# Joint Torque Feedback for Motion Training with an Elbow Exoskeleton

Hubert Kim

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

Mechanical Engineering

Alan T. Asbeck, Chair

Alexander Leonessa

Maury A. Nussbaum

Pinhas Ben-Tzvi

Alfred L. Wicks

September 14, 2021

Blacksburg, Virginia

Keywords: Physical Human-Robot Interaction, Joint-torque Feedback, Motion training, Haptic Exoskeleton, Torque Stiffness, Just Noticeable Difference.

Copyright 2021, Hubert Kim

# Joint Torque Feedback for Motion Training with an Elbow Exoskeleton

Hubert Kim

(ABSTRACT)

Joint torque feedback (JTF) is a new and promising means of kinesthetic feedback to provide information to a person or guide them during a motion task. However, little work has been done to apply the torque feedback to a person. This project evaluates the properties of JTF as haptic feedback, starting from the fabrication of a lightweight elbow haptic exoskeleton. A cheap hobby motor and easily accessible hardware are introduced for manufacturing and open-sourced embedded architecture for data logging. The total cost and the weights are $500 and 509g. Also, as the prerequisite step to assess the JTF in guidance, human perceptual ability to detect JTF was quantified at the elbow during all possible static and dynamic joint statuses. JTF slopes per various joint conditions are derived using the Interweaving Staircase Method. For either directional torque feedback, flexional motion requires 1.89-2.27 times larger speed slope, in $\mathrm{mNm}/(^\circ/\mathrm{s})$, than the extensional motion. In addition, we find that JTFs during the same directional muscle's isometric contraction yields a larger slope, in $\mathrm{mNm}/\mathrm{mNm}$, than the opposing direction (7.36 times and 1.02 times for extension torque and flexion torque). Finally, the guidance performance of the JTF was evaluated in terms of time delay and position error between the directed input and the wearer's arm. When studying how much the human arm travels with JTF, the absolute magnitude of the input shows more significance than the duration of the input ($p$-values of $< 0.0001$ and $0.001$). In the analysis of tracking the pulse input, the highest torque stiffness, $95$ $\mathrm{mNm}/\mathrm{deg}$, is responsible for the smallest position error, $6.102 \pm 5.117^\circ$, despite the applied torque acting as compulsory stimuli.

# Joint Torque Feedback for Motion Training with an Elbow Exoskeleton

Hubert Kim

(GENERAL AUDIENCE ABSTRACT)

Joint torque feedback (JTF) is a new and promising means of haptic feedback to provide information to a person or guide them during a motion task. However, little work has been done to apply the torque feedback to a person, such as determining how well humans can detect external torques or how stiff the torque input should be to augment a human motion without interference with the voluntary movement. This project evaluates the properties of JTF as haptic feedback, starting from the fabrication of a lightweight elbow haptic exoskeleton. The novelty of the hardware is that we mask most of the skin receptors so that the joint receptors are primarily what the body will use to detect external sensations. A cheap hobby motor and easily accessible hardware are introduced for manufacturing and open-sourced software architecture for data logging. The total cost and the weight are $500 and 509g. Also, as the prerequisite step to assess the JTF in guidance, human perceptual ability to detect JTF was quantified at the elbow during all possible static and dynamic joint statuses. A psychophysics tool called Interweaving Staircase Method was implemented to derive torque slopes per various joint conditions. For either directional torque feedback, flexional motion requires 1.89-2.27 times larger speed slope, in $\mathrm{mNm}/(^\circ/\mathrm{s})$ than the extensional motion. In addition, the muscles' isometric contraction with the aiding direction required a larger slope, in $\mathrm{mNm}/\mathrm{mNm}$ than the opposing direction (7.36 times and 1.02 times for extension torque and flexion torque). Finally, the guidance performance of the JTF was evaluated in terms of time delay and position error between the directed input and the wearer's arm. When studying how much the human arm travels with JTF, the absolute magnitude of the input shows more significance than the duration of the input ($p$-values of $< 0.0001$ and $0.001$). In the analysis of tracking the pulse

input, the highest torque stiffness, $95$ mNm/deg, is responsible for the smallest position error, $6.102 \pm 5.117°$, despite the applied torque acting as compulsory stimuli.

# Dedication

*This dissertation is dedicated to those who try to understand human perception about the elbow joint torque and hopefully improves their design of exoskeletons.*

# Acknowledgments

My Ph.D. has been an exciting journey of making a dream come true: becoming a robotics engineer. Those experiences were possible because of the following invaluable people.

First and foremost, I would appreciate my committee chairman, Dr. Alan Asbeck, for being a supportive advisor and a sincere friend. Throughout six years of study, Dr. Asbeck always trusted my decisions and respected my perspectives, which became a strength to accomplish the long race. As the first student who met Dr.Asbeck in VT, I was fortunate enough to see and experience every step of building up the Assistive Robotics Lab. The figuring-out process from starting over together with Dr. Asbeck has provided me a unique experience of problem-solving. Along with the hands-on mechatronics skills and intuitions about mechanical design, his vision of "making an impact in the world by helping people regain capabilities they lost" has been the driving force of my research as well. Most of all, his consistent curiosity about the fundamental phenomena is an exemplary attitude I learned as a scholar whenever approaching research questions.

I would also thank my committees, who gave me objective and helpful feedback after the preliminary exam. Dr. Asbeck and I tend to plan too broad and often stray away from the viable plan. Dr. Nussbaum and Dr. Leonessa's feedback were very accurate and helpful in designing the follow-up study after the preliminary presentation. Dr. Ben-Tzvi and Dr. Wicks's feedback allowed me to reconsider the key research questions and kept me objective when writing out the research papers. Moreover, I appreciate Dr. Batra, who helped me grant the ICTAS scholarship to study without financial problems. ICTAS program has been beneficial in pursuing the study as it covered the conference traveling fee and summer research fund.

I should thank my lab mates, Brandyn, Josh, and Ranger, who had a constant interest in my research. Also, Athulya, Tim, Hani, and Taylor for helping me out whenever I need help with experiments.

Furthermore, I would like to give Yegyu Han and Sungjae Ohn special thanks for answering numerous questions about the statistical and motor running problems.

I would appreciate the full support from my parents and my sister for being there for me with encouragement. The study would not have been finished without their assistance.

Lastly, I thank God, who led my life here and gave me the strength to finalize the project.

# Contents

**List of Figures** xiii

**List of Tables** xxii

**1 Introduction** 1

1.1 Contributions . . . . . 3

1.2 Hypotheses . . . . . 5

**2 An Elbow Exoskeleton for Haptic FDBK made with a Direct Drive Hobby Motor** 7

2.1 Hardware in context . . . . . 9

2.2 Hardware description . . . . . 10

2.2.1 Exoskeleton hardware . . . . . 11

2.2.2 Software architecture . . . . . 13

2.2.3 Initial position detection . . . . . 13

2.3 Design files . . . . . 15

2.4 Bill of materials . . . . . 15

2.5 Build instructions . . . . . 17

2.5.1 Base motor setup . . . . . 17

Constructing base frame . . . . . 19
System integration test . . . . . 20
2.5.2 Index-pin calibration . . . . . 22
Custom microcontroller code setup . . . . . 23
Processing code setup and serial communication with CCS . . . . . 25
Assemble the joint actuator . . . . . 25
Perform index-Pin calibration . . . . . 26
2.5.3 Finish exoskeleton . . . . . 29
Fabricate braces and adjust their ergonomic locations . . . . . 29
Conduct position error-driven torque control . . . . . 30
2.5.4 Design considerations . . . . . 31
Designing the human interface . . . . . 31
Sizing motors for passive control . . . . . 33
Improving communication . . . . . 34
2.6 Operation instructions . . . . . 37
2.6.1 Exoskeleton donning instructions . . . . . 37
2.6.2 Activation instructions . . . . . 37
2.7 Validation and characterization . . . . . 39
2.8 Acknowledgements . . . . . 41
2.9 Declaration of interest . . . . . 42
2.10 Human and animal rights . . . . . 42
**3 Just Noticeable Difference of Elbow Joint Torque Feedback** **43**

3.1 Introduction . . . . . 44

3.2 Results . . . . . 49

3.2.1 Static JND Torques . . . . . 49

3.2.2 Dynamic JND Torques . . . . . 51

3.3 Discussion . . . . . 53

3.3.1 Static Conditions . . . . . 53

3.3.2 Dynamic Conditions . . . . . 54

3.3.3 Static vs. Dynamic Conditions . . . . . 56

3.3.4 Summary and Future Work . . . . . 59

3.4 Conclusions . . . . . 60

3.5 Methods . . . . . 60

3.5.1 Elbow Exoskeleton. . . . . 60

3.5.2 Software Systems. . . . . 62

Psychophysics Tools. . . . . 62

Psychophysics Tool for Dynamic JND Testing. . . . . 64

3.5.3 Participants. . . . . 67

3.5.4 Experimental Protocol. . . . . 68

Calibration. . . . . 69

JND for static condition. . . . . 70

JND for dynamic joint condition. . . . . 71

3.6 Acknowledgements . . . . . 72

3.7 Data Availability . . . . . 72

3.8 Author contributions statement . . . . . 72

3.9 Additional information . . . . . 72

**4 The Effects of Torq. Mag. and Stiffness in Arm Guidance through Joint Torque Feedback 73**

4.1 Introduction . . . . . 74

4.2 Materials and Methods . . . . . 80

4.2.1 Materials . . . . . 80

Hardware . . . . . 80

Software . . . . . 81

4.2.2 Methods . . . . . 85

Participants . . . . . 85

Pre-programmed Input . . . . . 86

Experimental Protocol . . . . . 90

4.3 Results . . . . . 91

4.3.1 Current Pulse . . . . . 92

4.3.2 Position Pulse . . . . . 93

4.3.3 Position Step . . . . . 94

4.4 Discussion . . . . . 96

4.5 Conclusion . . . . . 102

4.6 Acknowledgment . . . . . 103

**5 Conclusion and Future Work** **104**

5.1 Hypothesis Evaluation . . . . . 106

**Bibliography** **108**

**Appendices** **119**

**A Contributions for HardwareX paper** **119**

# List of Figures

2.1 Left: side view of an arm wearing the exoskeleton. Right: Isometric view of the arm with the exoskeleton. Subjects wear the sleeve and the foam, followed by tightening the straps. Aluminum support is attached at the third brace to allow stable positioning on the testbed. . . . . 11

2.2 System overview and interrupt routines. Key components with the corresponding peripherals are addressed. (* marked priorities have to be set manually by users). . 12

2.3 Primary hand tools for fabricating the frame. These include: Ⓐ FMHT73756 tin snips, ($16-18) – used to cut H1, a long thin beam, into shorter pieces; Ⓑ Metal Hand Punch ($25-39) – used to punch holes in beams; Ⓒ PVC Pipe ($3-10) and Ⓓ Rubber Mallet ($2-10) – used to bend the aluminum braces into the curved arm shape; and Ⓔ Hollow Punch Tool ($10-20) – creating mounting holes in fabric. Additional required equipment not shown above are: screw drivers, a sewing machine, a soldering iron, a table mount, and a power supply (or a battery). . . . . 17

2.4 Overview of the assembly process, showing the three stages of assembly discussed in this paper. . . . . 18

2.5 Overview of the mechanical assembly procedure in each stage. . . . . 18

2.6 (a), CAD Models with colors indicating the primary method for fabrication. Detailed information about the CAD files (.prt) are ①: forearm_screw_cover (A6a), ②: ENCDAdapter (A6b), ③: dustCover_Top (A6c), ④: EncoderFrame_Bottom (A6d), and ⑤: M4Spacer (A6e). (b), Cross-sectional view with the fasteners indicated. The H4d screws securing the ④ to the Upper arm Frame should be countersunk in the direction from the ④ toward the Upper arm Frame. Similarly, H4g screws assembling Upper arm Frame, ④, and A1 should be countersunk in the direction from the frame to the motor. The holes for setscrews (H4c) of ② should be threaded after being printed or manufactured. Screws not specifically indicated are H4e, M3 × 10mm; nuts not specifically indicated are H4k, M3. . . . . 19

2.7 Mechanical hard stops limit the arm movement from 0 to 120°. These mitigate any mechanical risks from accidentally overpowering the motor. . . . . 19

2.8 (a) Wiring for the DRV8305EVM motor controller shield and (b) Settings for the C2000 board ([63]). In (a), pin 21 and 22 are used for running the Miniature 5V Cooling fan. In (b), **White** indicates the jumper and switch settings; **Blue** is where the motor shield should go; **Purple** is where the encoder should be attached. . . . . 20

2.9 (a), Angle estimation process with TI's sensored control. The code is programmed to initialize both mechanical and electrical angle to zero as the current is injected. (b), Angle estimation through a predefined index pin, as is done in our system. Current injection is a very useful approach, yet is not feasible for a direct-drive wearable device. . . . . 23

2.10 Left: four highlighted screw (H4j) points to secure the Nylon sheet to the frame. Middle: manufacturing guideline for straps (H4a) and buckles (H4b) as where to sew, punch, and screw. Right: mounting process to frames. From the medial to the lateral side, secure the screws in the order of: screw (H4e); washer (H4m); straps (H4a); lock nut (H4k). . . . . 29

2.11 (a), Simulation of the exoskeleton described in this paper. The motion amplitude achievable at different frequencies. Both conditions yield motion amplitudes of more than 300° at 6 Hz, which is larger than our target amplitude of 50°. Note that the peak-to-peak angle of the motion is twice the amplitude. The simulations assume a voltage of 24 V and a maximum operating current of 8 A. (b), Summary of the calculations for how to determine the required motor size. In the last steps for finding $|A_{speed}|$ and $|A_{torque}|$, $\cos(2\pi ft)$ or $\sin(2\pi ft)$ are set to 1, as that is the maximum value. $J_{motor}$ is identified from lab2c and converted using an equation (Section 10.5 in [62]). The inertia of the exoskeleton frame is neglected in this example. . . . . 32

2.12 Data Package for logging from CCS. The package is made of 10 bytes of ID and 10 bytes of data (five 16 bit data). Five data IDs, 0x66, 0x73, 0x74, 0x46, and 0x54, are represented as `dataID_01`, `dataID_02`, `dataID_03`, `dataID_04`, and `dataID_Timer` in `void serialEvent()`, Table 2.2. The IDs are arbitrary chosen and could be changed to different characters. The five variables currently being logged are motor position, filtered torque, current feedback, commanded current, and timer counter. . . . . 36

2.13 Instructions for donning the exoskeleton, in order counter-clockwise. First, a sleeve is put on the wearer, followed by wrapping the arm in foam. Next, the exoskeleton is wrapped around the foam, and the straps tightened. . . . . 38

2.14 (a), Step responses with proportional gains (3.0 and 5.0). (b) and (c), Position tracking with kinesthetic guidance. (c) shows close-ups of different portions of (b). In all of the plots, the user's arm angle is shown with the solid red line. The blue line indicates the input position. . . . . 40

3.1 (a), Illustration of dissected arm view, showing the primary proprioceptors involved in joint torque sensing as well as how the exoskeleton applies pressure and masks skin receptors. (b), Picture of a participant's arm with the exoskeleton worn, showing the exoskeleton components. (c), Illustration of how each propioceptor is triggered under different joint conditions. The left column shows how the Muscle Spindles (MS) lengthen or shorten depending on the arm's movement. The right column illustrates how isometrically contracted flexor or extensor activity affects the Golgi-Tendon Organ (GTO). (d), Illustration of the ten studied torque interactions, showing the kinesthetic stimuli (black arrows), contracted muscles (red highlighted muscle shapes), and direction of arm motion (blue arrows). . . . . 47

3.2 Summary of the test results. For the icons below the x-axis illustrating each experimental condition, please refer to Fig. 3.1(d). (a), The JND resulting from the static tests. All subject data is shown with gray lines, and the black line is the mean of the data. The data was analyzed with a Repeated Measures ANOVA and Tukey post-hoc test. Statistically significant pairs are indicated with asterisks. (b), The JND torque from dynamic tests, where the arm is in motion at different speeds. The black line is the median of the data. The black $\times$s indicate that the data exceeds the maximum device output, which is $1,200$ mNm. The data was analyzed with the Kruskal-Wallis test and followed by the Wilcoxon signed-rank test to compare pairs of conditions. Statistically significant pairs are indicated with asterisks. . . . 50

3.3 (a) GUI screens for the static (left) and dynamic (right) mini-games. The score for the mini-game provides participants whether the current user's response is correct or not. Additionally, the red block of the dynamic condition is only shown during the demonstration to supplement the ideal timing window for response. The white box represents the arm position that moves along three boxes that correspond the arm's range of motion. From (b) to (d) are examples of testing of the static (flexor muscle contracted), dynamic, and static (extensor muscle contracted) conditions, respectively. . . . . 61

3.4 (a), Schematic diagram of state machine for static condition. (b), Example of a static JND test result. . . . . 63

3.5 (a), GUI system guiding the arm movement. Participants are instructed to move their arm (white bar) to match the guiding block (blue block). The horizontal position of the arm position corresponds to the actual elbow joint angle. (b), Illustration of the commanded arm angular position versus time for the fast (1 Hz, top) and slow (0.5 Hz, bottom) velocities. The average speed over the center of the arm position range is shown with the red line. The gray shaded ranges of arm position correspond to the portions where the arm is changing speed (① and ③); the white regions are where the arm was moving at roughly a constant speed, and torque was applied (②). (c), State machine illustrating how torque is applied to the arm depending on the participant's current arm position and if their arm is moving in flexion or extension. State transitions 1 and 4 occur when the arm changes its direction of motion; the other state transitions occur based on the arm position. (d), Example results from a trial. The top graph illustrates how the random on/off works on each extensional/flexional torque input. The bottom graph shows the subject moving their arm back and forth (red line) and the subject reacting to the current input (blue line). The black asterisks show the moments when the subject responded. . . . . 66

4.1 Conceptual diagram showing the different types of proprioception [29, 51] and its role in arm movement control [9]. (a) Obtained/Active proprioception. (b) Imposed/Passive proprioception. (c) Mixed proprioception. . . . . 76

4.2 During arm guidance, an exoskeleton provides a joint torque $\tau_{JTF}$ as a function of the error $\theta_{error}$ between the desired and actual arm angle. This diagram presents several options for the function relating the error and applied torque. The torque stiffness, $\frac{\Delta\tau}{\theta_{error}}$, is small on the left and large on the right. The three stiffness levels involve: **Compliant JTF**- shallow torque slope, JTF provides cues but does not hinder the motion. **Rigid**- steep torque slope, JTF may constrain the wearer's arm. **On-off control**- vertical torque slope, JTF delivers a binary (on-off) cue. $\tau_{stall}$ is the torque limit due to the hardware limit; $\tau_p$ is the perceptual resolution; $\theta_{stall}$ is the position threshold until JTF becomes the $\tau_{stall}$; and $\theta_{threshold}$ is the position threshold until JTF becomes the $\tau_p$. . . . . 77

4.3 (a) A subject participating in the test. (b) The aluminum test setup for holding the arm 45° inclined. (c) A sample GUI for the demonstration. The white arm indicates the participant's arm position while the black arm guides the desired arm motion. The black arm is not present once the test begins. . . . . 79

4.4 Four different levels of torque stiffness. Boxes made of dashed lines represent each position threshold ($\theta_{threshold}$ from the Figure 4.2). Also, black dotted lines extending down to the x-axis indicate the $\theta_{stall}$. Three gains shown are Gain One (G1, shallow gain-12 mNm/deg); Gain Two (G2, steep gain-38 mNm/deg); and Gain Three (G3, virtual wall-95 mNm/deg). Note that the mixed gain GM is designed to have the same torque slope as G1, with an offset of 30°. The positive position error indicates the flexional arm displacement; the negative error is extensional arm motion. $\tau_{stall}$ is the system's peak torque. $\tau_p$ represents the perceivable threshold that is modified according to the exoskeleton environment. . . . . 83

4.5 Sample test results showing how piece-wise analysis is conducted. Blue is the commanded input, and Red is the wearer's arm movement. The three types of input are shown in different subfigures. (a) Current Pulse. Magenta is the segment; green and red circles show the beginning and ending points; cyan and blue circle indicate first and second peaks. (b) Position Pulse. Green and cyan circles are commanded position peak and arm movement peak. (c) Position Step. The red circle is the steady-state commanded position; the Black circle illustrates $90$ % of the arm's steady-state position. The positive direction illustrates the arm flexional motion; negative y-values show arm extension movement. . . . . 84

4.6 Test results for several examples of each input type. In each plot, the solid line shows the mean arm angle as a function of time; the shaded area shows one standard deviation from the mean. **(a)**: Current Pulse result of $10$ A magnitude with $0.3, 0.5, 1,$ and $2$ seconds of duration. The dashed line shows the applied current. **(b)**: Position Pulse result of $45°$ Pulse Input with $0.6, 1, 2, 4,$ and $8$ seconds of duration. **(c)**: Position Step result of increasing angle from $0$ to $45°$ and $0$ to $90°$ and decreasing angle from $90$ to $45°$ and $90$ to $0°$. Positive angle indicates the flexional arm movement; the negative numbers are extensional arm angle. The dashed line shows the commanded step input. . . . . 89

4.7 Measurements of the tracking data for the Current Pulse. (a): Arm's Traveling distance as JTF (left) and Arm's moving speed (right) as fixed effects. (b): Interaction effect between the magnitude of the JTF and the movement speed on Arm's traveling distance. The arm travel distances plotted are the absolute value of the travel distance for ease of comparing the positive and negative torques; the negative torques resulted in negative travel distances. (c): and the peak-to-peak time delay from a Current Pulse. JTF of the positive values are flexional input; the negative values are extensional feedback. For (a) and (c), The statistically different pairs are highlighted with asterisks. . . . . 92

4.8 Absolute Peak-to-peak Error for Position Pulse for the gain effect (left) and peak-to-peak time delay for the gain effect (right). Further details about the significant pairs are represented as asterisks. . . . . 94

4.9 Unpooled Means and Stds of Position Pulse, peak-to-peak position error (left) and the peak JTF (right) . . . . . 95

4.10 Unpooled Means and Stds of the Position Pulse's bode plot. The magnitude, dB, is calculated by the ratio of the arm's peak to the input peak. The raised cosine function is treated like a sinusoidal wave when the data was converted into phase. The figure on the right side shows the gain effect on the phase delay. . . . . 97

4.11 Unpooled mean values of steady-state Position Step error represented as a vector table. The x-axis indicates the arm's initial position and the y-axis is the final position. The Left panel is the G1 condition and the Right panel is G2. $0^\circ$ is where the arm is fully extended. N/M indicates that the corresponding initial-final step input pair was not measured. . . . . 98

# List of Tables

2.1 Bill of materials. "Desig." = Designator; "Qty." = Quantity; T=thickness; W=width; L=length. . . . . 16

2.2 Pseudo-code for communication. The ASCII IDs tagged for the current data package are c, 125, 124, and 123 in decimal. Those are arbitrarily chosen and can be modified to be other ASCII codes. Switching to other variables can be done in `updateDLogPkg()`. . . . . 35

2.3 Values for BRR, the 16-bit baud-select register. The default enumeration structure defined in `sci.h` assumes a serial clock of 15 MHz (low speed peripheral clock, LSPCLK), where setting BRR to 194 leads to a 9600 baud rate. This system described in this paper uses a Baud rate of 57600. For the detailed table, refer to SPRUH18G [61], Table 13-11. . . . . 37

3.1 Description of the different conditions examined in this paper. . . . . 49

3.2 Statistical significance values for static and dynamic testing. Shown are the values from Tukey HSD Analysis (for the static tests) and the Wilcoxon Signed Rank test (for the dynamic tests) for each pair of conditions that was statistically different. (Flx. = flexional; Ext. = extensional; Torq. = torque.) . . . . . 51

3.3 Unpooled means ± standard deviations of the static JND torque. The first row (Ext. Torq.) corresponds to the FCE2, FCE1, NTE, ECE1, ECE2 trials, while the second row (Flx. Torq.) corresponds to the FCF2, FCF1, NTF, ECF1, ECF2 trials. (Flx. = flexional; Ext. = extensional; Torq. = torque). . . . . 51

3.4 Medians of the dynamic JND condition. The first row (Ext. Torq.) corresponds to the FME2, FME1, NTE, EME1, EME2 trials, while the second row (Flx. Torq.) corresponds to the FMF2, FMF1, NTF, EMF1, EMF2 trials. (Flx. = flexional; Ext. = extensional; Torq. = torque.) . . . . . 52

4.1 Significant levels of interaction effect between gain and duration of Position Pulse. The compared DVs are Peak-to-peak position error (left) and peak JTF within the trial (right). A-G are the group labels where the fixed effects with the same label are not statistically significant. . . . . 96

# List of Abbreviations

BLDC Brushless DC Motor

CCS Code Composer Studio

CCW Counter Clockwise

CNS Central Nervous System

CW Clockwise

FOC Field Oriented Control

FSR Force Sensitive Resistors

GTO Golgi Tendon Organ

GUI Graphical User Interface

HIL Human in the loop

JND Just Noticeable Difference

JTF Joint Torque Feedback

LPF Low Pass Filter

MS Muscle Spindle

PWM Pulse Width Modulation

SCI Serial Communication Interface

SISO Single Input Single Output

JND is a term conventionally used to describe the smallest amplitude of the stimuli for a person to perceive.

# Chapter 1

# Introduction

Joint Torque Feedback is an intuitive haptic means imposed by wearable robots for various robot-mediated training. It is natural feedback in the sense that it stimulates the same sensory organs—proprioceptors—that we use in daily activities to get feedback about our body's position and the forces in our muscles ( [29, 51]). They serve as the primary means of sensory information when we feel "being guided" when others pull or push a body segment. Conventional application of the torque feedback is in physiatry ( [10, 16, 46]) where the primary focus is to rehabilitate the patient's mobility or teaching a new movement.

The haptic guidance is achieved through delivering the temporal and spatial information via physical contact ( [18, 27, 65]), similarly to how vibrotactile motors can convey information [4]. As a baseline study, researchers explored the minimum perceptual limit for the kinesthetic feedback on the arm ( [18, 27, 65]). Although useful as they are, the study above can be improved in a couple of points. First, the wearable device should isolate the kinesthetic sensation from the tactile sensation as much as possible since the skin feedback tends to be more sensitive than the joint receptors ( [29]). This way, the perceptual resolution measured is less dependent on the surface pressure from various shape factors of the exoskeleton. Second, when measuring the perceptual threshold of the body segment, the dynamic and static conditions should be treated separately as each condition might trigger different primary proprioceptors ( [18, 65]).

Keeping the "transparency" of an exoskeleton to its wearer is a critical design element to deliver haptic feedback. Any backlash or time delays between a person's motion and the exoskeleton's

response could generate parasitic torque and impair the wearer's desired movement ( [56]). The typical approach to handle the resistance torques was to loop back the sensor readings to compensate for the error. (e.g. in Harmony- [32] and MGA- [10]). However, completely preventing the person from feeling the parasitic torque is impossible. Even State-Of-The-Art control systems incorporate parasitic resistance torques.NEUROExos ( [66]) has $0.5$ Nm of friction torque, Harmony ( [32]) has $0.3$ Nm, Martelli 2014 (hip exoskeleton) has $1.4$ Nm ( [48]). Those torques are likely to affect the wearer's proprioceptive perception as those are larger than the human proprioceptive threshold ( [18, 35]). Alternatively, direct-driving the motor on the joint could minimize the inertia ending up reducing the response time ( [22, 26, 42, 45, 49]). The exoskeleton manufactured in this project is designed for applications where the device does not overpower the wearer but rather delivers a noticeable cue to encourage the wearer to move ( [53]). Such a system can be implemented for enhancing the user experience in Virtual Reality or for replacing the repetitive therapy in rehabilitation or acquiring new skills ( [22]).

Analyzing the smallest requirements for torque perception provides intuitions for not only efficient mechanical design and but also optimal temporal and spatial information transfer to the wearer. In this project, we examine the magnitude of JTF on various aspects to understand how fast and accurately guide the individual's right arm through physical contact.

Before moving further, understanding how the human body comprehends the external torque during the physical guidance would be helpful. While being guided with the wearable device, the person's neural system simultaneously processes two types of proprioceptions, obtained and imposed proprioception ( [29]). Although both proprioceptions refer to a sense of body movement, the obtained proprioception occurs when individuals voluntarily execute a motion ( [9]). In contrast, the imposed proprioception is the sensation of body coordination when someone's limb is passively driven ( [51]). In this regard, JTF is distinctive from other tactile feedback in that the proprioceptors for sensing and actuating shared the same sensory organs. Therefore, when mea-

suring perceptual ability, one should also consider the obtained proprioception effect generated from participants actuating the arm.

One approach to measuring perceptual ability is to utilize the psychophysics tool ( [18, 27]). This section explores the minimum human perception boundaries of externally applied joint torques (Just Noticeable Difference, or JND). Human perception in physical Human-Robot Interaction (pHRI) for rehabilitation plays a key role because the subject's active participation determines motor learning performance. Consequently, excessive guidance with no consideration on the human perceptual level could hinder the learning process ( [57, 69]). A few groups of researchers have studied the human perception ability regarding kinesthetic interactions. [18] and [65] investigated flexional kinesthetic stimuli on the arm with a hand-held device and exoskeleton, respectively. Those studies are useful yet did not cover the entire joint combination of the arm joint. In this project, we address the masking effect of ten different dynamic and static joint conditions.

The empirical research above was conducted based on the idea of isolating the wearer's perceptual ability, imposed proprioception, from its after-effect, obtained proprioception. However, our observation of JND cannot simply be applied to the real-life tracking scenario because the measured JND of the imposed proprioception is perpetually affected by the wearer's movement. Instead of directly measuring the perceptual resolution, another approach is to look for the frequency-domain properties ( [19]). In this case, the wearer's perceptual ability is measured based on how well one can track the pre-programmed path via JTF. As the desired motion is easily achievable to anyone, we believe that the dominant factor of the tracking is from how well one can feel the guidance rather than how good one's muscle strength is.

## 1.1 Contributions

We have made the following contributions in this thesis:

- We designed and fabricated a lightweight arm exoskeleton made of a Brushless DC motor and corresponding data logging modules. The microcontroller code for the exoskeleton has a feature where an encoder with an index-pin replaces the functionality of Hall effect sensors. This allows a range of different motor models, even those without Hall effect sensors, to be used in robotics research. Additionally, we present calculations for choosing motor sizes for joint torque feedback, and present initial results of guiding the arm with very low torques ($< 1Nm$). The draft and the required code for running the BLDC motor and the serial communication tool are open-sourced via Hardware-X. Thus, the joint actuator can be implemented in any application requiring a backlash-free torque since the device is direct-drive.

- We investigated the perceptual resolution of torque feedback at the elbow joint. Specifically, we determined the minimum external torque that the elbow could sense under ten different conditions. These conditions included with the arm isometrically contracted in both flexion and extension, and with the arm moving back and forth at two speeds. The Interweaving Staircase Method was implemented to find the converging torque values of the joint under these ten conditions, including separately determining the elbow's sensitivity in flexion and extension. In all cases, we masked the touch receptors on the arm, such that the experiment isolated the muscle spindles and Golgi Tendon Organ to detect the stimuli. To hold the participant's attention while keeping the task accessible, we introduced a multi-modal User Interface where visual and auditory feedback was provided.

- We determined the human response to torque pulses of different amplitudes and frequencies provided with an elbow exoskeleton. This provided a baseline of how far the elbow would move under different conditions, with the human participant attempting to follow the exoskeleton's applied torque. Subsequently, we evaluated the human response to pulses where the arm was guided with a PD-controller running on an arm exoskeleton. If the human

moved differently than a commanded trajectory, the exoskeleton would apply a torque to the elbow proportional to the angular error. With this system, we evaluated the human tracking performance with four different torque slopes. During the experiments, we randomly combined each signal to mitigate human anticipatory behavior. We finally evaluated the human response to step inputs with different initial and final angles. This further allowed us to understand the sensitivity of the arm in different ranges of motion. We found that guiding the right arm with a $12$ mNm/deg of torque stiffness led to a similar phase delay observed with subjects driven by contralateral kinesthetic input (around $-50°$ at 1 Hz). With the step input experiment, we found that the response to extensional step motions was different than the response to flexion step motions.

These contributions are detailed in three papers, which are chapters 2-4. These are:

Chapter 2: H. Kim and A. Asbeck, "An elbow exoskeleton for haptic feedback made with a direct drive hobby motor." *HardwareX* 8, e00153, DOI: https://doi.org/10.1016/j.ohx.2019.e00060 (2020).

Chapter 3: H. Kim and A. Asbeck, "Just Noticeable Differences for Elbow Joint Torque Feedback." *Scientific Reports*, (Under review).

Chapter 4: H. Kim and A. Asbeck, "The Effects of Torque Magnitude and Stiffness in Arm Guidance through Joint Torque Feedback," *IEEE Access*, (Submission imminent).

