speaker Verification

DIP: Distal Interphalangeal

DOF: Degree of Freedom

EEG: Electroencephalogram

EER: Equal Error Rate

EMG: Electromyography

FA: False Acceptance

FMD: Flicker Material Dataset

FSCE: Force-Sensitive Conductive Elastomer

FSR: Force Sensitive Resistor

GAP: Global Average Pooling

HMI: Human Machine Interface

HMM: Hidden Markov Model

IMU: Inertial Measurement Unit

IoU: Intersection over Union

LSEA: Linear Serial Elastic Actuator

MAE: Mean Absolute Error

mAP: mean Average Precision

MAPE: Mean Absolute Percentage Error

MBN: MobileNetV1

MCP: Metacarpophalangeal

Mel: Mel-spectrum

MFCC: Mel Frequency Cepstral Coefficients

MINC: Material Recognition in the Wild with the Materials in Context Database

MLP: Multi-Layer Perception

MSE: Mean Squared Error

NLP: Natural Language Processing

PCB: Printed Circuit Board

PID: Proportional–Integral–Derivative

PIP: Proximal Inter-Phalangeal

ROC: Receiver Operating Characteristic

RPM: Round Per Minute

RSEA: Rotary Serial Elastic Actuator

SEA: Serial Elastic Actuator

SF: Surface Fitting

SSD: Single Shot Detector

STT: Speech To Text

TA: True Acceptance

TAP: Temporal Average Pooling

WER: Word Error Rate

Chapter 1

Introduction

1.1 Motivation

Brachial Plexus Injuries (BPI) are commonly caused by motorcycle or snowmobile accidents, resulting in damage to the neural system of the hand, arm, and shoulder, leading to mobility and sensation loss. Research studies indicate that 90% of individuals involved in motorcycle accidents sustain BPI, and approximately 80% of these patients are men aged between 25 and 35 [20]. Moreover, 66.67% of BPI patients report difficulties in performing daily activities that involve hand, arm, and elbow movements [20]. While surgical interventions can partially restore arm and shoulder mobility, they are ineffective in restoring hand functionality [21]. Consequently, assistive robotic exoskeleton gloves find widespread use in post-surgical physical therapy for BPI patients [22, 23, 24, 25, 26]. Exoskeleton gloves are promising as a physical therapy method to prevent hand muscle atrophy [27]. Additionally, wearable robotic exoskeleton gloves function as assistive devices beyond physical therapy, enabling grasping for BPI patients and allowing their participation in daily activities [28, 29].

Fig. 1.1 illustrates a successful application of the exoskeleton glove in restoring grasp functionality for patients with BPI. Fig. 1.1 (A) portrays a patient with BPI on her right hand, exhibiting muscle atrophy. Despite undergoing surgery to reconstruct the nerve system in her shoulder and forearm, significant weakness persists. When grasping objects, as depicted in Fig. 1.1 (C) - (F), she relies on her healthy hand to lift her paralyzed arm and hand. To

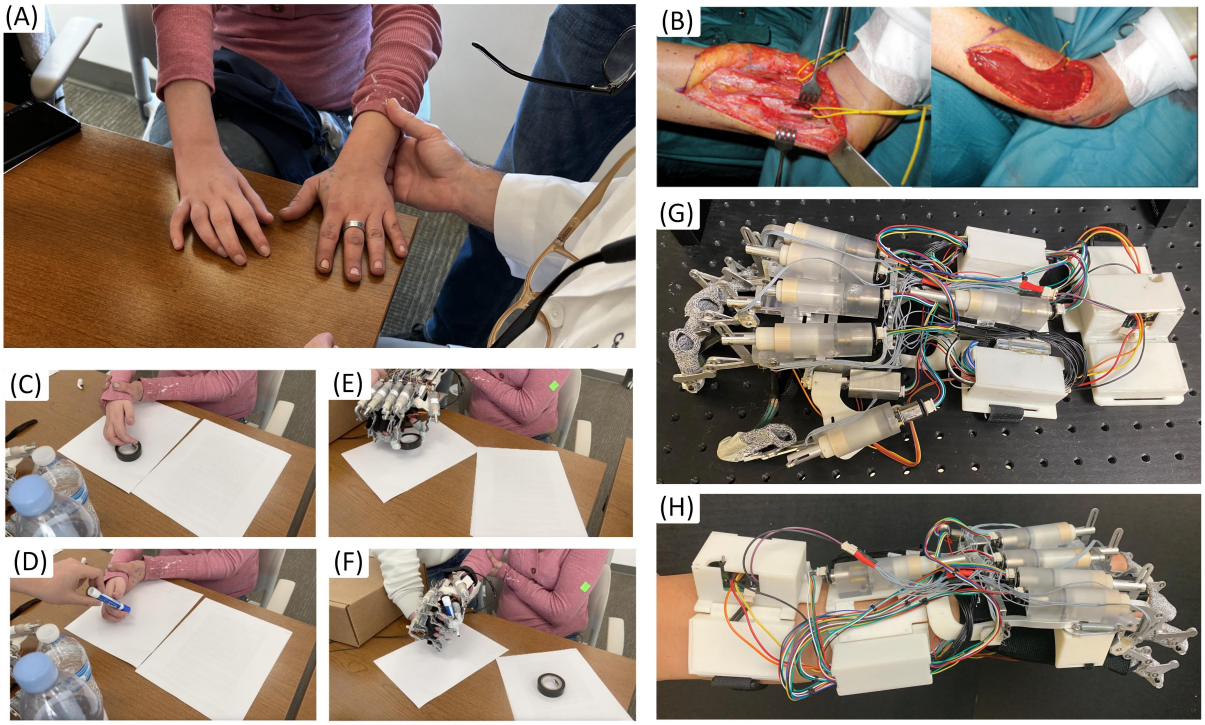


Figure 1.1: (A) Patient with BPI on her right hand. (B) Surgical option to reconstruct the nerve system on the forearm. (C) The patient failed to grasp duct tape. (D) The patient failed to grasp a marker pen. (E) The patient successfully grasped duct tape with the help of an exoskeleton glove. (F) The patient successfully grasped a marker pen with the help of an exoskeleton glove. (G) Top view of the exoskeleton glove used in this clinical experiment. (H) Side view of the exoskeleton glove used in this clinical experiment.

help patients with BPI, a voice-controlled robotic exoskeleton glove system was proposed to assist in partially restoring hand functionality (illustrated in Fig. 1.1 (E) and (F)).

The motivation of this research is to develop an easy-to-use assistive robotic exoskeleton glove for patients with BPI, aiming to enhance their quality of life through its application in post-surgery physical therapy and as an assistive device. This research addresses three key questions: (1) What techniques are necessary to achieve precise actuator force control in the exoskeleton glove designed for patients with BPI? (2) What techniques are required to ensure robust grasp force planning in the exoskeleton glove designed for patients with BPI? (3) Which Human-Machine Interface (HMI) is the most suitable for the exoskeleton glove

designed for patients with BPI?

By addressing these questions, this research aims to offer valuable insights and advancements in the development of assistive technology for individuals with BPI.

1.2 Literature Review

In order to answer the three primary questions mentioned above, a thorough review of the most up-to-date studies in each field was conducted.

1.2.1 Robotic Exoskeleton Gloves

Previous researchers have designed multiple robotic exoskeleton gloves [30]. Wearable exoskeleton gloves attract more attention from researchers due to their portability. Compared to stationary exoskeleton gloves, wearable robotic exoskeletons offer a lightweight, portable design and are equipped with a portable power source. As a result, these wearable robotic exoskeleton gloves can be carried by users to assist them in performing daily living tasks.

Wearable exoskeleton gloves can be classified into two main categories: soft robotic exoskeleton gloves [2, 3, 7, 29, 31, 32, 33, 34, 35, 36, 37] and rigid robotic exoskeleton gloves [4, 13, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51].

Soft exoskeleton gloves, characterized by their lack of a rigid frame, are widely recognized for their superior wearability. The actuation of these gloves is achieved through the contraction and extension of inflatable materials or artificial tendon mechanisms. Soft exoskeleton gloves are particularly well-suited for post-stroke rehabilitation scenarios, where users only require partial assistance from the exoskeleton to support their muscles. However, a major drawback of soft exoskeleton gloves is the absence of a rigid linkage [52]. Consequently, a

significant portion of the force generated by the actuator is transmitted to the user's body instead of being directed towards the object being grasped. This limitation renders soft exoskeleton gloves unsuitable for individuals seeking an assistive device that can fully replace the functionality of their hand.

As previously mentioned, rigid exoskeleton gloves serve as suitable assistive devices capable of fully replacing the functionality of the hand. These gloves feature rigid linkages that effectively transmit force from the actuator to the grasp object. Rigid exoskeleton gloves can be classified into two primary categories, namely remote-driven rigid exoskeleton gloves [38, 49, 50, 51] and direct-driven exoskeleton gloves [4, 13, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48].

Remote-driven rigid exoskeleton gloves typically employ pneumatic actuators [49, 50, 51] or wire-driven linkages [38] to allow for the remote placement of the compressor or actuator, thus saving space. However, it is worth noting that the size of the compressor and air cylinder is typically large [35] (shown in Fig. 1.2 (A)), which can affect the overall portability of the system. On the other hand, using wire-driven linkages necessitates intricate wire routing and linkage structures, which occupy a significant amount of space on the hand (shown in Fig. 1.2 (F) and (H)).

Direct-driven exoskeleton gloves utilize electric actuators to directly drive the linkages, which has emerged as a more popular solution among researchers due to its simplicity in linkage design (shown in Fig. 1.2 (E) and (I)). However, the limited space available on the hand necessitates a trade-off between the number of actuators and the output of each actuator. For instance, if five powerful actuators are employed in the exoskeleton, the size of the exoskeleton tend to be bulky which affect wearability (shown in Fig. 1.2 (D)). Conversely, keeping the exoskeleton compact requires a limited number of actuators can be accommodated on the exoskeleton (shown in Fig. 1.2 (I)).

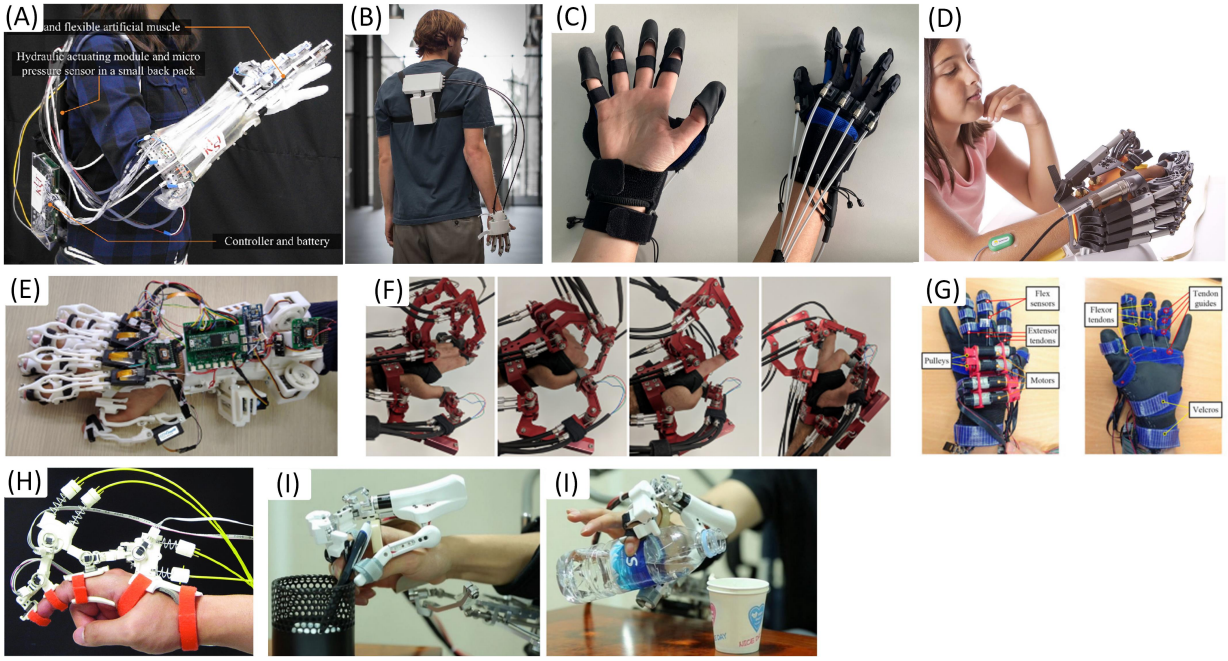


Figure 1.2: The state-of-the-art wearable robotic exoskeleton glove systems. (A) Exoskeleton glove designed by Ryu et al. [1]. (B) Exoskeleton glove designed by Nycz et al. [2]. (C) Gloreha Lite exoskeleton glove [3]. (D) Hand of Hope [4]. (E) Exoskeleton glove designed by Dragusanu et al. [5]. (F) Exoskeleton glove designed by Ferguson et al. [6]. (G) Graspy Glove [7]. (H) Exoskeleton glove designed by Agarwal et al. [8]. (I) KULEX Hand [9].

An overview of the essential specifications of state-of-the-art wearable exoskeleton gloves is provided in Tab. 1.1. The corresponding exoskeleton gloves are also listed in Fig. 1.2 in the same order.

1.2.2 Actuator Force Control Techniques for Rigid Robotic Exoskeleton Gloves

In a comprehensive review of exoskeleton gloves conducted in 2021 [53], it was surprising to discover that less than half of the reviewed papers mentioned force control. This finding is remarkably low considering the critical importance of applying the correct amount of force to ensure a stable grasp. The majority of researchers primarily rely on position control for

Table 1.1: Summary of hand exoskeleton grasp assistance systems

Device	Actuation, Transmission, Category	Weight: glove/total	Peak Fingertip Force	Fingers
Ryu et al. [1]	hydraulics, cable, Soft	2.62 kg / 2.74 kg	12 N	3
Nycz et al. [2]	linear motor, cable, Soft	113 g / 867 g	8.7 N	4
Gloreha Lite [3]	motor, cable, Soft	80 g / 5 kg	5 N	5
Hand of Hope [4]	linear motor, linkage, Rigid	700 g / -	12 N	5
Dragusanu et al. [5]	linear motor, linkage, Rigid	- / around 500 g	-	5
Ferguson et al. [6]	motor, cable, Rigid + Remote	- / -	-	3
Graspy Glove [7]	motor, cable, Soft	250 g / 340 g	16 N	4
Agarwal et al. [8]	SEA, cable, Rigid + Remote	80 g / -	-	1
KULEX Hand [9]	motor, linkage, Rigid	- / 172 g	10 N	2

grasping [45, 54, 55, 56], which is not an ideal approach as it lacks generalizability across different objects. It is worth noting that the focus of this research is on designing an assistive exoskeleton glove for patients with BPI. Based on the literature review of existing hardware, this study will specifically concentrate on a direct-driven rigid exoskeleton glove featuring electric motors as actuators.

The control methods for rigid exoskeleton actuators can be classified into two distinct categories: SEA force control [10, 11, 12, 13] and sensor-based force control [47, 57, 58, 59]. Both methods are discussed in the following sections.

Series Elastic Actuator (SEA) Force Control

The concept of SEA was first proposed by the MIT Artificial Intelligence Laboratory in 1995 [60]. This design involves connecting a torsion spring in series between the input and output shaft of the electric motor. The output force is then determined by measuring the torsion spring twist angle. SEAs offer several advantages over rigid actuators, such as precise force

sensing and a wider control bandwidth. Springs provide more linear and repeatable force sensing compared to Force-Sensing Resistors (FSRs). Additionally, they act as low-pass filters, attenuating high-frequency motion and allowing for a broader control bandwidth.

Since the inception of SEA designs, they have been used in robotic exoskeletons. Kim et al. [61] created hydraulic SEAs to help with walking and standing up motions in exoskeletons. Karavas et al. [62] developed electric SEAs for lower limb exoskeleton joints. These designs provide precise force control and a high output force. However, they necessitate large hydraulic or electrical power sources, as well as metal housings and output shafts. The main difficulty in applying these SEAs to robotic exoskeleton gloves is their size and weight.

Measuring force on the fingertips of an exoskeleton glove using SEAs requires two principles to be followed. First, the elastic material used should have low stiffness, which makes it easier to measure deformation. Second, all components involved in force transformation, including SEAs and exoskeleton parts, must be highly rigid. This ensures that the deformation measured is mainly from the elastic material. However, it is difficult to implement these two principles in exoskeleton glove design. High rigidity often leads to bulky and heavy designs, while low stiffness limits the maximum output force of the SEAs.

Previous research in this area has encountered issues such as inaccurate force feedback due to exoskeleton deformation or inadequate force output that does not meet user requirements.

Bianchi et al. [11] proposed an exoskeleton glove with SEA using titanium alloy as an elastic material. This design is depicted in Fig. 1.3 (B). The exoskeleton can output 20N of force on each fingertip. However, there are three drawbacks to this design. First, due to the high force output of the SEA, the size of the motor used on such an SEA needs to be powerful but large. The size of the exoskeleton limited the number of motors that can be used. The design coupled the middle, ring, and little finger in one motion to keep the design compact, reducing

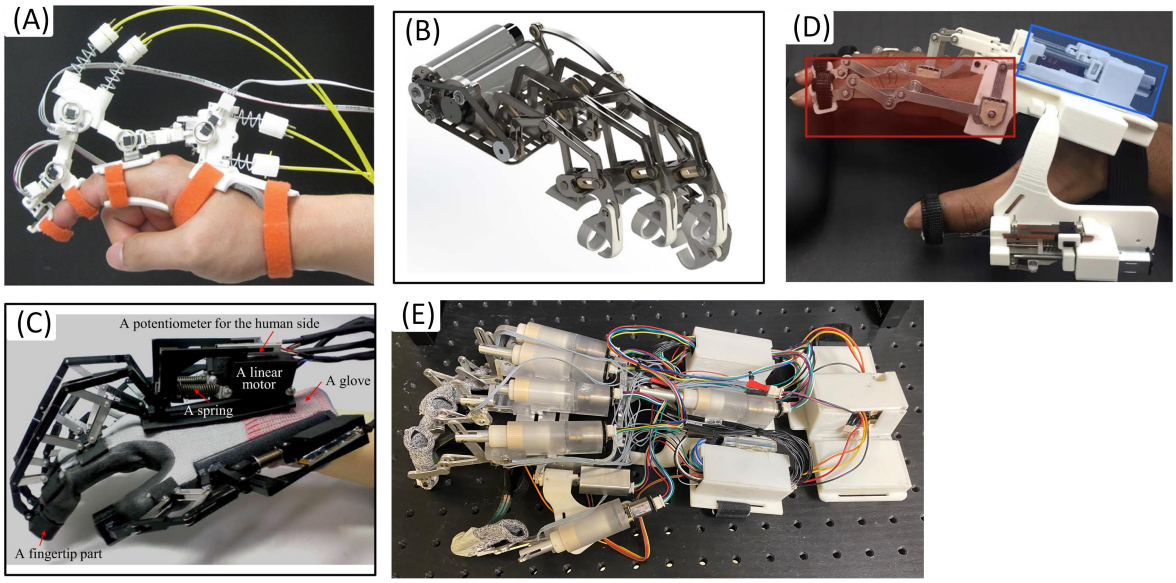


Figure 1.3: Examples of exoskeleton gloves which used SEA. (A) An index finger exoskeleton with series elastic actuation for rehabilitation [10]. (B) Series elastic transmission hand exoskeletons [11]. (C) A wearable and force-controllable hand exoskeleton system [12]. (D) A general-purpose robotic hand exoskeleton with series elastic actuation [13]. (E) A robotic glove system for patients with Brachial Plexus Injuries [14].

degrees of freedom (DOFs) will limit the functionality of such an exoskeleton glove. Second, the use of a high-stiffness elastic material makes the goal of accurate force measurement more difficult to achieve. The elastic material has a stiffness constant of 58 N/mm. This research does not propose a valid method for measuring fingertip linkage force.

Agarwal et al. [10] designed a rigid link cable-driven exoskeleton glove using two SEAs on each finger. This previous design is shown in Fig. 1.3 (A). The force control used in this previous research has about 10% error and can produce a 0.3Nm peak torque on each SEA output joint. The main issues with this design are size and weight. To output a 0.3Nm torque to one finger requires two large RE-Max 29 motors. The user must carry the power source (motor, motor pulley, battery), dramatically reducing mobility. Moreover, instead of measuring the force at the fingertips of the exoskeleton linkage, this design only measures the force at the SEA output joint. According to experiments, there usually exists a huge

difference between the SEA output force and the fingertip linkage output force. Failing to measure the force on the fingertip linkage will affect the grasping of the exoskeleton [63].

Jo et al. [12] designed a compact linear SEA for robotic exoskeleton gloves. This design is shown in Fig. 1.3 (c). Accurate force control is achieved with a linear-quadratic tuned proportional-derivative controller to control the distance and a disturbance observer to model the uncertainty. However, the spring constant is very low (0.343 N/mm), resulting in a maximum output force of 9N to the SEA output shaft. This research does not report the output force measured on the fingertip linkage.

Refour et al. [13] designed a compact Linear Series Elastic Actuator (LSEA) for robotic exoskeleton gloves, which can output 20N on each fingertip linkage. This design is shown in Fig. 1.3 (D). One of the main issues with this design is the accuracy of the force control. According to the author, the LSEA experiences 2-3N of output force error when measuring force on the SEA output shaft. However, due to the deformation of the plastic LSEA housing and the thin aluminum linkage, the force output to the fingertip linkage may be inaccurate. This research does not report the accuracy of the force control when measuring the force on the fingertip linkage.

Xu et al. [64] designed a compact LSEA shown in Fig. 1.3 (E). This LSEA has a high force output of 30N at the output shaft and 10N at each fingertip. The LSEA uses thin aluminum linkages with 3D-printed plastic SEA output shafts and housing. Due to the high output and low rigidity of the LSEA and exoskeleton linkage, both the linkage and the LSEA output shaft will deform when a large force is applied. This deformation causes the force output of the LSEA to be non-linear to the strain of the elastic material, thus making fingertip linkage experience up to 50% force measurement error.

Force Sensor Based Force Control

The use of force sensors for force control in rigid exoskeleton gloves is a popular approach. Force sensors are a more compact alternative to SEAs and can be easily installed on the fingertips, allowing direct force measurement.

Lee developed the iSAFER glove, which used a Force-Sensitive Conductive Elastomer (FSCE) to measure the force exerted on the fingertip [65]. However, this approach has drawbacks such as noisy data and the need for individualized manufacturing and calibration for each new user, based on their unique finger shape, making it difficult to achieve generalizability.

Ma et al. [48] designed a robotic exoskeleton glove with strain gauges as a tactile sensor to measure force. The strain gauges are mounted on the retaining ring, where users mount their fingers on the exoskeleton linkage. This approach has advantages such as direct force measurement, good repeatability, and a compact size. However, it requires calibration for each new user to adapt to different finger sizes, otherwise inaccurate force measurement will occur.

Diez et al. [59] introduced a robotic exoskeleton that incorporates an optical force sensor as the tactile sensor, addressing the accuracy limitations observed with FSRs. This approach enables a consistent and precise force measurement on the fingertip linkage of the exoskeleton glove without the need for customized calibration. However, the optical nature of the sensor requires substantial signal amplification and processing equipment, compromising the portability of the system.

1.2.3 Grasp Force Planning Techniques for Robotic Exoskeleton Gloves

Research on force planning methods for robotic exoskeleton gloves has been focused on improving and increasing the muscle movements of users, particularly in the context of rehabilitation [17, 66, 67, 68, 69, 70]. However, these studies do not take into account patients with BPI, who often lack voluntary muscle movements. Therefore, there is a lack of developed exoskeleton force planning methods for this group of people. To bridge this gap, this section also reviews force planning methods for robotic grippers, which have a similar fundamental idea. These methods focus on determining the necessary force to achieve a stable grasp on a target object.

Grasp force planning methods can be broadly categorized into three main groups: traditional force planning [46, 71], machine learning based force planning [72, 73, 74], and slip-grasp force planning [65, 75, 76].

Traditional force planning techniques necessitate the utilization of equations with precise grasp position, friction coefficient, and weight to calculate the optimal contact forces. Teja et al. [46] developed two methods to meet the two conditions required for a secure grasp. These include the prevention of deformation of soft objects and the maintenance of force and moment equilibrium of the objects being grasped. Cheng et al. [71] used the compact-dual linear programming method to determine the force distribution for a robotic grasping system called DIGITs. Nevertheless, setting up a physical model can be difficult, since the grasping position, angle, and contact point can be varied.

Machine learning based force planning is analogous to traditional force planning techniques, but instead of constructing intricate equations, neural networks are used to simulate the physical environment. Youshen et al. [72] suggested the use of recurrent neural networks

for optimizing grasp forces for multi-fingered robotic hands. Xiong et al. [73] proposed an algorithm based on artificial neural networks to determine the joint torques for a multi-finger robotic gripper. Liang et al. [74] proposed the use of deep reinforcement learning to identify the optimal force distribution for a multi-finger robotic gripper.

Grasp force planning, both through traditional and machine learning based methods, requires precise information about the grasping position, object weight, surface material, and size of the object in order to calculate the optimal force. However, when using assistive exoskeleton gloves, it is often necessary to grasp objects of varying sizes and weights without having explicit prior measurements. This presents a challenge to the algorithms mentioned above, making their use in such situations difficult.

The slip-grasp force planning method can identify adequate force through trial and error, without the need to explicitly measure the object. Lee et al. [65] proposed a custom-made Force-Sensitive Conductive Elastomer (FSCE) sensor to detect slippage in the fingertips of the iSAFER exoskeleton glove. The force controller adds force to the exoskeleton glove fingertip linkages based on the slippage between the fingertips and the grasped item in both of the methods mentioned above. However, the reinforcement process usually leads to a laborious grasping process where the user has to find the optimal grasp force through failures, which is not practical for exoskeleton glove users. Additionally, slip detection on a robotic exoskeleton glove is different from a robotic hand or gripper due to space and size constraints. Previous researchers have designed multiple slip-detection sensors for robotic hands and grippers and have achieved good results using slip-grasp force planning method [75, 76]. But unlike a robotic gripper, there is not enough space for larger and more accurate slip-detection sensors to be fitted at the fingertips in an exoskeleton glove application. The limitation on sensors makes the slip-grasp force planning method suffer from accuracy issues.

1.2.4 Human-Machine Interface (HMI) for Robotic Exoskeleton Gloves

HMI is a key factor in the exoskeleton glove system, allowing users to operate the device with minimal effort. HMIs of the exoskeleton glove can be divided into four types: Electromyography (EMG)-based [17, 66, 67, 68, 69, 70], Electroencephalogram (EEG)-based [15, 16], voice-based [17], and vision-based [19]. Nevertheless, each of these approaches has its own drawbacks.

Fig. 1.4 (C) illustrates an EMG-based HMI which uses muscle signals from the forearm to control an exoskeleton glove. This type of HMI is the most popular one for post-stroke rehabilitation, as it allows for real-time user-guided force planning. Unfortunately, it is not suitable for patients with BPI, since they lack control over their muscles.

Noninvasive EEG-based HMIs use brain signals to control an exoskeleton glove, requiring minimal effort from the user. However, this system is not very portable due to the need for EEG probes (as seen in Fig. 1.4 (A) and (B)). Additionally, it is not cost-effective.

Vision-based HMIs, as illustrated in Fig. 1.4 (E), use cameras to identify the user's point of attention and detect the object at that point. This HMI offers good portability and requires little effort from the user. However, vision-based HMI is heavily dependent on the accuracy of object detection. Additionally, it can be difficult for vision-based HMI to accurately detect objects when multiple items are present in the field of view [19, 77, 78].

Voice-based HMIs (as seen in Fig. 1.4 (C)) allow users to control the exoskeleton by issuing commands through their voice. This approach is cost-effective and portable, however, the lack of speaker verification and authentication can be a hindrance when using the exoskeleton glove [17, 79, 80].

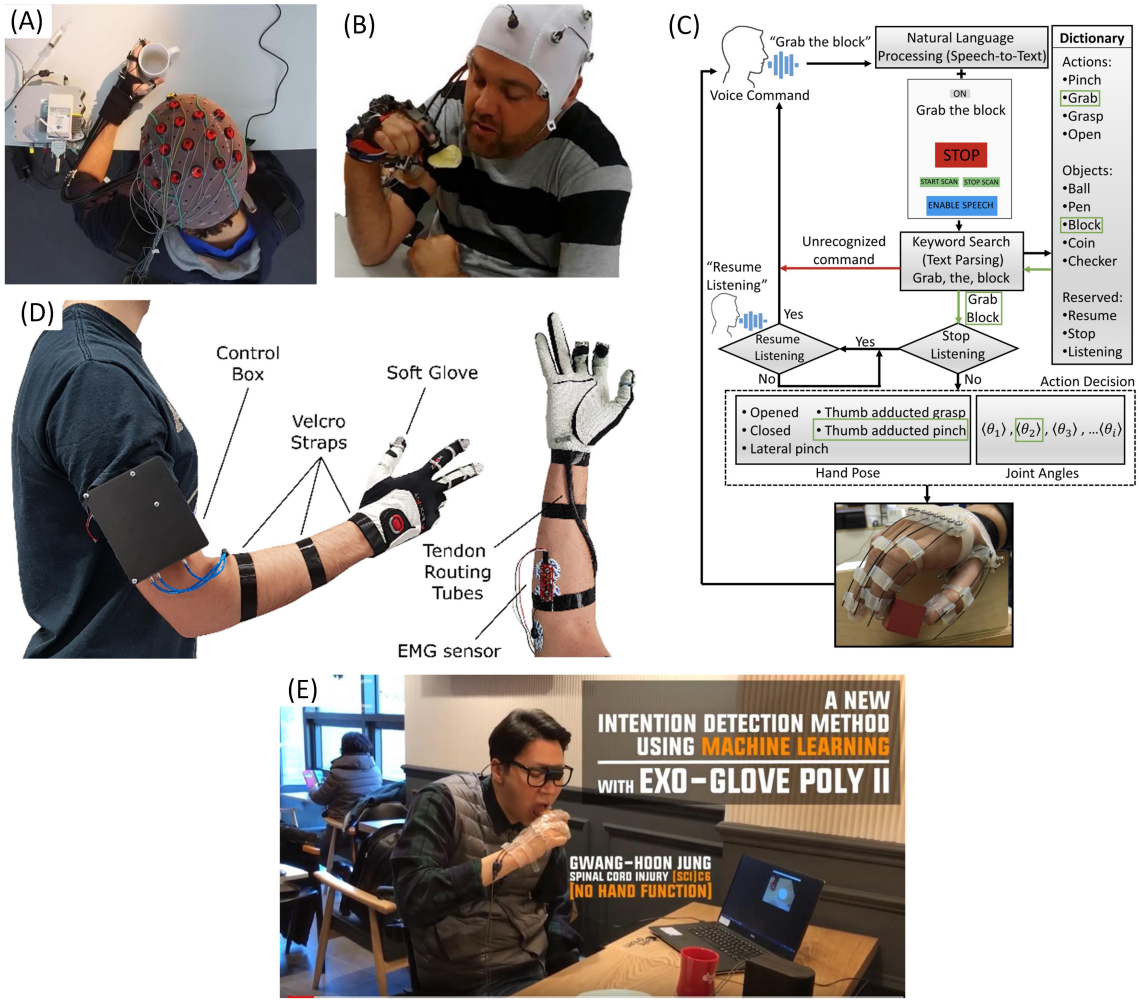


Figure 1.4: Human-Machine Interface (HMI) designed by previous researchers. (A) EEG-based HMI designed for neurorehabilitation [15]. (B) EEG-based HMI designed for patients with Quadriplegia [16]. (C) Voice-base HMI designed for patients with hand disabilities [17]. (D) EMG-based HMI designed for patients with hand disabilities [18]. (E) Vision-based HMI designed for Spinal Cord Injuries [19].

1.3 Problem Statement and Proposed Solution

Assistive exoskeleton gloves face challenges in three main areas: controlling the power of the actuator, organizing the force, and the human-machine interface. Research has been conducted to address the problems in these areas.

1.3.1 Actuator Force Control

A comprehensive analysis of the current actuator force control techniques employed in rigid exoskeleton gloves has revealed a major issue: the lack of precision caused by friction forces and unquantifiable deformations in the linkages. **To tackle this issue, this research focuses on addressing the challenge by introducing data-driven calibration and control methods. These approaches aim to enhance the precision and reliability of actuator force control in rigid exoskeleton gloves.**

1.3.2 Exoskeleton Glove Force Planning

Humans possess the capacity to gauge the grasp force through sight. When the real size, weight, and surface friction coefficient of an object are in agreement with the estimation, the anticipated force is likely to be close to the ideal grasp force. Haptic feedback is employed to identify slippage in cases where the estimated force is inaccurate, allowing humans to modify the grasp force as necessary [74].

Despite a thorough examination of existing research, no one has come up with a reliable way to calculate the weight and size of the target objects. **The aim of this study is to create a force planning system that mimics human-like abilities. To do this, a vision-based object and material detection method is used to estimate the grasp force, as well as a slip-grasp force planning method to adjust the grasp force during object manipulation.**

1.3.3 Human-Machine Interface (HMI) Design

After conducting literature review, it has been found that EMG-based HMIs are not suitable for patients with BPI. EEG-based HMIs, on the other hand, are not ideal due to their high cost and lack of portability. Vision-based HMIs heavily rely on object detection accuracy and may not provide a natural user experience. Voice-based HMIs emerge as the most suitable option for exoskeleton gloves. However, a significant issue associated with voice-based HMIs in exoskeleton gloves is the lack of speaker verification, which increases the risk of misoperation of the exoskeleton due to voice commands from individuals other than the user. **This research aims to address this challenge by introducing a speaker verification system to the voice-based HMI, making the voice-based more convenient to use.**

1.4 Dissertation Structure

A brief overview of each chapter is listed as follows:

Chapter 1: Introduces the motivation of this study, briefly reviews related research, and states the existing challenges of the state-of-the-art research. Additionally, it outlines the proposed solutions and the contribution to this research.

Chapter 2: Introduces the hardware design of an assistive exoskeleton glove.

Chapter 3: States the challenges of existing actuator force control methods and proposes a data-driven actuator force calibration and control method.

Chapter 4: States the challenges of existing force planning methods and proposes a vision-based object size and weight estimation method to bridge the gap between human-like grasping and exoskeleton grasping. Additionally, it describes performance improvements of the

existing slip-grasp force planning method by proposing a hybrid slip-grasp force planning policy.

Chapter 6: Proposes a voice-based HMI with text-independent speaker verification feature and a more advanced voice-based HMI with text-dependent speaker verification feature.

Chapter 5: Demonstrates the experimental results of the proposed prototype.

Chapter 7: Summarizes this research and proposes further improvements.

1.5 Summary of Contributions

This study contains joint work from the robotic exoskeleton glove research team in the Robotics and Mechatronics Lab at Virginia Tech. My main contributions are summarized as follows:

1: Development of a proof-of-concept robotic exoskeleton glove prototype with onboard sensing, computation, power, and control designed for BPI patients.

