Actuation, Control, and Customization: Key Elements Toward Real-world Deployment of Powered Exoskeleton Systems

by

Ung Hee Lee

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in the University of Michigan
2022

Doctoral Committee:

Professor Elliott J. Rouse, Chair
Professor Robert D. Gregg
Professor Sehoon Ha
Professor Ramanarayan Vasudevan

Ung Hee Lee

unghee@umich.edu

ORCID iD: 0000-0001-5655-9354

Dedication

To my parents, for their love and prayers

Acknowledgments

First and foremost, I would like to thank my advisor Elliott Rouse, whose guidance and expertise made the completion of this work possible. He taught me how to write scientific papers, honed my presentation skills, and deepened my knowledge in the field of wearable robotics – making me a well-rounded engineer and scientist. He was always patient with me, and complimented my achievements. He was supportive of me pursuing my research interest, and provided opportunities to work in industry. Most importantly, I appreciate him allowing me to stay with my family when we moved to Princeton during my final year of study. Without his generosity and support I wouldn't have made it so far.

I would also like to express my sincere gratitude to my committee members, Robert Gregg, Ram Vasudevan, and Sehoon Ha, for their advice and encouragement.

I would also like to thank Jason Martz for providing the opportunity to work with him and stay at UM, and allow me to pursue the area that I'm passionate about when I was transitioning labs.

I am also thankful to my lab members who provided invaluable insights and suggestions for my dissertation. I thank Alejandro Azocar, Yves Nazon, Kim Ingraham and Max Shepherd. Especially, I would like to thank Varun Shetty for his help conducting my final studies while I was working remotely at Princeton.

I am grateful to my friends and colleagues, Woodo Lee, Suhak Lee, Hyungsuk Kwon, Eunjeong Hyun, Yeonjoon Cheong, Tae Gone Jung, Hyung Woo Choi, Jin Young Son, and

Changwoo Lee. In addition, I'd like to thank Il Young Chun, who provided guidance and sincere career advice. They have made my long doctoral journey joyful and provided emotional support when I most needed it.

I would also like to thank Jie Tan, Georgios Evangelopoulos for their guidance during my internship at X. In addition, I would like to thank David Fouhey for advising my last chapter of my dissertation.

Most importantly, I thank my parents Sang Heon Lee, Jung Ja Moon and my sister Jiseon Lee for their love, prayers, and caring for the past seven years. I also thank my in-law family for their love and support. Finally, a huge thank you to my wife, Hyeon Young Ro, and my son, Jay Ung Lee, who are the reason I live and laugh every day.

Preface

Chapter 2 to 5 are written as a separate manuscript; due to this, there may be some overlap on introductions and backgrounds between the chapters.

Table of Contents

List of Tables

List of Figures

Abstract

Lower limb robotics holds a potential to transform human mobility by assisting locomotor activities. Especially, robotic exoskeletons which assist lower limbs by offloading the body from weight, can enable people to walk and run farther with less effort, and carry heavy loads. Despite the promise of wearable technologies, we don't often see these devices in daily life. I argue that there is a gap in how we design and control robotic systems. To close this gap, I focus on three key elements: actuation, control, and customization. First, I characterize a high-performance brushless DC (BLDC) motor, which is an emerging actuation system for lightweight and efficient robots. Second, I expand upon my first work, and provide a tutorial for robot designers who want to utilize these emerging motors for their robotic applications. This includes a consolidated set of governing equations for modeling BLDC motors, and common mistakes from designers when interpreting specifications provided by motor manufacturers. Third, I transition to the control aspect, I describe an online customization framework by optimizing user preference for controlling robotic exoskeleton. Lastly, I present an intent recognition system which predicts users' intent ahead of time, allowing seamless control across multiple activities. With these key elements being addressed, my dissertation lays the groundwork for translating these technologies outside of the lab, in the real world.

Chapter 1

Introduction

Lower-limb exoskeletons have the potential to improve the quality of life by assisting the limbs by offloading the body from the stress and loading associated with the labor [1]. Due to the recent advancement in hardware technology, there has been a proliferation of commercially available lower-limb exoskeletons for the past 20 years [2, 3, 4, 5]. The increase of numbers in exoskeletons are largely driven by the accessibility of electric motors which accounts for 72 % of lower limb exoskeletons [6]. Despite the growth of exoskeletons in the market, we don't see these technologies regularly in our daily life. Assistive or rehabilitation exoskeletons, which are targeted for clinical populations (*e.g.* paraplegic), are often heavy or either fixed on a treadmill [7, 8, 9]. Vast majority of them use servo motors which accompany gear reduction and sensors for precise position measurement [10]. The introduction of gears and sensors results in increase of weight and frictions which reduces the agility while limiting the operation time of the device due to the inefficiency of the system. On the other hand, augmentative lower limb exoskeletons, augments the able-bodied users' activities by following users' movement [11]. There has been innovative lightweight designs introduced in the market such as Honda Walking Assist [12] and Samsung GEMS [5], which employs high torque-density motors (*e.g.* brushless motors); however, they are limited in providing the degree of torque assistance (peak torque of 6Nm and 12Nm for Honda Walking Assist and GEMS, respectively) that might be nec-

essary to support the user. In addition, often the controllers accompanied with commercially available exoskeletons, apply predefined kinematic or kinetic trajectories, which are nonintuitive and may interfere with users intended movement [11]. Therefore, the use of these technologies in daily lives remains limited.

A critical step in the development of lightweight, high performance exoskeletons is the modeling and selection of brushless electric DC motors (BLDC) motors. Due to multiple benefits (*e.g.* reduction in friction, torque density) accompanied with BLDC motors over brushed DC motors, BLDC motors has been gaining popularity as a choice for the lightweight robotic applications [13, 14, 15, 16, 17]. The electromechanical modeling analyses are often conducted as a priori, to ensure the motor is properly sized for the application. In other words, modeling is conducted to ensure the motor meets the torque-velocity requirements of the application while minimizing the mass of the motor. This process is especially important for lightweight robotic systems, where mass dictates the success of the applications. However, the proper assessment of BLDC motors for use in these systems has been challenging due to its obscurity in motor manufacturers' datasheets, from which motor's characteristic are obtained. A BLDC motor datasheet often contains inconsistent representation across different manufacturers, while omitting important information required for analyzing the motors. For example, torque constants are typically reported without any representations described (*e.g.* phase, RMS, q-axis); since BLDC motors have three alternating currents, it is unclear with respect to which currents are used to obtain the constants. For certain BLDC motors, they are application specific (*e.g.* drone motors), which challenges the use in wearable robotics despite their benefits [18]. The confusion is largely stemming from lack of understanding of the governing equations of BLDC Motors and error in converting BLDC characteristics into 'brushed motor' analogue. Hence, proper interpretation of the datasheet and understanding the operation principle of BLDC motors are required to utilize BLDC motors for building efficient and high performance

exoskeletons.

Despite the recent advancement in hardware driven by BLDC motors, developing an intuitive and autonomous controller remains a challenge. Modern control strategies for wearable robots include controller parameters to describe assistance profiles to the wearer [19]. The strategies are comprised of torque control [16, 20, 21], position control [22, 23], and impedance control [24, 25, 26, 27], etc. These parameters should be tuned to account for specific activity or individuals. The choice of parameters are paramount since they alter the gait mechanics, ultimately affecting physiology and experience of the user. In general, there are two approaches to adjust these parameters: 1. expert-driven tuning [28, 29, 30], 2. human-in-the-loop optimization [31, 20, 32, 33]. Expert-driven tuning involves gait experts who observe the walking behavior of the user and adjust parameters to match the kinetic or kinematic profile of natural biomechanics. Human-in-the-loop uses an optimizer which automatically identifies the settings that minimize a physiological objective (*e.g.* metabolic cost). While encouraging, expert-driven tuning requires expert knowledge, and human-in-the-loop optimization requires a certain time and a suite of sensor setup to measure metabolic energy, which are both time and resource consuming. Therefore, it is impractical to use these methods in real-world scenarios. Especially, human-in-the-loop optimization uses a single physiological objective to minimize; where it may not account for the myriad of factors that contribute to users while performing daily activities. For example, when individuals are going down the stairs, they may prioritize multiple objectives at the same time such as balance, walking speed and comfort etc. One way to overcome the conventional methods is to tune parameters based on *what people want*, using user preference. The user-preference-based tuning can attain more than one objective while not relying on expert knowledge nor expensive sensors, which allows optimizing controller parameters outside of laboratories.

Recently, studies on using user preference have been emerging as a strategy for con-

trolling wearable robots. One line of work investigates approaches where the user self-explores the controller space (*i.e.* user-driven, self-tuning methods) [34, 35, 36]. While promising, these tuning methods require the user to keep track of their exploration to identify the settings, and every time they have to go through the procedure from the start. In other words, it is unlikely users would remember the settings they explored from the other day when they tune the settings. In addition, these approaches might not scale up with the dimensions of the parameters. Another line of work studies tuning controllers automatically based on preferences [37, 38, 39]. These studies used Bayesian optimization by modeling a utility function using Gaussian Process and sample control parameters to maximize the function. To validate their system, they applied it to Altlante, a full lower body exoskeleton developed by Wandercraft combined with a controller stemming from the bipedal robot community. As the robot is intended for clinical populations, it drives and overrides the user's movement. Therefore, it is unclear how this approach performs on identifying users preference with augmentative robots where the controller has a symbiotic relationship with the users.

In large, the goal of exoskeleton is to build a controller which does not interfere with the intended movement of the wearer while assisting the intended movement. In addition, to enable the use of exoskeletons in daily lives, the controller should account for multiple activities while transitioning to these activities seamlessly and autonomously. Many applications enable these activity transitions by using external commands from the users such as key-fob cues, which are unintuitive and induce cognitive burden [40]. Thus, it is crucial to develop a controller which infers the wearer's intention in advance and acts autonomously so that the wearer can perform and transition to different activities in a seamless manner. One approach to infer the wearer's intended activity is to use an *intent recognition framework* [41, 42, 43, 44]. The approach predicts the upcoming activities of the user in every walking step using sensor information from the user, robotic system, and

environment with data-driven modeling and sensor fusion. Intent recognition system is used as a high-level controller where it assigns activity-specific instruction to mid-level controllers for how to provide mechanical effort. While promising, the activity-specific controller assigned from the intent recognition system typically assigns a set of generic control parameters independent from users. In other words, the assistance from the exoskeleton might not be optimal for individuals. Therefore, connecting the controllers with mid-level controllers that personalize and optimize to users should be accompanied as a holistic controller design.

While numerous designs and control of exoskeletons have been proposed, there is *a gap* in how we design and control these systems efficiently and translate them into the real-world. The main objective of my proposal is to close this gap by *modeling and optimizing the lower-limb exoskeleton from actuator to human level*. The objective will be achieved by following aims: characterization and modeling of BLDC motors (Chapter 2 and 3), preference-in-the-loop optimization (Chapter 4), development of intent-recognition system (Chapter 5). I believe these three aims lay the groundwork for the core elements of exoskeletons: actuation, control and customization, and will translate exoskeleton technologies in everyday lives.

Chapter 2

Empirical Characterization of a High-performance Exterior-rotor Type Brushless DC Motor and Drive

2.1 Abstract

Recently, brushless motors with especially high torque densities have been developed for applications in autonomous aerial vehicles (*i.e.* drones), which usually employ exterior rotor-type geometries (ER-BLDC motors). These motors are promising for other applications, such as humanoids and wearable robots; however, the emerging companies that produce motors for drone applications do not typically provide adequate technical specifications that would permit their general use across robotics—for example, the specifications are often tested in unrealistic forced convection environments, or are drone-specific, such as thrust efficiency. Furthermore, the high magnetic pole count in many ER-BLDC motors restricts the brushless drives able to efficiently commutate these motors at speeds

This chapter is currently published in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

needed for lightly-geared operation. This paper provides an empirical characterization of a popular ER-BLDC motor and a new brushless drive, which includes efficiencies of the motor across different power regimes, identification of the motor transfer function coefficients, thermal response properties, and closed loop control performance in the time and frequency domains. The intent of this work is to serve as a benchmark and reference for other researchers seeking to utilize these exciting and emerging motor geometries.

2.2 Introduction

The usage of electric motors in autonomous aerial vehicles (*i.e.* drones) has advanced the proliferation of new brushless motor sizes and geometries that may have important implications in other areas of robotics. These motor geometries were developed to provide the high torque required for drone applications, without the need for a transmission. Transmissions are often undesirable because they universally add weight and audible noise, as well as reduce the system's efficiency and specific power—concepts which are especially important for drones and other applications, such as wearable robots. Consequently, motor manufacturers have expanded the availability of high torque exterior-rotor type (ER-) BLDC motors, which can be used in direct drive implementations for autonomous aerial vehicles. These ER-BLDC motors are usually larger in diameter, ranging from $2 \sim 11$ cm in diameter, while only being $1 \sim 2$ cm in axial length. The large diameter stems from the substantial torque that can be required, as motor torque is proportional to the square of the radius.

The development and usage of ER-BLDC motors in the drone industry have outpaced rigorous electromechanical characterization of the motors and associated drives. These motors have gained popularity due to their low cost and high performance; however, obtaining specifications for these motors can be challenging. With the exception of a few re-

search groups that study the implementation of ER-BLDC motors in robotic applications [13, 15, 45], the motor specifications that can be obtained are often sparse, result from unrealistic testing conditions, or in units that are specific to drone applications (*e.g.* thrust efficiency). Furthermore, ER-BLDC motors utilized in drone applications implement open-loop commutation systems, known as 'electronic speed controllers' (ESCs). These ESCs cannot operate at low speeds, and thus are not appropriate for wearable robotics tasks (*e.g.* locomotion). In addition, many common commercially-available BLDC drives cannot efficiently commutate some ER-BLDC motors, and more specialized options must be used. The high number of pole pairs causes an increase in the transition rate between motor windings, which can exceed the computational limitations of modern BLDC drives [46]. Fortunately, new drive systems have been developed specifically to commutate the high number of pole pairs in some ER-BLDC motors at speeds required for some applications [47]. However, a rigorous, empirical characterization of these motors and drives is yet to be completed.

In this paper, we performed an empirical electromechanical characterization of a new ER-BLDC motor (T-motor U8-KV100, Nanchang, Jiangxi, China), originally created for the drone industry. In addition, our characterization provides insight into the performance of a recently-developed brushless drive (model: Dephy Actuator Package (DAP), Dephy Inc., Maynard, MA, USA) capable of efficiently commutating the high number of pole pairs (N=21). The main contributions of this paper are the complete characterization of a popular ER-BLDC motor and accompanying drive, which will serve as a benchmark for use of these motors in wearable robotic applications; the characterization included identification of the motor's transfer function coefficients (*e.g.* torque / back-EMF constants, rotor inertia, and viscous damping parameters), the empirically-determined efficiencies across a range of torque-speed regimes, the thermal response properties, and the closed-loop control performance in the time and frequency domains.

2.3 Characterization Methods

This study focuses on the empirical characterization of a popular ER-BLDC motor that has had recent success in autonomous robotic applications, namely the T-motor U8-KV100 (Nanchang, Jiangxi, China) [25, 48, 15, 49, 50, 51]. To this end, we investigated two versions of the U8-KV100; the first version was the unmodified U8-KV100, and the second version included a custom modified housing, which facilities use in our application area of interest (*e.g.* wearable robotics). The modified U8-KV100 was designed by Dephy Inc., and is slightly lighter (230 g vs. 240 g) and substantially thinner (17 mm vs. 26 mm) than the standard U8-KV100. Both motor versions have a diameter of 87 mm and the electromagnetic properties of the motors are identical. The brushless drive used in this study was also developed by Dephy Inc., and is based on the open-source Flexible, Scalable Electronics Architecture (FlexSEA) project developed for wearable robotics applications [47]. The custom modified U8-KV100 motor, coupled to a commercial version of the FlexSEA brushless drive, are known together as the *Dephy Actuator Package* (*DAP*). The DAP enables field oriented control of the 21 pole pairs at speeds relevant for wearable robotics ($\sim$300 rad/s) and uses a PWM frequency of 20 kHz. The DAP includes built-in low level control, including position, current, and impedance control options. Within the drive, position and current feedback loops are closed at 1 kHz and 10 kHz respectively. Onboard sensing includes electrical states of the motor windings and bus, as well as a nine-axis inertial measurement unit (model: MPU-9250, InvenSense, San Jose, CA, USA), temperature sensing, and a 14-bit absolute encoder (model: AS5047P ams AG, Premstaetten, Austria). Motor line to line currents are measured via a 12-bit current sensor within the drive and line to line voltages are provided from the drive, obtained by multiplying PWM duty cycle by bus voltage. Depending on the application, peripheral electrical hardware can communicate with the DAP via Bluetooth, Universal Serial Bus (USB), Serial Peripheral Interface

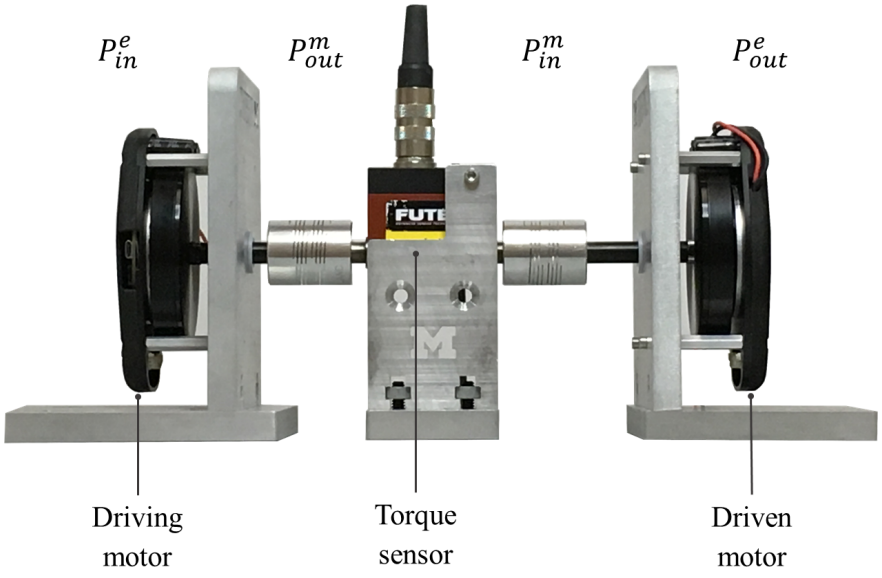

Figure 2.1: Experimental setup for characterization of the ER-BLDC motor and drive. Two identical motors were coupled through a common shaft which included a torque sensor. During operation, the driving motor converted electrical power (P^e_{in}) into mechanical power (P^m_{out}), while the driven motor converted mechanical power (P^m_{in}) into electrical power (P^e_{out}).

(SPI), Inter-Integrated Circuit (I^2C), and RS-485. The DAP can be used with higher-level control systems to select control type, set controller gains, provide reference signals, and acquire data [47, 25]. Detailed specifications of the motors are described in Table 3.2.

In this paper, the electrical state of the windings is represented using the brushed DC electromechanical model for convenience and simplicity. That is, for our analysis, we converted the 3-phase sinusoidal winding currents and voltages to fictitious direct current and voltage (*i.e.* quadrature quantities). Using the direct-quadrature transformation[52], we report values in terms of the q-axis, which rotates along the rotor shaft. For a sinusoidally wound BLDC motor, this conversion is equivalent to the root-mean-square of the 3-phase currents, and enables discussion of electrical states without including sinusoidally varying voltages and currents.

2.3.1 Efficiency

The efficiency of a motor and drive can be especially important for many applications. Since both power and loss scale with current and velocity in the electrical and mechanical domains, respectively, motor efficiency is dependent on the specific current-voltage regimes of operation (or equivalently, torque-speed regime). To this end, we developed an experimental dynamometer to assess motor and drive efficiency. Two ER-BLDC motors (DAPs) were coupled through a common rotational shaft. A contactless torque sensor was secured within the common rotational shaft (model: TRS605, Futek Advanced Sensor Technology, Inc., Irvine, CA, USA) (Fig. 2.1). The analog voltage from the torque sensor was acquired by a 16-bit analog to digital converter (model: ADS 1115, Texas Instruments, Dallas, TX, USA), which was recorded using a microcomputer (model: Raspberry Pi 3 Model B, Raspberry Pi Foundation, Cambridge, UK). The microcomputer also communicated via USB to the ER-BLDC drives, acquiring digitized data regarding motor electromechanical state, angular velocity, bus current, winding current, bus voltage, and winding voltage were obtained at 320 Hz, and stored with digitized torque data for post processing. Both motors were powered by a common 36 V lithium polymer (LiPo) battery with a capacity of 950 mAH (model: 25087X2, Venom Power, Rathdrum, ID, USA). In this experiment, one motor operated in positive power (driving motor), while the other motor provided a resistance, operating in negative power (driven motor). The use of two identical motors (driving and driven) enabled quantification of efficiency for two (nearly) opposite current-voltage conditions for a given current-voltage test. The equations governing the steady state operation of coupled motors are described as follows, converted to the quadrature-based brushed electromechanical model. Applying Newton's Second's

Law yields:

$$\text{Driving: } \cancel{J\ddot{\theta}} = k_t i_+ - b\dot{\theta} - \tau_L \tag{2.1}$$

$$\text{Driven: } \cancel{J\ddot{\theta}} = -k_t i_- - b\dot{\theta} + \tau_L \tag{2.2}$$

where i_+ and i_- are the q-axis winding currents of driving and driven motor, respectively; J is the rotor's moment of inertia; $\dot{\theta}$ is the angular velocity; $\ddot{\theta}$ is the angular acceleration; k_t is the q-axis torque constant; τ_L is the load torque, and b is the viscous damping coefficient. $J\ddot{\theta}$ are neglected, since both motors are steady state (*i.e.* constant velocity).

Using the traditional brushed electromechanical model, winding voltage can be described by applying Kirchoff's Voltage Law:

$$\text{Driving: } v_+ = k_b\dot{\theta} + i_+ R + \cancel{L\frac{di}{dt}} \tag{2.3}$$

$$\text{Driven: } v_- = -k_b\dot{\theta} + i_- R + \cancel{L\frac{di}{dt}} \tag{2.4}$$

where v_+ and v_- are the q-axis winding voltages applied to driving and driven motor, respectively, k_b is the back-EMF constant, and R is the phase resistance of the windings. Inductances ($L\frac{di}{dt}$) are omitted as the motors are at steady state and the q-axis current is a constant value. Finally, it should be noted that the motors were placed face-to-face; thus, the angular velocities had opposite signs, while the torque had a consistent sign.

Efficiencies were investigated across a range of current-voltage conditions. In general, the driving motor was operated at a fixed, open-loop q-axis voltage across the windings that varied from 0 $\sim$ 39.2 V at 2.4 V intervals. In contrast, the driven motor was generally operated in closed-loop current control, with desired q-axis currents ranged from 0 $\sim$ 19.8 A at 0.8 A intervals. Tested current-voltage conditions were required to satisfy the inequality: $v + 3.2i \leq 80.8$ where v, i are commanded voltage and current, respectively.

In total, 678 combinations of voltage and current sets were tested. For each combination of current and voltage, applied motor voltage was increased linearly for one second, with one second of steady state behavior before the voltage was decreased linearly to zero over an additional one second. Each test was a total a three seconds in duration. Three second duration was chosen to balance the time required to safely reach steady state, while limiting the thermal rise in winding temperature.

To quantify motor power and efficiency, the data were analyzed in MATLAB (The MathWorks, Inc., Natick, MA, USA). The efficiency of driving motor was calculated as:

$$\eta_{+} = \frac{\tau_L \dot{\theta}}{i_{+}^{b} v_{+}^{b}} \tag{2.5}$$

where i_{+}^{b} is the bus current, v_{+}^{b} is the bus voltage, and their product $i_{+}^{b} \cdot v_{+}^{b}$ is the input electrical power, averaged across the one second interval at steady state. The bus power represents the electrical state of the power source (*i.e.* batteries). Similarly, we calculated the mechanical power as $\tau_L \cdot \dot{\theta}$, also averaged across the one second interval. The efficiency of driven motor was quantified as:

$$\eta_{-} = \frac{i_{-}^{b} v_{-}^{b}}{\tau_L \dot{\theta}} \tag{2.6}$$

which is the reciprocal of the driving motor's efficiency, where the motor is driven by the mechanical power from the driving motor, generating electrical power ($i_{-}^{b} \cdot v_{-}^{b}$). Any condition where the angular velocity was less than 0.1 rad/s was considered a stall condition, and was not analyzed.