## 1.2 Hypotheses

The following is a list of the hypotheses examined in this thesis:

- H0: We hypothesize that the elbow joint can be guided through an arbitrary trajectory with torques as little as 1 Nm, and a bandwidth that depends on the torque applied as a function

of the arm's position error from the desired trajectory.

- H1: We hypothesize that masking the tactile sensations at the skin of the arm, such that only the sensory organs deep in the muscle and tendon are used to detect external forces, will yield elbow JND torques of a higher magnitude than those evaluated while holding a stylus in the hand.
- H2: We hypothesize that the JND torque at the elbow joint measured during arm movement will have a value larger than JND torques measured while the arm is stationary and isometrically contracted.
- H3: We hypothesize that a higher torque stiffness during elbow guidance with joint torques (i.e., a larger amount of torque given a certain angular error between a target joint angle and the actual joint angle) will lead to better tracking results, but torque slopes that are too steep will behave like an on-off controller and result in poor tracking performance.

# Chapter 2

# An Elbow Exoskeleton for Haptic FDBK made with a Direct Drive Hobby Motor

**Authors:** *Hubert Kim*[1] *and Alan T. Asbeck*[1]

**Affiliations:** [1]Mechanical Engineering Department, Virginia Tech, Blacksburg, VA 24061, USA.

**Contact email:** aasbeck@vt.edu

**Abstract:**

A direct drive motor is one of the simplest mechanisms that can be used to move a mechanical joint. In particular, a brushless direct current (BLDC) motor with no gearing produces a low parasitic torque due to its backdrivability and low inertia, which is ideal for some applications such as wearable systems. While capable of operating with a higher power density than brushed motors, BLDC motors require accurate position feedback to be controlled via vector control at slow speeds. The MotorWare™ library from Texas Instruments (TI), which is designed to run with a C2000 microcontroller, is written to run BLDCs. However, the code was written to run the motor continuously with an incremental encoder and requires further engineering to be used at low speeds such as in an exoskeleton. In this paper, we present the design of an elbow exoskeleton that can be used for haptic feedback. We provide instructions to build the exoskeleton hardware, custom code to modify software provided by TI so that a motor can

provide a controlled torque at low speeds, code to enable the microcontroller to communicate with a computer for high-level commands and data storage, and also provide an overview of how alternate motors could be used with this software setup.

**Keywords:** brushless DC motor control; direct drive actuator; haptic feedback; joint torque feedback; exoskeleton

**Specifications table:**

| | |
|---|---|
| **Hardware name** | Elbow Haptic Exoskeleton |
| **Subject area** | Mechatronics |
| **Hardware type** | Joint actuator |
| **Open source license** | Berkeley Source Distribution (BSD) license |
| **Cost of hardware** | USD $509 |
| **Source file repository** | http://dx.doi.org/10.17632/skm88ynyhv.4 |

## 2.1 Hardware in context

There are numerous applications for an actuator that can provide controlled torques at slow speeds, including robots, test fixtures, or force feedback systems. In this paper, we focus on a wearable force feedback system (an exoskeleton) that can provide a torque to a person's joint. Such a system can be used for motion training for sports, rehabilitation, or skills acquisition [22]. The exoskeleton presented in this paper is designed for applications where the device does not overpower the wearer but rather delivers a small torque to train a person in how to move [53].

For rehabilitation or haptic feedback applications, the "transparency" of an exoskeleton to its wearer is an important design factor. Delays between a person's motion and the exoskeleton's response can cause parasitic resistances, which can hinder the wearer's natural motion. Typically, researchers address resistance torques by using a closed-loop system where the interaction forces between the human and exoskeleton are sensed, and then the actuators are moved to reduce unwanted forces (e.g. in Harmony [32] and MGA [10]). However, even good control systems can have parasitic resistance torques of 0.3-1 Nm [32, 48, 66], which somewhat penalizes the device's transparency, and which is much more than the just noticeable difference (JND) joint torque for haptic feedback applications [18, 35]. An alternate approach is to use a large-diameter direct drive motor (no gear train) to move a joint, which has minimal inertia and can respond to inputs quickly [22, 26, 42, 45, 49]. While the peak torque is smaller than other methods due to the lack of a gearbox, the resistance torques are close to zero. We use this approach for an elbow exoskeleton, presented in this paper, so the

system can be used for haptic feedback experiments such as measuring the JND [35]. If desired, the system presented in this paper can be adapted to use a small gear ratio in order to increase the torque output, at the cost of transparency.

Kinesthetic feedback with our exoskeleton could be delivered during many possible activities, including sports, teleoperation, force feedback (e.g., in virtual reality), or motor recovery for neurologic patients, in addition to fundamental haptic feedback experiments as we have done ([35]). For all of these applications, the exoskeleton's role is to convey information through a torque or to guide the wearer to a new joint angle, similar to a trainer pushing on the arm with their hands. The paper describes the mechanical design of our exoskeleton, the microcontroller software needed to drive it, and the computer software used to provide high-level commands to the exoskeleton and log data. These modules can be easily be adapted to a wide range of other applications.

## 2.2 Hardware description

The hardware described in this paper is a one degree of freedom (DoF) arm exoskeleton, with the joint actuated by a direct-drive brushless direct current (BLDC) motor (Figure 2.1). It can generate a torque of 0.5 Nm at stall for 60 seconds with a current of 10 A without any heating issues. While this peak torque is small, it is strong enough to convey useful torque feedback, and utilizes a very light-weight and low profile motor. If desired, it is straightforward to replace the motor with a more powerful BLDC motor to generate more torque. The mechanical hardware constitutes simple plastic plates and aluminum frames that are easily manufactured in a laboratory or hobby shop environment. In addition to providing details about the exoskeleton structure, we provide source code files for running the motor with full torque at stall conditions as well as software to communicate with a computer and log data from the exoskeleton.

Thus, our system is useful for researchers who are interested in the following tasks:

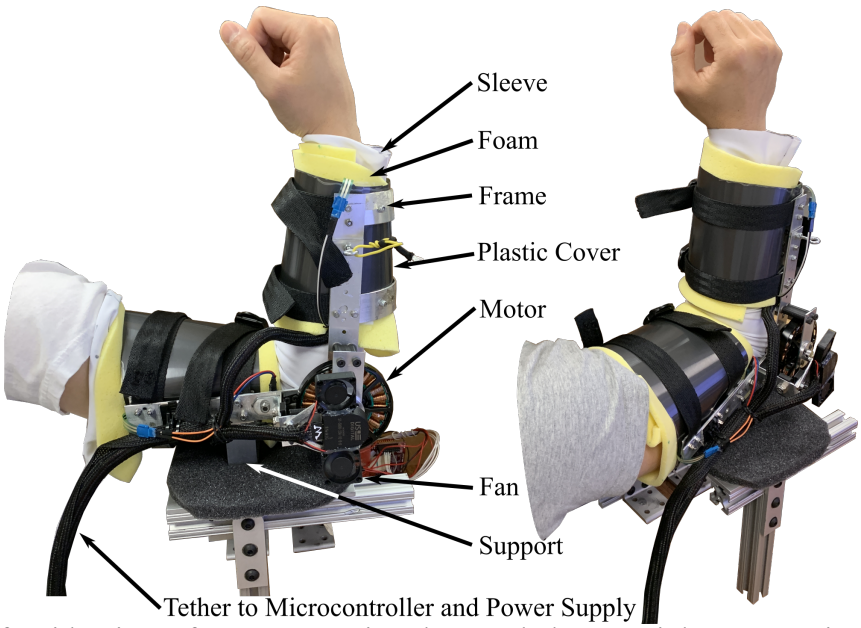


Figure 2.1: Left: side view of an arm wearing the exoskeleton. Right: Isometric view of the arm with the exoskeleton. Subjects wear the sleeve and the foam, followed by tightening the straps. Aluminum support is attached at the third brace to allow stable positioning on the testbed.

- Motion training or haptic feedback using an exoskeleton, for example in sports training, rehabilitation, or virtual reality scenarios;
- Laboratory experiments with haptic feedback or wearable robotics; and
- Robotics using high-torque, low-cost hobby motors.

### *2.2.1 Exoskeleton hardware*

The actuator interface is constructed with easily-accessible materials such as 6061 Aluminum and plastic sheets. The actuator interface is a semi-rigid structure made of aluminum frames with plastic covers wrapping around the forearm and upper arm. The motor is mounted to the frame at the hinge. The exoskeleton is tethered to the microcontroller and the power supply. The aluminum frame is on the outside of the wearer's arm, and wraps half way around the exterior of the arm. The plastic cover wraps around the anterior side of the arm, where muscle volume changes occur during device operation. Foam pads the user inside the plastic covers, and straps hold the wearer

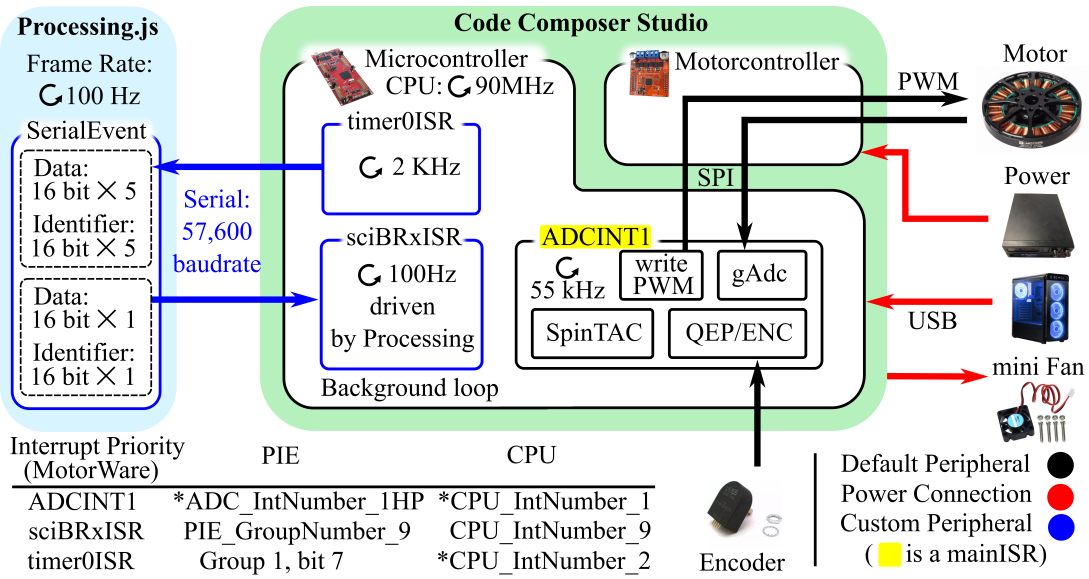


Figure 2.2: System overview and interrupt routines. Key components with the corresponding peripherals are addressed. (* marked priorities have to be set manually by users).

inside the exoskeleton.

The exoskeleton is driven by a T-Motor MN7005-KV115 outrunner brushless motor, and its position is sensed by a US Digital S1-5000 optical encoder. The motor is controlled by a Texas Instrument (TI) C2000 TMS320F28069M launchpad as the main processing unit. The microcontroller directly communicates with an attached motor shield, a DRV8305EVM.

Two miniature fans are attached above the motor to circulate the air through it, since it cannot dissipate heat effectively at stall. From testing, we found that the actuator system could handle the heat generated from 10 A for more than 30 seconds (out of a peak current of 15 A). The exoskeleton is designed to create up to 0.5 Nm of torque while keeping the weight at 500 g. These specifications are dependent on the motor chosen; with more powerful motors, the maximum torque can be increased.

### *2.2.2 Software architecture*

The embedded software platform is Code Composer Studio (CCS) with a Texas Instrument (TI) C2000 TMS320F28069M as the main processing unit. The microcontroller directly communicates with the attached motor shield, a DRV8305EVM, via Texas Instrument's MotorWare™ Library (Figure 2.2). The TI MotorWare library has been used by several groups for motor control ([6, 24]) but the details of many elements needed for its effective use have not been provided previously. The MotorWare library employs Field Oriented Control (FOC, equivalent to Vector Control), where the pulse-width modulation (PWM) waveforms are generated based on the relative position between the stator and rotor in the motor. FOC is suitable for controlling Permanent Magnet Synchronous Motors (PMSM) and Brushless DC motors (BLDC).

Our code is an encapsulated library that includes serial communication, timer interrupt, and motor-related functionalities. We use the library in conjunction with TI's tutorial project files ([62], which provides several "labs" demonstrating the MotorWare code). In our code, we estimate the mechanical angle through a predefined index pin, which allows us to use optical encoder as an absolute encoder while also providing an electrical angle. Bi-directional communication is achieved through a timer interrupt and serial interrupt module. Instead of using the provided position control project (lab13b) where a single bandwidth is used as a tuning parameter, we use a proportional-derivative (PD) controller because it allows more control. It is also more intuitive to tune the PD controller, as each of the gains corresponds to the position and speed error, respectively.

### *2.2.3 Initial position detection*

Estimating an electrical angle is necessary for accurate torque production in vector control. The measured or calculated electrical angle is transformed into three PWM phasors for running the motor, through the Park Transformation and Space Vector Module. The electrical angle is determined based on the position relative to the North and South poles on the rotor and can be measured by a Hall effect sensor. Additionally, for a motor to be used as a direct-drive actuator, a measurement

of the mechanical angle (i.e., the exoskeleton joint angle) is necessary. A common approach to obtain both types of angle information is to use both an absolute encoder and a Hall effect sensor. Depending on the choice of an incremental or absolute encoder, the mechanical angle could be relative to the encoder power-up position or the absolute rotation angle, respectively.

Alternatively, TI MotorWare provides three methods for angle estimation: sensorless, hall sensor, and only encoder. The first two cases rely on an internal estimator. In the estimator, the motor model is reconstructed based on the back-electromagnetic field (back-EMF) readings and predefined motor specifications. However, usable back-EMF readings are only available under sufficient speed, and the initial angle estimation becomes challenging during the initial start-up where the back-EMF is weak. Therefore, those methods are not appropriate for a manipulator operating at slow speeds or stall.

The third method implements a current injection technique. The technique is valid in vector control, where a three-phase PWM waveform (a-b-c axes) is transformed from d-q variables. MotorWare uses the phase magnitude invariance method (Chapters 3 and 5 in [58]) for the conversion. The current applied on the quadrature axis (q-axis) is proportional to the generated torque and spatially perpendicular to the main flux direction (d-axis). The injected current on the d-axis should not affect the torque intensity nor causing motor rotation, so long as the estimated d and q axes correctly line up with the actual axes. Therefore, applying current is used as a prerequisite step before running the motor, and it aligns the rotor to the stator position. It also measures the coil resistance for accurate motor control. However, it is not ideal when the load is too heavy to move. In a direct-drive application, or an application with a low-backlash gearbox connected to a load, the motor is not able to move sufficiently with this technique to align the rotor and stator.

Instead, we estimate the mechanical angle via the index-pin, where the initial angle for vector control is measured with an optical encoder and its predefined index pin. By calculating the mechanical angle based on the pre-memorized position through the index pin, we use an incremental encoder

like an absolute encoder, and we re-calculate the electrical angle based on the mechanical angle (Figure 2.9). As this approach disables the existing initial current injection, which measures the coil resistance, we use `runRsOnLine()` from [62] to update the coil resistance periodically. This also mitigates the coil resistance drop during the temperature change from stall operation, enabling accurate motor control.

## 2.3 Design files

CAD models for 3D printing, drawings for aluminium frame fabrication, and CCS scripts for motor control are provided.

| Design filename | File type | Open source license | Location of the file |
|---|---|---|---|
| CAD_Files.zip | 3D print files | CC BY-NC-SA 4.0 | Mendeley Dataset |
| CCS_codes.zip | CCS scripts | BSD 3-Clause License | Mendeley Dataset |
| Processing_codes.zip | Processing scripts | MIT License | Mendeley Dataset |
| frame_drawings.pdf | pdf Schematic | CC BY-NC-SA 4.0 | Mendeley Dataset |

## 2.4 Bill of materials

Following is a list of components with their labels. Actuator modules are grouped as A; the hardware structure for the exoskeleton is presented in group H; software groups are in S.

| Desig. | Component | Qty. | Cost per unit (USD) | Total cost (USD) | Source of materials |
|---|---|---|---|---|---|
| A1 | MN7005 KV115 | 1 | 249.9 | 249.9 | T-Motor |
| A2 | LAUNCHXL-28069M | 1 | 24.99 | 24.99 | TI |
| A3 | BOOSTXL-DRV8305EVM | 1 | 93.22 | 93.22 | Digi-key |
| A4 | S1 Optical Shaft Encoder | 1 | 89.80 | 89.80 | US digital |
| A5 | Miniature 5V Cooling fan | 2 | 3.50 | 7.00 | Adafruit |
| A6 | A6a. forearmScrewHousing | 1 | - | - | 3D print |
| | A6b. encoderMotorAdapter | 1 | - | - | 3D print |
| | A6c. encoderAdjustment | 1 | - | - | 3D print |
| | A6d. encoderFrameAdapter | 1 | - | - | 3D print |
| | A6e. frameSpacing | 1 | - | - | 3D print |
| H1 | 6061 Aluminum, 1/16”T, 1”W | 1 | 4.74 | 4.74 | McMaster |
| H2 | Foam Sheet, 1/4”T, 18” by 20” | 1 | 16.54 | 16.54 | McMaster |
| H3 | Nylon Sheet, 6”W, 0.015”T | 1 | 12.80 | 12.80 | McMaster |
| H4 | H4a. Webbing Strap, W: 1” L: 1’ | 1 | 0.91 | 0.91 | McMaster |
| | H4b. Sew-On Buckles for 1” Webbing | 4 | 0.28 | 1.12 | McMaster |
| | H4c. M3-0.5×3mm Set Screw | 2 | 0.05 | 0.10 | McMaster |
| | H4d. M3-0.5×4mm Flat Head | 4 | 0.15 | 0.60 | McMaster |
| | H4e. M3-0.5×10mm Flat Head | 18 | 0.05 | 0.90 | McMaster |
| | H4f. M3-0.5×39mm Flat Head | 2 | 0.10 | 0.20 | McMaster |
| | H4g. M4-0.7×8mm Flat Head | 2 | 0.05 | 0.10 | McMaster |
| | H4h. M5-0.8×10mm Flat Head | 2 | 0.07 | 0.14 | McMaster |
| | H4i. M5-0.8×6mm Flat Head | 2 | 0.32 | 0.64 | McMaster |
| | H4j. M3-0.5×6mm Flat Head | 4 | 0.05 | 0.20 | McMaster |
| | H4k. M3-0.5 Nylon-Insert Locknuts | 24 | 0.04 | 0.80 | McMaster |
| | H4l. M5-0.8 Nylon-Insert Locknuts | 4 | 0.01 | 0.04 | McMaster |
| | H4m. M3 3.2mm ID 6mm OD Washer | 8 | 0.03 | 0.24 | McMaster |
| | H4n. M4-0.7×15mm Socket Head | 1 | 1.56 | 1.56 | McMaster |
| | H4o. M4-0.7 Nylon-Insert Locknuts | 1 | 0.12 | 0.12 | McMaster |
| | H4p. L-Bracket, 7/8”×7/8”×5/8” | 2 | 0.43 | 0.86 | McMaster |
| S1 | S1a. `HardwareX_comm_N_timer.h` | - | - | - | CCS script |
| | S1b. `HardwareX_comm_N_timer.c` | - | - | - | CCS script |
| | S1c. `main.h` | - | - | - | CCS script |
| | S1d. `HardwareX_haptic_joint.c` | - | - | - | CCS script |
| S2 | S2a. `Processing_HardwareX.pde` | - | - | - | Opensource |
| | S2b. `receive_Stat.java` | - | - | - | Opensource |
| | S2c. `traj_1.txt` | - | - | - | Opensource |

Table 2.1: Bill of materials. “Desig.” = Designator; “Qty.” = Quantity; T=thickness; W=width; L=length.

Figure 2.3: Primary hand tools for fabricating the frame. These include: Ⓐ FMHT73756 tin snips, ($16-18) – used to cut H1, a long thin beam, into shorter pieces; Ⓑ Metal Hand Punch ($25-39) – used to punch holes in beams; Ⓒ PVC Pipe ($3-10) and Ⓓ Rubber Mallet ($2-10) – used to bend the aluminum braces into the curved arm shape; and Ⓔ Hollow Punch Tool ($10-20) – creating mounting holes in fabric. Additional required equipment not shown above are: screw drivers, a sewing machine, a soldering iron, a table mount, and a power supply (or a battery).

In addition to the above materials, several hand tools are needed to complete the build process. These are shown in Figure 2.3.

## 2.5 Build instructions

The assembly process involves three primary stages, as seen in Figure 2.4: Base Motor Setup, Index-Pin Calibration, and Final Exoskeleton Assembly. Each step includes both hardware and software steps. An overview of the hardware assembled during each step is in Figure 2.5. The scope of contents addressed in this paper is more focused on the engineering side based on the existing manuals. Therefore, technical basics provided from [62], such as importing the MotorWare library or checking the target configuration, are not discussed.

### *2.5.1 Base motor setup*

*Constructing base frame*

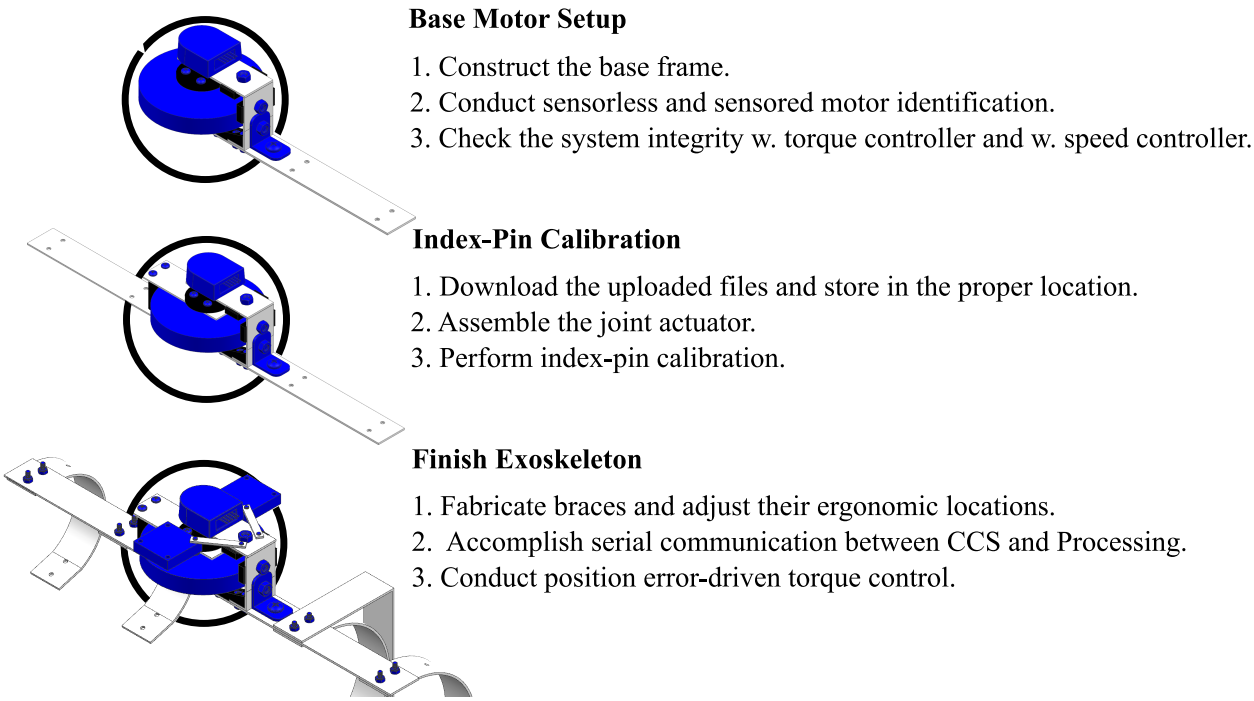


Figure 2.4: Overview of the assembly process, showing the three stages of assembly discussed in this paper.

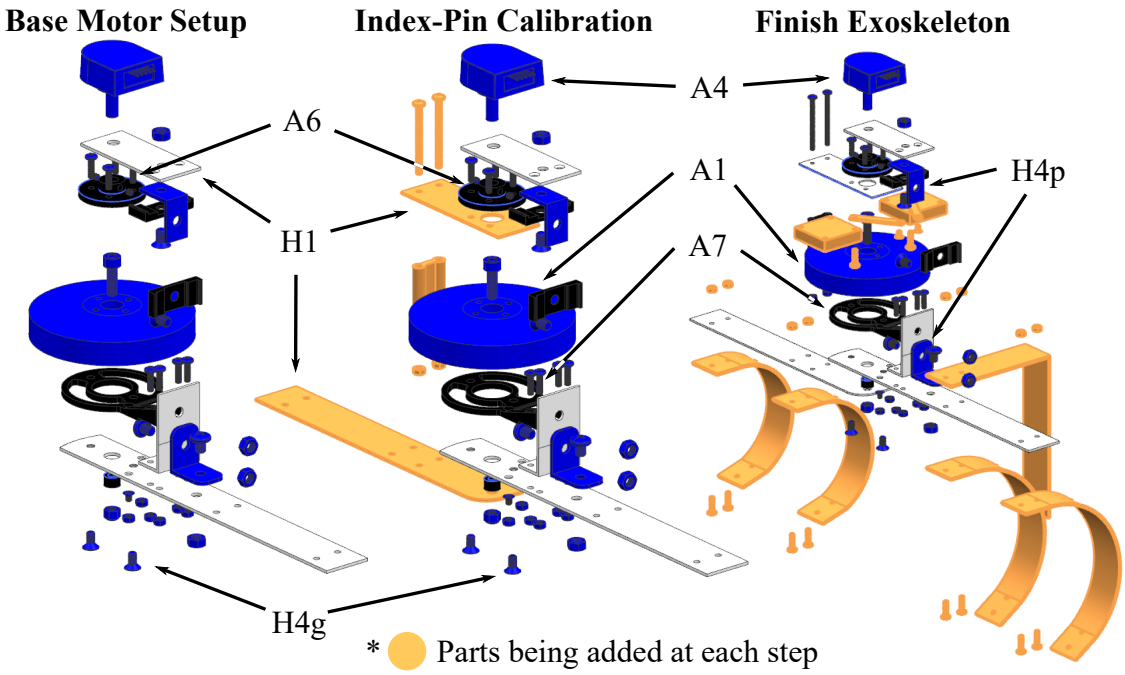


Figure 2.5: Overview of the mechanical assembly procedure in each stage.

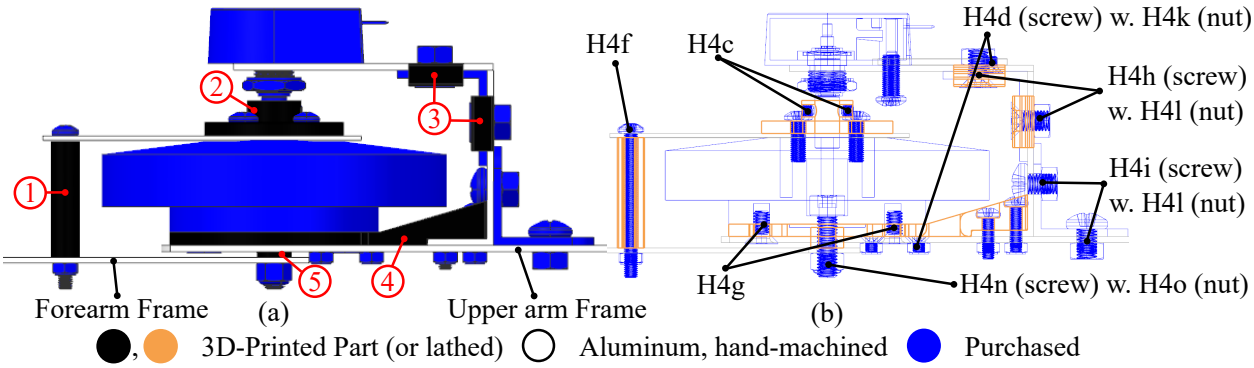


Figure 2.6: (a), CAD Models with colors indicating the primary method for fabrication. Detailed information about the CAD files (.prt) are ①: forearm_screw_cover (A6a), ②: ENCDAdapter (A6b), ③: dustCover_Top (A6c), ④: EncoderFrame_Bottom (A6d), and ⑤: M4Spacer (A6e). (b), Cross-sectional view with the fasteners indicated. The H4d screws securing the ④ to the Upper arm Frame should be countersunk in the direction from the ④ toward the Upper arm Frame. Similarly, H4g screws assembling Upper arm Frame, ④, and A1 should be countersunk in the direction from the frame to the motor. The holes for setscrews (H4c) of ② should be threaded after being printed or manufactured. Screws not specifically indicated are H4e, M3 × 10mm; nuts not specifically indicated are H4k, M3.

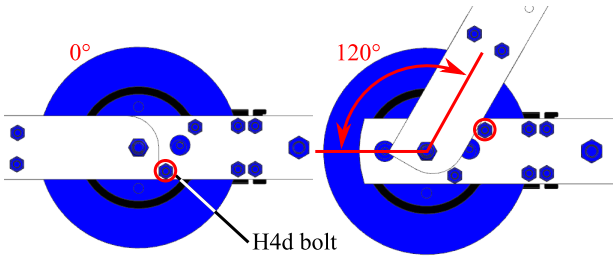


Figure 2.7: Mechanical hard stops limit the arm movement from 0 to 120°. These mitigate any mechanical risks from accidentally overpowering the motor.

The first manufacturing steps are to attach the motor and encoder to the base frame and ensure they are functioning correctly. To complete the base frame, four CAD files (②, ③ × 2, and ④ in Figure 2.6) should be built. Most pieces of the frame are thin aluminum sheet metal (H1), so simple hand tools suffice for most of the manufacturing steps. The fabrication process is easily performed by printing out the downloaded CAD drawings (one to one scale in A3 paper) and pasting them to the Aluminum sheet, which provides a template for cutting or drilling the metal sheet (Ⓐ and Ⓑ in Figure 2.3). When fabricating sheet metal parts, be careful to keep metal dust away from the motor, as it can get stuck between the rotor and stator and cause it to jam. ② in Figure 2.6 is the

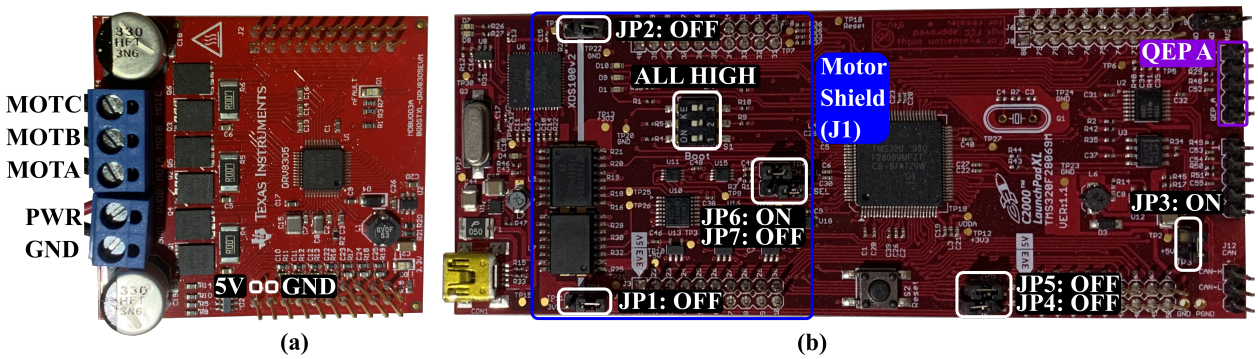


Figure 2.8: (a) Wiring for the DRV8305EVM motor controller shield and (b) Settings for the C2000 board ([63]). In (a), pin 21 and 22 are used for running the Miniature 5V Cooling fan. In (b), **White** indicates the jumper and switch settings; **Blue** is where the motor shield should go; **Purple** is where the encoder should be attached.

most critical part in the fabrication process as the motor control relies on the encoder readings. The best way to manufacture ② in Figure 2.6 is to use a lathe, although high-resolution 3D-printing works as well. Whether the encoder shaft functions properly or not can be tested using MotorWare lab12b, sensored speed control. Two holes require tapping, indicated by red circles in Figure 2.7. These are travel limits for the exoskeleton to prevent over-rotation. When the motor is securely mounted on the upper arm frame through ④ and aligned with the encoder, the motor is ready to be tested.

*System integration test*

Following the mechanical assembly of the motor, encoder, and base frame, the hardware and software functionality should be verified. The primary focus of this part is to check if the hardware and original MotorWare projects function appropriately before applying the custom code provided with this paper. First, the jumpers on the C2000 board must be set up correctly, as shown in Figure 2.8(b). The motor controller should be attached to the C2000 board, and the motor wires and encoder connected, as in Figure 2.8(a).