2: Development of machine learning based SEA fingertip force prediction methods to accurately predict force applied to each exoskeleton glove fingertip.

3: Development of an overarching force planning method which uses SEA and FSR sensors to adjust the overall grasp force to avoid slippage in real-time.

4: Development of a vision-based method using machine learning to estimate the size, weight, and surface material of objects for accurate force planning.

5: Development of a bio-authenticated voice-based human machine interface for grasp activation.

1.6 Selected Publications

Disclosure: The content of these publications is used in this dissertation.

Peer-Reviewed Journal Paper

1: Guo, Y., Xu, W., Pradhan, S., Bravo, C., & Ben-Tzvi, P. (2021). Data driven calibration and control of compact lightweight series elastic actuators for robotic exoskeleton gloves. IEEE sensors journal, 21(19), 21120-21130. * Selected as a featured article in Volume 21, Issue 19

2: Guo, Y., Xu, W., Pradhan, S., Bravo, C., & Ben-Tzvi, P. (2022). Personalized voice activated grasping system for a robotic exoskeleton glove. Mechatronics, 83, 102745.

3: Xu, W., Guo, Y., Bravo, C., & Ben-Tzvi, P. (2022). Design, Control, and Experimental Evaluation of a Novel Robotic Glove System for Patients With Brachial Plexus Injuries. IEEE Transactions on Robotics, 39(2), 1637-1652.

4: Guo, Y., Xu, W., Pradhan, S., Bravo, C., & Ben-Tzvi, P. (2023). Vision Based Human Machine Interface for an Assistive Robotic Exoskeleton Glove. Autonomous Robots, Submitted.

5: Xu, W., Guo, Y., & Ben-Tzvi, P.(2023). Development of a Novel Compact Exoskeleton Glove with Reinforcement Learning Control. ASME Journal of Mechanisms and Robotics, Submitted.

Peer-Reviewed Conference Paper

1: Xu, W., Guo, Y., Bravo, C., & Ben-Tzvi, P. (2022, October). Development and Experimental Evaluation of a Novel Portable Haptic Robotic Exoskeleton Glove System for Patients with Brachial Plexus Injuries. In 2022 IEEE/RSJ International Conference on Intelligent

Robots and Systems (IROS) (pp. 11115-11120).

2: Xu, W., Guo, Y., & Ben-Tzvi, P. (2023, Aug), Robotic Exoskeleton Glove System Design and Simulation for Patients with Brachial Plexus Injuries. In 2023 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC-CIE 2023).

3: Guo, Y., Xu, W., & Ben-Tzvi, P. (2024), Controlled Human-Machine Interface for an Assistive Exoskeleton Glove Aiding Patients with Brachial Plexus Injuries. In 2024 IEEE International Conference on Robotics and Automation (ICRA 2024), Pending Submission.

Patent

1: Guo, Y., Ben-Tzvi, P., Refour, E., Sebastian, B., Xu, W., & Pradhan, S. (2020, June), Robotic Exoskeleton Glove System. U.S. Patent Application No. 16/888,993. (to be issued)

Chapter 2

Exoskeleton Glove Hardware Introduction

According to the literature review in Sec. 1.2.1, the rigid linkage-driven exoskeleton glove appears to be more desirable for patients with BPI. However, researchers face challenges in balancing the size, weight, and force output of the glove. In this chapter, a prototype is proposed to address the aforementioned challenges. This prototype contains the joint work of my research team. For more details, see [23].

The designed prototype features 7 Degrees Of Freedom (DOF), allowing for finger extension, contraction, thumb rotation, and wrist bending. Additionally, passive mechanisms are employed for finger abduction and adduction. The proposed exoskeleton glove can perform five different types of grasp as shown in Fig. 2.1. The mechanical design of the prototype incorporates five SEAs with aluminum side-mounted linkages (shown in Fig. 2.2). This combination optimizes the force output while minimizing the size and weight of the linkages [23].



Figure 2.1: Five different types of grasp can be performed by the proposed exoskeleton glove. (A) Cylinder grasp. (B) Sphere grasp. (C) Tripod grasp. (D) Tip grasp. (E) Lateral grasp.

2.1 Mechanical Design

2.1.1 Exoskeleton Glove Linkage Optimization

The side-mounted LSEA linkage used in this design is first proposed by Vanteddu et al. [46] and redesigned by Xu et al. [64]. The linkage design is shown in Fig. 2.3.

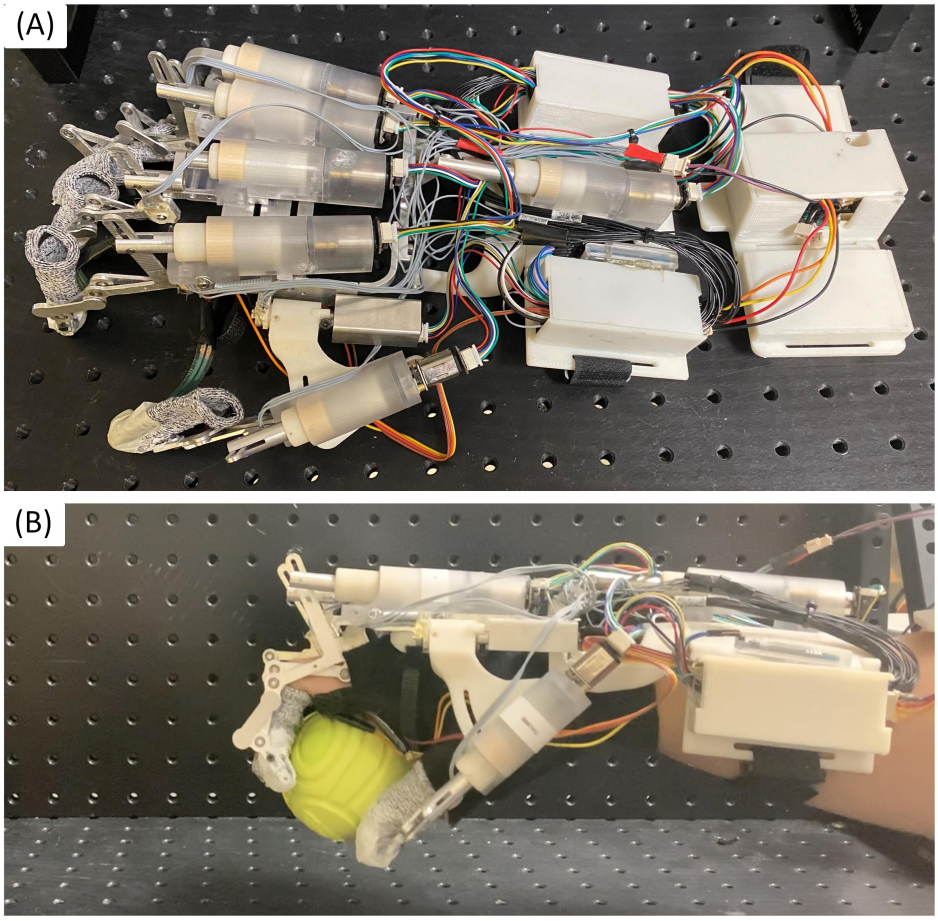


Figure 2.2: Exoskeleton glove prototype. (A) Top view. (B) Side view.

This design contains three main linkages: l_P, l_I, and l_D. These three linkages correspond to the three bones of the human finger. Similarly, θ_1, θ_2, and θ_3 correspond to the Metacarpophalangeal (MCP), Proximal Inter-Phalangeal (PIP), and Distal Interphalangeal (DIP) joints of the human finger. There are also two constraint linkages c_1 and c_2 to limit the motion of these linkages. c_1 and c_2 are shown in Eq. 2.1 and Eq. 2.2,

$$c_1^2 = [l_P \cos\theta_1 - d_I \sin(\alpha_I - \theta_2) - d_B \sin\alpha_B]^2 + [l_P \sin\theta_1 - d_I \cos(\alpha_I - \theta_2) - d_B \cos\alpha_B]^2 \tag{2.1}$$

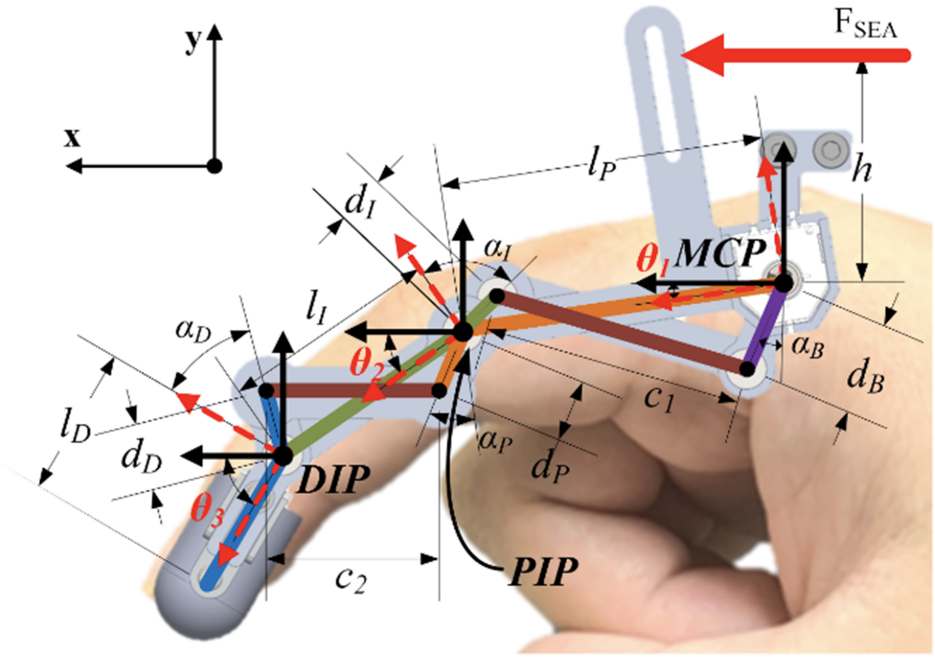


Figure 2.3: Index finger LSEA linkage design.

$$\begin{aligned} c_2^2 =& [l_I \cos\theta_2 + d_D \sin(\theta_3 - \alpha_D) - d_P \sin(\alpha_P - \theta_1)]^2 + [l_I \sin\theta_2 - d_D \cos(\theta_3 - \alpha_D) \\ & - d_P \cos(\alpha_P - \theta_1)]^2 \end{aligned} \tag{2.2}$$

where, c_1 and c_2 are the lengths of the constraint links. d_B, d_P, d_I and d_D are the lengths of the corresponding linkages (shown in Fig. 2.3). α_B, α_P, α_I, and α_D are the angles between the linkages and the y-axis. θ_1 is driven by the actuator directly, given a particular θ_1, θ_2 and θ_3 can be derived based on the above constraint equations. θ_2 and θ_3 can be expressed as Eq. 2.3 and Eq. 2.4.

$$\theta_2 = -\tan^{-1}\left\{\frac{\sqrt{-[A+(c_1-d_I)^2][A+(c_1+d_I)^2]}}{-c_1^2-A+B+d_I^2}- \frac{2d_I l_P\cos(\theta_1)-2d_B d_I\sin(\alpha_B)}{-c_1^2-A+B+d_I^2}\right\}+\alpha_I \tag{2.3}$$

where, $A = 2d_B l_P \sin\alpha_B + \theta_1 - l_P^2 - d_B^2$, $B = 2d_I l_D \sin\theta_1 - 2d_B d_I \cos\alpha_B$.

$$\theta_3 = \tan^{-1}\left\{\frac{\sqrt{-[C+(c_2-d_D)^2][C+(c_2+d_D)^2]}}{-c_2^2-C+D+d_D^2} -\frac{2d_D l_I\cos(\theta_2)-2d_P d_D\sin(\alpha_P-\theta_1)}{-c_2^2-C+D+d_D^2}\right\}+\alpha_D \tag{2.4}$$

where, $C = 2d_P l_I \sin\alpha_P - \theta_1 + \theta_2 - l_I^2 - d_P^2$, and $D = -2d_D d_P\cos(\alpha_P-\theta_1) + 2d_D l_I \sin\theta_2$. My teammates derived the above kinematics equations, and more details are included in our publication [23].

The linkage motion must perfectly mimic the human motion to provide good wearability. HUST dataset [81], which include joint angles of human hand while executing the grasping tasks of Feix taxonom, is used as a reference to optimize the length of each linkage. The motion of the human finger during grasping is plotted in multicolor lines in Fig. 2.4 (A) and (B).

The joint position of MCP is set to $(0,0)$ as shown in Fig. 2.4. The PIP joint position is calculated by Eq. 2.5. The DIP joint position is calculated by Eq. 2.6. The fingertip position is calculated by Eq. 2.7.

$$P_{PIP} = (l_P\cos\theta_P, -l_P\sin\theta_P) \tag{2.5}$$

$$P_{DIP} = (l_P\cos\theta_P + l_I\cos\theta_I, -l_P\sin\theta_P - l_I\sin\theta_I) \tag{2.6}$$

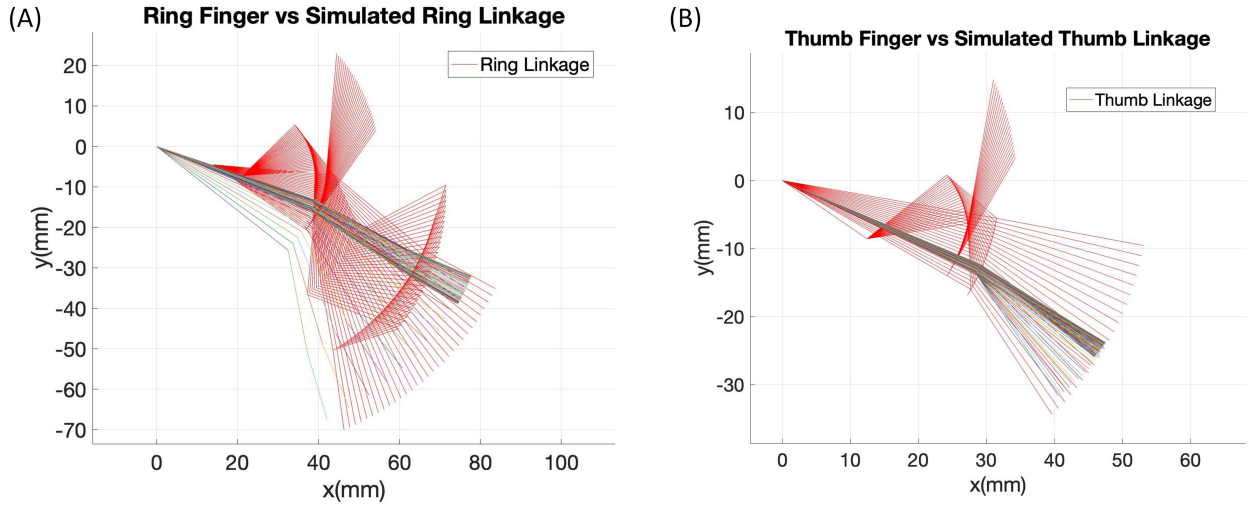


Figure 2.4: LSEA linkage optimization results. The red line represents the optimized linkages motion. Multi-color lines represent human finger motion. (A) Ring finger optimization result. (B) Thumb finger optimization result.

$$P_{TIP} = (l_P \cos\theta_P + l_I \cos\theta_I + l_D \cos\theta_D, -l_P \sin\theta_P - l_I \sin\theta_I - l_D \sin\theta_D) \tag{2.7}$$

where, θ_P, θ_I, and θ_D are human MCP, PIP, and DIP joint angle. Similarly, l_P, l_I, and l_D are the finger length corresponding to the length of bones in human finger.

During the linkage optimization process, the length of the exoskeleton l_P, l_I, and l_D links is selected based on the average length of the human finger. The rest of the linkages are optimized by minimizing the weighted Euclidean distance of the DIP joint position and the fingertip position between the human finger and exoskeleton linkages. The joint position of the exoskeleton link P_{EPIP}, P_{EDIP}, and P_{ETIP} can be calculated using Eq. 2.5, Eq. 2.6 and Eq. 2.7, Where θ_1 is equal to the given human MCP joint angle θ_P. θ_2 and θ_3 are calculated on the basis of the linkage length using Eq. 2.3 and Eq. 2.4. The cost function is shown in Eq. 2.8.

$$C(\theta_P, \theta_I, \theta_D) = w_1 d(P_{DIP}, P_{EDIP}) + w_2 d(P_{TIP}, P_{ETIP}) \tag{2.8}$$

where, θ_P, θ_I, and θ_D are given in the HUST dataset. w_1 and w_2 are the weights. d is the Euclidean distance function.

The optimization calculation is performed using the built-in optimization function of MATLAB. The optimization result is shown in Fig. 2.4. The optimization exoskeleton links have a trajectory similar to that of a human finger.

2.1.2 Linear Series Elastic Actuator (LSEA) Design

The LSEA design of the exoskeleton glove is shown in Fig. 2.5. LSEAs provide extension and contraction motion to the exoskeleton finger linkages. A linear spring is added between the input shaft and the output shaft of the linear actuator. The output force of the LSEA is described in Eq. 2.9.

$$F_{SEA} = k\Delta s = k\left[\frac{l\Delta ticks}{N_{cpr}\, i} - h(\tan\theta - \tan\theta_i)\right] \tag{2.9}$$

where, k is the spring constant. l is the lead of the leadscrew. i is the motor gearbox reduction rate. h is the height from the LSEA output shaft to the MCP joint. $ticks$ are the encoder ticks measured by the magnetic encoder mounted at the end of each LSEA motor. N_{cpr} is the counts per revolution of the encoder. θ_i is the angle measured by the angular potentiometer mounted on the MCP joint when the SEA input shaft is at its initial position. θ is the current MCP joint angle.

In addition to its capacity for contraction and extension motion, the LSEA base also incorporates a passive abduction-adduction mechanism. This mechanism enables the LSEA to move along with the user's finger during grasping, maximizing comfort during operation.

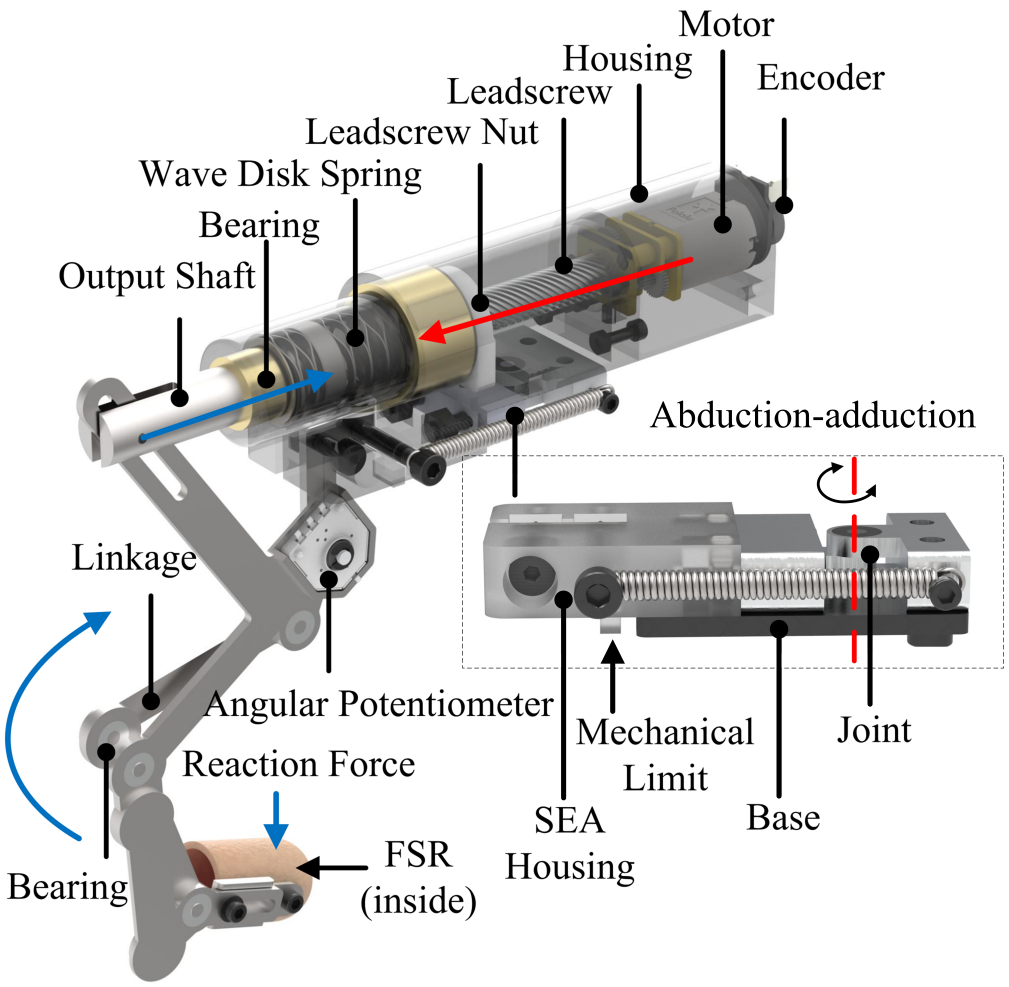


Figure 2.5: Actuator, linkage, and abduction-adduction mechanism design of the LSEA.

2.1.3 Rotary Series Elastic Actuator (RSEA) Design

The design of the RSEA is depicted in Fig. 2.6. It uses a torsion spring as the elastic material (as shown in Fig. 2.6). To eliminate backlash, the torsion spring is preloaded. The RSEA incorporates an aluminum input shaft and a combined plastic-aluminum output shaft to enhance its structural integrity, allowing it to generate forces of up to 15N. Friction is minimized by implementing ball bearings between the RSEA shaft and the housing (shown in Fig. 2.6). A mechanical limit is incorporated to prevent back-driving of the SEA, enabling it to support the weight of the user's thumb. The mechanical limit is highlighted in the red

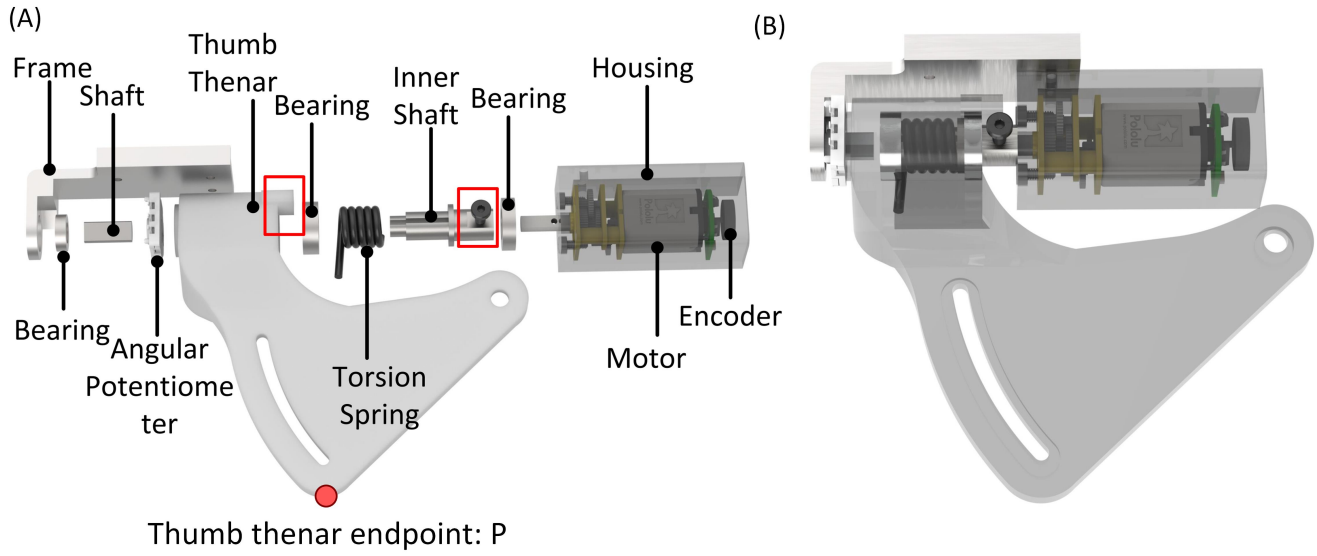


Figure 2.6: RSEA design. (A) Exploded view of RSEA. The red box shows the mechanical limit to prevent SEA from back-drive. (B) RSEA side view.

box in Fig. 2.6. The output force of the RSEA can be calculated using Eq. 2.10.

$$F = K(\theta - \frac{\Delta ticks}{360 N_{cpr} i}) \tag{2.10}$$

where, F is the output force on the exoskeleton thumb thenar end point P, shown in Fig. 2.6 (A). K is the measured spring constant. θ is the RSEA output shaft angle measured using a potentiometer. $ticks$ are the encoder ticks measured by the magnetic encoder mounted at the end of each SEA motor shown in Fig. 2.6 (A). N_{cpr} is the count per revolution of the encoder.

2.2 Electrical Design

The proposed exoskeleton glove features a modular and portable electronic design, with all electronic components integrated within the glove itself. This modular design allows for

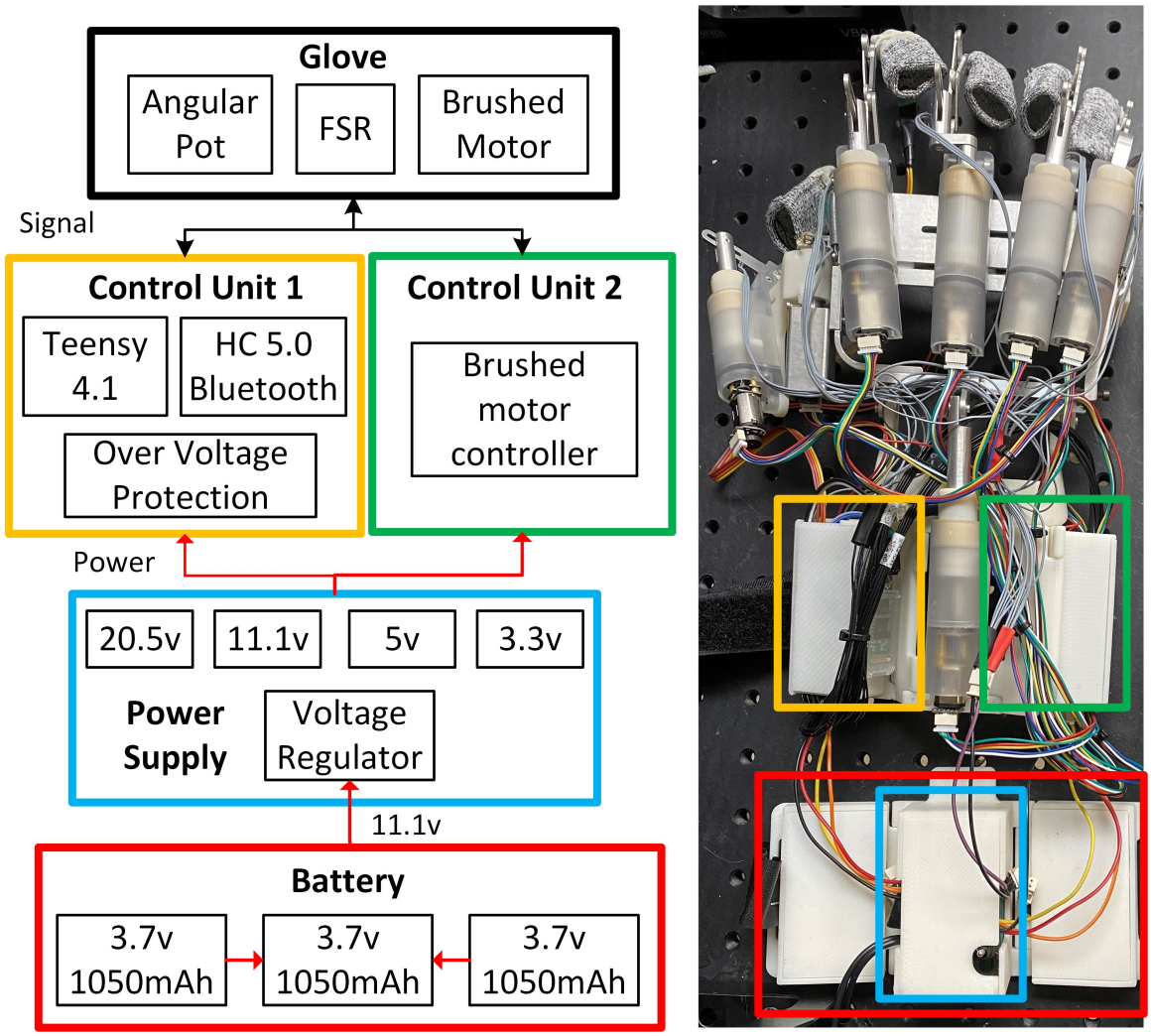


Figure 2.7: Electronics design overview. Yellow: microcontroller board. Green: brushed motor controller board. Blue: Power conversion board. Red: 3-in-1 battery package.

easy replacement of individual parts and eliminates the need for external electrical wiring to control the exoskeleton glove. The electronics block diagram is presented in Fig. 2.7. The electronics are divided into three parts, all mounted on the user's forearm.

The first part comprises a battery box that houses three parallel-connected 3.7V 1050mAh batteries, along with a power conversion unit that provides voltage outputs of 3.3V, 5V, 11.1V, and 20.5V.

The second part consists of a two-piece control unit. The left Printed Circuit Board (PCB)

incorporates a Teensy 4.1 microcontroller and an HC-05 Bluetooth unit, while the right PCB contains DVR8801 motor drivers. The Teensy 4.1 microcontroller handles sensor readings and performs low-level control tasks. FreeRTOS is used on the Teensy 4.1 microcontroller to enable multi-threading. The usage of real-time system ensures that all the computation is finished in its designed time interval. The microcontroller receives commands from a computer via a Bluetooth connection.

The third part of the exoskeleton glove employs five dual-shaft 12V 380:1 Pololu metal gear motors, providing sufficient power and torque for each finger's LSEA. Additionally, two dual-shaft 12V 1000:1 Pololu metal gear motors are utilized in the LSEA and RSEA for the thumb thenar and wrist movements. Each motor is paired with a DVR-8801 motor driver and a 12-Count Per Revolution (CPR) magnetic encoder. Seven Bourns 3382G angular potentiometers are used to calculate the spring compression for each SEA, with a 20.5V voltage supplied to maximize their resolution. Furthermore, two Interlink FSR-402 force-sensitive resistors (FSRs) are placed on wearable finger sleeves, specifically on the thumb and index finger.

Chapter 3

Exoskeleton Glove Series Elastic Actuator (LSEA) Force Control

The importance of actuator control in ensuring the accurate transmission of the desired force to the exoskeleton fingertip linkage is undeniable. This chapter described the development process of data-driven SEA fingertip force prediction methods that use experimental data to create a physical model for the actuators. Compared to traditional kinematics-based force calculation methods, which often experience significant errors during LSEA calibration, the data-driven approach is more effective when considering parameters such as linkage deformation and friction force. Accurate force prediction will greatly assist in force control of the exoskeleton glove. This chapter first discusses the drawbacks of traditional kinematics-based LSEA calibration and then presents data-driven methods to achieve more precise force calibration and control.

3.1 LSEA Force Modeling

This section explains the traditional LSEA force modeling approach through kinematic calculations. Fig. 3.1 displays the index finger LSEA. When the SEA is stationary, the force can be determined using Eq. 3.1. The encoder can measure the distance between DE and the potentiometer can measure $\angle ABC$ (MCP joint). CB and CE are fixed and can be

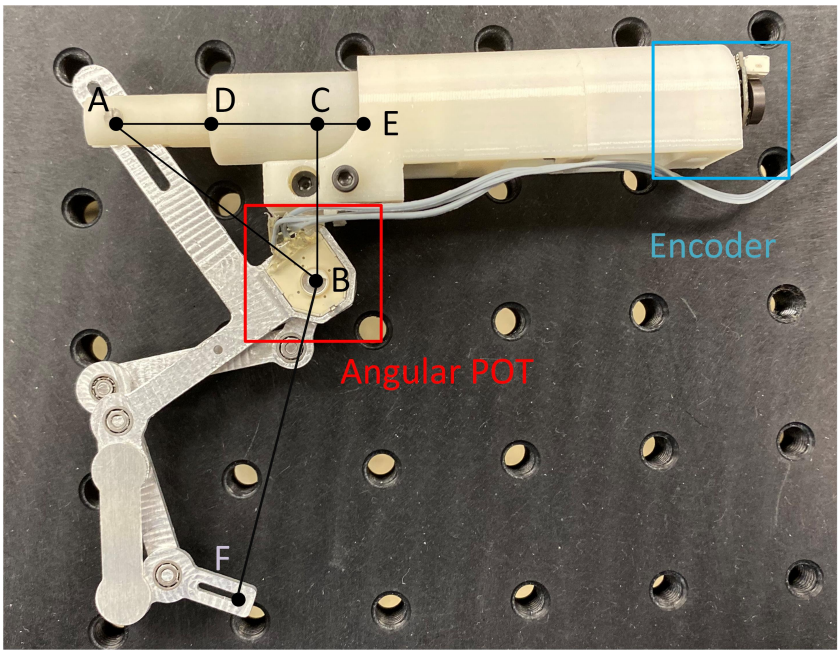


Figure 3.1: The legend for the calculation of the SEA force is as follows: Point A is the endpoint of the output shaft of the LSEA. Point B is the MCP joint. Points C and D are located on the input shaft of the LSEA. Point E is situated on the LSEA housing, which is stationary. Point F is the fingertip endpoint of the LSEA linkage. The 12-CPR magnetic encoder, which is marked in blue, is used to measure the length of DE. The rotary potentiometer, which is marked in red, is used to measure the angle of the MCP joint.

directly measured. δl_s can be calculated using Eq. 3.2. F_{fa} and F_{fl} can be adjusted with a constant value during force calibration. R_l is the LSEA output shaft to fingertip linkage force transformation ratio. The linkage transformation ratio changes as the MCP angle changes, detail is discussed in the following subsection.

$$F_t = (\delta l_s \times k_s - F_{fa}) \times R_l - F_{fl} \tag{3.1}$$

$$\delta l_s = AD_i - (tan(\angle ABC) \times CB - (DE - CE)) \tag{3.2}$$

where,

F_t – Force output on the fingertip linkage, measured in direction that is perpendicular to the last linkage

δl_s – LSEA spring compression

F_{fa} – Friction force between the actuator shafts and actuator housing

R_l – LSEA output shaft to fingertip linkage force transformation ratio

l – Uncompressed spring length

F_{fl} – Friction force of the linkage joints

AD_i – The initial length of AD when no force is applied on the SEA

3.1.1 Rotatory Encoder Setup

The 12 Count Per Revolution (CPR) Pololu quadrature magnetic encoder is employed to measure the displacement of the LSEA input shaft. This encoder is connected to a 12V Pololu micro-gear motor that is coupled with a 380:1 gearbox. The lead screw has a ratio of 20:1 between the distance (measured in millimeter) and revolution. The displacement of the actuator input shaft δDE can be determined using Eq. 3.3.

$$\begin{aligned} DE &= \frac{\frac{R_{ld}}{R_g}}{N_{cpr}} \times N_{enc} = \frac{\frac{20}{380}}{12} \times N_{enc} \\ &\simeq 0.00438596 \times N_{enc} \end{aligned} \tag{3.3}$$

where,

DE – Displacement of DE measured in millimeters

R_{ld} – Distance to revolution ratio of the lead screw, 20mm : 1rev

R_g – Output shaft to input shaft radio of gearbox, 380:1

N_{cpr} – Count per revolution of encoder, 12

N_{enc} – Encoder count

3.1.2 Angular Potentiometer Setup

The angular potentiometer is employed to measure the angle of $\angle ABC$ (the MCP joint), which is then used to calculate the displacement of the LSEA output shaft with the help of trigonometry. This potentiometer has a range of 15 to 345 degrees and is connected to a 3.3V 12bit ADC bus. To improve the resolution, the voltage of the potentiometer is increased to 20.5V. This voltage increase allows for the effective range to be calculated using Eq. 3.4. To protect the microcontroller from overvoltage damage, a 20 degree mechanical offset is introduced. The angle of $\angle ABC$ is calculated by Eq. 3.5.

$$ER_{pot} = \frac{V_{adc}}{V_s} \times R_{pot} = \frac{3.3}{20.5} \times 330^\circ \simeq 53.12195^\circ \tag{3.4}$$

$$\angle ABC = \frac{ER_{pot}}{Re_{adc}} \times Pot - \angle O + \angle S \simeq \frac{53.12195^\circ}{2^{12}} \times Pot - 20^\circ + 15^\circ \simeq 0.012969 \times Pot - 5^\circ \tag{3.5}$$

where,

ER_{pot} – Effective measuring range of potentiometer

V_{adc} – ADC bus reference voltage, 3.3V

V_s – Source voltage of potentiometer, 20.5V

R_{pot} – Measuring range of potentiometer, 330 degrees

Re_{adc} – ADC resolution, 12bit, 4096

Pot – Raw potentiometer adc reading

$\angle O$ – Potentiometer offset angle, 20 degrees

$\angle S$ – Potentiometer start angle, 15 degrees

3.1.3 Force Measurement at LSEA Fingertip Linkage

It is essential to measure the force at the LSEA fingertip linkage in order to comprehend the actual output force from the LSEA. This is usually done with a load cell. However, it can be difficult to place the load cell perpendicularly to the last linkage when measuring force directly from the linkage end. Therefore, in this research, two different load cells were placed at different locations to accommodate the linkage at different angles, as illustrated in Fig. 3.2. The force measured at the linkage end was then transformed back to the actuator output shaft through inverse kinematics.