2.3.2 System Identification

Torque / Back-EMF constant

A motor's torque (or back-EMF) constant scales the conversion from electrical power to mechanical power of the motor. We completed two tests to obtain estimates of the q-axis torque constant and back-EMF constant separately. To quantify torque constant, the motor's output shaft was mounted to the torque sensor, which was mechanically grounded (*i.e.* in a stall condition). Current commands were provided to the drive by the microcomputer. Currents were tested from -5 to 5 A at 1.0 A intervals. Each trial had a duration of nine seconds following the onset of torque; first and last one second were removed from analysis to ensure steady state behavior. Five trials were repeated at each current level. Data collected by the microcomputer included desired winding current, measured winding current, and measured torque at a sample rate of 250 Hz. The motor torque and current during each trial were averaged across the seven second trial duration. Subsequently, the data from all trials were averaged to obtain a single current-torque value for each current condition. The torque constant was obtained by linear regression between the average measured torques and average measured winding currents. Goodness of fit was determined by the coefficient of determination (*i.e.* R-squared).

Similarly, to quantify the back-EMF constant, the motor's angular velocity was measured as a function of applied winding voltages. The motor's rotor was able to spin freely, operating at the no-load speed for each applied voltage. To calculate the back EMF voltage, we subtracted the voltage drop across the winding resistance ($i \cdot R$) from the applied voltage, the difference of which is the back-EMF (Eq. 2.3, 2.4) during steady state conditions. The motor was operated at a fixed, open-loop voltage across the windings, tested between -22 and 22 V at 5.4 V intervals. Data collection protocol and post-processing procedures

were identical to characterization of the torque constant.

Inertia and Viscous Loss

The rotor inertia and viscous loss properties have important effects on the dynamic response, especially for ER-BLDC motors, which often have greater rotor inertia values. In this experiment, the two coupled motors interacted, and the interaction dynamics were used to estimate the inertia and viscous loss of the motor. The driven motor had the motor leads open with no control commands present, and thus, acted as a pure inertia and viscous load. To estimate motor properties, we used a non-parametric system identification technique implemented in the frequency domain. The reference position trajectory was provided by the microprocessor, and was a Gaussian random process, low-pass filtered using a third order Butterworth filter with a cutoff frequency of 40 Hz, and was scaled to a peak value of $\pm$ 180°. Five trials were conducted, each with a duration of 60 seconds, with data collected at a sample rate of 250 Hz. Following data collection, MATLAB was used to perform the system identification analyses with an input of load torque and output of angular velocity. The frequency response was determined using Blackman-Tukey spectral analysis—where the auto-spectrum and cross-spectrum are divided in the frequency domain [53], with a window length of 100 samples and an overlap of zero samples. Following estimation of the frequency response, a first-order model was fit to the frequency response data:

$$\frac{\dot{\Theta}(s)}{T_L(s)} = \frac{1}{Js + b} \tag{2.7}$$

The sum-squared error between the magnitude of the second-order model and the non-parametric frequency response was calculated in the range of logarithmically spaced frequencies where coherence was greater than 0.85. Coherence values closer to 1.0 indicate linearity between the input and output, supporting our choice to use a linear model across

the range of frequencies tested. The parameters for the first-order model (inertia and damping) were chosen that minimize the corresponding error in the frequency response. A single set of inertia and damping values were obtained by averaging the optimized parameters across all trials. The estimated parameters were used to compute the output (angular velocity) of a separate validation dataset, and variance accounted for (VAF) was used to estimate model agreement. Lastly, we repeated the same experiment without the driven motor, thereby measuring any inertia and damping associated with the torque sensor and driving shafts. The inertia and damping of the sensor and shafts were removed from the values obtained using both motors.

Additionally, damping coefficient was measured using a separate experiment investigating the relationship between angular velocity and winding current. Voltages were commanded from -22 to 22 V at 5.4 V intervals to the drive, which corresponded to -160 rad/s to 160 rad/s at 27 rad/s intervals. The current and angular velocity for each set were obtained by averaging the measured current and velocity during seven second duration, and averaged again across the five trials for each set.

2.3.3 Thermal properties

Joule heating of the motor's windings is among the chief limitations of system performance. The flow of heat flux to the environment is governed by a combination of convection and conduction—first order ordinary differential equations (ODEs) that can be modeled in the electrical domain (Fig. 2.2). To this end, we conducted experiments to quantify the thermal properties of the windings and housing structures of the DAP and the T-motor KV100 separately. We applied 8 A of DC current across two of the three ER-BLDC leads. Data were recorded for 65 minutes, which was known to enable the system to reach a steady state temperature. Two approaches were used to determine the thermal proper-

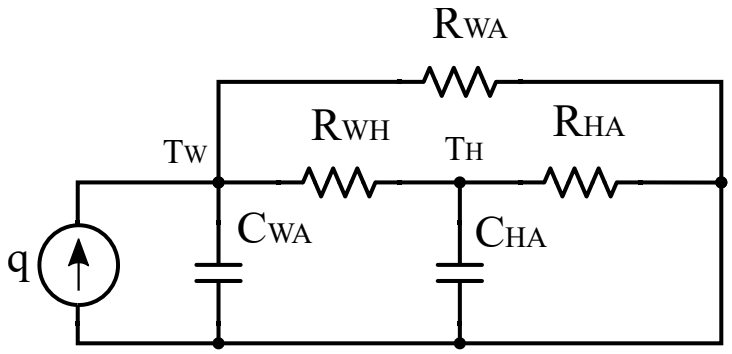

Figure 2.2: Analogous electrical circuit of the thermal modeling comprised of thermal resistances and capacitances.

ties, differing in how the temperature of the windings were obtained. First, temperatures were directly acquired by infrared temperature camera (model: FLIR One Pro, Flir System Inc., Wilsonville OR, USA). The temperature of the windings and housing were measured and recorded every minute (1/60 Hz) in software. The overall winding temperature was a weighted sum of the two different measured winding temperatures. Second, we recorded the current and voltage applied to the motor using a 16-bit data acquisition system (model: USB-6218, National Instruments Corporation, Austin, TX, USA) sampling at 580 Hz. Measured currents and voltages were used to compute the instantaneous resistance, which is related to the temperature via temperature coefficient of resistivity (TCR)[54]. Thermal resistances and capacitances were obtained by fitting parameters of the first order ODEs relating applied heat flux to temperature rise. Thermal impedances of each motor were obtained by Newton's law of cooling:

$$\frac{Q(s)}{\Delta T(s)} = \frac{1}{R_{WA}} + \frac{1}{R_{WH} + \frac{R_{HA}}{C_{HA}R_{HA}s+1}} + C_{WA}s \tag{2.8}$$

where Q is the heat flux, R_{WH}, R_{HA}, R_{WA} are thermal resistances of the winding to housing (WH), housing to ambient (HA) and winding to ambient (WA), respectively and

When current passes between two leads in a delta winding configuration, two windings experience half the current of the third winding; thus, the total heat generated can be calculated as a weighted sum.

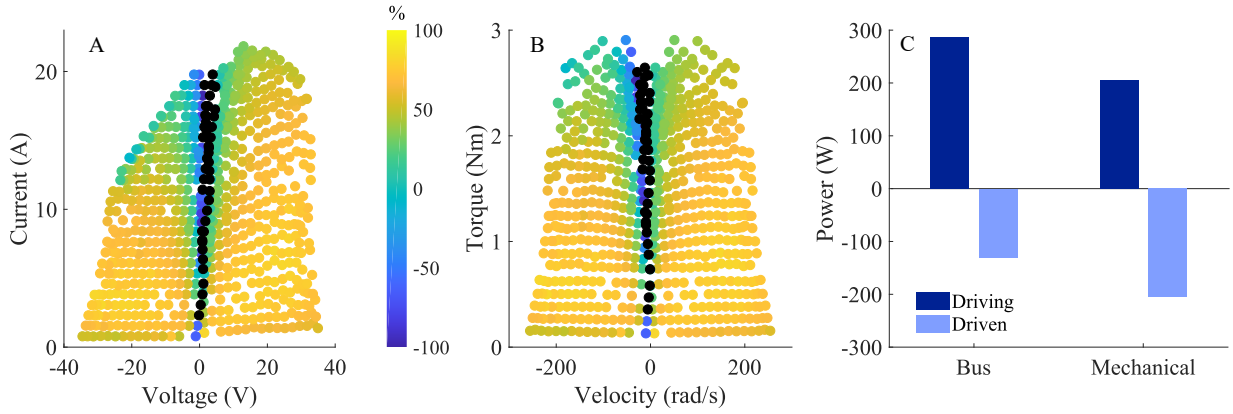

Figure 2.3: Total efficiencies (bus electrical power in; mechanical power out) of the actuator in different current and voltage (a) and torque and velocity conditions (b). The yellow color spectrum represents positive efficiency and the blue spectrum shows negative efficiencies. Negative efficiency occurs when the voltage across the winding is larger than back-EMF, and only occurs during negative mechanical power. Efficiencies less than -100% are depicted in black dots. (c) Plot depicts electrical power of the bus (c, left) and mechanical power (c, right) of the motor for the 15.8 A and 24.5 V condition. During each current-voltage condition, one motor operated in positive mechanical power, while the other provided resistance, operating in negative mechanical power. The relationships between the input and output powers provided the ability to assess total efficiency across a range of torque-speed combinations. Note that the mechanical power coupling the two motors is identical.

C_{WA}, C_{HA} are capacitances of the winding to ambient and housing to ambient, respectively. We used a genetic algorithm-based optimization routine to minimize the sum-squared error between the measured temperatures acquired by the thermal camera and the temperatures governed by the ODEs from the thermal circuit. The parameters (thermal resistances and capacitances) of the second-order model were chosen that minimizes the error. The steady state temperatures of the windings and housing were obtained by multiplying the heat flux (i^2R) by the relevant thermal resistance. Note that the heat flux changes as function of time, because the electrical resistance of the windings increases as the temperature rises.

2.3.4 Closed-loop Control

The drive within the DAP includes on-board closed loop control for position and current, the performance of which were assessed in this section. For the position control tests, the motor was able to spin freely; in contrast, for the current control tests, the motors rotor was grounded (as in the previous experiments determined the back-EMF and torque constants). Reference trajectories were provided by the microcomputer at a rate of approximately 725 Hz over USB to the drive.

We conducted step response tests to quantify the closed-loop controller performance in the time domain. Reference step inputs were provided at 180°, 360°, and 720° for the position control tests, and 3.3 A, 6.6 A, and 9.9 A for the current control tests (which approximately corresponds to 0.5 Nm, 1 Nm, 1.5 Nm respectively). Each trial had a duration of three seconds and five trials were repeated at each step reference level. Rise time, overshoot, and steady-state error were calculated to quantify controller performance. Subsequently, these values were averaged across all trials and reference levels to obtain single value for each parameter for both the position and current controllers.

We conducted frequency response tests to quantify the closed-loop controller performance in the frequency domain. A Gaussian random process was low-pass filtered using a third order Butterworth filter, which provided the time-varying reference input trajectories. The cutoff frequency of the low-pass filter was 70 Hz and 350 Hz for the position and current control, respectively. The reference trajectories were then scaled to a peak value of 180°, 360°, and 720° for the position control tests; 3.3 A, 6.6 A, and 9.9 A for the current control tests. All trials had a duration of 10 seconds, and five trials were repeated at each reference level. The frequency response was determined using Blackman-Tukey spectral analysis with a window length of 1000 samples. Bandwidth was determined by the frequency where the magnitude crossed -3 dB. Bandwidth values were averaged across all

trials and reference levels to obtain single value for the position and current controllers.

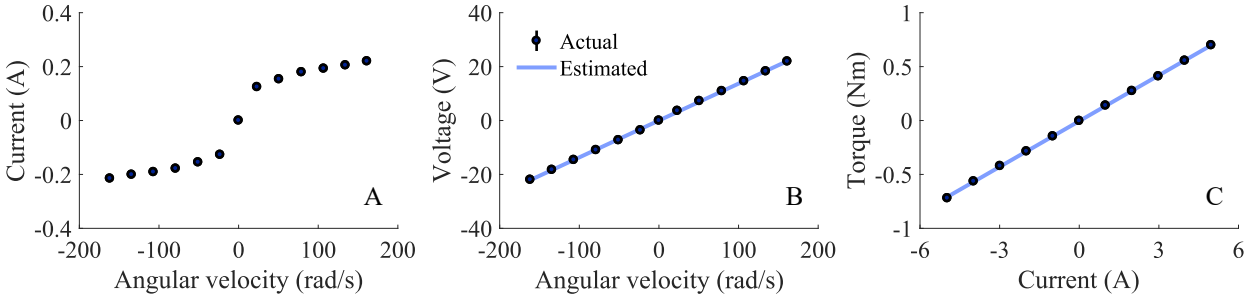

Figure 2.4: Plots of no-load current (a), voltage-speed (b), torque-current (c) relationships. The slope of the voltage-speed relationship is the back EMF constant, and the slope of the torque-current relationship is the torque constant. The standard deviations of angular velocities, winding currents and voltage are depicted vertically. Standard deviations for all lines are not visible as their errors are smaller than the scale of both axes. Lines of best fit are shown.

2.4 Results

2.4.1 Efficiency

Efficiencies of the ER-BLDC motor were calculated across the current-voltage and torque-speed regimes (Fig. 2.3a, b), with a representative trial shown in Fig. 2.3c. In general, efficiencies during positive power (quadrant one, >0 V, >0 rad/s) are all greater than zero and over 65%, except during higher current ($>\sim$20 A, $>\sim$2.8 Nm) conditions. During negative power (quadrant two, <0 V, <0 rad/s), both positive and negative efficiencies were observed. Low and negative efficiencies were more prominent at higher current conditions ($>\sim$13 A, $>$1.8 Nm), but were present across all tested currents at low velocity ($\sim$ 0V). The mean value of positive efficiencies across all conditions was 56 ± 20 %.

2.4.2 System Identification

Torque / Back-EMF constant

We observed the characteristic linear relationships between torque and q-axis current, as well as q-axis voltage and angular velocity (Fig. 2.4b, c). The slope of the line of best fit for the torque-current data (*i.e.* torque constant) was 0.141 Nm/A with 95% confidence interval (C.I.) of [0.140, 0.142] Nm/A. The R-squared value for the line of best fit was 0.999. The slope of the line of best fit for the voltage-velocity data (*i.e.* back-EMF constant) was 0.137 V/rad/s (95% C.I.: of [0.135, 0.138] V/rad/s). The R-squared value for the line of best fit was 0.999.

Inertia and Viscous Loss

The optimized values of inertia and damping coefficients of the second order model across all trials (Eq. 2.7) were $1.21{\cdot}10^{-4} \pm 1.95{\cdot}10^{-6}$ kg$\cdot m^2$ and 2.6 $\pm$ 0.35 mNm/rad/s, respectively. The data fit the model well, with a VAF of 83%. When the model was tested with a separate, validation input, the VAF was also 83%.

The damping property of the motor was also investigated by analyzing the current-angular velocity relationship at no-load speed (Fig. 2.4a). When linear regression was used to estimate the current as a function of positive velocity, the slope was 1.12 mA/rad/s (95% C.I.: [0.41, 1.82] mA/rad/s) and the offset was 65 mA. Using the torque constant and representing these data in terms of torque, the slope was 0.16 mNm/rad/s (95% C.I.: [0.06, 0.26] mNm/rad/s) and the offset was 9.1 mNm. The y-axis intercept indicates that the damping behavior of the motor is nonlinear. The R-squared value for the line of best fit was 0.77.

2.4.3 Thermal properties

The thermal responses of both the T-motor (Fig. 2.5a) and DAP (Fig. 2.5b) show steep rise in temperature initially, and gradually levels off as they reach steady-state. Measured temperatures of the housing were lower than the winding for both T-motor and the DAP. Temperatures estimated by the ODE (Eq. 2.8) showed a better agreement with the thermal image data than temperatures estimated by the TCR. The steady state temperature of the DAP ($96.9°$C) was lower than that of the motor ($102.7°$C). This discrepancy agrees with the fact that the additional structural mass (7075-T6 Aluminum) of the DAP acted as a heat sink, lowering the thermal resistance.

2.4.4 Closed-loop Control

Step responses of the current and position feedback control (first row of Fig. 2.6) show accurate and persistent tracking performance (steady state error $< 0.6\%$) across all reference inputs in time domain. For the position control, the average rise time and settling time across all trials were 22.2 ± 6.1 ms and 60.3 ± 14.8 ms, respectively; for the current control, the rise time and settling time were 5.0 ± 0.7 ms and 30.2 ± 16.9 ms, respectively. Overall, the settling times of both controllers increase as the amplitude of reference step inputs increase. Note that the rise and settling time of the current controller were less than that of the position controller.

Bode plots for magnitude and phase (second and third rows of Fig. 2.6) shows varying frequency response characteristics of the closed-loop systems at different reference inputs. In general, for the position control, the bandwidths were approximately $15 \sim 20$ Hz; for the current control, they were greater than 320Hz.

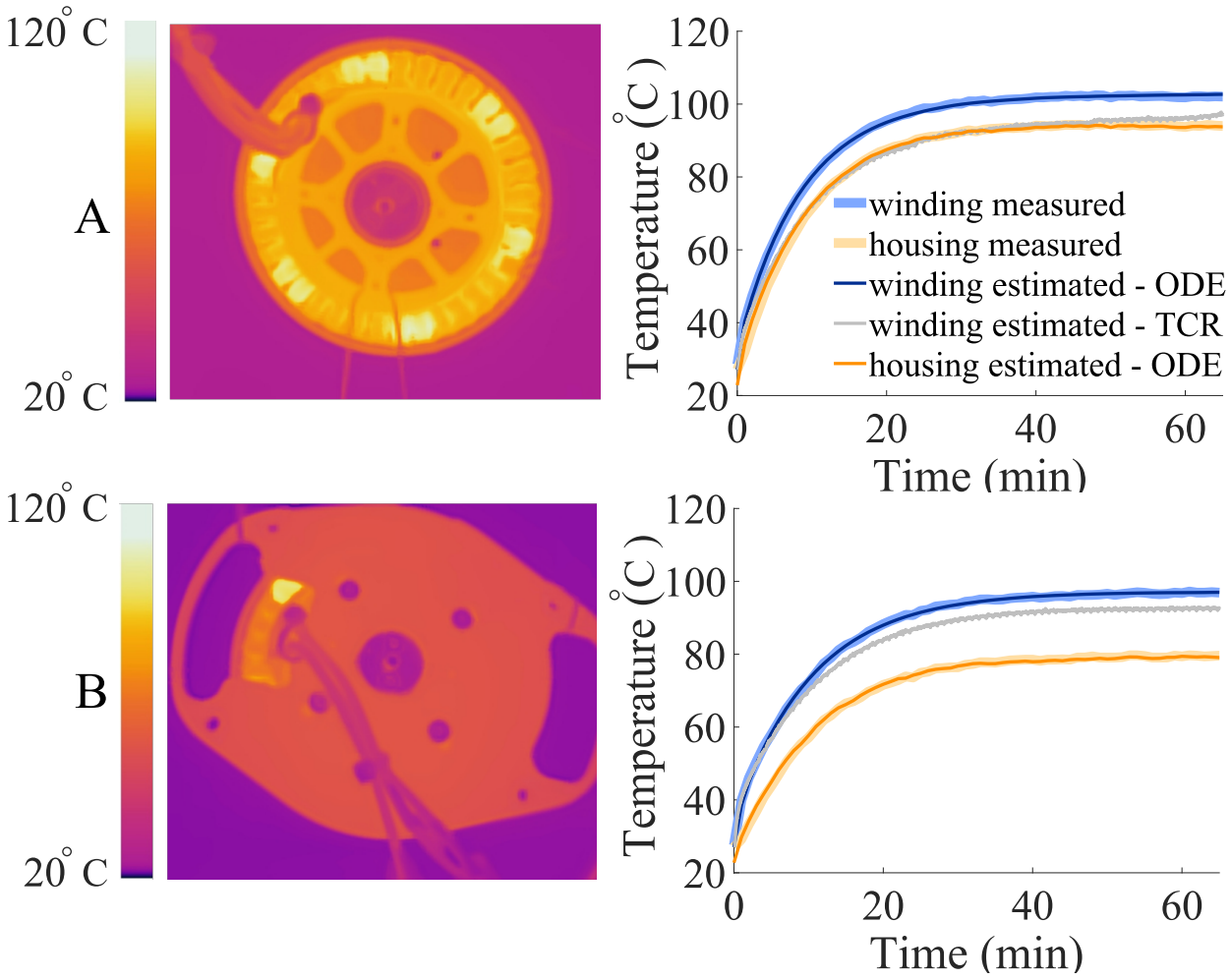

Figure 2.5: Infrared thermal image (left) and thermal responses (right) of the T-motor (a) and DAP (b) at steady state. The measured temperatures using the thermal imaging camera; estimated temperatures of the winding and housing using the thermal model (Fig. 2.2) and TCR are depicted.

2.5 Discussion

In this paper, we quantified the time and frequency domain characteristics of a high-performance ER-BLDC motor and the closed-loop control performance of an accompanying brushless drive. The motivation of this paper is to provide accurate and detailed specifications of a common ER-BLDC motor, the parameters of which are often unreported or not generalizable beyond drone applications. We characterized the motor's transfer func-

tion, efficiency across torque-speed regimes, as well as thermal properties. Accurate and detailed motor specifications are increasingly important, especially as ER-BLDC motors developed for the drone industry are more commonly used in general robotic applications.

2.5.1 Efficiency

The efficiency pattern across torque-speed and current-voltage regimes were asymmetric about the origin (Fig. 2.3). For the current-voltage representation, the asymmetry stems, in part, from the x-axis representation of total voltage across the windings, rather than the back-EMF. Thus, for a fixed voltage magnitude and current—that is, two points on either side of the y-axis—the velocities associated with these current $\pm$voltage pairs are different (but the voltage drop across the resistance is identical); this mismatch in velocity causes substantially different viscous losses on either side of the current-voltage regions. This also accounts for the right-slanted lean in the negative power quadrant of the current-voltage efficiency plot. Each combination tested at a fixed current has a winding voltage difference of $2k_b\dot{\theta}$ (Eq. 2.3, 2.4). For the torque-speed representation, there is also some asymmetry about the origin. This asymmetry is likely due to the differences in efficiencies of the internal processes within the drive. The conversion of bus voltage to winding voltage is more efficient when the system is operating in positive electrical power, as compared to negative electrical power.

The total efficiency was often less than zero for negative power regimes (this can be observed by the blue spectrum of colors in the efficiency plot (Fig. 2.3a-b, quadrant two)). Efficiencies less than zero occur in negative power conditions when the voltage drop across the winding resistance is larger than that of the driven motor's back-EMF voltage. In other words, this occurs when the winding current is larger than the maximum current induced

from electromagnetic induction:

$$i_{-} > i_{max} = \frac{k_b \dot{\theta}}{R} \tag{2.9}$$

where the induced maximum current (i_{max}) is defined as induced current when the winding voltage is zero (*i.e.* leads are shorted). Alternatively, in terms of electrical power, the power loss across the winding resistance is greater than the mechanical power input. Thus, positive electrical power is required from the power source, leading to an efficiency less than zero. In certain cases, the negative efficiency may be $<$ -100%. This occurs when the power loss across the winding resistance is more than twice the mechanical power input.

2.5.2 System Identification

Torque / Back-EMF constant

In theory, the torque and back-EMF constants are equivalent in the brushed DC model. In our characterization, there was a slight discrepancy (3% error) between these values. This discrepancy may be explained by practical considerations, including that the motor may not be perfectly sinusoidally wounded (*i.e.* has slight variation in amplitude of phase back-EMF profile), in addition to any noise in the experimental measurements.

Viscous Loss

Nonlinear friction was observed in current-velocity relationship when the motor was spinning freely (Fig. 2.4a). This nonlinearity (current-velocity values near the origin) indicates the presence of coulomb friction, with an offset of 65 mA (9.1 mNm). For velocities greater than zero, a constant slope was observed, which indicates a linear model may not fit the data near zero velocity. We used a linear, time invariant approach to estimate the

inertia and damping properties separately, and showed strong agreement between output velocity and model-predicted values (VAF: 83%). This agreement indicates that inertia dominated the torque response, with a limited contribution from the damping component. Using the input velocity and estimated inertia and damping values, the damping component contributed approximately 19% of the output torque and the inertia component contributed approximately 81% of the output torque. Thus, although the damping behavior was nonlinear, the usage of linear methods did not likely affect the analysis.