Two steps must be completed before running the motor: (1) figuring out the correct combination of motor windings, and (2) identifying the sensorless motor parameters, such as the winding's inductance, resistance, and magnetic flux. Both steps are performed simultaneously through TI's sensorless parameter identification project (lab2a or lab2c, provided by MotorWare [62]). Both projects yield the `user_j1.h` file for sensorless control, which contains the required parameters; the projects function the same way, but lab2c is for low inductance motors. With a BLDC motor with many pole pairs (i.e., 24N28P), potentially lab2a yields too low of an inductance reading ($<$50 $\mu$H) which will possibly result in an error during other projects. In this case, lab2c should be used. Aside from lab2a and lab2c, the other projects in MotorWare require the identified motor parameters and so cannot be run until the motor parameters are determined. Accordingly, lab2a or lab2c must be used to figure out the correct mapping between motor windings (U, V, and W) and the motor controller's PWM channels (MOTA, MOTB, and MOTC in Figure 2.8(a)). An appropriate winding combination should lead to finishing the parameter identification project (lab2a or lab2c) without any pause in the motor's rotation, vibration, or unusual behavior (i.e., rotating backward).

MotorWare projects share two versions of motor parameter files; one for the sensorless control (FOC) and another for the sensored (InstaSPIN). Both motor parameter files are named `user_j1.h` but stored in different paths. Once the sensorless `user_j1.h` is obtained through lab2a or lab2c, the sensored motor parameters, like motor inertia and friction, should also be measured in lab12a. Sensorless projects utilize the FAST estimator, which runs the motor model internally based on the back-EMF readings. In comparison, sensored projects (e.g. lab12a) use encoder readings. Thus, those two types of projects share different motor parameter files.

To run lab12a, `USER_MOTOR_MAX_SPEED_KRPM` in `user_j1.h` has to be set to be 1. While running lab12a, misalignment of the encoder shaft could lead to skipping the index reading, resulting in a malfunction of the motor. The connection between the encoder and motor can be adjusted if this occurs. We use an encoder with a built-in shaft (A4) because in prototype development we found

that it was extremely difficult to align a shaft-less encoder with the motor properly.

Upon both the sensorless `user_j1.h` and sensored `user_j1.h` being ready, simple motor control can be tested by running the torque controller (lab4) and speed controller (lab12b) that come with MotorWare. A successful run on project lab4 ensures that the sensorless parameters are identified appropriately, and running the speed controller (lab12b) verifies the encoder setup and other aspects of the hardware system.

Following is a summary of the software steps to this point:

- Install MotorWare (S1).
- Prepare the C2000 microprocessor with the proper jumper configuration (Figure 2.8).
- Perform Motor Identification (lab2a or lab2c and lab12a) to create both sensorless and sensored versions of `user_j1.h`.
- Perform torque control (lab4) and speed control (lab12b) to check the system functionality.

### *2.5.2 Index-pin calibration*

In this stage of the build process, we install the custom microcontroller code provided with the paper, as well as set up the encoder and align its index location within an appropriate rotation range of the motor. It is necessary to calibrate the index pin before before finalizing the whole assembly because the inertia of the exoskeleton hinders the motor alignment if done at the end.

This section involves changing the peripherals' priorities. When modifying the peripherals' priorities all the projects linked to the source file should be updated simultaneously. Thus, after this step, all other projects in MotorWare will not compile.

#### *Custom microcontroller code setup*

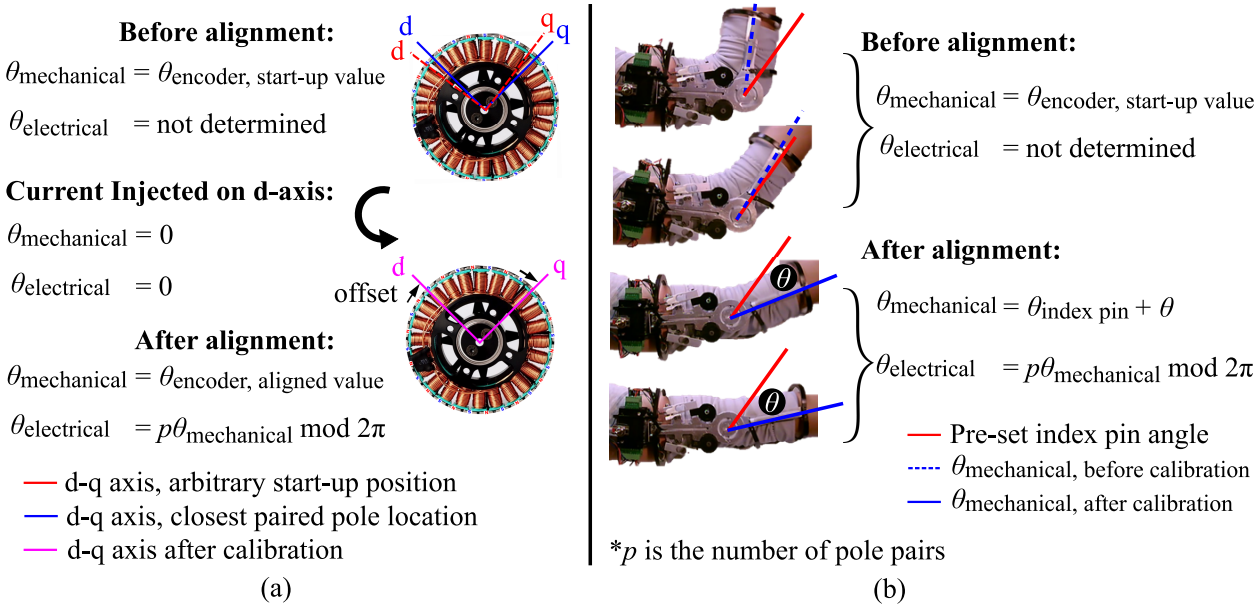


Figure 2.9: (a), Angle estimation process with TI's sensored control. The code is programmed to initialize both mechanical and electrical angle to zero as the current is injected. (b), Angle estimation through a predefined index pin, as is done in our system. Current injection is a very useful approach, yet is not feasible for a direct-drive wearable device.

Following the basic motor setup and testing in the Base Motor Setup stage, this section addresses how to set up the custom microcontroller code provided with this paper. The microcontroller source code files provided with this paper (S1) are based on lab12b and thus share all the embedded features with that project. Linking scripts (S1b and S1d) to the lab12b workspace (WS) and replacing one header file (S1c) enables us to access the key features in MotorWare. All files (S1) should be located on the specific path `MW_INSTALL_DIR\sw\solutions\instaspin_motion\src`. The default path of MotorWare is `C:\ti\motorware\motorware_1_01_00_18\`and its relative path is defined by Eclipse as `MW_INSTALL_DIR`. One essential script that is not included on the list but still required to be linked to the WS is the `sci.c` file. The file can be found in `MW_INSTALL_DIR\sw\drivers\sci\src\32b\f28x\f2806x`. To link the script to an existing WS, use "Add files" from the project context menu of lab12b, which is accessible by right-clicking the project name on the Project Explorer. Then locate where the downloaded the scripts (S1b and S1d) are. Create links to the files relative to PROJECT_LOC. Detailed steps to link files to the current WS is explained

in Section 6.7 in [64]. After linking the custom code, remove the original `lab12b.c` link file as the .obj file is one-to-one mapped with the .c file.

In the custom code, we introduce several interrupt routines to the system. Since the main Interrupt Service Routine (ISR) of the MotorWare is directly related to the PWM generation, ordering the priorities of these interrupts is critically important for the system's proper functioning. Figure 2.2 shows two different priorities, CPU and PIE. Both are part of the interrupt management levels of Digital Signal Processing. If an interrupt event takes place and interrupt flag bit is on, the interrupt request is delivered to the PIE controller. The request is then managed at the CPU level (Chapter 1.7 in [61]). Among priorities illustrated in Figure 2.2, the PIE priority of the `timer0ISR()` is the default setting and does not need any modification. Both PIE and CPU settings of the `sciBRxISR()` are declared as the uploaded `HardwareX_comm_N_timer.c` is linked to the WS. The CPU priority of the `timer0ISR()` and both priority settings of `ADCINT1()` need to be manually changed. Two relevant files, `hal.h` and `hal.c` of the Hardware Abstract Layer, address the general peripherals' priorities and initialization ([64]) and must be edited as follows:

In `hal.h`:

| | |
|---|---|
| at "the globals", | add: `extern interrupt void sciBRxISR(void);`<br>add: `extern interrupt void timer0ISR(void);` |
| at `HAL_acqAdcInt()`, | change the parameter of `PIE_clearInt()` from<br>`PIE_GroupNumber_10` to `PIE_GroupNumber_1` |
| at `HAL_initIntVectorTable()`, | change the PIE priority of `pie->ADCINT1 = &mainISR;`<br>from<br>`pie->ADCINT1` to `pie->ADCINT1_HP`<br>add: `pie->SCIRXINTB = &sciBRxISR;`<br>add: `pie->TINT0 = &timer0ISR;` |

Similarly, in `hal.c`:

| | |
|---|---|
| at `HAL_enableAdcInts()`, | change the parameter of `PIE_enableAdcInt()` from `ADC_IntNumber_1` to `ADC_IntNumber_1HP` |
| | change the parameter of `CPU_enableInt()` from `CPU_IntNumber_10` to `CPU_IntNumber_1` |
| at `HAL_enableTimer0Int()`, | change the parameter of `CPU_enableInt()` from `CPU_IntNumber_1` to `CPU_IntNumber_2`; |

*Processing code setup and serial communication with CCS*

Following the setup of the initial microcontroller code, we need to prepare the computer code. Download and prepare the Processing script and select the appropriate COM port for serial communication. If CCS has trouble connecting to the board, make sure that the TI's COM Port has the virtual terminal option on, in the Device Manager. The port number set should be an auxiliary port for TI. Two supplementary files, the communication states file (`receive_Stat.java` - S2b) and the pre-programmed trajectory (`traj_1.txt` - S2c), have to be in the same folder with the `Processing_HardwareX.pde`.

*Assemble the joint actuator*

We next assemble the forearm side of the exoskeleton frame. The forearm assembly is attached to the Base Motor Setup (Figures 2.5 and 2.6) with the 3D-printed spacer (⑤ in Figure 2.6) in between the aluminum frames. An M4 screw (H4n) passes through the motor (A1), motorFrameAdapter ④, and frameSpacing ⑤ , and is secured with a lock nut (H4o). The most important part of the assembly is that the index pin should be in the range of motion. To assure this, the reading values for the index pin should be monitored during the assembly process via the Real-Time Watch Window in CCS. The entire operation steps are closely related to TI's state machines of: `Flag_enableSys` – system flag to initiate background loop, `Flag_Run_Identify` – flag to check the motor parameter before

Controller API, `CtrlState` – state for the Controller API, and `EstState` – state for Estimator API. With the motor and encoder connected, but not yet secured:

1. **CCS**: At the top of the `haptic_joint.c`, locate where all the `defines` are declared. Uncomment `AngleResetViaIndxPin` and comment `AngleResetViaIndxPin_Dbug` out.
2. **CCS**: Build the project: Debug project, Enable Silicon Real-time mode, Enable continuous refresh, and Resume.
3. **Processing**: Execute the `Processing_HardwareX.pde` by pressing Run button. This initiates interrupt events for `sciBRXISR()` and `timer0ISR()` on the microcontroller. Check if `Flag_enableSys` is 1, and `Flag_Run_Identify` is 0 in the Real-Time Watch Window, Expressions tab.
4. **CCS**: Create a variable `mtrPosInitiated` in bool format in the Real-Time Watch Window, Expressions tab.
5. Try rotate the motor from the initial arm angle to the desired maximum angle, and see if the `mtrPosInitiated` variable changes to 1.

Repeat step 5 until the index-pin is detected within the range of motion. The index pin on the encoder should be verified to be located within the arm's traveling range before finalizing the forearm frame's location relative to the motor and proceeding to subsequent steps (Figure 2.9). Secure the encoder to the motor with setscrews (H4c) in part ② after the index pulse is observed and terminate the system by pressing the space bar key.

*Perform index-Pin calibration*

The following are the steps to calibrate the index pin (Figure 2.9) after the hardware assembly. As the encoder reading determines the electrical angle, running the motor without calibration might lead to unexpected motor operation.

1. **CCS**: At the top of the `haptic_joint.c`, locate where all the `defines` are declared. Uncomment `AngleResetViaIndxPin` and comment `AngleResetViaIndxPin_Dbug` out.
2. **CCS**: Build the project: Debug project, Enable Silicon Real-time mode, Enable continuous refresh, and Resume.
3. **CCS**: Create variables, `debug_IndxCnt_1` and `debug_zeroOffset` in the Real-Time Watch Window, Expressions tab.
4. **CCS**: Stop the code and terminate.
5. **CCS**: Uncomment both `define`:`AngleResetViaIndxPin` and `AngleResetViaIndxPin_Dbug`.
6. Make the exoskeleton fully extended (straightened) before initializing the code. Also, make sure the motor is flat on the ground to mitigate possible tilting.
7. **CCS**: Build the project: Debug project, Enable Silicon Real-time mode, Enable continuous refresh, and Resume.
8. **Processing**: Execute the `Processing_HardwareX.pde` by pressing Run button. This initiates interrupt events for `sciBRXISR()` and `timer0ISR()` on the microcontroller. Check if both `Flag_enableSys` and `Flag_Run_Identify` are 1 in the Real-Time Watch Window, Expressions tab.
9. **CCS**: As the current is injected, motor aligns itself to the closest paired pole location as in Figure 2.9(a). The current injection process also can be monitored through the power supply. Occasionally, the motor rotates even after the alignment. In this case, restart from step 1.
10. **CCS**: Record value in `debug_zeroOffset`, save it for later use. This is the offset between the desired initial position and the closest paired pole location.
11. Swing the exoskeleton to the maximum range of motion.
12. **CCS**: Record the `debug_IndxCnt_1` value for later use. This shows where the index pin is located.
13. **Processing**: Press a space bar to terminate the system.

14. In the `timer0ISR()`, `haptic_joint.c`, replace the default value in `ENC_setZeroOffset()` with the `debug_zeroOffset`. Also, replace the default value for `IndxOffset` with the measured `debug_IndxCnt_1`.
15. **CCS**: This step is to verify the current setting works. Rotate the motor to an arbitrary position. Uncomment `AngleResetViaIndxPin` and comment `AngleResetViaIndxPin_Dbug` out. Repeat the Step 6.
16. Execute the `Processing_HardwareX.pde` by pressing Run button. Check if `Flag_enableSys` is 1 and `Flag_Run_Identify` is 0 in the Real-Time Watch Window, Expressions tab.
17. **CCS**: Add a variable `mtrPosMrev_iq` in the Q-Value(24) format in Real-Time Watch Window. The reading from the variable is an arbitrary reading recorded when powering up the encoder.
18. **CCS**: Move the motor until `mtrPosInitiated` changes from 0 to 1. When the index-pin is detected, the value of `mtrPosMrev_iq` should be updated to the calibrated position.
19. **Processing**: Press a space bar to terminate the system.

Whenever the actuator module requires re-assembly, the index pin should be re-calibrated. However, once we set the index pin value, the value should not be changed unless the encoder shaft slips or is detached. In summary,

- Download the code files associated with this paper (S1) and extract them into the appropriate path (Section 2.5.2). Note that `main.h` replaces the original file and the other files are simply pasted there.
- Link files, S1b and S1d, to the lab12b WS (including `sci.c`).
- The existing priority settings should be adjusted in `hal.c` and `hal.h`.
- Set up Processing (S2) and test the serial communication.
- Assemble the encoder and the forearm frame, making sure the index pin location is in the range of the exoskeleton's rotation.

- Calibrate the index pin following the procedure in Section 2.5.2.

### *2.5.3 Finish exoskeleton*

*Fabricate braces and adjust their ergonomic locations*

In this stage, we finish assembling the aluminum braces, taking into consideration the ergonomic requirements of the arm. Just like the aluminum manufacturing process for the Base Motor Setup stage, please refer to the CAD drawings (`frame_drawings.pdf`). Figure 2.5 shows the new components added in this stage.

To assemble the rest of the exoskeleton, first bend the aluminum strips using the PVC pipe Ⓒ and hammer Ⓓ (see Figure 2.3). Once the aluminum frames and braces are assembled, attach the Nylon sheet (H3) to the frames with screws (H4j) (Figure 2.10). The Nylon sheet is cut to 12 cm $\times$ 38 cm for the forearm piece, and 11 cm $\times$ 48 cm for the upper arm piece. Foam (H2) is also cut to be similar sizes. It could be attached to the Nylon sheet with hot glue, although in our instructions we leave them as separate pieces. The foam provides additional comfort to the wearer and helps the exoskeleton conform to various arm geometries. Next, attach the miniature fans (A5) which are

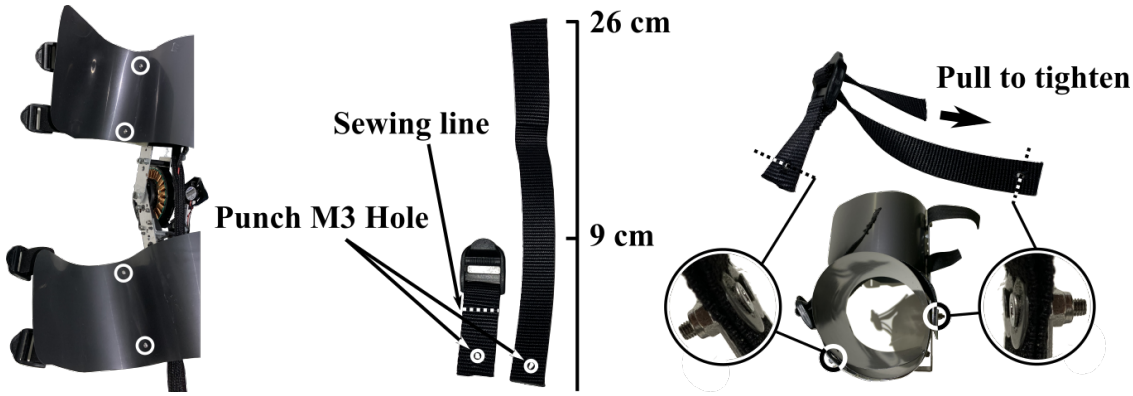


Figure 2.10: Left: four highlighted screw (H4j) points to secure the Nylon sheet to the frame. Middle: manufacturing guideline for straps (H4a) and buckles (H4b) as where to sew, punch, and screw. Right: mounting process to frames. From the medial to the lateral side, secure the screws in the order of: screw (H4e); washer (H4m); straps (H4a); lock nut (H4k).

necessary to allow the motor to dissipate heat during stall with screws (H4d and H4e). Finally, sew and punch (Ⓕ in Figure 2.3) the straps (H4a and H4b), and attach them to the braces with screws (H4e in Table 2.1).

*Conduct position error-driven torque control*

The exoskeleton provides a torque as a function of the position error:

$$I = P(x - x_d) - D\dot{x} \tag{2.1}$$

$$\tau = K_t\, I \tag{2.2}$$

where $I$ is the current, $P$ is the stiffness gain, $D$ is the damping gain, $x$ is the mechanical position, $x_d$ is a specified desired position transmitted from Processing, and $\dot{x}$ is the filtered speed feedback. $\tau$ is the torque created by the motor, and $K_t$ is the motor torque constant. In the microcontroller code, $x$ and $x_d$ have units of [revolutions], $\dot{x}$ has units of [krpm], $\tau$ has units of [Nm], and $I$ has units of [PU] ("per unit") where the current is divided by a user-defined full-scale current. Our Processing code provides two different modes for position input, such that the controller could be tuned before applying a predefined path. One is direct-position input, where the experimenter adjusts the position input by pressing up/down on the keyboard. This mode is appropriate for tuning the PD controller before applying any motion trajectory. Another mode is the preprogrammed trajectory. The sample external trajectory file, `traj_1.txt`, is loaded via multi-threading and sent to CCS. The provided sample path is in degree and is updated in `Processing` at 100 Hz. Users can switch between the two modes by toggling `positionInputMode`: 0 for direct-input, and 1 for preprogrammed path.

The P and D gains for the torque control are empirically tuned and can be adjusted according to the desired actuator bandwidth. The current feeding into the system can be observed by adding a variable `iqReference` in the Q-Value(24) format in Real-Time Watch Window. When performing

real-time tuning, variables corresponding to the P, D gains should be typed in Real-Time Watch Window: `tuning_KP` and `tuning_KD` in the Q-Value(24) format. Start with a very low number (i.e., $0.001$) to begin the tuning safely and then increase the value after observing how the exoskeleton responds. Usually, the P gain value should be tuned first to roughly estimate how fast the motor should rotate, followed by adjusting the D gain to alleviate the overshoot. Typically, motors with strong static friction require higher P gains. If the motor is unstable upon powering up (i.e., vibrates), the system requires a higher damping value to be stable. In summary, in this stage the steps are:

- Fabricate the braces and attach them.
- Attach the Nylon sheet (H3).
- Connect the fans.
- Sew and punch (Ⓕ) straps, tailor them, and link them to the braces.
- Empirically tune the P and D gains for position-error based torque controller.

### *2.5.4 Design considerations*

In this section, we elaborate on design considerations for the system. Specifically, we provide additional details on designing the human interface, sizing a motor, and improving communication between Processing and the microcontroller. The process outlined in this paper is adaptable to other motors besides the one selected for this project; this section explains how to choose a new motor and use it with the software.

#### *Designing the human interface*

The frame design should allow the exoskeleton to be comfortable throughout the arm's range of motion. Instead of a metal frame completely surrounding the arm, the arm braces on the medial

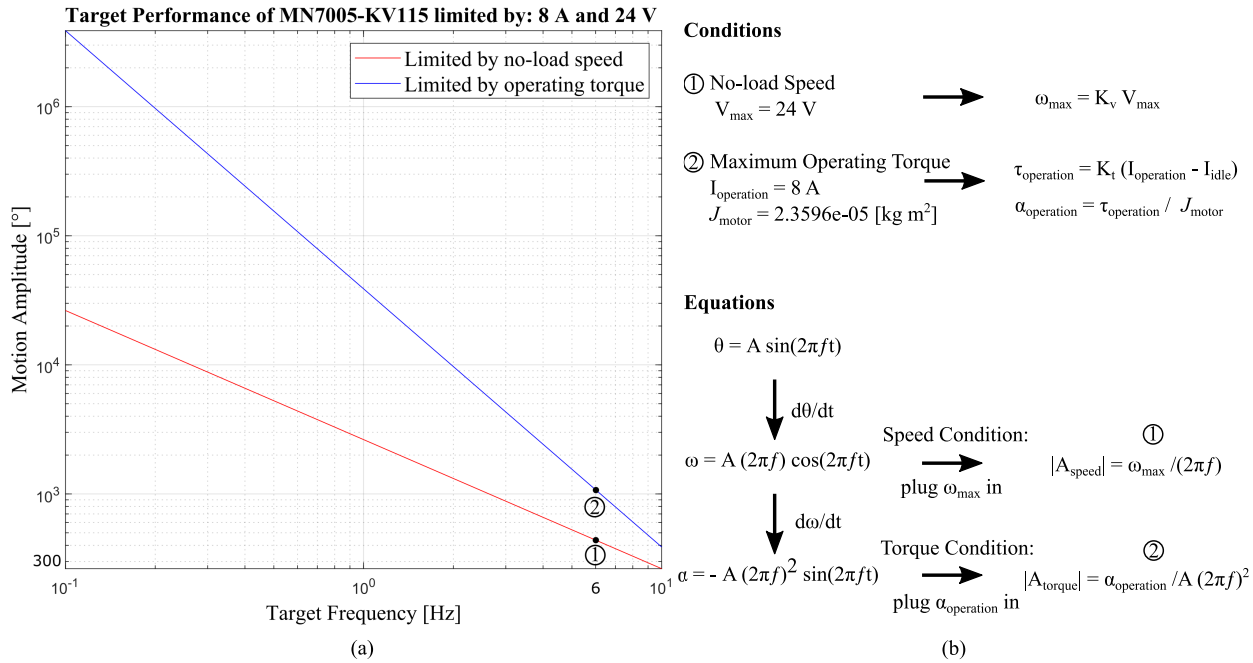


Figure 2.11: (a), Simulation of the exoskeleton described in this paper. The motion amplitude achievable at different frequencies. Both conditions yield motion amplitudes of more than 300° at 6 Hz, which is larger than our target amplitude of 50°. Note that the peak-to-peak angle of the motion is twice the amplitude. The simulations assume a voltage of 24 V and a maximum operating current of 8 A. (b), Summary of the calculations for how to determine the required motor size. In the last steps for finding $|A_{speed}|$ and $|A_{torque}|$, $\cos(2\pi ft)$ or $\sin(2\pi ft)$ are set to 1, as that is the maximum value. $J_{motor}$ is identified from lab2c and converted using an equation (Section 10.5 in [62]). The inertia of the exoskeleton frame is neglected in this example.

side are made of soft fabric while the bottom half remains made of rigid parts. This enables the braces to deliver the torque from the motor while not impeding the range of motion when the arm is fully flexed. Second, the exoskeleton should adapt to the muscle volumes, as the arm muscles can change drastically during arm movement. This is accomplished by the plastic sheets held in place by elastic straps. Third, the foam on the inside of the exoskeleton conforms to the wearer, as the plastic sheets are cylindrical, and also provide a soft interface. The exoskeleton can be adjusted for different wearers' geometries by bending the aluminum or altering the foam thickness.

*Sizing motors for passive control*

Passive control in motion training can be simplified to an actuator's motion: the actuator applies the desired motion regardless of the wearer's force feedback. The problem can be easily analyzed if the desired motion is a simplified sinusoidal motion, and we ignore any interaction forces with the wearer. An overview of the calculations needed to select a motor are shown in Figure 2.11, and the equations are also as follows:

$$\omega_{max} = K_v V_{max} \tag{2.3}$$

$$\tau_{operation} = K_t(I_{operation} - I_{idle}) \tag{2.4}$$

$$\alpha_{operation} = \tau_{operation}/J_{motor} \tag{2.5}$$

$$\theta = Asin(2\pi ft) \tag{2.6}$$

$$\omega = A(2\pi f)\cos(2\pi ft) \tag{2.7}$$

$$\alpha = A(2\pi f)^2 \sin(2\pi ft) \tag{2.8}$$

$$|A_{speed}| = \omega_{max}/(2\pi ft) \tag{2.9}$$

$$|A_{torque}| = \alpha_{operation}/(2\pi ft)^2 \tag{2.10}$$

where $V_{max}$ is the maximum voltage; $\omega_{max}$ is the maximum motor speed; $K_v$ is the motor velocity constant; $K_t$ is the motor torque constant; $I_{operation}$ is the maximum operation current; $I_{idle}$ is the no-load current; $J_{motor}$ is the motor rotor inertia; $\alpha_{operation}$ is the angular acceleration of the motor; $\tau_{operation}$ is the motor torque; $\theta$ is simulated motion; $A$ is the magnitude of the trigonometric functions; and $f$ is the frequency, which ranges from 0.1 to 10 Hz in Figure 2.11. The angular speed (Eq. 2.7) and angular acceleration (Eq. 2.8) are generated from the simplified motion equation (Eq. 2.6) and taking derivatives.

The motor performance is evaluated in two aspects: no-load speed and maximum operating torque.

The chosen motor's velocity constant ($K_v$) determines the no-load speed (Eq. 2.3), given the maximum voltage. In comparison, the motor torque constant ($K_t$), motor inertia, and the maximum operating current decide the operating torque condition (Eqs. 2.4 and 2.5). The maximum operating current should be smaller than the rated peak current since the joint actuator mostly operates in a stall condition.

To determine the sinusoidal motion amplitude possible at different frequencies, the maximum angular velocity $\omega_{max}$ is substituted into Eq. 2.7 and the amplitude is solved for, after setting $\cos(2\pi ft)$ to $1$ since that is the maximum value. This results in Eq. 2.9, which is plotted as the red line in Figure 2.11(a). Similarly, the amplitude limited by the maximum torque is found by substituting a constant $\alpha_{operating}$ into Eq. 2.8 and setting $\sin(2\pi ft)$ to $1$, and this is plotted in Figure 2.11(a) as the blue line. The target performance of 50° amplitude for 6 Hz as the maximum human arm motion can be easily achieved with the given motor inertia. However, the inertia of the attached exoskeleton frame is neglected in the simulation.

Some parameters utilized in the calculation are retrieved from the specification sheet [59], which is measured under continuous rotation. Therefore, parameters sensitive to the heat, such as the coil resistance, could be different when it comes to running the motor under stall condition, and some care should be taken when doing calculations.

In general, for direct-drive applications, outrunner BLDC motors are especially suitable because of their high torque-to-weight ratio. When selecting a specific motor, a high pole-pair number leads to a lower cogging effect from the magnets being attracted to the stators.

*Improving communication*

Table 2.2 provides an overview of the communication between CCS and `Processing`. As shown there, the communication process between the microcontroller and Processing involves three inter-

**Communication Algorithm**

`serialEvent()` in Processing

```
toggle ← False
command ← desiredInput
if serial port available then
   incoming8bit ← in short
   if toggle == True then
      switch state_dataID
         case == state_Data01
            Store: LSB → MSB and Compose
         case == state_Data02
            Store: LSB → MSB and Compose
         case == state_Data03
            Store: LSB → MSB and Compose
         case == state_Data04
            Store: LSB → MSB and Compose
         case == state_Timer
            Store: LSB → MSB and Compose
            Transmit: command

            Update: savingDataArray
         end switch
   else
      switch incoming8bit


         case == dataID_01
            state_dataID ← state_Data01
         case == dataID_02
            state_dataID ← state_Data02
         case == dataID_03
            state_dataID ← state_Data03
         case == dataID_04
            state_dataID ← state_Data04
         case == dataID_Timer
            state_dataID ← state_Counter
         default
            toggle ← True
            LSB ← 0

            MSB ← 0
         end switch
      end if
end if
```

`sciBRxISR()` in Code Composer Studio

```
toggleCMDIn ← False
dataGet8bit ← getDataNonBlocking()
if toggleCMDIn == True then
   Store: LSB → MSB and Compose
else
   switch dataGet8bit
      case c
         toggleCMDIn ← True
      case 125
         enable: Motor State and timer0ISR
      case 124
         disable: Motor State and timer0ISR
      case 123
         initializeExperiment ← True
   end switch
end if
```

`timer0ISR()` in Code Composer Studio

```
flat_updatePkg ← False
HighOrLow ← False
updateDLogPkg()
dataPointer = (uint16*) dataPointerArray
[cntr]
if HighOrLow == True then
   cntr++
   transmit8bit ← (*dataPointer >> 8) & 0xFF
else
   transmit8bit ← (*dataPointer) & 0xFF
end if
SCI_putDataNonBlocking(transmit8bit)
HighOrLow ^= 1
cntr % = size(dataArray)
if cntr == 0 then
   flat_updatePkg ← True
end if
```

`updateDLogPkg()` in Code Composer Studio

```
if flat_updatePkg == True then
   Update: dataArray
   flat_updatePkg ← False
end if
```

Table 2.2: Pseudo-code for communication. The ASCII IDs tagged for the current data package are c, 125, 124, and 123 in decimal. Those are arbitrarily chosen and can be modified to be other ASCII codes. Switching to other variables can be done in `updateDLogPkg()`.

rupt routines, `serialEvent()` of `Processing`, and the `timer0ISR()` and `sciBRxISR()` of CCS ([61]). The current serial communication setting has one stop bit with no parity check. To transmit

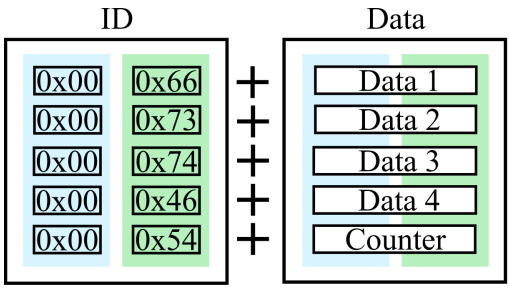


Figure 2.12: Data Package for logging from CCS. The package is made of 10 bytes of ID and 10 bytes of data (five 16 bit data). Five data IDs, 0x66, 0x73, 0x74, 0x46, and 0x54, are represented as `dataID_01`, `dataID_02`, `dataID_03`, `dataID_04`, and `dataID_Timer` in `void serialEvent()`, Table 2.2. The IDs are arbitrary chosen and could be changed to different characters. The five variables currently being logged are motor position, filtered torque, current feedback, commanded current, and timer counter.

20 Bytes of the data package, illustrated in Fig. 2.12, the number of actual bits delivered via serial frames is 200 bits (Table 2.2). For the selected baud rate of 57600 (Table 2.3), which is a bit rate of 1.74E-5, the data package can be sent at around 288 Hz. This is more than twice as fast as the update framerate of `Processing_HardwareX.pde`, which is 100 Hz.