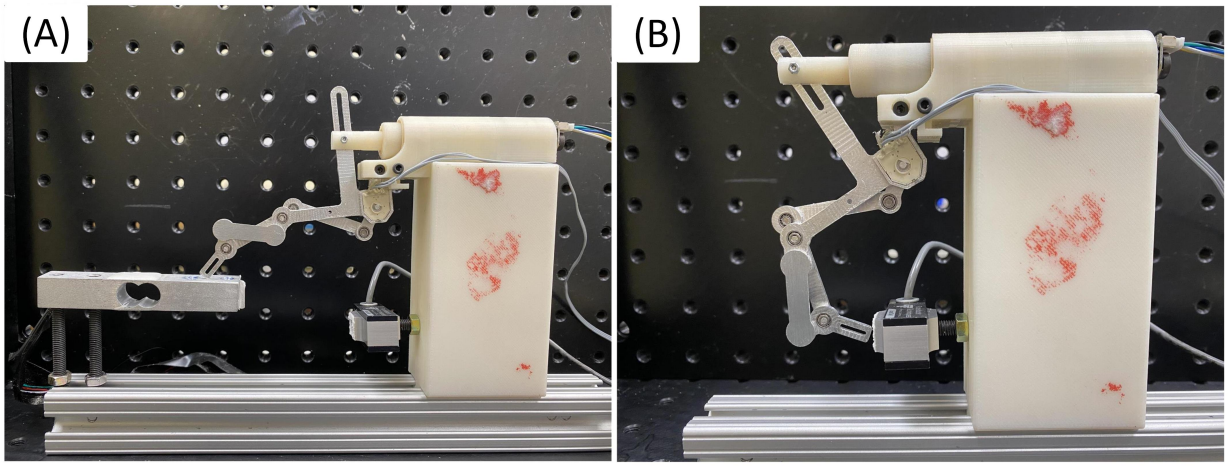


Figure 3.2: Fingertip linkage force measurement setup for LSEA. (A) Horizontally mounted load cell used for LSEA fingertip linkage force measurement when the actuator is not fully extended. (B) Vertical mounted load cell used for LSEA fingertip linkage force measurement when the actuator is fully extended.

3.1.4 Linkage Force Transformation Ratio

The Linkage Force Transformation Ratio (R_l) transmits the force from the LSEA output shaft to the perpendicular director of the LSEA fingertip linkage. The load cell is mounted in two different positions to accommodate different angles of the $\angle ABC$ (MCP joint), resulting in two different equations for R_l based on the position of the load cell. The relationship between angle R_l is illustrated in Fig. 3.3. The Linkage Force Transformation Ratio used to transfer the force associated with the small angle of the MCP joint is expressed as a fifth-order equation (H_{fit}). The Linkage Force Transformation Ratio used to transfer the force associated with the large angle of the MCP joint is expressed as a second-order equation (V_{fit}). The combined equation is given in Eq. 3.6.

$$R_l = \begin{cases} H_{fit} & \text{if } 5^\circ \leq \angle ABC \leq 16^\circ \\ V_{fit} & \text{if } 16^\circ < \angle ABC \leq 28^\circ \end{cases} \tag{3.6}$$

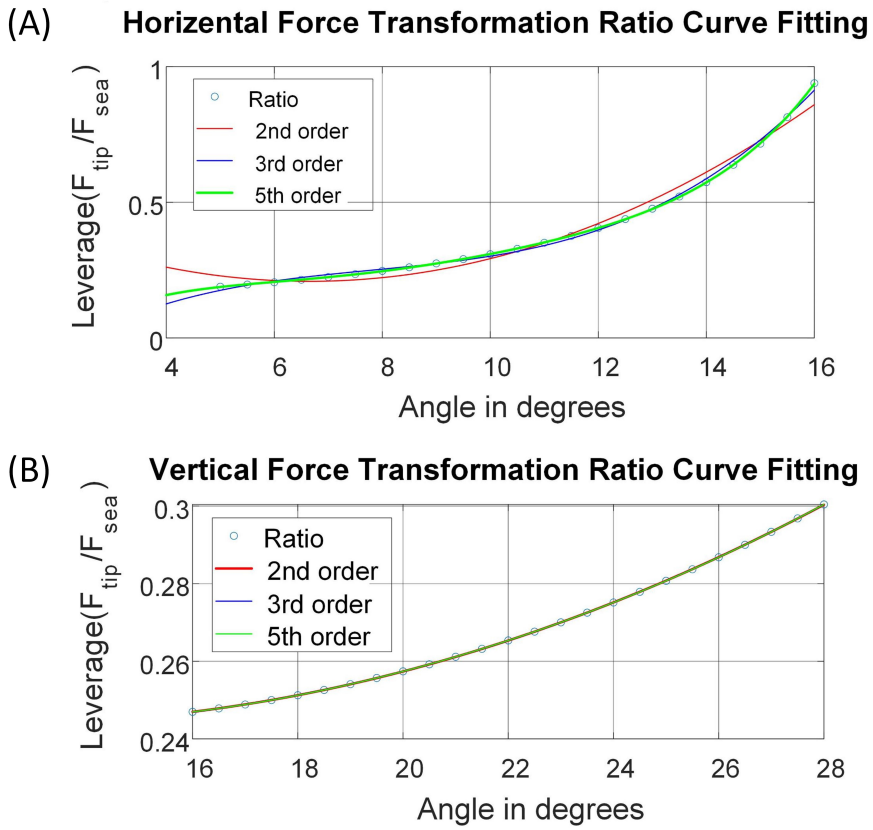


Figure 3.3: LSEA actuator output shaft to fingertip linkage force transformation ratio. (A) Index finger LSEA linkage force transformation ratio when $\angle ABC$ is between 5 degree and 16 degree. (B) Index finger LSEA linkage force transformation ratio when $\angle ABC$ is between 16 degree and 28 degree.

3.2 Challenges of LSEA Force Calibration

This section discussed the difficulties encountered in traditional LSEA force calibration, leading to a discrepancy between the desired and measured forces. The force output of the SEA can be easily calculated using the equations derived in the preceding sections. The force calibration result is quantified by comparing the calculated force with the force measured by a load cell. The measured force and the calculated force should be comparable, but when testing the LSEA, the two forces have a discrepancy of up to 45%. Two major issues that prior research has yet to tackle are friction and deformation, which will be discussed in the

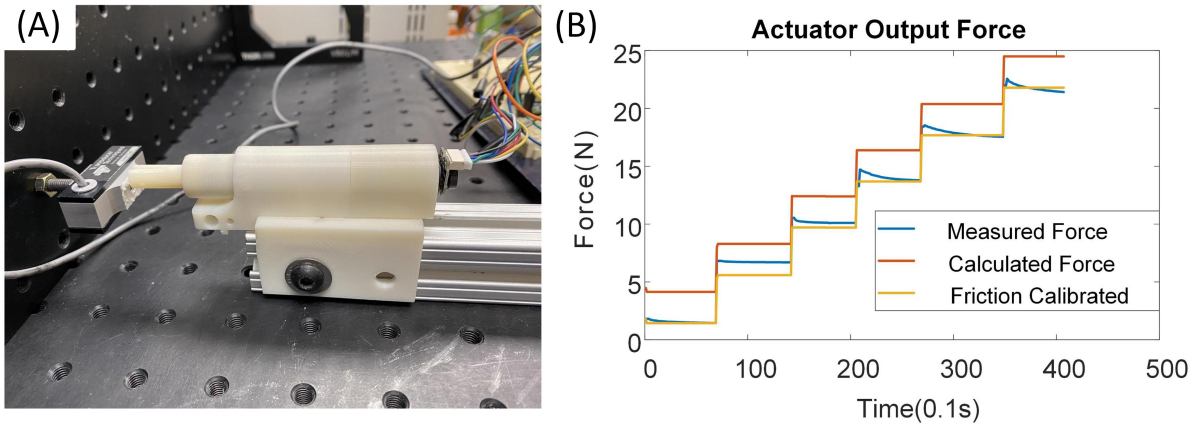


Figure 3.4: Difference of the actuator output force and measured force due to friction force. (A) Force output measurement setup. (B) A comparison was made between the calculated (ideal), measured, and calibrated output force. The calibrated value was expected to be the same as the measured value, however, it did not match.

following subsections.

3.2.1 Friction Force Between Actuator Shafts and Housing

This section investigates the influence of the friction force between the LSEA housing and the shafts. An experiment was conducted with the setup shown in Fig. 3.4 (A). The ideal output force was expected to be equal to the calculated force (illustrated by the blue and red lines in Fig. 3.4 (B)). However, the experimental result in Fig. 3.4 (B) showed that the measured force was generally lower than the calculated force due to the considerable amount of friction force between the LSEA shafts and housing that affects the output force.

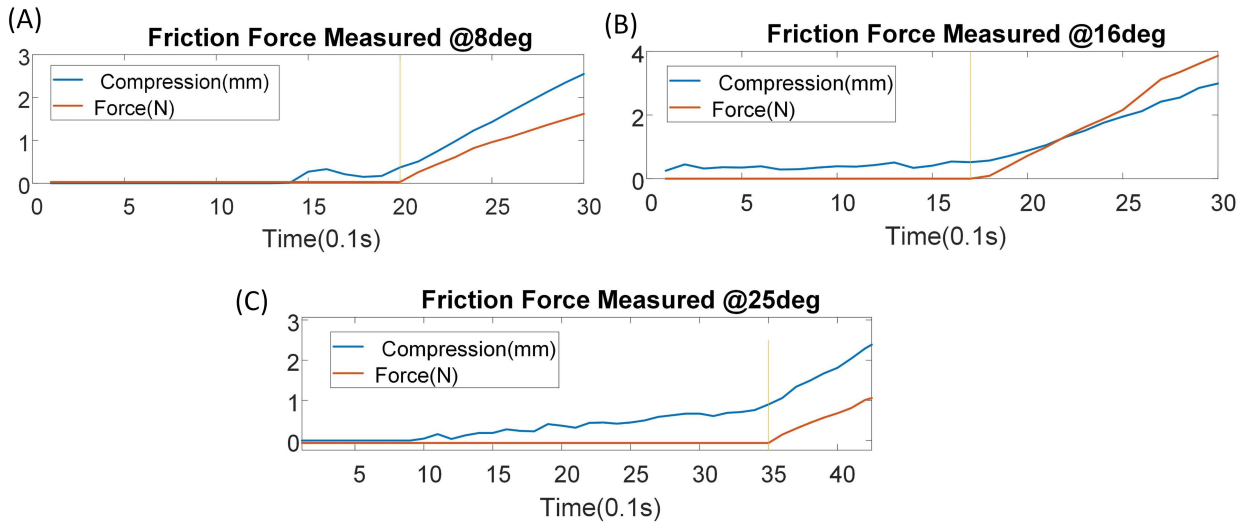


Figure 3.5: Different index finger linkage angle have different friction force. (A) The measured friction force at 8 degree is 2.1N (measured by 0.51mm spring compression). (B) The measured friction force at 16 degree is is 2.4N (measured by 0.57mm spring compression). (C) The measured friction force at 25 degree is is 4.45N (measured by 1.06mm spring compression).

3.2.2 The Combined Friction Force of the Linkage Joints and the Actuator Shafts

This section examines the influence of friction force on the LSEA linkage and the actuator Shafts. In this experiment, the SEA with the linkage is actuated against a load cell at different angles. The compression of the SEA will increase, but the load cell reading will stay at zero, indicating that the spring is compressed due to friction force. The load cell will start to register readings when the applied force is greater than the friction force. The compression measured when the load cell has readings represents the amount of friction force of the LSEA. Fig. 3.5 shows that the friction force varies at different angles.

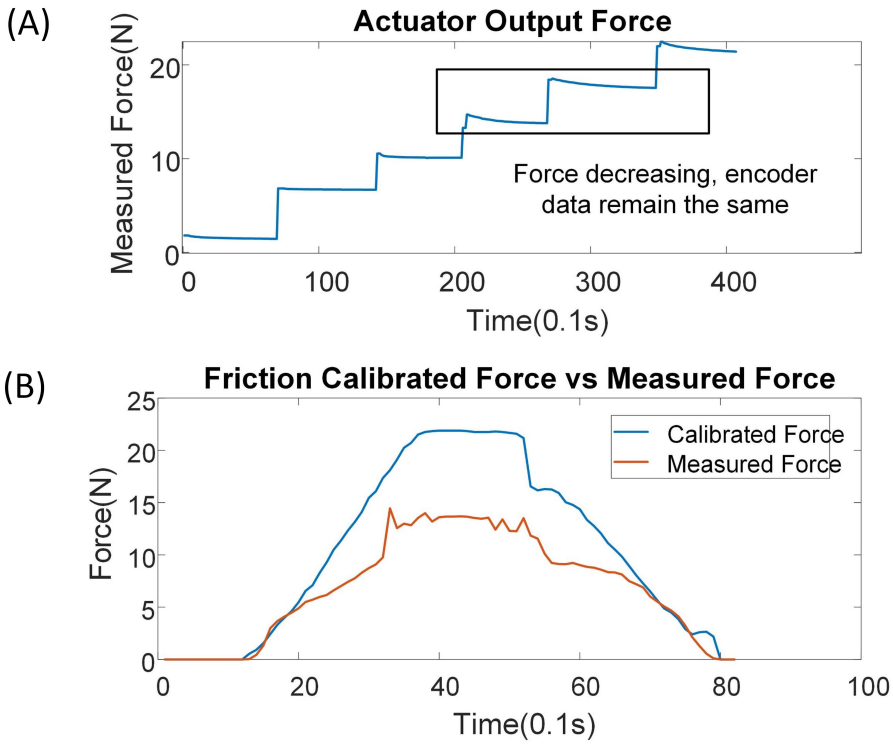


Figure 3.6: Force measurement error caused by deformation. (A) Inconsistency in force measurement caused by actuator shaft deformation. (B) Force measurement error caused by linkage deformation.

3.2.3 Deformation of the Actuator Shafts and Linkages

This section illustrates that the LSEA output force is affected by the deformation of the actuator shafts and linkages. Two experiments were conducted to demonstrate how deformation can influence force calibration.

The experiment depicted in Fig. 3.4 (A) was the first to be carried out. The force that was measured, as seen in Fig. 3.6 (A), does not perfectly match the calibrated force. This indicates the LSEA output shaft is deformed, causing the error in force measurement.

The second experiment is setup as shown in Fig. 3.2. This experiment compared the desired fingertip output force with the measured fingertip output force, as illustrated in Fig. 3.6 (B). The desired force was calibrated to account for the friction force, yet there was a 48%

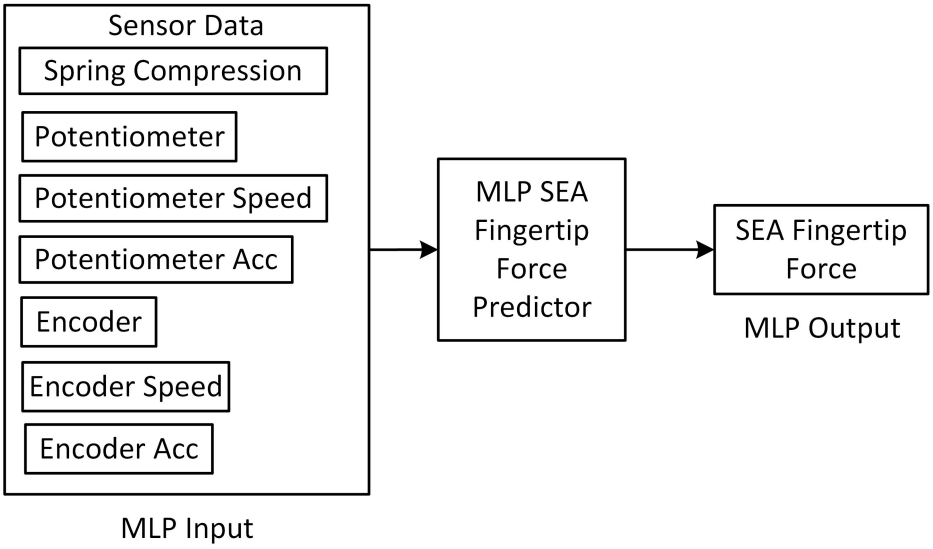


Figure 3.7: The Multi-Layer Perceptron (MLP) LSEA fingertip force prediction method.

discrepancy between the measured and calibrated force. Additionally, Fig. 3.6 (B) demonstrated that the difference increases as the force increases. This experiment indicates that the deformation of the LSEA linkages can cause errors in the force measurement proccess.

3.3 LSEA Force Calibration using Multi-Layer Perceptron (MLP) Method

The Kinematics-based SEA fingertip force calibration accuracy is impaired by the presence of the hard-to-measure friction force and immeasurable deformation, leading to significant force calibration error. To address this issue, the linkage and LSEA actuator must be considered as a single system and calibrated using a data-driven approach in which a machine learning model is used to take the sensor input and predict the output force of the fingertip linkage. The kinematics calculation can be replaced by a Multi-Layer Perceptron (MLP) model (illustrated in Fig. 3.7), which predicts the output force based on the sensor readings.

The MLP network has seven inputs, including calculated spring compression, filtered potentiometer reading, raw encoder reading, and speed and acceleration of raw encoder reading and filtered potentiometer reading. Speed and acceleration are selected due to the friction force having a different direction when the system is in motion, and this direction may vary depending on speed and acceleration. The sensor inputs do not contribute equally to the force output, and thus the sensor inputs are weighted. The weighted MLP methods are used for force calibration.

3.3.1 Network Tuning and Lost Function Selection

The performance of the MLP model can be greatly improved through network tuning and loss function optimization. Hyperparameter tuning is used to adjust the calibration network, which involves finding the optimal layer, number of nodes, and combination of cost functions.

Two cost functions are compared for the MLP method: the Mean Absolute Error (MAE) and the Mean Squared Error (MSE). MAE (Eq. 3.7) measures the absolute difference between the prediction and the measured value, in this case the difference in force. MSE (Eq. 3.8) measures the square of the difference between the prediction and the measured value, and is more punishing of outlier data than MAE.

$$E = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y_j}| \tag{3.7}$$

$$E = \frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2 \tag{3.8}$$

3.4 LSEA Force Calibration using Surface Fitting (SF) Method

The Surface Fitting (SF) method is a low computational cost approach that is often used as a comparison baseline for the MLP method. This method requires minimal computational cost, making it suitable for calibrating multiple SEAs on a single microcontroller. The inputs for this approach are angle and compression, and the output is the force. One downside is that it does not take into account the direction of the friction force. Compared to the MLP approach, this method requires less computational power due to its fewer inputs and the use of third-order linear regression. However, this efficiency may come at the cost of lower accuracy.

3.5 LSEA Force Control using Multi-Layer Perceptron (MLP) Method

The LSEA is controlled by a PID spring compression controller running on a Teensy 4.1 micronctoller. The microcontroller samples sensor signals locally and transmits the sampled signals via Bluetooth to a computer to process the data through the MLP predictor. The output of the MLP predictor is the desired spring compression, which is sent back to the microcontroller to be used as the PID spring compression controller setpoint. The software architecture is illustrated in Fig. 3.8.

This structure was chosen because the MLP is computationally complex to run on Teensy 4.1 as there are 7 SEAs and each has its own MLP predictor. The MLP predictor and the PID compression controller are running at 10Hz to accommodate the restrictions imposed by the

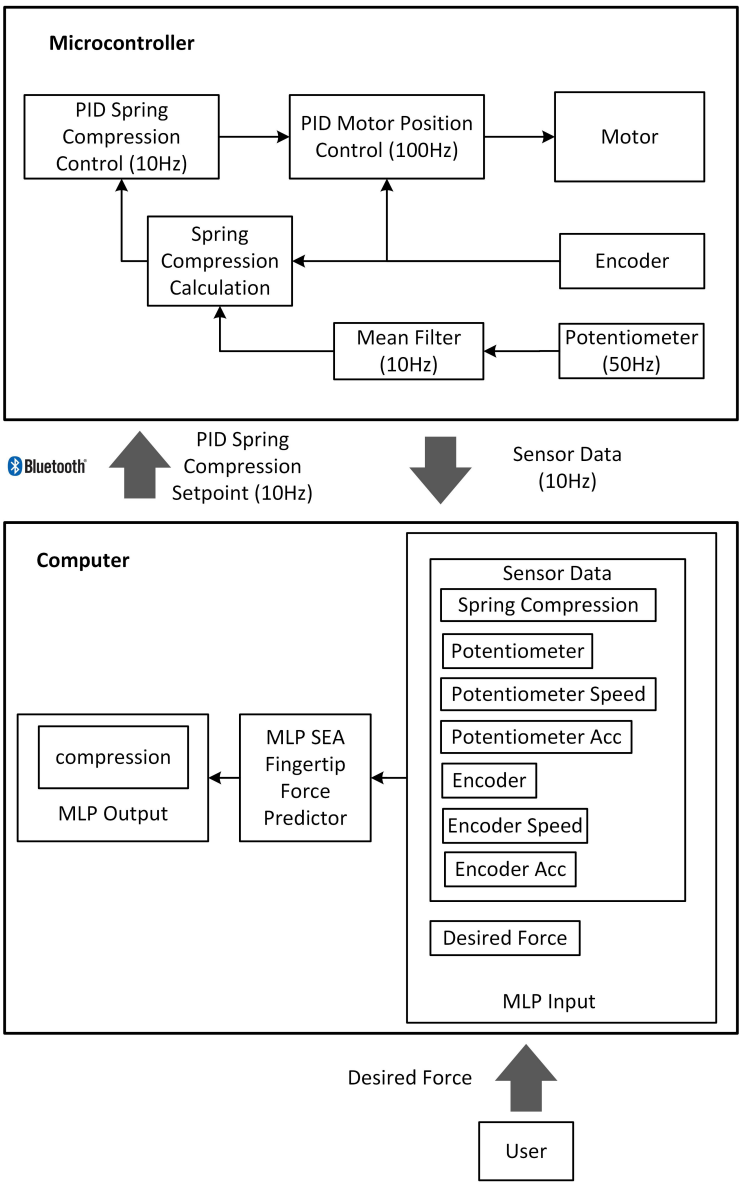


Figure 3.8: SEA control software architecture using MLP method. The low-level control is running on a microcontroller. The desired compression is fed into the microcontroller by the high-level MLP SEA force predictor through cloud computing.

wireless Bluetooth connection. The control system is split into two parts: a high-level spring compression estimator and a low-level compression regulator. This separation of control tasks enables SEAs to be managed efficiently and effectively, despite the computational limitations of microcontrollers.

3.5.1 Low-level Control Method

The microcontroller is running a PID spring compression control and a PID motor position control. The control structure is illustrated in Fig. 3.8. The PID spring compression control output is the setpoint of the PID position controller. The motors are controlled by the PID position controller based on the reading of the encoder. The PID position controller serves as a safety feature and ensures that the exoskeleton is reset to a safe position in the following circumstances. First, the user does not feel comfortable operating the exoskeleton glove; second, the Bluetooth connection is lost; third, the position limits are reached. The potentiometer has a data collection rate of 50Hz and runs an average filter that outputs the filled potentiometer reading to the PID spring compression controller at 10Hz. The PID spring compression controller runs at 10Hz, and the PID position controller runs at 100Hz. The microcontroller calculates seven variables locally, spring compression, filtered potentiometer reading, raw encoder reading, and speed and acceleration of the filtered potentiometer reading and raw encoder reading. These seven variables are sent to a computer in the cloud at 10Hz using Bluetooth. The microcontroller received a compression setpoint from the SEA force predictor at 10Hz.

SEA spring compression calculation is sensitive to small changes in motor position or potentiometer reading. For example, 1 revolution of the motor or 1 deg of the potentiometer reading can cause a 0.08 mm spring compression difference. A 0.1mm dead zone is implemented for the PID compression control such that when the error is smaller than 0.lmm, the PID controller is not updated to avoid oscillation.

3.5.2 High-Level Multi-Layer Perceptron (MLP) Compression Predictor

The LSEA compression predictor has been modified on the basis of the force calibration network. This network takes eight inputs and predicts the desired LSEA spring compression, which will be fed into the low-level compression controller at a rate of 10Hz. The input contains the desired calculated spring compression, filtered potentiometer reading, raw encoder reading, and the speeds and accelerations of the raw encoder and filtered potentiometer. This network is trained using the same data as the calibration method and fine-tuned using a hyperparameter tuning method. This network can predict the desired compression based on sensor data (illustrated in Fig. 3.8). Data is fed into the predictor to update the predicted compression. The desired compression is then sent to the low-level compression controller to carry out precise compression control.

3.6 LSEA Force Control using Surface Fitting (SF) Method

The Surface Fitting (SF) method is a low computational cost approach that is often used as a comparison baseline for the MLP method. This method requires minimal computational cost, making it suitable for control of multiple SEAs on a single microcontroller. This approach can be used as an alternative to the high-level MLP technique, while the low-level control remains unchanged. The control software architecture is shown in Fig. 3.9. Instead of running on a separate computer and transmitting the results to the microcontroller via a wireless connection, this method can be executed directly on the microcontroller.

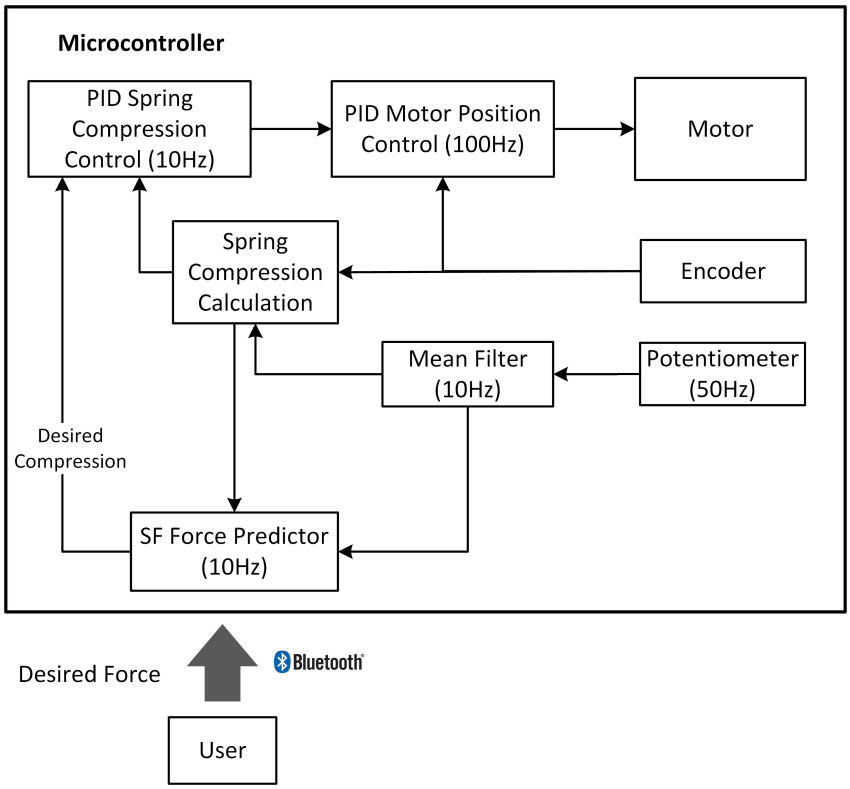


Figure 3.9: SEA control software architecture using SF method. All computation is performed locally on microcontroller. The SF method uses spring compression and filtered potentiometer data to predict the desired spring compression.

3.7 Experimental Results

The experimental results section is composed of three subsections. The first subsection details the data preparation process. The second subsection displays the force calibration results of both the Surface Fitting (SF) technique and the Multi-Layer Perceptron (MLP) technique. The third subsection shows the force control results of both the SF and MLP technique.

3.7.1 Data Preparation

The LSEA sensor data was gathered during the data preparation process and the force of the fingertip linkage was measured with a load cell. Two load cells are used to measure the force in two different directions, which produces two distinct reference force values (illustrated in Fig. 3.2). In order to prepare the data for training, the measured force must be pre-processed. The Linkage Transformation Ratio is used to convert the force at the fingertip linkage to the SEA output shaft, which is then used as a reference value. This value is a consequence of deformation, friction, and sensor noise. For instance, the force output of an SEA is 10N, however, 5N of that is lost in the transmission process. Thus, the net output force is 5N.

The training dataset consists of 10,400 data points obtained from two load cells at 12 different contact angles. The test and validation datasets were randomly chosen from the training dataset, with 2,416 and 140 data points respectively.

3.7.2 Force Calibration Results

The Surface Fitting (SF) force calibration model, depicted in Fig. 3.10, utilizes a 3rd-order polynomial and fits most points to the surface. This graph indicates that the force output is not completely consistent. Even when angles and compression are similar, there is a slight amount of error in the force output.

The results of the hyperparameter tuning for the MLP approach and SF approach are presented in Tab. 3.1. The optimal network for the MAE loss function consists of three layers, each with 8, 16, and 32 nodes. For the MSE loss function, the optimal network has two layers, each with 16 and 32 nodes.

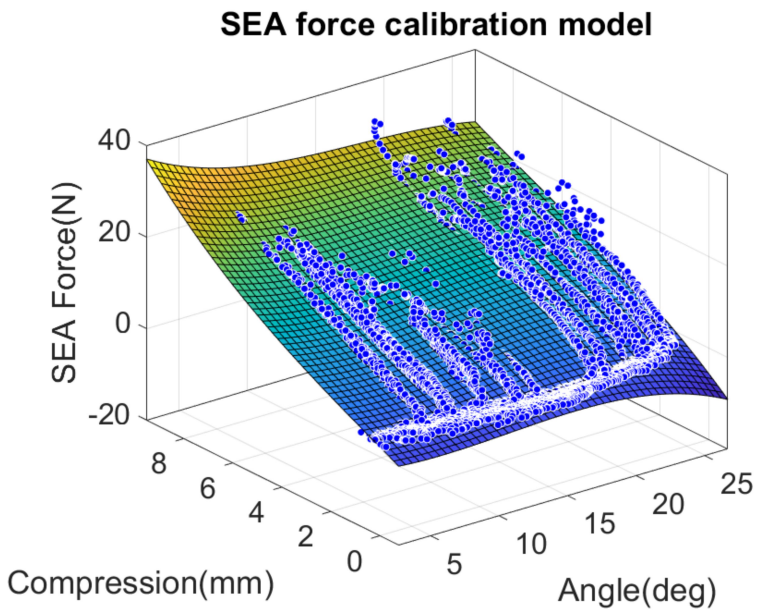


Figure 3.10: LSEA fingertip linkage force calibration model built using Surface Fitting (SF) method.

Table 3.1: SEA fingertip force calibration model hyper-parameter tuning results

Method	Loss function	Opt Network Struc	Error
MLP	MSE	16-32	1.432
MLP	MAE	8-16-32	2.13
SF	MSE	N/A	1.855
SF	MAE	N/A	0.987

The result of the MLP model is compared with the SF method on the testing dataset. The results are shown in Tab. 3.1. For the MLP method, the MSE loss function has a performance gain of 22% over the SF method. When using the MAE loss function, the SF is 72.5% more accurate. The MSE loss function of MLP and MAE of SF is selected to make further comparisons. Fig. 3.11 shows a demo of force calibration using the above two methods in the validation dataset. The MLP method that uses the MSE loss function has a better fit under 15N, while the surface fitting has a better fit over 15N.

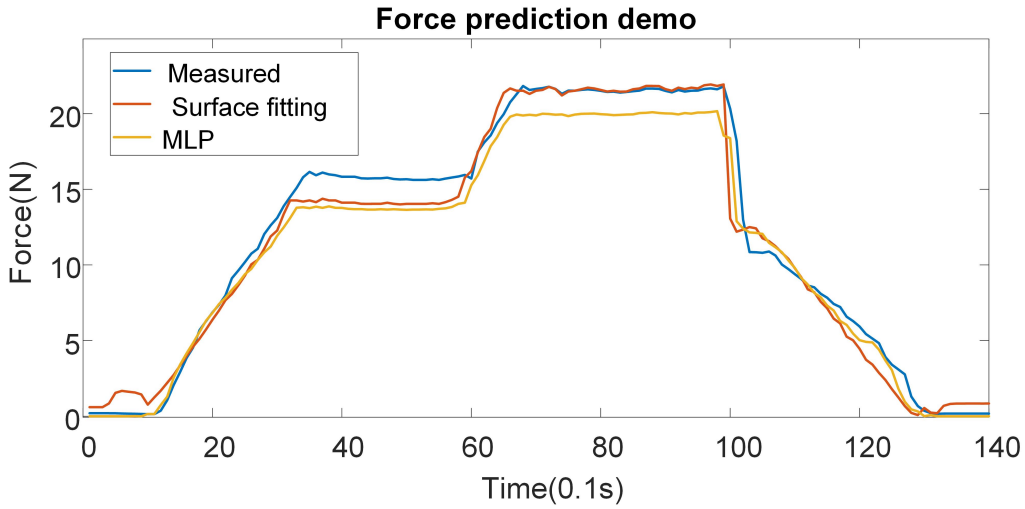


Figure 3.11: LSEA force calibration simulation performed using data from the validation dataset.