2.5.3 Thermal properties

Ultimately, a motor's peak performance is limited by its ability to dissipate heat generated by joule heating of the windings. To obtain the thermal characteristics of the motor, current was applied across two leads and we measured the resulting temperature change. These data were used to develop a thermal model, which enables simulation of the rise in winding temperatures for arbitrary current inputs across different robotic applications (assuming the conduction and convection-based thermal resistances do not change appreciably). The thermal model predicts a maximum continuous q-axis current of 7.6 A and 7.7 A; a maximum instantaneous current (for 20 s) of 23.8 A and 28.7 A for the T-motor and DAP, respectively, assuming a maximum permissible winding temperature of 125°C.

2.5.4 Closed-loop Control

The closed loop performance differed substantially between control types (*i.e.* position control and current control). The position control bandwidth was between 15 - 20 Hz, while the current control bandwidth was greater than 320 Hz. In the time domain, the position control has a consistent velocity during the transient aspect of the step response; that is, response did not vary with step size. This nonlinear behavior stems from the maximum

voltage that can be applied, namely, the back-EMF voltage is limited by the power source voltage ($\sim$36 V), known as actuator saturation. The step response for the current control had significant overshoot and a fast rise time ($\sim$5 ms), when compared to position control. The increased overshoot can be explained, in part, by the numerator dynamics in the plant's transfer function for current control ($\frac{I(s)}{V(s)}$). The faster rise time may be explained by the greater bandwidth of the electrical dynamics, when compared to the mechanical dynamics of the plant ($\frac{\theta(s)}{V(s)}$), as well as the lack of saturation.

2.5.5 Comparison to Maxon EC30 4-pole

ER-BLDC motors have several advantages over commonly used motors often designed for "pick and place" automation tasks. In the field of wearable robotics, the Maxon EC30 4-pole is a common motor [55, 56], with a rated power of 200 W and a mass of 300 g. The ER-BLDC motors, such as the T-motor described in this work, have three main advantages: 1) they have a higher torque density, 2) they are more efficient, and 3) they have a better thermal resistance. The high torque density of ER-BLDC motors stems from the larger motor radius, which increases torque by radius squared. A higher torque constant permits a lower transmission ratio, which has several important advantages, including mass, complexity, efficiency, and controllability. In addition, due to geometric factors, ER-BLDC motors are more efficient. This can be observed by its greater motor constant, when compared to the Maxon EC30 (T-motor: 0.23 Nm/$\sqrt{W}$ vs. Maxon EC30: 0.04 Nm/$\sqrt{W}$). The motor constant describes the thermal loss per unit torque. Finally, ER-BLDC motors may be better at dissipating heat, resulting in a lower thermal resistance. This likely stems from the increase in surface area to volume ratio, which predicts a 28% reduction in thermal resistance, which in actuality, the thermal resistance of the T-motor (4.7 K/W) is 38% less than that of the Maxon (7.6 K/W), where $R_{WA} \gg R_{HA}, R_{WH}$.

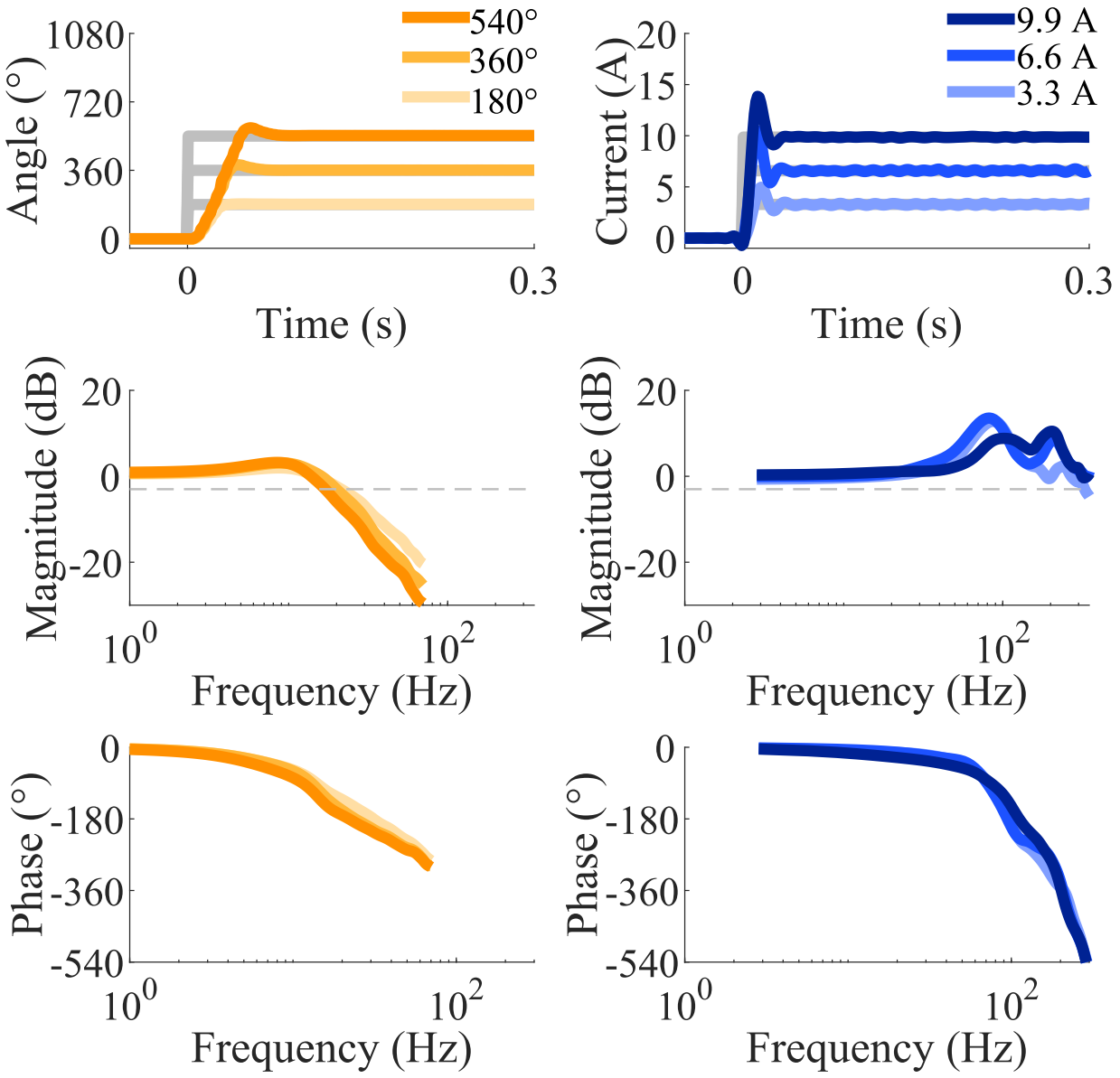

Figure 2.6: Position (yellow) and motor current (blue) step responses (first row) and bode plots (second and third row) for the motor. Bandwidth of the motor was determined by the the frequency where the magnitude crossed -3 dB (dashed line).

Table 2.1: T-Motor & Dephy Actuator Specifications

	T-motor	Dephy Actuator
Characteristics		
Terminal Resistance (mΩ)	186	
Terminal Inductance (μH)	138[a]	
Torque Constant, k_t (Nm/A)	0.14[b]	
Inertia (g·m^2)	0.12 ± 0.002	
Damping (mNm/(rad/s))	0.16	
Mechanical Time Constant (ms)	24.7 ± 1.2	
Bus Voltage (V)	36	
Max Continuous Current (A)	7.6[b]	7.7[b]
Peak Current - 20 s (A)	23.8[b]	28.7[b]
Winding Type	Sinusoidal, Delta	
Thermal		
Thermal Resistance		
Winding-Ambient (K/W)	9861.6	3416.3
Housing-Ambient (K/W)	4.2	3.5
Winding-Housing (K/W)	0.5	1.1
Time Constant Winding (s)	12.9	65.0
Time Constant Motor (s)	548.5	670.6
Control		
Position Steady-state Error (%)		0.4 ± 0.2
Position Rise Time (ms)		22.2 ± 6.1
Position Settling Time (ms)		60.3 ± 14.8
Position Overshoot (%)		8.45 ± 3.59
Position Bandwidth (Hz)		$15 \sim 21$
Current Steady-State Error (%)		0.6 ± 0.6
Current Rise Time (ms)		5.0 ± 0.7
Current Settling Time (ms)		30.2 ± 16.9
Current Overshoot (%)		39.8 ± 12.2
Current Bandwidth (Hz)		> 320
Embedded System		
Commutation Type		FOC
PWM Frequency (kHz)		20
Other Specifications		
Number of Pole Pairs	21	
Number of Phases		3
Product		
Total Weight (g)	240	230[c]

[a] Values reported by Dephy Inc.
[b] Quadrature quantities
[c] Weight excluding the embedded system

Chapter 3

How to Model Brushless Electric Motors for the Design of Lightweight Robotic Systems

3.1 Abstract

A key step in the development of lightweight, high performance robotic systems is the modeling and selection of permanent magnet brushless direct current (BLDC) electric motors. Typical modeling analyses are completed *a priori*, and provide insight for properly sizing a motor for an application, specifying the required operating voltage and current, as well as assessing the thermal response and other design attributes (*e.g.* transmission ratio). However, to perform these modeling analyses, proper information about the motor's characteristics are needed, which are often obtained from manufacturer datasheets. Through our own experience and communications with manufacturers, we have noticed several common errors in modeling BLDC motors that stem from a lack of understanding of the governing equations, compounded by vague or inconsistent terminology used in motor datasheets. The purpose of this tutorial is to concisely describe the governing equa-

This chapter is currently in preparation for submitting to a journal.

tions for BLDC motor analyses and highlight three potential errors. We present a power-invariant conversion from phase and line-to-line reference frames to a familiar q-axis DC motor representation, which provides a “brushed" analogue of a three phase BLDC motor that is convenient for analysis and design. We highlight common errors including incorrect calculations of winding resistive heat loss, improper estimation of motor torque via the motor’s torque constant, and incorrect estimation of the required bus voltage or resulting angular velocity limitations. A unified and condensed set of governing equations is available for designers in Appendix 6.3. The intent of this work is to provide a consolidated mathematical foundation for modeling BLDC motors that addresses existing confusion and fosters high performance designs of future robotic systems.

3.2 Introduction

The success of numerous modern robotics and automation applications is predicated on the use of brushless permanent magnet direct current (BLDC) electric motors. Electric motors convert electrical energy to and from mechanical energy, which requires coordinated interaction between electric and magnetic fields that produce torque via the Lorentz Law. Brushed electric motors often use mechanical contacts—termed *brushes*—to provide energy to the motor’s winding that produces torque. More recently, BLDC motors have been developed to address challenges associated with mechanical brushes, where BLDC motors use multiple windings energized via electrical switching, a process known as *electric commutation*. The absence of mechanical brushes enables BLDC motors to have improved efficiency, power density, longevity, and reduced audible noise [57, 58]. Consequently, BLDC motors are a compact, highly controllable, and efficient actuation method popular across a wide range of robotics applications.

The electromechanical design process often requires careful assessment of motor tech-

nologies to ensure the motor is properly matched to the application. This is especially important for certain areas of robotics, such as legged robots, wearable robots, or autonomous vehicles, where actuator mass is a critical aspect of success. To ensure the motor is properly matched, the torque-velocity requirements of the application are translated to the required current, voltage, and thermal demands, typically assessed *a priori*. The required motor torque-velocity and current-voltage relationships are typically analyzed, in addition to modeling the thermal dynamics during different use cases. This analysis is critical to the design process because the results are used to specify the final components that comprise the system. For example, the required voltage and current govern the power supply demands, the required current is used to specify an appropriate motor driver, and thermal rise governs whether the motor is properly sized for the application. To complete these analyses in the design process, the parameters that govern BLDC motor operation (*e.g.* torque constant, winding resistance, thermal resistance, *etc.*) are used, and are often obtained from manufacturer-provided sources of information (*i.e.* motor datasheets).

Motor datasheets provide important specifications for determining the most appropriate motor for a specific application; however, the ways these specifications are provided can lead to common errors in motor analysis. Roboticists usually simplify three phase BLDC motors to an analogous single phase “brushed" motor, which is convenient and intuitive to assess, since it is described by a single current, voltage, resistance, inductance, torque constant, *etc*. To this end, care must be used when reducing to the brushed electromechanical model when assessing BLDC motor technologies. For example, the winding resistance provided in manufacturer datasheets is often the *terminal resistance*, which is conveniently measured across two leads of the windings; however, terminal resistance is challenging to use in an equivalent single-phase “brushed" motor model analysis, and if used improperly, causes inaccurate estimates of heat produced (*i.e.* Joule heating) that can be over or under-estimated by as much as 100% (Appendix 6.7). Thus, clarity, stan-

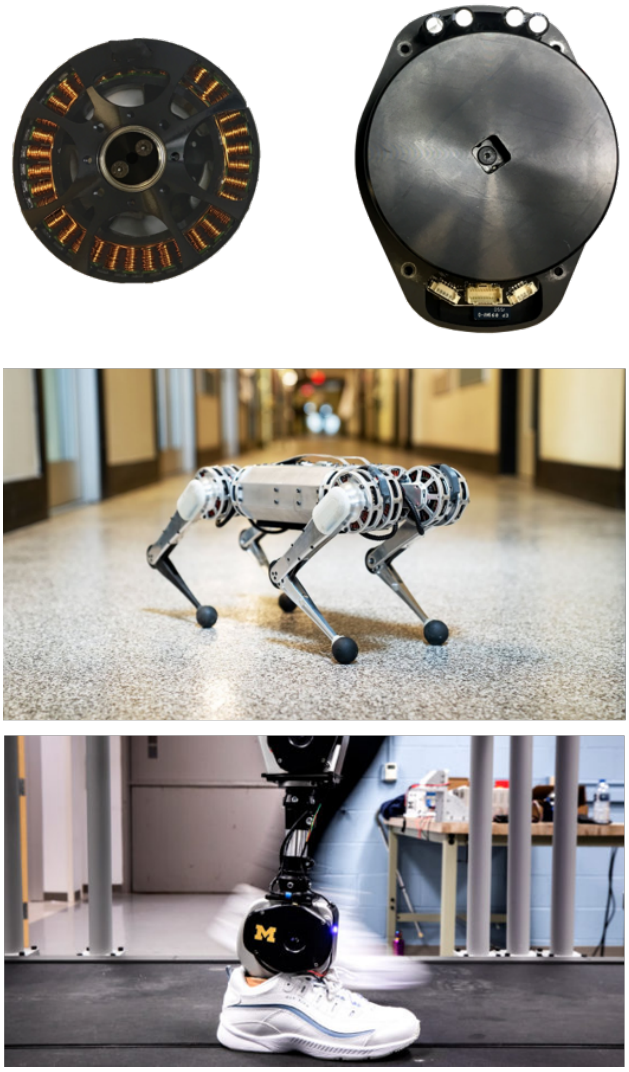

Figure 3.1: The picture of the high-torque exterior-rotor type Brushless DC motor (T-Motor 100 KV, top left; Dephy Actuator Package, top right), and examples of robots using BLDC motors: MIT Cheetah Robot [59] (middle), the University of Michigan's Open Source Bionic Leg [60] (bottom). Figures used with permission.

dardization, and a strong mathematical foundation are needed to ensure BLDC motors are able to be conveniently and accurately modeled for future robotics applications, especially when motor mass is a driving design factor.

We believe this tutorial will be a helpful resource to designers studying the use of high-performance BLDC motors, particularly in mobile robot applications (Fig. 3.1). In addi-

tion to the importance of motor modeling in the design process, other recent work has highlighted the need for greater standardization of analyses and terminology. This is evidenced by recent characterizations of new motors that do not have adequate manufacturer specifications [18]. In addition, researchers have occasionally obtained parameters that differ from what is provided by motor manufacturers [61]. Thus, an improved understanding of motor modeling and the underlying parameters will reduce the time and cost associated with actuator design, use, and characterization [61, 62, 15, 63, 64, 65, 66, 67, 68, 69, 70, 26].

There are many different types of brushless motor topologies, which can add to challenges modeling their performance. In the most general description of brushless motors, there are two types, namely BLDC motors and asynchronous electric motors (*i.e.* AC induction motors). Within BLDC motors, further separation can be made based on the shape of their back electromotive force (back-EMF) profile, being either sinusoidal or trapezoidally shaped. The waveform of the back-EMF profile stems from the distribution of each winding and the shape of stator slots and magnetic poles. The back-EMF shape affects the performance of motors and plays a role in the selection of an appropriate commutation paradigm to maximize efficient torque production [71]. Some previous works have denoted motors with trapezoidal back-EMF profiles as BLDC motors, with permanent magnet synchronous motors (PMSM) being used to denote motors with sinusoidal profiles [58, 72]. This naming convention has become confounded by newly popularized "drone" style exterior-rotor type motors being described as BLDC, despite having a sinusoidal back-EMF profiles. Thus, in this work, we do not distinguish between PMSM and BLDC motor types, and instead we limit our analysis to BLDC motors with a sinusoidal back-EMF profile for ease of analysis.

In this paper, we describe the governing equations for brushless motor operation, and highlight inconsistencies that stem from misinterpretation of motor datasheets. We pro-

vide conversions from phase-based representations of motor current and voltage to a DC representation using the quadrature-direct transformation; by converting to a DC representation, the familiar and convenient single-phase (*i.e.* brushed) DC motor analogue can be used for interpretation and design. We introduce three common errors that arise when motor characteristics from datasheets are used without proper conversions, namely incorrect calculations of resistive losses, inaccurate estimation of motor torque via the torque constant, and improper determinations of maximum voltage needed for power supply specification. The intent of this work is to improve understanding of how brushless motors are described by manufacturer datasheets, and to enable accurate and high performance designs of future robotic systems.

3.3 Electromechanical Modeling of Motors

In this section, we describe the equations that govern brushed and BLDC motors, and introduce the concept of electrical commutation for BLDC motors. Most importantly, we present a brushed motor representation of the BLDC motor using direct-quadrature (d-q) transformation. In this work, we assume these properties and assumptions for the BLDC motor:

- Non-salient
- Three phase sinusodial back-EMF profile
- Identical mutual and self inductances
- d-q transformation is in phase with the rotor ($I^d = 0$)

In this paper, we define saliency as magnetic saliency [73, 74], which describes if a motor's winding inductances vary as a function of rotor angle. Note that this is different from salient poles, which stems from the physical characteristics of the magnetic poles. A motor could have salient poles but be a non-salient motor under this definition; typical hobbyist brushless motors fall under this category [75].

These assumptions provide a more convenient analytical approach while also representing the real-world use case of BLDC motors. For example, an exterior rotor type BLDC motor (Fig. 1) satisfies these assumptions when used with an off-the-shelf field oriented control (FOC) controller and operating with a sufficient DC bus voltage / power supply (Section 3.4.1) [17].

In this paper, we define the *winding reference frame* to collectively describe the different electrical references where superscript ϕ, l, ll and q denote a phase, line, line-to-line, and q-axis quantities, respectively. **Bold font** denotes vectors or matrices, $\bar{bar}$ represents amplitude of a sinusoidal quantity, and we boxed the key equations of each sections. All boxed equations are provided together in Appendix 6.3.

3.3.1 Brushed DC motors

To accurately understand BLDC motors, we first describe the equations that govern ideal brushed DC motors in the electrical and mechanical domains. By applying Newton's Law, we can obtain

$$\boxed{J\frac{d^2\theta_m}{d^2t} = K_t I^a - b\frac{d\theta_m}{dt} - \tau_L} \tag{3.1}$$

where J is the inertia of the rotor, K_t is the torque constant, I^a is the armature current, b is the viscous damping coefficient, θ_m $\frac{d\theta_m}{dt}$ are the rotor angle and the rotor's angular velocity, respectively, and τ_L is the load torque. Applying Kirchoff's Voltage Law (KVL) across the winding yields:

$$\boxed{V = RI^a + K_b\frac{d\theta_m}{dt} + L\frac{dI^a}{dt}} \tag{3.2}$$

where V is the voltage applied across the winding, K_b, is the back-EMF constant, R is the resistance of the rotor winding, and L is the inductance of the winding. The term $K_b\frac{d\theta_m}{dt}$ is known as the back electromotive force (back-EMF), which is the generated voltage that

opposes the voltage across the winding (V). These two equations are the fundamental relationships that govern ideal brushed DC motor operation.

3.3.2 BLDC motors

In this section, we describe the governing equations that underlie the operation of ideal BLDC motors, beginning with a single phase, and expanding to three phase operation. Similar to the equations governing brushed DC motors, we summarize BLDC models for both the mechanical and electrical domains.

Per-phase modeling

We first begin by understanding how a single phase produces torque. Per-phase torque production from a BLDC motor can be expressed as

$$\boldsymbol{\tau}^{\phi} = \boldsymbol{F} \times \boldsymbol{r} = p(NI^{\phi}\boldsymbol{B}^{\phi}(\theta) \times \boldsymbol{\ell}) \times \boldsymbol{r} \tag{3.3}$$

where p is the number of magnetic pole pairs, ϕ denotes the phase quantities, $\boldsymbol{F}$ is the electromagnetic force vector calculated from the Lorentz Law, N is the number of individual coils in the winding, $\boldsymbol{B}$ is the per-phase magnetic flux density vector, and $\boldsymbol{\ell}$ is the vector length of a coil side along the axis parallel to the rotation axis (perpendicular to the magnetic field), $\boldsymbol{r}$ is the rotor radius, and I^{ϕ} is the current flowing through one phase as a function of time (Fig 3.2).

In this work, all angles for BLDC motors are represented in magnetic angles. When BLDC motors have multiple pole pairs, one magnetic rotation is defined by a $1/p$ rotation of the rotor:

$$\theta = \theta_m \cdot p \tag{3.4}$$

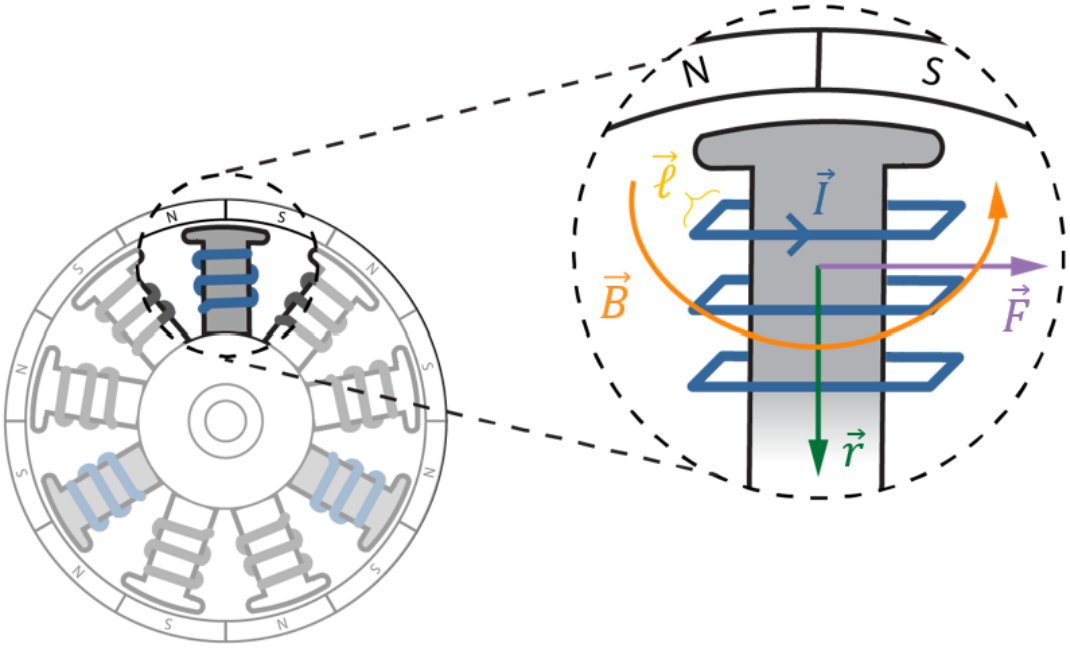

Figure 3.2: A simplified depiction of an exterior-rotor type brushless motor shown with electromagentic force diagram. The gray colored poles indicates that they are part of the same phase. Note that for visualization purposes, electromechanical quantities are depicted as a single pole value, although they are per-phase quantities (magnified view of the figure).