From `timer0ISR()`, the data array is a pointer array and updated every other `timer0ISR()` call (Table 2.2). The bit shifting on CCS side is written as little-endian as C2000 only supports the little-endian format. To implement the data package in both Processing and CCS, we use `short` in Processing and `unsigned int16` in CCS as the data type storing the received data. Note that `timer0ISR()` triggers `updateDLogPkg()` with the MotorWare's default framerate of the timer0, 2 kHz. To synchronize all five data (including the counter in Figure 2.12), `updateDLogPkg()` does not refresh until `timer0ISR()` finishes sending the whole packet. Ten iterations of data transmission are required for every data logging packet (five 16-bit numbers, transmitted 8 bits at a time at each `timer0ISR()` interrupt) and so the full data packet updates at 200 Hz. Within `Processing`, the function `void keyPressed()` enables users to use keyboard as an input source. Currently, the coded keys are: 1) arrow up/down, 2) Enter, and 3) space bar. These can be switched to be different keyboard inputs.

| BRR in decimal | BRR in binary | Baud rate, 15 MHz of LSPCLK | Baud rate, 90 MHz of LSPCLK |
|---|---|---|---|
| 194 | 11000010 | 9600 | 57600 |
| 97 | 1100001 | 19200 | 115200 |
| 33 | 100001 | 57600 | 330882 |
| 15 | 1111 | 115200 | 703125 |
| 9 | 1001 | 1125000 | 1125000 |

Table 2.3: Values for BRR, the 16-bit baud-select register. The default enumeration structure defined in `sci.h` assumes a serial clock of 15 MHz (low speed peripheral clock, LSPCLK), where setting BRR to 194 leads to a 9600 baud rate. This system described in this paper uses a Baud rate of 57600. For the detailed table, refer to SPRUH18G [61], Table 13-11.

## 2.6 Operation instructions

### *2.6.1 Exoskeleton donning instructions*

The donning process is shown in Figure 2.13. First, wear the foam to mitigate any possible discomfort between the plastic cover and skin. The foam is simply wrapped around the arm such that different sizes of arm volumes can be accommodated easily. Second, align the elbow joint with the motor. Third, make sure the plastic cover overlaps internally and tighten the straps. The plastic material cover should slide in easily and secure the foam to the arm.

### *2.6.2 Activation instructions*

Initialization steps are conducted in both CCS as a debugging tool and `Processing` as a User Interface. Steps for enabling the ISRs and activating the motor are separated, as the index pin reset module instantly updates the current arm position relative to the index pin position. This causes

Figure 2.13: Instructions for donning the exoskeleton, in order counter-clockwise. First, a sleeve is put on the wearer, followed by wrapping the arm in foam. Next, the exoskeleton is wrapped around the foam, and the straps tightened.

an abrupt torque surge if it occurs when the motor is enabled. The steps to use the device as as follows:

1. **CCS**: At the top of the `haptic_joint.c`, locate where all the `defines` are declared. Uncomment `AngleResetViaIndxPin` and comment `AngleResetViaIndxPin_Dbug` out.
2. **CCS**: Build the project: Debug project, Enable Silicon Real-time mode, Enable continuous refresh, and Resume.
3. **Processing**: Set a proper mode for position input by toggling `positionInputMode`. Setting the variable to 0 enables the direct-position input; while applying as 1 starts preprogrammed trajectory mode. When using the preprogrammed trajectory mode, make sure to load a correct path file. The path file should be in the same directory with the `.pde` file.
4. **Processing**: Execute the `Processing_HardwareX.pde` by pressing Run button. Check if `Flag_enableSys` is 1, and `Flag_Run_Identify` is 0 in the Real-Time Watch Window, Expressions tab.
5. **CCS**: Swing the arm so that the index-pin is detected while the arm travels. Detecting the index-pin initiates interrupt events for `sciBRXISR()` and `timer0ISR()` on the microcontroller. Also, `Flag_Run_Identify` becomes 1 in the Real-Time Watch Window, Expressions tab.

6. **CCS**: Wait until `CtrlState` and `EstState` become Online in the Real-Time Watch Window, Expressions tab.
7. **Processing**: Serial communication can be tested by whether or not the position data is logged without any missing bits. By default, a simple GUI should pop up after execution that displays the current joint angle in revolutions.
8. **Processing**: Move the arm to the position where the initial position input is programmed, and press the Enter key to enable the motor. Depending on the mode chosen in Section 2.5.3, exoskeleton will either guide the arm to the zero degrees position or apply the pre-programmed path. If the P gain is set too high or the inertia of the system is too low, the position error surges, resulting in the motor becoming unstable. However, please note that when a human is wearing the exoskeleton, the human arm provides additional damping to the system, and the proportional gains can be set to be much larger than if the exoskeleton is operating without a human inside. For the direct-position input, pressing up/down is programmed to increase/decrease the position by 0.01 rev.
9. **Processing**: Upon finishing the experiment, pressing the space bar key in the GUI terminates the `Processing` program and transmits a termination code to CCS. Both `Flag_enableSys` and `Flag_Run_Identify` become 0 in the Real-Time Watch Window, Expressions tab. This terminates `timer0ISR()` and motor control in CCS side, and `Processing` stores the logged data in a sub-folder, `log/` under the installed folder.

## 2.7 Validation and characterization

Several different demonstrations of the system guiding a human's arm are shown in Figure 2.14. In each case, a position trajectory was transmitted from `Processing` to the PD controller in CCS. During the motion guidance, the exoskeleton applies a torque pushing the wearer toward the nomi-

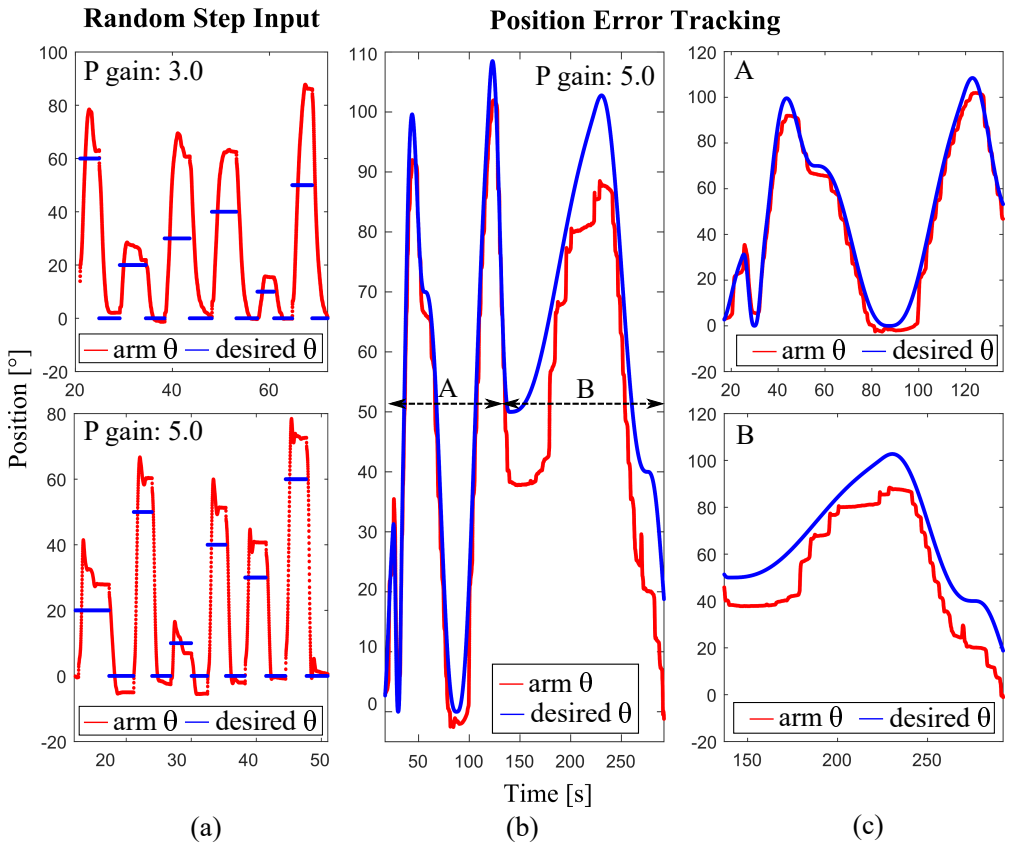


Figure 2.14: (a), Step responses with proportional gains (3.0 and 5.0). (b) and (c), Position tracking with kinesthetic guidance. (c) shows close-ups of different portions of (b). In all of the plots, the user's arm angle is shown with the solid red line. The blue line indicates the input position.

nal position whenever the wearer deviates from that position. The subject was blinded during these tests and could not see their arm's position. The tests were conducted with the wearer seated with their arm resting on a platform, as illustrated in Figure 2.1.

In Figure 2.14(a), proportional gains of 3.0 and 5.0 were each tested with six different step inputs ranging from 10-60°. The position inputs (blue lines) were applied in a random order via the exoskeleton, and the wearer tried to follow the applied torque (red lines). As can be seen, the wearer cannot perfectly follow the applied torque, and overshoots the desired final arm angle. The overshoots are larger when the gain is 3.0 (top) than 5.0 (bottom) because the wearer has difficulty differentiating between small positional differences when the proportional gain is low.

We also created a minimal-jerk trajectory, ranging from 0° to 100° (Figure 2.14(b) and (c)). The red lines in these plots, which are the human arm angle, show a staircase pattern where the arm pauses for a moment followed by a more rapid motion to move toward the intended position. This trend is shown more distinctively in the part of the trajectory with a less-steep slope, B (Figure 2.14(c), bottom plot) as compared to the A portion of the trajectory (Figure 2.14(c), top). At the end of section A, the subject had an overshoot of approximately -12 degrees. They maintained a negative offset from the desired angle throughout section B. The tracking performance is determined by how well the subject perceives the kinesthetic cue of the external arm torque. Given that steeper slopes yield smaller position errors (A compared to B), we assume that a low rate of position change leads to poor perceptual resolution. When the desired position is slowly changing as in the B region, the wearer might have difficulty differentiating the torque change, leading to poor tracking performance. In general, we have observed that the test results can be affected by several factors, including: (1), how well the subject can focus on the arm torques; (2), how familiar the subject is with the test setup; (3), how tight the straps are; and (4), the types of foams and sleeve used in the exoskeleton.

While these results are very preliminary, they demonstrate the exoskeleton's ability to be used as a motion training tool. This capability can be potentially used for rehabilitation or sports training, and the different hardware and software components can also be used separately to implement low-cost hobby motors for other actuator applications.

## 2.8 Acknowledgements

The authors thank Hongxu H. Guo for assistance with early versions of the code and index pin reset concepts, and thank Sungjae Ohn for assistance with the concepts of the field oriented control. This work was supported by an Institute for Critical Technology and Applied Science (ICTAS)

Fellowship for H. Kim and a Google Faculty Research Award. The sponsors did not have any role in the study design; collection, analysis, and interpretation of data; writing; or decision to submit the article for publication.

## 2.9 Declaration of interest

We declare that we do not have any known competing financial interests and personal relationships with other people or organizations that could have appeared to influence the work reported in this paper.

## 2.10 Human and animal rights

The study was approved by the Virginia Tech Institutional Review Board (IRB #16-175), and informed consent was obtained prior to the experiment.

# Chapter 3

# Just Noticeable Difference of Elbow Joint Torque Feedback

**Authors:** *Hubert Kim*[1] *and Alan T. Asbeck*[1]

**Affiliations:** [1]Mechanical Engineering Department, Virginia Tech, Blacksburg, VA 24061, USA.

**Contact email:** aasbeck@vt.edu

**Abstract:** Joint torque feedback is a new and promising means of kinesthetic feedback imposed by a wearable device. The torque feedback provides the wearer temporal and spatial information during a motion task. Nevertheless, little research has been conducted on quantifying the psychophysical parameters of how well humans can perceive external torques under various joint conditions. This study aims to investigate the Just Noticeable Difference (JND) perceptual ability of the elbow joint to joint torques. The paper focuses on the ability of two primary joint proprioceptors, the Golgi-Tendon Organ (GTO) and Muscle Spindle (MS), to detect elbow torques, since touch and pressure sensors were masked. We studied 14 subjects while the arm was isometrically contracted (static condition) and was moving at a constant speed (dynamic condition). In total there were 10 joint conditions investigated, which varied the direction of the arm's movement and the preload direction as well as torque direction. The JND torques under static conditions ranged from 0.097 Nm with no preload to 0.197 Nm with a preload of 1.28 Nm. The maximum dynamic JND torques were 0.799 Nm and 0.428 Nm, when the arm was flexing and extending at 213 degrees per second,

respectively.

## 3.1 Introduction

Joint torque feedback (JTF) is the sense of being guided or given information by a joint torque generated from a wearable device. This method of haptic feedback is analogous to training someone in how to move via one person manually moving another person's joint through different motions. Typically, this is used in rehabilitation, where someone is re-learning a motor skill, or it can also be used to teach someone a new skill. Joint torque feedback could also be used to convey a general continuous signal to a person, similar to force feedback with a joystick. JTF is a new type of haptic feedback, and has only sparsely been explored [18, 33, 34, 56, 65].

In general, haptic information involves mechanotactile sensors, vibrotactile sensors, and proprioception [29, 50, 51]. Most prior research has focused on skin receptors due to their ease of stimulation, and has specifically focused on the spatial resolution and sensitivity of skin sensors, such as through vibrotactile [4], skin stretch [5], and mechanotactile [2] feedback. In contrast, this study focuses on the proprioceptors within the arm. Unlike the haptic modalities detected by sensors located under the skin, proprioceptors are situated inside a joint—inside the muscles and tendons [29, 51] (Fig. 3.1 (a)). The primary afferent joint proprioceptors are Muscle Spindles (MS) and Golgi Tendon Organs (GTO), and they are located in muscle fibers and tendons, respectively [51] (illustrated in Fig. 3.1 (a) and (c)). MS are broadly known to detect the change of length and rate of length change of muscle fibers, and thus detect the position and movement of a joint; GTO are used to measure the tension of the muscle fibers [51]. Our objective is to explore using these sensors to convey haptic information to a person through JTF. This is natural feedback in the sense that these are the same sensory organs—proprioceptors—that we use in daily activities to get feedback about our body's position and the forces in our muscles. In motion training scenarios, conveying infor-

mation through the proprioceptors is typically referred to as kinesthetic or proprioceptive feedback [29, 51]. Kinesthetic feedback usually consists of position or velocity feedback, or force feedback applied through a handle or joystick. Joint torque feedback is a subset of kinesthetic feedback, where we are specifically concerned with the effects of torques applied across a joint.

Some prior work has been done that is broadly related to the proprioceptive sensors: researchers have quantified how much torque is required to drive a joint passively [40], and the amount of torque a joint can generate during isometric contraction has also been characterized [12, 15, 25]. Also, human perception when the arm is passively-driven has been explored [1]. Early studies utilized a contralateral limb-matching test to investigate the force resolution. Subjects' perceptual ability was determined based on how well they could reproduce the left arm's perceived force on the right arm [31]. Researchers also explored identifying force magnitudes presented with a stylus [14]. However, the torque that encourages a person's voluntary arm motion has not been fully characterized.

For motion training, understanding the minimum detectable torque at a joint is important. In prior work on motion training, researchers have identified two extremes: a fully patient-driven strategy, where no external assistance is provided, and a fully robot-driven strategy where the robot drives a person's limb through a desired trajectory and the person does not need to exert any effort whatsoever [53, 57]. Studies have found that the optimal assistance to maximize learning should be strong enough to cue the person to move and to realize when they have made an error, but not strong enough to hinder the subject's voluntary motion or fully constrain the wearer's desired motion [57, 69]. To meet these requirements, the torque from a wearable device for motion training should be relatively small but still convey information to the wearer, similar to how vibrotactile motors convey information without causing motion. Thus, it is crucial to understand the minimum perceptual limits of joint torques, in order to optimally convey information through kinesthetic guidance.

Other benefits of understanding the human perceptual resolution include optimizing exoskeletons, teleoperation, and Virtual Reality (VR) systems. For rehabilitation and motion training applications, examining the minimum noticeable level of torque provides an insight into how much parasitic torque is permitted in an exoskeleton or force feedback system without disturbing the wearer. For example, Schiele et al. found that the interaction torque between an exoskeleton and wearer due to mechanical constraints and misalignments could be as large as $1.46$ Nm [56]. In powered exoskeletons, sensor noise and delays in control systems can cause parasitic torques of $0.3 - 1$ Nm [32, 48, 66]. Understanding the human perceptual resolution is also important in fields where the transmission of physical interaction is required, such as teleoperation or haptic motor learning [72]. The technique called lossy compression utilizes the perceptual resolution for selectively eliminating haptic information outside of the human perception threshold [72]; the reduced data size can improve the efficiency of teleoperation or VR systems.

In this paper, our objective is to understand the minimum necessary amount of external torque for a person to feel its presence (i.e., the Just Noticeable Difference (JND) torque), both when a joint is stationary (static condition) and when it is moving (dynamic condition). The current work is an extension of a prior experiment that solely examined the static condition [34]. In the static condition, a fixed reference torque is applied to the elbow joint and the subject holds stationary to resist that torque, thereby isometrically contracting their arm muscles. On top of the fixed reference torque, additional test torques are applied briefly in order to measure the JND. In the dynamic condition, the arm moves back and forth continuously, and test torques are applied briefly when the arm is at a near-constant velocity. In all of the experimental conditions, pressure and touch sensations from the skin are masked [60] over the arm area so as to isolate the proprioceptors inside the arm during Joint Torque Feedback.

Exploring the human perception under the joint's dynamic condition is motivated by our pilot test results [33] in which a person was following a continuous trajectory while being guided by JTF. We

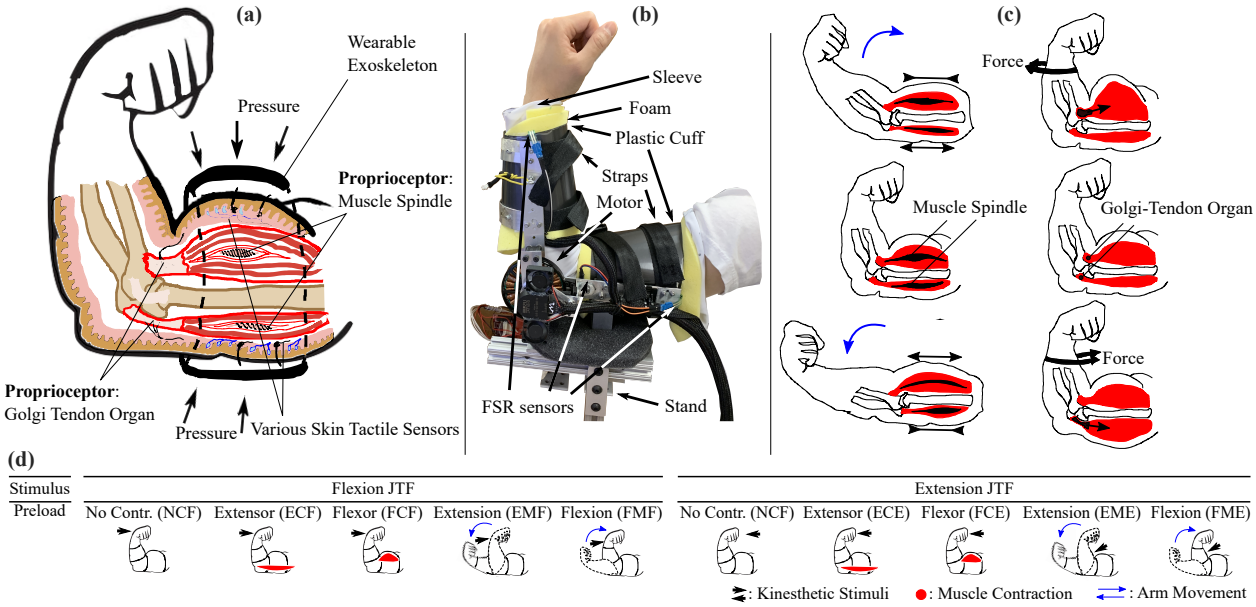


Figure 3.1: (a), Illustration of dissected arm view, showing the primary proprioceptors involved in joint torque sensing as well as how the exoskeleton applies pressure and masks skin receptors. (b), Picture of a participant's arm with the exoskeleton worn, showing the exoskeleton components. (c), Illustration of how each propioceptor is triggered under different joint conditions. The left column shows how the Muscle Spindles (MS) lengthen or shorten depending on the arm's movement. The right column illustrates how isometrically contracted flexor or extensor activity affects the Golgi-Tendon Organ (GTO). (d), Illustration of the ten studied torque interactions, showing the kinesthetic stimuli (black arrows), contracted muscles (red highlighted muscle shapes), and direction of arm motion (blue arrows).

noticed that the arm motion showed a staircase trend: the person felt the JTF at first, and moved their arm. Then, they paused for a moment, since while moving they could not sense the stimulus. The person waited until they could sense the torque, then moved again. This effect suggests that arm speed affects, and masks, the sensation of torque at a joint. Similarly, other researchers conducted perceptual studies of the whole arm with a haptic stylus [18, 70, 72]. The researchers found that the force feedback magnitude and arm speed influenced the JND, and the observed behavior seemed to have a slope following Weber's law. In psychophysics, our sensory organs are argued to follow Weber's law where the ratio of a detectable stimulus change to the initial stimulus intensity is a constant. Thus, each joint state (preload torque and joint speed) will have a different JND torque.

Several researchers have studied a subset of external conditions affecting the torque perception of the arm. Zadeh et al. studied the JND of kinesthetic stimuli that influenced the whole arm, with a hand-held stylus display [72] and with the arm in motion. Feyzabadi et al. also used a stylus-based haptic device to study the force perception of the wrist, elbow, and shoulder joints. For the elbow specifically, Feyzabadi et al.'s experiment examined flexional motion ranging from $20°$-$80°$ with a preload in the flexional direction. Similarly, Thomas et al. employed JTF for replacing the tactile sensation of amputees [65]. Cross-modal matching was conducted between the vibrotactile and the joint torque feedback. The torque was applied in the flexional direction, with the arm moving in free motion. These studies are useful yet do not fully address all of the conditions that would arise during motion training, i.e., different preloads and different speeds with the JTF in various directions. Furthermore, measuring the perception ability of a proximal joint (i.e., the elbow) with a distal manipulator (i.e., a hand-held stylus) could yield different results than observations through direct joint torques. For example, the perceptual resolution at the elbow using a hand-held device [18] could be affected by the neural receptors in the whole arm's afferent pathway. Also, these studies with a stylus did not isolate the proprioceptive sensors: there is the chance that skin receptors in the hand holding the stylus could sense forces through pressures, skin stretch, or other feedback signals, giving additional information to the test subject beyond that from the proprioceptors.

In this paper, we examine ten different test conditions to completely characterize the arm's response to both stationary (static) and in-motion (dynamic) conditions (Fig. 3.1 (d) and Table 3.1). The test conditions involve arm muscles' isotonic and isometric contraction representing dynamic and static joint conditions. All of the static experiments were conducted at the arm's neutral condition ($45°$). All test results were analyzed to derive the participants' temporal, spatial, and directional perceptions. The static test condition involves $0.89$ Nm and $1.28$ Nm of preload.In the dynamic condition, the subject's arm moves at approximately $100°/\text{s}$ and $200°/\text{s}$, which is equivalent to the elbow moving through a range of around $80°$ at $0.5$ Hz and $1$ Hz. In designing the arm's movement,

the highest speed does not exceed 2 Hz based on Neilson et al. [17].

| | | |
|---|---|---|
| NTF | Neutral, Flexional torque | Exoskeleton applies a flexion torque while the arm is posed at neutral without muscle contraction or arm movement. |
| ECF | Extensor muscles Contracted, Flexional torque | Exoskeleton generates a flexion torque while the Extensor muscles are isometrically contracted. |
| FCF | Flexor muscles Contracted, Flexional torque | Exoskeleton generates a flexion torque while the Flexor muscles are isometrically contracted. |
| EMF | Extensional Movement, Flexional torque | Exoskeleton generates a flexion torque while the arm is moving in the extensional direction. |
| FMF | Flexional Movement, Flexional torque | Exoskeleton generates a flexion torque while the arm is moving in the flexional direction. |
| NTE | Neutral, Extensional torque | Exoskeleton generates an extension torque while the arm is posed without muscle contraction or arm movement. |
| ECE | Extensor muscles Contracted, Extensional torque | Exoskeleton generates an extension torque while the Extensor muscles are isometrically contracted. |
| FCE | Flexor muscles Contracted, Extensional torque | Exoskeleton generates an extension torque while the Flexor muscles are isometrically contracted. |
| EME | Extensional Movement, Extensional torque | Exoskeleton generates an extension torque while the arm is moving in the extensional direction. |
| FME | Flexional Movement, Extensional torque | Exoskeleton generates an extension torque while the arm is moving in the flexional direction. |

Table 3.1: Description of the different conditions examined in this paper.

## 3.2 Results

### *3.2.1 Static JND Torques*

For the static tests, a repeated measures ANOVA (N=14) was conducted on the JND torque, with the converged torque as the dependent variable and the preload level as a fixed effect. All of the $p$-values were corrected with the false discovery rate (FDR) method to account for multiple hypothesis tests. For the pairwise comparisons, a Tukey post hoc test was conducted.

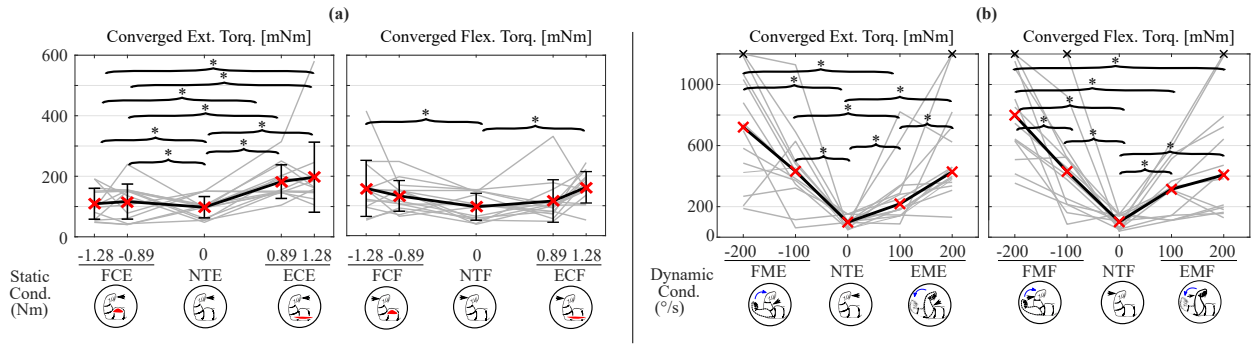


Figure 3.2: Summary of the test results. For the icons below the x-axis illustrating each experimental condition, please refer to Fig. 3.1(d). (a), The JND resulting from the static tests. All subject data is shown with gray lines, and the black line is the mean of the data. The data was analyzed with a Repeated Measures ANOVA and Tukey post-hoc test. Statistically significant pairs are indicated with asterisks. (b), The JND torque from dynamic tests, where the arm is in motion at different speeds. The black line is the median of the data. The black $\times$s indicate that the data exceeds the maximum device output, which is $1,200$ mNm. The data was analyzed with the Kruskal-Wallis test and followed by the Wilcoxon signed-rank test to compare pairs of conditions. Statistically significant pairs are indicated with asterisks.

The results for the static tests are shown in Fig. 3.2(a), and the $p$-values for the significant pairs are in Table 3.2. A summary of the means and standard deviations are listed in Table 3.3. Each of the interactions in Fig. 3.2 and Table 3.1 has two levels of fixed effects, except for the neutral conditions (NCF and NCE). The effect level is followed by the label. For example, ECF has $0.89$ Nm (ECF1) and $1.28$ Nm (ECF2) of preload torque pulling toward the flexional direction (that leads to the extensor muscles contracting). Pooling the static data, the fixed effect results were $F(4,52) = 3.91, p < 0.008$ for flexional direction torque and $F(4,52) = 8.05, p < 0.01$ for extensional direction torque.

The best-fit extension torque slopes for the static condition (i.e., the slope of the best fit line passing through the Neutral condition and the two levels of torque for a given condition) are $0.081$ mNm/mNm for the extensor muscle contracted (ECE: line through NTE, ECE1, ECE2) and $0.011$ mNm/mNm for the flexor muscle contracted (FCE). Similarly, the flexional torque slope was $0.046$ mNm/mNm and $0.047$ mNm/mNm for extensor (ECF) and flexor muscles contracted

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Tukey HSD Test | Flx. Torq. | NTF-ECF2 | | | | NTF-FCF2 | | |
| | t-test | 3.28 | | | | 3.10 | | |
| | $p$-value | 0.015 | | | | 0.025 | | |
| | Ext. Torq. | NTE-ECE2 | NTE-ECE1 | FCE2-ECE2 | FCE2-ECE1 | FCE1-ECE2 | ECE1-FCE1 | |
| | t-test | 4.36 | 3.72 | 3.86 | 3.21 | 3.54 | 2.90 | |
| | $p$-value | < 0.001 | 0.004 | 0.003 | 0.018 | 0.007 | 0.042 | |
| Wilcoxon Signed Rank Test | Flx. Torq. | FMF2-NTF | FMF1-NTF | FMF2-EMF1 | FMF2-FMF1 | FMF2-EMF2 | NTF-EMF1 | NTF-EMF2 |
| | Z-score | 4.482 | 4.114 | 3.885 | 2.780 | 2.412 | -3.426 | -4.252 |
| | $p$-value | < 0.001 | < 0.001 | < 0.001 | 0.005 | 0.016 | < 0.001 | < 0.001 |
| | Ext. Torq. | NTE-FME2 | NTE-FME1 | EME1-FME2 | EME1-FME1 | EME2-EME1 | EME1-NTE | EME2-NTE |
| | Z-score | 4.481 | 3.884 | 3.194 | 2.183 | -2.827 | -3.609 | -4.343 |
| | $p$-value | < 0.001 | < 0.001 | 0.001 | 0.029 | 0.005 | < 0.001 | < 0.001 |

Table 3.2: Statistical significance values for static and dynamic testing. Shown are the values from Tukey HSD Analysis (for the static tests) and the Wilcoxon Signed Rank test (for the dynamic tests) for each pair of conditions that was statistically different. (Flx. = flexional; Ext. = extensional; Torq. = torque.)

| | FC2 | FC1 | NT | EC1 | EC2 |
|---|---|---|---|---|---|
| Ext. Torq. [mNm] | $109.2 \pm 51.0$ | $116.3 \pm 57.9$ | $97.8 \pm 35.2$ | $182.3 \pm 55.5$ | $197.0 \pm 115.8$ |
| Flx. Torq. [mNm] | $158.0 \pm 93.7$ | $132.7 \pm 51.3$ | $96.5 \pm 45.0$ | $115.7 \pm 71.0$ | $161.5 \pm 52.5$ |

Table 3.3: Unpooled means $\pm$ standard deviations of the static JND torque. The first row (Ext. Torq.) corresponds to the FCE2, FCE1, NTE, ECE1, ECE2 trials, while the second row (Flx. Torq.) corresponds to the FCF2, FCF1, NTF, ECF1, ECF2 trials. (Flx. = flexional; Ext. = extensional; Torq. = torque).

(FCF), respectively.

All of the static conditions had response times of $0.445 \pm 0.108$ s, and these were not significantly different from each other.

### *3.2.2 Dynamic JND Torques*

During the dynamic condition, subjects moved their arms through a mean $\pm$ Std.Dev. range of $76.4° \pm 7.5°$. At the midpoint of the cycle, the mean absolute arm velocity was $97.1°/s \pm 23.2°/s$ for the $0.5$ Hz conditions and $213.1°/s \pm 23.5°/s$ for the $1$ Hz conditions. The entire cycle took $2.06 \pm 0.01$ s for the $0.5$ Hz conditions and $1.04 \pm 0.04$ s for the $1$ Hz conditions.

The results for the dynamic JND tests are shown in Fig. 3.2(b), and the $p$-values for the significant pairs are in Table 3.2. The dynamic measurements involved some trials where the converged JND torque values were beyond the hardware limit ($1,200$ mNm); these are shown in Fig. 3.2(b) with "$\times$" symbols. To handle the outlier trials, analysis was conducted based on ranks: non-parametric tests were used, specifically the Kruskal-Wallis test followed by a Wilcoxon signed-rank test to find significant pairs. The arm's movement speed was a fixed effect. In the dynamic conditions, there were two measured speeds, which were two levels of the fixed effect. For example, the EME has two levels of extensional movement speed: $0.5$ Hz (EME1) and $1$ Hz (EME2). Note that FMF and EME are conditions with the JTF aiding the arm movement, while EMF and FME have JTF resisting the arm motion.