Table 3.2: SEA compression prediction model hyper-parameter tuning results

Method	Loss function	Opt Network Struc	Error
MLP	MAE	8-16	0.19
MLP	MSE	8-16	0.17
SF	MSE	N/A	0.35
SF	MAE	N/A	0.24

3.7.3 Force Control Results

The SF force control model is shown in Fig. 3.12. The model used 3rd order polynomial and most points are fitted on the surface. This model describes the relationship between compression, force, and angle.

The MLP approach result is shown in Tab. 3.2. For the MAE loss function, the optimal network consists of two layers, each with 8,16 nodes. For the MSE loss function, the optimal network consists of two layers, each with 8,16 nodes.

The comparison of the weighted MLP model and the SF method on the testing dataset is presented in Tab. 3.2. The weighted MLP model with both MAE and MSE loss functions

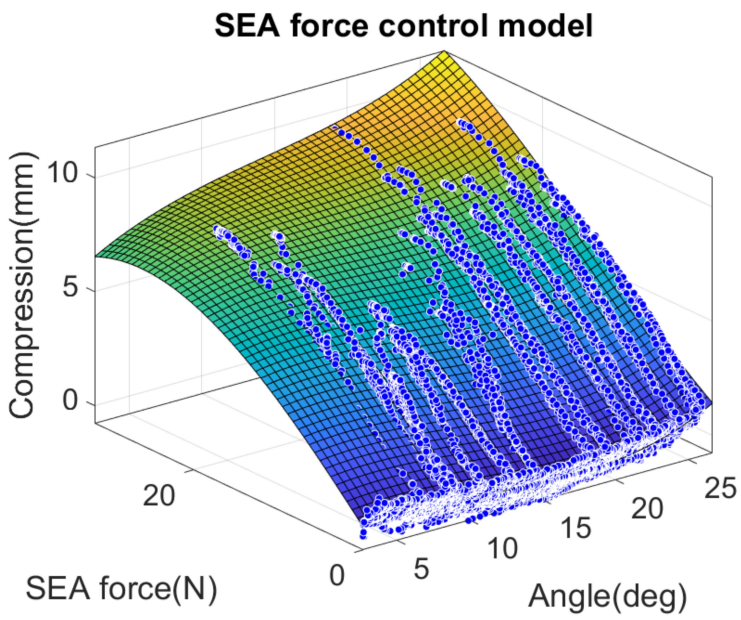


Figure 3.12: LSEA compression prediction model built using Surface Fitting (SF) method.

yields better accuracy than the SF method. The MLP model with MSE loss function has a 51.4% performance improvement over the SF model with MSE loss function. Additionally, the MLP model with MAE loss function has a 20.9% performance gain over the SF model with MAE loss function. Therefore, MSE is chosen as the optimal loss function for the MLP model due to its higher performance gain over the SF model. MAE loss function is selected for the SF model due to its lower performance difference compared to the weighted MLP model.

Fig. 3.13 illustrates a simulation of force control using the validation dataset with two different methods. The simulation used the measured force to predict the desired compression and then compared the predicted compression with the measured compression. The MLP method was more effective for low compression, while the SF method was more suitable for higher compression.

Fig. 3.14 illustrates the comparison between the weighted MLP prediction control and the SF control when test the control methods on actual LSEA hardware. The output force on

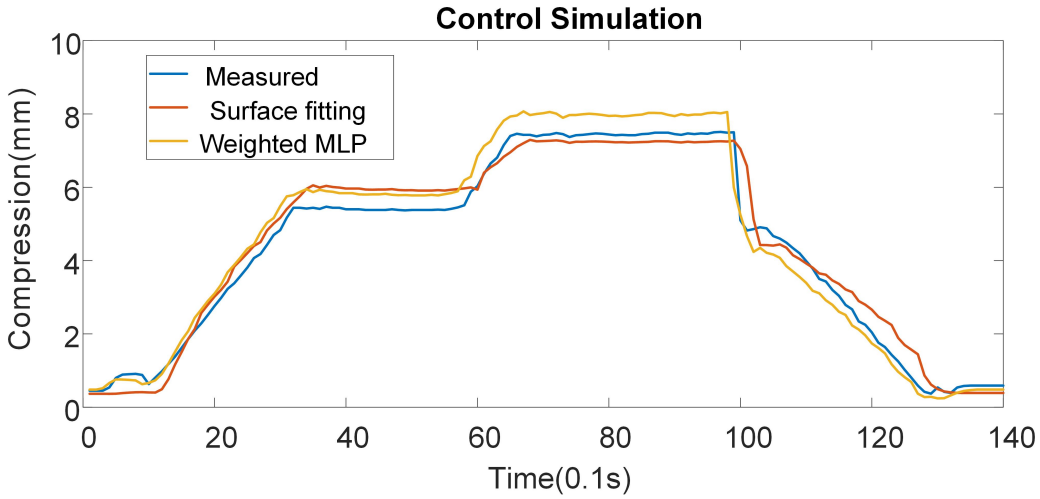


Figure 3.13: LSEA compression setpoint prediction simulation performed using data from the validation dataset.

the LSEA output shaft was set to 5N, 10N, and 15N, and the desired force on the fingertip was calculated through the Linkage Transformation Ratio. A load cell was used to measure the actual applied force. The results indicate that the MLP approach had a slightly more accurate force output, with an error of 0.3N, while the SF method had an error of 0.6N. These experiments demonstrate that the force control was successful when implemented on the LSEA hardware. The settling time increases as the output force increases due to a longer actuation time. Each control method outputs a force of 5/10/15N at ten different contact angles. The linkage starting angle will be 5 degrees. The settling time is measured when the linkage comes in contact with the load cell and ends when the output reaches the error bounds of 0.3N and 0.6N. The settling time and force error for different angles are averaged and shown in Tab. 3.3.

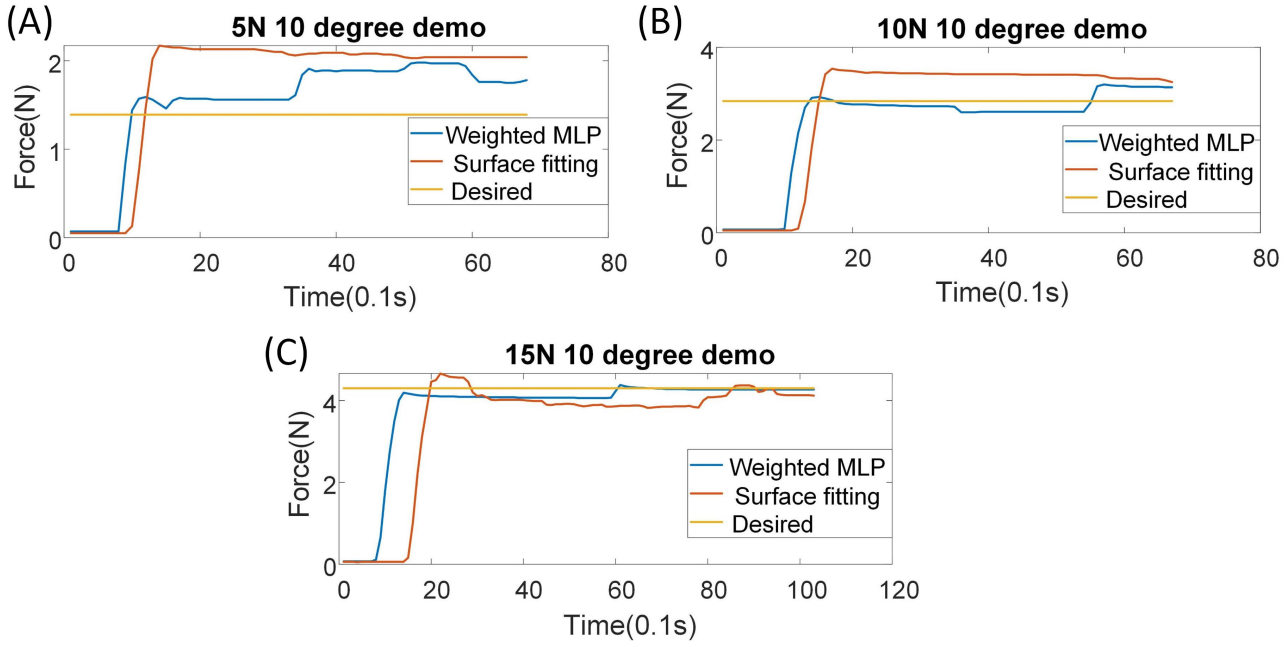


Figure 3.14: Index finger LSEA force control experiments. This experiment is performed on the LSEA hardware. The MCP joint begins at an angle of 5 degrees and makes contact with the load cell at 10 degrees. (A) The desired LSEA output shaft force is set to 5N. (B) The desired LSEA output shaft force is set to 10N. (C) The desired LSEA output shaft force is set to 15N.

Table 3.3: Force control error and settling time when experimenting on LSEA hardware

Force	Method	Force Error Avg	Settling Time Avg
5N	SF	0.29N	0.32s
10N	SF	0.41N	0.50s
15N	SF	0.18N	0.58s
5N	MLP	0.24N	2.72s
10N	MLP	0.30N	3.22s
15N	MLP	0.1N	3.15s

Table 3.4: Force calibration performance comparison on the validation datset

Method	Error Avg	Error Max	Error Avg	Error Max
Kinematics Calc	3.55N	9.38N	34.31%	44.7%
SF(MAE)	0.886N	7.261N	8.06%	35.72%
MLP(MSE)	1.120N	5.304N	10.18%	29.13%

3.8 Discussion

The contribution of this chapter is to demonstrate that friction force and linkage deformation can cause an average of 34.31% and a maximum of 44.7% difference in force measurement on the compact, lightweight SEA used in the exoskeleton glove prototype. These findings can be applied to most exoskeleton gloves that require force transformation from the SEA to the fingertips. Calibration is necessary for any compact, lightweight SEA in an exoskeleton glove. Two methods are proposed to control and calibrate a compact, lightweight linear SEA used in an exoskeleton.

When using the MSE loss function on the test dataset, the MLP method is 22% more accurate. When using the MAE loss function, the SF is 72.5% more accurate. Both methods showed a similar performance when the test was conducted using the validation dataset. MLP method has a slightly higher average difference but a smaller maximum difference. Both methods are more accurate 65% than the calibration of the kinematics calculation. The performance comparison of both methods on the validation dataset is shown in Tab. 3.4.

In the force control test, the MLP method is more accurate using both MAE and MSE loss functions. When using the MSE loss function, the MLP method is 51.4% more accurate. When using the MAE loss function, the MLP method is 29.1% more accurate. When tested using a validation dataset, the MLP method has a slightly higher average difference but a

Table 3.5: SEA force control performance comparison on the validation datset

Method	Error Avg	Error Max	Error Avg	Error Max
SF(MAE)	0.347mm	1.946mm	7.99%	38.15%
MLP(MSE)	0.409mm	1.505mm	9.38%	30.00%

smaller maximum difference. In real-world testing with the SEA hardware, the MLP method shows 27.5% less average force error than the SF method. However, the SF method costs 84. 6% less average settlement time than the MLP method. The SF approach can run on a microcontroller due to its simplicity. The performance comparison of both methods on the validation dataset is shown in Tab. 3.5.

Both methods have decent accuracy and an acceptable settlement time. For applications that do not require low settling time and have ample computational power, the MLP approach is slightly more accurate. The SF method is suitable for microcontroller applications with limited computational power. The SF method is also ideal for applications where control speed is critical.

Chapter 4

Exoskeleton Glove Force Planning Techniques

4.1 Introduction

Force planning method is important for the exoskeleton glove to maintain a stable grasp. The proposed exoskeleton glove aims to grasp objects of different weights and sizes; therefore, it requires various force outputs to the exoskeleton fingertip linkages. When the grasp force is insufficient, the object will fall out of hand. When the grasp force is excessive during the grasping of a cylinder or sphere, the target object might experience deformation during the grasping. When the grasp force is excessive during tip or tripod grasp, the target object may break the friction force and slip out of the optimal grasp position due to the limited DOF of the proposed exoskeleton glove.

This chapter introduces two innovative force planning methods to apply adequate grasp force to the exoskeleton glove: a hybrid slip-grasp force planning method and a vision-based initial grasp force planning method. A hybrid slip-grasp method inspired by Lee et al. [65] was implemented, which combined Series Elastic Actuators (SEAs) and Force-Sensitive Resistors (FSRs) to improve performance. However, the slip-grasp method relied on a time-consuming trial-and-error process. To overcome this issue, a vision-based initial grasp force estimation method was introduced. Using this method, a rough estimate of the required grasp force

could be obtained, significantly reducing the number of trials needed to achieve a stable grasp using the slip-grasp approach.

4.2 Slip Grasp Force Planning

A proposed solution to the problem of force planning is to employ a hybrid slip detection technique that combines SEA and FSR. This combination is designed to identify any slippage and adjust the output force accordingly.

4.2.1 Hybrid Slip Detection

SEAs are used to measure large exoskeleton fingertip force changes, and FSRs are used to measure the change in contact force in the fingerpad. SEAs do not provide good sensitivity for small force changes due to the relatively high spring stiffness, linkage deformation, linkage backlash, and indirect spring compression measurement. SEAs usually can only consistently sense force changes of at least 0.5N. But SEAs have good repeatability and can sense force changes above 0.5N. FSRs, on the other hand, have terrible repeatability [82] and can have more than $\pm 50\%$ errors on random raw data. But FSRs have a great sensitivity. These two sensors are combined and used in a hybrid slip detection system. A slip is detected if the SEA compression changes by more than 2mm in 0.3 seconds or the raw FSR reading changes by more than 10% in 0.3 seconds.

A slip-grasp policy is used to perform real-time force planning during grasping. First, apply a 2N force to the fingertips using SEA, which should grasp most objects used daily. If the grasp force is insufficient, the object will slip during grasping. 0.5N force is added to each LSEA and a 100Nmm torque is added to the RSEA each time the FSRs detect a slip. 1N

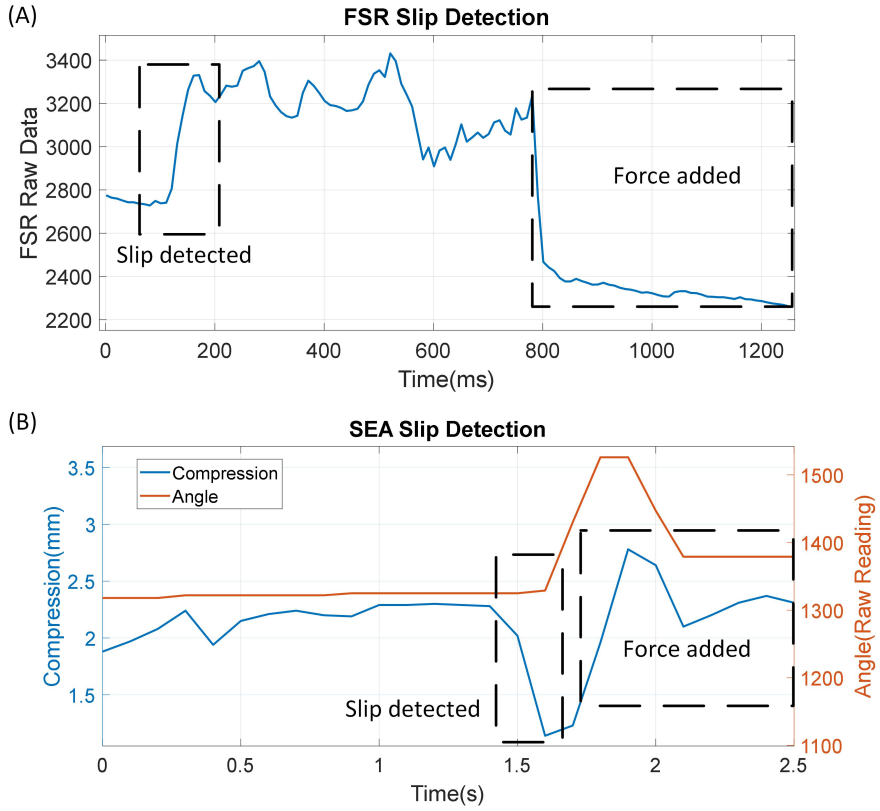


Figure 4.1: Hybrid slip detection and slip-grasp force planning method demonstration. (A) Slip detected using the FSR raw reading, the force added within 500ms after a slip is detected. (2) Slip detected using the SEA raw reading, the force added within 500ms after a slip is detected.

force is added to each LSEA and 200Nmm torque is added to the RSEA if both the FSRs and SEA detect a slip.

4.2.2 Experimental Results

The experiments were carried out on a subject with a healthy hand. The subject does not exert force on the exoskeleton during grasping. The setup of the experiment is shown in Fig. 4.2. The exoskeleton glove applied minimal grasp force to ensure that the subject could lift the object without slippage. After lifting the object, the subject shaken the object to produce slippage. Fig. 4.1 (A) and (B) show that the exoskeleton can properly adjust the

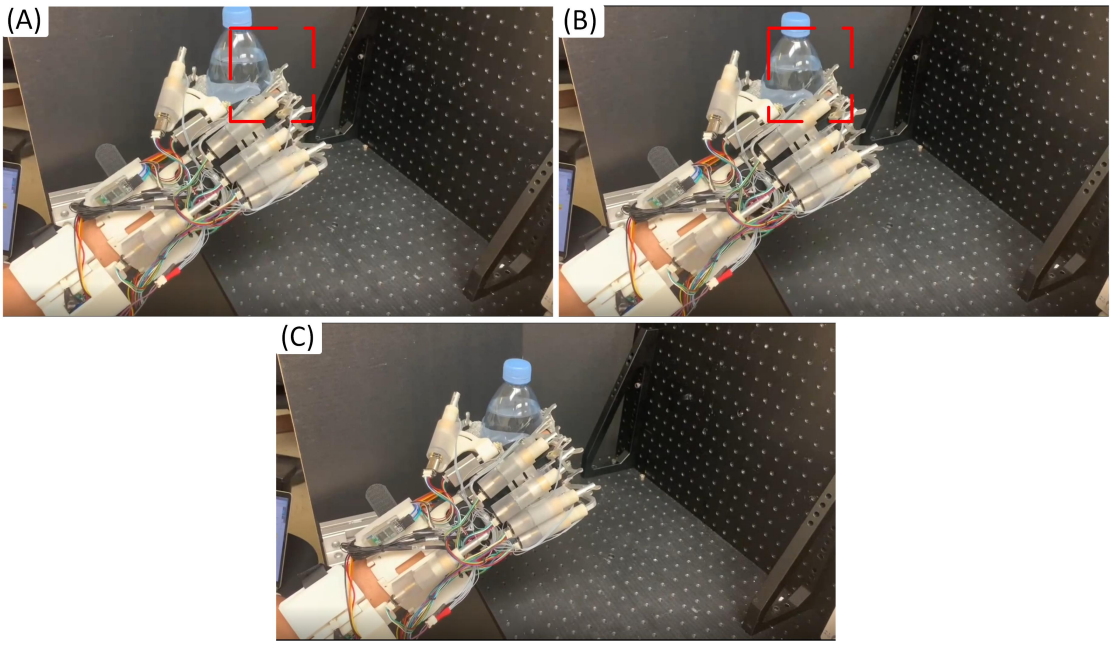


Figure 4.2: Slip detection and slip-grasp force planning method experiment setup. The red box showed the amount of slippage observed on the target object. (A) The user can lift the object without falling. (B) The user shakes the target object causing it to slip. (C) Force is added to stop the target object from slipping.

SEA output force to maintain stability when slippage is detected. The quantitative results of the slip-grasp method are listed later in this chapter when comparing the slip-grasp method with the vision-based force planning method (Sec. 4.3.7).

4.3 Vision-Based Force Planning for Exoskeleton Gloves

Humans can grasp and lift an object without knowing its weight, surface material, and size. Studies show that even with restricted haptic feedback, humans can still perform a stable grasp based on visual input [83]. Humans can use vision to estimate the grasp force. If the object's actual size, weight, and surface friction coefficient match the estimation, the predicted force will be close to the optimal grasp force. Haptic feedback is used to detect slippage when the estimated force is not accurate. Humans can adjust the grasp force based

on haptic feedback. This chapter proposes a vision-based method that mimics a human grasping procedure to solve the initial grasp force estimation issue on exoskeleton gloves. The computer vision algorithm estimates the size, weight, and surface material of the target object, and then calculates the initial grasp force based on static force analysis.

4.3.1 Review of Vision-Based Material Analysis

Sean et al. [84] built the Material Recognition in the Wild with the Materials in Context (MINC) dataset with images of human-labeled material classification in real world and proposed a deep learning based material segmentation method. Such a method uses a convolution neural network to generate a probability map and the Conditional Random Field (CRF) algorithm to calculate a label for each pixel. The advantage of this method is that it does not require pixel-wise labels, which is ideal for applications with limited segmented data. MINC-2500 is a subset of the MINC dataset, containing 57,500 image patches for 23 different types of materials. The vision-based force planning system proposed in this chapter is built based on the segmentation structure developed by Sean et al. [84] and uses the MINC-2500 dataset for training.

4.3.2 System Overview

The vision-based force planning method is designed to grasp an object without the need for detailed measurements in advance. The goal is to find the initial grasp force by estimating the size, shape, weight, and surface material of the object to be grasped.

This vision-based initial grasp force estimation method uses voice input from a microphone to initiate grasping and releasing (input voice command: “Grasp” and “Release”). Such a voice command system is described in [85]. After receiving a grasp command, the camera

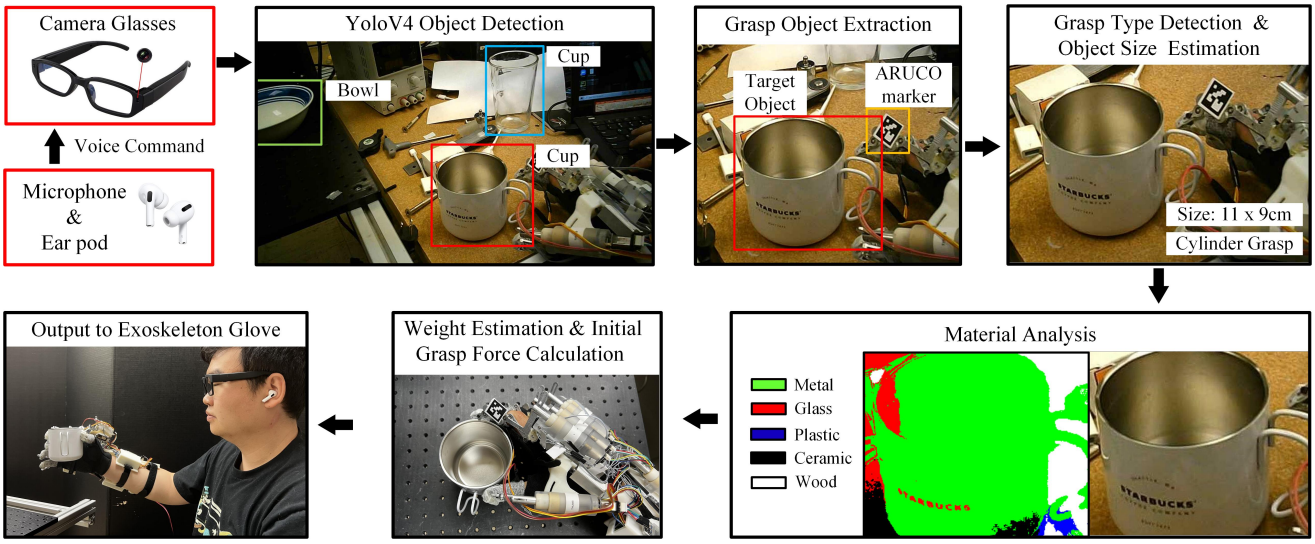


Figure 4.3: Overview of vision-based initial grasp force prediction procedure.

embedded in the glasses will start to take pictures and perform the following three steps on the image to calculate the initial grasp force.

(1) The input images are sent to an object detector trained on the Common Objects in Context (COCO) dataset. This step will help the vision-based force planning method to understand the environment by detecting all objects in the view and extracting the target object using an ARUCO marker on the exoskeleton glove (ARUCO marker is shown in Fig. 4.3). In this step, the target object category and size are acquired and the grasp type is determined according to the target object's category.

(2) The surface material of the target object is acquired by performing a material classification or material segmentation on the image patch of the target object. Given the object's size and surface material, the object's weight can be estimated.

(3) The initial grasp force is calculated based on the spatial location of the exoskeleton, the surface material of the target object, and the weight.

The initial grasp force is then sent to the exoskeleton. The SEAs are FSRs on the exoskeleton

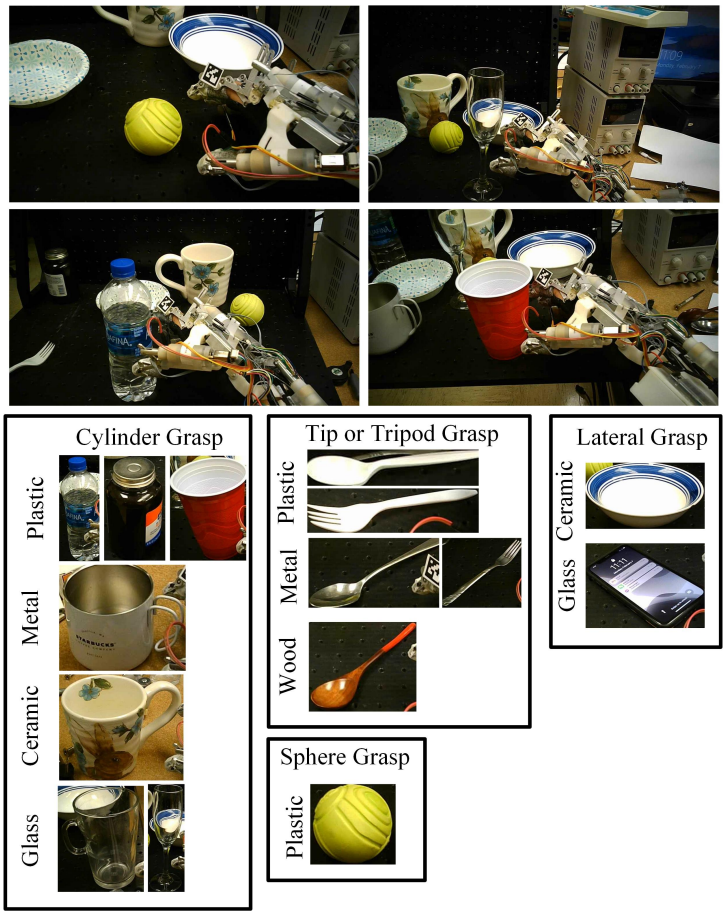


Figure 4.4: Sample images of the exoskeleton grasping environment, object category, and object material.

glove will detect slip while applying the predicted initial grasp force, and the slip-grasp method will adjust the grasp force as needed. The structure of the vision-based force planning method is shown in Fig. 4.3. Sample images for the exoskeleton grasping environment, object category, and object material are shown in Fig. 4.4.

4.3.3 Target Object Detection

There are two common approaches for detecting and locating an object in an image: object detection [86, 87, 88, 89] or instance object segmentation [90, 91]. Object detection requires

image annotation using a bounding box during training. The detection result for object detection is a bounding box that contains background information. Thus, object detection is faster during training and inference. Instance object segmentation requires pixel-wise image annotation for training, and the detection result consists of pixels of the object without backgrounds. Object segmentation can better understand the object's shape, but is slower during training and inference than object detection. In this research, object detection was used over object segmentation for two reasons.

(1) Object detection is faster than object segmentation during inference process. Two-stage object segmentation will first detect the object in a bounding box and then extract the object pixels from the background. Single-stage object segmentation uses a decoder network to find the object and an encoder to propagate the object's pixels. Both methods mentioned above need additional calculations during inference, thus being slower than object detection using bounding boxes. The need for speed in this application necessitated the use of object detection instead of object segmentation.

(2) Object detection techniques have better data availability. Object detection does not necessitate pixel-level labeling, and this study may address the difficulty of grasping items that are not included in publicly accessible datasets. To detect uncommon objects in a small-scale project, transfer learning or fine-tuning on a public dataset is usually employed. Therefore, object detection techniques are utilized in this research as they require less annotation and will have better data availability.

The state-of-the-art object detection methods are based on Single Shot Detector (SSD) [87], Faster R-CNN [88], EfficientDet [89], and YOLOV4 [86]. Researchers have previously tested these methods on the COCO dataset [92]. The inference speed and Mean Average Precision (mAP) at 50% Intersection over Union (IOU) of seven different object detection methods are compared on the collected validation dataset in order to select the most suitable object

detection method. Sample images of the collected validation dataset are shown in Fig. 4.4. The experimental results are shown in Fig. 4.9. According to the experiments, YOLOV4 was selected as the object detection method used in this research; it better balanced speed and mAP than other methods.

4.3.4 Size Estimation for Target Object

The data output from object detection will be an object category vector $\mathbf{c}$, an object bounding box vector $\mathbf{B}$, and an object center vector $\mathbf{S}$. The n^{th} object detected in an image belongs to category ${}^{n}c$.

$$\mathbf{c} = [{}^{1}c, {}^{2}c, ..., {}^{n}c] \tag{4.1}$$

For the n^{th} object detected in an image, the object's bounding box $\mathbf{{}^{n}b}$ is the combination of the upper left corner $\mathbf{{}^{n}p_{ul}} = ({}^{n}x_{ul}, {}^{n}y_{ul})$ and the lower right corner $\mathbf{{}^{n}p_{lr}} = ({}^{n}x_{lr}, {}^{n}y_{lr})$.

$$\begin{aligned} \mathbf{B} &= [\mathbf{{}^{1}b}, \mathbf{{}^{2}b}, ..., \mathbf{{}^{n}b}] \\ &= [({}^{1}x_{ul}, {}^{1}y_{ul}, {}^{1}x_{lr}, {}^{1}y_{lr}), ..., ({}^{n}x_{ul}, {}^{n}y_{ul}, {}^{n}x_{lr}, {}^{n}y_{lr})] \end{aligned} \tag{4.2}$$

For the n^{th} object detected in an image, the center of the pixel of the detected object is located at $\mathbf{{}^{n}s}$ calculated from the bounding box $\mathbf{{}^{n}b}$.

$$\begin{aligned} \mathbf{S} &= [\mathbf{{}^{1}s}, \mathbf{{}^{2}s}, ..., \mathbf{{}^{n}s}] \\ &= [({}^{1}x_{s}, {}^{1}y_{s}), ..., ({}^{n}x_{s}, {}^{n}y_{s})] \end{aligned} \tag{4.3}$$

The target object is selected on the basis of the distance to the ARUCO marker located on the exoskeleton glove. The output of the ARUCO Application Programming Interface (API) contains the center coordinate of the marker: $\mathbf{s_m} = (x_m, y_m)$.

The exoskeleton glove used in this research is right-handed with the ARUCO marker placed on the index finger linkage (see Fig.4.4). The object to be grasped is likely to be on the lower right of the ARUCO marker. A weighted distance function was customized to find the distance between the ARUCO marker center coordinate $\mathbf{s_m}$ and the detected n^{th} object center ${}^{\mathbf{n}}\mathbf{s}$:

$$\begin{aligned} {}^{n}d = w_0(x_m - {}^{n}x_s) + w_1({}^{n}y_s - y_m) \\ + \sqrt{(x_m - {}^{n}x_s)^2 + (y_m - {}^{n}y_s)^2} \end{aligned} \tag{4.4}$$

where, ${}^{n}d$ is the n^{th} object distance between the object center and the ARUCO marker center. w_0 is the weight that serves as the penalty for the object located on the right of the marker, and w_1 is the weight that serves as the penalty for the object located above the marker. $({}^{n}x_s, {}^{n}y_s)$ is the coordinate of the center of the object from the vector of the center of the object ${}^{\mathbf{n}}\mathbf{s}$. The grasped object's index i can be found by minimizing the customized distance function ${}^{n}d$:

$${}^{i}d = min({}^{1}d, {}^{2}d, ..., {}^{n}d) \tag{4.5}$$

The category of the target object is ${}^{i}c$, the bounding box is ${}^{\mathbf{i}}\mathbf{b}$, and the center coordinate is ${}^{\mathbf{i}}\mathbf{s}$.

The size of the target object is estimated on the basis of the size of the ARUCO marker. The marker's width and height are 2 centimeters. The coordinates are explained in Fig. 4.5. The coordinates of the bounding box of the detected object ${}^{\mathbf{i}}\mathbf{b}$ can be transferred from the pixel coordinates to the camera coordinates and then to the marker coordinates. The Euclidean distance between points **e** and **f** in the marker coordinates is the length of the object w in centimeters (points are marked in Fig. 4.5). The Euclidean distance between points **f** and **g** in the marker coordinates is the height of the object h in centimeters.

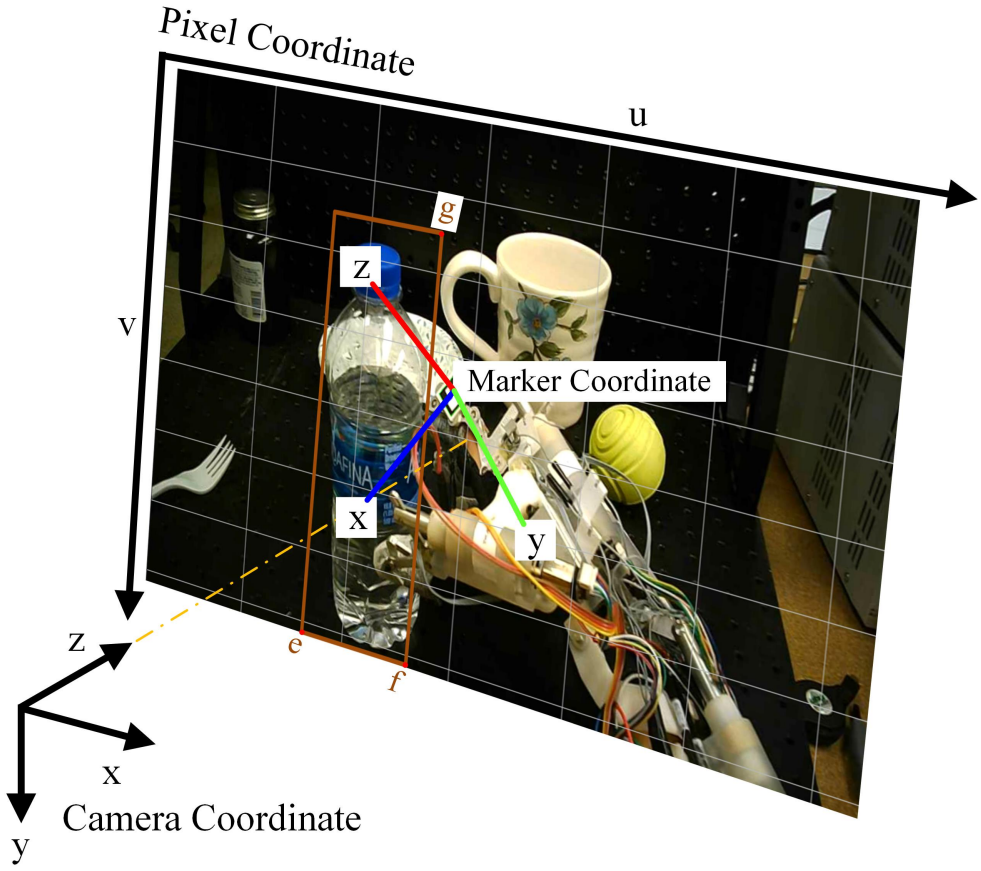


Figure 4.5: Illustration of the camera, marker, and pixel coordinates.