Therefore, to obtain properties of BLDC motors with multiple pole-pairs (*e.g.* K_v) using rotor displacement, the above transformation should be applied accordingly.

Since each winding is perpendicular to the magnetic flux density formed by the permanent magnets, the per-phase torque becomes:

$$\boldsymbol{\tau}^{\phi} = pNI^{\phi}\left\|\boldsymbol{B}^{\phi}(\boldsymbol{\theta})\right\|\|\boldsymbol{\ell}\|\|\boldsymbol{r}\|\boldsymbol{j} \quad (\boldsymbol{B}^{\phi}(\theta) \perp \boldsymbol{\ell} \perp \boldsymbol{r}) \tag{3.5}$$

$$\boldsymbol{\tau}^{\phi} = pNA\left\|\boldsymbol{B}^{\phi}(\boldsymbol{\theta})\right\|\left\|\boldsymbol{I}^{\phi}\right\|\boldsymbol{j} \tag{3.6}$$

where $A = \ell r$ and $\boldsymbol{j}$ is a unit vector perpendicular to both electromagnetic force and the rotor radius (vector going into the page in Fig. 3.2), where $\|\cdot\|$ represents the Euclidean norm. As we assume the magnetic flux density of a BLDC motor is sinusoidal, the magnetic flux density can be expressed as a function of the magnetic angle of the rotor:

$$\left\|\boldsymbol{B}^{\phi}(\boldsymbol{\theta})\right\| = \bar{B^{\phi}}\sin(\theta) \tag{3.7}$$

Therefore, the per-phase torque becomes:

$$\boldsymbol{\tau}^{\phi} = pNA\bar{B^{\phi}}I^{\phi}\sin(\theta)\boldsymbol{j} \tag{3.8}$$

where the $\bar{B^{\phi}}$ is the amplitude of the per-phase magnetic field.

Eq. (3.3) can be further simplified by defining the per-phase torque constant:

$$\boxed{\boldsymbol{\tau}^{\phi} = K_t^{\phi} I^{\phi}\boldsymbol{j}} \tag{3.9}$$

where the per-phase torque constant is defined as:

$$K_t^{\phi}(\theta) = pNA\bar{B^{\phi}}\sin(\theta) \tag{3.10}$$

and the per-phase torque constant is a function of the rotor's position, where $\bar{K^{\phi}}(\theta)$ is the amplitude of the per-phase torque constant:

$$K_t^{\phi}(\theta) = \bar{K_t^{\phi}}\sin(\theta) \tag{3.11}$$

The sinusoidally varying per-phase torque constant highlights the difference from the brushed motor torque constant, which stems from the shape of the rotor magnetic flux density across rotor angle [73].

The per-phase back-EMF can be calculated using Faraday's Law:

$$V_e^{\phi} = Nr(\boldsymbol{B}_{\phi} \times \frac{\boldsymbol{d\theta}}{\boldsymbol{dt}}) \cdot \boldsymbol{\ell} \tag{3.12}$$

where V_e^ϕ is the per-phase back-EMF. Since the magnetic flux density and radius of the coil are perpendicular, and the cross product of those are parallel to the vector length of the coil, the per-phase back-EMF equation reduces to:

$$V_e^\phi = NA\left\|\boldsymbol{B}^{\boldsymbol{\phi}}(\boldsymbol{\theta})\right\|\left\|\frac{\boldsymbol{d\theta}}{\boldsymbol{dt}}\right\| \quad (\boldsymbol{B}^\phi(\theta) \times \frac{\boldsymbol{d\theta}}{\boldsymbol{dt}} \parallel \boldsymbol{\ell}) \tag{3.13}$$

where magnetic flux density has a sinusoidal profile:

$$V_e^\phi = NA\bar{B}_\phi \sin(\theta) p\frac{d\theta_m}{dt} \tag{3.14}$$

where $\frac{d\theta}{dt} = p\frac{d\theta_m}{dt}$. By defining the back-EMF constant, the expression becomes:

$$V_e^\phi = K_b^\phi(\theta)\frac{d\theta_m}{dt} \tag{3.15}$$

where the per-phase back-EMF is the function of the rotor's position:

$$K_b^\phi(\theta) = pNA\bar{B}_\phi \sin(\theta) \tag{3.16}$$

$$K_b^\phi(\theta) = \bar{K_b^\phi} \sin(\theta) \tag{3.17}$$

The per-phase back-EMF has the same profile as the torque constant, which also originates from the shape of the flux density. Note that ideally the per phase back-EMF is identical to the per-phase torque constant (*i.e.* $K_b^\phi = K_t^\phi$).

Three-phase modeling

We now describe how individual phase models of a BLDC motor can be combined to obtain the three-phase model. Total produced torque of the BLDC motor is derived from

the three windings operating in parallel:

$$\boldsymbol{\tau} = (K^{\phi}_{t,A}(\theta)I_A + K^{\phi}_{t,B}(\theta)I_B + K^{\phi}_{t,C}(\theta)I_C)\boldsymbol{j} \tag{3.18}$$

which is the summation of each per-phase torque, where the phase torque constants are identical and 120° out of phase:

$$K^{\phi}_{t,A}(\theta) = \bar{K^{\phi}_t}\sin(\theta) \tag{3.19}$$

$$K^{\phi}_{t,B}(\theta) = \bar{K^{\phi}_t}\sin(\theta - \frac{2\pi}{3}) \tag{3.20}$$

$$K^{\phi}_{t,C}(\theta) = \bar{K^{\phi}_t}\sin(\theta + \frac{2\pi}{3}) \tag{3.21}$$

Similarly, the total back-EMF of the BLDC motor becomes:

$$\boxed{V_e = K^{\phi}_{b,A}(\theta)\frac{d\theta_m}{dt} + K^{\phi}_{b,B}(\theta)\frac{d\theta_m}{dt} + K^{\phi}_{b,C}(\theta)\frac{d\theta_m}{dt}} \tag{3.22}$$

We can apply KVL on each phase quantity, which can be compactly expressed in matrix form (Appendix 6.5).

3.3.3 Electric commutation for BLDC motors

To produce constant torque, BLDC motors require their windings to be sequentially energized at the appropriate time, a process known as electric commutation. This process is performed by a microcontroller known as a brushless motor drive. To maximize the torque production and minimize torque ripple, the driver should provide a current waveform that matches the back-EMF profile [73]. That is, for three-phase BLDC motors with sinusoidal back-emf profiles, the optimal currents to maximize smooth torque production

should be in phase with the back-EMF and sinusoidal:

$$I_A = \bar{I^\phi} \sin\theta \tag{3.23}$$

$$I_B = \bar{I^\phi} \sin\left(\theta - \frac{2\pi}{3}\right) \tag{3.24}$$

$$I_C = \bar{I^\phi} \sin\left(\theta + \frac{2\pi}{3}\right) \tag{3.25}$$

where $\bar{I^\phi}$ is the amplitude of the phase current and currents are 120° out of phase, identical to the phase of torque constants (3.19). For motors with a trapezoidal back-EMF profile, phase currents that match the profile as closely as possible (*i.e.* trapezoidal commutation) should be applied to maximize efficient torque production [76]. By applying these sinusoidal currents to the total torque (3.18, Fig. 3.3), the torque becomes:

$$\tau = \bar{K_t^\phi} \bar{I^\phi} \big[\sin^2(\theta) + \sin^2(\theta + \frac{2}{3}\pi) + \sin^2(\theta - \frac{2}{3}\pi) \big] \boldsymbol{j} \tag{3.26}$$

$$\boxed{\tau = \frac{3}{2} \bar{K_t^\phi} \bar{I^\phi} \boldsymbol{j}} \tag{3.27}$$

by using trigonometry, shown below:

$$\sin^2(\theta) + \sin^2(\theta + \frac{2}{3}\pi) + \sin^2(\theta - \frac{2}{3}\pi) = \frac{3}{2} \tag{3.28}$$

This simplification consists of only DC quantities (*i.e.* torque scales linearly with amplitude of the phase current) and allows us to represent BLDC motors using the brushed motor model. Some literature concludes the conversion to a single phase motor analogue with this equation (3.27) by defining the brushed motor equivalent torque constant as $K_t = \frac{3}{2} \bar{K_t^\phi}$ and the amplitude of the phase current ($\bar{I^\phi}$) as the brushed motor equivalent current [73]. These different definitions of torque constants are valid for calculating torque as long as the appropriate currents are used; however, these conversions result in

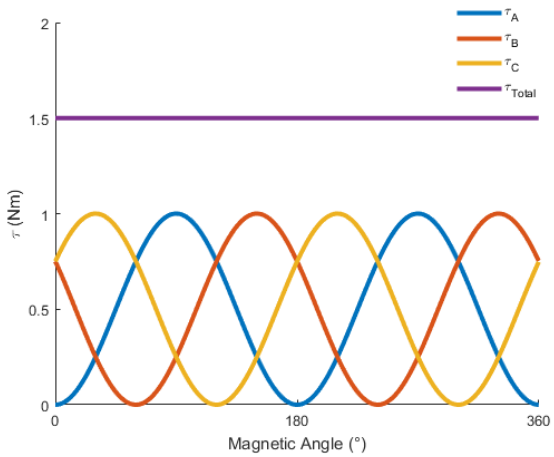

Figure 3.3: A demonstration of torque production of a BLDC motor (3.18). The sum of three phase-torques with constant amplitude results in total torque which is a DC value. Note that each winding produces a torque with a sine squared profile, which results from the product of the sinusoidal torque constant and the sinusoidal phase currents.

other analytic errors (*e.g.* inaccurate thermal loss). Consequently, in this work, we provide a version of the single phase motor representation of a BLDC motor which not only satisfies (3.27), but also enables accurate modeling of other factors (*e.g.* resistive power loss, back-EMF). Methods for appropriate conversions are detailed in the following sections.

3.3.4 d-q transformation

The power-invariant d-q transformation converts brushless motor quantities to a single phase "brushed" motor equivalent representation while preserving power of the system. The power invariant d-q transformation is a unitary transformation that converts three phase axes in the stator frame of the motor to two axes fixed to a rotating coordinate frame that rotates with the rotor's magnetic angle (*i.e.* d-q axis). Conversion to the rotating reference frame of the rotor transforms sinusoidal functions of angle, such as current and voltage, into effective DC values, forming an equivalent single phase **brushed** motor ana-

Sometimes, a different form of the d-q transformation is used which scales the currents such that the q-axis current is equal to the peak of a single phase current [77]. Since this amplitude-invariant form is a non-unitary operation, it renders power calculations inaccurate and is therefore avoided in this analysis.

logue for BLDC motors. The d-q transformation is composed of two transformations:

$$\text{Clarke } (\boldsymbol{C}): \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \tag{3.29}$$

$$\text{Park } (\boldsymbol{P}): \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{3.30}$$

where **C** and **P** are Clarke and Park transformations, respectively, and the matrix product of these transformations becomes the d-q transformation:

$$\boxed{\boldsymbol{P} \cdot \boldsymbol{C} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2}{3}\pi\right) & \cos\left(\theta + \frac{2}{3}\pi\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2}{3}\pi\right) & -\sin\left(\theta + \frac{2}{3}\pi\right) \end{bmatrix}} \tag{3.31}$$

By applying the d-q transformation to phase currents or voltages, the converted quadrature quantities are described as follows:

$$\begin{bmatrix} I^d \\ I^q \end{bmatrix} = \boldsymbol{P} \cdot \boldsymbol{C} \cdot \boldsymbol{I}^\phi = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2}{3}\pi\right) & \cos\left(\theta + \frac{2}{3}\pi\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2}{3}\pi\right) & -\sin\left(\theta + \frac{2}{3}\pi\right) \end{bmatrix} \cdot \begin{bmatrix} I_A^\phi \\ I_B^\phi \\ I_C^\phi \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 0 \\ \frac{3}{2}\bar{I^\phi} \end{bmatrix} \tag{3.32}$$

Thus, the q and d-axis currents are

$$\boxed{I^q = \sqrt{\frac{3}{2}}\bar{I}^\phi} \tag{3.33}$$

$$I^d = 0 \tag{3.34}$$

and the BLDC motor torque can be expressed as:

$$\boldsymbol{\tau}_q = \bar{K}_t^\phi \sqrt{\frac{3}{2}} \bar{I}^\phi \sqrt{\frac{3}{2}} \boldsymbol{j} = K_t^q I^q \boldsymbol{j} \tag{3.35}$$

where the q-axis torque constant is defined as:

$$\boxed{\bar{K}_t^\phi \sqrt{\frac{3}{2}} = K_t^q} \tag{3.36}$$

and superscript q denotes q-axis quantities of according constants, while the d-axis current is zero. Similarly, we can define the back-EMF constant using the same conversion:

$$K_b^q = \sqrt{\frac{3}{2}} \cdot K_b^\phi \tag{3.37}$$

where the phase back-EMF constant and phase torque constant are identical ($K_t^\phi = K_b^\phi$). This relationship shows that there is factor of $\sqrt{\frac{3}{2}}$ difference among quadrature and phase quantities. Therefore, by applying Newton's second law, we can obtain:

$$\boxed{J\frac{d^2\theta_m}{d^2t} = K_t^q I^q - b\frac{d\theta_m}{dt} - \tau_L} \tag{3.38}$$

Similarly, by applying the d-q transformation, the equivalent electrical circuit of a BLDC

motor becomes:

$$\boxed{V^q = R^\phi I^q + K_b^q \frac{d\theta_m}{dt} + L^e \frac{dI_q}{dt}} \tag{3.39}$$

where, L^e is an effective inductance (*i.e.* stator inductance is invariant of rotor position [78]). For full derivation, please refer to Appendix 6.6. Since neither self inductance nor mutual inductance may be accessible, the effective inductance can be obtained through terminal inductance as follows:

$$\boxed{\begin{aligned} &\text{Wye:} \quad L^e = \frac{3}{2} L^{ll} \qquad (3.40) \\ &\text{Delta:} \quad L^e = \frac{1}{2} L^{ll} \qquad (3.41) \end{aligned}}$$

where, L^{ll} represents the line to line or terminal inductance. This assumes the connection scheme of phases as Fig. 1 of [79]. In this work, we assume identical self and mutual inductances across different phases and their combinations. By applying the d-q transformation, the effective phase inductance is equal to the q-axis inductance (Appendix 6.6):

$$L^e = L^q \tag{3.42}$$

The d-q axis representation of a BLDC motor demonstrates that the motor can be fully transformed to the single phase “brushed" motor representation that is helpful for analysis and control.

3.3.5 BLDC motor winding configuration

To fully understand the relationships that govern BLDC motors and their characteristics, we must first understand the motor’s winding type. There are two types of common BLDC windings: wye (*i.e.* star) and delta (Fig. 3.4). The winding type of BLDC motors is deter-

mined by the manufacturer and can be chosen for a variety of factors. Delta wound motors are generally used when greater instantaneous current or torque is required. If equal line-to-line voltages are applied to both motors, the entire voltage is applied across each phase for a delta wound motor ($V^{ll} = V^{\phi}$), whereas the line-to-line voltage is divided between the phases for a wye wound motor ($V^{ll}/\sqrt{3} = V^{\phi}$). In addition, wye wound motors typically have greater efficiency in the lower velocity regions, while delta wound motors are typically used when greater efficiencies are desired in higher velocity regions [80]. Finally, specific winding types may have benefits in certain applications (*e.g.* induction motors) [73].

The winding type dictates how terminal (*i.e.* line-to-line) quantities are related to phase quantities. Lines are defined as the three conductors that branch from the windings (A, B, and C), and line-to-line voltage is defined as the voltages referenced between any two lines. The relationship between the phase quantities and line-to-line quantities for each winding configuration are as follows:

Wye:	$\bar{V}^{ll} = \sqrt{3}\bar{V}^{\phi}$	(3.43)
	$\bar{I}^{l} = \bar{I}^{\phi}$	(3.44)
Delta:	$\bar{V}^{ll} = \bar{V}^{\phi}$	(3.45)
	$\bar{I}^{l} = \sqrt{3}\bar{I}^{\phi}$	(3.46)

Note that currents are referenced within a line, and voltages are measured between lines (*i.e.* there is no such thing as line-to-line current). Depending on the winding type, appropriate conversion factors should be applied to accurately convert between reference frames (*e.g.* d-q transformation). Thus, knowing the winding type is crucial for estimating the motor properties, which ultimately enables accurate modeling, analyses, and commuta-

tion for robotics applications.

3.3.6 Resistive power loss

The heat produced from a motor's windings are among the chief limiting factors in a motor's performance. Thus, to appropriately size a motor for a specific task, the resistive power loss and temperature rise of the windings is often investigated. We can calculate the total resistive heating loss by summing individual losses of each phase:

$$P = (I_A^{\phi\,2} + I_B^{\phi\,2} + I_C^{\phi\,2})R^{\phi} \tag{3.47}$$

$$= \frac{3}{2}\bar{I^{\phi}}^2 R^{\phi} \tag{3.48}$$

where each phase currents are provided as (3.23). Similar to the total torque, the total power loss becomes a DC value. The power loss can also be represented in the q-axis frame:

$$\boxed{P = I^{q\,2} R^{\phi}} \tag{3.49}$$

where it reduces to the single phase brushed motor equivalent expression by using the conversion factors (3.33, 3.36). Note that resistance used in the q-axis power loss formulation is the resistance of a single phase.

3.4 Implications in BLDC Analyses

3.4.1 BLDC datasheet variation

To optimally select BLDC motors for robotics applications, *a priori* modeling analyses are conducted to properly specify the motor for the application; this is especially important when motor mass is a key driver of system performance. The specific parameters needed to conduct these analyses are often obtained from manufacturer datasheets. Inconsistencies and sparse or ambiguous information provided in these datasheets can lead to erroneous results (Table 3.2). That is, manufacturers sometimes report motor parameters with respect to different voltage and current references. Indeed, these parameters are often measured based on their simplicity or convenience, as opposed to what would most readily be used in modeling analyses. Several common reference frames are described in Table 3.3. In the proceeding sections, we describe common errors that arise from improperly using motor parameters in BLDC modeling analyses. We provide conversions from common values of voltage and current, and encourage motor manufacturers to specify the details of the provided parameters with emphasis on reporting with respect to the q-axis representation.

Brushless motor drives that perform field oriented control and report q-axis current and voltage often assume one motor winding type (*e.g.* wye) to perform their calculations. If the motor being used is delta wound, appropriate conversions factors will be needed to ensure accurate calculation of expected motor torque.

The analysis developed in this paper has neglected other sources of losses such as core losses due to hysteresis and eddy currents, bearing friction, air resistances *etc.* [74]. We chose to focus on Joule heating because it is the dominant factor in temperature rise, and it can be challenging to obtain information for other sources of losses from manufacturer datasheet.

Thermal loss, resistance, and inductance

Thermal loss is a key driver of electromechanical performance; to minimize motor mass, the windings are expected to rise in temperature, without exceeding the maximum permissible temperature. To correctly model the resistive heat loss, the heat produced by all three windings is considered (3.47). Equivalently, the thermal loss can be calculated from a single phase "brushed" motor analogue (3.49) using the q-axis current and phase resistance. However, the resistance and current values used in these equations must be paired appropriately; when modeling the resistive heat loss using the q-axis current, the required winding resistance is the phase resistance. This is different from the motor's terminal resistance, which can be reported on BLDC motor datasheets. To convert from terminal resistance to phase resistance, the winding type must be considered (Appendix 6.4). However, this conversion can be challenging, since many BLDC motor datasheets do not include the motor winding type, and thus obscure the ability to calculate phase resistance from terminal resistance. As an example of how resistive heat loss can be erroneously calculated by pairing the incorrect resistance and current, if terminal / line-to-line resistance is used directly instead of the phase resistance, the predicted power loss will be twice the magnitude for wye-wound motors and 33% less for delta-wound motors (Appendix 6.7).

To address this issue, terminal resistance can be scaled to the appropriate phase resistance by (3.50) and (3.51), and used to calculate total resistive power loss (3.49):

$$\boxed{\text{Wye:} \qquad R^{\phi} = \frac{1}{2} R^{ll}} \tag{3.50}$$

$$\boxed{\text{Delta:} \qquad R^{\phi} = \frac{3}{2} R^{ll}} \tag{3.51}$$

Following the determination of the appropriate resistance values to use in power loss

calculations, the expected rise in temperature can be obtained in simulation using the thermal resistances and capacitances for the coupled winding-housing system dynamics[18].

Torque

The calculation of expected motor torque is a critical aspect of the design process. There are two common errors that result from inaccurate usage of motor parameters; firstly, some recently developed motors from the drone industry provide the velocity constant (K_v) rather than torque constant, and this is sometimes erroneously converted directly to the motor's torque constant. Secondly, the current used in the formulation of the motor's torque constant reported in the datasheet may be ambiguous, and can cause torque estimation error if the current used to determine the torque constant does not agree with the current used to predict torque in modeling analyses. To address these challenges, appropriate conversion factors and explanations are provided in the paragraphs below.

To calculate expected motor torque, a single phase "brushed" motor analogue is typically used, in which the motor's torque constant is multiplied by the effective winding current. In some cases, motor manufacturers provide the voltage constant as a proxy for the torque constant, and leave the torque constant determination to the designer since these values are equivalent with proper commutation (*i.e.* the manufacturer only provides the reciprocal of the motor's K_b, known as K_v). This may lead to a common error—considering K_b equal to K_t—which is not necessarily true for brushless motors [74], depending on the winding reference frame. For example, a common motor used in some robotic applications is the T-motor KV100 (U8), where only the velocity constant K_v ($\frac{1}{K_b}$) is provided as 10.47 rad/V·s (100 RPM/V). However, the reciprocal of this value actually (nearly) describes the amplitude of the line-to-line sinusoidal back-EMF per unit angular velocity (*i.e.* $\bar{K_b^{ll}}$), which, because the motor is delta wound, represents the back-EMF amplitude for a single phase as a function of rotor velocity (3.45). However, in practice,

the amplitude of the line-to-line back-EMF is ∼5% to 10% greater than what would be expected using the manufacturer reported K_b^{ll} (potentially converted from K_v^{ll}). To obtain the q-axis torque constant for an equivalent single phase "brushed" motor, the amplitude-based $\bar{K_b^{ll}}$ is multiplied by $\sqrt{\frac{3}{2}}$, which can then be used with the q-axis current. Refer to Appendix 6.8 for full derivation. Generalized for both winding types, the conversions are:

$$\boxed{\begin{aligned} &\text{Wye:} \quad K_t^q = \frac{1}{\sqrt{2}}\bar{K_b^{ll}} \\ &\text{Delta:} \quad K_t^q = \sqrt{\frac{3}{2}}\bar{K_b^{ll}} \end{aligned}} \tag{3.52, 3.53}$$

The BLDC motor torque can be also calculated using the velocity constant, which is agnostic to winding type:

$$\boxed{\tau = \frac{\sqrt{3}}{2}\bar{K_b^{ll}}\bar{I^l}} \tag{3.54}$$

which can be derived from equations 3.59, 3.52, 3.53, 3.57 and 3.58.