A one-way chi-square test of independence was used to examine the relationship between arm movement speed and the ability to perceive the JTF in the flexional direction and extensional direction (N=14). The alpha level was 0.05 in the Kruskal-Wallis test. The relation between the flexion JTF and the arm speed effect was significant, $X^2$(4, $N$=14) = $39.13$, $p < 0.001$. There was also a significant relationship observed between the extensional JTF and arm speed variables. $X^2$(4,$N$=14) = $38.01$, $p < 0.001$. Significant pairs of conditions are shown in Fig. 3.2(b), and the medians of the data are given in Table 3.4.

| | FM2 | FM1 | NT | EM1 | EM2 |
|---|---|---|---|---|---|
| Ext. Torq. [mNm] | 720.7 | 431.7 | 98.3 | 218.7 | 428.3 |
| Flx. Torq. [mNm] | 798.9 | 430.3 | 100.5 | 313.8 | 408.0 |

Table 3.4: Medians of the dynamic JND condition. The first row (Ext. Torq.) corresponds to the FME2, FME1, NTE, EME1, EME2 trials, while the second row (Flx. Torq.) corresponds to the FMF2, FMF1, NTF, EMF1, EMF2 trials. (Flx. = flexional; Ext. = extensional; Torq. = torque.)

The best-fit extension torque slopes for the dynamic condition are $1.650$ mNm/(°/s) for the extensional (EME) and $3.112$ mNm/(°/s) for the flexional motion (FME), calculated the same way as in the static tests. Similarly, the flexional torque slope was $3.492$ mNm/(°/s) for the flexional (FMF)

and $1.537$ mNm/(°/s) for the extensional (EMF) arm movement.

In the dynamic condition, all of the trials except the fastest ones (FM2 and EM2) had response times of $0.431 \pm 0.107$ s (very similar to the static conditions), while the fastest ones had response times of $0.244 \pm 0.064$ s. This difference was due to the experimental setup, where the subjects were asked to answer within the arm's movement. With the faster arm motion, there was a shorter time window in which subjects could respond.

## 3.3 Discussion

Overall, the elbow torques required to be noticeable ($0.1 - 0.8$ Nm under various conditions) are 100-200 times smaller than the maximum isometric torques able to be produced by the elbow ($49 - 72$ Nm) [12, 15, 25]. The results from this study show that the perception of torque feedback is affected by various joint conditions.

### *3.3.1 Static Conditions*

The masking effect due to a static preload (Fig. 3.2(a)) resulted in a similar pattern with the previous test that was conducted with $0.64$ Nm and $1.28$ Nm [34], but with smaller JND values in all condition levels as compared to the prior experiment. Feyzabadi et al. found a value of $0.012$ Nm as the average of testing the neutral conditions. This is smaller than our results for NTE and NTF which were both very close to $0.097$ Nm. Similarly to our previous test [34], the JND of extension torques was larger when the subject was contracting their tricep (ECE) as compared to their bicep (FCE). It is unknown why this asymmetry occurs.

Additionally, the observed torque slope (JND torque divided by preload torque) in the current experiment was smaller than that found in the prior experiment. For the extension torque, it was

0.081 mNm/mNm for ECE (previously 0.118 mNm/mNm), and 0.011 mNm/mNm for the FCE (previously 0.026 mNm/mNm). The flexional torque slope was 0.046 mNm/mNm for ECF (previously 0.0692 mNm/mNm) and 0.047 mNm/mNm for FCF (previously 0.0548 mNm/mNm). These differences are likely due to increased amounts of training in this experiment (we examined additional conditions, as compared to previously when only the static cases were tested) or due to slightly different parameters in the psychophysics tool that allowed the tests to converge faster. Additionally, in this experiment the direction that converged first could keep converging, whereas previously it stopped after a fixed number of transition points.

### *3.3.2 Dynamic Conditions*

The dynamic conditions resulted in comparatively large values for JND torque (up to 1.2 Nm for some participants, with medians up to around 800 mNm for the fastest speeds), so motion does appear to cause a masking effect. For comparison, the maximum JND torque during the static conditions was only 200 mNm. Also, the dynamic conditions had a relatively large variance as compared to the static conditions (the largest observed JND torques were around 1 Nm more than the smallest JND torques). This may have been due to the relative difficulty of the experiment: participants needed to maintain a constant arm speed and track visual cues while responding to torque cues. Some participants may have been better able to focus on the torque cues while still maintaining the correct arm speed. The dynamic JND torques do increase roughly linearly with the arm speed. Thus, Weber's law appears to be in effect, similar to previous results [18, 72].

The torques, and torque slopes, in the dynamic tests were similar for a given arm motion direction, regardless of the direction of applied torque (Fig. 3.2(b)). That is, if the elbow was moving in flexion, the minimum perceivable torques were very similar for a given speed regardless if the applied torque was pushing in flexion or extension; the same was true if the elbow was moving

in extension. Also, for a given speed, the JND torques while the elbow was flexing were close to double those while the elbow was extending. Although not all pairs of fixed effect levels had significance, FME1 was larger than EME1, and FMF2 was larger than EMF2.

It is unclear why this asymmetry occurs. When the torques are applied, the arm is moving at nearly a constant velocity, and the velocity is the same in both directions. Thus, there is not a clear dominance of the flexors or extensors. It is possible that if the arm is flexing, then the biceps are more strongly contracted and if the arm is extending, then the triceps are more strongly contracted. However, this would result in an opposite pattern to that observed with the static conditions (FCE vs. ECE), where the bicep contracting had a smaller JND torque than when the tricep was contracting. While the flexors tend to create higher MVC forces [8, 37], the extensors have a larger cross-sectional area [37]. A larger cross-sectional area of the extensor with the corresponding larger MS population might be responsible for the lower medians in extensional motion.

Another possibility is that the pressure around the arm from the exoskeleton cuff might affect the results during elbow flexion. To mask the skin pressure, we applied distributed pressure around the arm. This pressure increased during arm flexion due to the expanded volume of the contracted bicep muscle (Fig. 3.1). Though we did not measure how much the FSR readings increased during full arm flexion, the escalation of the skin pressure might mask proprioceptive sensations even more, leading to the FME and FMF conditions being relatively insensitive.

The change rate of the perceived torque as a function of speed is much steeper than results from Zadeh et al. [72]. They used a Phantom haptic device that the user held in the fingers and moved back and forth laterally while a force either opposed or aided their motion (the force switched direction as the hand switched direction to maintain opposing or aiding). Zadeh et al.'s test yielded a JND of $222.12\ \mathrm{mN/(m/s)}$ for forces opposed to the movement, and $186.6\ \mathrm{mN/(m/s)}$ for aiding forces. To compare our results, we can convert our torque and speed based on an average length between

the elbow joint to the center of the hand. Considering this moment arm to be $32.7$ cm according to anthropometric data [11] results in torque slopes of $884$ mN/(m/s) for EME, $1668$ mN/(m/s) for FME, $1871$ mN/(m/s) for FMF, and $824$ mN/(m/s) for EMF. In the opposing direction (FME and EMF in our experiment), the smallest JND is $824$ mN/(m/s) (EMF), and the smallest JND in the aiding direction (FMF and EME) is $884$ mN/(m/s) (EME). Thus, our experiment found values that are around four times larger than those in the Zadeh et al. experiment.

These differences could be due to several factors. Since the Zadeh et al. experiment used a Phantom haptic device held in the fingers, there are many more proprioceptors that could have been stimulated. Not only could mechanotactile and vibrotactile sensors on the fingertips be used to sense forces, but other proprioceptors along the wrist and hand could have provided additional information. Second, the input feedback was triggered when the averaged arm speed (over the entire back-and-forth cycle or multiple cycles) was at certain values. The force reversed directions at the ends of travel, and it is possible that the user could have sensed those transitions in some circumstances. Finally, the speed level investigated in the experiment was much slower than our experiment. The speed levels we chose are nearly two and four times larger than their experiment's high velocity ($0.22-0.28$ m/s). For both Zadeh et al.'s experiment and our test, slower speeds yield better perceptual resolution. Our higher arm speeds may have led to larger torque slopes through some nonlinear mechanism.

### *3.3.3 Static vs. Dynamic Conditions*

Comparing the JND values for the static versus dynamic conditions, the converged torque values under the dynamic conditions (medians ranging from $218.7$ Nm to $798.9$ Nm) were substantially larger than the values with the static conditions ($96.5$ Nm to $197.0$ Nm, including the neutral conditions) (see Fig. 3.2 and Tables 3.3 and 3.4).

Recall that GTO are traditionally considered to measure muscle force while MS measure position and velocity of a joint [51]. We thus consider which sensors are being used to detect the joint torques in this experiment. One possibility is that the GTO are being used to detect the force change in the muscles. In most of the static conditions, the muscles are flexed due to the preloads on the arm. Since GTO measure the muscle tension, and the muscle tension changes slightly due to the torque from the exoskeleton, the GTO could be detecting the exoskeleton's effects. However, we note that in the dynamic conditions, the thresholds for noticing the applied torque were much higher than the thresholds for the static conditions (more than four times higher for fast velocities). We then must consider the amount of muscle activity in the biceps and triceps during the dynamic conditions. At the ends of the joint travel, there will be substantial muscle activity due to accelerating and decelerating the arm's inertia. However, in the center of the range of travel, the arm is moving at approximately steady state (Fig. 3.5(b)). While we did not measure muscle activity during the experiment, we can roughly surmise the muscle activity in the biceps and triceps. By taking the second derivative of the measured elbow angle and assuming a nominal moment of inertia for the forearm and hand, we find that the elbow torques even in the center of the range of motion are frequently 1-3 Nm, which must be created by the biceps and triceps muscles. Additionally, the muscles surely have some co-contraction due to needing to maintain control and follow the desired trajectory. Thus, there can be muscle contraction in the dynamic conditions that is even larger than that tested in the static conditions. Even if the GTO were the sole arm sensor being used to measure the torque pulses from the exoskeleton, and the GTO are not sensitive to arm motion (as is commonly understood), then the additional muscle activation in the arm could account for much of the difference between the static and dynamic conditions.

An alternative hypothesis is that some combination of the MS and GTO are contributing to the sensation of torque from the exoskeleton. Since the MS primarily respond to length and velocity of the muscle fibers, they could be stimulated if the exoskeleton caused small changes in the arm's

position which were then detected and used to determine that the exoskeleton caused torque on the arm. One study [28] found that people could correctly sense the direction of motion if their elbow was displaced only $0.1 - 0.2°$ at angular velocities of greater than $1°/s$. In our experiment, during the static condition torque pulses the exoskeleton itself moved at a maximum speed of $1 - 2°/s$ with displacements of $1 - 2°$ during many of the pulses. While the exoskeleton's motion is not exactly the same as the arm's motion due to the intervening padding, it is likely that the arm moved more than 0.1-0.2° and thus could be triggering the MS with the exoskeleton's torque pulses. During the dynamic conditions, the arm was moving at close to $97°/s$ or $213°/s$ in the different conditions. Due to the arm's motion, it is extremely difficult to determine if any additional motion was caused by the exoskeleton. However, since the torques applied during the dynamic conditions were generally larger than those applied during the static conditions, it is likely that some perceptible arm motion occurred as well.

If the MS are contributing to the sensation of torque, then their contribution could be affected by the arm's velocity. Assuming that Weber's law holds, neural signals due to the arm's motion could be thus masking the additional length changes due to the exoskeleton torque. Indeed, a recent model of the arm's proprioceptive abilities supposes that the MS and GTO combine their signals [36], and that the overall neural stimulus is a combination of the force and velocity.

An alternate reason why the dynamic JND is much larger than the static JND could be thixotropy [51]. Briefly, the response of the MS is affected by the muscle's activation immediately preceding a test condition. A conditioning contraction will increase the background firing rate of the MS [23, 68], which would imply that a larger stimulus would be required to produce a noticable response. In the dynamic tests, the arm muscles create large torques (up to 8-10 Nm in the faster condition) when reversing the arm's direction of motion. These large torques could mask any sensations in the center of the arm's range of motion where the torques were lower, if the MS were involved in sensing the exoskeleton torque.

A final reason for the difference between the dynamic and static conditions could be possible psychological distractions during the test itself. The static condition experiment was comparatively much simpler than the dynamic condition, where people needed to first maintain a steady arm motion and then secondly detect disturbance torques. Thus, a larger stimulus might be needed to gain their notice. However, this is unlikely to explain all of the differences observed.

### *3.3.4 Summary and Future Work*

We have quantified the perceptual ability of the elbow to sense torques via a psychophysics tool. However, the accuracy of the converged values could be improved. The step size we implemented has a limitation as it is pre-defined as the exponential function. If subjects missed initial trials, it took a long time for the torque magnitude to get back to the perceivable range, which decreased resolution. The step difference could be adjusted to an individual's learning speed using various adaptive algorithms in the future. Moreover, measuring electromyography (EMG) concurrently with torque inputs could better compare the arm's muscle activity between dynamic and static conditions. EMG sensors could clearly show the muscle activity during the dynamic conditions, and better be able to determine if the increased JND values were due solely to increased muscle activity or if the MS must be involved. Furthermore, choosing subject groups with different age groups or strength-training proficiency will be another helpful reference for simulating real-life training scenarios because they have different muscle properties and proprioceptors. Younger people tend to have stronger MVC and larger cross-sectional muscle area [37], and may have different proprioceptive abilities [51].

## 3.4 Conclusions

In conclusion, this work provides a comprehensive examination of the minimum elbow torques required to be sensed under various conditions with the GTO and MS in the arm. The analysis suggests that the perception of elbow JTF is influenced by the arm's contraction condition and movement, and that the sensitivity of these proprioceptors is less sensitive than the sensitivity of sensors in the skin. In general, we observed a more significant masking effect from arm movement than from isometric contraction. This work is useful for motion training systems, exoskeleton design, and force-feedback systems.

## 3.5 Methods

### *3.5.1 Elbow Exoskeleton.*

The hardware setup was an evolution of that used in a prior study [33, 34], and is shown in Fig. 3.3. The exoskeleton's range of motion is mechanically designed to reach 120° when the arm is fully flexed. However, when it is worn with the foam, the range is reduced to 90-95° depending on the subject's arm volume. The chosen motor was Antigravity MN7005 KV115 (24N 28P). To provide more torque to the arm for the dynamic condition experiment, the maximum current magnitude was increased to $14$ A. The torque in the system was calibrated using a FUTEK LBB200 load cell with a Tacuna Systems Amplifier v.2.3. The torque constant was found to be $0.072$ Nm/A.

As briefly described in the prior study [34], we calculated how much pressure was required to mask the skin receptors based on measurements [60]. Tan et al. [60] explained that the human tactile system is sensitive to the perimeter of the contact between the arm and another object. The JND force threshold per contact perimeter was found to be $0.06-0.09$ N/cm [60]. The effective surfaces of the exoskeleton are the two plastic cuffs (Fig. 3.1(b)), of which the outer edge is $34.5$ cm for each side and the surface area is $414$ cm$^2$. When choosing $0.09$ N/cm for the force threshold at the

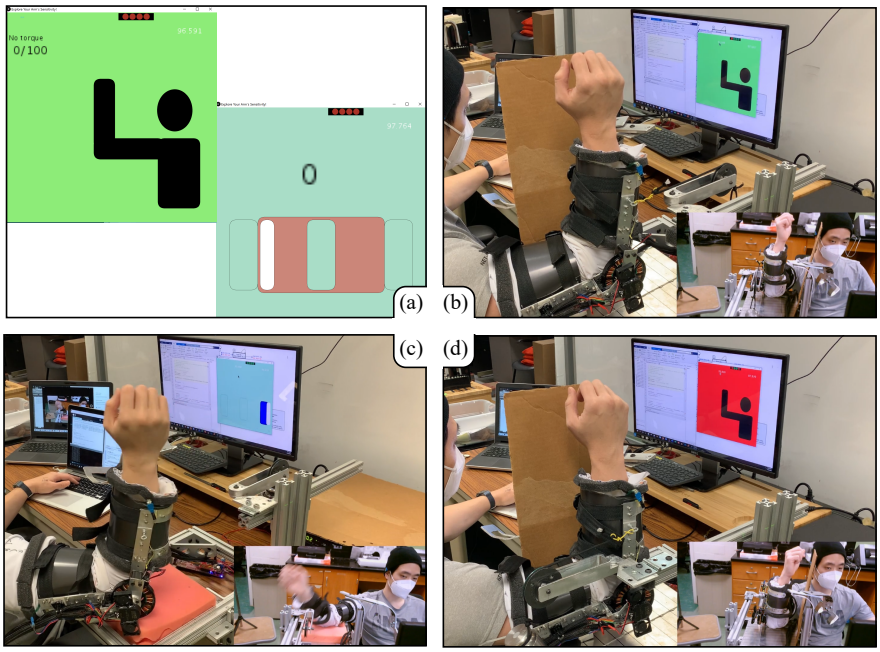


Figure 3.3: (a) GUI screens for the static (left) and dynamic (right) mini-games. The score for the mini-game provides participants whether the current user's response is correct or not. Additionally, the red block of the dynamic condition is only shown during the demonstration to supplement the ideal timing window for response. The white box represents the arm position that moves along three boxes that correspond the arm's range of motion. From (b) to (d) are examples of testing of the static (flexor muscle contracted), dynamic, and static (extensor muscle contracted) conditions, respectively.

perimeter, the minimum pressure to mask the tactile sensors becomes $0.015\ \mathrm{N/cm^2}$. After donning the exoskeleton and tightening the straps, four FSR sensors (size: $4.5$ by $4.5$ cm) are measured to make sure that each sensor reading is more than $0.015\ \mathrm{N/cm^2}$. The four FSR sensor readings are shown in the top right corner of the GUI screen as four red circles (Fig. 3.3(a)). Because we wrapped the arm with foam, the actual pressure required to be noticeable is assumed to be larger than the value calculated from the periphery.

*3.5.2 Software Systems.*

The experiment was programmed in three different platforms: Processing, Texas Instruments Code Composer Studio (CCS), and MATLAB. Processing was utilized to make a Graphical User Interface (GUI) to give cues to the participants (Fig. 3.3); the subjects responded to perceived joint torques through a computer keyboard. Processing was also used to take the user's responses and calculate the magnitude of the upcoming applied torque via a psychophysics tool. The calculated torque was then transmitted back to a Texas Instrument(TI) C2000 TMS320F28069M microcontroller (programmed in CCS) that commanded the TI DRV8305 motor shield to apply the torque to the arm. When the current was applied, to mitigate abrupt current surges that might trigger the wearer's vibrotactile sensation, we apply a first-order Low Pass Filter with a rise time of 0.5 seconds. Finally, MATLAB took the logged data from Processing and summarized the user's perceptual abilities. The microcontroller transmitted 24 byte data packages at 200 Hz, while the UI platform logged the data at 200 Hz and refreshed the GUI at 100 Hz.

*Psychophysics Tools.*

A psychophysics tool similar to that used in Kim et al. [34], the interweaving Staircase Method, was used to assess perception capability for both static and dynamic conditions. The Staircase Method is a recurring algorithm that updates the stimulus reflecting the user's response (Fig. 3.4). It updates like a staircase; while the input alternates its direction randomly, the consecutive magnitude of the input increases or decreases based on the user's response [18, 72].

In the Static experiment, the state machine (Fig. 3.4(a)) constitutes two states: the Apply state and the Calculate state. In the Apply state, the torque is provided to the user's arm, and they make a response. The torque is set to zero in the Calculate state, and the state machine is updated to prepare for the next torque application. When torque is applied, the user's response could be: (1) torque felt in the flexional direction, (2) torque felt in the extensional direction, or (3) did

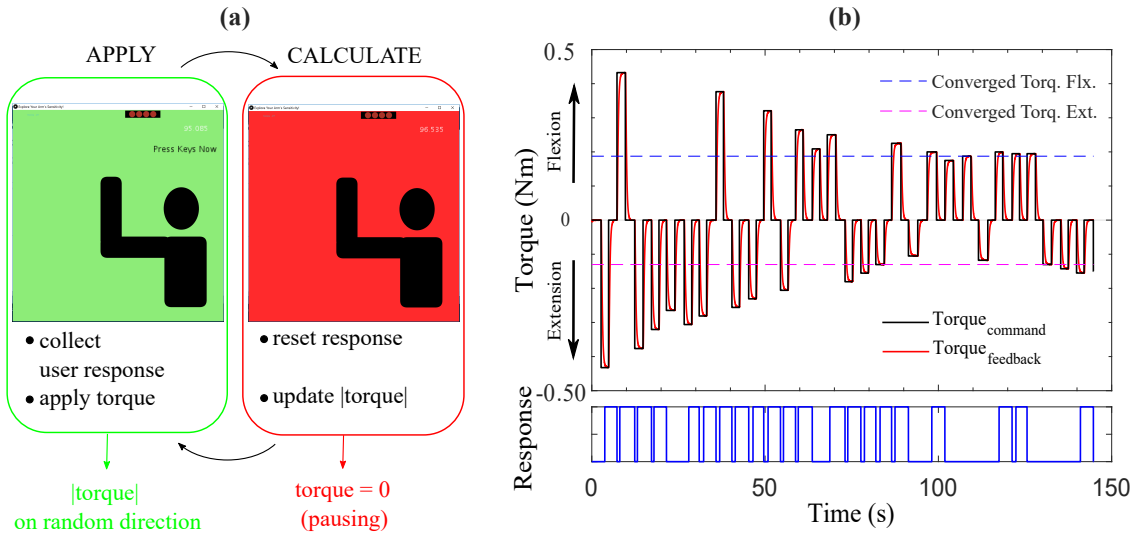


Figure 3.4: (a), Schematic diagram of state machine for static condition. (b), Example of a static JND test result.

not feel any torque. A series of answers of "YES-torque felt" are responsible for decreasing the magnitude until the answer switches to "NO-did not feel torque" and vice versa. Whenever the trend of magnitude change switches from increasing to decreasing and vice versa, the transition is counted, and the step size of the magnitude is reduced to yield a high-resolution converging value. The step difference is calculated with the exponential decay $StepSize[n] = Ae^{-0.1(n-1)}$, where $n$ is the transition point number (starting at $n = 1$), and $A$ is a fixed scalar of $A = 100$ mNm for all conditions. The exponential function is tuned such that the initial adjustment can be done with the large step size and start decreasing as the transition happens. Thus, the Step Difference does not change until the transition happens. For example, when there were a series of YES responses, the system subtracted the same magnitude of Step Difference from the original torque value until the subject answered NO. When the answer switches from YES to NO, then the step difference's size was changed. However, the initial magnitude of the torque is set considering the difficulty of the task. We found that the subjects had difficulty responding whenever the required arm movement is the same as the direction of applied torque from the pilot test. Therefore, the initial values for those conditions (FMF, EME) were set to 900 mNm, while other dynamic conditions have 500 mNm.

The initial values for the static condition were $400$ mNm. Even though more transition points directly contribute to a higher resolution of the converged value, the experiment needs to balance the resolution and fatigue from wearing the exoskeleton. Therefore, the static experiment continued until both the flexion and extension directions had at least eight transitions. Typically one of the directions converged before the other; the direction that converged first continued updating until the other direction reached eight transitions. At the end of the test, the mean of the final two transition points was recorded as the converged value. In the Pausing state, the experiment could also be stopped by the experimenter for an extended period of time. This was used whenever the participant's arm strayed away from the initial neutral position too much (i.e., the forearm was more than around 30° away from vertical). If the experiment was stopped, the experimenter told the participant how to move their arm, so it was once again close to the neutral position, and then the experiment was resumed. The effect levels for the static condition are slightly different from our prior test [34], specifically the smaller amount of preload was increased from $0.64$ Nm to $0.89$ Nm. The design change was made to anticipate more significant changes in user response with the higher preload.

*Psychophysics Tool for Dynamic JND Testing.*

The psychophysics tool for dynamic testing shares many properties with the staircase method with the static condition, including the same exponential function to calculate the step difference. However, the dynamic JND experiment required additional design considerations because the participants need to maintain the required arm speed while reacting to the psychophysical input. The first design consideration was how to discretize the arm movement. We pilot tested applying torque during the whole arm cycle, as did prior researchers [72]; in this case, the subjects were asked to move the arms back and forth with the same average speed, and the torque magnitude and direction did not change during one complete cycle of extension and flexion. However, we noticed that

users' answers were mainly acquired when their arms were changing direction. At these moments, the velocity was very slow and not distinctly in either flexion or extension. We speculated that the users were feeling the torque in a condition that was essentially static. So, the final system only applied torque in the center of each arm's movement while the arm had substantial velocity.

Fig. 3.5 gives an overview of the GUI, arm motion, state machine, and example data for the dynamic conditions. To guide the participants' motion, a GUI was presented on a monitor. Fig. 3.5 (a) shows a series of green rectangles which are examples of the GUI at different points in the arm's motion. Within each green rectangle, there are three blocks (outlines of rectangles) that correspond to the limits of the arm's range of motion and the center of the range of motion. These are labeled ① - ③ at the very bottom of the sequence of green rectangles. Each block changes its color to blue sequentially to guide the wearer to move to that arm angle. A moving white bar displays the wearer's current arm position. The border of each block becomes thicker when the arm position hits the block, indicating the arm reached different points in the range of motion. Additionally, to cue the user, the moving white block turns red when the subject presses the keyboard. This process ensured that the subjects moved through the full range of motion. The approximate arm angles swept by the participant are shown in Fig. 3.5(b). The gray shaded regions and labels ① - ③ correspond to the regions and labels indicated at the bottom of Fig. 3.5(a).

The state machine, Fig. 3.5(c), updates and generates the torque signal based on the participant's past reply during one motion direction (i.e., flexion or extension). The state machine calculates the stimulus based on the real-time arm position. Whenever the subject's arm moved the wrong way or missed the end block, the state machine detected it and paused the internal update such that the user's response was not recorded and no input magnitude change was made. When subjects respond, because two options (Yes or No) are given instead of three (flexion, extension, No), the system is equipped with a random-on-off system to mitigate subjects guessing the incoming stimuli. Fig. 3.5(d) shows how the random decision on each extension or flexion affects the torque

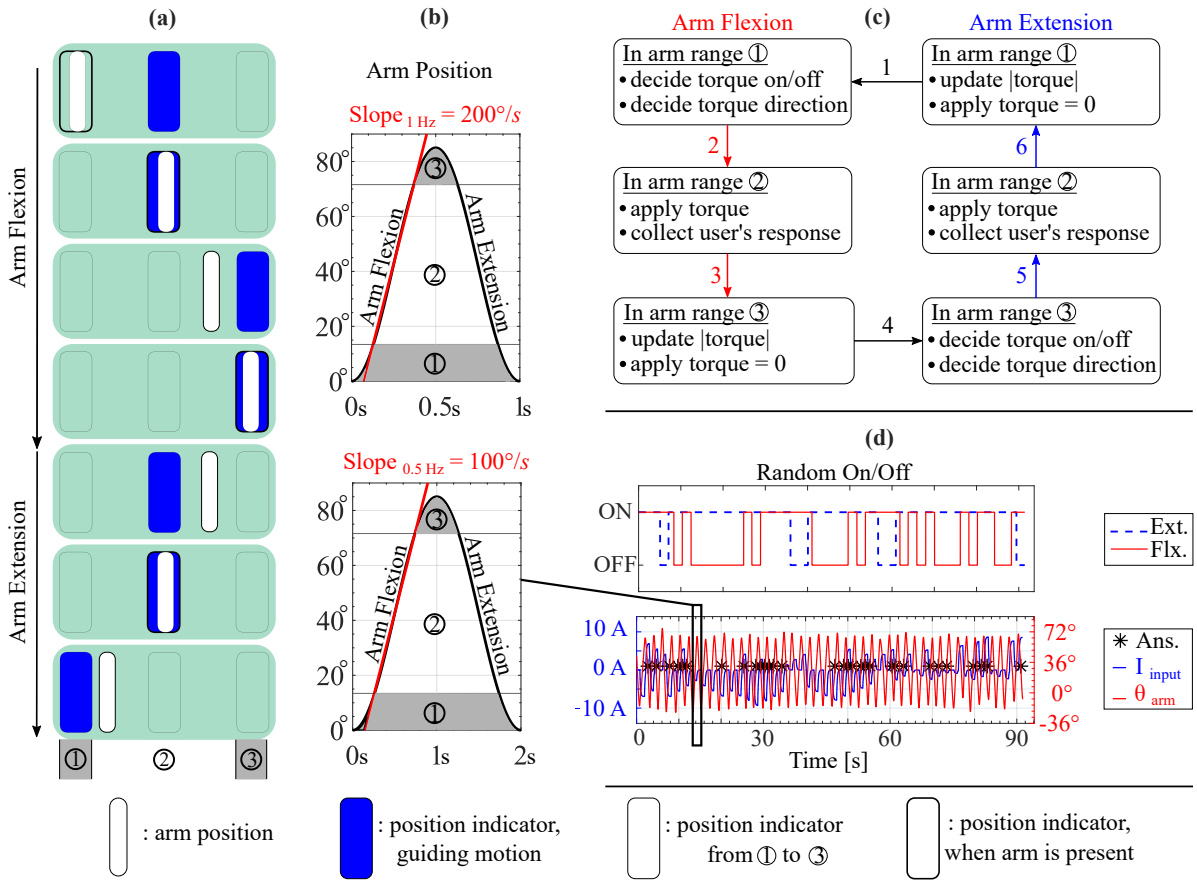


Figure 3.5: (a), GUI system guiding the arm movement. Participants are instructed to move their arm (white bar) to match the guiding block (blue block). The horizontal position of the arm position corresponds to the actual elbow joint angle. (b), Illustration of the commanded arm angular position versus time for the fast (1 Hz, top) and slow (0.5 Hz, bottom) velocities. The average speed over the center of the arm position range is shown with the red line. The gray shaded ranges of arm position correspond to the portions where the arm is changing speed (① and ③); the white regions are where the arm was moving at roughly a constant speed, and torque was applied (②). (c), State machine illustrating how torque is applied to the arm depending on the participant's current arm position and if their arm is moving in flexion or extension. State transitions 1 and 4 occur when the arm changes its direction of motion; the other state transitions occur based on the arm position. (d), Example results from a trial. The top graph illustrates how the random on/off works on each extensional/flexional torque input. The bottom graph shows the subject moving their arm back and forth (red line) and the subject reacting to the current input (blue line). The black asterisks show the moments when the subject responded.

generation.

The second design problem in the dynamic conditions was the means to effectively guide the par-

ticipant into moving their forearm back and forth at the right velocity. In pilot testing, we tested solely using a visual cue to guide arm motion, similar to the study from [72]. However, the subjects were easily distracted by the arm motion directions and could not focus on the psychophysics sensations. Therefore, we used multimodal cues (an auditory cue in addition to the GUI) to facilitate subjects adapting to the test environment faster. We first attempted to use a continuous sound where the pitch was synchronized with the arm's motion, but this was not helpful. Even with a small deviation from the desired movement, the sound changes were confusing to the subjects. We then sectioned the arm's range of motion into five regions and assigned music notes of do-le-mi-fa-sol to each region using the Processing 3.0 Minim library. The programmed sound played whenever the arm position was within the region. Even with this system, the motion was not as smooth as expected. Eventually, we decided to use two different beeping sounds played whenever the arm motion should change direction. The subjects were instructed to move their arm to hit the end blocks (Fig. 3.5(a)) concurrently with the beeping sounds. This synchronization prevented the torque from being applied during the wrong arm movement.

Comparing Fig. 3.5(b) and (d), the users nominally would move their arm through a range of $90°$. In practice, people changed their arm's motion direction as soon as they hit the end blocks (① and ③), without fully flexing or extending their arm. This led to a slightly reduced range of motion during the experiment.

### *3.5.3 Participants.*

We fully randomized the session orders for the static and dynamic conditions via a Standard Latin-square design to account for any fatigue effects and learning effects. The recruited sample size was $N = 14$ where 12 were male, 2 were female. The average age was $29.31 \pm 4.00$ years, and all were healthy right-handed individuals. The experiment was approved by the Virginia Tech In-

stitutional Review Board (IRB #16-175) and the Human Research Protection Program at Virginia Tech for the COVID-19 mitigation process, and all regulations and procedures were followed during the experiment. All the participants also followed the COVID-19 safety guidelines. Subjects provided informed consent and signed a consent form, then were instructed about the purpose of the experiment before the test. The entire experiment took 45-50 minutes, including a 5-minute break between the two sessions.

### *3.5.4 Experimental Protocol.*

All of the experiments began with calibration, discussed in the next section. The tests constituted a randomized combination of static and dynamic sessions. The static and dynamic sessions were made up of a practice session, followed by two test sessions with different effect levels. Thus, all the test sessions, including static and dynamic tests and fixed effect levels, were fully randomized. The practice session was intended to facilitate learning how to respond with the keyboard given a JTF input. Due to the difficulty of the experiment setup for the dynamic conditions, a practice session was conducted whenever the effect level changed. For instance, when the speed level was increased, another practice session was conducted immediately before the test. The practice session for the static condition was conducted only at the neutral arm posture, with no preload torques.