The following method can be used to convert points from pixel coordinates to marker coordinates. The ARUCO API outputs the rotation vector ($\mathbf{r}$) in the axis-angle representation and the center coordinate ($\mathbf{t}$) of the marker in the camera coordinates. The target object is assumed to have the same distance from the camera as the marker. To transfer a point $\mathbf{p_p} = (u, v)$ from the pixel coordinates to the camera coordinates $\mathbf{p_c} = (x_c, y_c, z_c)$, the following equations are used:

$$x_c = \frac{u - s_x}{f_x} d_z \tag{4.6}$$

$$y_c = \frac{v - s_y}{f_y} d_z \tag{4.7}$$

where, d_z is the distance from the marker to the camera in the camera coordinates. s_x and s_y are the coordinates of the principle point in the camera coordinates (640 and 360 in this application). f_x and f_y are focal lengths of x and y axes in pixels (1184 and 1249 in this application).

To transfer a point $\mathbf{p_c} = (x_c, y_c, z_c)$ from the camera coordinate to the marker coordinate $\mathbf{p_m} = (x_m, y_m, z_m)$, the following equations are used:

$$\mathbf{R} = Rodrigues(\mathbf{r}) \tag{4.8}$$

$$\mathbf{p_m} = \mathbf{R}^T(\mathbf{p_c} - \mathbf{t}) \tag{4.9}$$

where, Rodrigues formula was used to build a transformation matrix $\mathbf{R}$ from the axis-angle representation rotation vector $\mathbf{r}$. $\mathbf{t}$ is the marker coordinate center represented in the camera coordinates.

4.3.5 Material Classification for Target Object

There are two common approaches to detect the surface material of an object, including image classification based on center pixels and semantic segmentation on the entire image [84, 93, 94]. The most widely used material classification datasets are the Flicker Material Dataset (FMD), MINC, and open surface datasets. There are only limited pixel-wise annotated images provided, and most of these annotated images are furniture from the interior of a house, which is very different from this application. Due to the limited availability of annotated data, a pixel-wise supervised classification method such as UNet [91, 94] cannot

be used. For this application, the center pixel classification method was used to classify the material of a given object image, and the conditional random field (CRF) [95] method was used for segmentation.

Since this application focuses on grasping objects that are used daily, as shown in Fig. 4.4, the number of classes in MINC-2500 was reduced from 23 to 5, which include ceramics, metal, glass, plastic and wood.

Material Classification Challenges

Initially, the classification of deep learning material was trained and tested on MINC-2500 and achieved good accuracy. The original MINC dataset material patch classification was trained on VGG-16, AlexNet, and InceptionV1 in 2014. The VGG-16 architecture was used as a performance baseline to test the new networks, which achieved high classification accuracy in the ImageNet challenge: InceptionResNetV2 and ResNet152V2. Moreover, networks that achieve similar classification accuracy were tested, but have faster inference speeds: InceptionV3, ResNet50V2, and MobileNetV2. In addition to different network architectures, the NetVLAD pooling method was tested, which is a clustering-based pooling method commonly used in speaker verification, face detection, and place recognition [96].

The model weight is transferred from ImageNet, and the training is terminated if the validation loss does not decrease for ten consecutive epochs. The training result was tested on a small dataset similar to the use case of this research, which contains images from the FMD dataset and images collected online. Some sample images from the data can be visualized in Fig. 4.8. The dataset contains 169 images for each of the five categories.

The training results and model performance comparison are shown in Tab. 4.1.

According to training results, ResNet50V2, MobileNetV2, and InceptionV3 are the top 3

Table 4.1: Results of training on MINC-2500 and testing on the collected dataset

Network	MINC-2500 ACC	Collected Dataset ACC	Speed*
VGG-16	71%	22%	279ms
InceptionV3	**83%**	21%	**215ms**
VGG-16-N*	68%	21%	292ms
InceptionV3-N*	77%	22%	225ms
MobileNetV2	75%	20%	**173ms**
ResNet50V2	78%	**23%**	228ms
ResNet152V2	84%	22%	487ms
InceptionResNetV2	81%	21%	472ms

Speed*: the inference time is measured by inference of one image on a E5-1260 CPU.
-N*: NetVALD layer with 32 clusters is added after the last convolution.

networks that achieve a good time and performance balance in the MINC-2500 validation set. However, the MINC-2500 does not have a perfect generalization to material classification. The context in the MINC dataset is very different from that of this application, which prevents the network from finding a correct label during testing on the collected dataset. NetVALD clustering pooling layer also does not improve accuracy.

To solve the generalization issue, transfer learning was performed to retrain the model in the collected dataset. Transfers from ImageNet and MINC-2500 weight were experimented. The results are shown in Tab. 4.2. The results show that the transfer from MINC-2500 using ResNet50V2 has the best accuracy when testing on the collected dataset.

Table 4.2: Results of transfer ImageNet and MINC-2500 weight on the collected dataset

Network	Transfer MINC-2500 ACC	Transfer ImageNet ACC
ResNet50V2	**79%**	76%
MobileNetV2	72%	71%
InceptionV3	75%	72%

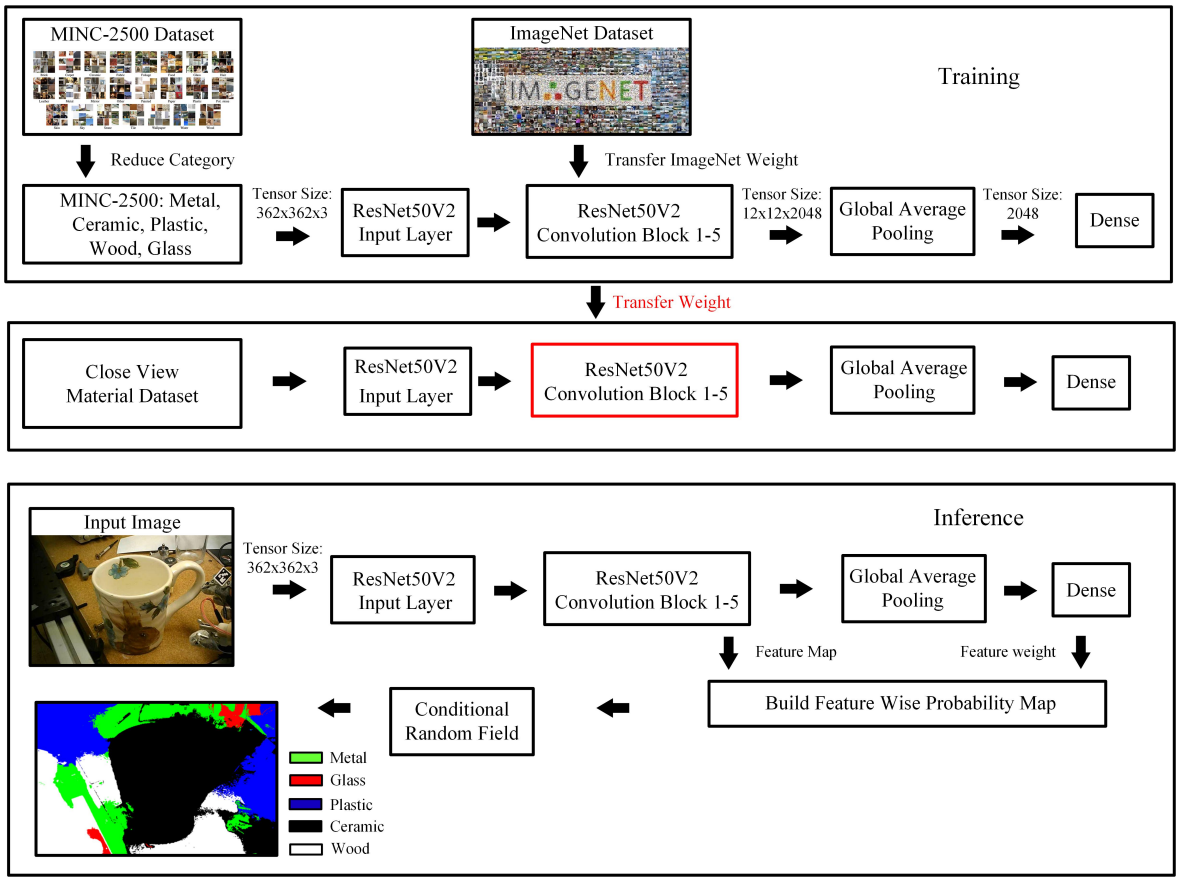


Figure 4.6: Training and inference procedure for vision-based material classification and segmentation.

Transfer Leaning using ResNet50V2

Based on the experimental results from the previous section, ResNet50V2 was used to transfer the weight from ImageNet to the MINC-2500 dataset. The number of material classes in MINC-2500 is reduced to metal, ceramic, plastic, glass, and wood. The input layer is modified to match the MINC-2500 size, the ResNet50V2 convolution blocks have not been modified, and the weight is trained using the initial ImageNet value. The output of the convolution layer consists of 2048 feature maps $\mathbf{M}_{[\mathbf{12x12x2048}]}$. The pooling layer uses Global Average Pooling (GAP) to group the feature maps $\mathbf{M}_{[\mathbf{12x12x2048}]}$ to $\mathbf{M}_{[\mathbf{1x1x2048}]}$ and classified into five classes multiplied by weight $\mathbf{W}_{[\mathbf{5x2048}]}$. Due to the low generalization accuracy of

the MINC-2500 dataset on this application, the MINC-2500 weight was transferred to the collected dataset using the same architecture. Fig. 4.6 shows the training and inference procedure.

When inferring on a sample image, the ResNet50V2 network was modified to output a class probability map ${}_{\mathbf{c}}\mathbf{P}_{\mathbf{[1x5]}}$ and a feature-map-sized class probability map ${}_{\mathbf{f}}\mathbf{P}_{\mathbf{[12x12x5]}}$ using Grad-CAM [97]. The Grad-CAM is generated using the following equation:

$$ {}_{\mathbf{f}}\mathbf{P} = \sum_{n=1}^{2048} {}^{\mathbf{n}}\mathbf{W}^{\mathbf{n}}\mathbf{M} \tag{4.10} $$

where, ${}^{\mathbf{n}}\mathbf{M}$ is the n^{th} feature map and ${}^{\mathbf{n}}\mathbf{W}$ is the weight of the n^{th} feature map. The probability map ${}_{\mathbf{f}}\mathbf{P}_{\mathbf{[12x12x5]}}$ will be resized to pixel level probability map ${}_{\mathbf{p}}\mathbf{P}_{\mathbf{[362x362x5]}}$ using cubic spline interpolation. The probability map ${}_{\mathbf{p}}\mathbf{P}_{\mathbf{[362x362x5]}}$ and colored image $\mathbf{I}_{\mathbf{[362x362x3]}}$ are input into a Conditional Random Field (CRF) algorithm to perform pixel-level unsupervised segmentation by minimizing the following energy function [95]:

$$ {}^{c}E(\mathbf{x}) = \sum_{i} U(i) + \sum_{(i,j)} Par(i,j) \tag{4.11} $$

where, ${}^{c}E(x)$ is the energy function for class c. $\mathbf{x}$ is the set of all pixels in image $\mathbf{I}$. i and j are pixel indexes in set $\mathbf{x}$. i and j control a nested loop to pair each pixel with all other pixels without repetition. $U(i)$ is the unary energy that is the negative log probability of a pixel belonging to class c. $Par(i,j)$ is the pairwise energy that measures the pixels' spacial and color similarity. The unary and pairwise energy is defined in the following equations:

$$ U(i) = -log({}^{\mathbf{i}}_{\mathbf{p}}\mathbf{P}_{\mathbf{c}}) \tag{4.12} $$

$$Par(i,j) = exp(-\frac{|^{i}p - {}^{j}p|^2}{2s_p^2} - \frac{|\mathbf{^{i}I} - \mathbf{^{j}I}|^2}{2s_c^2}) \tag{4.13}$$

where, $\mathbf{{}^{i}_{p}P_{c}}$ is the probability of the pixel level of the i^{th} pixel in the image belonging to class c. ${}^{i}p$ and ${}^{j}p$ are the positions of the i^{th} and j^{th} pixels. $\mathbf{^{i}I}$ and $\mathbf{^{j}I}$ are the RGB values of i^{th} and j^{th} pixels. Long-range connections were used in the energy calculation. Thus, the pairwise energy contains only the appearance kernel. s_p and s_c are the position similarity and color similarity parameters, respectively. The parameter values s_p and s_c were chosen to be 60 and 10, respectively, based on [95]. The results of the CRF algorithms will be an updated pixel-level probability map $\mathbf{_{crf}P_{[362x362x5]}}$.

The classification results can be found by finding the maximum value of the class probability map $\mathbf{_{c}P}$. The results can be directly used to estimate the grasp force. The segmentation results can be used to perform pixel-wise classification when the target object contains different materials.

Weight Estimation for Target Object

The estimated size and material of the target object can be obtained based on the methods described in the previous sections. However, the information is insufficient to estimate the weight, and some assumptions need to be made to calculate the target object's volume.

The target objects in this application can be classified into four categories: fork/spoon, bottle/cup/wine glass, sports ball, and apple/cell phone. An apple and cell phone's weight is not affected much based on size; thus, the average weight of an apple and cell phone can be used as the weight of the target object. Sports balls are usually very light, so it was assumed that a sports ball weighs 20 grams if it has a diameter less than 5cm, weighs 100 grams if it has a diameter between $5 - 10cm$, and weighs 250 grams if the diameter is larger

than $10cm$.

The shape of a spoon or fork can be simplified to a plate with a $0.1cm$ thickness. Thus, the weight of a spoon or fork can be estimated using:

$$v_{sf} = 0.1wh \tag{4.14}$$

$$s_{sf} = v_{sf}\rho \tag{4.15}$$

where, w and h are the estimated width and height of the target object, respectively. ρ is the density of the material of the target object. v_{sf} is the volume of the object. s_{sf} is the weight of the target spoon or fork.

The shape of a bottle, cup, and wine glass can be simplified to a hollow truncated cone. It is assumed that the truncated cone has two thirds of the volume of a cylinder of the same height. The thickness can be assumed to be $0.2cm$. Thus, the weight of a bottle when filled with water can be estimated using the following.

$$\begin{aligned} v_b &= \frac{2}{3}(v_o - v_i) \\ &= \frac{2}{3}(\pi(\frac{w}{2})^2 h - \pi(\frac{w}{2} - 0.2)^2(h - 0.4)) \end{aligned} \tag{4.16}$$

$$s_b = v_b\rho + v_i\rho_w \tag{4.17}$$

where, v_b is the volume of the material to form the bottle. v_o is the outer volume, v_i is the inner volume. s_b is the weight of the bottle. ρ is the density of the material of the bottle. ρ_w is the density of water.

The weight of a cup can be estimated as similar to that of a bottle. The only difference is that a cup might have a handle and will make volume calculation inaccurate. The size of the handle was assumed to be 30% of the weight of the cup w. Thus, the weight of a cup when full of water can be estimated using the following.

$$\text{if } h \geq w: \quad v_c = \frac{2}{3}(v_o - v_i) = \frac{2}{3}(\pi(\frac{w}{2})^2 h - \pi(\frac{w}{2} - 0.2)^2(h - 0.2)) \tag{4.18}$$

$$\text{if } w \geq h: \quad v_c = \frac{2}{3}(v_o - v_i) = \frac{2}{3}(\pi(\frac{0.7w}{2})^2 h - \pi(\frac{0.7w}{2} - 0.2)^2(h - 0.2)) \tag{4.19}$$

$$s_c = v_c\rho + v_i\rho_w \tag{4.20}$$

where, v_c is the volume of material to form the cup. v_o is the outer volume, and v_i is the inner volume. s_c is the weight of the bottle. ρ is the density of the material of the cup. ρ_w is the density of water. A wine glass is a special cup with a long leg, so it was assumed that the capacity of the glass is 50% of a normal cup. Thus, the weight of a wine glass when full of water can be estimated using the expression:

$$s_{wg} = v_c\rho + 0.5v_i\rho_w \tag{4.21}$$

4.3.6 Gasp Force Estimation

The initial grasp force is calculated based on the predicted weight and shape of the target object. Fig. 4.7 illustrates the coordinate systems for initializing the force. The origin of the world coordinates is placed at the object's center. The exoskeleton glove coordinates are located in the center of the Inertial Measurement Unit (IMU). The IMU is calibrated to align with the world coordinates in the beginning. Assuming that there is no torque applied on the object and the contact forces are normal to the last link of each of the exoskeleton fingers, for an arbitrary object, the force equilibrium equation can be expressed as:

$$\sum_i \mu {}^w\mathbf{R}_e {}^e\mathbf{R}_i {}^e\mathbf{F}_i + M\mathbf{g} = \mathbf{0} \tag{4.22}$$

where, $i \in \{thumb, index, middle, ring, little\}$, ${}^w\mathbf{R}_e$ is the rotation matrix from the exoskeleton glove coordinates to the world coordinates, which is calculated based on readings from the IMU. ${}^e\mathbf{R}_i$ is the rotation matrix from the fingertip i to the exoskeleton glove coordinates, which is calculated based on the forward kinematics of the glove [64]. ${}^e\mathbf{F}_i$ is the vector of the contact force applied to the fingertip i, which is measured based on a calibrated linear series elastic actuator [63]. M is the object's mass and $\mathbf{g}$ is the vector of gravitational acceleration.

For the cylinder and tip grasp, the direction of the friction force on each fingertip is always opposite to gravity. Therefore, the above equation can be simplified to $\sum_i \mu F_i = Mg$.

4.3.7 Experimental Results

The experiment section encompassed three primary components. Initially, the datasets utilized for object detection validation and material classification were introduced. Subsequently, the performance of object detection, size estimation, and material classification

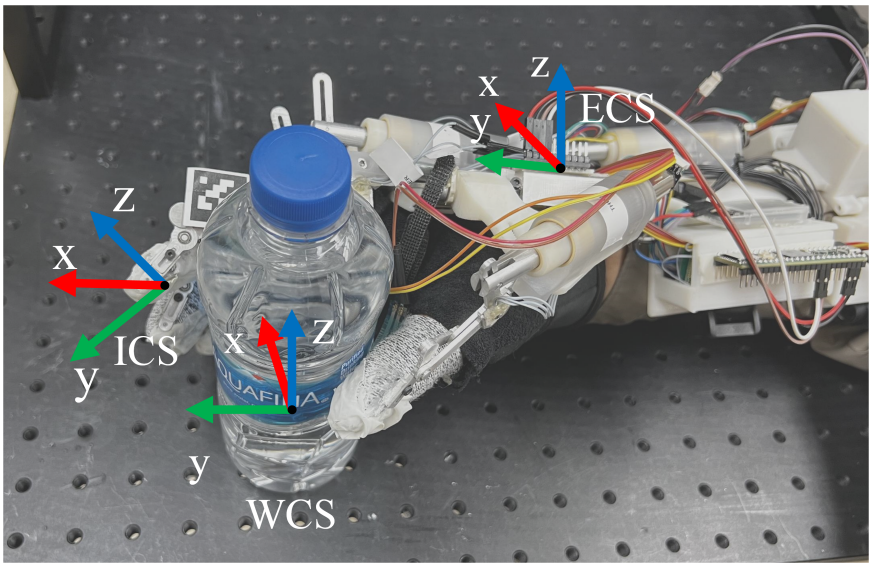


Figure 4.7: The coordinate systems for initial force estimation. WCS: world coordinate system. ECS: exoskeleton glove coordinate system. ICS: i-th fingertip coordinate system.

within these datasets was assessed. Lastly, a vision-based HMI was integrated as an extension of the slip-grasp force planning method for the exoskeleton glove. The experiments were structured to contrast the combined approach of vision and the slip-grasp method against the exclusive use of the slip-grasp force planning method.

Datasets

Two small datasets were built to verify this application (object detection dataset); one for vision-based grasp force planning method validation and one for transfer learning material classification (material classification dataset).

The dataset for vision-based grasp force planning method validation has 30 images taken from 1080P SVWSUN Video Glass worn by an exoskeleton glove user. Each grasp object is labeled using a bounding box. Sample images are shown in Fig. 4.4.

The dataset for transfer learning consists of five labels: ceramic, plastic, metal, wood, and

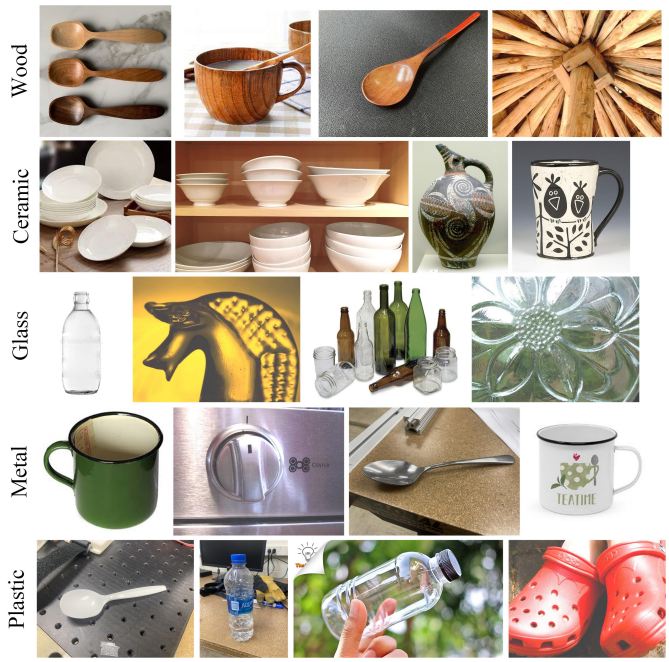


Figure 4.8: Sample images used in the material classification training.

glass. Each class has a training set of 119 images, a testing set of 30 images, and a validation set of 20 images. Each image is labeled on the basis of the object's center material. This dataset contains images from an online image search, the FMD dataset, and images taken for the grasp objects used in this research. Sample images are shown in Fig. 4.8. Images in this dataset have more details and fewer contexts than images in MINC-2500.

Object Detection and ARUCO Marker Detection

The labeled object detection validation dataset was used to test the performance of different networks trained on the COCO dataset. A mean Average Precision (mAP) at 50% Intersection over Union (IoU) was used to quantify object detection performance. The speed was measured based on the average inference time of 10 images using the E5-1260 CPU. The results are shown in Fig. 4.9. Multiple networks were tested and YOLOV4 with a 0.75 threshold was selected based on mAP and speed.

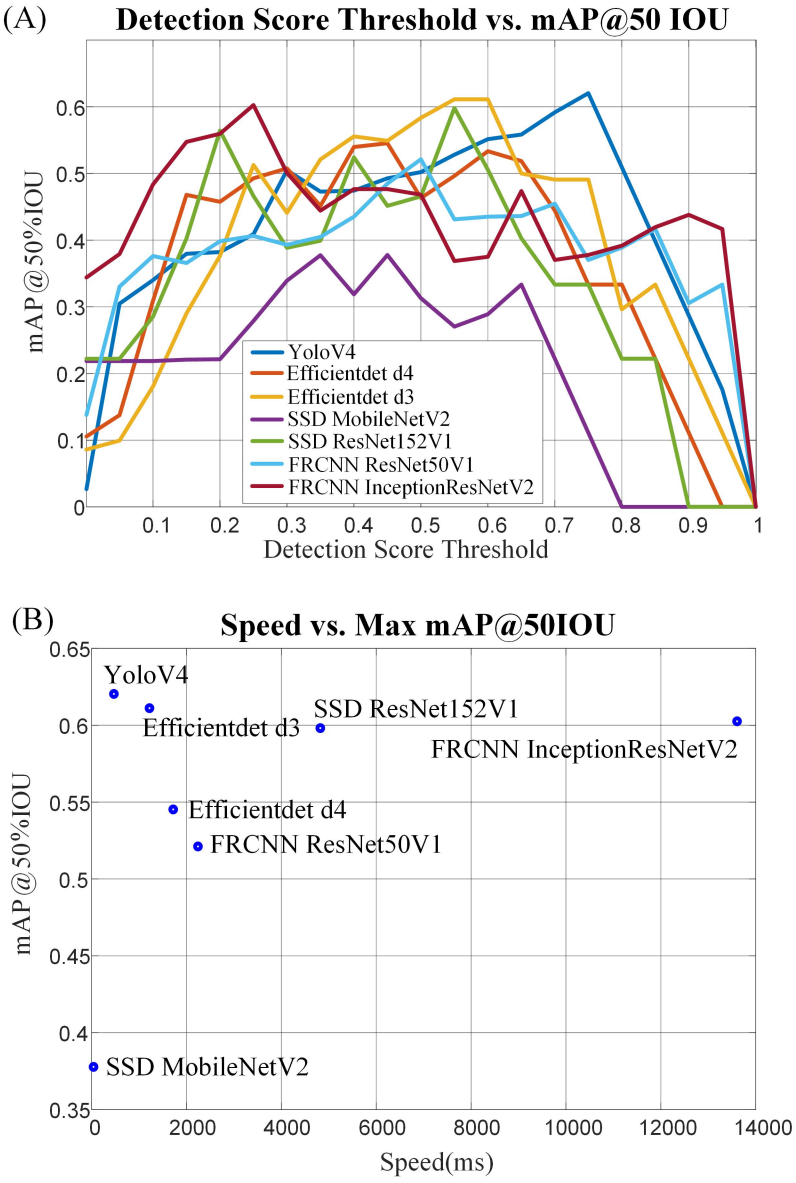


Figure 4.9: Performance comparison of different Object detection methods. (A) mean Average Precision (mAP) at 50% Intersection over Union (IoU) of 7 different state-of-the-art neural networks. (B) mAP vs. average inference time of each neural network.

The successful detection rate R_s of object detection and ARUCO marker detection can be calculated using the following equation:

$$R_s = \frac{TP - FP}{n} \tag{4.23}$$

where, TP is true positive, which means that the ARUCO API detection successfully detects the marker, and the object detection successfully identifies the center object. FP is false positive, which means that the marker detection recognized the wrong marker or the object detection detects the wrong center object. n is the total number of test images. The experiments yielded a 90% success rate in the object detection validation dataset.

Object Size Estimation

The experiment involved evaluating the object detection validation dataset by comparing the detected target object's size with the ground truth sizes. For this purpose, images successfully detected by both the YoloV4 object detector and the ARUCO marker detector were utilized. This dataset comprised 27 images featuring 15 different objects observed from various angles. To obtain the predicted size for each object, the average of the estimated sizes from different angles was taken. The ground truth sizes were determined based on the width and height of the orthographic projection, as illustrated in Fig. 4.10.

The obtained results are presented in Tab. 4.3. To quantify the difference between the predicted and actual object sizes, the percentage difference between the products of width (w) and height (h) was calculated. This evaluation metric is termed the Mean Absolute Percentage Error (MAPE). The MAPE difference between the predicted and actual object sizes was found to be 26.9%. The main source of this error was identified as the estimation process, particularly when utilizing the bounding box to estimate the object's dimensions.

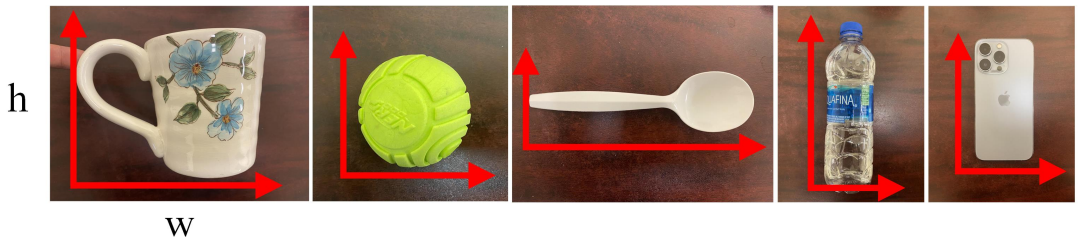


Figure 4.10: Examples of ground truth object size measurements.

This error tends to occur when the object is placed at an angle during detection.

Table 4.3: Object size estimation experimental results

Object	**Actual* (cm)**	**Predicted* (cm)**	**Diff**
Plastic Bottle A	6.5x11	8x13	45.4%
Plastic Bottle B	7x20	8x21	20%
Plastic Spoon	13x4	14x3.5	5.8%
Plastic Fork	14x4	14x3.5	12.5%
Plastic Cup	12x10	14x11	28.3%
Plastic Ball	7x7	6.8x6.8	5.6%
Metal Spoon	18x3.5	14.1x6.7	49.9%
Metal Fork	18x2.5	12x6	60%
Metal Cup	14x9	13.4x10.6	12.7%
Wood Spoon	16.5x4	11x7.8	30%
Glass Cup	12x14	12.9x15.1	15.9%
Wine Glass	20x5.5	19.4x7.3	28.7%
Ceramic Cup	19x11.5	18.8x15.2	30.8%
Ceramic Bowl	17.5x17.5	17x10	44.5%
Cell Phone	15x7.5	12x8.2	12.5%
MAPE	-	-	26.9%

Actual*: The actual size is defined by the width times height in centimeters.
Predicted*: The predicted size is defined by the width times height in centimeters.

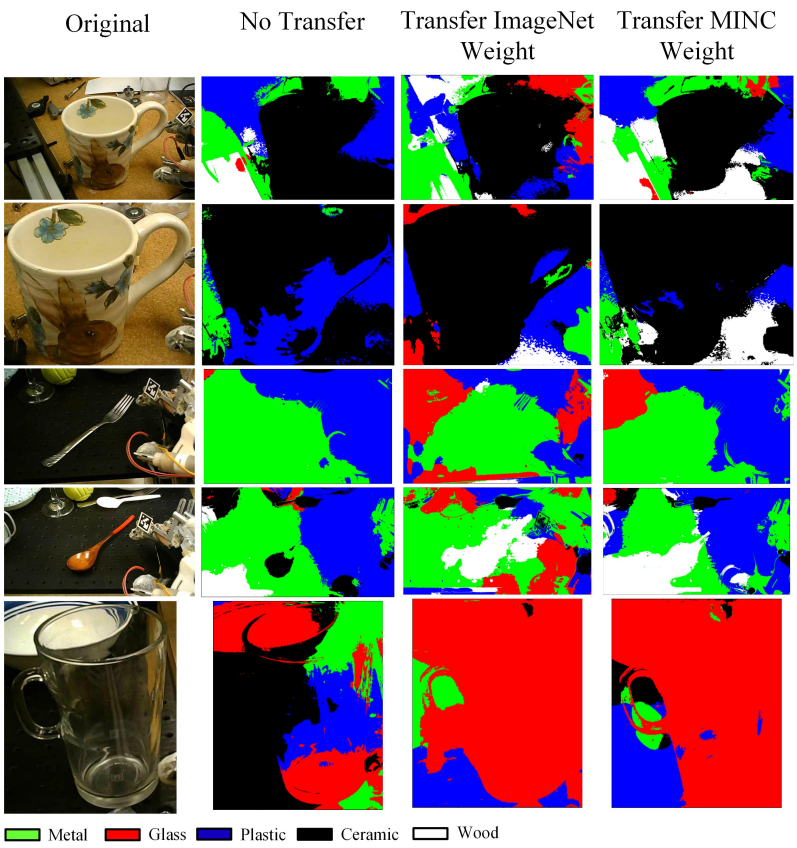


Figure 4.11: Sample material segmentation results.

Object Material Classification

The training and testing results in the proposed material classification dataset are shown in Tab. 4.2. According to the accuracy and speed of classification, the material classification network used is ResNet50V2. The weight is transferred from the MINC-2500 dataset.

Material classification validation was also performed on the object detection dataset. The material classification accuracy for all detected objects was 96%. In addition to material classification, material segmentation is performed using the CRF method to visualize the result of material classification. Sampe images of material segmentation are shown in Fig. 4.11.

Object Weight Estimation

This experiment comprised 27 detected images used in the size estimation process, weight estimation was based on estimated sizes and material obtained in the previous section. It is assumed that the materials have the following densities: plastic ($0.92g/cm^3$), metal ($7.85g/cm^3$), glass ($2.7g/cm^3$), ceramic ($6g/cm^3$), and wood ($0.9g/cm^3$).

The results of these experiments are presented in Tab. 4.4. However, it should be noted that the weights of the containers varied due to differences in fluid level. For consistency, it was assumed that all containers were full. To assess the accuracy of the weight estimation, the Mean Absolute Percentage Error (MAPE) was used as the evaluation metric. The MAPE between the predicted and actual object weights was found to be 59.8%. The relatively large weight estimation error can be attributed to the following factors. First, weight estimation is heavily influenced by size estimation, which in turn can be affected by the angle at which the object appears in the camera. Second, the assumption of standard shapes for all objects, such as cylinders or boxes, may not hold true for most cases, where cups might have handles, and wine glasses may have long legs, leading to deviations from the standard shapes used in the estimation process. Furthermore, despite some instances of substantial percentage errors, the overall weight difference remains acceptable. For instance, the metal fork experienced a weight estimation error of 35g, representing a 159.1% overestimation compared to its actual size. The average weight difference across all objects is only 173g, which still provides meaningful information for initial grasp force planning.

Object Grasp Experiments

The experimental procedure involving human subjects in this study received approval from the Carilion Clinic Institutional Review Board (IRB-19-330). Due to the nature of the

Table 4.4: Object weight estimation experimental results

Object	**Actual(g)**	**Predicted(g)**	**Diff**
Plastic Bottle A	12-512	698	36.3%*
Plastic Bottle B	207	432	108.7%
Plastic Spoon	3	5	66.7%
Plastic Fork	3	5	66.7%
Plastic Cup	11-502	881	75.5%*
Plastic Ball	69	100	45%
Metal Spoon	48	74	54.2%
Metal Fork	22	57	159.1%
Metal Cup	172-576	619	7.5%*
Wood Spoon	7	8	14.3%
Glass Cup	358-779	1459	89.5%*
Wine Glass	188-369	399	8.1%
Ceramic Cup	480-1059	1756	65.8%*
Ceramic Bowl	315	596	89.2%
Cell Phone	222	200	10%
MAPE	-	-	59.8%

*: Containers have various weights due to content. During weight estimation, all containers are assumed to be full of water.

exoskeleton glove used in this research, which is a rigid linkage exoskeleton, the user cannot apply any force to the fingertips of the exoskeleton linkages when wearing it.