Modeling expected motor torque can be further hindered by unclear representations of torque constants from manufacturer datasheets. Informal communications revealed several different current conventions used when reporting torque constants, which were rarely explained on their datasheets or product literature. While our understanding of the currents used by these manufacturers is reported in Table 3.4, it should be noted that it was challenging to obtain the exact definitions of the currents used in torque constants (which illustrates the issue) though we are grateful to the manufacturers who worked with us. Thus, care must be used when interpreting torque constants, and we recommend conversion of torque constants to the q-axis representation for accurate and convenient modeling analysis. The following equations can be used to convert torque constants provided with different current references to the equivalent torque constant using the q-axis winding reference frame:

$$\boxed{\begin{aligned} &\text{Wye:} \qquad K_t^q = \sqrt{\frac{3}{2}}\bar{K_t^\phi} \\ &\text{Delta:} \qquad K_t^q = \sqrt{\frac{3}{2}}\bar{K_t^\phi} \end{aligned}} \tag{3.55, 3.56}$$

For convenience, we also provide the equations for q-axis current obtained using the different winding reference frames:

$$\boxed{\begin{aligned} &\text{Wye:} \qquad I^q = \sqrt{\frac{3}{2}}\bar{I}^\phi = \sqrt{\frac{3}{2}}\bar{I}^l \\ &\text{Delta:} \qquad I^q = \sqrt{\frac{3}{2}}\bar{I}^\phi = \frac{1}{\sqrt{2}}\bar{I}^l \end{aligned}} \tag{3.57, 3.58}$$

Thus, using the appropriately paired q-axis current and torque constant, the expected torque can be modeled by:

$$\boxed{\tau = K_t^q I^q} \tag{3.59}$$

Voltage

An important factor in the design process is choosing the required bus (*i.e.* battery or power supply) voltage. The bus voltage governs the maximum velocity at which a motor can operate, assuming the motor is operating within other speed restrictions (*e.g.* bearing maximum angular velocity). In a brushed “single phase" motor, the calculation of maximum no-load velocity is trivial, simply obtained using the bus voltage divided by the back-EMF constant. However, when analyzing brushless motors, obtaining the maximum velocity requires a more detailed analysis. To determine the maximum no-load velocity, we need to consider the maximum amplitude of the motor’s line-to-line voltage, governed by the motor’s winding type. The amplitude of the line-to-line voltage may exceed the q-axis voltage, and thus designing based on the q-axis voltage may result in unexpected

limitations to motor velocity. The amplitude of the motor's line-to-line voltage can be obtained using (3.60) and (3.61):

$$\text{Wye:} \quad \bar{V}^{ll} = \sqrt{2}V^{q} = \sqrt{2}K_b^q \frac{d\theta_m}{dt} \tag{3.60}$$

$$\text{Delta:} \quad \bar{V}^{ll} = \sqrt{\frac{2}{3}}V^{q} = \sqrt{\frac{2}{3}}K_b^q \frac{d\theta_m}{dt} \tag{3.61}$$

where K_b^q is the q-axis back-EMF constant (equivalent in magnitude to the q-axis torque constant) and V^q is obtained using the back-EMF constant and angular velocity. Solving these equations for velocity and substituting V^{bus} for $\bar{V}^{ll}$ provides an estimate for the maximum angular velocity for a given bus voltage under no load conditions:

$$\text{Wye:} \quad \frac{d\theta_m}{dt}_{max} = \sqrt{\frac{1}{2}}\frac{V^{bus}}{K_b^q} \tag{3.62}$$

$$\text{Delta:} \quad \frac{d\theta_m}{dt}_{max} = \sqrt{\frac{3}{2}}\frac{V^{bus}}{K_b^q} \tag{3.63}$$

In addition, there are other methods to calculate maximum velocity (*e.g.* using phase quantities, including $\bar{K_b^{\phi}}$), however, we recommend using the q-axis representation for consistencies with other analyses.

3.4.2 Standardizing Motor Analysis

Accurate modeling of the torque, velocity, current, and voltage needed for robotic applications is an important step in the design process. To ensure motor manufacturers provide the necessary information needed to perform these analyses, we suggest a standard set of

Some BLDC drives will modify the commutation routine when $\bar{V}^{ll}$ is saturated at V^{bus}–this enables the motor to achieve a greater allowable velocity for a fixed V^{bus}. When this occurs, the voltage and current profiles can be become non-sinusoidal. This alteration generally causes a non-zero d-axis current that effectively modifies the motor's velocity constant. This approach can be used to increase or decrease the motor's velocity constant for a corresponding loss in efficiency but is outside the scope of this tutorial.

information should be provided. Ideally, motor manufacturers would report information with respect to the q-axis reference frame, however, if this is impractical, we recommend the following. Datasheets could include line-to-line (terminal) resistance and inductance, in addition to the winding type, which enables conversion to the q-axis reference frame. The torque constant can be provided with respect to any winding reference frame, but the current used in the quantification of torque constant must be clearly described. In addition, while some manufacturers provide information on thermal dynamics (*e.g.* winding / housing thermal resistances and time constants), many omit this information, leaving this characterization to the community. Overall, each parameter should clearly indicate how it relates to a specific winding reference frame, and, if winding type is provided, conversions can be performed with ease.

3.5 Conclusion

In this tutorial, we present the underlying mathematical modeling of brushed and brushless DC motors, as well as electrical commutation and winding configurations of BLDC motors. Most importantly, we described common sources of error in BLDC motor modeling, which often stem from inconsistencies and misinterpretation of manufacturer datasheets. To address these errors, we provide explanations and conversions to the direct-quadrature reference frame, which facilitates a convenient DC representation while conserving key motor properties (*e.g.* resistive power loss). The intent of this tutorial is two fold: 1. to guide engineers and robot designers who utilize BLDC motors in their application to more accurately model and select a motor that is optimal for their application; and 2. to highlight the need for greater standardization and details to be provided in manufacturer datasheets (*e.g.* unit clarity, winding configuration, *etc.*). We hope that this work can serve as a benchmark for standardizing BLDC motor modeling analyses while steering

engineers and designers to more accurate and high performing system designs.

Table 3.1: Nomenclature of electromechanical quantities

Notation	Units	Description
$\bar{[\cdot]}$		Amplitude of a sinusoidal quantity
$[\cdot]^{\phi}$		A quantity in the phase reference frame
$[\cdot]^{ll \text{ or } l}$		A quantity in the line-to-line (terminal) or line reference frame
$[\cdot]^{q}$		A quantity in the q-axis reference frame
$[\cdot]^{RMS}$		The root-mean-square of a phase quantity
$[\cdot]_{A,B, \text{ or } C}$		Specific phase of a BLDC winding
I	A	Current
I^a	A	Armature current
V	V	Voltage
V^{bus}	V	Bus voltage
V^e	V	Back-EMF voltage
R	Ω	Electrical resistance
L	H	Inductance
L^e	H	Effective inductance
K_t	Nm/A	Torque constant
K_b	Vs/rad	Back-EMF constant
K_v	rad/Vs	Velocity constant
τ	Nm	Torque
P	W	Resistive power loss
B	T	Magnetic flux density
F	N	Electromagnetic force
J	Kgm^2	Rotor inertia
θ	rad	Rotor angle
b	Nm/rad/s	Damping coefficient
ℓ	m	Length of a coil
r	m	rotor radius
d	m	Half width of a coil
N		Number of coils in the winding
p		Number of pole pairs
$\boldsymbol{j}$		Unit vector

For example, $\bar{K_v^{ll}}$ is the line-to-line velocity constant, which represents the no load velocity divided by the amplitude of the sinusoidal line-to-line (terminal) voltage (sometimes known as a motor's K_v number), is obtained by combining three different rows of the table.

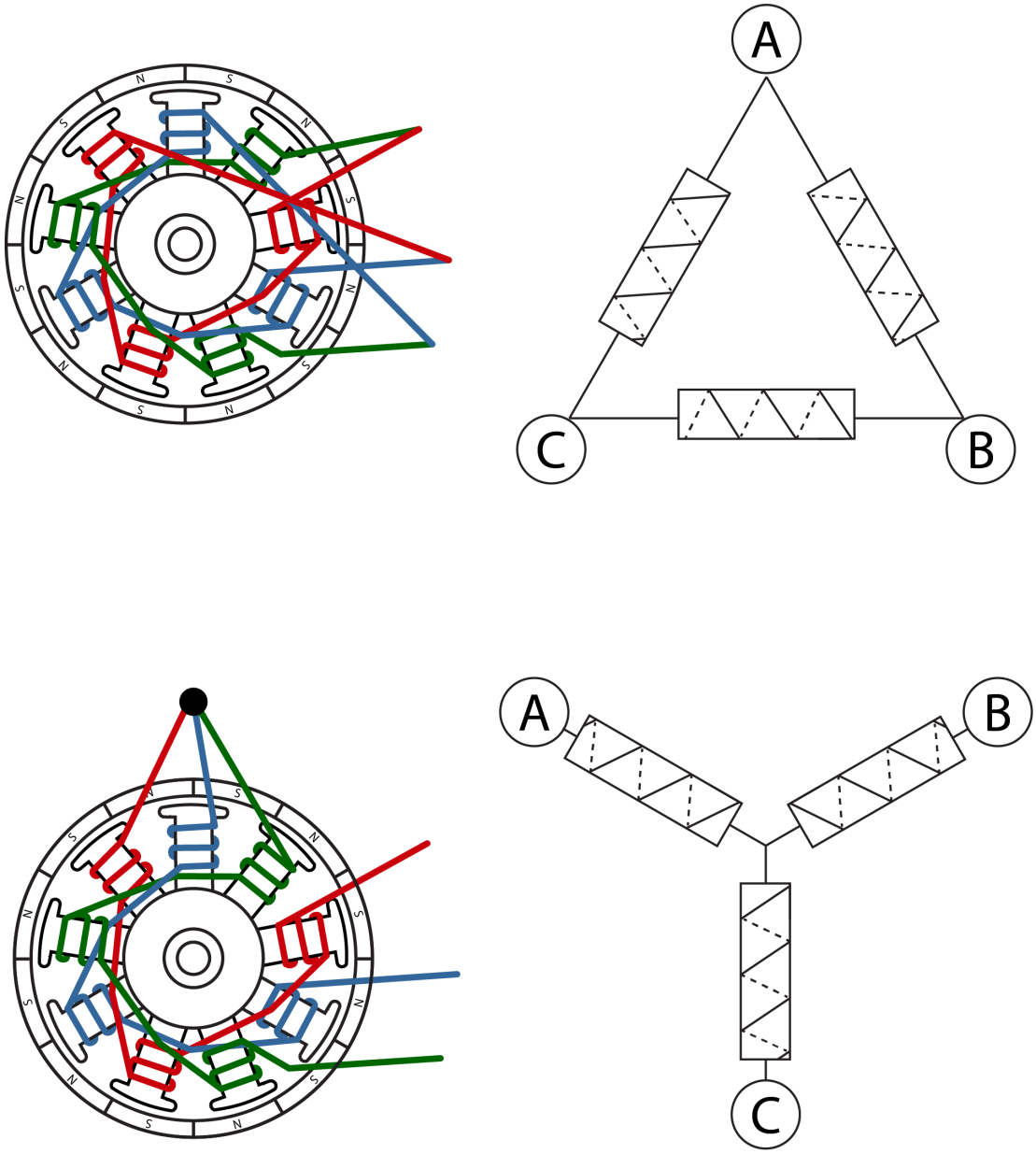

Figure 3.4: Depictions of delta (top) and wye (bottom) winding configurations.

Table 3.2: Brushless Motor Specifications

Characteristics	Definition & Comments
Terminal Resistance (Ω)	Represents electrical resistance across two motor leads (line-to-line or phase-to-phase resistance). Note that to estimate the phase resistance (neutral-to-line), one should know the winding configuration of the motor, which is typically not provided by the motor manufacturers. Appendix 6.4 details how to determine the winding type.
Torque Constant, K_t (Nm/A)	Ratio of torque output to provided current. Most of the cases for BLDC motors, the correct unit of the current is not provided (e.g. whether this is phase current or line current), which is the paramount factor for leading to incorrect power loss of the motor.
Back-EMF Constant, K_b (V/(rad/s))	Induced voltage created by rotating a motor. Typically provided in line-to-line reference frame, however; often the representations are omitted in the datasheet.

Table 3.3: Common representations of current and voltage in brushless motors

Notation	Description
$\bar{I^q}, \bar{V^q}$	q-axis current / voltage, equivalent to RMS current / voltage of all three phases[†]
I^ϕ, V^ϕ	Peak current/voltage of one phase, equivalent to q-axis current/voltage using the power-varying transform
I^l, V^{ll}	Line current and line-to-line voltage, measured within or across two of the motor leads, respectively. Conversion factors vary depending on winding type.
I^{RMS}, V^{RMS}	RMS current/voltage for one phase

[†] $\sqrt{\frac{\int_0^T (I_A^{\phi^2} + I_B^{\phi^2} + I_C^{\phi^2})}{T} dt} = \sqrt{\frac{3}{2}} \bar{I^\phi} = I^q$

Table 3.4: Different manufacturers' representation of torque constants

Manufacturer	Current in K_t
Maxon	$\bar{I^{\phi}}$
T-Motor	I^{bus}
TQ Drive	$\bar{I^{\phi}}$
Micromo †	I^{RMS}
Kollmorgen	I^{q}
Parker	I^{RMS}

Note that the above description is based on informal conversations with engineers from each manufacture, not from official product documentation (it was unlisted). In addition, the table only applies to FOC/Sine commutation.

† It is unclear whether they use RMS of three phases or a single phase, which should be further investigated.

Chapter 4

Preference-in-the-loop Optimization for Robotic Ankle Exoskeleton

4.1 Abstract

A major challenge to the widespread success of augmentative exoskeletons is accurately 'tuning' the controller to provide useful and symbiotic assistance. Often, the controller parameters are adjusted to optimize a physiological or biomechanical objective (*e.g.* metabolic rate, kinematics). However, these approaches are time consuming and resource intensive, while only enabling optimization of a single objective. In reality, exoskeleton user experience is derived from a myriad of factors, including comfort, fatigue, stability, among others. The purpose of this work is to conveniently and automatically tune the exoskeleton controller settings to maximize user preference in real time (*i.e.* preference-in-the-loop control). We propose a machine learning-based optimization framework to personalize controller settings in real-time. We utilize a previously-collected dataset to learn the generic preference landscape with a neural network, then the learned landscape is used

This chapter is currently in preparation for submitting to a journal.

to inform suggestions for an evolutionary strategy that optimizes the controller settings of a novel user. The user provides their pairwise choice feedback through a interface via a touchscreen. Our innovations are threefold: 1. optimizing user preference for a lower-limb exoskeleton in a multi-dimensional controller space; 2. using a neural-network to learn the preference landscape across multiple users; 3. employing black-box optimization + a neural network to efficiently identify the user's preferred settings. Our results indicate that users were able to identify optimized controller settings over randomly generated settings with rate of 88% on average.

4.2 Introduction

Wearable robots, including exoskeletons and powered prostheses, hold a potential to improve the quality of life by transforming the mobility of human [10, 26]. Millions of people suffers from mobility issues due to aging, weakened muscles, stroke, and limb loss etc [81, 82]. These individuals are likely to walk slower, fatigued easily, and prone to fall which lead to secondary conditions including obesity, and depression [82, 83, 84, 85]. Robotic assistive technology could potentially address the issue stemming from mobility by augmenting the locomotion by battery and motor. Advancement in microprocessor and electric motors enabled plethora of assistive robots; however, we don't see these technologies often in everyday life, and one of the main reasons is that controlling these devices in useful and desired way is challenging [26].

Modern control strategies typically includes a set of parameters (*i.e.* control laws) to mathematically describe the assistance of wearable robots. These parameters should be adjusted adequately such that it can be accustomed to an individual or an activity [19]. To tune these control parameters, there are two conventional methods: 1. Expert-based tuning by mimicking natural human locomotion. 2. Automatic tuning based on

metabolic cost. Although these approaches hold potentials, one of the most limiting factor is that they focus on a single objective – they either only focus on kinematics / kinetics or metabolic cost [31, 20]. In reality, there are multiple factors influences objectives, such as stability and comfort, and these single-objective based tuning approaches may not capture these aspect [36]. As an alternative, we propose a paradigm shift in tuning control parameters by leveraging user preference – asking people what they want. We argue that human preference encompasses multiple physiological or biomechanical objectives such as comfort, balance, fatigue etc. User preference-based tuning approach has a promise of transforming the concept of tuning by easily being adopted by users without the need of external devices (apart from the robot) while meeting the objectives of the users.

Recently, using user preference has been emerging as a tool for designing control strategies. Previous works from our group used user-driven tuning approaches to identify users preference for controlling exoskeletons and prostheses [35, 36]; however, since these approaches rely on user to explore the controller space, it may not scale to higher dimension. There has been automatic tuning methods to optimize preference using Bayesian optimization; however, they primarily used a full-body robot exoskeleton (Atlante) where robot drives the motion stemming from the field of bipedal robotics [38, 37, 39]. Therefore, it is unclear how this approach would apply to devices that have symbiotic relationship with the users (*e.g.* ankle exoskeletons, prostheses) where user drives the motion while robot assists the movement in tandem.

In this work, we present a sample-efficient active learning strategy- based controller for optimizing preference while user is in the loop. We used forced pairwise preference feedback to elicit user preference. To this end, we first describe our novel algorithmic contribution using a neural network and evolutionary strategy. Then we validate the proposed algorithm both in simulation and hardware. For simulation, we study the sample efficiency of the proposed method compared to the baseline evolutionary optimization

strategy. For hardware validation, we conducted human subject experiment using ankle exoskeleton system. Our contributions are threefold: 1. devising a sample efficient optimization algorithm for optimizing user preference 2. learning latent users preference landscape across multiple subjects 3. optimizing user preference for controlling lower-limb exoskeletons in high dimensional (four) space. The average validation accuracy of 88% on human subject testing demonstrates the feasibility of our approach in real-world use case. The intent of our work is to present a framework for users to tune their controllers automatically; we hope that this work can be a starting point for translating these wearable technologies outside of laboratories, ultimately, in our daily lives.

4.3 Method

The proposed approach uses an active learning paradigm, where a learning algorithm interactively queries a user's preference to identify the optimal settings (*e.g.* torque assistance setting for a wearable robot). The design principle of the algorithm is to build a system which converges to preferred setting while minimizing the users' query. There are two components in the optimization algorithm, one is modeling the latent preference function of the user, and the other is elicitation strategy which efficiently queries the users based on the inferred preference function. For modeling the preference function, we employ neural networks (*i.e.* RankNet [86]) to estimate preferred settings given a pair of parameter samples. In addition, the model provides estimation of a score for each parameter setting, which will be used to rank the given set of parameters (more than two). We exploit an evolutionary algorithm (CMA-ES, $1 + \lambda$ [87, 88]) as an elicitation method to sample parameters for querying the user. Given the estimated score from RankNet, CMA-ES selects the best parameter among the population in each generation. Our approach is inspired by [89]; however, it differs with the work by using RankNet instead of

RankSVM [90], which accounts for nonlinearity of the preference model. We validated the proposed algorithm, namely *RankCMAES*, in both simulation and robotic hardware with human subjects. The following sections describe the detail of the algorithm and validation methodologies.

4.3.1 Modeling Human Preference

We modeled human preference, in other words, estimated users' latent preference function to identify which settings are better than the other. We employed a RankNet, which is a gradient descent method originally developed for learning ranking problem (*e.g.* ranking search results on the internet). For our case, we are not necessarily interested in ranking the entire order of a set of settings, but more in modeling the landscape of human preference to identify the best setting in the set. To elicit user's preference, we used forced pairwise choice paradigm – asking people to ask to choose the setting they relatively prefer. Previous works demonstrated that pairwise feedback system is more reliable than numerical scores (*i.e.* 1-10 absolute scale) [91, 92]. The feedback was provided as labels for modeling the preference function. In mathematical formulation, given a true preference function $f : R^d \rightarrow R$, and a pair of settings x_a, x_b, a user prefers setting a over b ($a \succ b$) if $f(x_a) > f(x_b)$. We denote $\succ$ to indicate users preference between a pair of settings. To learn the estimate $\hat{f}$ of the function, we minimized the cross-entropy loss between the estimation of preference and true preference [93, 94]:

$$loss(\hat{f}) = -\sum \delta_a \log \hat{P}[x_a \succ x_b] + \delta_b \log \hat{P}[x_a \prec x_b] \tag{4.1}$$

where $\hat{P}$ denotes the probability of preferring a setting using a logistic function:

$$\hat{P}[x_a \succ x_b] = \frac{\exp\Big\{\hat{f}(x_a)\Big\}}{\exp\Big\{\hat{f}(x_a)\Big\} + \exp\Big\{\hat{f}(x_b)\Big\}} \tag{4.2}$$

and δ is a function which outputs binary number $0, 1$ depending on the true preference (*i.e.* training label):

$$\begin{aligned} \delta_a = 1, \delta_b = 0 \quad \text{if} \quad x_a \succ x_b \\ \delta_a = 0, \delta_b = 1 \quad \text{if} \quad x_a \prec x_b \end{aligned} \tag{4.3}$$

4.3.2 Learning Preference Function

Dataset

We created a dataset by using a previous study from our group which identified preferred assistance settings across multiple users [36]. The assistance control parameters included peak torque timing and peak torque, which are used to shape a torque profile to be prescribed to a bilateral ankle exoskeleton [20] (Details in Sec. 4.3.5). We converted 12 subjects' (Naive subjects) preferred setting composed of two parameters into pairwise preferences, where each datum had two sets of parameters and corresponding label which denotes the preference. For example, if $x_a \succ x_b$, two settings are x_a, x_b and the label is $[1, 0]$. To create user scores, we first fitted 2 dimensional Gaussian for individual users, and compared the score from the estimated Gaussian to generate pairwise labels. In total, the size of the data was 60,000 samples, where we uniformly sampled the settings in 2D parameter space defined by the bounds in [36].

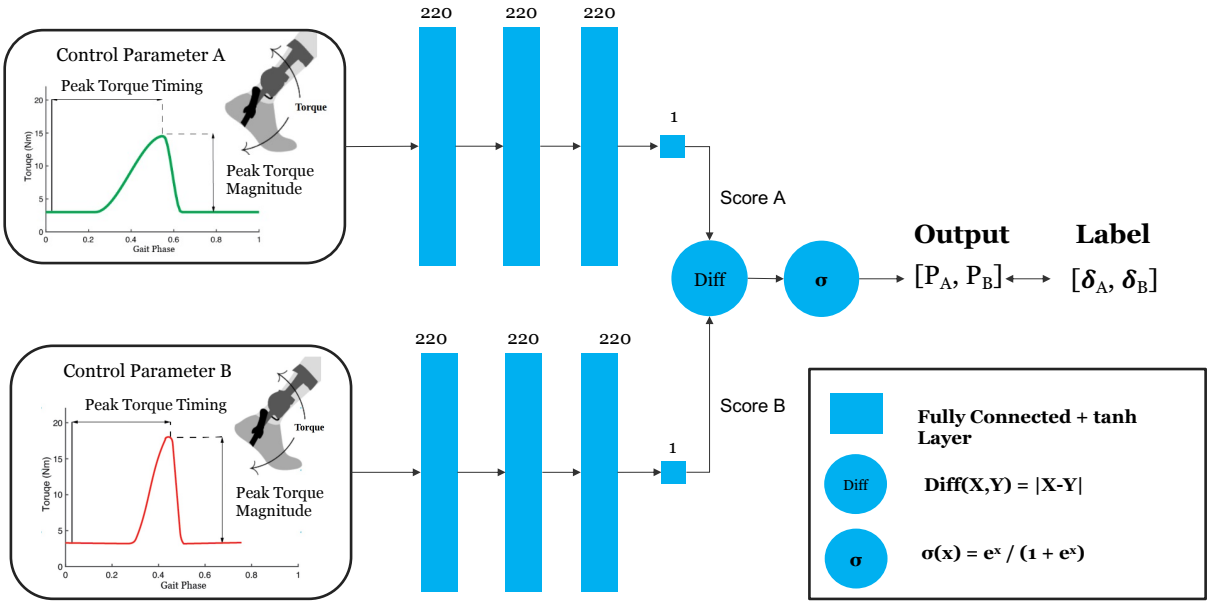

Figure 4.1: Ranknet Architecture. During the training of the neural network, Ranknet takes two sets of input (controller parameters) and minimize the cross-entropy loss between the output of the Ranknet and ground-truth preference label. Each control parameters goes through the feed-forward network backbone which is converted to an estimated score of corresponding setting. Then the difference of the scores are calculated and converted to a probability of selecting one setting over another using sigmoid function.

Ranknet Architecture and hyperparameter search

We used a multi-layer feed forward neural network to model the preference function $\hat{f}$, which maps input (*i.e.* controller settings) to a score (*i.e.* latent user score). To find the optimal number of hidden layers, neurons, and hyperparameters including regularization rate and batch size, we performed Bayesian optimization (BO) to maximize the estimated preference score, which is the output of the Ranknet. BO was used to identify the parameters in relatively few iterations compared to a grid search [95]. After the tuning, the number of hidden layers, neurons per layer, type of activation function, regularization rate, and batch size were chosen as 3, 200, tanh, 1e-4 and 16, respectively.

Training Ranknet

We divided the dataset into training and validation set with 8:2 ratio across all subjects data, and trained the Ranknet on the training set to minimize the cross-entropy loss (4.1). We employed a stochastic gradient-based optimizer ADAM [96] with L1 and L2 regularization to prevent overfitting. A number of epochs of 10 was used during the training process.