All the practice sessions were comprised of a mini-game. Participants were rewarded with 10 points for a correct answer and punished with -5 points for a wrong answer. Fig. 3.3(a) illustrates the GUI screen for the static (left) and the dynamic (right) mini-games. Upon responding, the correct answer is shown on the screen by updating the score. Subjects are required to score 100 to transition into the test sessions. The mini-game that took 3-4 minutes in our previous work [34] was extended to be 10 minutes long to ensure each participant got used to the system. After each session, to promote engagement, a brief summary of the participant's performance was provided to them. The

provided analysis included the smallest magnitude the subject could perceive in each direction and how quickly they responded.

Throughout the test, two extended screens were used for the experiment: one for the subject and another for the experimenter. Thus, the experimenter monitored the subject's converging dependency to determine if the subjects were correctly pressing the buttons or were distracted from the environment. If a subject was distracted, the experiment was paused and the experimenter reminded them to pay attention.

*Calibration.*

Subjects first adjusted the seat and height of the platform where their elbow was resting (labeled "stand" in Fig. 3.1(b)) until their upper arm was parallel to the ground. The exoskeleton was adjusted so the motor was aligned with the center of rotation of the elbow joint. Next, calibration of the strap pressure was conducted in the arm's neutral position (at a right angle), to account for the expanded volume of the arm flexors. During the donning process, the subject's arm was kept fully supinated. Straps were fastened until four FSR readings reached the minimum required values, which were indicated on the GUI as red circles on the top right screen (Fig. 3.3(a)). Due to the relatively high pressures from the exoskeleton cuffs, subjects were asked to provide feedback on whether it was too snug while the system was being demonstrated, and the strap pressure was adjusted accordingly. Lastly, subjects finalized the calibration process by resetting the index pin. As soon as the encoder detects the index pin during the arm swing, it triggers the timer interrupts in the microprocessor and prepares the motor controller.

Once the calibration is finished, the GUI indicator directs the instruction to conduct the Interweaving Staircase Method. The state machine indicates when each experiment terminates, then resumes once the conditions for the next experiment are ready, such as changing the weight used to create

the preload torque in the static conditions.

*JND for static condition.*

The preload torques were applied by a string pulling on the forearm of the exoskeleton. The string passed over a pulley, and a mass of $700$ g or $1,000$ g of mass was attached to the string on the far side of the pulley (Fig. 3.3(b) and (d)). This created a constant preload torques of $0.89$ Nm and $1.28$ Nm due to the string pulling on the exoskeleton forearm at a distance of $13$ cm from the elbow joint. On top of these preload torques, the exoskeleton motor provided test torques.

The GUI notifies the subject that a torque event occurs via a color change, the only visual data transferred. The green background on the screen (Fig. 3.3(b)) means the torque is being applied, and the subject should respond if they felt the torque feedback. The red screen (Fig. 3.3(d)) indicates that the torque is not applied and the subject should prepare for the upcoming torque input. While the red background is shown, any user response is not recorded. From pilot testing, we found that the recovery torque (i.e., torque returning to zero after a test stimulus is applied) behaves as a phantom stimulus in the opposite direction. The event cue (color change only when the stimulus is applied) keeps subjects from reacting to the restoring torque.

A panel obscured the exoskeleton from the participants' view to prevent them from responding by watching the arm movement (Fig. 3.3(b),(d)). The participants were instructed to not touch the panel, as the mechanoreceptors around fingers might respond more sensitively than the kinesthetic receptors. This better matches a typical motion training scenario that assumes a floating limb while correcting the motion. Test subjects are required to relax their right arms in an L-shape, not fully supinated nor pronated. Subjects cannot see their arm, so they can easily deviate from the neutral position when torques are applied. If this occurs, the experimenter pauses the state machine and provides verbal correction to be in the neutral position again.

*JND for dynamic joint condition.*

The dynamic condition sessions required knowledge that could be difficult to explain by simple verbal explanation. Therefore, a pre-recorded video of the instructor reacting to the GUI with the exoskeleton was prepared. It contained what the screen was showing, the expected response, and what feedback the system would give when the answers were given to the system. The video allowed the participants to learn the system faster. As a result, the total experiment time could be reduced, which could be critical when testing with wearable devices. Additionally, at the beginning of each practice session, the subjects were asked to move their arm while wearing the exoskeleton but without torque feedback. Torque was applied only after they had adapted to the visual and auditory interfaces and were able to generate the expected speed.

The test was separated into two sessions: torque in the opposing direction and torque in the aiding direction. When aiding, the torque was applied in the same direction as the arm's motion, and when opposing it was in the opposite direction. Consequently, the flexional torque of FMF and extensional torque of EME were grouped together as the aiding trials, while EMF and FME were applied as part of the same session as the opposing trials. The expected user response was either yes or no (felt or did not feel the torque), and each direction's torque was converged based on its own transition counter.

The dynamic condition had ten transition points to convergence. For the practice session of the dynamic test, we increased the number of transition points to fifteen so that the subjects could adapt to the exoskeleton and the GUI system before the test sessions. All the test combinations, including the static and dynamic, were fully randomized; some subjects had harder tasks to begin with than others. Specifically, those who began with the dynamic aiding conditions required additional time to learn the system.

### 3.6 Acknowledgements

This work was supported by a Virginia Tech ICTAS Doctoral Scholars Fellowship (H. Kim) and a Google Faculty Research Award.

### 3.7 Data Availability

All data generated or analyzed during this study are included as Supplementary Information.

### 3.8 Author contributions statement

H.K. and A.A. conceived the experiments, analyzed the results, and wrote and reviewed the manuscript. H.K. conducted the experiments.

### 3.9 Additional information

**Competing interests** The authors declare no competing interests.

# Chapter 4

# The Effects of Torq. Mag. and Stiffness in Arm Guidance through Joint Torque Feedback

**Authors:** *Hubert Kim*[1] *and Alan T. Asbeck*[1]

**Affiliations:** [1]Mechanical Engineering Department, Virginia Tech, Blacksburg, VA 24061, USA.

**Contact email:** aasbeck@vt.edu

**abstract** Joint torque feedback is a new and intuitive way of delivering kinesthetic feedback to a person or guiding them during a motion task via wearable devices. However, few works have been done when it comes to how stiff the torque input should be to guide a motion without interference with voluntary movement. Many studies have focused on the high-level control approach to improve the guiding experience. In comparison, the low-level control involves subject-dependent elements and thus being left unexplored. We study the low-level control mechanism on persons' arms that could be a reference for rehabilitation applications. This study adopted the widely accepted position error-based PD controller to evaluate the torque condition that leads to the minimum position error and time delay. The hardware setup is similar to our prior study, where the arm exoskeleton masks the touch and pressure sensation of the subject's arm while applying Joint Torque. We found that the highest torque stiffness is responsible for the smallest position error

despite the applied torque acting as compulsory stimuli. When the joint's final angle increases, the error also increases with a significant change over the gain effect. Furthermore, we examined how people react to the external current pulse when duration and magnitude change. Both duration and magnitude are shown to be significant on the arms' traveling distance.

## 4.1 Introduction

Haptic feedback has been a traditionally effective means for delivering training in movement sequences. The tactile feedback is beneficial for its design flexibility as the actuators are small and easy to construct; whereas, kinesthetic feedback has the advantage of intuitive feedback for the participants to follow through. Bark et al. fabricated a wearable robot made of four vibrotactile motors and explored one to four degrees of freedom arm motions [4]. Skin stretch was also tested for supplementing spatial perception [5] while mechanotactile replaces the proprioceptive perception for arm amputation [3]. In terms of kinesthetic feedback, Carignan et al. utilized a multi-joint system [10] of joint torque feedback (JTF) to demonstrate the virtual wall during a wall-painting task. Thomas et al. adopted JTF and vibrotactile to perceive the different surface shapes of objects [65]. In addition, researchers investigated kinesthetic feedback through a manipulandum display for training tennis motions [47] and rowing training [52].

JTF is haptic feedback delivered by a wearable device at a joint level: the device creates a small torque around a joint to guide the wearer's motion, similar to how a physical trainer might guide a person's limb during rehabilitation or motion training. JTF is distinctive from other tactile feedback because the stimulated sensory organs are the same proprioceptors used for body coordination. Proprioceptive responses can be accessed with two approaches, obtained and imposed Proprioception [29]. The obtained Proprioception is the body sensation realized by active movement, where the fusimotor action and joint receptors feedback are involved (Figure 4.1(a)). The conven-

tional experimental method asks the subject to move the limb to a certain position without seeing it [30, 39, 67]. In comparison, the imposed Proprioception occurs when the body segment is externally driven (Figure 4.1(b)). When the external force moves the limb, the skin receptors around the joint are known to be the principal source of measurement [20] and [73]. Experiments were conducted on the blindfolded subjects who notified the experimenter when detecting the passively-driven motion [7, 43, 71]. In the kinesthetic tracking test, subjects were instructed to move their right arms according to the angle change of the left elbow [17]. Thus, while the observations above are resourceful, physical guidance in motion training would accompany both active and passive Proprioception (Figure 4.1(c)).

JTF is also differentiated from the kinesthesis of the manipulandum devices. Gillespie et al. argued that the force interaction made in the end-effector is delivered to the trainers as the combination of the device's internal dynamics and desired behavior, and people might find it challenging to differentiate the desired two [21]. Even in the physiological perspective, proprioception felt via an end-effector takes place with multiple haptic sensory organs. For example, measuring force feedback on the arm's elbow through Phantom Omni device accompanies tactile sensors on the palm, joint receptors on the wrist, and elbow [18]. On the other hand, JTF typically uses a single direct-drive actuator per joint degree of freedom [10, 33, 34, 65].

There are numerous ways to guide an arm with JTF. Figure 4.2 illustrates the idea of torque stiffness in motion guidance (top) and simplified figures of the torque slope (bottom). The torque stiffness is from compliant (left) to rigid (right), and the steeper the torque slope becomes, the faster to reach the stall torque leading to the stronger corrective stimuli. Torque stiffness determines the rise time and torque threshold to get to the person's noticeable level. Different gain leads to the different threshold and rise time.

Three levels of torque stiffness are defined here:

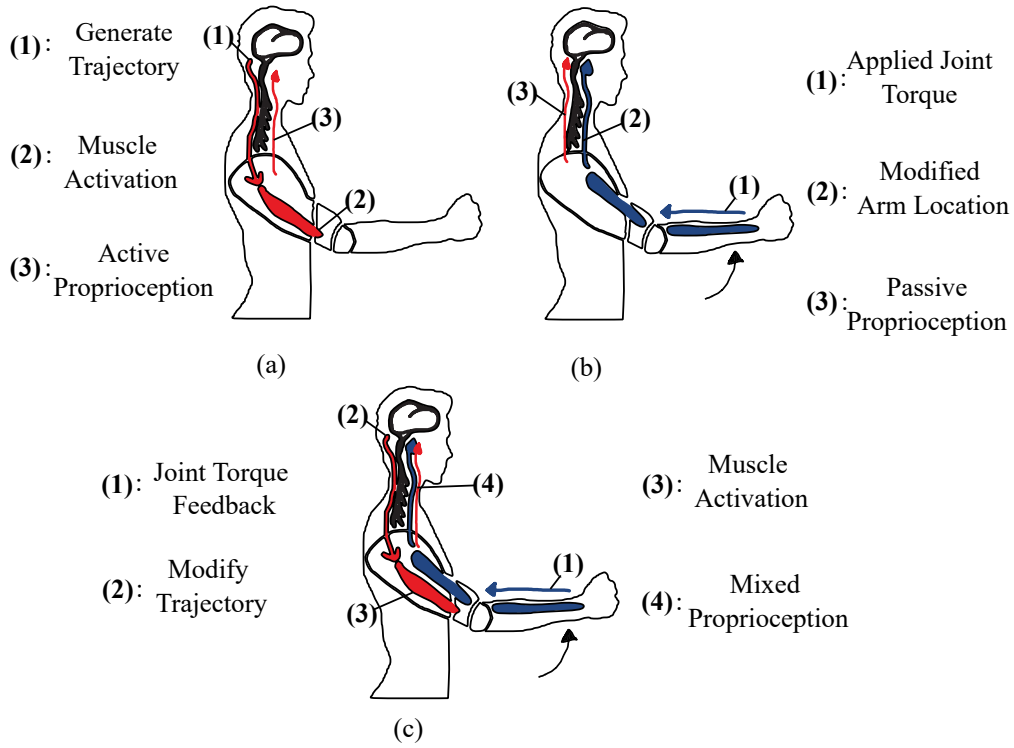


Figure 4.1: Conceptual diagram showing the different types of proprioception [29, 51] and its role in arm movement control [9]. (a) Obtained/Active proprioception. (b) Imposed/Passive proprioception. (c) Mixed proprioception.

(1) Torque Stiffness, JND, where the subject perceives a torque at a JND level. The slope for the torque controller becomes zero, and JTF works only as an event cue. It only tells the discrete information as the arm is following the trajectory correctly or not. This level of stiffness could be useful in the teleoperation industry, where compressing the haptic information plays a vital role [27].

(2) Torque Stiffness, Compliant, where the torque change over the position error is larger than the JND but does not disturb the arm motion. JTF works as both an event cue and spatial error. We believe that this level of torque controller is appropriate for a patient-driven control strategy [54].

(3) Torque Stiffness, Virtual Wall, where the initial perceivable sensation is a strong torque. Tan et al. [60] found that 24.2 N/mm was required to be felt like the rigid wall virtual wall when vision was blocked. Later, O'Malley and Goldfarb [49] further developed the virtual stiffness in a

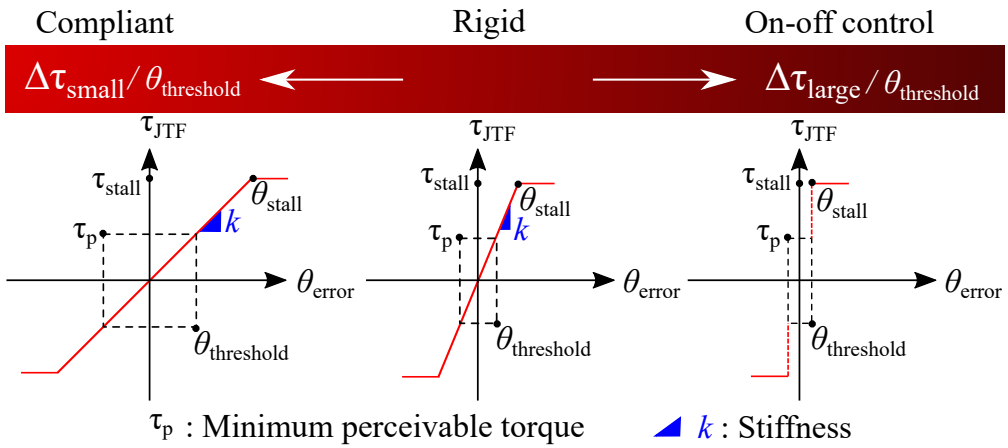


Figure 4.2: During arm guidance, an exoskeleton provides a joint torque $\tau_{JTF}$ as a function of the error $\theta_{error}$ between the desired and actual arm angle. This diagram presents several options for the function relating the error and applied torque. The torque stiffness, $\frac{\Delta\tau}{\theta_{error}}$, is small on the left and large on the right. The three stiffness levels involve: **Compliant JTF**- shallow torque slope, JTF provides cues but does not hinder the motion. **Rigid**- steep torque slope, JTF may constrain the wearer's arm. **On-off control**- vertical torque slope, JTF delivers a binary (on-off) cue. $\tau_{stall}$ is the torque limit due to the hardware limit; $\tau_p$ is the perceptual resolution; $\theta_{stall}$ is the position threshold until JTF becomes the $\tau_{stall}$; and $\theta_{threshold}$ is the position threshold until JTF becomes the $\tau_p$.

stylus-based and found that as small as 0.3-0.4 N/mm seemed to be the maximum distinguishable stiffness. In this level, $\theta_{stall}$ becomes almost $\theta_{threshold}$.

At first glance at the stiffness chart (Figure 4.2), it seems evident that applying a high gain would lead to better tracking. However, closer examination reveals that simply increasing the stiffness does not necessarily lead to better tracking performance. According to the guidance hypothesis, the corrective cues derived from a position error could be helpful yet should not be the primary source to be dependent on [69]. Excessive guidance without considering the human perceptual level could dominate the subject's perceptual ability, and there is little autonomy left for the wearer to play any role in tracking [55, 57, 69]. This argument is in agreement with the observations from [47] and [52] that the position controller yielded better tracking accuracy than the path control yet degraded the participants' self-initiation. Moreover, prior experiments have found that proprioceptors have a velocity-dependent or muscle contraction-dependent Weber fraction [18, 27, 34]. The stronger

arm is isometrically contracted, the inferior the perceptual resolution becomes, and the faster the arm moves, the lessened perception ability (Chapter 3).

Another element to notice in the toque stiffness is whether the JTF could move the arm and impede the motion or not. The torque required to move the relaxed arm is measured from 1 Nm/rad to 4 Nm/rad, depending on the joint conditions [41]. Though we cannot completely decouple the arm moved from JTF and the voluntary movement, we expect to observe that keeping the magnitude at the perceptual level could not hinder the arm movement. This observation is anticipated to be shown in the frequency-domain analysis.

Lastly, the JTF becomes an on-off controller with the fixed torque input when the slope becomes too large. For the JTF to be utilized as haptic guidance, it should deliver "when to move," and "how much travel" regardless of the wearer's joint condition [38]. The torque slope realized in the PD controller has a stall torque when it is out of the position threshold boundary, and the position required to reach the stall torque is represented as $\theta_{stall}$ in Figure 4.2. Simply increasing the gain would eventually lead to a discrete-like JTF, far-right slope in the figure where the $\theta_{threshold}$ becomes $\theta_{stall}$. Our pilot test found that the on-off controller made of JTF with a Just Noticeable Difference (JND) level could not provide sufficient information to the subjects. When the position threshold was set to narrow, the exoskeleton was unstable and constantly changed the direction of the torque. Even when there was enough position threshold, the subject could not tell how much one's arm should move.

Therefore, the magnitude of the JTF is suggested to be closer to the JND level, remaining as just enough amount of torque to be noticeable and not disturb the motion. Meanwhile, it should still provide spatial information corresponding to the Compliant zone from Figure 4.2.

In this paper, we study the JTF perception during tracking in three different approaches:
First, we examine how well people move their arms with different magnitudes and durations of

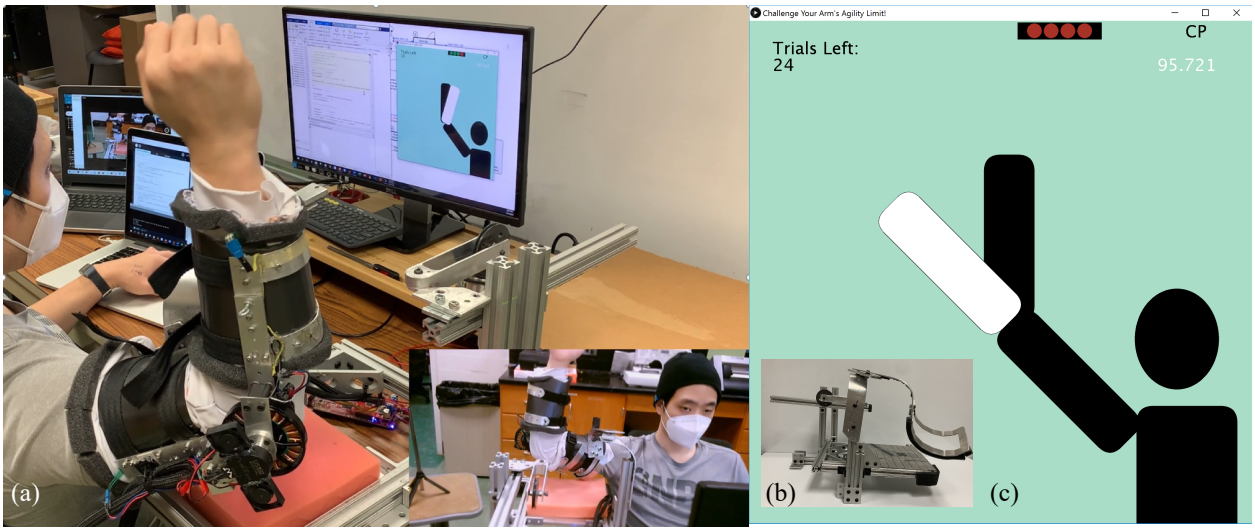


Figure 4.3: (a) A subject participating in the test. (b) The aluminum test setup for holding the arm 45° inclined. (c) A sample GUI for the demonstration. The white arm indicates the participant's arm position while the black arm guides the desired arm motion. The black arm is not present once the test begins.

torque from a Current Pulse (CP). A pure torque pulse is applied to determine if the torque's magnitude is more effective than the pulse duration. This also allows us to investigate if the previously measured JND is strong enough to guide the arm ([34]). Second, different levels of gain within the compliant range are studied with a Position Pulse (PP). The JTF implemented in this study is limited to about 1 Nm, which is not nearly as strong as constraining the arm to the target trajectory. Accordingly, the user's arm dynamics and perceptual ability are the two critical elements in determining the tracking performance. The stiffness responsible for the highest tracking performance could also guide a motion without interference with voluntary arm movement, promoting use-dependent plasticity. Third, the experiment explores how the joint's physiological condition affects the wearer's perception when tracking a Position Step (PS). From Latash et al.'s review, the human joint has inherent stiffness that varies depending on the angle ([41]). Thus, even the same magnitude of JTF may lead to a different final angle when the initial angle is different.

## 4.2 Materials and Methods

### *4.2.1 Materials*

#### *Hardware*

The same right-arm exoskeleton used for our prior study is made of an aluminum frame wrapped around a plastic cuff and actuated by a brushless DC (BLDC) motor (Antigravity MN7005 KV115, 24N 28P) [33, 34]. To operate the motor, a Texas Instruments (TI) C2000 TMS320F28069M microcontroller with the TI DRV8305 motor shield was used. The maximum operating current limit is set to $14.4$ A, which is nearly the hardware's maximum capacity (the motor shield's maximum allowable current is $15$ A). During transmitting the Step Input, the position data sent to the motor shield passes through a first-order Low Pass Filter with a rise time of 0.5 seconds. The exoskeleton is designed to mask the pressure sensation ($0.015$ N/cm$^2$) over the arm. The internal pressure is measured with four FSR sensors (size: $4.5 \times 4.5$ cm) when donning the exoskeleton. In addition to the exoskeleton, the computer screen and speaker provide visual direction and sound cues. Figure 4.3(c) shows the User Interface on screen. The FSR readings are displayed on the top right corner of the screen as four red circles. When the pressure reading becomes more than the threshold, it turns green. We used MotorWare's built-in electrical angle module instead of the custom electrical angle estimation feature from [33] to improve the BLDC motor's stability. The testbed, Figure 4.3(b), for holding the upper arm $45°$ is built. Rather than initiating the arm movement from full flexion, the inclined reference position accounts for the possible misalignment between the exoskeleton and the various arm geometries. Thus, the range of motion is designed to be $45 \pm 45°$ The torque constant, $0.072$ Nm/A, implemented in the result, was calibrated with a FUTEK LBB200 load cell with a Tacuna Systems Amplifier v.2.3.

#### *Software*

Three software platforms were implemented. Processing 3.0 is used for realizing the Graphical User Interface; Texas Instruments Code Composer Studio (CCS) for sensor reading and actuating the motor; MATLAB for the debugging.

The randomized experiment pairs for different subject groups are defined in Processing 3.0. Upon executing the software, it loads all the predefined segments of path files into object arrays. Then, the internal state machine transmits each position input in a random order to CCS. The GUI system realized with Processing is run at $100$ Hz while data logging speed was $200$ Hz based on the SCI communication between the Processing and CCS. Next, the CCS puts the position input to the corresponding controller. The MATLAB processed the received data from Processing and presented the tracking performance. After each trial was conducted, the visualized results were provided via MATLAB.

The position error-driven controller (PD controller) with different gain conditions are defined in CCS (Figure 4.4). From the implemented PD controller, the human arm has much inertia compared to the exoskeleton, and the torque stiffness is only controlled with the proportional gain. The damping term is only tuned once without the human arm. Hence, the corrective effort suitable for JND is evaluated by the torque slope or the Proportional gain.

We chose four different levels of torque stiffness illustrated in Figure 4.4. Gain One (G1) is a shallow gain with $12$ mNm/deg; Gain Two (G2) is a steep gain of $38$ mNm/deg; Gain Mixed (GM) is a combination of G1 and G2; Gain Three (G3) is a virtual wall, with $95$ mNm/deg. From Figure 4.4, the slope is steepest to shallowest in the order of G3, G2, GM, and G1. The figure also shows the position threshold that defines the boundary where proportional spatial information can be delivered: any position error beyond this boundary corresponds to the motor providing a fixed maximum stall torque. The position thresholds for the different conditions, from smallest to largest, are in the order G1, GM, G2, G3.

From pilot testing, we found that when the position threshold became smaller than $10°$, it caused an unstable feedback system to the wearer by constantly switching the direction of the torque. Therefore, when sizing the shallow torque slope (G1), we used the smallest position threshold ($10°$) in conjunction with the converged torque value from our prior study, which is about $0.12$ Nm [34]. The G1 slope passes close to this point. The G3 is the maximum torque slope that was achieved for our hardware setting. In this case, we chose the slope so that the maximum torque provided by our exoskeleton (1.1 Nm) occurred with close to $10°$ of position error.

The position before reaching the stall torque for each gain condition ($\theta_{stall}$) are $87°$, $55°$, $27°$, and $11°$ for G1, GM, G2, and G3. The GM level is explicitly implemented to investigate whether the change of slope with the position threshold ($\theta_{threshold}$) could affect the tracking accuracy. Specifically, we shifted the G1 with the $30°$ offset around the breakpoint of $0.5$ Nm and $13°$ to design GM (Figure 4.4). By setting the offset, we hypothesized that the GM would yield about $30°$ more accurate result than the G1.

From figure 4.2, the torque slope and the position threshold are determined based on the perceptual threshold, $\tau_p$, which was $0.14 - 0.18$ Nm from our prior study [34]. The corresponding position threshold ($\theta_{threshold}$) becomes $1, 3,$ and $10°$ for G3, G2, and G1, respectively. However, as explored in the pilot test, too small position threshold potentially leads to unstable system. Thus, we set the $\tau_p$ to be $0.5$ Nm such that the steep slope condition, G3, has the $\theta_{threshold}$ of $5°$ on the each direction.

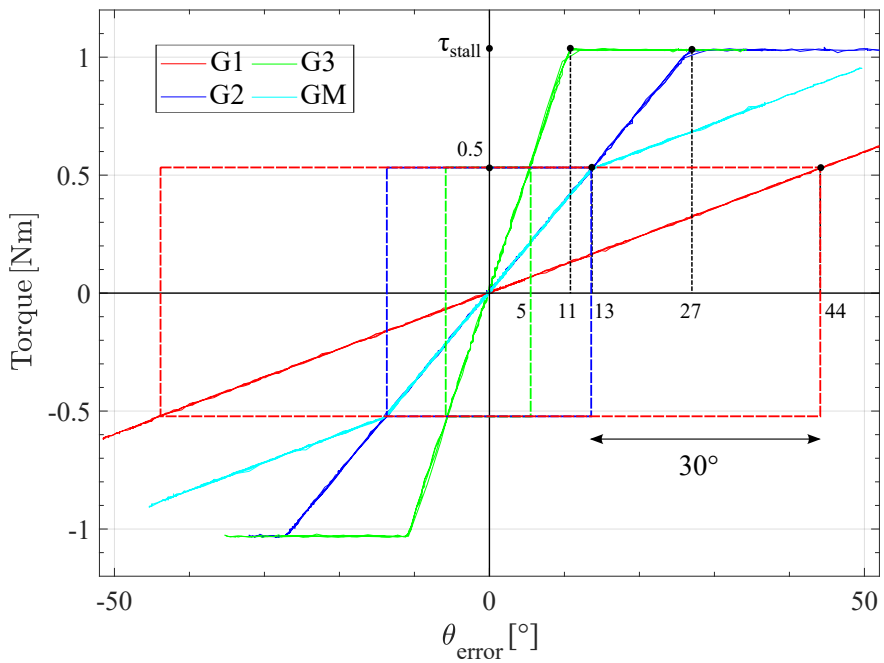


Figure 4.4: Four different levels of torque stiffness. Boxes made of dashed lines represent each position threshold ($\theta_{threshold}$ from the Figure 4.2). Also, black dotted lines extending down to the x-axis indicate the $\theta_{stall}$. Three gains shown are Gain One (G1, shallow gain-12 mNm/deg); Gain Two (G2, steep gain-38 mNm/deg); and Gain Three (G3, virtual wall-95 mNm/deg). Note that the mixed gain GM is designed to have the same torque slope as G1, with an offset of 30°. The positive position error indicates the flexional arm displacement; the negative error is extensional arm motion. $\tau_{stall}$ is the system's peak torque. $\tau_p$ represents the perceivable threshold that is modified according to the exoskeleton environment.

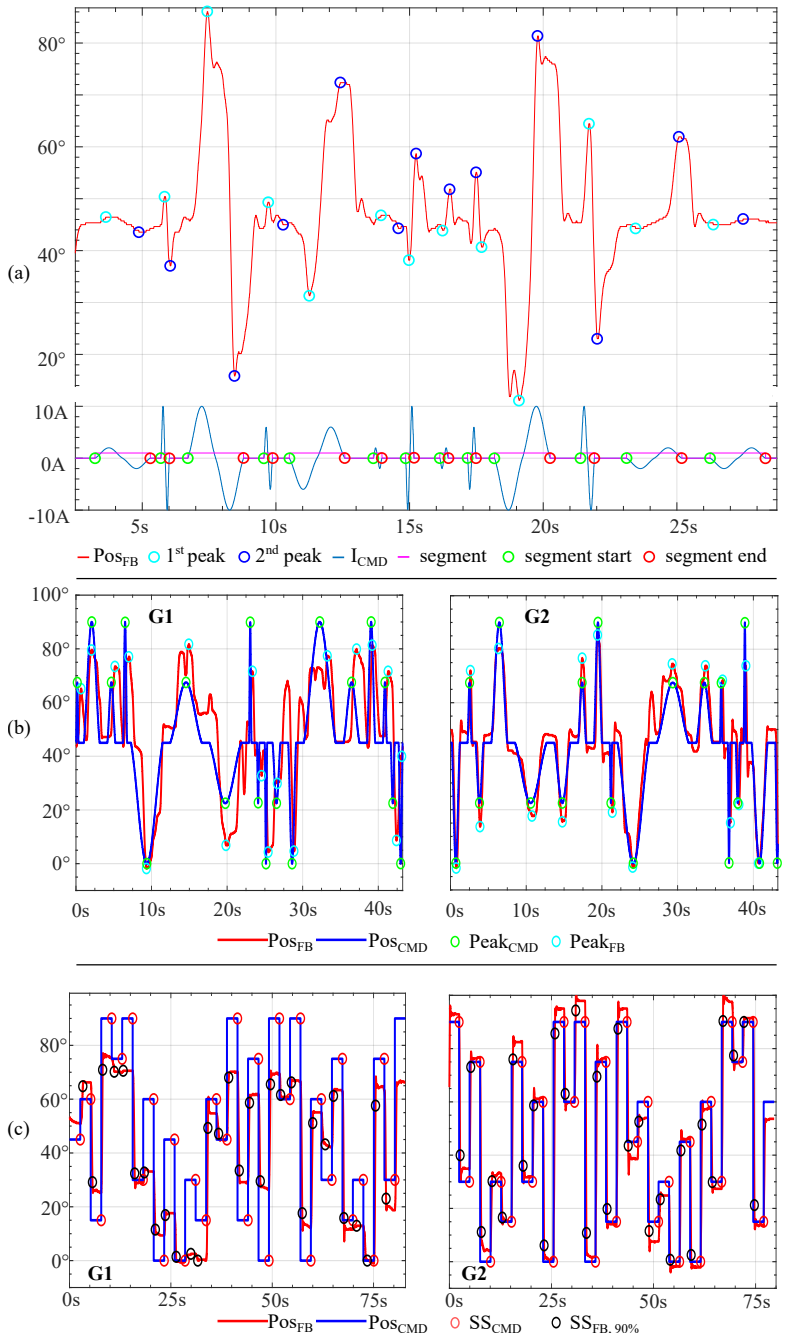


Figure 4.5: Sample test results showing how piece-wise analysis is conducted. Blue is the commanded input, and Red is the wearer's arm movement. The three types of input are shown in different subfigures. (a) Current Pulse. Magenta is the segment; green and red circles show the beginning and ending points; cyan and blue circle indicate first and second peaks. (b) Position Pulse. Green and cyan circles are commanded position peak and arm movement peak. (c) Position Step. The red circle is the steady-state commanded position; the Black circle illustrates 90 % of the arm's steady-state position. The positive direction illustrates the arm flexional motion; negative y-values show arm extension movement.