The grasp procedure is as follows: The user initiates the system using a personalized voice command system [98] to capture a 1280x760 pixel image. By employing the methods proposed in previous sections, the size and weight of the grasped object can be calculated. The 9-DOF MPU-9250 IMU detects the pitch, yaw, and roll of the exoskeleton glove using an AHRS filter. Using the weight of the object and the IMU data, the initial grasp force is

computed, and the exoskeleton glove applies this force to each fingertip [63]. The slip-grasp system is then utilized to stabilize the grasp.

During the experiment, each of the 15 objects present in the object detection dataset was subjected to 2-6 grasping attempts from various angles and water levels (for containers), resulting in a total of 64 grasp trials. Among these trials, 6 experienced failure of object detection, while 5 encountered errors in material classification. The grasp success rate is defined as the success in picking up the target object. The overall grasp success rate using vision-based grasp force planning method combined with the slip-grasp method was 87.5%.

Comparison Between Vision-based Force Estimation and Slip Grasp Force Planning

To demonstrate the effectiveness of the vision-based force estimation method. 64 experiments were performed using only the slip-grasp force planning method and achieved a grasp success rate of 71.9%, while the vision-based method achieved 87.5%. The success rate for each grasp category is shown in Fig. 4.12.

Table 4.5: Comparison between vision-based force estimation and slip-grasp force planning

Object	**Slip-Grasp (succ/total trials)**	**Vision (succ/-total trials)**	**Slip (index force/thumb torque)**	**Vision (index force/thumb torque)**
Plastic Bottle	3/6	5/6	3.67N / 367Nmm	2.73N / 459Nmm
Wine Glass	6/6	6/6	2.67N / 267Nmm	1.59N / 267Nmm
Plastic Spoon	2/4	4/4	2N / 200Nmm	0.75N / 31.6Nmm
Metal Spoon	3/4	4/4	2N / 200Nmm	1.2N / 50.8Nmm

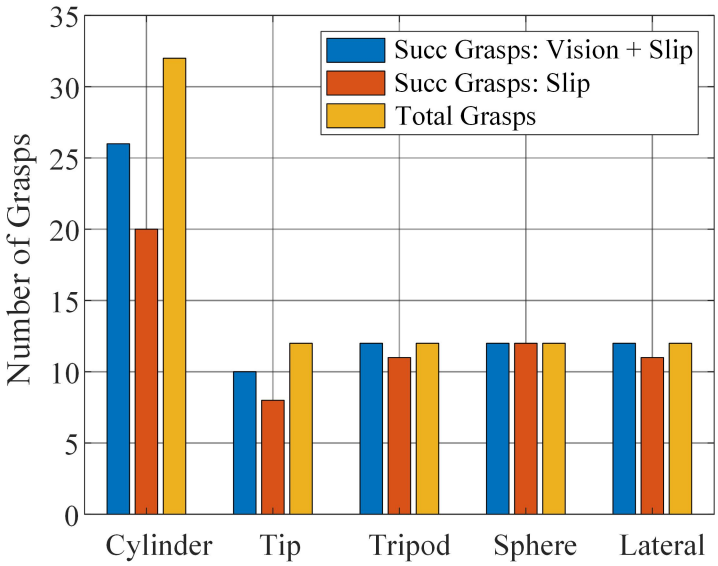


Figure 4.12: Experimental results of grasping daily used objects using vision-based initial grasp force prediction method and slip-grasp method. Blue: number of successful grasps performed using the vision-based initial force estimation with slip-grasp method. Red: number of successful grasps performed using only the slip-grasp method. Yellow: the total number of grasps for each individual method.

The comparison experiment reveals that utilizing a combination of vision-based force estimation with the slip-grasp system leads to a higher success rate compared to using only the slip-grasp system. To demonstrate the benefits of utilizing the vision-based initial force estimation technique, 20 additional grasp trials were carried out that involved four different items: a plastic bottle, a wine glass, a plastic spoon, and a metal spoon. These particular objects were chosen based on their notable performance in previous grasp experiments.

For the vision-based method, the initial grasp force was determined using the vision-based force estimation system, and the slip-grasp method was not utilized in this experiment. For the slip-grasp method, a predefined initial grasp force of 2N and 200Nmm is used. This method adjusted the grasp force based on slippage to achieve a stable grasp (details can be found in previous sections of this chapter).

The grasping process was facilitated by 6 Series Elastic Actuators (SEA) as depicted in Fig.

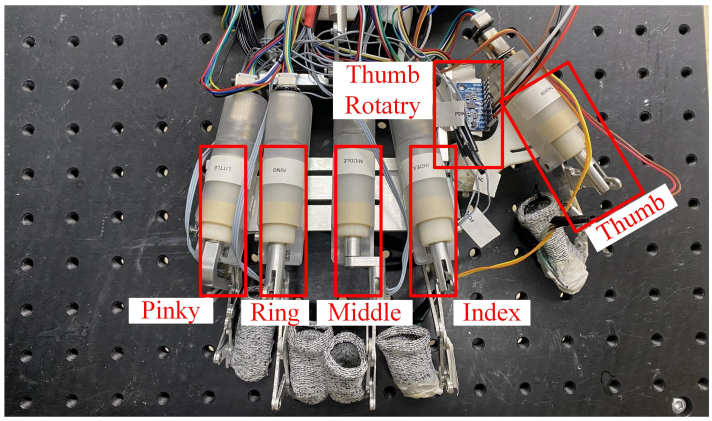


Figure 4.13: Series Elastic Actuators (SEA) are used to apply force on the exoskeleton glove in the grasp experiment.

4.13. The force and torque output of the index finger and thumb rotatory SEAs, which are the most critical actuators during grasping, were measured and reported in Tab. 4.5.

The results from the additional 20 grasp experiments are presented in Tab. 4.5 and Fig. 4.14, demonstrate that the vision-based force estimation system can produce reasonable initial grasp forces for various objects. This offers three main advantages during grasping. First, the initial grasp force estimate helps prevent the application of insufficient thumb torque, which can result in slippage. For example, in Fig. 4.14 (B), the plastic water bottle could not be lifted by the slip-grasp method due to the insufficient predefined thumb torque. Second, the initial grasp force can prevent the application of excessive force and torque. For example, in Fig. 4.14 (F), the plastic spoon could not be lifted by the slip-grasp method due to excessive fingertip force and thumb torque. Third, even for objects that can be successfully lifted by the slip-grasp method, incorporating a vision-based force estimation system allows for a reduction in the applied force (as shown in Tab. 4.5), thereby optimizing the grasping process.

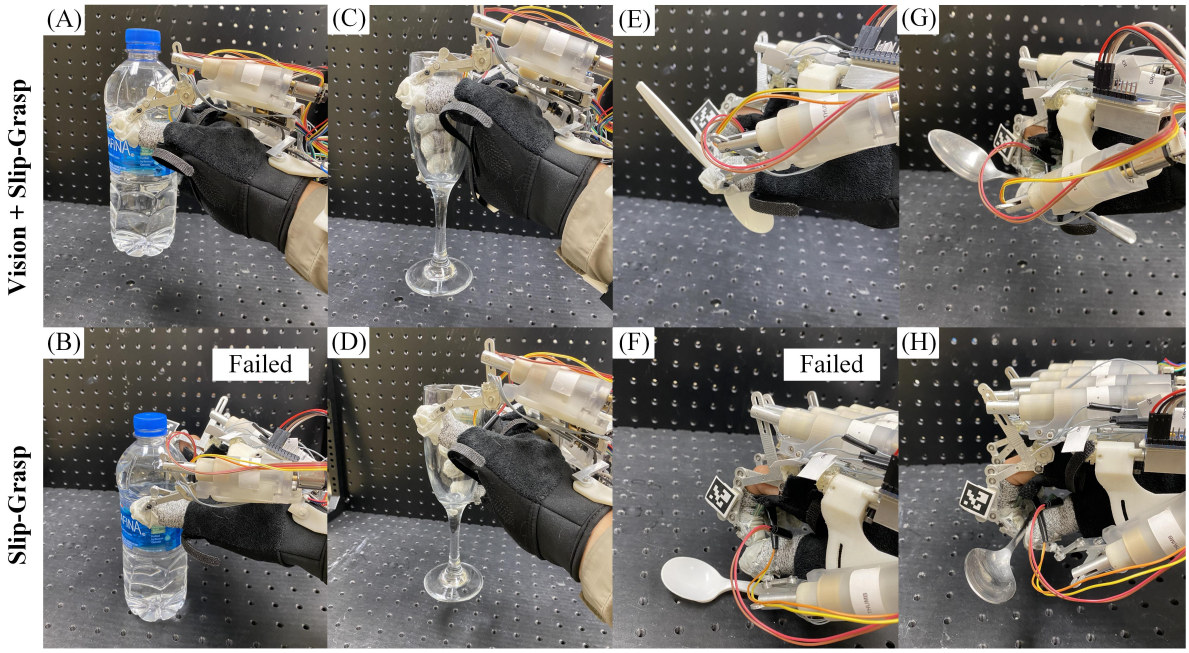


Figure 4.14: Demonstration of grasping daily used objects using vision-based initial grasp force prediction method and slip-grasp method. (A) Successfully grasp a 512g water bottle with vision system. (B) Failed to grasp a 512g water bottle using the slip-grasp method due to inadequate thumb torque. (C) and (D) Successfully grasp an 188g wine glass with both the vision system and the slip-grasp method. (E) Successfully grasp a 3g plastic spoon with vision system. (F) Failed to grasp a 3g plastic spoon using the slip-grasp method due to excessive force and torque. (G) and (H) Successfully grasp a 48g metal spoon with both the vision system and the slip-grasp method.

System Latency

The image processing is running on a desktop server with an E5-1260 CPU, and there is no GPU involved. The estimated size, weight, and surface friction coefficient are sent to the exoskeleton's onboard microcontroller, which generates the initial grasp force using IMU data and operates the exoskeleton. The computation time for processing a single image is around 700 ms. The processing time meets the application's requirements as only one image needs to go through the complete processing per grasp. The time consumption for processing one image is shown in Tab. 4.6.

Table 4.6: Inference speed of processing one 1280x760 pixel image using the vision-based grasp force planning method

Section	**Speed*(ms)**
ARUCO marker Detection	7
Object Detection	470
Material Classification	228
Size and Weight Estimation	3
Total	708

Speed*: the inference time is measured by averaging the inference time of ten images on a E5-1260 CPU.

4.3.8 Conclusion

This chapter presented a novel vision-based Human-Machine Interface (HMI) aimed at estimating the initial grasp force required to manipulate a target object using an assistive exoskeleton glove designed for patients with Brachial Plexus Injuries.

The proposed approach employed object detection and material classification techniques to predict the initial grasp force, using information about the weight, size, and material of the object. In the validation dataset, the object size estimation produced a Mean Absolute

Percentage Error (MAPE) of 26.9%, while the object weight estimation showed a MAPE of 59.8%. Although the MAPE of weight and size estimation was relatively high, vision-based initial grasp force estimation still managed to produce a meaningful result to assist grasping.

The vision-based HMI successfully distinguished between different materials and accurately predicted the initial grasp force for objects of varying weights. When integrated with the pure slip-grasp method, the combined approach attained an impressive 87.5% success rate, outperforming the standalone slip-grasp method (71.9%). These results highlighted the importance of estimating the initial grasp force to prevent slippage caused by inadequate or excessive application of force and torque.

Chapter 5

Voice-Based Human-Machine Interface (HMI) Design

5.1 Introduction

Based on findings from the literature review in Sec. 1.2.4, none of the currently available exoskeleton glove Human-Machine interfaces (HMI) incorporate speaker verification, which can lead to user authentication issues. Over the past few decades, researchers have worked extensively on developing speaker verification methods. This chapter specifically focuses on the application of existing speaker verification techniques for exoskeleton gloves, aiming to address the user authentication challenges associated with these devices.

5.2 Voice-Based HMI with Text-Independent Speaker Verification

Preliminary work and results related to Configurable Voice Activation and Speaker Verification (CVASV) HMI for the proposed exoskeleton glove are available in [85]. This voice-based HMI has not yet been fully tested using public and collected datasets. This section described additional experiments performed on the public VoxCeleb dataset [99] and human subjects

to justify the performance of the CVASV HMI.

5.2.1 Review of Configurable Voice Activation and Speaker Verification (CVASV) HMI

The CVASV HMI can perform voice activation when given a customized keyword and perform bio-authentication on each input command. This HMI contains two stages. First, the command recognition stage includes signal prepossessing, command aggregation, and Speech-To-Text (STT) conversion using Google's Web Speech Application Programming Interface (API). Second, the text-independent speaker verification stage contains a neural network feature extractor and similarity comparator. Stage one is running continuously to monitor the environment and perform the STT conversion. If the input command is valid, stage two is triggered by stage one to perform speaker verification. The CVASV HMI is designed to run on a mobile device where power consumption is critical. The two-stage design of this HMI can minimize the computation cost of the neural network by offloading the computation to the cloud server. Stage two also uses a low computational cost neural network to reduce the delay in the system along with power consumption. The overview of the CVASV HMI system is shown in Fig. 5.1.

Justification of MobileNet V1 Computational Cost

MobileNet uses a deep-wise separable convolution to replace the traditional convolution. The cost of a traditional convolution is shown in Eq. 5.1. The deep-wise separable convolution is shown in Eq. 5.2. Compared to traditional convolution, MobileNet's deep-wise separable convolution is faster on the basis of Eq. 5.3.

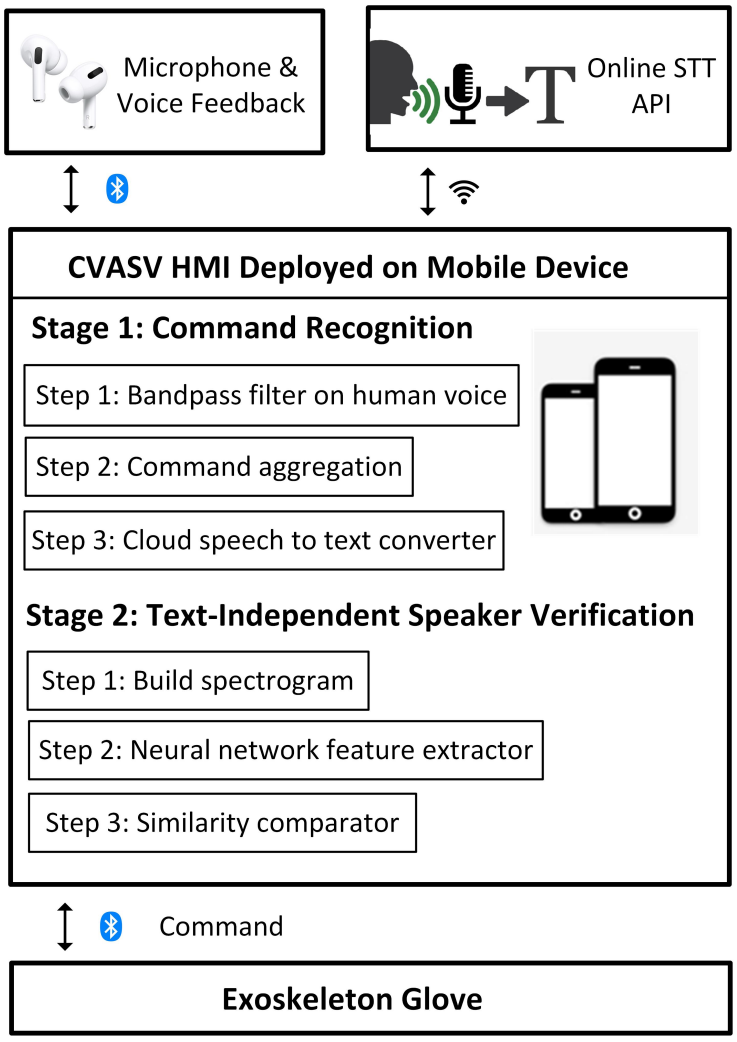


Figure 5.1: CVASV HMI system overview.

$$C_{traditional} = H_i \times W_i \times C_i \times H_k \times W_k \times C_o \tag{5.1}$$

$$C_{mobilenet} = H_i \times W_i \times C_i \times H_k \times W_k + C_i \times C_o \times H_i \times W_i \times W_k \tag{5.2}$$

$$\frac{C_{mobilenet}}{C_{traditional}} = \frac{1}{C_o} + \frac{1}{H_k \times W_k} \tag{5.3}$$

where, $C_{mobilenet}$ is the computation cost of MobileNetV1, H_i is the height of the input array, W_i is the width of the input array, C_i is the channel of the input array, H_k is the height of the kernel, W_k is the width of the kernel, and C_o is the channel of the output array.

The VGG-M neural network is used to perform speaker verification on the VOXCeleb dataset. The VGG-M network is modified on the basis of the image classification network VGG16. MobileNet-224 has a classification accuracy of 70.6%, which is comparable to the 71.5% accuracy of VGG16 in the ImageNet dataset. MobileNet-224 has only 4.2 million parameters, which is much less than VGG16's 138 million. Therefore, it is possible to use MobileNet to accelerate the VGG16-based speaker verification model.

Command Recognition

The command recognition stage is designed to detect commands when the user is speaking. This stage uses an online STT converter to extract the command content. This stage includes the following steps: signal prepossessing, command aggregation, and STT conversion using Google's Web Speech API. Each component is described in detail below.

First, using a Fourier transform, the signal prepossessing step converts the raw input from the microphone to the time-independent frequency domain. High-frequency and low-frequency data are considered noise and are subsequently trimmed from the spectrogram. The remaining frequencies contain potential human voices transferred back to the time domain. The data is then passed to the command aggregation step. The time-independent Fourier transform of the original and filtered audio data is shown in Fig. 5.2.

Second, the input audio is sliced into 0.5 seconds segments, and these audio segments are combined into complete voice commands based on sound intensity. This research assumed that users input commands in a clear, loud, and steady voice. Thus, the sound intensity

should increase if the user inputs the voice command, and the intensity will decrease after the input ends. The margin between words in a command sentence is usually 0.5 seconds at maximum. Based on these hypotheses, the audio aggregation feature is designed to monitor the intensity of every 0.5 seconds audio segment. If a sequence of audio segments has a higher intensity than a threshold, these audio segments are combined into a complete voice command and fed to the STT API. The audio aggregation feature is implemented according to the flowchart described in Fig. 5.2.

Third, the Google Web Speech API is used to recognize voice commands. After verifying that the trigger word or command is valid, the audio data are passed to the second stage for speaker verification.

Text-Independent Speaker Verification

The text-independent speaker verification method was improved based on the VGG-M speaker verification method by changing the VGG-M deep neural network into the low computational cost MobileNetV1 network. The size of the MobileNetV1 network was modified by reducing it to 75% of the original network by adjusting the alpha value in MobileNetV1 [100]. The input layer was adjusted to adapt 2D spectrogram input.

During training, the audio data are converted into a 2D spectrogram and fed into the MobileNetV1 network. The speakers' utterances are grouped using the Global Average Pooling (GAP) into 768 features and then classified using a dense layer. The neural network is trained as speaker classification model. The loss function used for classification was the standard Softmax loss function given by:

$$P(y = j|x) = \frac{e^{X^T W_j}}{\sum_{k=1}^{K} e^{X^T W_k}} \tag{5.4}$$

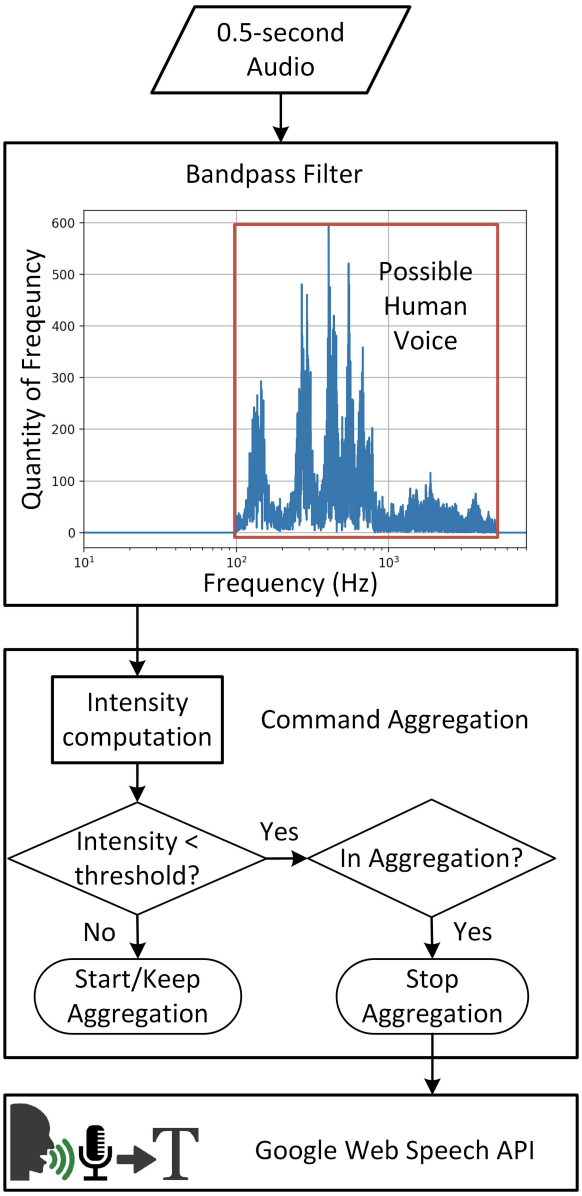


Figure 5.2: Flowchart of the command recognition stage in the CVASV HMI. Demonstrating bandpass filter and command aggregation logic.

where, x is the spectrogram input, j is the label of a certain speaker, X is the output of the feature of the neural network, k is the number of classes, and W is the weighting vector. The training process is shown in Fig. 5.3.

During speaker verification, the trained neural network is used as a feature extractor. The

last dense layer of the MobileNetV1 is detached. Each input audio that goes through the feature extractor will result in an array that contains 768 utterance features. Each speaker has a unique utterance feature vector. The feature vector A from an enrolled speaker and test-feature vector B from an unknown speaker are compared. If the distance is less than a threshold, these two audios are considered to be from the same speaker. The verification procedure is shown in Fig. 5.3. The distance between features is calculated by the cosine of the distance D_c defined as:

$$D_c(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5.5}$$

During the enrollment process, the user needs to record a x-second long enrollment audio. The audio is then randomly cut into n audio segments, each having a length of 3 seconds. The neural network processes these n audio clips, and each audio clip is turned into a feature vector F as discussed above . The average enrolled feature vector F_{avg_enroll} can be calculated using Eq. 5.6.

$$F_{avg_enroll} = \frac{\sum_{k=1}^{n} F_n}{n} \tag{5.6}$$

The average distance between the same speaker and different speakers in the verification training dataset can be calculated and used as a verification threshold. The verification training dataset contains K speakers. The threshold T is chosen by calculating the mean distance between the average enroll-feature F_{avg_enroll} and the average test-feature F_{avg_test} over K speakers, as shown in Eq. 5.7.

$$T = \frac{\sum_{k=1}^{K} D_c(F_{avg_enroll}, F_{avg_test})}{K} \tag{5.7}$$

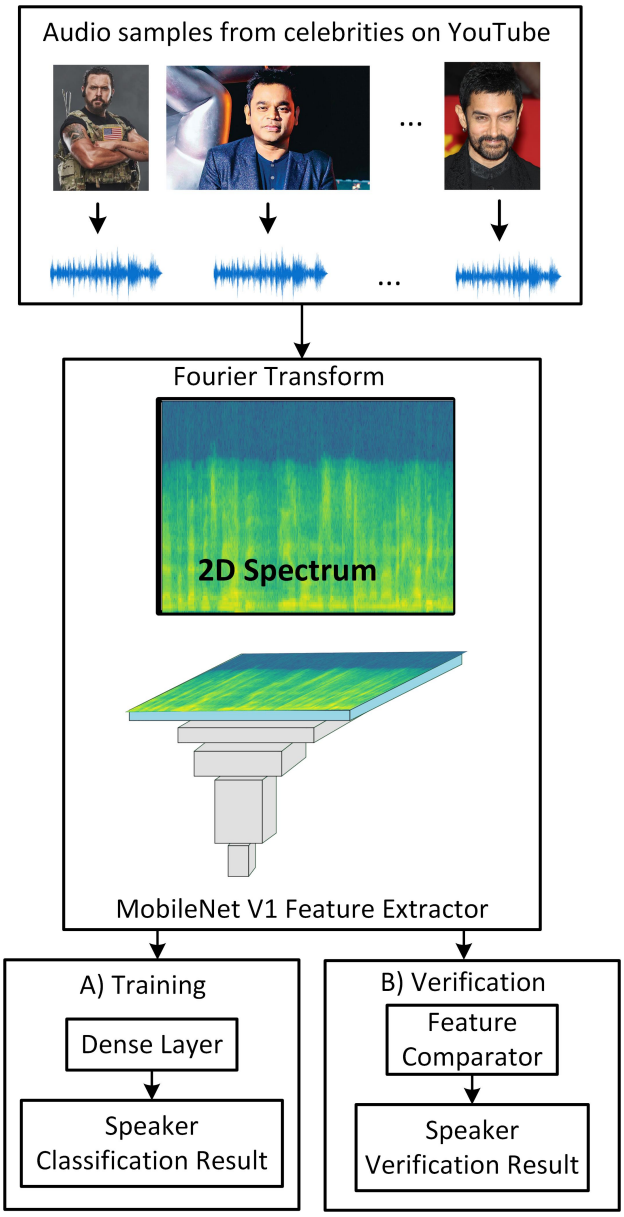


Figure 5.3: Speaker verification model training and inference procedure using MobileNetV1. (A) Train the MobileNetV1 feature extractor by performing speaker classification on the VoxCeleb1 dataset. (B) The last dense layer of the trained speaker classification model is removed. The model is used as speaker utterance feature extractor. The extracted features of the enrolled and tested speakers are compared to verify the enrolled speaker.

Table 5.1: Performance comparison between modified MobileNetV1 and the baseline VGG-M method

	MBN	**VGG-M**
Classification Accuracy	**80.0%**	79.2%
EER	12.4%	**10.0%**
Time Cost	**73.2ms**	88.9ms
Total Parameters	**1,832,544**	17,691,328

5.2.2 Experimental Results

Experiments were carried out to justify the effectiveness of the MobileNetV1 model in the aspect of speaker verification. In addition, the CVASV HMI was tested on the collected dataset and compared with other state-of-the-art voice-based HMIs.

MobileNet V1 Feature Extractor Performance

The feature extractor training is performed on the VoxCeleb1 dataset. The VGG-M method will be used as a performance baseline. The VoxCeleb1 [101] dataset contains 1,251 celebrities giving presentations in different environments from which more than 100,000 utterances and 40 speakers were chosen for the verification test. The dataset contains noisy data, which is suitable for training a robust speaker verification model. A 3 seconds audio clip is randomly extracted from each audio file and converted to a 512x300 spectrogram using Fourier transform. The input spectrogram is then classified into 1251 classes using MobileNetV1. The classification result is shown in Tab. 5.1.

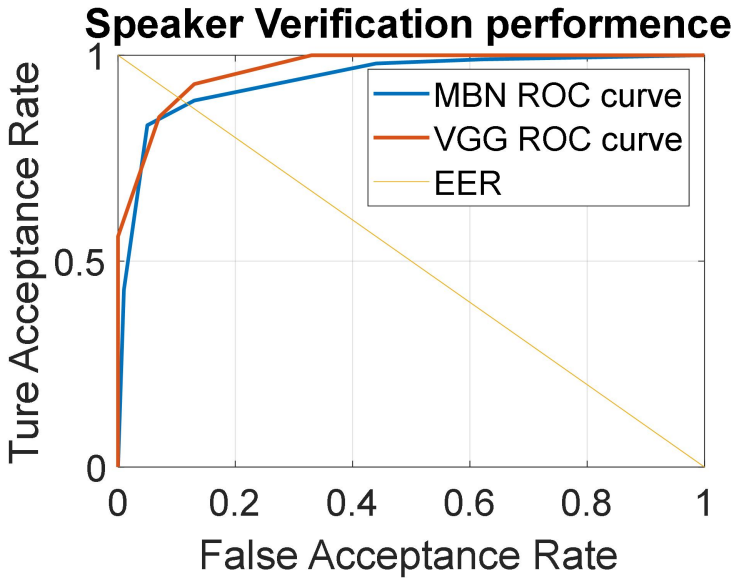


Figure 5.4: ROC curve of the proposed speaker verification method tested on VoxCeleb1 test dataset.

Speaker Verification Performance

The 40 speakers chosen to be used for the verification data have IDs from 270-309. Two audios are given during the verification testing process: one is used as enrollment data, and the other is used as test data. The verification result can be visualized by the Receiver Operating Characteristic (ROC) curve shown in Fig. 5.4

The performance of using MobileNetV1 is now compared to the original method, which uses VGG-M. The performance comparison between the MobileNetV1 (MBN) and the VGG-M is shown in Tab. 5.1. The modified MobileNetV1 achieves a 0.8% higher classification accuracy and has a 2.4% higher EER than the baseline VGG-M in the verification section. However, using MobileNetV1 as feature extractor reduces the total parameters in the neural network by 89.6% and was, on average, 17.7% faster to perform a verification than using VGG-M.

Tab. 5.2 compares the proposed speaker verification method with the state-of-the-art speaker verification methods tested on the Voxceleb1 dataset. The baseline method in this research

Table 5.2: Performance comparison between the modified MobileNetV1 and the state-of-the-art speaker verification methods on VoxCeleb1 dataset

Author	Network	Aggregation	EER
Proposed	MBN	Rand3S	12.4%
Proposed	VGG-M	Rand3S	10.0%
Nagrani et al.[102]	GMM-UBM	N/A	15.0%
Nagrani et al.[102]	I-vectors+PLDA	N/A	8.8%
Nagrani et al.[102]	VGG-M	TAP	10.2%
Chung et al.[103]	VGG-M-SC	TAP	5.94%
Xie et al.[104]	Thin-ResNet-34	TAP	10.48%
Xie et al.[104]	Thin-ResNet-34	GhostVLAD	3.22%

TAP: Temporal Average Pooling.
VGG-M-SC: VGG-M network with softmax and contrastive loss.
Rand3S: average of random 3-second audio clip.

has a slightly different EER than the EER reported by Nagrani et al. [102] due to the different aggregation methods. Nagrani et al. [102] featured a variable audio length for audio clip longer than 3 seconds. Whereas this research chose not to adopt different input lengths since this application is used for short commands less than 3 seconds long.

CVASV HMI performance

The collected speaker data contain voice from six males and four females, including Asian accents, Indian accents, and native North American English speakers. The collected data for each speaker contains a 20 seconds enrollment audio, one type of activation keyword, six types of grasp commands, and unknown commands. The chosen activation keyword is “Hey, glove”. The commands chosen are “Grasp toothbrush”, “Grasp bottle”, “Grasp cup”, and “Release”. Each commands are repeated 4-6 times by the speaker. The collected dataset is used to verify the feasibility of short command classification and verification.

The CVASV HMI is configured to use the Google Web Speech API for command classification. The classification accuracy is 94.1%. Successfully classified commands will enter the speaker verification section, which uses MobileNetV1 as a feature extraction network to extract utterance features and then uses a cosine distance function to measure the similarity of input and enrollment utterance features. The threshold for speaker verification is selected to be 0.30. In the validation experiment, voice commands from each speaker were tested against voice commands from other speakers. The experimental result is visualized using a confusion matrix shown in Fig. 5.5. Speaker verification performance is quantified using the True Acceptance (TA) rate and the False Acceptance (FA) rate. The CVASV HMI performance tested on the collected dataset is shown in Tab. 5.3.

Table 5.3: CVASV HMI performance tested on the collected dataset at cosine distance threshold = 0.30

Speaker	TA rate	FA rate
A	95%	3.1%
B	100%	9.3%
C	100%	10.7%
D	100%	8.6%
E	100%	12.4%
F	92%	4.9%
G	92%	4.3%
H	100%	11.1%
I	96%	7.6%
J	96%	6.3%

Fig. 5.6 shows the statistical results of seven additional voice commands used to test the CVASV HMI. These commands are later used in clinical experiments to test the exoskeleton glove in patients with BPI. Each command has been tested 20 times, 10 of which should be accepted and the other ten should be rejected. If a command is not recognized by the

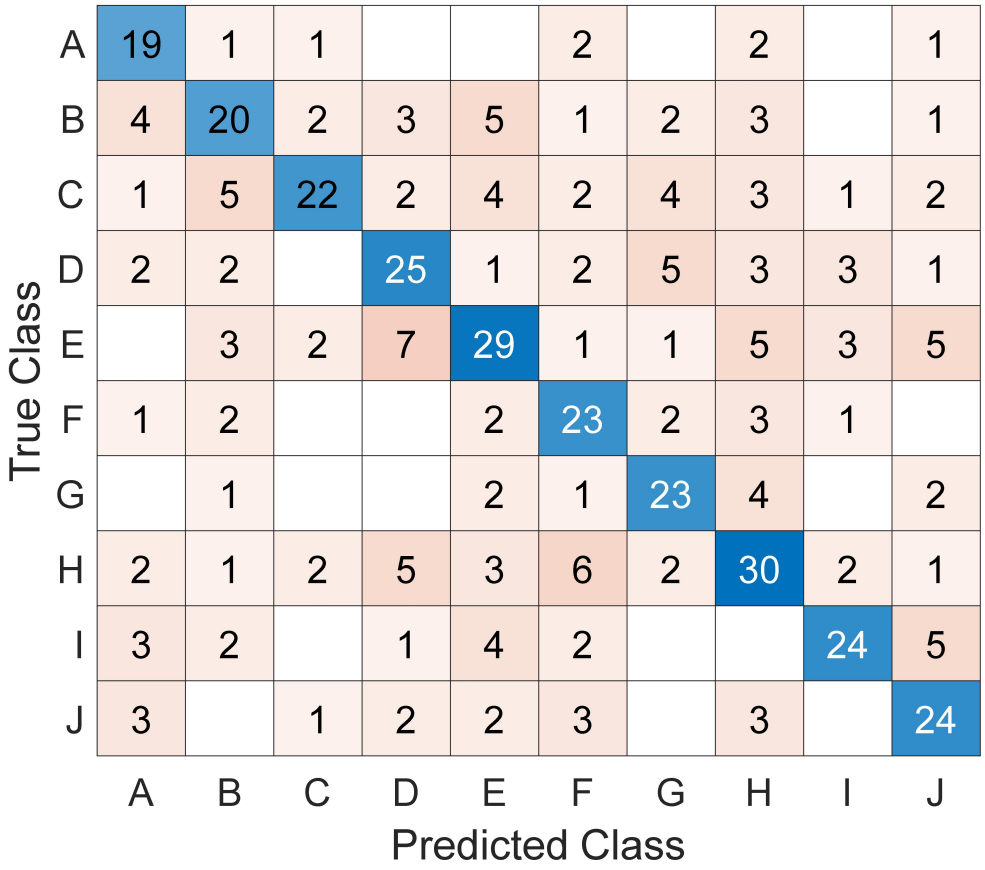


Figure 5.5: Confusion matrix of CVASV HMI validation experimental results on the collected dataset at cosine distance threshold = 0.30. Label A-J represents speaker A-J. The amount of true positive commands are marked in blue and the amount of false positive commands are marked in red.