4.3.3 Optimization of Human Preference

CMA-ES

We developed a sample efficient black box optimization to optimize human preference (*i.e.* preferred parameters). The black box optimization, namely Covariance matrix adaptation evolution strategy (CMA-ES), was used since the true preference function of human is unknown [87]. The overall procedure of optimization is as following: first, a population of samples (*e.g.* controller settings) is generated from a multivariate normal distribution. Second, the best candidate sample (*i.e.* elite) is compared with the parent sample, in which the comparison is conducted by the user. Third, if the new candidate sample is better than the parent, update the shape of the normal distribution (covariance matrix adaptation). Compared to standard ES, the mean of the population's distribution is updated based on covariance matrix such that the likelihood of selection (*e.g.* preference) increases [87]. We used $1 + \lambda$ CMA-ES [88], which is a variant of traditional μ, λ CMA-ES [87], where μ, λ denotes number of parent, and offspring, respectively and , signifies that a new generation is selected from only children / offspring, while $+$ denotes that new generation is selected from a pool of children and parent. $1 + \lambda$ CMA-ES exploits success rule based size step adaptation as opposed to path length control [88]. The main reason for the choice of the

particular CMA-ES is that it has faster convergence (*i.e.* less number of function evaluation before convergence) [97, 88]). Despite the fact that $1 + \lambda$ strategy may be prone to converging to sub-optimal local minima, because of the expensiveness of function evaluation due to nature of human subject experiment, we focused more on the sample efficiency of the algorithm.

Combining Ranknet with human preference

We combined the model of human preference using Ranknet into the CMA-ES. We name this algorthim as *RankCMAES*. The main innovation of the algorithm is that we combined Ranknet to increase the sample efficiency by reducing the amount of queries we ask to the user. The contribution on the algorithm are as follows: 1. Sorting offsprings is performed by Ranknet. The best offspring was chosen based on the estimated score from the pre-trained Ranknet, instead of eliciting users' preference for all offspring candidates. This significantly reduces the amount of queries required by users. 2. User's pairwise comparison is directly used for updating CMA-ES parameters. Specifically, user provides pairwise preferences over a pair of single parent and best offspring. To balance the model inaccuracy of Ranknet, we leverage user's response for updating covariance matrix of the CMA-ES. Therefore, only one user response per generation will be required to proceed with the optimization. 3. Number of lambda decreased as optimization progressed. In every N number of iterations, the number of offsprings (*i.e.* lambda) reduced by half, which eventually reduces to 1+1 CMAES towards the end of the optimization. The scheduling of lambda is exploited such that it balances the convergence speed with accuracy of optimization. Because the inevitable source of error in modeling of human preference, as the parent approaches to the optimal parameters, it may lack the granularity of closely reaching the optimal points compared to 1+1 CMAES. Therefore we can achieve fast convergence speed while not sacrificing the optimization accuracy by leveraging the Ranknet

when accuracy less matters, and gradually turning off the contribution of the Ranknet and relying on the user more as generation increases. 4. Ranknet infers the estimated scores based on the subspace of parameter space of the optimization. For example, Ranknet estimates the score on based on peak torque timing and peak torque among four ankle exoskeleton optimization parameters space (Sec. 4.3.5). This allows to use existing dataset (Section 4.3.2), which prevents the need of extra data collection effort, which is expensive due to the nature of human subjects experiment. The overall optimization procedure is described in Algorithm 1.

Algorithm 1 Rank-CMAES

Parameter

x - input parameter for optimization
x^g_{parent} - parent in g generation
x^g_i - ith offspring in g generation
K - Number of total generation until termination
λ_{succ} - number of successful candidates
p_{succ} - success rate
C - covariance matrix

procedure RANKCMAES(a, b)
  Init x^g_{parent}
  while $g < K$ **do**
    for i=1,..,λ **do**
      $x^{g+1}_i \sim N(x^g_{parent}, C)$
    end for
    $p_{succ} = \lambda_{succ}/\lambda$
    UPDATESTEPSIZE(p_{succ})
    $x^{g+1}_{best} = Rank(x^{g+1}_{i=1,..,\lambda})[0]$ ▷ Sort by scores from Ranknet
    if $x^{g+1}_{best} \prec x^{g+1}_{parent}$ **then** ▷ User provides the preference
      $x^{g+1}_{parent} \leftarrow x^{g+1}_{best}$
      UPDATECOVARIANCE($x^{g+1}_{parent}, C, p_{succ}$)
    end if
    $g \leftarrow g + 1$
  end while
end procedure

4.3.4 Validation of Optimization: Simulation

We analyzed the performance of the proposed algorithm *RankCMAES* in simulation. In large, the performance of the optimization can be measured by the number of generations before the convergence, and how closely the optimization reaches the global optimum. The procedure for the validation is as following: 1. we defined a synthetic preference profile using 4 dimensional Gaussian for 12 synthetic users. Two parameters (peak torque timing, peak torque) among 4 dimensional space was chosen in dataset described in Section 4.3.2, in other words, only other two parameters (fall time, rise time) were synthetically generated. This choice of preferred settings was to resemble the real world case as much as possible. 2. We performed 12 cross-validation to train Ranknet and test the optimization. Cross-validation was repeated for three times to account for stochasticity of the algorithm. For the optimization, we fixed the number of generation to 150, and the hyperparemters for Ranknet and CMA-ES was fixed across all validations. The number of generations was selected based on the pilot study for a single synthetic subject. All scores (*i.e.* function evaluations) were normalized to 0 to 1 scale.

We conducted ablation studies to further analyze *RankCMAES*. To study the effect of Ranknet, we compared the proposed algorithm with 1+1 CMAES, which has same amount of user response per generation. In addition, to investigate the effect of scheduling of lambda – decreasing the number of offsprings as number of generation increases – we compared the performance of the optimization with and without the scheduling. Initially the number of lambda was 8, and every 10 generation, the number of lambda reduced to half until it became one. The number of lambda and reduction frequency were determined by grid search to maximize the performance while other hyperparameters were fixed. The procedure for validation was identical to the above.

4.3.5 Validation of Optimization: Human Subject Experiment

Experimental Protocol

The purpose of the experiment was to validate the proposed optimization algorithm on human subject testing (Fig. 4.2). To this end, subjects provided their pairwise preferences by interacting with a touch screen tablet installed to the instrumented treadmill (Fig. 4.5), while control parameters were optimized in real-time. Subjects walked on the treadmill on a fixed speed of 1.2 m/s. A pair of controller setting was presented to the user using a touch screen, and user provided their preference via the buttons on the user interface on the screen (Sec. 4.3.5). As user pressed the button, the user experienced the change to the corresponding setting. Typically it took 3-4 steps before the change was applied, which we informed the users prior to the experiment to account for delay before making a selection. The change of assistance was prescribed at the heel contact of the following side of limb, right after the button was pressed. Once the users knows which setting they prefer, they confirmed the setting via the user interface. The users were instructed to try out the settings as much as possible without any time limit until they are certain which setting they prefer. After user repeated the selection for total 50 times, user entered a validation trial without being aware of it. We fixed the number of generations to 50, where the number was decided based on pilot study that was known to reach convergence. The optimized parameter was defined as the last parameter user selected before entering the validation. For validation trial, the protocol was identical to the main optimization trials, except one of the two settings corresponded to the optimized parameter while another was a randomly generated control parameter. There was total 10 selections during the validation trial.

The protocol composed of two practice sessions, and three main optimization sessions

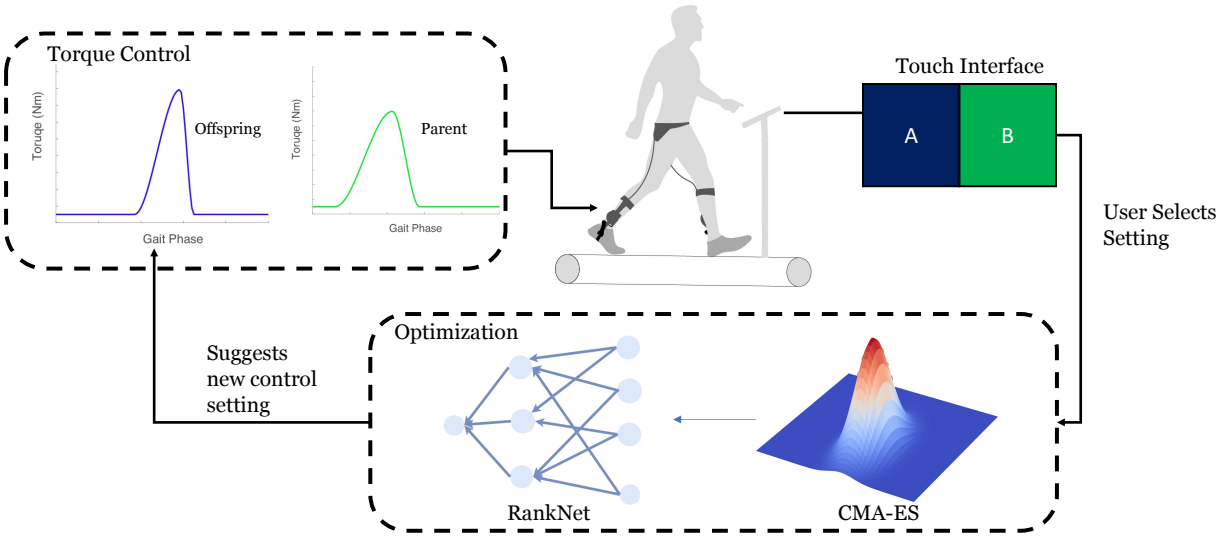

Figure 4.2: Experimental setup for preference-in the-loop-optimization. A method which optimizes users preferred torque profile by using pairwise preferences. Given two controller settings (parent, best offspring), user provides their selection of preferred setting using a touch screen interface. Using user feedback information, CMA-ES updates parameter accordingly and generates new set of offsprings consists of control parameters. Ranknet sorts the offsprings and present the best one to the user with the parent.

consists of optimization and validation trial as described above (Fig. 4.3). During first practice session, subjects donned the ankle exoskeleton with actuation off while walking on the treadmill. This session was to make sure the users are adjusted to the treadmill. On the second practice session, subjects walked on the treadmill with the exoskeleton actuation and optimization on while eliciting their preferences via tablet computer. User was asked to repeat the settings until they are comfortable with the overall interface. After the practice session ended, the main optimization session started. In every 10 minutes and between every sessions, users took break which was notified via the touch screen by prompting a pop-up window. The breaks were designed such that minimize users' fatigue which may detriment the quality of the selection.

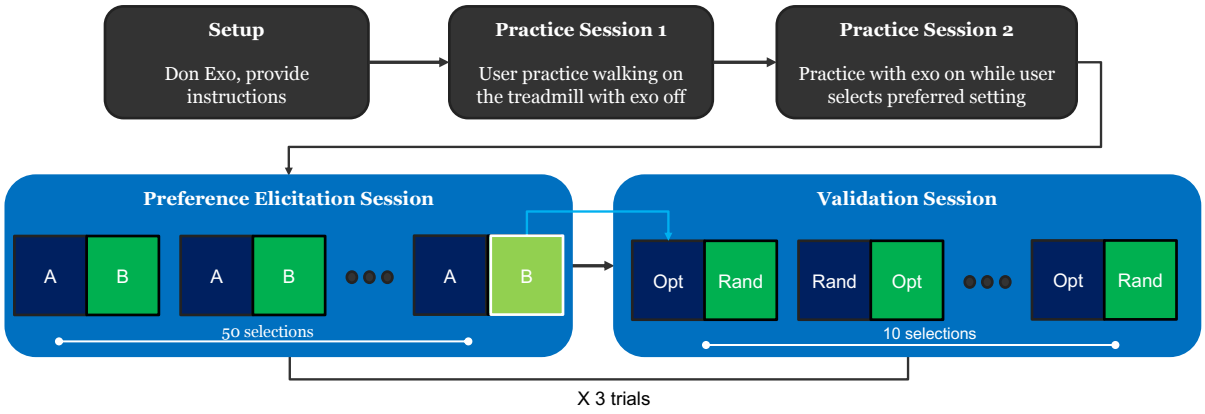

Figure 4.3: Experimental protocol for human subject testing. The protocol consists of setup, two practice sessions, preference elicitation, and validation sessions. The practice sessions were designed so that users become familiar to the instruments and preference elicitation process. During the preference elicitation session, user provided their preference between two settings (A, B). The last selected setting was defined as final optimized setting and passed on to the validation session. Users were not aware of the validation session, where users were comparing between the randomly generated control parameters with the optimized parameters. The sequence of settings in each selection were randomized to prevent users from memorizing the settings.

Subjects

We recruited total 14 able-bodied subjects (10 male and 4 female; average age 26.8 years; height 173 cm, weight 69.3 Kg), which were composed of 8 users with prior experience with a robotic exoskeleton or researchers in the field of wearable robotics (experienced subjects), and 6 users with no prior experience with exoskeletons (naive subjects). The information was self-reported from a questionnaire we provided prior to the experiment. All subjects provided consent to a protocol approved by the University of Michigan Institutional Review Board.

Ankle Exoskeleton System

Subjects donned powered bilateral ankle exoskeleton system (Exoboot-45, Dephy Inc., Maynamrd, MA), where both sides are symmetrical (Fig. 4.4 A). The exoskeleton provides unidirectional actuation, which is able to provide up to peak plantar-flexion torque of 30 N m about the ankle joint. The actuation are provided by brushless motors (T-motor U8 KV 100, Nanchang, Jiangxi, China) by pulling in the cable that is connected to the boot's lever arm. The cable produces a nonlinear transmission (4$\sim$17:1) over the range of ankle angle motion. The torque is transmitted to the leg via shin cuffs and carbon fiber plate embedded in the boots that is rigidly attached to a strut where the actuator is mounted. The exoskeleton is equipped with onboard sensors including ankle joint, motor encoder, IMU, motor current and voltage sensor. Each side of the exoskeleton had a mass of 1.4 Kg, and peripherals including battery (25V LiPo) and the microprocessor (Raspberry Pi 3B+, Cambridge, UK) for controlling the actuator system weighed 1.2 Kg. The microprocessor, which included higher level controller logic, communicated with the embedded system (FlexSEA [47]) of the actuator via USB cables. The actuator system provides Python API (Dephy ActPack) to command actuation and read sensors from the system. The main controller script on the microprocessor ran on 200 Hz, communicating with a closed-loop controller within the motor drive at 1 KHz and 10 kHz for position, and current control, respectively [25].

Ankle Exoskeleton Control

The exoskeleton provided torque as a function of gait phase. Commanded ankle torque (τ_{ankle}) was converted to a desired current (i) using ankle transmission (TR) and motor torque constant (K_T):

$$i = \tau_{ankle}/TR/K_t \tag{4.4}$$

where current and torque constant were represented in q-axis frame. For details on the quantity and the unit please refer to Chapter 1 of the dissertation. The assistance torque profile was shaped using four parameters: rise time, peak torque, fall time, and peak torque magnitude, where the polynomial was defined by interpolation of the parameters (Fig. 4.4 B). The methodology of shaping was originally developed by Zhang et al [20].

We used a finite-state-machine controller API for Dephy Exoboot developed by Shepherd et al [98]. The controller had four states to reflect human ankle joint kinematics. During state 1 (reel-out), the cable was loosen using position controller to reach a desired slack of the cable so that it minimized the resistance in swing-phase dorsiflexion while providing a ground clearance. In state 2 (swing), the cable maintained a desired slack to be much as transparent as possible by exerting zero torque. In state 3, after the heel strike,

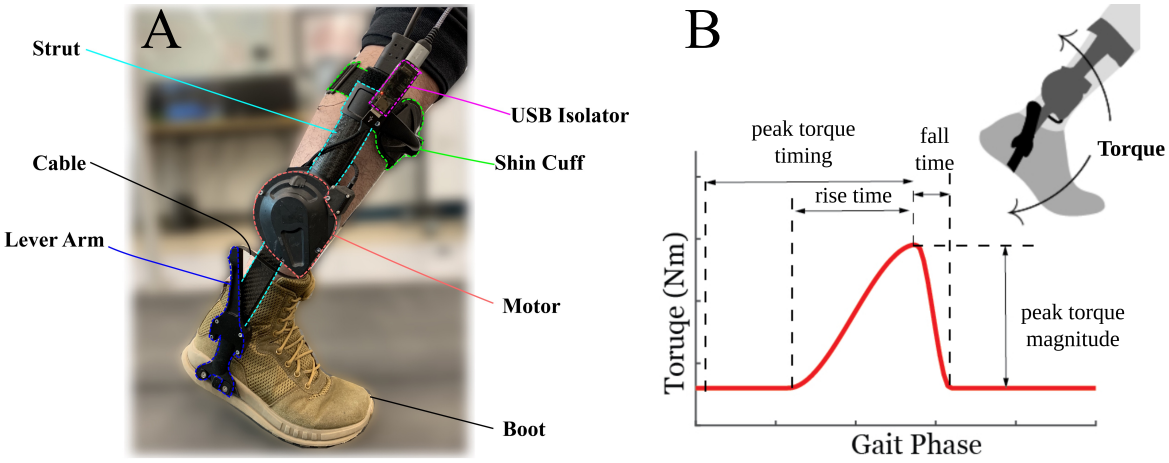

Figure 4.4: (**A**) Bilateral ankle exoskeleton system (Dephy Exoboot). The exoskeleton provides unidirectional assistance through brushless motor and cable transmission during push-off. (**B**) Shaping of the torque profile for controlling the exoskeleton. Four parameters: rise time, peak torque timing, fall time, and peak torque magnitude, determine the shape of a polynomial.

the cable was spooled into the actuator to be taught and ready to prescribe actuation in the following state. In state 4, assistance torque profile was applied using open-loop torque control, where the shape of the profile was defined by the method described in the previous paragraph. During the last state, assistance profile had a bias toruqe of 3 Nm before rise time and after fall time.

The controller parameters were sampled from the CMAES optimization while abiding by the bounds we predefined. The bounds of four control parameters was informed by the literature's on ankle exoskeleton [36, 20]. The bounds for the rise time, peak torque timing, and fall time were from 10-40 % , 30-60 % , 5-20 % gait cycle, respectively, and the bound for the peak torque magnitude was from 7.8 to 20.7 Nm.

Design of Graphic User Interface

The touch screen was composed of a total 4 buttons, which two buttons were 'tryout' buttons (Setting A, Setting B), and other two were 'confirmation' buttons under each tryout buttons (Fig 4.5). Two settings were provided by the optimizer, where one was parent, and another was offspring selected by Ranknet (Algorithm 1). Each tryout buttons of setting A and setting B corresponded to a torque assistance setting of parent and offspring. Whenever the user selected any of the tryout buttons, the respective button glowed to show the user which setting they were experiencing. There was no limit on how many times users can press these tryout buttons, since we allowed users to try as many times they want before making a decision which one they prefer. Once the users are certain which control setting they preferred, they pressed the respective confirm button. Based on the selection made by the user, in the next pairwise comparison, the non-confirmed setting was replaced by a set of control parameters recommended by the optimizer (*i.e.* offspring). The placement of the previously selected control setting and the newly recommended control setting was randomized so as to avoid scenarios where users remember

the order of the placement. In case of validation trial, one of the tryout buttons correspond to the optimized parameter, and another was the randomly generated parameter while the procedure was identical to that of the main optimization trial as discussed above.

When users made a decision on which settings they preferred, user clicked on button "confirm A" or "confirm B". Once user pressed the confirm buttons, a pop up window appeared on the screen with a button "Click to Start Next Comparison" – by clicking this button, the users started the next comparison and were presented with the same screen with the 4 buttons. This pop up was added so that users could drive the pace of the experiment. To prevent cases where user accidentally presses confirm before exploring two settings, a warning pop-up window was prompted to the user in this scenario – a pop up appeared on the screen with the message "Please explore both the control setting

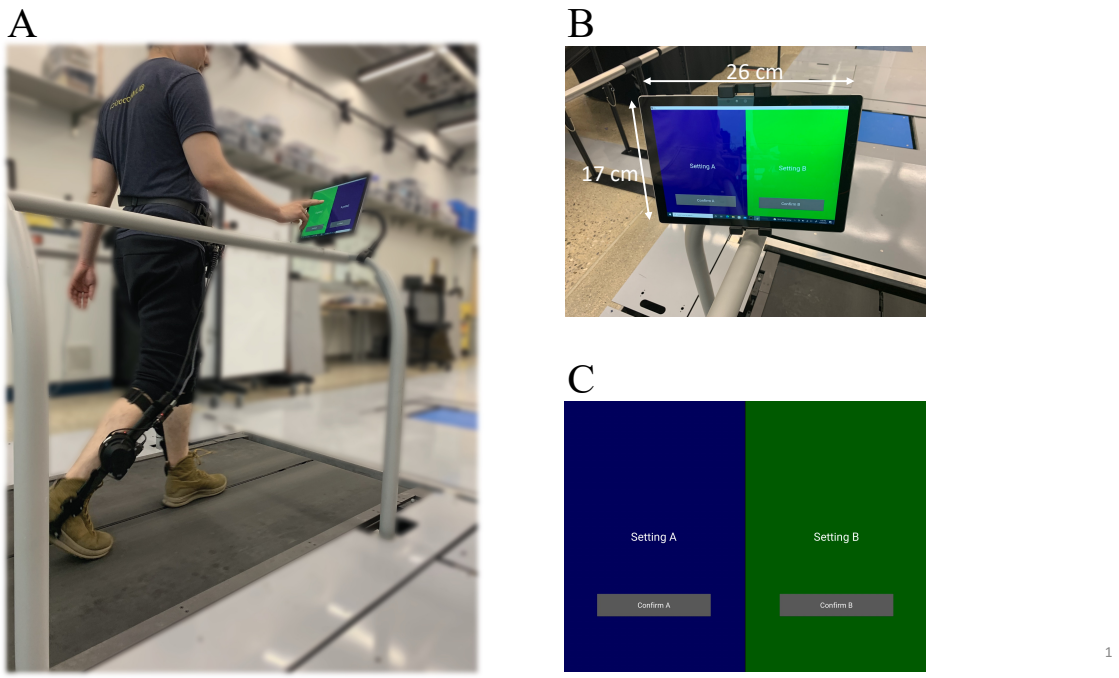

Figure 4.5: Setup of a graphic user interface (GUI). (**A**) A picture of subject interacting with the GUI while walking on a treadmill with the exoskeleton. (**B**) The placement and dimensions of Microsoft Surface tablet where the GUI was presented. (**C**) Buttons on the GUI: Setting A, Setting B, confirm A, confirm B. Setting buttons allowed users to try out controller setting corresponding to each button. Once user decides which setting they preferred, they clicked the according confirm button under the setting.

before pressing confirm". This ensured that the users experience both the control settings before making a selection. Additionally, a pop up appeared with the message "Please reset the Exoskeleton, Break Time!" and a button which enabled to rejoin the session every 10 minutes when the break time was reached. Also a pop up appeared on the screen with the message "End of Trial!" after the completion of a single session (*i.e.* 60 selections).

Design of Communication System

We designed a multi-device wireless communication system for real-time optimization while user is in the loop (Fig. 4.6). We used gRPC Remote Procedure Call (gRPC) for communication between the three devices: Microsoft Surface tablet PC (GUI), Computer, Raspberry Pi (RPi). gRPC is a communication framework which is lightweight, environment agnostic and efficient, which is suitable for our application. When the user makes a selection (*i.e* clicks button on the touch screen), the GUI sends the user's selection to the computer via a wireless gRPC message. Based on the selection received from the GUI, the desktop computer runs the optimization and sends the control parameters to the RPi. The RPi updates the controller parameters based on the control parameters received from the computer. The RPi sends the control commands to the embedded system (FlexSEA [47]) on the Exoboot via a USB connection, and the embedded system sends the motor commands to the brushless DC Motor on the Exoboot. When one session block ends, (*i.e.* user has made 60 selections), the computer sends a message to GUI to to appear the pop-up window on the screen to inform the user that session has ended. Additionally, once break time is reached (every 10 minutes), the RPi sends a grpc message to the GUI to make another pop-up appear.