### *4.2.2 Methods*

*Participants*

We recruited 15 subjects, 13 male and 2 female, ages $29.31 \pm 4.00$ years, who were all healthy right-handed participants. The experiment was reviewed by the Human Research Protection Program at Virginia Tech for the COVID-19 mitigation process and approved by the Virginia Tech Institutional Review Board (IRB #16-175).

All the participants were informed about the COVID-19 safety guidelines along with the study's goal. Specifically, the individuals were instructed not to entirely relax the arm during the test and pay attention to how they feel on their right arm. Also, they are asked not to wiggle the arm when they are not sure about whether the force is applied or not. Some subjects with thin arms had additional padding foam inside the exoskeleton cuff. The authors could not find any supporting details about the gender difference or the muscular proportion effect on the perceptual level. Therefore, the data was not normalized per each subject weight.

Subjects are randomly assigned to two groups, Position Step or Position Pulse. Both groups first participated in the Current Pulse test, followed by either Position Step or Position Pulse, depending on the assigned group. The test takes about 45 minutes (10 minutes of Current Pulse, 30 minutes of Position Pulse or Position Step, with 12 minutes (4 times 3 minutes) break between sessions). A demonstration trial was preceded for each session. $10 monetary compensation was provided for the participation.

*Pre-programmed Input*

From the pilot study, we noticed a staircase trend was observed during tracking minimal jerk trajectory [33]. At first, the person moved the arm as long as one felt it. Then they waited until perceiving a stronger sensation. We believe that this observation could be further studied by changing the torque stiffness and potentially be useful when analyzed in the frequency domain.

Because of human anticipatory behavior, the traditional system identification method using a sine-sweep does not work. One option could be implementing the filtered white noise [19]. However, due to the prerequisite condition of masking the skin pressure, the time wearing the exoskeleton should be minimized. Accordingly, we need to examine the specific input condition within a reasonable time. Unfortunately, though the filtered white noise provides good randomness, we could not regulate the total time window sufficiently.

Another approach is to combine the pre-selected inputs randomly. We prepared a single input for the current pulse, position pulse, and position step with specified magnitude and duration. The pulse position trajectory for both Current and Position is characterized by a raised cosine function with the roll-off factor of 0.5 and limited to positive values:

$$Torque = \max\left(\left(\frac{sin(\pi t/T)}{\pi t/T}\right)\left(\frac{\cos(0.5\pi t/T)}{1-(t/T)^2}\right),\ 0\right)$$

where $Torque$ is the torque applied, $t$ is time, and $T$ is the period. With $T = 1$, this results in a pulse shape from $t = -1$ to $+1$, with a maximum amplitude of $1$ at $t = 0$. A single waveform was generated in this manner and then scaled to match the amplitudes and durations for each condition. Unlike the Position Pulse being applied as a single pulse, the Current Pulse is applied as a pair of either positive-negative or negative-positive waves (Figure 4.5).

The programmed state machine in Processing 3.0 checks if the whole combination is finished. Then, it randomly selects the base magnitude and duration. Lastly, the system resizes the original

input array and exerts the input.

**Current Pulse (CP)**. The chosen magnitude for the CP involves $2, 6, 10A$, which corresponds to torques of $0.144, 0.432, 0.72$ Nm. The duration set is $0.3, 0.5, 1, 2$ seconds (which we plot as $3.3, 2, 1, 0.5$ Hz), leading to twelve input combinations. However, we applied two pulses consecutively in the opposite direction for the CP as a pair. Therefore, the CP state machine randomly chooses the combination pair of the three magnitudes, four duration with each positive-negative and negative-positive pair. The CP only informs subjects when to initiate the arm motion. Accordingly, the CP should be paired to apply the restoring force with a beeping sound being emitted as a termination stimulus upon the completion of the trial. Please find Figure 4.5(a) for the sample CP result and Figure 4.6(a) for the segmented $10$ A input.

**Position Pulse (PP)**. We created pulse signals from five different durations ($0.6, 1, 2, 4, 8$ seconds, corresponding to frequencies of $1.67, 1, 0.5, 0.25, 0.125$Hz) with the magnitudes of $\pm 22.5°$ and $\pm 45°$. The generated pulses are played in a random order. The state machine programmed in Processing randomly chooses the magnitude and the duration at every trial until all combinations of pairs are finished. It always started at $45°$, going to a peak angle of either $67.5°$ or $90°$. To mitigate human guessing, the system also adds or subtracts a small amount of magnitude $(0 - 10°)$ from the peak. Next, the input always goes back to $45°$. Thus, we compare the four different perceptual conditions, assuming that the arm responds similarly in terms of its range of motion. The time in between each pulse is 2 seconds. Please find Figure 4.5(b) for the sample Position Pulse result and Figure 4.6(b) for the segmented $45°$ input.

**Position Step (PS)**. The Low-Pass Filter implemented is the first-order filter with a $1$ second rise time. Then each segment of input lasts 5 seconds. The initial and final angles are selected from $\{0°, 15°, 30°, 45°, 60°, 75°\}$. Based on our pilot test, we eliminated some combinations where the input is sometimes too small for the wearer to perceive it and others to reduce the total experiment

time. The excluded pairs are: $0°$ to $15°$, $15°$ to $45°$, $30°$ to $45°$, $45°$ to $75°$, and $60°$ to $75°$ for the increasing and $15°$ to $0°$, $45°$ to $15°$, $45°$ to $30°$, $75°$ to $45°$, and $75°$ to $60°$ for the decreasing order. Please find Figure 4.5 (c) for the sample Position Step result and Figure 4.6 (c) for the segmented $0°$ to $90°$ input.

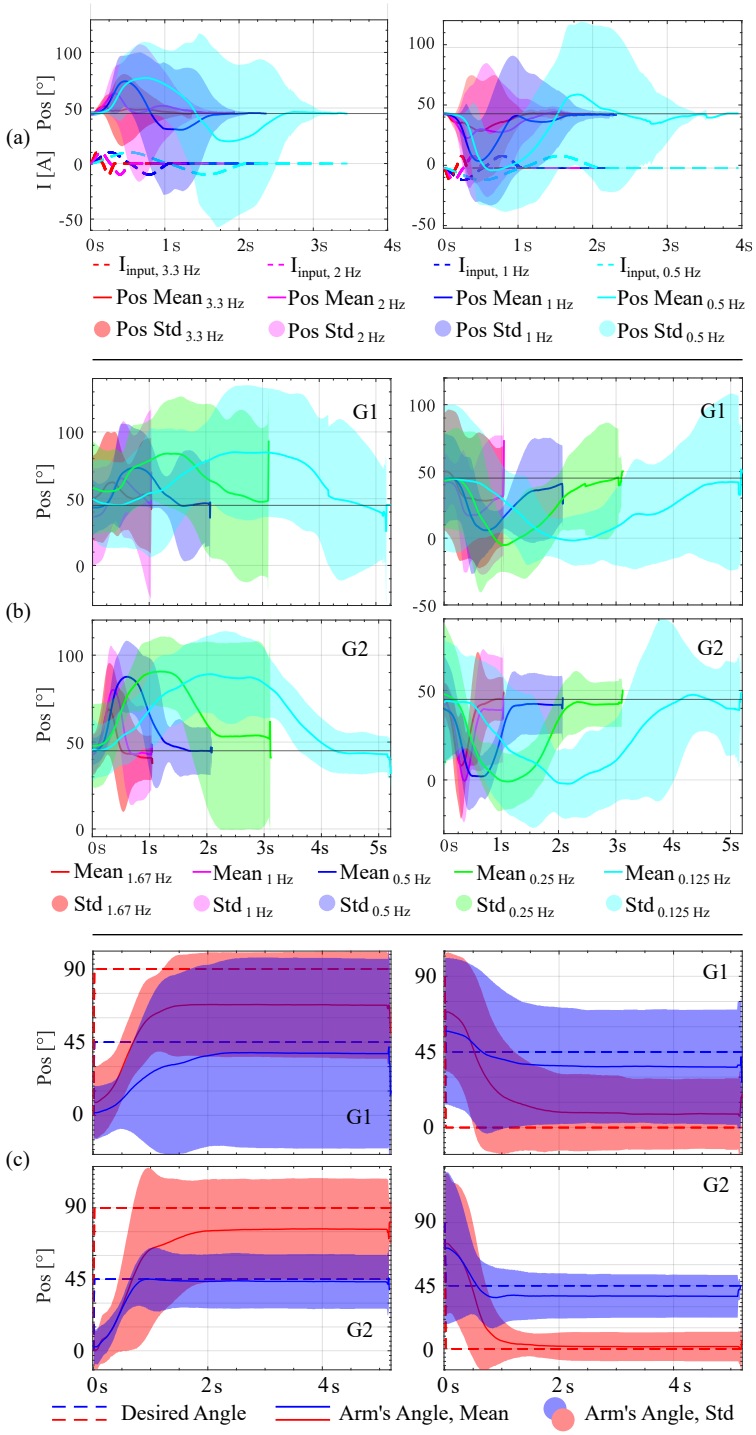


Figure 4.6: Test results for several examples of each input type. In each plot, the solid line shows the mean arm angle as a function of time; the shaded area shows one standard deviation from the mean. **(a)**: Current Pulse result of $10$ A magnitude with $0.3, 0.5, 1$, and $2$ seconds of duration. The dashed line shows the applied current. **(b)**: Position Pulse result of $45°$ Pulse Input with $0.6, 1, 2, 4$, and $8$ seconds of duration. **(c)**: Position Step result of increasing angle from $0$ to $45°$ and $0$ to $90°$ and decreasing angle from $90$ to $45°$ and $90$ to $0°$. Positive angle indicates the flexional arm movement; the negative numbers are extensional arm angle. The dashed line shows the commanded step input.

*Experimental Protocol*

The experiment was carried out in two groups, one group with Current Pulse-Position Pulse and another with Current Pulse-Position Step. For both groups, the Current Pulse was tested before the other experiment. Also, a tutorial session was preceded for every session. The tutorial's goal was to provide the correct answer such that one could grasp the knowledge of their behavior's result. The results shown on the screen during the tutorial were the visualized individual's arm and desired location, and average time delay and position error.

The break time in between the sessions was selected to be at least 5 minutes. During the break, the motor is powered off, and the subjects are allowed to take off and relax the arm.

Amid the break between sessions, the individual's tracking result of the relevant trial was visualized on the screen. The screen included the robot's desired path and the participant's arm movement. The report was introduced to motivate and facilitate the participant's engagement, just like the "attention checks" sections in a psychological survey. Since the proprioception is a result of physiological and psychological efforts [29], paying attention throughout the experiment played a key role.

Here are the shared experimental procedures applied to all three experiments. The test starts with the experimenter helping the participant wear the exoskeleton. When donning the exoskeleton, the subjects' elbow was made to be aligned with the motor. Also, to mitigate the tilting of the shoulder and unexpected fatigue, individuals were asked to sit as close as possible to the aluminum pad. In addition arms were fully extended and supinated when tightening the straps. Next, the participant changed one's posture for the comfortable stance. The experimenter adapted the sitting height and the platform's location based on the arm's volumes and weight. Then, the tester ran the GUI system and enabled the CCS script. Then, the exoskeleton calibrates the index pin to reset the encoder. The

data logging began. For both Position Step and Position Pulse, the initial arm angle is 45 degrees inclined from the desk to minimize the gravity effect near the reference angle. However, some subjects with long upper arm length tended to incline too much toward the desk and the support had to be adjusted accordingly. Finally, the GUI setup and required rules were explained to the subjects before the practice session. During the practice, subjects were allowed to wiggle the arm to feel how it would be to be driven. Yet, they were asked not to be hesitant during the actual sessions.

Unlike Position Pulse or Position Step, subjects could not tell when the input execution process is finished during the Current Pulse because the end of the second torque pulse was difficult to detect. In general, the torque pulses frequently led to the arm moving with a net displacement away from the starting position. Therefore, a beep sound is introduced to cue subjects when the exoskeleton finished applying torque in both directions. Upon the beeping sound, subjects are asked to move the arm back to the neutral position. The experimenter confirms that the arm is neutral before proceeding to the subsequent trial.

The experiment was conducted in an empty space apart from the participant and the experimenter. During the tracking, subjects were allowed to close their eyes for better concentration except for the mini-game session, which provided visual information.

## 4.3 Results

In this study, JTF perception during tracking was evaluated regarding position error and time delay. Instead of the conventional system identification method, we applied the randomly ordered discrete unit input. For each session, the continuous time-domain data (Figure 4.5) is discretized into each segment, Figure 4.6 via MATLAB. The test data is analyzed as a repeated measure design. The p-value from the Repeated Measures ANOVA (RMANOVA) was corrected with the false discovery

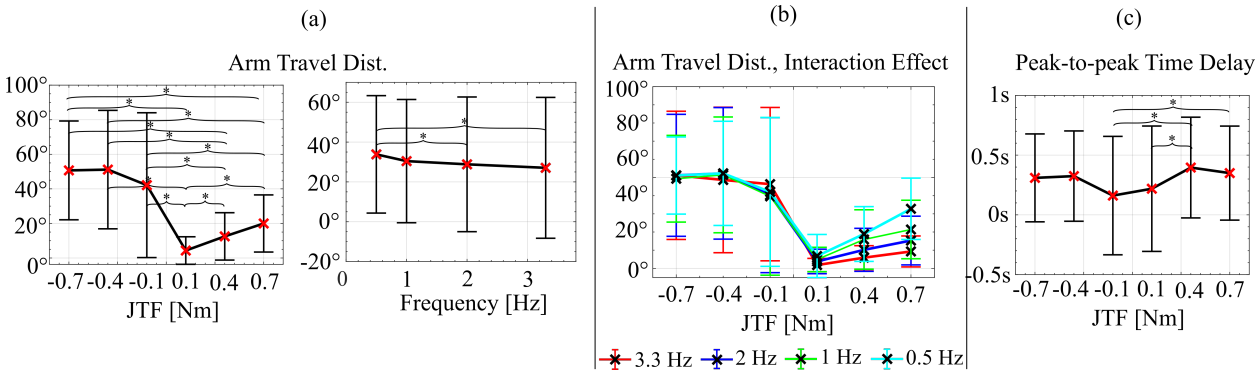


Figure 4.7: Measurements of the tracking data for the Current Pulse. (a): Arm's Traveling distance as JTF (left) and Arm's moving speed (right) as fixed effects. (b): Interaction effect between the magnitude of the JTF and the movement speed on Arm's traveling distance. The arm travel distances plotted are the absolute value of the travel distance for ease of comparing the positive and negative torques; the negative torques resulted in negative travel distances. (c): and the peak-to-peak time delay from a Current Pulse. JTF of the positive values are flexional input; the negative values are extensional feedback. For (a) and (c), The statistically different pairs are highlighted with asterisks.

rate (FDR) method to take care of multiple hypothesis testing. Pairwise comparisons were made with Tukey post-hoc test. The software used for the statistical analysis was JMP Pro 15.

### *4.3.1 Current Pulse*

The goal of the study about Current Pulse is to determine if which of the two, the current magnitude and duration, are the major source of driving the arm's movement.

For the Current Pulse, the following equation is implemented for data processing, for each segment $i$:

$$\Theta_{Current\ Pulse,i} = |\Theta_{end\ arm,i} - \Theta_{start\ arm,i}|$$

where the $\Theta_{end\ arm,i}$ is the arm position at the endpoint of segment $i$, and $\Theta_{start\ arm,i}$ is the arm's position at the beginning of the segment. Figure 4.5(a) illustrates how the segmentation is performed. Since the current pulse was applied as a double-shape (positive-negative or negative-positive), two

peak points were identified.

The RMANOVA was conducted with the current magnitude and duration as the Independent Variables (IVs) and the arm's traveling distance as the Dependent Variable (DV), Figure 4.7. Significant results were found for the duration effect ($F(3, 41.81) = 6.747,\ p = 0.001$) and magnitude effect ($F(5, 60.27) = 115.741,\ p < 0.0001$). Noticeable significant interactions in Figure 4.7(b) are the JTFs applied on the flexional direction. From the pair-wise t-test, the input of $0.74$ Nm on the flexional direction has statistical significance between $3.3$ and $0.5$ Hz ($t$= $4.81,\ p = 0.0008$) and between $2$ and $0.5$ Hz ($t$= $3.757,\ p = 0.042$).

On the other hand, the peak-to-peak time delay only yields statistical significance for the magnitude ($F(5, 71.23) = 4.107,\ p = 0.0074$).

### *4.3.2 Position Pulse*

We determined the statistical significance of Position Pulse using the peak-to-peak position error per segment. Figure 4.5 (b) illustrates how the segmentation is performed.

Position magnitude, pulse duration, and the toque slope are selected as the IVs. The DV was the absolute peak-to-peak error per each segment. The fixed effect test on the gain yielded ($F(3, 12.98) = 19.151, p = 0.0003$) and on the duration ($F(4, 22.28) = 3.945, p = 0.034$). The significant pair was observed between the $1.677$ and $0.125$ Hz to be $13.206 \pm 11.922°$ and $8.877 \pm 7.212°$ ($t$= $3.419,\ p = 0.0185$). However, we found no significance on the position magnitude between $22.5°$ and $45°$. The unpooled means and standard deviations of the gain effect are: $16.91 \pm 12.61°$ for G1, $9.279 \pm 6.939°$ for G2, $6.102 \pm 5.117°$ for G3, and $11.789 \pm 9.701°$ for GM.

The peak-to-peak time delay is also examined with respect to gain, duration, and the input magnitude. The gain effect is found ($F(3, 18.02) = 6.971, p = 0.0182$). The unpooled means and stan-

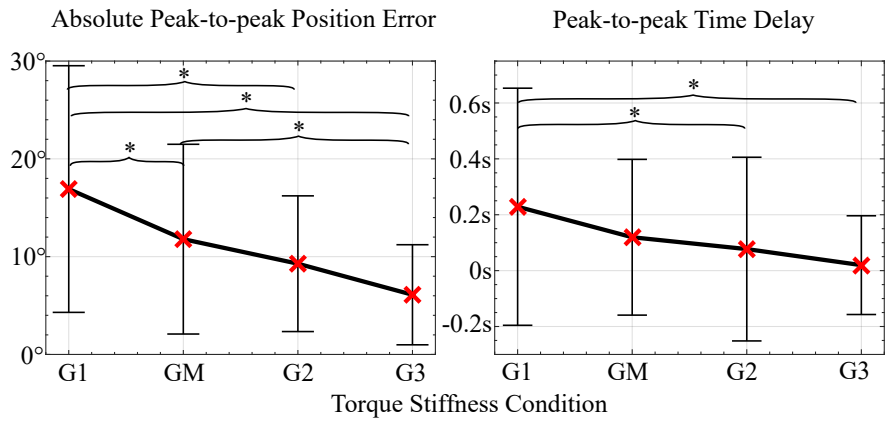


Figure 4.8: Absolute Peak-to-peak Error for Position Pulse for the gain effect (left) and peak-to-peak time delay for the gain effect (right). Further details about the significant pairs are represented as asterisks.

dard deviations of the gain effect are: $0.229 \pm 0.424$ s for G1, $0.077 \pm 0.329$ s for G2, $0.0196 \pm 0.177$ s for G3, and $0.119 \pm 0.279$ s for GM. The duration of the pulse and the magnitude did not have any significance.

We also studied the relationship between the peak JTF and the torque stiffness, Figure 4.9. For the gain effect, the Repeated Measures ANOVA yields $F(3, 16) = 45.628,\ p < 0.0001$. For duration effect, $F(4, 22.01) = 30.814,\ p < 0.0001$. The interaction effect has a significant effect between gain and duration ($F(12, 65.43) = 4.224,\ p < 0.0001$).

Moreover, the test result was evaluated in the frequency domain. The transfer function of the arm's position to the robot's pre-defined trajectory is represented in Figure 4.10.

### *4.3.3 Position Step*

The Position Step was analyzed as the ratio of absolute steady-state position error:

$$\Theta_{Position\ Step,i} = |\Theta_{end,robot,i} - \Theta_{end,arm,i}|$$

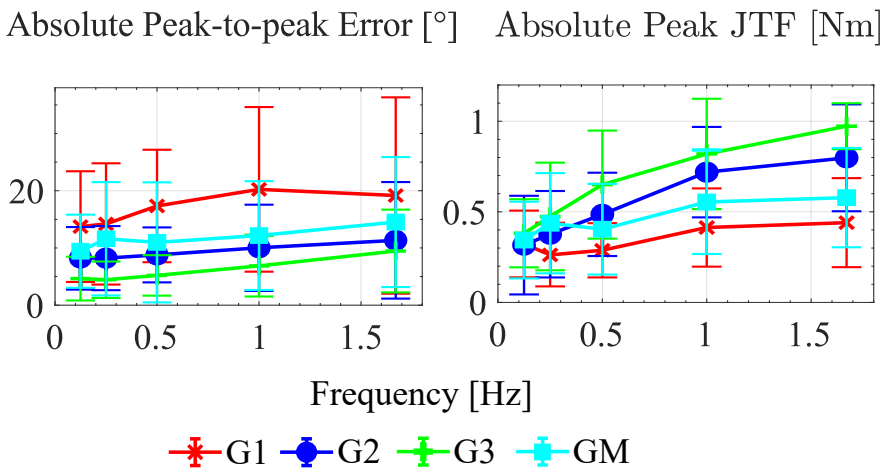


Figure 4.9: Unpooled Means and Stds of Position Pulse, peak-to-peak position error (left) and the peak JTF (right)

where $\Theta_{end\ robot,i}$ is the end position of the desired trajectory for segment $i$ and $\Theta_{end\ arm,i}$ is the end position of the arm (measured by the exoskeleton). Figure 4.5 (c) illustrates how the segmentation is performed.

The fixed effects included gain, absolute position magnitude (i.e., absolute value of the change in arm position), and the arm's moving direction (i.e., flexion or extension). The FDR-corrected $p$-value for the gain effect is $F(1, 6.241) = 15.198,\ p = 0.035$. The unpooled means and standard deviation are $15.48 \pm 12.44\,^\circ$ for the G1, while $9.04 \pm 9.96\,^\circ$ for the G2. We also found significance between the extensional motion and flexional motion with the FDR correction ($F(1, 7.769) = 11.465,\ p = 0.035$). The detailed means and standard deviations are: $8.95 \pm 7.49\,^\circ$ for the extensional input, while $15.29 \pm 14.00\,^\circ$ for the flexional feedback. However, we could not find the significance among the input magnitudes.

In terms of response time, we analyzed the $90\%$ rise time in terms of gain, direction, and magnitude of the step input. The G1 is shown to be different from G2: $F(1, 8.65) = 16.685,\ p = 0.021$ (FDR corrected). Specifically, the G1 yields $0.638 \pm 0.434$ s while G2 is $0.468 \pm 0.296$ s. However, no significance was found among the magnitude and directions of the input.

| [Hz] | Gain | Position Eror | | | | | | Peak JTF | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.125 | G1 | A | B | C | D | E | | | | | | | F | G |
| | GM | | B | C | D | E | F | | | | | E | F | G |
| | G2 | | | C | D | E | F | | | | | | | G |
| | G3 | | | | | | F | | | | | E | F | G |
| 0.25 | G1 | A | B | C | D | | | | | | | | | G |
| | GM | A | B | C | D | E | F | | | | D | E | F | G |
| | G2 | | | C | D | E | F | | | | | E | F | G |
| | G3 | | | | | | F | | | | D | E | F | G |
| 0.5 | G1 | A | B | C | | | | | | | | | | G |
| | GM | A | B | C | D | E | F | | | | | E | F | G |
| | G2 | | | C | D | E | F | | | C | D | E | F | G |
| | G3 | | | | | E | F | | B | C | D | | | |
| 1 | G1 | A | | | | | | | | | | E | F | G |
| | GM | A | B | C | D | E | F | | | C | D | E | F | |
| | G2 | | B | C | D | E | F | | B | C | | | | |
| | G3 | | | | D | E | F | A | B | | | | | |
| 1.67 | G1 | A | B | | | | | | | | D | E | F | G |
| | GM | A | B | C | D | E | | | B | C | D | E | | |
| | G2 | A | B | C | D | E | F | A | B | | | | | |
| | G3 | | | C | D | E | F | A | | | | | | |

Table 4.1: Significant levels of interaction effect between gain and duration of Position Pulse. The compared DVs are Peak-to-peak position error (left) and peak JTF within the trial (right). A-G are the group labels where the fixed effects with the same label are not statistically significant.

Unlike PP which always begins at $45°$, PS has several beginning positions and end positions. Because of this, we could create a map between the start and end angle like Figure 4.11. The values represented are unpooled means.

## 4.4 Discussion

Although both magnitude and duration of the pure torque pulse are responsible for increasing the arm's travel distance, more numbers of statistically significant pairs are observed on the magnitude side (Figure 4.7(a)) with smaller $p$-value ($<0.0001$ for the magnitude and $0.0012$ for the duration).

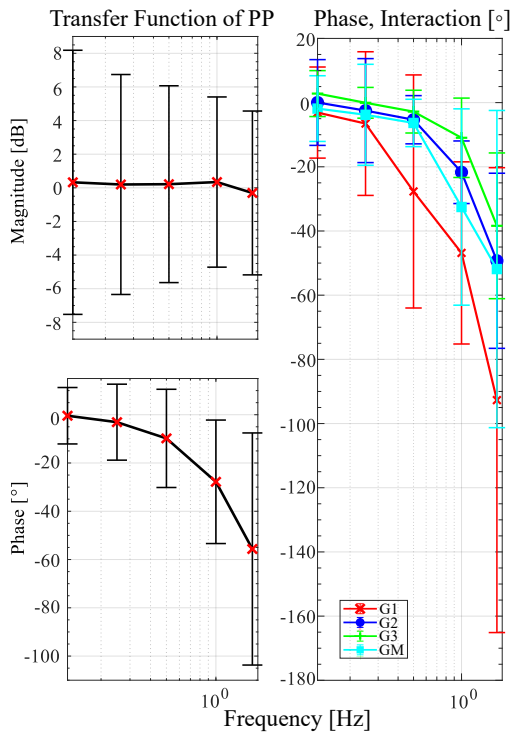


Figure 4.10: Unpooled Means and Stds of the Position Pulse's bode plot. The magnitude, dB, is calculated by the ratio of the arm's peak to the input peak. The raised cosine function is treated like a sinusoidal wave when the data was converted into phase. The figure on the right side shows the gain effect on the phase delay.

Specifically, in Figure 4.7(b), the flexional JTF of $0.74$ Nm, which is equivalent with 10 A current, have different arm's traveling distance per different frequencies of the torque input. On the contrary, the extensional direction of torque inputs is likely to lead to the end of the arm's range of motion ($45°$) regardless of the arm's speed. Interestingly, the input of 2 A current pulse, which is $0.144$ Nm, is similar to the smallest noticeable torque for the static condition ($0.098-0.101$ Nm) [34]. Subjects could initiate the exoskeleton's guidance to some extent, but the tracking performance was inferior. For the extensional direction, whether applying $0.144$ Nm or $0.72$ Nm does not change a lot in arm's traveling distance. In contrast, the flexional direction movement, $0.144$ Nm of torque only drives roughly $5°$ no matter how fast the robot drives the arm. The results illustrate that driving the wearer's arm is tentatively easier in extensional direction than the flexional motion, and potentially moving the arm in flexional direction could be influenced by higher mechanical joint stiffness.

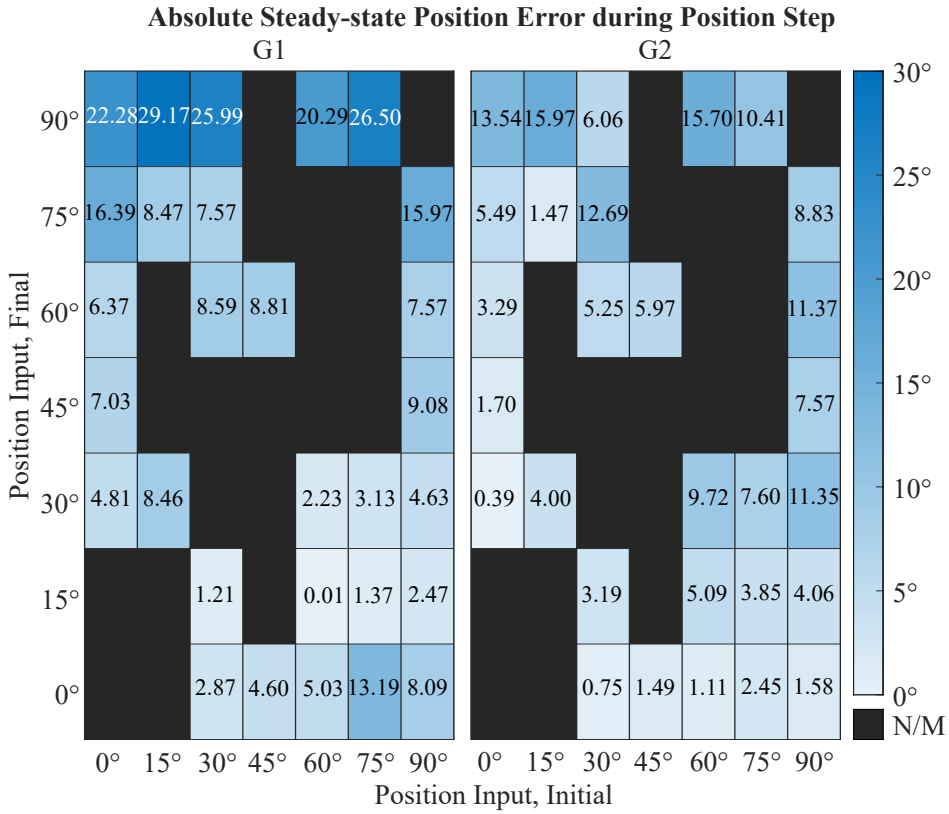


Figure 4.11: Unpooled mean values of steady-state Position Step error represented as a vector table. The x-axis indicates the arm's initial position and the y-axis is the final position. The Left panel is the G1 condition and the Right panel is G2. $0\,^{\circ}$ is where the arm is fully extended. N/M indicates that the corresponding initial-final step input pair was not measured.

However, the peak-to-peak time delay at different arm speeds is indistinguishable at all pulse durations. In terms of magnitude effect, an almost symmetric pattern is showing from Figure 4.7(c) with three significant pairs. For the 2 A current input, the $0.25$ Hz condition resulted in a large standard deviation and even a slightly negative peak-to-peak time delay, meaning that the participants reached a maximum before the actual torque did, presumably because they had a hard time detecting the torque. Therefore, we can argue that the JND level of torque could be helpful in the initiation of the movement, yet insufficient in pursuing tracking tasks.

Contrary to the CP, the position controller concurrently corrects the arm movement during tracking in the Position Pulse (PP) condition. Thus, gain level dominates over other fixed effects, the pulse's duration or magnitude. We hypothesized in the Introduction that once the torque slope becomes so large that $\theta_{threshold}$ becomes $\theta_{stall}$, the tracking performance might be degraded. However, higher

gain leads to better accuracy in speed and position even for our steepest gain, G3 (Figure 4.8). The result contradicts our pilot test, where the subject was guided with the pure digital-like JND of JTF. During the pilot test, subjects were confused about the feedback and could not collect the data; the tracking behavior was accompanied by overshoot and multiple back-and-forth iterations. The critical difference between the test result and the pilot test was that the highest gains still provide angular information, despite the applied torque reaching the stall torque instantaneously. Alternatively, the slope of G3 could be not strong enough to realize the actual virtual wall effect of the system.

Another noticeable point is that GM acts as a slope between G1 and G2 despite the fact the slope G1 applied from $0-13°$ and $13°$- for G2. In terms of both position error and time delay, GM positioned in between G1 and G2 most of the cases, Figure 4.8. The tracking performance could be listed as G3-G2-GM-G1 from the most accurate result to the least accuracy. Those order is completely reverse of the position stall ($\theta_{stall}$) to reach stall torque level: $11°$(G3)-$13°$(G2)-$27°$(GM)-$87°$(G1). This could imply that the participants perceive the JTF created by the GM as the average of G1 and G2, rather than recognized by two discrete torque sections. Additionally, $\theta_{stall}$ seems to be the most critical element determining the performance of the tracking. Moreover, we anticipated that G1 would have $30°$ of offset larger than GM because of $\theta_{threshold}$ difference between the two gain conditions, Figure 4.4. Yet, those yielded nearly $15\ °$ difference, Figure 4.8. This means that the mixed gain, GM, does not simply behave as an offset of the G1. Rather, we assume that the G1 applied from $0-13°$ compensated the offset effect.