STT conversion API, the command is classified as unrecognized. If a command is rejected but accepted, that command will be marked as falsely accepted. If a command should be accepted but rejected, that command will be marked as true rejected. The overall success rate is 89.3%.

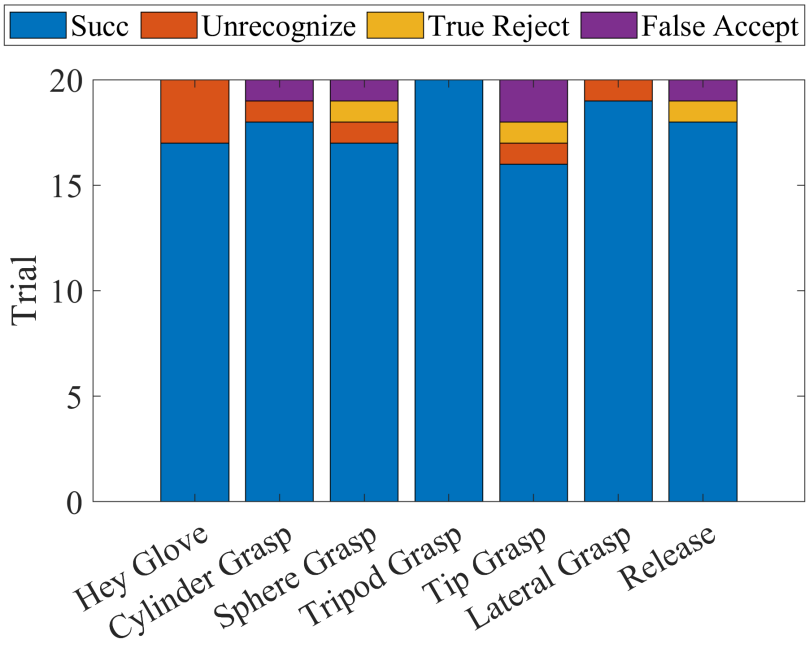


Figure 5.6: HMI operation success rate for 140 voice commands from 2 subjects. Each keyword is tested 20 times, 10 of which should be accepted, and the other ten should be rejected.

System Latency

The CVASV HMI is designed to run reliably in real time on a portable device. This experiment used a computer with a 2.2GHz six-core Intel i7 processor to simulate the mobile device. The program uses only one thread with a CPU frequency limited to 2GHz, and RAM limited to 2 gigabytes. The latency for each component is shown in Tab. 5.4. The entire system's latency to process one command is 182.58ms, faster than most HMIs, and can run in real-time on a portable device.

Energy Consumption

The CVASV HMI has not yet been deployed on a smartphone. The power consumption experiment is designed to approximate battery usage and improve the system before it is

Table 5.4: Time cost to process one audio command using CVASV HMI

Subsections of CVASV HMI	Time Cost
Noise Reduction Filter	2.12ms
Loudness Filter(D_e)	2.45ms
Audio Collection Subsystem	5.02ms
spectrogram Conversion	12.32ms
Voice Recognition API	34.47ms
MobileNetV1 Feature extractor	126.2ms
Voice Recognition Subsystem	182.58ms

deployed on a smartphone. In this experiment, a laptop with a 5953mAh, 11.4v battery is used. The computer screen was turned off during the experiment and no additional programs were running besides the CVASV HMI. The CPU frequency is limited to 2GHz, and the RAM is limited to 2 Gigabytes. The result is measured as a percentage of loss of battery. The idle status measured the power consumption when the system was placed in a noisy room for 2 hours. The active status measured the power consumption when the system was activated 50 times and three commands were inputted per activation. The results are shown in Tab. 5.5.

Table 5.5: Power consumption of CVASV HMI

HMI Status	Loss in Battery	Power(W)
Idle	9%	2.925
Active	21%	6.825

Comparison with Existing Voice-Based Exoskeleton Glove HMI

The CVASV HMI proposed in this chapter was proven to be fast, accurate, portable, and secure. The CVASV HMI can distinguish between different speakers and recognize various

Table 5.6: Comparison between CVASV HMI and other state-of-the-art voice-based HMI system

Author	Method	Acc*	CV-SV
Proposed	GoogleAPI+CNN	89.3%	Yes
He et al.[105]	GoogleAPI	92%	No
El-emary et al.[106]	GMM	<85%	No
Gomez et al.[107]	MG GMM+SM	88%	No
Gomez et al.[107]	HMM	100%	No
Megalingam et al.[108]	PocketSphinx:HMM	90%	No
Pleva et al.[109]	Julius: HMM	91%	No
Guo et al.[110]	LD3320 speech chip	94%	No

CV-SV: Customized voice activation and speaker verification.
Acc*: For HMI without speaker verification, Acc is command classification accuracy. For HMI with speaker verification, Acc is voice control successful rate at 0.3 thresholds.
GMM: Gaussian Mixture Model
MG GMM+SM: Mouth gesture based detection using GMM and state machine.
HMM: Hidden Markov Model.

commands. After being deployed in the initial prototype, the system achieved less than 200ms latency to process one command and has an average 89.3% success rate when tested by human subjects. Tab. 5.6 showed the performance comparison between the CVASV HMI and other state-of-the-art short voice command systems.

5.3 Voice-Based HMI with Text-Dependent Speaker Verification

The CVASV voice HMI is less practical than was expected during clinical experiments. It has two major issues. First, the CVASV HMI requires an internet connection to perform Speech-To-Text (STT) conversion and Natural Language Processing (NLP) to recognize

voice commands. When the Internet connection speed is low, the voice HMI takes more than 700 ms to respond. Second, the voice command is usually short, which causes the real-world text-independent speaker verification performance (22% EER) not to be as good as the performance in the VoxCeleb validation dataset (12.4% EER). Retrain the model by including the user's voice can improve the performance, but it takes more than 30 hours to retrain the model using GTX Titan XP GPU.

Based on the challenges mentioned above, the CVASV HMI has been modified in the following aspects. First, STT and NLP are replaced with a fixed command keyword detector that does not require an Internet connection. Second, fixed commands are used to operate the exoskeleton glove; thus, the text-independent speaker verification method can be replaced with a text-dependent speaker verification method based on one-shot learning. The detail of the above improvements is explained in the following sections.

5.3.1 Command Detection

In the previous CVASV HMI, STT and NLP methods were used to extract voice commands due to their robustness. For instance, phrases like "grasp a cup" or "grasp a water bottle" effectively triggered a cylinder grasp by the exoskeleton glove. However, based on valuable patient feedback, a more user-friendly approach was favored, allowing patients to directly choose the grasp type, typically involving shorter commands. Given that the exoskeleton used in this research accommodates five fundamental grasp types, users found it convenient to memorize these commands. To enhance the accuracy of fixed command detection, fixed commands were incorporated into the grammar of the STT and NLP models, effectively reducing the Word-Error-Rate (WER). This new approach was introduced midway through the clinical experiments and garnered positive feedback from patients.

Following the conclusion of the experiments, the need for employing the STT and NLP methods as a command detector became redundant due to the use of fixed commands. Researchers have explored various efficient approaches for fixed command detection, with two common methodologies prevailing. Some have advocated for a one-shot learning method, comparing the input voice command with enrolled commands [111]. Conversely, others have employed neural network methods as feature extractors, coupled with the Hidden Markov Model (HMM) approach to decipher the letter sequence of the input command [112, 113]. The field has witnessed extensive research efforts, resulting in the availability of multiple APIs. Notably, the Picovoice Porcupine wake word detection API has been favored over other commonly used APIs such as Pocketsphinx [114] or Snowboy [111], primarily due to its superior performance. This selection aligns with the API's minimal computational demands and its capacity to function without Internet connectivity.

5.3.2 Text-dependent Speaker Verification

Two prevalent text-dependent speaker verification approaches have been widely explored. The first approach embraces a one-shot learning methodology for text-dependent speaker verification [115, 116]. This approach offers the advantage of requiring significantly less data and training time, focusing solely on the development of a comparison network [117]. Nevertheless, it necessitates specific training data aligned with the verification command, which can be labor intensive to collect for each unique command.

In the second approach, some researchers have opted to split the text-dependent speaker verification task into two distinct tasks: Speech-to-Text (STT) and text-independent speaker verification. The text-independent speaker verification method leverages a Convolution Neural Network (CNN) feature extractor and a cosine distance comparator to differentiate be-

tween different speakers [118, 119]. This approach avoids the data availability issue by drawing from a text-independent speaker verification dataset like VoxCeleb. While theoretically more robust, practical performance is influenced by factors such as the length and type of voice commands, as well as the speaker's characteristics. This CNN-cosine comparator method was employed in the previous HMI.

To select the most suitable solution for this application, both methods are implemented and compared.

The Collected Speaker Verification Dataset

To evaluate the performance of both speaker verification methods, a dataset encompassing approximately 2000 voice commands from seven distinct subjects (referred to as Speaker A to G) was meticulously curated. It is noteworthy that all seven subjects share several common attributes: they are all males aged between 24 to 30, and they possess a Mandarin language background. Within this dataset, each subject contributed a set of five standardized commands, which include: "hey glove," "cylinder grasp," "tripod grasp," "lateral grasp," and "release object."

In the case of the one-shot learning method, the one-shot comparison network was trained using data from five specific speakers (Speakers A to E), while the remaining two speakers (F and G) were exclusively designated for validation purposes. The validation process involved executing speaker verification for each command uttered by Speakers F and G against the corresponding command articulated by Speakers A to E.

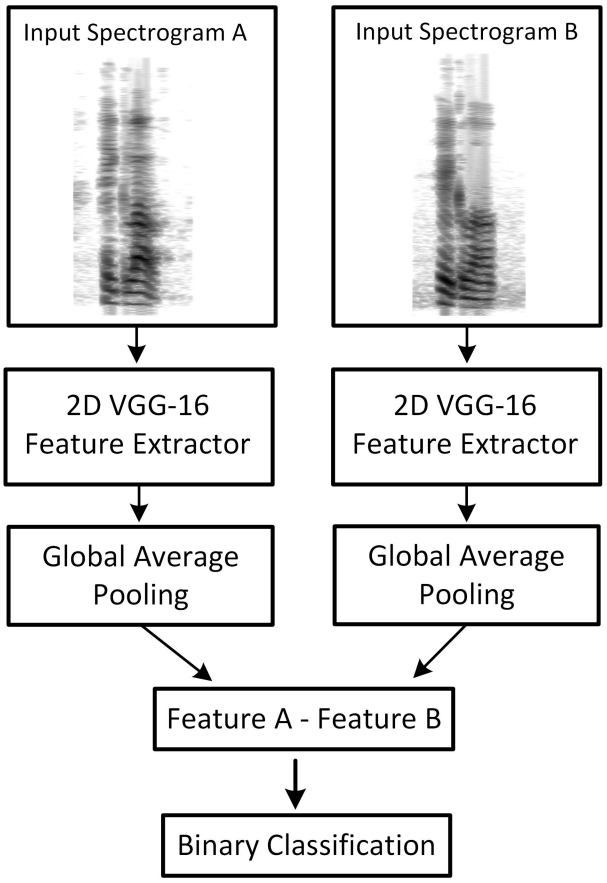


Figure 5.7: The neural network structure of the one-shot-based text-dependent speaker verification method.

One-shot-based Text-Dependent Speaker Verification

The essence of one-shot learning lies in the creation of a comparison neural network responsible for binary classification based on the similarity between two inputs. The structural components of the one-shot learning method are illustrated in Fig. 5.7. The algorithm can be dissected into two fundamental segments: preprocessing and the comparison neural network. Below, a closer examination is conducted to elucidate the design choices underpinning this framework.

First, voice commands are recorded using a single channel microphone at a sampling frequency of 16,000 Hz. To standardize input data, a fixed length L is established for each valid

voice command, depending on the length of the command itself. If the input voice command exceeds this predetermined length, it is trimmed at both ends. Conversely, if the input voice command falls short of L, zero-padding is employed to fill the void. To enable input compatibility with convolutional neural networks, the data undergoes a conversion process into a spectrogram. Two widely adopted audio preprocessing conversion methods were evaluated: Mel Frequency Cepstral Coefficients (MFCC) and Mel-spectrum (Mel). Both methods were rigorously assessed on speaker verification validation tasks, with performance gauged using the Equal Error Rate (EER). After meticulous tuning of the number of coefficients and Mel bands, the optimal results are presented in Tab. 5.7. It was discerned that the Mel-spectrum method outperformed the MFCC method.

Second, the performance of several commonly used comparator networks was analyzed, including VGG-16, MobileNet V2 (MBN V2), and Resnet-50 (Res-50). The results are summarized in Tab. 5.7. It was evident that the VGG-16 network achieved the lowest EER, establishing its superiority in this context.

Table 5.7: Comparison between different prepossessing techniques and comparator networks used for one-shot learning speaker verification

Preprocess	**Comparator Networks**	**EER**
MFCC	VGG-16	34%
Mel	VGG-16	23%
Mel	MBN V2	28%
Mel	Res-50	23%

Speaker Verification Comparison

This subsection evaluated the effectiveness of text-dependent and text-independent speaker verification by measuring their performance using the Equal Error Rate (EER). A lower EER indicates a more accurate speaker verification.Tab. 5.8 offers a comprehensive view of

the EER for speaker verification across two distinct commands. These specific commands have been chosen as illustrative examples to underscore the impact of command type and speaker variability on EER. Notably, both methods exhibit a similar average EER when considering all commands, as indicated in Tab. 5.8. It is difficult to come to a definite conclusion regarding which method is more effective due to the restricted amount of data available.

Table 5.8: Performance comparison between CNN-cosine speaker verification method and one-shot speaker verification method

Method	**Command**	**Task: speaker x vs. x**	**EER**
CNN-Cosine	Hey, glove	F vs. A-E	27%
CNN-Cosine	Hey, glove	G vs. A-E	18%
One-shot	Hey, glove	F vs. A-E	9%
One-shot	Hey, glove	G vs. A-E	32%
One-shot-retrain	Hey, glove	F vs. A-E	**9%**
One-shot-retrain	Hey, glove	G vs. A-E	25%
CNN-Cosine	Cylinder grasp	F vs. A-E	23%
CNN-Cosine	Cylinder grasp	G vs. A-E	22%
One-shot	Cylinder grasp	F vs. A-E	39%
One-shot	Cylinder grasp	G vs. A-E	21%
One-shot-retrain	Cylinder grasp	F vs. A-E	27%
One-shot-retrain	Cylinder grasp	G vs. A-E	**12%**
CNN-Cosine	**All**	F, G vs. A-E	22%
One-shot	**All**	F, G vs. A-E	23%
One-shot-retrain	**All**	F, G vs. A-E	**16%**

However, it is worth highlighting that the one-shot method emerges as the more suitable choice for this application. It achieved comparable performance to the CNN-cosine comparator method, while demanding significantly fewer data and shorter training times, as evidenced in Tab. 5.9. Additionally, inference times were measured by running both methods on an Intel E5-1260 CPU, while training times were determined by executing both

methods on an NVIDIA GTX Titan XP GPU. Furthermore, exploiting the rapid retraining capability of the one-shot method, which allows the inclusion of one of the held-out speakers for testing against other speakers, can enhance speaker verification EER on the collected dataset. As shown in Tab. 5.8, this enhancement leads to an improvement from 23% to 16% in the EER of speaker verification.

Table 5.9: Speaker verification system training and inference speed comparison

Method	**CNN-Cosine**	**One-shot**
Inference Speed	∼94ms	∼190ms
Training Time	30+hr	∼20min
Training Data Size	300,000+ utterances	∼2000 utterances

5.3.3 One-shot Learning Based Voice HMI System Design

The voice-controlled HMI was modified based on the previously mentioned discoveries. The design of the new voice HMI is shown in Fig. 5.8. It functions by capturing input from the microphone, recognizing the command through the Picovoice Porcupine wake-word API, and subsequently transmitting the recognized command to a one-shot learning speaker verification model.

5.3.4 Experimental Results

The speed comparison between the previous and new HMI is shown in Tab. 5.10. The inference time is measured by running both methods on an Intel E5-1260 CPU. The one-shot method is around 100ms slower than the previous CVASV method (assuming using an ideal internet connection) because all computations were performed locally.

A comparison of the processing speed between the previous and new HMIs is detailed in Tab.

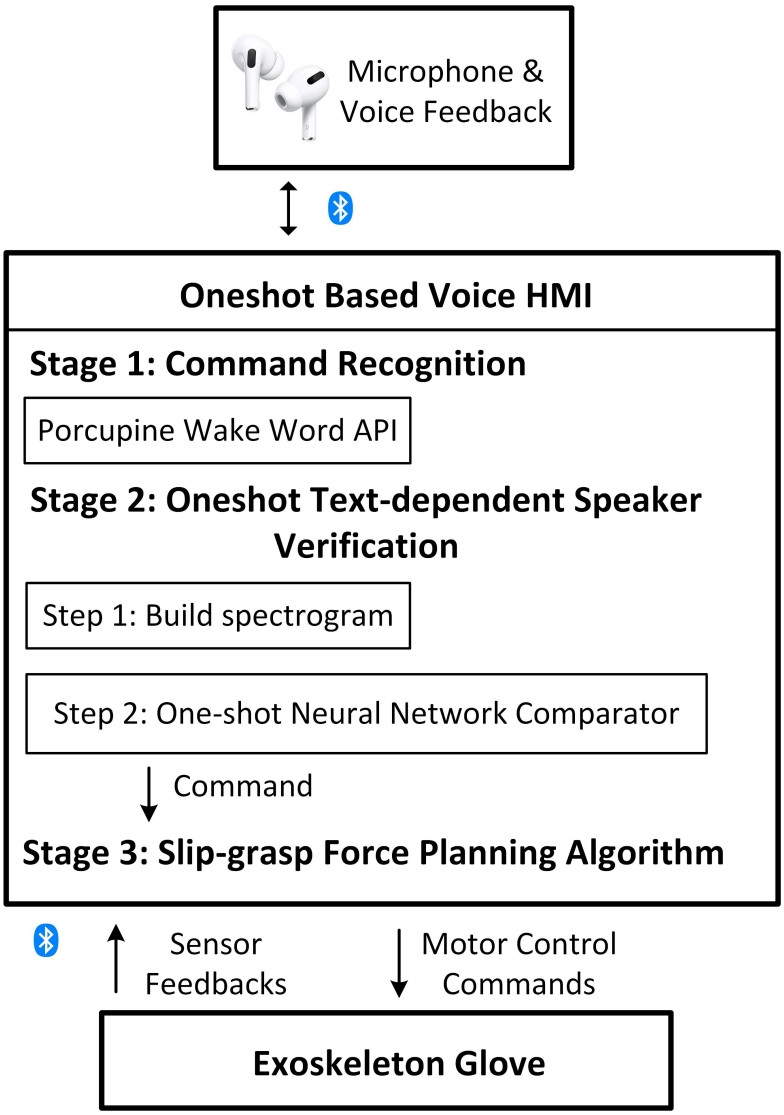


Figure 5.8: The structure of the modified voice-based HMI using the one-shot learning method.

5.10. The inference time was evaluated by executing both methods on an Intel E5-1260 CPU. It is noteworthy that the one-shot method was marginally slower by approximately 100 ms compared to the previous method, given the assumption of an ideal Internet connection. This discrepancy can be attributed to the fact that all computations were executed locally.

In addition, an experimental assessment was conducted on the complete HMI system, involving 200 voice commands sourced from two human subjects. This data set encompassed five

Table 5.10: HMI system processing speed comparison

	Google API+CNN-Cosine	**Porcupine+One-shot**
CR	58+ ms	∼110ms
SV	∼94ms	∼190ms
Total	152+ms	∼300ms

CR: Command Recognition.
SV: Speaker Verification.
The speed is measured by process one voice command.

types of command, each type comprising 20 trials per subject. Approximately 50% of the data was allocated to retraining the model. The resulting performance was tested against the previous HMI and other state-of-the-art HMIs, as illustrated in Tab. 5.6. The proposed HMI demonstrated an impressive recognition and verification rate of 94.5% in the 200 voice command tests.

Table 5.11: Comparison between the proposed HMI systems and other state-of-the-art voice-based HMI systems

Author	**Method**	**Acc***	**SV**
Proposed	Porcupine+One-shot	94.5%	Yes
Yunfei et al.[80] (old HMI)	GoogleAPI+CNN	89.3%	Yes
He et al.[105]	GoogleAPI	92%	No
El-emary et al.[106]	GMM	<85%	No
Gomez et al.[107]	MG GMM+SM	88%	No
Gomez et al.[107]	HMM	100%	No
Megalingam et al.[108]	PocketSphinx:HMM	90%	No
Pleva et al.[109]	Julius: HMM	91%	No
Guo et al.[110]	LD3320 speech chip	94%	No

SV: speaker verification.
Acc*: For HMI without speaker verification, Acc is command classification accuracy. For HMI with speaker verification, Acc is voice control successful rate.
GMM: Gaussian Mixture Model
MG GMM+SM: Mouth gesture based detection using GMM and state machine.
HMM: Hidden Markov Model.

Chapter 6

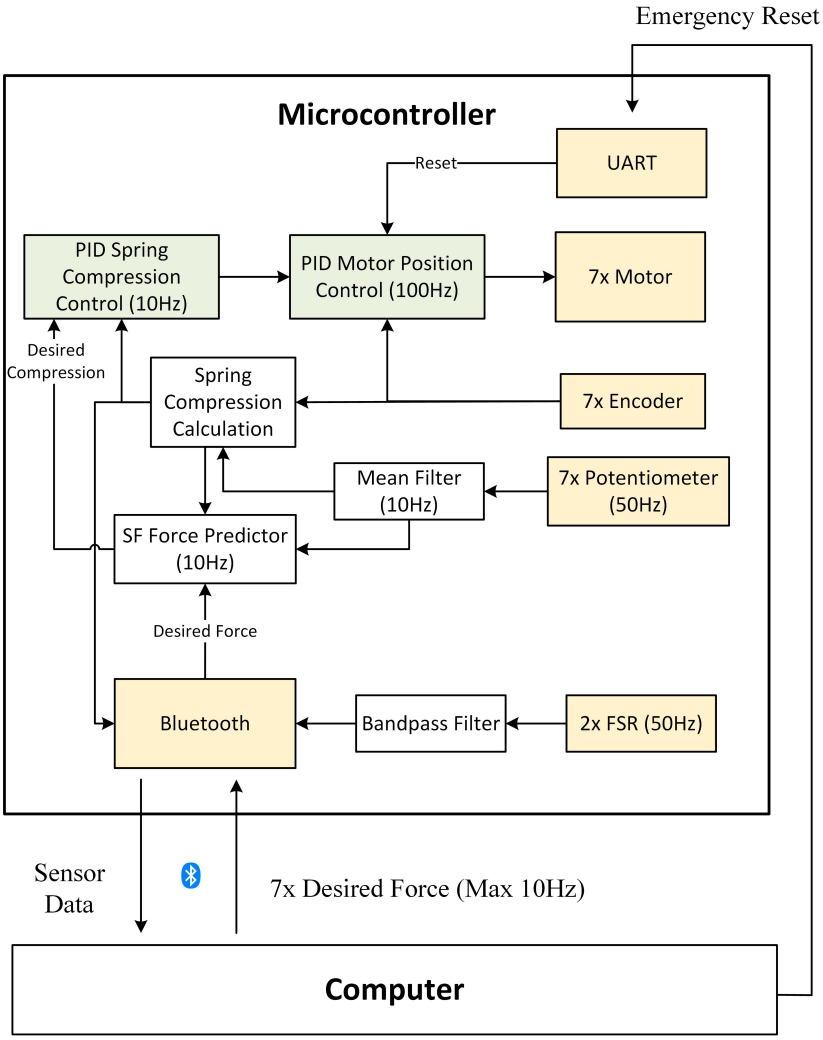
Exoskeleton Glove System Integration and Experiments

In this chapter, the system integration and experiments of the exoskeleton glove prototype are presented. The prototype was first used with human subject to perform experiments that helped validate its effectiveness. Subsequently, the prototype was used to perform clinical experiments with patients that suffer from Brachial Plexus Injuries (BPI).

The exoskeleton glove incorporates Linear Series Elastic Actuator (LSEA) force control using the Surface Fitting (SF) method, a hybrid slip-grasp force planning method, and a text-independent Configurable Voice Activation and Speaker Verification (CVASV) Human-Machine Interface (HMI). The selection of these control methods and HMI is aimed at achieving optimal stability and performance, ensuring that the glove can be safely and effectively used on patients during clinical experiments.

6.1 Control Architecture

The exoskeleton control system can be divided into a low-level controller running on the Teensy 4.1 microcontroller and a high-level controller running on a computer.



Sensor Data, consumed by slip-grasp force planning
2x Spring Compression (10Hz)
2x FSR Reading (50Hz)
Optional Sensor Data, used for monitor the exoskeleton glove
5x Spring Compression (10Hz)
7x Filtered Potentiometer (10Hz)
7x Encoder (10Hz)

Figure 6.1: Microcontroller software architecture. Hardware is marked in yellow. Controller is marked in green. The LSEA fingertip force prediction is running locally using SF method. The microcontroller sends sensor data to the computer and receives force command from the computer.

6.1.1 Low-level Controller

The low-level controller runs on a Teensy 4.1 microcontroller which is overclocked at 800MHz. FreeRTOS is used to perform multi-threading on Teensy 4.1 which increases code readability. The real-time system ensures that all computation can be performed in the designed time interval. The microcontroller software architecture is depicted in Fig. 6.2. The low-level controller serves three main functions.

First, the low-level controller controlled the SEA fingertip force through an open-loop control. The Surface Fitting SEA fingertip force predictor takes the desired fingertip force as input to predict the desired SEA spring compression based on sensor data. The force predictor is running at 10Hz. The desired spring compression is used as the setpoint of a PID SEA spring compression controller, which takes the calculated SEA spring compression as input and outputs the desired actuator input shaft position. The PID spring compression controller runs at 10Hz. The desired actuator input shaft position is used as the setpoint of a PID motor position controller, which takes encoder data as the input and output PWM signals to control the DC motor. The PID motor position controller is running at 100Hz.

Second, the microcontroller sends sensor data to the computer to perform slip-grasp force planning and receive a force command from the computer. The microcontroller sends the index and thumb finger FSR data to the computer at a rate of 50Hz. The spring compression of the index and rotatory SEA is sent to the computer at a rate of 10Hz. These data are consumed by the slip-grasp force planning program running on the computer. The desired fingertip force of each finger generated by the slip-grasp force planning method is sent to the microcontroller whenever a new force is generated. The force planning method will generate the desired force at a maximum rate of 10Hz. In addition, all sensor data can be sent to the computer at 10Hz as an option to monitor the exoskeleton glove.

Third, the exoskeleton glove has an emergency reset feature to ensure the user's safety during clinical experiment. The exoskeleton glove is equipped with an emergency stop button to shut off power. However, in most cases, an emergency reset button would be more suitable for this purpose when the user feels uncomfortable while grasping. The emergency reset feature will reset the actuator to its original position when it receives a reset command from UART.

6.1.2 High-level Controller

The high-level controller consists of two components. The first component is the CVASV HMI, which is capable of recognizing custom activation keywords and performing voice-based biometric authentication. The CVASV HMI will pass the grasp command to the second component: the slip-grasp force planning. The slip-grasp force planning program initiates the grasp according to the received grasp command. The desired fingertip force is sent to the microcontroller. Sensor data are received to detect slippage of the object. If the object slips while being lifted, the system will apply force to each SEA. The HMI and slip-grasp force planning run on the same computer in parallel, allowing voice commands to be received and executed while sending commands to the low-level controller.

6.2 Integration of CVASV HMI

The CVASV HMI was tested on a prototype to demonstrate its feasibility. Three healthy individuals participated in the experiment. Each of them wore the exoskeleton glove and performed a grasp without any intentional movement. Each object was grasped five times by each participant. For each trial, the exoskeleton glove was activated by the participant's

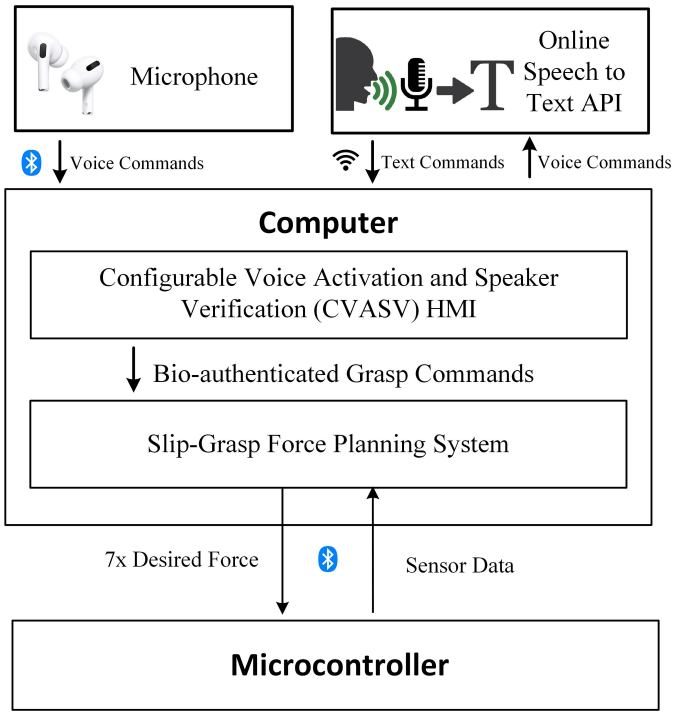


Figure 6.2: Computer software architecture. The speaker verification and slip-grasp force planning is performed on the computer.

voice command with the words "Hey, glove". After hearing the feedback "Activated" from the earphone, the participant had to indicate the type of grasp based on the target object by voice command. Upon receiving the feedback "Start" from the earphone, the exoskeleton glove would execute the grasp according to the specified type. The exoskeleton utilized a slip-grasp force planning method to ensure a stable grasp. The participants held the object and shook it to demonstrate a secure grasp. When the grasp was completed, the participant had to say "Release" to make the glove return to its original position. Fig. 6.4 shows the statistical results of the experiment.

6.3 Power Consumption

The average power consumption (P) of the exoskeleton during the grasp experiment is measured by monitoring the instant current (I_{inst}) of an external power source when performing the grasping experiments above and calculated using,

$$P = \frac{U \cdot \sum I_{inst}}{t} \tag{6.1}$$

where, U is the 11.1v supplied voltage, and t is the total time cost during the grasping.

The power consumption of the exoskeleton glove is listed in Tab. 6.1. The average power consumption for active grasping (P_{grasp}) is 4.68 watts and 1.665 watts for idle (P_{idle}). The battery time can be calculated using the formula,

$$E_{Wh} = Q_{mAh} \cdot \frac{U}{1000} = 11.655Wh \tag{6.2}$$

where, Q_{mAh} is 1050mAh and U is 11.1v supplied voltage.

The battery time during continuous grasping can be calculated by,

$$T_{batt} = \frac{E_{Wh}}{P_{grasp}} = 2.49h \tag{6.3}$$

Thus, the battery can hold up to 2.49 h of continuous operation or 6.67 h of idle.

6.4 Human Subject Experiments

Prior to conducting tests on patients with BPI, comprehensive evaluations were carried out on subjects with healthy hands to ensure consistent and reliable system performance. These

Table 6.1: Power consumption of the proposed exoskeleton glove

Status	Power (W)
Idle	1.665
Cylinder Grasp (209g)	4.368
Cylinder Grasp (400g)	4.548
Sphere Grasp (69g)	3.912
Tip Grasp (37g)	5.436
Tripod Grasp (48g)	4.044
Lateral Grasp (316g)	5.726

tests were conducted to verify that the system functions consistently as expected, providing a solid foundation before progressing to clinical experiments with BPI patients. Ten target objects (listed in Tab. 6.2) were selected to perform five different types of grasp: Cylinder grasp, Sphere grasp, Tripod grasp, Lateral grasp, and Tip grasp.

Table 6.2: Target objects selection

Name	Grasp Type	Weight (g)
Bottle	Cylinder	400
Jar	Cylinder	209
Tape	Tip	15
Small Box	Tip	5
Small ball	Tip	5
Screw Driver	Tripod	48
Pen	Tripod	15
Flat Box	Lateral	142
Bowl	Lateral	316
Ball	Sphere	316

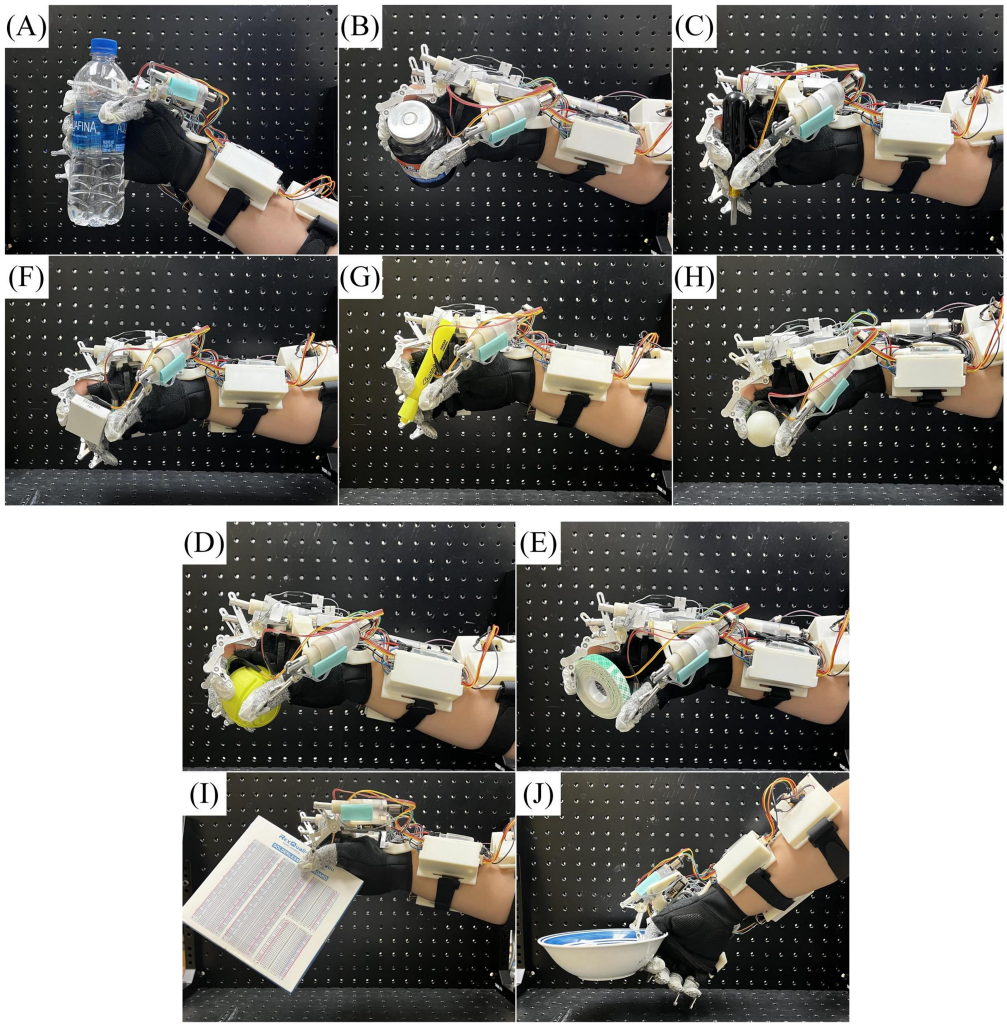


Figure 6.3: Examples of successful grasps. Each object is grasped five times by a single participant. (A) bottle, (B) jar, (C) screwdriver, (D) ball, (E) tape, (F) small box, (G) pen, (H) small ball, (I) flat box, (J) bowl.