To have a seamless real-time communication between the three device, we used multi-threading to run multiple process in parallel. The controller on the RPi was running two processes simultaneously, one to continuously listen to the message which contains the

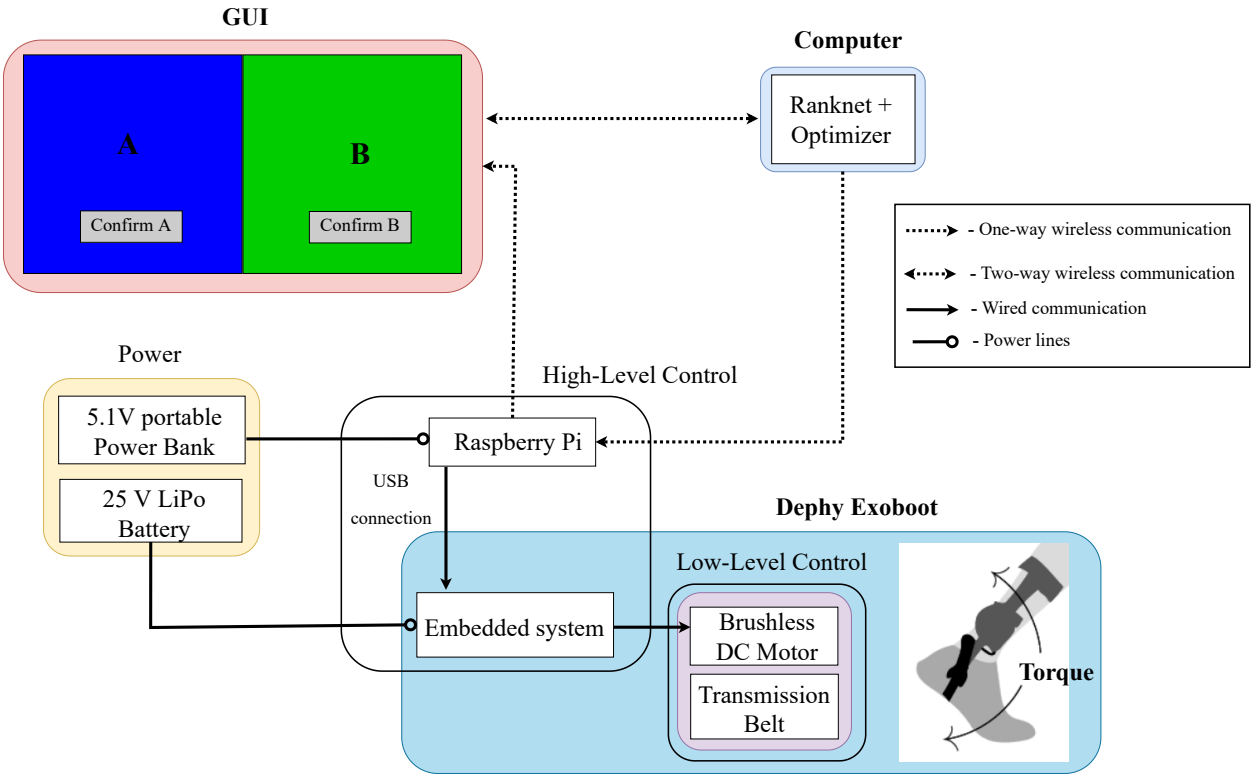

Figure 4.6: Schematic diagram of the communication system. The figure shows the communication loop between three devices: Microsoft Surface tablet PC (GUI), desktop computer and Raspberry Pi+Exoboot.

control parameters sent by the computer, and another process to send the control commands to the embedded system. The surface tablet also was running two processes simultaneously, one to display the GUI interface on the screen, and another to continuously listen to the messages sent by the RPi and desktop computer to indicate that the break time / end of session has reached.

Statistical Analysis

To quantify the success of the optimization in human subject testing, we used a metric of validation accuracy – number of times user selected optimized torque control parameters over randomized parameters. During the main optimization sessions, we recorded the control parameters during all generations including that of validation trial. First we

analyzed the validation accuracy by calculating the mean and standard deviation across trials per subjects. In addition, we compared the validation accuracy between naive and knowledgeable subjects by using two-tailed t tests. The t tests were calculated across all trials for both subject groups. The significance level was defined at 0.05

We hypothesized that as randomized profile deviates farther from optimized profile, users will be able to distinguish the profiles better than it is close to each other. To evaluate the hypothesis, we used a linear mixed-effects models to measure the relationship between RMSE of randomized and optimized profile, and number of times user selected randomized over optimized profile (NTSR). NTSR was used as response variable, RMSE as a fixed-effects term, and a random intercept that varies by subject. We calculated the significance of the fixed effect coefficient to assess the statistical significance of RMSE on NTSR. The significance level was defined at 0.05. All statistical calculation were performed using MATLAB (MathWorks, Natick, MA).

4.4 Results

4.4.1 Learning Preference Function: Training Ranknet

We visualized Ranknet which was trained using the dataset as described in Sec. 4.3.2. Estimated preference landscape from Ranknet had high score around peak torque timing larger than 55 % gait cycle, in accordance with the dataset where most of the users' preferred settings are clustered (Fig. 4.7). This means that during optimization, Ranknet will sort offsprings based on the order in which has higher peak torque timing.

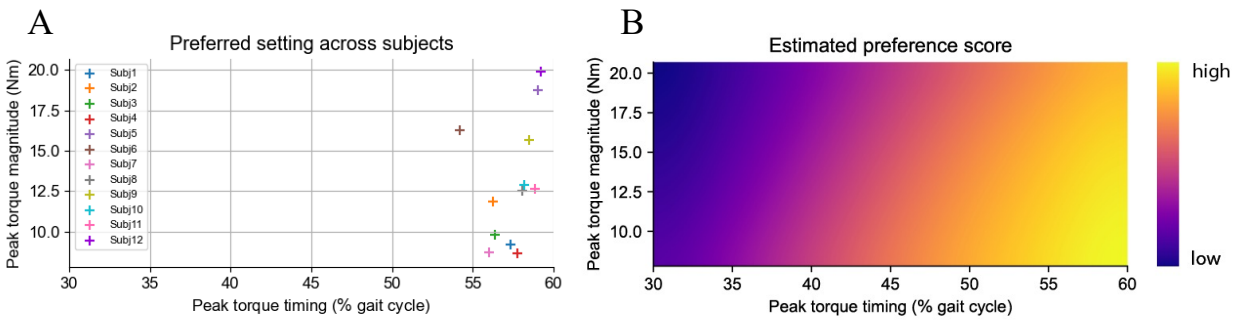

Figure 4.7: (**A**) Users' preferred controller settings in previous work[36]. Note that the controller settings are two dimension: peak torque timing and magnitude, and users are different from the subjects participated in this study. (**B**) Estimated preference score from Ranknet. In accordance with subplot A, scores are higher in the region where users settings are clustered.

4.4.2 Validation of Optimization: Simulation

We analyzed the performance of *RankCMAES* in simulated environment. The proposed algorithm outperformed baseline 1+1 CMA-ES in 12 cross-validation. In general, *RankCMAES* the performance gap between two algorithms occurs around $\sim$ 30 generations, where Ranknet assists the CMA-ES optimization. This highlights the effect of Ranknet in the optimization. In addition, we studied the effect of lambda scheduling in performance of *RankCMAES*. We can observe that the effect of scheduling is paramount, when there is no scheduling, *RankCMAES* suffers to reach the global optimum.

4.4.3 Validation of Optimization: Human Subject Experiment

We calculated the validation accuracy rate of users selecting optimized control parameters over randomly generated parameters. First, we analyzed the mean and standard deviation across trials per subject (Fig. 4.9 A). We noticed subjects with lower accuracy had higher

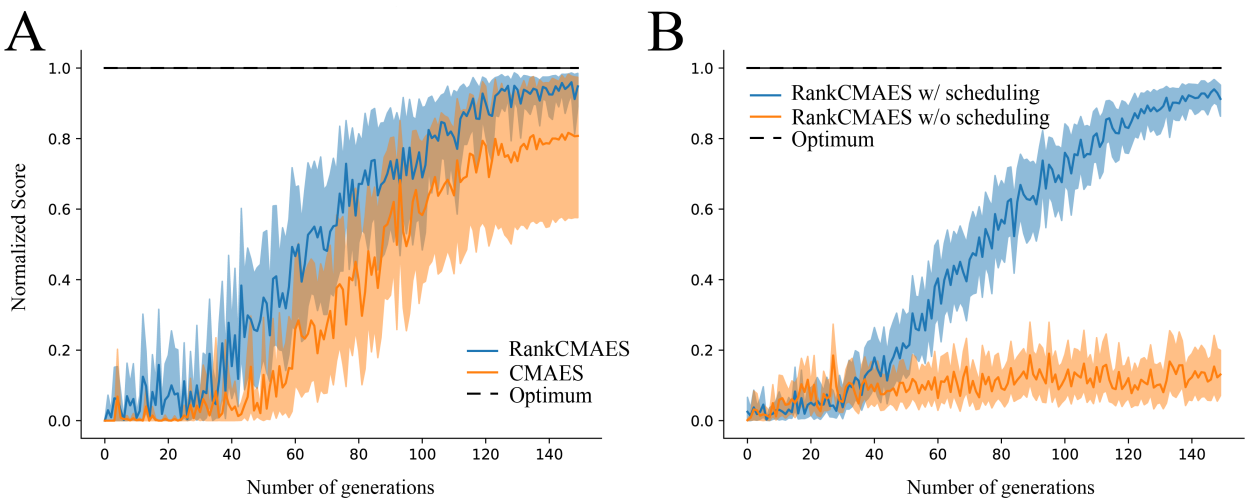

Figure 4.8: Performance validation of *RankCMAES* in simulation. (**A**) Effect of Ranknet. Comparison of optimization results between *RankCMAES* and 1+1 CMAES. *RankCMAES* (blue) outperforms baseline CMAES (orange). The dashed line shows the score of global optimum (upper bound). (**B**) Effect of lambda scheduling. Comparison of optimization performance when the scheduling is off verses on. *RankCMAES* with scheduling (blue) outperforms *RankCMAES* without the scheduling (orange).

variance. Overall, the average and standard deviation across all subjects were $87.8 \pm 10.6\%$, which shows that users were able to discern and select optimized parameters. Secondly, we calculated the mean and standard deviation of validation accuracy across knowledgeable and naive subject groups (Fig. 4.9 B). The accuracy for knowledgeable subject group and naive subject group were $93.8 \pm 8.8\%$ and $80.0 \pm 17.1\%$, respectively. Knowledgeable subjects statistically performed better in validation trial than naive subjects ($P = 0.001$).

To further check the plausibility of the validation trial, we analyzed the correlation between deviation of optimized and randomized torque profile and number of times user selected randomized over optimized profile. To measure the deviation, we used root-mean-square-error (RMSE) between optimized and randomized profile. We binned the range of RMSE into three groups: RMSE of 0-3, 3-6, 6-9, to calculate the number of times user selected randomized profile in each group. Since RMSE is a continuous variable, it is nearly impossible for a RMSE to have more than one corresponding counts of selection, which necessitates to group RMSE of certain ranges. The number of times user selected

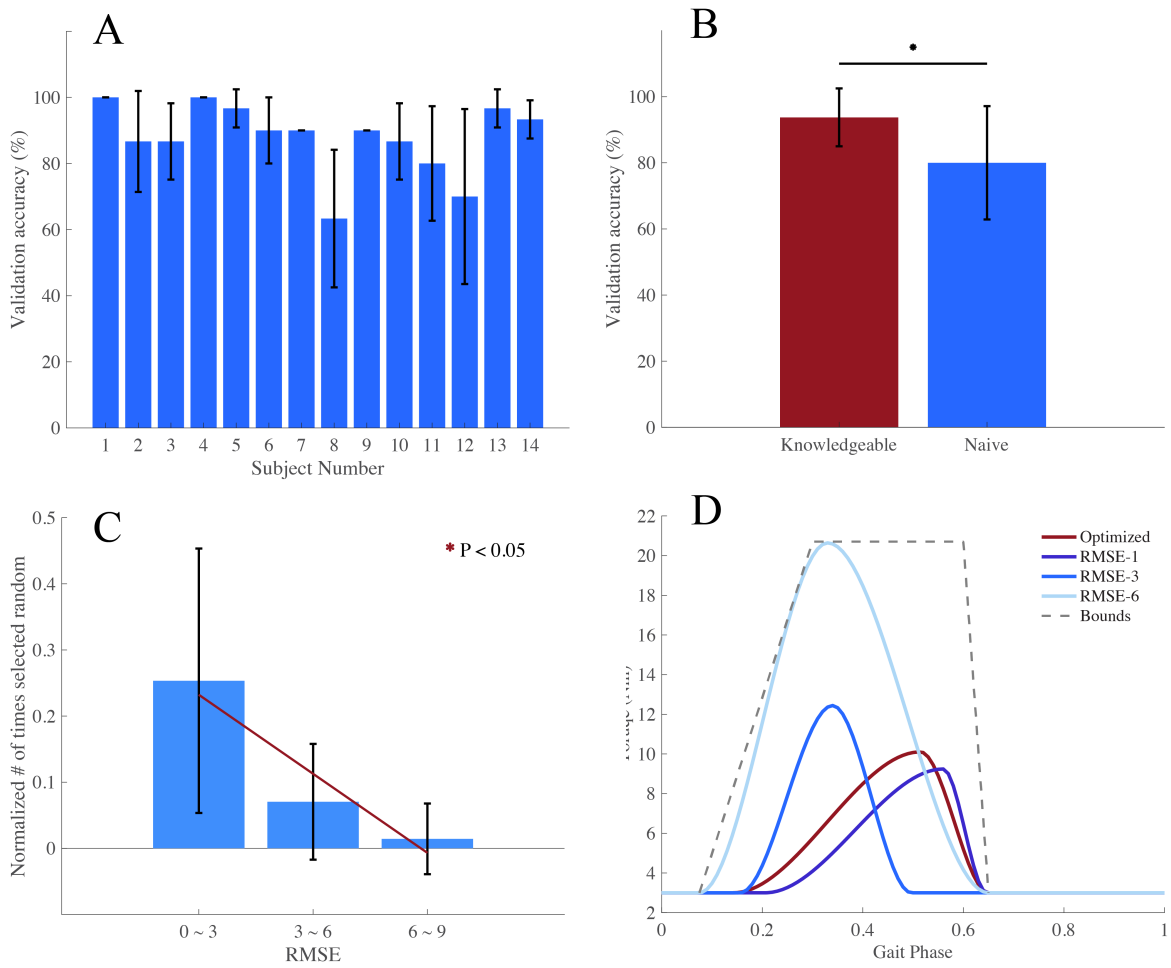

Figure 4.9: **(A)** Mean and standard deviation of validation accuracy across trials for all subjects. **(B)** Mean and standard deviation of validation accuracy across knowledgeable and naive subject groups, respectively. Error bar denotes a standard deviation, and star shows a statistical difference ($P < 0.05$). (C) Mean and standard deviation of normalized number of times user selected randomized profile over optimized profile (NTSR) across subjects. NTSR was measured by comparing between randomized and optimized profile in torque-gait phase space during validation trials, and grouped in three different root mean square error (RMSE). Line depicts the slope of linear-mixed effects models. (D) An example plot of torque profiles which represents each RMSE group..

randomized profile was normalized (NTSR) to the total number user selections per RMSE group to account for the stochasticity of sampling process of randomized profile. We analyzed the effects of RMSE on NTSR using linear mixed-effects model (LMEM). The model showed negative linear significant relationship between RMSE and NTSR ($P < 10^{-6}$), where the level of significance was defined at $\alpha = 0.05$ (Fig. 4.9C).

We measured and visualized the average of the optimized control parameters across trials for individual subjects. Overall, users had distinctive optimized parameters over the range of permissible parameters (Fig. 4.10A). For peak torque timing, the parameters ranged from 40 to 58 % gait cycle, while most of the subjects were between 53 to 58 % gait cycle except two subjects. For peak torque magnitude, the parameters were between 12 to 20 Nm. The optimized rise time ranged from 20 to 33 % gait cycle, while the optimized fall time ranged from 6 to 30 % gait cycle. Additionally, we visualized the optimized torque profile for all subjects. We were able to observe that optimized profiles covered wide area within the allowable range of torque; however, peak torque timing were rather concentrated in relatively narrow range across subjects.

We visualized the progression of optimization for one representative subject (no. 7, trial 3). We depicted the progression of selected torque profiles (Fig. 4.11 A), and selected and non selected profiles (Fig. 4.11 B) from start to the end of the optimization (50 generations). We were able to observe that the users explored wide range of profiles before converging to the optimal profile. During the validation trial, user mostly selected the optimized over randomized parameters (Fig. 4.11 C), except one case where randomized profile was close to that of the optimized (Fig. 4.11 D).

4.5 Discussion

In this work, we proposed a sample-efficient active-learning strategy for optimizing users preference while human is in the loop. To this end, we described the novelty in the algorithm, and validated the performance in both simulation and human subject experiment with a robotic ankle exoskeleton. The overall validation accuracy in human subject testing were in average 88% across subjects. This accuracy indicates that user were able to reliably distinguish and select preferred settings, and the algorithm were able to iden-

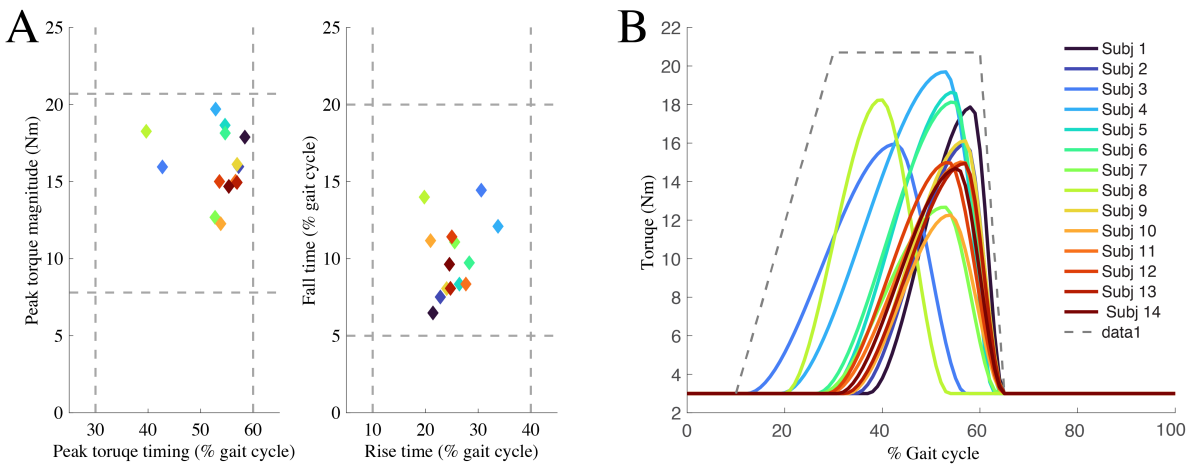

Figure 4.10: (**A**) Users optimized control parameters averaged across trials. Left plot shows the control parameters of peak torque timing and magnitude, while right plot depicts the optimized control parameters of rise time and fall time. The gray lines depicts the bounds of each control parameters. Different color codes represents different subjects which corresponds to the subject number in subplot B. (**B**) Average torque profiles of all subjects. The toe-off point was at 65% of the gait cycle, where the profile was not permitted to prescribe larger than the bias torque (3 Nm). The gray lines depicts the bounds of the profile imposed by control parameters.

tify users' preferred control parameters. We also studied whether knowledge in wearable robotics increases the validation accuracy, where knowledgeable users had statistically significant higher accuracy. The motivation of this work is to provide a framework for automatically tuning users preference when users need to. Our algorithm enables for users to tune the controller in real-time, by asking a simple binary question for current pair of settings, without requiring users to remember and keep track of the previous settings they have selected. This simple form of feedback significantly reduces the cognitive burden for eliciting preferences. The work lays a cornerstone for automated tuning system of powered exoskeleton based on what people want, which can attain multiple objectives such as comfort, balance and stability without the need of extra equipment. We believe that a user preference has a strong potential to bring these emerging wearable robotics technology outside of the lab, in community and at home.

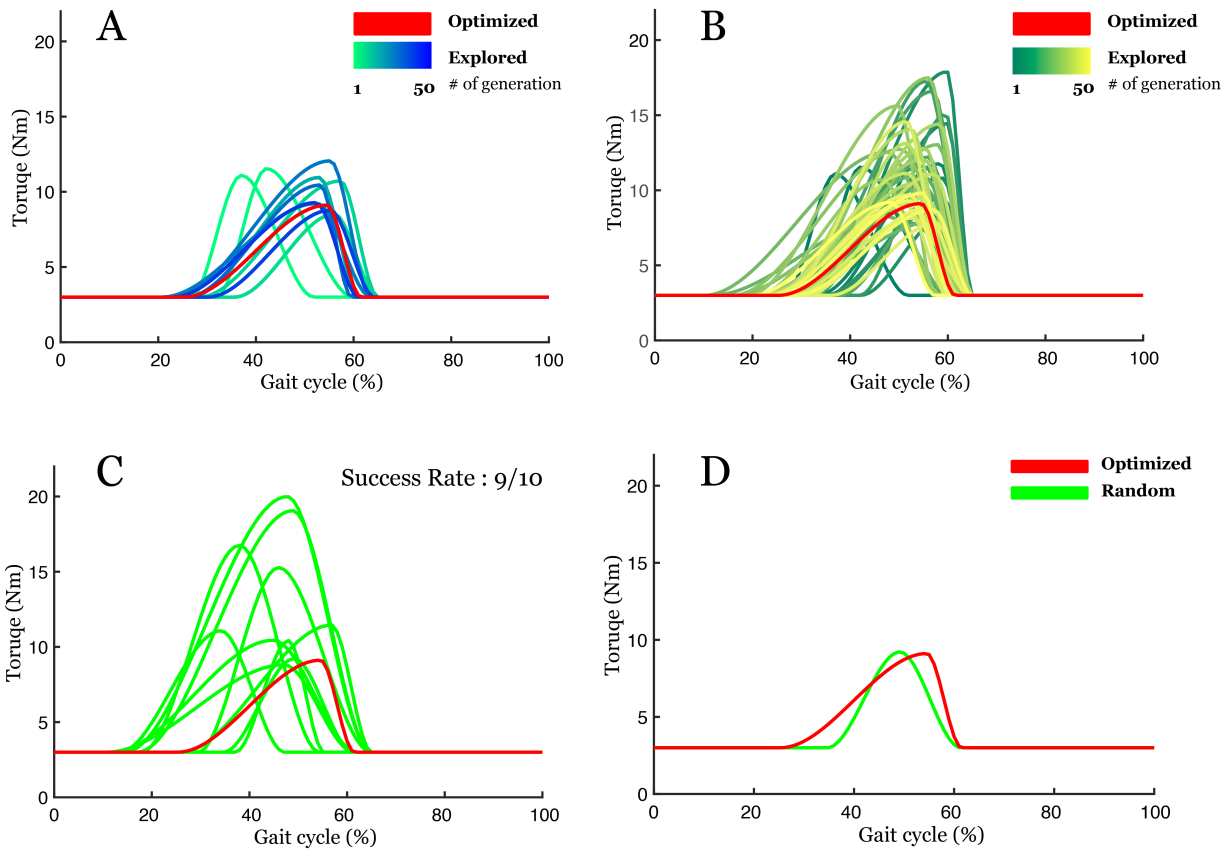

Figure 4.11: Progression of torque profiles of one representative subject (no.7, trial no.3) during main optimization trial (**A**) Selected torque profiles across number of generations. Red line depicts the optimized profile (a profile user selected at the end of the optimization) (**B**) Selected and non selected torque profiles across number of generations. (**C**) Torque profiles across validation trial. Green line represents randomly generated torque profile. (**D**) Torque profiles when user selected randomized profile over the optimized. For this particular subject, user selected the randomized (green) over optimized (red) for only one time during the validation trial.

4.5.1 Limitation

The human preference model (*i.e.* Ranknet) is pretrained and fixed during the optimization procedure. In other words, the model is not updated based on a novel user. *RankCMAES*, which is rooted on CMAES, still optimizes and adapts to a new user; however, since the model is fixed, a novel user's preference landscape is unknown. Currently, the model creates a single model across multiple user information, in other words, there is no user-specific information contained in the model. This can potentially be an issue when a new user does not follow the trend from the model. For example, subject no. 8 showed a low optimized peak torque timing (38%) compared to other subjects $(\sim 55\%)$. In general, Ranknet recommends settings that has higher peak torque timing during the optimization process (Fig. 4.7), which may hindered the performance of the algorithm. Therefore, this particular subject may have converged to a sub-optimal setting which resulted in the lowest validation accuracy among subjects.