Figure 4.9 and Table 4.1 further illustrate the relationship between the torque slope and the JTF. In general, a steep slope is responsible for the motor getting to its stall condition more quickly. The interesting observation is that the averaged peak JTF (i.e., the actual torque applied during the PP feedback) of GM with slow motions ($< 0.2$ Hz) seems to be larger than the peak JTF with the G2 condition. Then, the peak torque for GM becomes smaller than the peak torque for G2 at

frequencies $> 0.2$ Hz. GM has a steep slope within $\pm 13°$ of the position threshold. It is observed that the slow-motion region is affected by the steep slope, resulting in a larger peak JTF than the other side of the motion. Although all of the mean values from Figure 4.9 are above the JND of static condition [34], the torque slope of G1 is so shallow that the overall peak JTF observed through the tracking had the smallest amplitude with no statistical significance. G1 also yielded the highest position error and was not statistically significant, either.

Interestingly, at $0.125$ Hz, all of the gains required the same current during the experiment (i.e., had almost exactly the same peak JTF torque), but G3 resulted in better accuracy than G1 (and the other gains were in between the two). Thus, the steeper slope of G3 allowed participants to track the pulse more precisely.

Moreover, we examined the frequency elements of the PP Figure 4.10. The magnitude element of the transfer function did not have noticeable changes over the arm speed. However, the phase plot of the PP per different gain shows a comparable phase delay. Previously kinesthetic guidance was studied with the left arm's position guiding the right arm's movement [17]. Neilson et al. found that around $1$ Hz, phase had $-50\,°$ delay. From our test result, G2 and GM yielded similar phase values around $0.834$ Hz. On the other hand, G1 had nearly $-90\,°$, which is much slower than Neilson et al.'s kinesthetic response test. The G3 result had even less of a phase lag, around $-40\,°$.

Similar to the gain effect for PP, results from the Position Step (PS) indicate that the higher gain leads to smaller spatial errors and time delays. Particularly, we focused on the different tracking performances in either direction. The directional difference in statistical analysis is further explored in Figure 4.11, in terms of the full combination of input-output position. The trend was consistent throughout the map that the flexional motion yielded relatively more position error than the extensional motion. Those arguments are in agreement with our test results for both G1 and G2 conditions. G1 and G2's top row has large position errors where the values correspond to the arm

flexing. On the other hand, the bottom row of both conditions has small values corresponding to the arm extending. The position errors from Figure 4.11 did not have significant change over the initial angle (per column). However, they yielded noticeable changes over the final position (per row).

One possible reason supporting the observation is related to the mechanical properties of the elbow joint. According to Latash et al.'s study, passive joint stiffness refers to the mechanical properties of body tissues, such as muscle-tendon complex, and has a property of varying depending on the initial joint position [41]. The author highlighted that passive joint stiffness increases with elbow flexion [41, 44]. Given this, if the exoskeleton is moving the arm and the body is sensing the displacement after the exoskeleton moves it, the exoskeleton will move the arm more in extended positions as compared to flexed positions due to the lower passive stiffness. If the arm moves more, presumably, you have an easier time sensing that. So, we would expect that extending your arm (and ending up in an extended position) will result in better Position tracking accuracy than being guided to flex your arm. But, this is a passive stiffness that occurs when the arm is externally driven, not an active stiffness that is happening during the active tracking, as in this experiment. Therefore, it cannot explain the whole position error more significantly in the flexional direction.

Another perspective might be a perceptual difference, meaning that persons perceive flexional or extensional differently. In previous work about the JND torques in dynamic conditions (Chapter 3), the flexion directional torque feedback under arm's flexional movement at $200\ ^{\circ}/\mathrm{s}$ conditions had a higher median value than the extensional directional torque input during arm's extensional movement at $200\ ^{\circ}/\mathrm{s}$ ($0.799$ and $0.428$ Nm, respectively, for the faster movement speed). So, more torque is required to sense flexion torques than extension torques. Besides, from the tests of CP (Figure 4.7), we learned that the flexional direction of torque drives the arm smaller than the extensional input. Thus, if the exoskeleton is causing the arm to move in flexion, we would expect the accuracy to be worse than if the exoskeleton was causing the arm to move in extension.

## 4.5 Conclusion

In this study, we explored human tracking performance with respect to the exoskeleton's torque guidance. From the current pulse, we found that the JND level of torque could initiate the tracking motion yet led to poor tracking accuracy. Moreover, when the magnitude of the current is small, the arm traveling distance was indistinguishable. At the same time, the large amplitude was responsible for no statistical difference among time delays. Then tracking the performance of different linear slopes of JTF was evaluated. We did not see the degradation of performance due to a steep torque slope (gain G3); rather, the higher gain was responsible for a higher peak torque during the tracking and better tracking accuracy. From the frequency domain of arm tracking, we noticed that the exoskeleton's kinesthetic guidance of G2 and GM yielded similar to the pattern with the kinesthetic guidance via contralateral arm proprioception. Most importantly, through observing the different gain conditions, we learned that the size of the position threshold primarily determined the tracking performance. Lastly, the torque feedback on extensional and flexional direction was compared during the step input. Our test results are in agreement with the prior study about the passive joint study.

Nevertheless, there is a limitation in our frequency domain analysis. The initial assumption was to separate the constant delay term with the frequency-dependent terms. The idea behind the assumption was that our neural delay should function like the constant term while the physiological reaction behaves like the second-order model plant. Also, we could not fully mitigate the humans guessing the external input. In some cases, the phase delay is shown to be negative, unlike our expectation. This shows that people react to external force as soon as they perceive it. As our prior experiment indicates, the wearer has even more difficulty with the perception resolution after the motion initiates, ending up in overshoot.

We were also unable to determine if the people were sensing force in their muscles or were sensing

some small amount of arm movement due to the applied torque. Finally, the implemented torque stiffness in this study (with a linear PD controller) could be improved. In the future, other types of controllers, such as impedance controllers or other functions relating position error to applied torque, could replace the linear slope and potentially achieve better tracking performance. It would also be interesting to study the optimized torque profile by adopting a higher-level controller.

## 4.6 Acknowledgment

This work was supported by a Virginia Tech ICTAS Doctoral Scholars Fellowship (H. Kim) and a Google Faculty Research Award.

# Chapter 5

# Conclusion and Future Work

In this project, we closely examined how to apply the torque feedback to the elbow joint and determined the psychophysical parameters of how well humans can detect external torques or how stiff the torque input should be to augment a human motion without interference with the voluntary movement.

First, we provided step-by-step instructions for building a lightweight elbow exoskeleton. As a partial-assistive device, the output was expected to be less than 2 Nm, which was less than $5$ % of the maximum torque that a typical adult male could generate at their elbow. Thus, A cheap Brushless DC hobby motor and easily accessible hardware were introduced for the manufacturing process and the open-sourced embedded architecture for data logging. The total cost and the weight of the joint actuator are $500$ g and $509$ $\$$.

Second, we quantified the human perceptual ability to detect JTF at the elbow during all possible combinations of preload torques or every attainable sequence of arm movement. We assessed perception capability using the Interweaving Staircase Method and derived torque slopes per various joint conditions. For either directional torque feedback, flexional motion required a 1.89-2.27 times larger speed slope, in mNm/(°/s) than the extensional motion. In addition, the muscles' isometric contraction with the aiding direction required a larger slope, in mNm/mNm than the opposing direction (7.36 times and 1.02 times for extension torque and flexion torque).

Finally, the guidance performance of the JTF was evaluated in terms of time delay and position

error between the directed input and the wearer's arm. When studying how much the human arm travels with JTF, the absolute magnitude of the input shows more significance than the duration of the input ($p$-values of $< 0.0001$ and $0.001$). In the analysis of tracking the pulse input, the highest torque stiffness, $95$ mNm/deg, is responsible for the smallest position error, $6.102 \pm 5.117°$, despite the applied torque acting as compulsory stimuli. By applying different levels of the gain slope, we learned that the position threshold corresponding to stall torque primarily determined the tracking performance.

Nevertheless, there is a limitation in our study. Inherently, the measured data via psychophysics tool relies on the user response. Because it is not something directly measured, it is difficult to differentiate the primary source to derive the responses. For example, we were unable to determine if the people were sensing force in their muscles or were sensing some small amount of arm movement due to the applied torque. Moreover, we could not fully mitigate the humans guessing the external input. In some cases, the phase delay is shown to be negative, unlike our expectation. This shows that people react to external force as soon as they perceive it, and it happens before the peak moment of the input. The implemented torque stiffness in this study (with a linear PD controller) could be improved as well. In the future, other types of controllers, such as impedance controllers or other functions relating position error to the applied torque, could replace the linear slope and potentially achieve better tracking performance. It would also be interesting to study the optimized torque profile by adopting a higher-level controller. We could also expand the evaluation into real-world tasks such as throwing a dart with JTF, which could be similar to the prior study of Obayashi et al. [13]. Studies such as this would allow us to explore the motor learning effects of JTF.

## 5.1 Hypothesis Evaluation

We next discuss a summary of the hypotheses evaluated in this thesis, and whether they were supported or not. We present each hypothesis in turn (duplicated from the Introduction), followed by a discussion of its evaluation.

- *H0: We hypothesized that the elbow joint can be guided through an arbitrary trajectory with torques as little as 1 Nm, and a bandwidth that depends on the torque applied as a function of the arm's position error from the desired trajectory.*

  This hypothesis was supported by our experiments. During tracking tests, a torque was applied to the elbow based on the angular error between the arm and the desired trajectory. With a gain of 12 mNm/deg, the elbow can follow a sinusoidal pulse input with a phase shift of $-45°$ at a frequency of 1 Hz, while with a gain of 95 mNm/deg the elbow can follow a sinusoidal pulse input with a phase shift of $-45°$ at a frequency of more than 1.67 Hz. In both cases the peak torque was less than 1 Nm.

- *H1: We hypothesized that masking the tactile sensations at the skin of the arm, such that only the sensory organs deep in the muscle and tendon are used to detect external forces, will yield elbow JND torques of a higher magnitude than those evaluated while holding a stylus in the hand.*

  This hypothesis was supported by our experiments. The JND torques able to be sensed at the elbow while masking sensations at the skin were more than four times larger than those sensed in a different experiment while a stylus was held in the hand. This was true for both trials where the arm was stationary and where it was in motion.

- *H2: We hypothesized that the JND torque at the elbow joint measured during arm movement will have a value larger than JND torques measured while the arm is stationary and isometrically contracted.*

  This hypothesis was supported by our experiments. The JND torques sensed while the arm

was stationary and resisted static torques of 0-1.28 Nm in both flexion and extension were much less than those sensed while the arm was moving back and forth at around $100°/s$ and $200°/s$. The JND torques with the arm stationary ranged from 97-197 mNm, while the JND torques with the arm in motion ranged from 219-799 mNm.

- *H3: We hypothesized that a higher torque stiffness during elbow guidance with joint torques (i.e., a larger amount of torque given a certain angular error between a target joint angle and the actual joint angle) will lead to better tracking results, but torque slopes that are too steep will behave like an on-off controller and result in poor tracking performance.*

  This hypothesis was partially supported by our experiments. We found that higher torque stiffnesses did result in smaller errors during arm guidance. However, we did not see a decline at the highest stiffness we tested (95 mNm/deg). It may be that at even higher stiffnesses than this, that the performance declines.

# Bibliography


[1] Fumihiro Akatsuka and Y. Nomura. Velocity difference perception in passive elbow flexion movement: -effects on before/during/after acceleration times. 2017.

[2] Christian Antfolk, Marco D'Alonzo, Marco Controzzi, Göran Lundborg, Birgitta Rosén, Fredrik Sebelius, and Christian Cipriani. Artificial redirection of sensation from prosthetic fingers to the phantom hand map on transradial amputees: Vibrotactile versus mechanotactile sensory feedback. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 21, 09 2012. doi: 10.1109/TNSRE.2012.2217989.

[3] Christian Antfolk, Marco D'Alonzo, Marco Controzzi, Göran Lundborg, Birgitta Rosén, Fredrik Sebelius, and Christian Cipriani. Artificial redirection of sensation from prosthetic fingers to the phantom hand map on transradial amputees: Vibrotactile versus mechanotactile sensory feedback. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 21, 09 2012. doi: 10.1109/TNSRE.2012.2217989.

[4] K Bark, E Hyman, F Tan, E Cha, S A Jax, L J Buxbaum, and K J Kuchenbecker. Effects of Vibrotactile Feedback on Human Learning of Arm Motions. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 23(1):51–63, 2015. doi: 10.1109/TNSRE.2014.2327229.

[5] Karlin Bark, Jason Wheeler, Pete Shull, Joan Savall, and Mark Cutkosky. Rotational skin stretch feedback: A wearable haptic display for motion. *IEEE Transactions on Haptics*, 3(3): 166–176, 2010.

[6] Gareth Barnaby and Anne Roudaut. Mantis: A scalable, lightweight and accessible architecture to build multiform force feedback systems. pages 937–948, 10 2019. ISBN 978-1-4503-6816-2. doi: 10.1145/3332165.3347909.

[7] Bruce D. Beynnon, Steven H. Ryder, Lars Konradsen, Robert J. Johnson, Kelly Johnson, and A. Renström. The effect of anterior cruciate ligament trauma and bracing on knee proprioception. *The American Journal of Sports Medicine*, 27(2):150–155, 1999. doi: 10.1177/03635465990270020601. URL https://doi.org/10.1177/03635465990270020601. PMID: 10102093.

[8] Evan Brody, Brad Hatfield, Thomas Spalding, Mardon Frazer, and Francis Caherty. The effect of a psyching strategy on neuromuscular activation and force production in strength-trained men. *Research quarterly for exercise and sport*, 71:162–70, 07 2000. doi: 10.1080/02701367.2000.10608894.

[9] F.M.M.O. Campos and J.M.F. Calado. Approaches to human arm movement control—a review. *Annual Reviews in Control*, 33(1):69–77, 2009. ISSN 1367-5788. doi: https://doi.org/10.1016/j.arcontrol.2009.03.001. URL https://www.sciencedirect.com/science/article/pii/S1367578809000091.

[10] C. Carignan, J. Tang, and S. Roderick. Development of an exoskeleton haptic interface for virtual task training. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3697–3702, 2009.

[11] R. F. Chandler, C. E. Clauser, J. Mcconville, H. Reynolds, and J. Young. Investigation of inertial properties of the human body. page 171, 03 1975.

[12] Dale Chapman, Michael Newton, and Kazunori Nosaka. Eccentric torque-velocity relationship of the elbow flexors. *Isokinetics and exercise science*, 13:139–145, 05 2005. doi: 10.3233/IES-2005-0192.

[13] Tomohiro Shibata Chihiro Obayashi, Tomoya Tamei. Assist-as-needed robotic trainer based on reinforcement learning and its application to dart-throwing. *Neural Networks*, 53(4):52–60, 2014. doi: 10.1016/j.neunet.2014.01.012.

[14] Steven Cholewiak, Hong Tan, and David Ebert. Haptic identification of stiffness and force magnitude. pages 87–91, 04 2008. ISBN 978-1-4244-2005-6. doi: 10.1109/HAPTICS.2008.4479918.

[15] Martin A Colson S, Pousson M and Van Hoecke J. Isokinetic elbow flexion and coactivation following eccentric training. *J Electromyogr Kinesiol*, 9(1):13–20, Feb 1999. doi: 10.1016/s1050-6411(98)00025-x.

[16] X Cui, W Chen, X Jin, and S K Agrawal. Design of a 7-DOF Cable-Driven Arm Exoskeleton (CAREX-7) and a Controller for Dexterous Motion Training or Assistance. *IEEE/ASME Transactions on Mechatronics*, 22(1):161–172, 2017. ISSN 1083-4435 VO - 22. doi: 10.1109/TMECH.2016.2618888.

[17] Neilson P. D. Speed of response or bandwidth of voluntary system controlling elbow position in intact man. *Medical & biological engineering*, 10(4):450–459, 1972. doi: https://doi.org/10.1007/BF02474193.

[18] Shams Feyzabadi, Sirko Straube, Michele Folgheraiter, Elsa Kirchner, Su-Kyoung Kim, and Jan Albiez. Human Force Discrimination during Active Arm Motion for Force Feedback Design. *IEEE Transactions on Haptics*, 6:309–319, 7 2013. doi: 10.1109/TOH.2013.4.

[19] M J Fu and M C Cavusoglu. Human-Arm-and-Hand-Dynamic Model With Variability Analyses for a Stylus-Based Haptic Interface. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(6):1633–1644, 2012. ISSN VO - 42. doi: 10.1109/TSMCB.2012.2197387.

[20] S.C. Gandevia, D.I. McCloskey, and D. Burke. Kinaesthetic signals and muscle contraction. *Trends in Neurosciences*, 15(2):62–65, 1992. doi: 10.1016/0166-2236(92)90028-7.

[21] Brent Gillespie, M modhrain, Philip Tang, David Zaretzky, and Cuong Pham. The virtual teacher. *Proceedings of the ASME International Mechanical Engineering Conference and Exposition*, 01 1998.

[22] R.A.R.C. Gopura, D S V Bandara, Kazuo Kiguchi, and G K I Mann. Developments in hardware systems of active upper-limb exoskeleton robots: A review. *Robotics and Autonomous Systems*, 75:203–220, 2016. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2015.10.001.

[23] JE Gregory, DL Morgan, and U Proske. Aftereffects in the responses of cat muscle spindles and errors of limb position sense in man. *Journal of neurophysiology*, 59(4):1220–1230, 1988.

[24] Felix Grimminger, Avadesh Meduri, Majid Khadiv, Julian Viereck, Manuel Wüthrich, Maximilien Naveau, Vincent Berenz, Steve Heim, Felix Widmaier, Jonathan Fiene, Alexander Badri-Spröwitz, and Ludovic Righetti. An open torque-controlled modular robot architecture for legged locomotion research. *IEEE Robotics and Automation Letters*, 5:3650–3657, 2020.

[25] A Gupta and M K O'Malley. Design of a haptic arm exoskeleton for training and rehabilitation. *IEEE/ASME Transactions on Mechatronics*, 11(3):280–289, 2006. ISSN VO - 11. doi: 10.1109/TMECH.2006.875558.

[26] A Gupta and M K O'Malley. Design of a haptic arm exoskeleton for training and rehabilitation. *IEEE/ASME Transactions on Mechatronics*, 11(3):280–289, 2006. ISSN VO - 11. doi: 10.1109/TMECH.2006.875558.

[27] Mehrdad H. Zadeh, David Wang, and Eric Kubica. Human Factors for Designing a Haptic Interface for Interaction with a Virtual Environment. *HAVE 2007 - The 6th IEEE International*

*Workshop on Haptic, Audio and Visual Environments and Games, Proceedings*, pages 21–26, 11 2007. doi: 10.1109/HAVE.2007.4371580.

[28] Lesley A Hall and DI McCloskey. Detections of movements imposed on finger, elbow and shoulder joints. *The journal of physiology*, 335(1):519–533, 1983.

[29] Jia Han, Gordon Waddington, Roger Adams, Judith Anson, and Yu Liu. Assessing proprioception: A critical review of methods. *Journal of Sport and Health Science*, 90, 2 2015. doi: 10.1016/j.jshs.2014.10.004.

[30] Prawit Janwantanakul, Mary Magarey, Mark Jones, and Brenton Dansie. Variation in shoulder position sense at mid and extreme range of motion. *Archives of physical medicine and rehabilitation*, 82:840–4, 07 2001. doi: 10.1053/apmr.2001.21865.

[31] Lynette A Jones. Matching forces: Constant errors and differential thresholds. *Perception*, 18(5):681–687, 1989. doi: 10.1068/p180681. URL https://doi.org/10.1068/p180681. PMID: 2602094.

[32] Bongsu Kim and Ashish D Deshpande. An upper-body rehabilitation exoskeleton Harmony with an anatomical shoulder mechanism: Design, modeling, control, and performance evaluation. *The International Journal of Robotics Research*, 36(4):414–435, 2017. doi: 10.1177/0278364917706743.

[33] H. Kim and A. Asbeck. An elbow exoskeleton for haptic feedback made with a direct drive hobby motor. *HardwareX*, 8:e00153, 2020. doi: https://doi.org/10.1016/j.ohx.2019.e00060.

[34] H. Kim, H. H. Guo, and A. T. Asbeck. Just noticeable differences for joint torque feedback during static poses. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11096–11102, 2020.

[35] Hubert Kim, Hongxu H. Guo, and Alan T. Asbeck. Just noticeable differences for joint torque feedback during static poses. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11096–11102, 2020. doi: 10.1109/ICRA40945.2020.9197537.

[36] Dinant A Kistemaker, Arthur J Knoek Van Soest, Jeremy D Wong, Isaac Kurtzer, and Paul L Gribble. Control of position and movement is simplified by combined muscle spindle and golgi tendon organ feedback. *Journal of neurophysiology*, 109(4):1126–1139, 2013.

[37] Cliff Klein, Charles Rice, and Gregory Marsh. Normalized force, activation, and coactivation in the arm muscles of young and old men. *Journal of applied physiology (Bethesda, Md. : 1985)*, 91:1341–9, 09 2001. doi: 10.1152/jappl.2001.91.3.1341.

[38] Attila J. Kovacs, Jason Boyle, Nicole Grutmatcher, and Charles H. Shea. Coding of online and pre-planned movement sequences. *Acta Psychologica*, 133(2):119–126, 2010. ISSN 0001-6918. doi: https://doi.org/10.1016/j.actpsy.2009.10.007. URL https://www.sciencedirect.com/science/article/pii/S0001691809001541.

[39] R Larsen, H Lund, R Christensen, H Røgind, B Danneskiold-Samsøe, and H. Bliddal. Effect of static stretching of quadriceps and hamstring muscles on knee joint position sense. *Br J Sports Med*, 39, 1 2005. doi: 10.1136/bjsm.2003.011056.

[40] M L Latash and G L Gottlieb. Compliant Characteristics of Single Joints: Preservation of Equifinality with Phasic Reactions. *Biol. Cybern.*, 62(4):331–336, 2 1990. ISSN 0340-1200. doi: 10.1007/BF00201447. URL http://dx.doi.org/10.1007/BF00201447.

[41] Mark L. Latash and Vladimir M. Zatsiorsky. Joint stiffness: Myth or reality? *Human Movement Science*, 12(6):653–692, 12 1993. ISSN 0167-9457. doi: 10.1016/0167-9457(93)90010-M. URL https://www.sciencedirect.com/science/article/pii/016794579390010M.

[42] Ung Hee Lee, Chen-Wen Pan, and Elliott J Rouse. Empirical Characterization of a High-performance Exterior-rotor Type Brushless DC Motor and Drive. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8018–8025. IEEE, 2019.

[43] S. M. Lephart, J. J. Warner, P. A. Borsa, and F. H. Fu. Proprioception of the shoulder joint in healthy, unstable, and surgically repaired shoulders. *Journal of shoulder and elbow surgery*, 3, 11 1994. doi: 10.1016/S1058-2746(09)80022-0.

[44] W A MacKay, D J Crammond, H C Kwan, and J T Murphy. Measurements of human forearm viscoelasticity. *Journal of Biomechanics*, 19(3):231–238, 1986. ISSN 0021-9290. doi: 10.1016/0021-9290(86)90155-7.

[45] Karon Maclean. Emulation of haptic feedback for manual interfaces. 1 2011.

[46] Y Mao, X Jin, G Gera Dutta, J P Scholz, and S K Agrawal. Human Movement Training With a Cable Driven ARm EXoskeleton (CAREX). *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 23(1):84–92, 2015. ISSN 1534-4320 VO - 23. doi: 10.1109/TNSRE.2014.2329018.

[47] L Marchal-Crespo, G Rauter, D Wyss, J von Zitzewitz, and R Riener. Synthesis and control of an assistive robotic tennis trainer. In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 355–360, 2012. ISBN VO -. doi: 10.1109/BioRob.2012.6290262.

[48] Dario Martelli, Federica Vannetti, Mario Cortese, Peppino Tropea, Francesco Giovacchini, Silvestro Micera, Vito Monaco, and Nicola Vitiello. The effects on biomechanics of walking and balance recovery in a novel pelvis exoskeleton during zero-torque control. *Robotica*, 32 (8):1317–1330, 2014.

[49] Marcia O'Malley and Michael Goldfarb. The Effect of Virtual Surface Stiffness on the Haptic

Perception of Detail. *Mechatronics, IEEE/ASME Transactions on*, 9:448–454, 7 2004. doi: 10.1109/TMECH.2004.828625.

[50] Marcia O'Malley and Abhishek Gupta. *Haptic Interfaces*, pages 25–73. 12 2008. ISBN 9780123740175. doi: 10.1016/B978-0-12-374017-5.00002-X.

[51] U Proske and S C Gandevia. The proprioceptive senses: their roles in signaling body shape, body position and movement, and muscle force. *Physiol Rev*, 92(4):1651–1697, 2012. doi: 10.1152/physrev.00048.2011.

[52] Georg Rauter, Roland Sigrist, Robert Riener, and Peter Wolf. Learning of temporal and spatial movement aspects: A comparison of four types of haptic control and concurrent visual feedback. *IEEE Transactions on Haptics*, 8(4):421–433, 2015. doi: 10.1109/TOH.2015.2431686.

[53] Naqash Rehmat, Jie Zuo, Wei Meng, Quan Liu, Sheng Q Xie, and Hui Liang. Upper limb rehabilitation using robotic exoskeleton systems: a systematic review. *International Journal of Intelligent Robotics and Applications*, 2(3):283–295, 2018. ISSN 2366-598X. doi: 10.1007/s41315-018-0064-8.

[54] Naqash Rehmat, Jie Zuo, Wei Meng, Quan Liu, Sheng Q Xie, and Hui Liang. Upper limb rehabilitation using robotic exoskeleton systems: a systematic review. *International Journal of Intelligent Robotics and Applications*, 2(3):283–295, 2018. ISSN 2366-598X. doi: 10.1007/s41315-018-0064-8. URL https://doi.org/10.1007/s41315-018-0064-8.

[55] David Reinkensmeyer and Michael Boninger. Technologies and combination therapies for enhancing movement training for people with a disability. *Journal of neuroengineering and rehabilitation*, 9:17, 03 2012. doi: 10.1186/1743-0003-9-17.

[56] André Schiele and Frans van der Helm. Influence of attachment pressure and kinematic con-

figuration on phri with wearable robots. *Applied Bionics and Biomechanics*, 6:157–173, 07 2009. doi: 10.1080/11762320902879961.

[57] Roland Sigrist, Georg Rauter, Robert Riener, and Peter Wolf. Augmented visual, auditory, haptic, and multimodal feedback in motor learning: A review. *Psychonomic bulletin & review*, 20, 11 2012. doi: 10.3758/s13423-012-0333-8.

[58] Seung□Ki Sul. *Control of Electric Machine Drive Systems*, volume 1 of *1*. Wiley-IEEE Press, 1 edition, 12 2010. ISBN 9780470590799. An optional note.

[59] T-MOTOR, LLC. Antigravity 7005 KV230 Specifications. https://store-en.tmotor.com/goods.php?id=475.

[60] Hong Z. Tan, Brian Eberman, Mandayam A. Srinivasan, and Belinda Cheng. Human factors for the design of force-reflecting haptic interfaces. In *Dynamic Systems and Control Division (Publication) DSC*, volume 55-1, pages 353–359. American Society of Mechanical Engineers, 1994.

[61] *TMS320x2806x Piccolo Technical Reference Manual*. Texas Instruments, 2011. URL https://www.ti.com/lit/pdf/SPRUH18G. Rev. 2017.

[62] *InstaSPIN-FOC™ and InstaSPIN-MOTION™ User's Guide*. Texas Instruments, 2013. URL https://www.ti.com/lit/pdf/SPRUHJ1H. Rev. 2019.

[63] *LAUNCHXL-F28069M Overview User's Guide*. Texas Instruments, 2015. URL https://www.ti.com/lit/pdf/SPRUI11. Rev. 2019.

[64] *Hardware Abstraction Layer (HAL) Module of MotorWare™*. Texas Instruments, 2017. URL https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/171/motorware_5F00_hal_5F00_tutorial.pdf. release in Motorware 18 package.

[65] Neha Thomas, Garrett Ung, Colette McGarvey, and Jeremy D Brown. Comparison of vibrotactile and joint-torque feedback in a myoelectric upper-limb prosthesis. *Journal of NeuroEngineering and Rehabilitation*, 16(1):70, 2019. ISSN 1743-0003. doi: 10.1186/s12984-019-0545-5. URL https://doi.org/10.1186/s12984-019-0545-5.

[66] Nicola Vitiello, Tommaso Lenzi, Stefano Roccella, Stefano Marco Maria De Rossi, Emanuele Cattin, Francesco Giovacchini, Fabrizio Vecchi, and Maria Chiara Carrozza. NEUROExos: A powered elbow exoskeleton for physical rehabilitation. *IEEE Transactions on Robotics*, 29 (1):220–235, 2012.

[67] T Willems, E Witvrouw, J Verstuyft, P Vaes, and De Clercq D. Proprioception and Muscle Strength in Subjects With a History of Ankle Sprains and Chronic Instability. *J Athl Train*, 37, 12 2002.

[68] Linda R Wilson, Simon C Gandevia, and David Burke. Increased resting discharge of human spindle afferents following voluntary contractions. *The Journal of physiology*, 488(3):833–840, 1995.

[69] Carolee Winstein, Patricia Pohl, and Rebecca Lewthwaite. Effects of Physical Guidance and Knowledge of Results on Motor Learning: Support for the Guidance Hypothesis. *Research quarterly for exercise and sport*, 65:316–323, 1 1995. doi: 10.1080/02701367.1994.10607635.

[70] Bing Wu, Roberta L. Klatzky, and Ralph L. Hollis. Force, torque, and stiffness: Interactions in perceptual discrimination. *IEEE Transactions on Haptics*, 4(3):221–228, 2011. doi: 10.1109/TOH.2011.3.

[71] Kazuhiro Yasuda, Yuki Sato, Naoyuki Iimura, and Hiroyasu Iwata. Allocation of attentional resources toward a secondary cognitive task leads to compromised ankle proprioceptive per-

formance in healthy young adults. *Rehabilitation research and practice*, 2014:170304, 01 2014. doi: 10.1155/2014/170304.

[72] Mehrdad Hosseini Zadeh, David Wang, and Eric Kubica. Perception-based lossy haptic compression considerations for velocity-based interactions. *Multimedia Systems*, 13:275–282, 2008. ISSN 09424962. doi: 10.1007/s00530-007-0106-9.

[73] Bohdanna T. Zazulak, Timothy E. Hewett, N. Peter Reeves, Barry Goldberg, and Jacek Cholewicki. The Effects of Core Proprioception on Knee Injury:A Prospective Biomechanical-Epidemiological Study. *The American Journal of Sports Medicine*, 35, 3 2007. doi: 10.1177/0363546506297909.

# Appendix A

# Contributions for HardwareX paper

In this page, we highlight the key contributions of the HardwareX paper.

We provided a custom module that modifies the TI MotorWare Library to make the Brushless DC motor (BLDC) work without hall-sensor readings. We achieved this by using the index-pin variable that stores the predefined position of the offset to be used during an Initial Pose Detection stage. The accurate position measurement for the BLDC is critical in Field Oriented Control (FOC) because the torque is generated based on the electric and magnetic angle. Typically, a hall sensor is used to measure those angles, along with the encoder measuring a mechanical angle. Since the TI solution relies on the strong back-EMF to calculate the electrical angle, applications where the range of motion is limited need an alternative way of calculating the electrical angle. We estimate the mechanical angle via the index pin, where the initial angle for FOC is measured with an optical encoder and its predefined index pin. By calculating the mechanical angle based on the pre-memorized position through the index pin, we use an incremental encoder like an absolute encoder, and we re-calculate the electrical angle based on the mechanical angle (Figure 2.9). The benefit of utilizing this method is that one could use the MotorWare to run BLDC in a small back-EMF environment such as the wearable robot or manipulator joint.

We conducted an analysis of and simulated the motor torque requirements and how to select a motor for different limb joint amplitudes and frequencies. This allows motors for different joints in the body to be easily selected.

We created code scripts for data logging between the Code Composer Studio and Processing have

been open-sourced. TI's C2000 TMS320F28069M microcontroller with DRV8305 motor shield provides a real-time debugging method through Eclipse User Interface but is limited to very few data packets. Using the receipt interrupt feature of the Serial Communication Interface (SCI), we presented the tools for data logging with the adjustability of the transmitting speed and the size of the data packet.

We demonstrated the application of a joint torque feedback elbow exoskeleton, showing how a human arm could be guided by very small torques through an arbitrary trajectory of joint angle versus time. We provided examples of this tracking as a function of different P-gains in the exoskeleton controller.

We produced hardware drawings and directions to fabricate a one degree of freedom joint actuator. The device runs based on a direct-drive mechanism with back-drivability and low latency. This could be useful in applications where a precise output is required, such as haptic feedback or Virtual Reality.