6.4.1 Experiments on Subjects with Healthy Hand

The grasp experiments were performed on three healthy subjects. Fig. 6.4 shows the statistical results of 150 trials. Fig. 6.3 shows an example of a successful grasp with ten different objects.

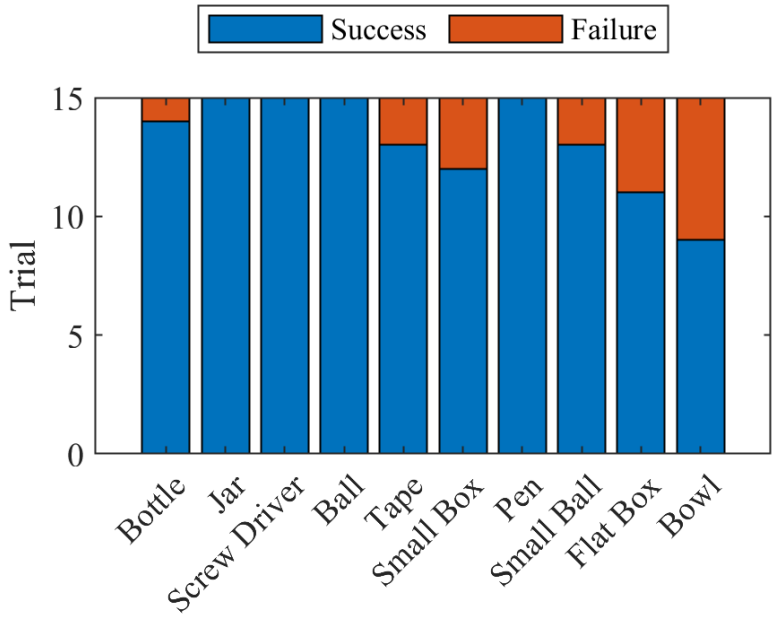


Figure 6.4: Grasp success rate for 150 grasping trails. This experiment is performed on 3 subjects with healthy hand.

6.4.2 Experiments on Subjects with BPI

Four clinical trials were conducted on three individuals with hand impairments caused by brachial plexus injury using the proposed exoskeleton. Three of these experiments were deemed successful, indicating that the proposed exoskeleton glove can partially restore the patient's grasp ability. Fig. 6.5 shows a comparison of the patient's grasping of objects with and without the exoskeleton gloves.

6.5 Comparison with State-of-the-Art Exoskeleton Gloves

The proposed prototype was compared to nine other devices, as shown in Tab. 6.3. It is noteworthy that the measurements of the peak fingertip force varied from device to device. The maximum posted pinch force or measured fingertip force was used for comparison. Most

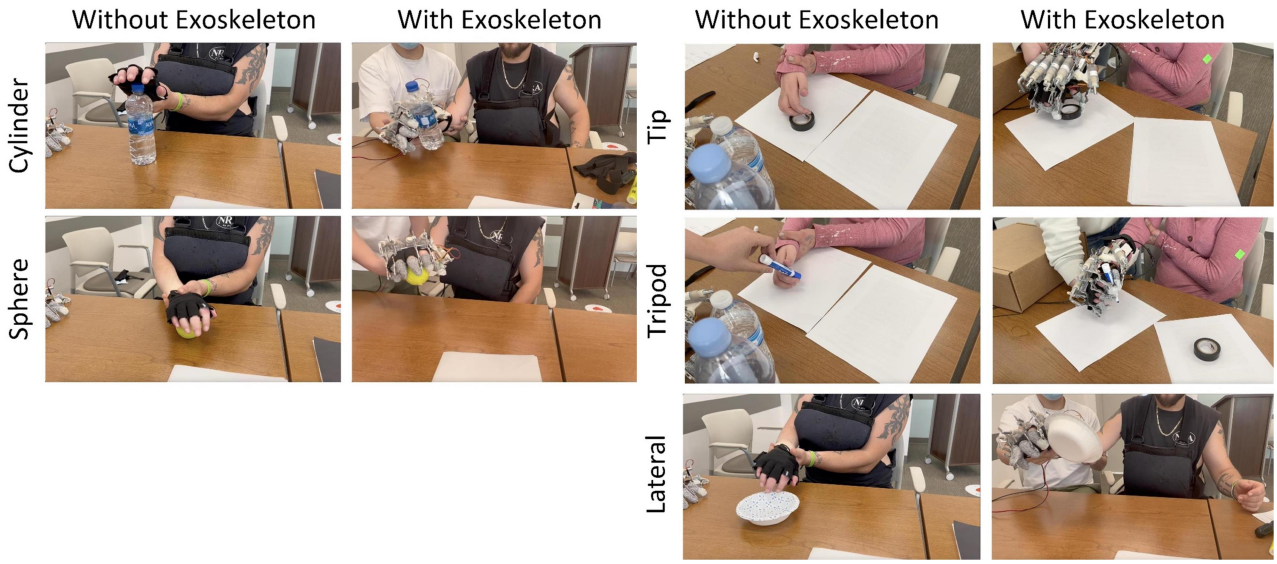


Figure 6.5: Comparison between grasping objects with and without the proposed exoskeleton glove. This experiment is performed on patients with BPI.

glove devices are lightweight, but require heavy actuators and are connected to a workstation. Only a few devices are designed to actuate all five fingers. Gloves that cannot actuate all fingers reduce the glove's capability. The comparison reveals that the proposed exoskeleton glove achieves an ideal balance between size, weight, and force output while still providing complete functionality, including force control and an intelligent Human-Machine Interface (HMI).

Table 6.3: Summary of hand exoskeleton grasp assistance systems

Device	Actuation, Transmission	Weight: glove/total	Peak Fingertip Force	Fingers	Wrist	Force/Torque Sensing	Intelligent Activation
Ryu et al.[1]	hydraulics, cable	2.62 kg / 2.74 kg	12 N	3	No	Yes	Yes
Nycz et al.[2]	linear motor, cable	113 g / 867 g	8.7 N	4	No	No	No
Gloreha Lite [3]	motor, cable	80 g / 5 kg	5 N	5	No	No	No
Hand of Hope [4]	linear motor, linkage	700 g / -	12 N	5	No	No	No
Dragusanu et al. [5]	linear motor, linkage	- / around 500 g	-	5	Yes	Yes	No
Ferguson et al. [6]	motor, cable	- / -	-	3	No	No	No
Graspy Glove [7]	motor, cable	250 g / 340 g	16 N	4	No	No	No
Agarwal et al. [8]	SEA, cable	80 g / -	-	1	No	Yes	No
KULEX Hand [9]	motor, linkage	- / 172 g	10 N	2	No	No	No
Proposed	**SEA, linkage**	**383 g/759 g**	**10 N**	**5**	**Yes**	**Yes**	**Yes**

Chapter 7

Conclusion and Future Work

7.1 Summary

This dissertation introduced a novel assistive robotic exoskeleton glove designed for patients with Brachial Plexus Injuries (BPI). It encompassed the development of force control methods, force planning techniques, and the design of a Human-Machine Interface (HMI).

In the design phase of the force control methods, a data-driven Series Elastic Actuator (SEA) fingertip force prediction method was developed to assist force control. This method used data-driven techniques to predict the fingertip force output of the Linear Series Elastic Actuator (LSEA) based on sensor readings. The proposed data-driven approach was advantageous because it accounted for friction forces and deformation in the exoskeleton glove linkages.

During the force planning phase, a slip-grasp force planning method was proposed to account for variations in the size and weight of the object. A vision-based grasp force estimation method was introduced to estimate the size and weight of the object, generating an estimated grasp force before grasping. This force estimation method mimicked the human grasping process, reducing the number of trials required for stable grasping using the slip-grasp technique.

In the HMI design phase, the Configurable Voice Activation and Speaker Verification (CVASV)

HMI was thoroughly tested and integrated into the exoskeleton glove. Alternative options for HMIs were also explored, with a one-shot learning based method identified as a superior alternative to CVASV. It achieved a similar Equal Error Rate (EER) without the need for an Internet connection and required less training time.

In conclusion, force control, force planning, and HMI were successfully integrated into the exoskeleton glove prototype, which was then used for clinical experiments to validate its effectiveness in assisting patients with BPI. This dissertation contributed significantly to the field of assistive robotics by presenting the developmental processes of an innovative exoskeleton glove that offered improved functionalities to address the specific needs of patients with BPI. Advancements in force control methods, force planning techniques, and HMIs provided a foundation for future research and development in robotic exoskeletons for assistive purposes.

7.2 Directions for Future Work

This research can be further improved in two ways. Firstly, it is essential to assess the proposed oneshot learning based voice HMI quantitatively and to obtain feedback from subjects with BPI. Secondly, the force planning method can be extended by including more hand movements. At present, the vision-based force estimation and hybrid slip-grasp method only covers grasp and lift motions, but it would be advantageous to consider a wider range of movements, such as shaking and flipping, in the future. This expansion of motion could lead to a more reliable force planning method. By addressing these two aspects, future work can enhance the performance of the HMI system and broaden the scope of force planning.

Bibliography

[1] 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 *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3028–3033, IEEE, 2008.

[2] C. J. Nycz, T. Bützer, O. Lambercy, 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*, vol. 1, no. 2, pp. 976–983, 2016.

[3] "Gloreha glove.." https://www.gloreha.com/. Accessed: 2022-02-02.

[4] X. Hu, K. Tong, X. Wei, W. Rong, E. Susanto, and S. Ho, "The effects of post-stroke upper-limb training with an electromyography (emg)-driven hand robot," *Journal of Electromyography and Kinesiology*, vol. 23, no. 5, pp. 1065–1074, 2013.

[5] M. Dragusanu, M. Z. Iqbal, T. L. Baldi, D. Prattichizzo, and M. Malvezzi, "Design, development, and control of a hand/wrist exoskeleton for rehabilitation and training," *IEEE Transactions on Robotics*, 2022.

[6] P. W. Ferguson, B. Dimapasoc, Y. Shen, and J. Rosen, "Design of a hand exoskeleton for use with upper limb exoskeletons," in *International Symposium on Wearable Robotics*, pp. 276–280, Springer, 2018.

[7] D. Popov, I. Gaponov, and J.-H. Ryu, "Portable exoskeleton glove with soft struc-

ture for hand assistance in activities of daily living," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 865–875, 2016.

[8] P. Agarwal, J. Fox, Y. Yun, M. K. O'Malley, and A. D. Deshpande, "An index finger exoskeleton with series elastic actuation for rehabilitation: Design, control and performance characterization," *The International Journal of Robotics Research*, vol. 34, no. 14, pp. 1747–1772, 2015.

[9] M. B. Hong, S. J. Kim, Y. S. Ihn, G.-C. Jeong, and K. Kim, "Kulex-hand: An underactuated wearable hand for grasping power assistance," *IEEE Transactions on Robotics*, vol. 35, no. 2, pp. 420–432, 2018.

[10] P. Agarwal, J. Fox, Y. Yun, M. K. O'Malley, and A. D. Deshpande, "An index finger exoskeleton with series elastic actuation for rehabilitation: Design, control and performance characterization," *International Journal of Robotics Research*, vol. 34, no. 14, pp. 1747–1772, 2015.

[11] M. Bianchi, M. Cempini, R. Conti, E. Meli, A. Ridolfi, N. Vitiello, and B. Allotta, "Design of a Series Elastic Transmission for hand exoskeletons," *Mechatronics*, vol. 51, no. July 2017, pp. 8–18, 2018.

[12] I. Jo and J. Bae, "Design and control of a wearable and force-controllable hand exoskeleton system," *Mechatronics*, vol. 41, pp. 90–101, 2017.

[13] E. M. Refour, B. Sebastian, R. J. Chauhan, and P. Ben-Tzvi, "A General Purpose Robotic Hand Exoskeleton With Series Elastic Actuation," *Journal of Mechanisms and Robotics*, vol. 11, no. 6, 2019.

[14] W. Xu, S. Pradhan, Y. Guo, B.-T. Pinhas, and C. Bravo, "A Novel Design of A Robotic Glove System For Paitients with Brachial Plexus Injuries," in *Proceedings of the 2020*

ASME IDETC/CIE, 44th Mechanisms {\&} Robotics Conference, (St. Louis, MO), 2020.

[15] L. Randazzo, I. Iturrate, S. Perdikis, and J. d. R. Millán, "mano: A wearable hand exoskeleton for activities of daily living and neurorehabilitation," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 500–507, 2017.

[16] C. E. Bouton, A. Shaikhouni, N. V. Annetta, M. A. Bockbrader, D. A. Friedenberg, D. M. Nielson, G. Sharma, P. B. Sederberg, B. C. Glenn, W. J. Mysiw, *et al.*, "Restoring cortical control of functional movement in a human with quadriplegia," *Nature*, vol. 533, no. 7602, pp. 247–250, 2016.

[17] X. Wang, P. Tran, S. M. Callahan, S. L. Wolf, and J. P. Desai, "Towards the development of a voice-controlled exoskeleton system for restoring hand function," in *2019 International Symposium on Medical Robotics (ISMR)*, pp. 1–7, 2019.

[18] L. Gerez, J. Chen, and M. Liarokapis, "On the development of adaptive, tendon-driven, wearable exo-gloves for grasping capabilities enhancement," *IEEE Robotics and Automation letters*, vol. 4, no. 2, pp. 422–429, 2019.

[19] D. Kim, B. Kang, K. B. Kim, H. Choi, J. Ha, K.-J. Cho, and S. Jo, "Eyes are faster than hands: A soft wearable robot learns user intention from the egocentric view," *Science Robotics*, p. 2949, 01 2019.

[20] C. J. González Lemus and F. X. Romero Prieto, "First study of the prevalence and characterization of brachial plexus injuries in guatemala," *Journal of Brachial Plexus and Peripheral Nerve Injury (JBPPNI)*, vol. 16, no. 01, pp. e46–e50, 2021.

[21] L. Rasulić, A. Savić, B. Živković, F. Vitošević, M. Mićović, V. Baščarević, V. Puzović, N. Novaković, M. Lepić, M. Samardžić, *et al.*, "Outcome after brachial plexus injury

surgery and impact on quality of life," *Acta neurochirurgica*, vol. 159, pp. 1257–1264, 2017.

[22] Q. A. Boser, M. R. Dawson, J. S. Schofield, G. Y. Dziwenko, and J. S. Hebert, "Defining the design requirements for an assistive powered hand exoskeleton: A pilot explorative interview study and case series," *Prosthetics and Orthotics International*, p. 0309364620963943, 2020.

[23] W. Xu, Y. Guo, C. Bravo, and P. Ben-Tzvi, "Design, control, and experimental evaluation of a novel robotic glove system for patients with brachial plexus injuries," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1637–1652, 2023.

[24] W. Xu, Y. Liu, and P. Ben-Tzvi, "Development of a novel low-profile robotic exoskeleton glove for patients with brachial plexus injuries," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11121–11126, 2022.

[25] E. K. Jian, D. Gouwanda, T. K. Kheng, *et al.*, "Wearable hand exoskeleton for activities of daily living," in *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, pp. 221–225, IEEE, 2018.

[26] P. Tran, D. Elliott, K. Herrin, and J. P. Desai, "Towards comprehensive evaluation of the flexotendon glove-iii: a case series evaluation in pediatric clinical cases and able-bodied adults," *Biomedical Engineering Letters*, pp. 1–10, 2023.

[27] N. Smania, G. Berto, E. La Marchina, C. Melotti, A. Midiri, L. Roncari, A. Zenorini, P. Ianes, A. Picelli, A. Waldner, *et al.*, "Rehabilitation of brachial plexus injuries in adults and children," *Eur J Phys Rehabil Med*, vol. 48, no. 3, pp. 483–506, 2012.

[28] T. Bützer, O. Lambercy, J. Arata, and R. Gassert, "Fully wearable actuated soft

exoskeleton for grasping assistance in everyday activities," *Soft robotics*, vol. 8, no. 2, pp. 128–143, 2021.

[29] Y. Chen, Z. Yang, and Y. Wen, "A soft exoskeleton glove for hand bilateral training via surface emg," *Sensors*, vol. 21, no. 2, p. 578, 2021.

[30] R. A. Bos, C. J. Haarman, T. Stortelder, K. Nizamis, J. L. Herder, A. H. Stienen, and D. H. Plettenburg, "A structured overview of trends and technologies used in dynamic hand orthoses," *Journal of NeuroEngineering and Rehabilitation*, vol. 13, no. 1, pp. 1–25, 2016.

[31] 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*, vol. 73, pp. 135–143, 2015.

[32] S. Kudo, K. Oshima, M. Arizono, Y. Hayashi, and S. Moromugi, "Electric-powered glove for CCI patients to extend their upper-extremity function," in *2014 IEEE/SICE International Symposium on System Integration, SII 2014*, pp. 638–643, Institute of Electrical and Electronics Engineers Inc., 1 2014.

[33] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *The International Journal of Robotics Research*, vol. 35, pp. 161–185, 1 2016.

[34] N. Takahashi, S. Furuya, and H. Koike, "Soft exoskeleton glove with human anatomical architecture: production of dexterous finger movements and skillful piano performance," *IEEE Transactions on Haptics*, vol. 13, no. 4, pp. 679–690, 2020.

[35] H. K. Yap, B. W. Ang, J. H. Lim, J. C. Goh, and C. H. Yeow, "A fabric-regulated soft robotic glove with user intent detection using EMG and RFID for hand assistive appli-

cation," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, pp. 3537–3542, Institute of Electrical and Electronics Engineers Inc., 6 2016.

[36] L. Gerez, A. Dwivedi, and M. Liarokapis, "A hybrid, soft exoskeleton glove equipped with a telescopic extra thumb and abduction capabilities," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9100–9106, IEEE, 2020.

[37] A. Mohammadi, J. Lavranos, P. Choong, and D. Oetomo, "Flexo-glove: a 3d printed soft exoskeleton robotic glove for impaired hand rehabilitation and assistance," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2120–2123, IEEE, 2018.

[38] P. Agarwal, P.-H. Kuo, R. R. Neptune, and A. D. Deshpande, "A novel framework for virtual prototyping of rehabilitation exoskeletons," in *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, pp. 1–6, IEEE, 2013.

[39] Z. Ma, P. Ben-Tzvi, and J. Danoff, "Sensing and Force-Feedback Exoskeleton Robotic (SAFER) Glove Mechanism for Hand Rehabilitation," *Proceedings of the ASME Design Engineering Technical Conference*, vol. 5A-2015, pp. 1–8, 2015.

[40] M. Aiple and A. Schiele, "Pushing the limits of the CyberGrasp™ for haptic rendering," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3541–3546, 2013.

[41] D. Leonardis, M. Barsotti, C. Loconsole, M. Solazzi, M. Troncossi, C. Mazzotti, V. P. Castelli, C. Procopio, G. Lamola, C. Chisari, M. Bergamasco, and A. Frisoli, "An EMG-controlled robotic hand exoskeleton for bilateral rehabilitation," *IEEE Transactions on Haptics*, vol. 8, no. 2, pp. 140–151, 2015.

[42] N. S. Ho, K. Y. Tong, X. L. Hu, K. L. Fung, X. J. Wei, W. Rong, and E. A. Susanto, "An EMG-driven exoskeleton hand robotic training device on chronic stroke subjects: Task training system for stroke rehabilitation," in *IEEE International Conference on Rehabilitation Robotics*, 2011.

[43] I. Sarakoglou, A. Brygo, D. Mazzanti, N. G. Hernandez, D. G. Caldwell, and N. G. Tsagarakis, "Hexotrac: A highly under-actuated hand exoskeleton for finger tracking and force feedback," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1033–1040, IEEE, 2016.

[44] M. Malvezzi, T. L. Baldi, A. Villani, F. Ciccarese, and D. Prattichizzo, "Design, development, and preliminary evaluation of a highly wearable exoskeleton," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1055–1062, IEEE, 2020.

[45] 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*, vol. 17, no. 5, pp. 884–894, 2011.

[46] T. Vanteddu, B. Sebastian, and P. Ben-Tzvi, "Design optimization of rml glove for improved grasp performance," in *Proceedings of the ASME 2018 Dynamic Systems and Control Conference, DSCC 2018*, vol. 1, pp. 1–8, 2018.

[47] 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*, vol. 24, no. 12, pp. 1323–1332, 2016.

[48] Z. Ma and P. Ben-tzvi, "Sensing and force-feedback exoskeleton (safe) glove," in

Proceedings of the 2015 ASME IDETC/CIE, 39th Mechanisms Robotics Conference, (Boston, MA), 2015.

[49] E. Refour, B. Sebastian, and P. Ben-Tzvi, "Two-digit robotic exoskeleton glove mechanism: Design and integration," *Journal of Mechanisms and Robotics*, vol. 10, 4 2018.

[50] M. Bouzit, G. Burdea, G. Popescu, and R. Boian, "The rutgers master ii-new design force-feedback glove," *IEEE/ASME Transactions on mechatronics*, vol. 7, no. 2, pp. 256–263, 2002.

[51] Z. Sun, X. Miao, and X. Li, "Design of a bidirectional force feedback dataglove based on pneumatic artificial muscles," in *2009 international conference on mechatronics and automation*, pp. 1767–1771, IEEE, 2009.

[52] D. Chiaradia, M. Xiloyannis, M. Solazzi, L. Masia, and A. Frisoli, "Comparison of a soft exosuit and a rigid exoskeleton in an assistive task," in *Wearable Robotics: Challenges and Trends: Proceedings of the 4th International Symposium on Wearable Robotics, WeRob2018, October 16-20, 2018, Pisa, Italy 3*, pp. 415–419, Springer, 2019.

[53] P. Tran, S. Jeong, K. R. Herrin, and J. P. Desai, "Hand exoskeleton systems, clinical rehabilitation practices, and future prospects," *IEEE Transactions on Medical Robotics and Bionics*, vol. 3, no. 3, pp. 606–622, 2021.

[54] J. Iqbal, H. Khan, N. G. Tsagarakis, and D. G. Caldwell, "A novel exoskeleton robotic system for hand rehabilitation–conceptualization to prototyping," *Biocybernetics and biomedical engineering*, vol. 34, no. 2, pp. 79–89, 2014.

[55] D. Leonardis, M. Barsotti, C. Loconsole, M. Solazzi, M. Troncossi, C. Mazzotti, V. P. Castelli, C. Procopio, G. Lamola, C. Chisari, *et al.*, "An emg-controlled robotic hand

exoskeleton for bilateral rehabilitation," *IEEE transactions on haptics*, vol. 8, no. 2, pp. 140–151, 2015.

[56] S. Soekadar, M. Witkowski, C. Gómez, E. Opisso, J. Medina, M. Cortese, M. Cempini, M. C. Carrozza, L. Cohen, N. Birbaumer, *et al.*, "Hybrid eeg/eog-based brain/neural hand exoskeleton restores fully independent daily living activities after quadriplegia," *Science Robotics*, vol. 1, no. 1, p. eaag3296, 2016.

[57] J. Lee and P. Ben-Tzvi, "Design of a wearable 3-DOF forearm exoskeleton for rehabilitation and assistive purposes," in *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, vol. 3, pp. 1–10, 2017.

[58] T. Vanteddu and P. Ben-Tzvi, "Stable grasp control with a robotic exoskeleton glove," *Journal of Mechanisms and Robotics*, vol. 12, no. 6, p. 061015, 2020.

[59] J. A. Díez, A. Blanco, J. M. Catalán, F. J. Badesa, L. D. Lledó, and N. García-Aracil, "Hand exoskeleton for rehabilitation therapies with integrated optical force sensor," *Advances in Mechanical Engineering*, vol. 10, no. 2, pp. 1–11, 2018.

[60] G. A. Pratt and M. M. Williamson, "Series elastic actuators," *IEEE International Conference on Intelligent Robots and Systems*, vol. 1, pp. 399–406, 1995.

[61] H. g. Kim, J. w. Lee, J. Jang, S. Park, and C. Han, "Design of an exoskeleton with minimized energy consumption based on using elastic and dissipative elements," *International Journal of Control, Automation and Systems*, vol. 13, no. 2, pp. 463–474, 2015.

[62] N. C. Karavas, N. G. Tsagarakis, and D. G. Caldwell, "Design, modeling and control of a series elastic actuator for an assistive knee exoskeleton," *Proceedings of the IEEE RAS*

and EMBS International Conference on Biomedical Robotics and Biomechatronics, pp. 1813–1819, 2012.

[63] Y. Guo, W. Xu, S. Pradhan, C. Bravo, and P. Ben-Tzvi, "Data driven calibration and control of compact lightweight series elastic actuators for robotic exoskeleton gloves," *IEEE Sensors Journal*, vol. 21, no. 19, pp. 21120–21130, 2021.

[64] W. Xu, S. Pradhan, Y. Guo, C. Bravo, and P. Ben-Tzvi, "A novel design of a robotic glove system for patients with brachial plexus injuries," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 83990, p. V010T10A042, American Society of Mechanical Engineers, 2020.

[65] J. B. Lee, "Development of Intelligent Exoskeleton Grasping Through Sensor Fusion and Slip Detection Development of Intelligent Exoskeleton Grasping," Master's thesis, Virginia Polytechnic Institute and State University, 2018.

[66] Y. Chen, Z. Yang, and Y. Wen, "A soft exoskeleton glove for hand bilateral training via surface emg," *Sensors*, vol. 21, no. 2, 2021.

[67] S. Cheon, D. Kim, S. Kim, B. Kang, J. Lee, H. Gong, S. Jo, K.-J. Cho, and J. Ahn, "Single emg sensor-driven robotic glove control for reliable augmentation of power grasping," *IEEE Transactions on Medical Robotics and Bionics*, vol. PP, pp. 1–1, 12 2020.

[68] K. Li, Z. Li, H. Zeng, and N. Wei, "Control of newly-designed wearable robotic hand exoskeleton based on surface electromyographic signals," *Frontiers in Neurorobotics*, vol. 15, p. 121, 2021.

[69] Y. Yun, S. Dancausse, P. Esmatloo, A. Serrato, C. A. Merring, P. Agarwal, and A. D. Deshpande, "Maestro: An emg-driven assistive hand exoskeleton for spinal cord in-

jury patients," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2904–2910, 2017.

[70] T. Lalitharatne, K. Teramoto, Y. Hayashi, and K. Kiguchi, "Towards hybrid eeg-emg-based control approaches to be used in bio-robotics applications: Current status, challenges and future directions," *Paladyn, Journal of Behavioral Robotics*, vol. 4, pp. 147–154, 12 2013.

[71] F. . Cheng and D. E. Orin, "Efficient algorithm for optimal force distribution-the compact-dual lp method," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 2, pp. 178–187, 1990.

[72] Youshen Xia, Jun Wang, and Lo-Ming Fok, "Grasping-force optimization for multifingered robotic hands using a recurrent neural network," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 549–554, 2004.

[73] C. Xiong and Y. Xiong, "Neural-network based force planning for multifingered grasp," *Robotics and Autonomous Systems*, vol. 21, no. 4, pp. 365–375, 1997.

[74] H. Liang, L. Cong, N. Hendrich, S. Li, F. Sun, and J. Zhang, "Multifingered grasping based on multimodal reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1174–1181, 2021.

[75] R. A. Romeo and L. Zollo, "Methods and sensors for slip detection in robotics: A survey," *IEEE Access*, vol. 8, pp. 73027–73050, 2020.

[76] J. W. James and N. F. Lepora, "Slip detection for grasp stabilization with a multifingered tactile robot hand," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 506–519, 2020.

[77] T.-H. Pham, A. Kheddar, A. Qammaz, and A. A. Argyros, "Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2810–2819, 2015.

[78] D.-K. Ko, K.-W. Lee, D. H. Lee, and S.-C. Lim, "Vision-based interaction force estimation for robot grip motion without tactile/force sensor," *Expert Systems with Applications*, vol. 211, p. 118441, 2023.

[79] P. Tran, S. Jeong, and J. P. Desai, "Voice-controlled flexible exotendon (flexotendon) glove for hand rehabilitation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4834–4839, 2019.

[80] Y. Guo, W. Xu, S. Pradhan, C. Bravo, and P. Ben-Tzvi, "Personalized voice activated grasping system for a robotic exoskeleton glove," *Mechatronics*, vol. 83, p. 102745, 2022.

[81] M. Liu, "Hust dataset (march 2016)," Mar 2016.

[82] D. Giovanelli and E. Farella, "Force Sensing Resistor and Evaluation of Technology for Wearable Body Pressure Sensing," *Journal of Sensors*, vol. 2016, 2016.

[83] K. Stone and C. Gonzalez, "The contributions of vision and haptics to reaching and grasping," *Frontiers in Psychology*, vol. 6, p. 1403, 2015.

[84] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Material recognition in the wild with the materials in context database," *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[85] Y. Guo, "Personalized voice activated grasping system for a robotic exoskeleton glove," Master's thesis, Virginia Polytechnic Institute and State University, 2021.

[86] C. Gao, Q. Cai, and S. Ming, "Yolov4 object detection algorithm with efficient channel attention mechanism," in *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, pp. 1764–1770, 2020.

[87] S. Chen, J. Hong, T. Zhang, J. Li, and Y. Guan, "Object detection using deep learning: Single shot detector with a refined feature-fusion structure," in *2019 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 219–224, 2019.

[88] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015.

[89] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10778–10787, 2020.

[90] X. Liang, L. Lin, Y. Wei, X. Shen, J. Yang, and S. Yan, "Proposal-free network for instance-level object segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2978–2991, 2018.

[91] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, vol. 9, pp. 82031–82057, 2021.

[92] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[93] H. Zhang, J. Xue, and K. Dana, "Deep ten: Texture encoding network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2896–2905, 2017.

[94] C. Zhao, L. Sun, and R. Stolkin, "A fully end-to-end deep learning approach for real-time simultaneous 3d reconstruction and material recognition," in *2017 18th International Conference on Advanced Robotics (ICAR)*, pp. 75–82, 2017.

[95] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *Advances in neural information processing systems*, vol. 24, 2011.

[96] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5297–5307, 2016.

[97] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

[98] Y. Guo, W. Xu, S. Pradhan, C. Bravo, and P. Ben-Tzvi, "Integrated and configurable voice activation and speaker verification system for a robotic exoskeleton glove," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 83990, American Society of Mechanical Engineers, 2020.

[99] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech and Language*, vol. 60, p. 101027, 3 2020.

[100] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[101] A. Nagraniy, J. S. Chungy, and A. Zisserman, "VoxCeleb: A large-scale speaker iden-

tification dataset," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2017-Augus, pp. 2616–2620, 2017.

[102] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Science and Language*, 2019.

[103] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxceleB2: Deep speaker recognition," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. i, pp. 1086–1090, 2018.

[104] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5791–5795, IEEE, 2019.

[105] S. He, A. Zhang, and M. Yan, "Voice and motion-based control system: Proof-of-concept implementation on robotics via internet-of-things technologies," *ACMSE 2019 - Proceedings of the 2019 ACM Southeast Conference*, pp. 102–108, 2019.

[106] I. M. El-emary, M. Fezari, and H. Attoui, "Hidden Markov model/Gaussian mixture models (HMM/GMM) based voice command system: A way to improve the control of remotely operated robot arm TR45," *Scientific Research and Essays*, vol. 6, no. 2, pp. 341–350, 2011.

[107] J. B. Ǵomez, A. Ceballos, F. Prieto, and T. Redarce, "Mouth gesture and voice command based robot command interface," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 333–338, 2009.

[108] R. K. Megalingam, R. S. Reddy, Y. Jahnavi, and M. Motheram, "Ros based control of robot using voice recognition," in *2019 Third International Conference on Inventive Systems and Control (ICISC)*, pp. 501–507, 2019.

[109] M. Pleva, J. Juhar, S. Ondas, C. R. Hudson, C. L. Bethel, and D. W. Carruth, "Novice user experiences with a voice-enabled human-robot interaction tool," in *2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 1–5, 2019.

[110] S. Guo, Z. Wang, J. Guo, Q. Fu, and N. Li, "Design of the speech control system for a upper limb rehabilitation robot based on wavelet de-noising," in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 2300–2305, 2018.

[111] A. Rangapur, S. C. Sethuraman, *et al.*, "Efficientword-net: An open source hotword detection engine based on one-shot learning," *arXiv preprint arXiv:2111.00379*, 2021.

[112] M. Wu, S. Panchapagesan, M. Sun, J. Gu, R. Thomas, S. N. P. Vitaladevuni, B. Hoffmeister, and A. Mandal, "Monophone-based background modeling for two-stage on-device wake word detection," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5494–5498, IEEE, 2018.

[113] K. Kumatani, S. Panchapagesan, M. Wu, M. Kim, N. Strom, G. Tiwari, and A. Mandai, "Direct modeling of raw audio with dnns for wake word detection," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 252–257, IEEE, 2017.

[114] D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky, "Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices," in *2006 IEEE international conference on acoustics speech and signal processing proceedings*, vol. 1, pp. I–I, IEEE, 2006.

[115] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5115–5119, 2016.

[116] Y. Zhang, M. Yu, N. Li, C. Yu, J. Cui, and D. Yu, "Seq2seq attentional siamese neural networks for text-dependent speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6131–6135, IEEE, 2019.

[117] S. Kadam and V. Vaidya, "Review and analysis of zero, one and few shot learning approaches," in *International Conference on Intelligent Systems Design and Applications*, pp. 100–112, Springer, 2018.

[118] S. H. Mun, W. H. Kang, M. H. Han, and N. S. Kim, "Robust text-dependent speaker verification via character-level information preservation for the sdsv challenge 2020," *arXiv preprint arXiv:2010.11408*, 2020.

[119] A. Lozano-Diez, A. Silnova, B. Pulugundla, J. Rohdin, K. Veselỳ, L. Burget, O. Plchot, O. Glembek, O. Novotnỳ, and P. Matejka, "But text-dependent speaker verification system for sdsv challenge 2020.," in *INTERSPEECH*, pp. 761–765, 2020.