4.5.2 Validation of Optimization

The validation accuracy of $\sim$ 90% show that the proposed algorithm were able to optimized the control parameters based on user preferences. During validation trial, users were able to perceive the differences between optimized and randomized control parameters, while able to identify the preferred settings repeatedly. The blindfold validation test we designed is a way to indirectly validate the performance of the optimization, but it does not provide information on how close the optimized settings are to the global optimum in parameter space. To mitigate the shortfall of the validation test, we further analyzed how the accuracy (NTSR) decreases as the randomized parameters deviates from optimized parameters (RMSE). The significant negative linear relationship between two parameters: NTSR, RMSE, demonstrates that as the deviation of randomized and optimized parame-

ters increases, users were able to identify the difference better. This implicitly manifests the trend of the preference landscape, where hints the convexity of it. However, to accurately identify a novel user' landscape, a preference model should be fitted to the data collected from the novel user as described above.

We further analyzed the validation accuracy by dividing users into two different groups based on prior knowledge to exoskeletons (knowledgeable vs naive). We observed that knowledgeable users performed statistically better than naive users. This may indicate that knowledge on exoskeletons contribute to sensing and identifying preferred control settings. Similar study was conducted previously from our research lab, where the precision in identifying preferred settings was compared between two groups [36]. Although there were no statistical significance observed between two groups, knowledgeable group had higher precision in average. This finding supports the possibility of knowledgeable user having higher sensitivity than naive users on controller settings; however, separate sensitivity analysis should be performed to verify the hypothesis. There has been a study which investigated into measuring the sensitivity using the just-noticeable difference on powered ankle exoskeleton, but it did not distinguish the group of subjects based on prior knowledge on exoskeletons [21].

4.5.3 Future work

One way to address the limitation of the fixed preference model is to update or newly train Ranknet based on new users' preference feedback; however, the challenge is to carefully choosing the frequency of update, since we need to have abundant amount of data to train neural networks, and updating too frequently may lead to catastrophic forgetting. In addition, we can devise a neural-network architecture which contains user-specific information. By having additional user ID as an input to the network, the model may be able

to adapt to a novel user (*i.e.* meta-learning).

One of the novelties in our approach is that we trained and used Ranknet in lower dimension (two) and used for higher dimension (four) optimization problem. This demonstrates two advantages: 1. Utilizing pre-existing, non-pairwise preference data [36]. This removes a burden of collecting a new dataset, which is costly for human testing, and we can convert into the data format we desire and use it in our optimization. 2. Extensibility to a higher dimension optimization. Similar to the study, we can reuse the data we collected for four dimension and apply it to higher dimension problem. In other words, the preference feedback we collected during the optimization experiment, can be a new source of data for optimizing higher dimension control parameters. This does not mean that the algorithm cannot use Ranknet for the same dimensional optimization problem. To use Ranknet which is on the identical dimension, a data collection must be conducted as a *priori* procedure for training Ranknet.

Chapter 5

Image Transformation and CNNs: A Strategy for Encoding Human Locomotor Intent for Autonomous Wearable Robots

5.1 Abstract

Wearable robots have the potential to improve the lives of countless individuals; however, challenges associated with controlling these systems must be addressed before they can reach their full potential. Modern control strategies for wearable robots are predicated on activity-specific implementations, and testing is usually limited to a single, fixed activity within the laboratory (*e.g.* level ground walking). To accommodate various activities in real-world scenarios, control strategies must include the ability to safely and seamlessly transition between activity-specific controllers. One potential solution to this challenge is to the infer wearer's intent using pattern recognition of locomotion sensor data. To this end, we developed an intent recognition framework implementing convolutional neural

This chapter is currently published in IEEE Robotics and Automation Letters (Volume: 5, Issue: 4, Oct. 2020).

networks with image encoding (*i.e.* spectrogram) that enables prediction of the upcoming locomotor activity of the wearer's next step. In this paper, we describe our intent recognition system, comprised of a mel-spectrogram and subsequent neural network architecture. In addition, we analyzed the effect of sensor locations and modalities on the recognition system, and compared our proposed system to state-of-the-art locomotor intent recognition strategies. We were able to attain high classification performance (error rate: 1.1%), which was comparable or better than previous systems.

5.2 Introduction

Wearable robots, including powered prostheses and exoskeletons, have the potential to improve people's quality of life by enhancing their physical capabilities during locomotion [16, 25]. Despite the promise of these wearable technologies, challenges remain in the development of safe, intuitive, and versatile control systems. Recently, researchers have demonstrated exoskeletons that are able to apply substantial assistance, as well as reduce the metabolic expenditure during walking [16, 20]. To obtain these results, researchers typically develop control approaches that are intended for single activities, often tethered to a treadmill. For this approach to be applicable in daily life, these systems must be able to encompass multiple activities, including walking, running, and stair ascent or descent. To address the limitations associated with control systems meant for single activities, some researchers have developed methods for switching between multiple activity-specific controllers; however, often these transitions are initiated by commands such as visual, auditory, or touch (*e.g.* key-fob) cues which are non-intuitive and can increase cognitive burden [40]. Thus, for users to naturally perform the activities of daily life, it is imperative to develop control strategies that can infer the wearer's intended movement automatically without requiring external commands, and autonomously transition between

different activity-specific controllers.

One approach to infer the wearer's intended activity is to use an intent recognition framework [40, 44, 42]. Intent recognition typically includes predicting the upcoming activities of the user each step using information from the wearer, robotic system, or environment prior to completing the movement (*e.g.* before heel contact or toe off of the current step) [41]. There have been several works that implemented intent recognition strategies employing sensor fusion for improving the performance [44, 99]. While these strategies demonstrated high performance on classifying users' locomotor activities (error rate $< 2\%$), they often rely on hand-crafted features, such as the mean, standard deviation, maximum and minimum of time-series data. This can be challenging because it may require domain specific knowledge and trial and error approaches to extract meaningful features [100].

Deep learning (DL) has been emerging as a tool to classify activities in human activity recognition (HAR) or intent recognition tasks [101, 102, 103]. Especially, convolutional neural networks (CNNs) have been used over other DL methods, due to their local dependency and scale invariance, which captures the invariant features of the same activities with variations (*e.g.* walking) [101]. Combined with recent advancement in processing capability and miniaturization of graphics processing units (GPUs), CNNs have been extensively employed for mobile and wearable sensors based tasks. To increase the performance of CNNs, several researchers have configured CNN architectures by adding additional layers and nodes or combined with other DL architectures (*e.g.* CNN + Long Short Term Memory) [104, 105]. These approaches can increase the computational complexity, which may not be ideal for low-power on-board sensors or microcontrollers [106]. In addition, due to the increased number of parameters in these architectures, it may be challenging to determine the optimal parameters from relatively small datasets [107].

To obtain better performance while minimizing the computational efforts, researchers

in HAR have investigated various techniques for configuring input data, such as linear interpolation, distance matrices, *etc.* [108]. Among these techniques, the use of image transformation (*i.e.* 2D representation) of time-series data as an input to CNNs, have been employed for classifying activities [102]. Especially, conversion to the spectrogram captures frequency features of the signals and is robust against variance of sampling rate [106]. While these image configuration techniques have achieved promising results, many researchers have focused on classifying the activity *after* the movement completion, rather than predicting the activity *before* the completion (*i.e.* intent). Specifically for HAR tasks of walking activities, researchers have focused on classifying the activity of the current or past step, rather than the activity of the subsequent step of the gait cycle [101, 106]. CNNs have been used to predict locomotor intent for use in powered prostheses; however, they either have directly applied time-series data or hand-crafted features as an input to the CNN, which resulted in similar or inferior performance compared to the feature-based classifiers (*e.g.* Linear Discriminant Analysis) [103, 109, 110]. Thus, the impact of these image configuration methods applied to CNNs for locomotor intent recognition tasks remains unknown.

The contributions from this paper include: (1) We propose a spectrogram-based CNN recognition framework for predicting the *intent* of the lower-limb locomotor activities. Inspired by [106], we modified this approach to be suitable for our tasks by developing an analysis pipeline composed of a lightweight neural network architecture and a mel-scaled spectrogram. (2) We compared the performance of our system to the state-of-the-art (SOTA) locomotor intent recognition strategies using bilateral neuromechanical signals. The proposed system achieved a classification error rate of 1.1%, which outperformed or was comparable to previous works [44, 110]. (3) We characterized the effect of sensor locations and modalities on the classifier performance; finally, (4) we qualitatively identified the region of the gait cycle responsible for the intention by visualizing the activation

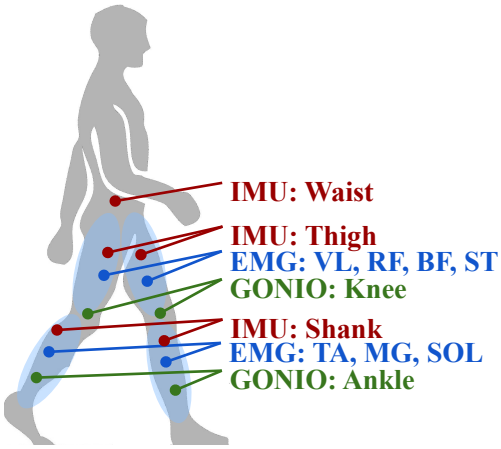

Figure 5.1: Instrumentation setup for the dataset.The EMG electrodes were placed on seven muscle groups responsible for lower limb locomotion: tibialis anterior (TA), medial gastrocnemius (MG), soleus (SOL), vastus lateralis (VL), rectus femoris (RF), biceps femoris (BF), and semitendinosus (ST). The GONIOs were placed on knee and ankle joints and the IMUs were placed on the thigh and shank to measure angular position and velocity. [111]

of the CNN. To our knowledge, this is the first work to use CNNs with image encoding of frequency content for lower-limb intent recognition with bilateral neuromechanical sensor fusion. The intent of our work is to enable future wearable robotic technologies to be used outside the laboratory, where a diverse range of activities is required.

5.3 System Design

5.3.1 Dataset

We used a publicly available dataset composed of kinematic and muscle activity signals to train our intent recognition framework. The dataset, named as the Encyclopedia of Able-bodied Bilateral Lower Limb Locomotor Signals (ENABL3S), was chosen over other datasets (*e.g.* UCI-HAR [112]), because it focuses on normal locomotion, includes rich biomechanical signals from multiple sensor modalities, and sampling rates are sufficient for online control purposes. The data were collected from wearable electrogoniometers

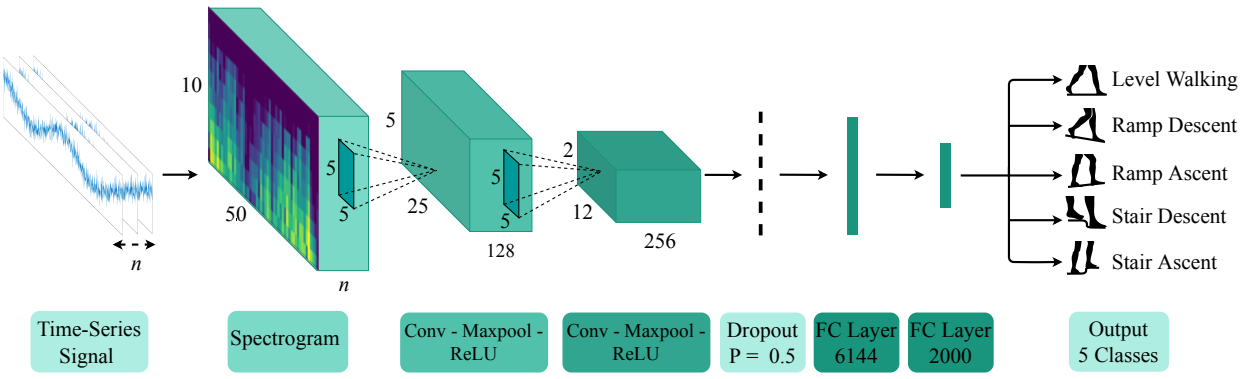

Figure 5.2: Our proposed intent recognition pipeline from frequency domain representation, CNN architecture to the output activities. Convolutional layers consisted of kernels sized of 5x5, stride of 1, padding of 2, and two sequential linear layers had hidden units of 6144 and 2000, respectively. The dropout with 0.5 probability was added to improve the generalizability of the proposed system.

(GONIO), surface electromyography (EMG) and internal measurement unit (IMU) sensors. The sampling rate of EMG, GONIO and IMU sensors were 1000, 500, and 500 Hz respectively and low-pass filtered at 350, 10 and 25 Hz respectively. All sensor data were processed to identify right and left heel contact and toe off (*i.e.* gait events). These sensors were placed on the lower limbs of 10 able-bodied human subjects (Fig. 1). Each subject performed 25 repetitions of a circuit consisting of walking on level ground (LW), ascending/descending a ramp with a 10 degree incline (RA/RD), and ascending/descending a four-step staircase (SA/SD). The odd-numbered trials had a sequence of these activities as follows: LW→ SA→ LW→ RD→ LW, while even-numbered trials had LW→ RA→ LW→ SD→ LW. The ground truth label was marked by the experimenter using a key fob. The preceding 500 ms of sensor data before each gait event was used as the input to our analysis pipeline and the activities after each gait event (*i.e.* upcoming activity) were used as the label for prediction [111].

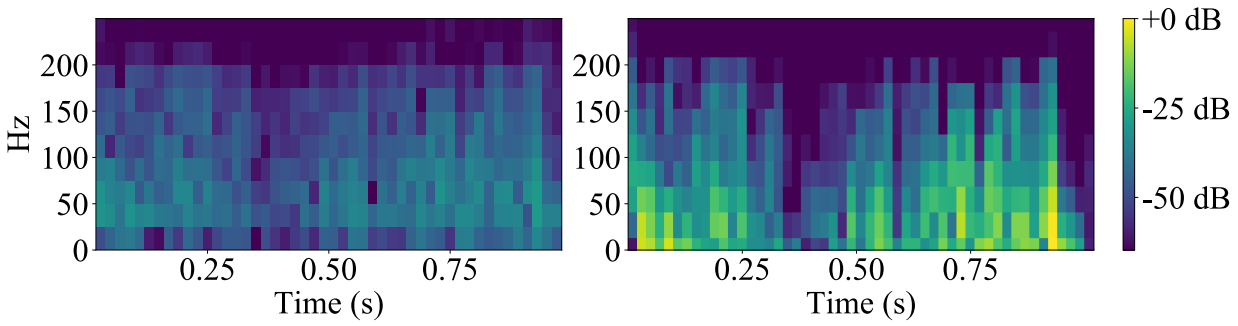

Figure 5.3: A sample spectrogram (left) and mel-spectrogram (right) generated from the EMG signal of the right TA. Lower frequency signals were amplified from the mel scale conversion.

5.3.2 CNN-based Intent Recognition

Image Encoding using Spectrograms

Due to the periodic nature of walking, we propose that the frequency domain information from the time-series data provides a more effective representation of lower-limb locomotor activities for CNN classification. To produce the frequency-domain representation, the Short-Time Fourier Transform (STFT) was performed on time-series data:

$$\mathrm{STFT}(x[n]; w, k) = \sum_{n=-\infty}^{\infty} x[n]w[n-k]e^{-j\omega n} \tag{5.1}$$

where the signal $x[n]$ was multiplied by a windowing function ω, shifted by an offset k. The squared magnitude of the STFT produced a spectrogram, and we further transformed the spectrogram using nonlinear scaling known as the *mel scale* (Eq. 5.2) which demonstrated its success as a pre-processing step in auditory classification tasks [113]. The mel scale originates from representing the human auditory system such that it has perceptually equal pitch (*i.e.* frequency-scale) increments; in other words, as frequency (Hz) increases, larger intervals of frequency are required to produce the same magnitude of pitch increments. This scaling was chosen so that it amplifies the lower frequency content

of the signal (Fig. 5.3), where much of the content of human locomotion is below 3.5 Hz [114]. The mel scale can be computed using

$$2595 \cdot log_{10}\left(1 + \frac{f}{700}\right) \tag{5.2}$$

Lastly, the amplitudes were squared to further attenuate the higher frequency content and transformed to a decibel scale. For the windowing function ω, a Hann window of length 20, with an offset k of 10 was used. When converting to the mel scale, the Hz scale was partitioned into 10 bins (user-defined) prior to the transformation being applied. The selection of 10 bins was to balance classifier performance and processing overhead. All the steps outlined in this section were performed using the LibROSA package in Python [115].

LIR-Net Architecture

We designed the CNN architecture which consists of a series of 2D convolutional layers and pooling layers, followed by fully-connected layers (Fig 5.2). This proposed network is called *Locomotor Intent Recognition-Net* (LIR-Net), which is lightweight but provides high performance when classifying lower-limb neuromechanical spectrogram images. The spectrogram produced in the image encoding step (Section 5.3.2) was provided as an input to the CNN and the softmax operation was applied to the output of the last linear layer, which represented probability distribution of the predicted class.

5.4 Experimental Protocol

The proposed system was evaluated against separate classifier configurations and compared with different classification strategies using the ENABL3S dataset. Furthermore,

we investigated the effect of sensor modalities and laterality groups, where modality describes sensor type (*e.g.* IMU, EMG, GONIO) and laterality describes the side of the leg where a gait event was detected (*e.g.* ipsilateral, contralateral, and bilateral) [111]. Lastly, we implemented feature (*i.e.* unit) visualization of LIR-Net to identify the frequency region of the input spectrogram where the units were most activated.

5.4.1 Classifier Configuration

Generic

A *generic configuration* is defined as when only signal data or features were given as an input to a certain classification strategy without any information (*i.e.* ground truth) from the current activity provided. The *current activity* was defined as the activity before each gait event (*i.e.* before movement completion), whereas *upcoming activity* was defined as the future activity after each gait event.

Mode-Specific

The mode-specific strategy encodes the environment knowledge by providing the information of the current locomotor activity [40]. Specifically, the strategy employs separate classifiers depending on the current activity (*i.e.* mode), which has different number of outputs for each classifier (*e.g.* for the RA classifier, only transition to RA or LW is allowed). Combined with heuristic feature-based classifiers (*e.g.* Linear Discriminant Analysis (LDA), Support Vector Machine (SVM)), this configuration demonstrated low error rates ($< 2\%$) when classifying the locomotor intent [44].

5.4.2 Classification Strategies

Random Guesser

To understand the effect of distribution of the dataset on the classifier configurations, we created a baseline system that predicts activity based on distribution of samples, thereby always predicting the activity with the greatest likelihood.

Generic: Provided only the signal data without activity information, such a system can be represented as follows:

$$\hat{i}_n = \arg\max_i P(i) \tag{5.3}$$

where $\hat{i}, i$ are the predicted class (*i.e.* upcoming activity) and true class label respectively, n represents the nth gait event, and P is the probability distribution of the class i. Therefore, the class with the largest representation (LW, Tab. 5.1) was chosen every time.

Mode-Specific: Given the signal data and the current class of the signal data, we represented a similar system as follows:

$$\hat{i}_n = \arg\max_i P(i|i_{n-1}) \tag{5.4}$$

where P is the probability distribution of the class i given a current activity (i_{n-1}) of nth gait event. For example, given a RA, the classifier only outputs RA since the data distribution of RA-RA is (13%) larger than RA-LW (2%).

Heuristic Feature-based Classifiers

LDA and SVM have demonstrated their validity as classifiers for intent recognition, because they provide low classification error while it is computationally efficient [40, 44].

Table 5.1: Data distribution of Enabl3s

Transition from	to	Number of Samples[#]
Level walking (LW)	LW	8886 (42.87%)
	RA	503 (2.43%)
	RD	474 (2.29%)
	SA	478 (2.31%)
	SD	477 (2.30%)
Ramp ascent (RA)	RA	2740 (13.22%)
	LW	481 (2.32%)
Ramp descent (RD)	RD	3416 (16.48%)
	LW	471 (2.27%)
Stair ascent (SA)	SA	934 (4.51%)
	LW	469 (2.26%)
Stair descent (SD)	SD	925 (4.46%)
	LW	476 (2.30%)

[#] The count of each activity transitions across all subjects.

Especially, LDA combined with the mode-specific configuration achieved SOTA performance (1.43%) for intent recognition tasks [44]. To this end, we used LDA and SVM for our baseline classifiers to be compared with our proposed system. For an input to the classifiers, we used features previously known to be important for intent recognition when controlling powered prostheses. Features were extracted from the time-series data, including mean, standard deviation, maximum, minimum, initial, and final value, *etc* [44, 111].

Generic: The features extracted from bilateral sensor set with all sensor modalities of ENABL3S were provided as an input to the classifier, which were 332 features in total. To be consistent with the existing work, for the LDA classifer, features were normalized and principal component analysis was applied to maintain 95% variance, while the prior was

set to be equally probable. For the SVM, a linear kernel was chosen with a regularization parameter of 10 [44]. The calculation was performed using the Sckit-learn software package in Python.

Mode-Specific: Separate LDA and SVM classifiers were trained to encompass all the gait events and locomotor activities. During the prediction, the mode-specific classifier was selected based on the current locomotion activity. The predictions (*i.e.* output) of the classifiers were limited by the number of transitions allowed on the previous activity.

LIR-Net

Generic: A generic configuration of LIR-Net followed the procedures of Section 5.3.2.

Mode-Specific: To provide the current activity information to the network, we provided the mode information as a one-hot encoding vector and concatenated into the first linear layer of our intent recognition pipeline (Fig. 5.2). We chose this approach rather than explicitly following the conventional mode-specific scheme (*i.e.* training separate classifiers with differing number of output depending on the mode), since the performance of DL will likely suffer from the scarcity of the training samples due to the splitting.

5.4.3 Performance Evaluation

We compared the offline performance using both classifier configurations. We divided the dataset, including all sensor laterality groups and modalities, into testing and training sets, which were divided in two ways: 1. the division was randomized by 10-fold cross validation (90:10 split) within all subjects' data (*i.e.* user-dependent); 2. one out of ten subject's data were withheld as testing set, while the other nine subjects were grouped as the training set. This was repeated 10 times until all subjects were tested once (*i.e.* leave-one-out or user-independent cross validation) [40]. Each classifier was trained on the training set

and evaluated on the testing set. For the user-dependent LIR-Net, the data were divided into training, validation, and test sets (80:10:10), and after finding the best hyperparameters, the validation set was added to the training set. Identical hyperparameters of the user-dependent LIR-Net were used for the user-independent classifier. The error rate of each classifier was determined by the number of correctly classified predictions divided by the number of each test set. Error rates were further categorized based on whether the misclassification occurred at the gait event where the previous and the following activities were identical (*i.e.* steady-state) or different (*i.e.* transitional) [44].

We conducted statistical analyses separately for each error types, and analyzed all classifiers on both classifier configurations and user-dependencies. We used three-way ANOVAs with error rate as a dependent variable, and classifier type, configuration, and user-dependency as independent variables, and subject as a random factor. We performed a *post hoc* comparison test using Tukey's Honestly Significant Difference Criterion (Tukey) to determine the statistical difference between the pairs of interest ($\alpha = 0.05$).

Training of LIR-Net

The network was trained to minimize the cross entropy loss which is described as:

$$Loss(q, p) = -\sum_{i} q(i) \log p(i) \tag{5.5}$$

where the $q(i)$ is the ground truth probability expressed as one-hot encoding and $p(i)$ is the predicted probability of class i. We used a stochastic gradient-based optimizer ADAM [96] with L2 regularization to prevent overfitting.

Hyperparameter Search of LIR-Net

We investigated different hyperparameters of LIR-Net to obtain the best performance. The parameters associated with spectrogram implementation were fixed (Section 5.3.2). A grid search was then performed on hyperparameters to maximize validation accuracy. The hyperparameters found to give greatest accuracy were a batch size of 32, learning rate of 10^{-5}, L2 regularization strength of 10^{-3}, and 200 epochs. All calculations were performed using the PyTorch package [116].

Classification Latency of LIR-Net

The time required for the classifier to make a prediction is a critical factor in the real-time usability of intent recognition systems. To evaluate the latency of LIR-Net, we measured the elapsed time from spectrogram generation to prediction per activity using all sensors. We calculated the latency using a single-board computer (model: Jetson Nano, NVIDIA, Santa Clara, CA) with GPU acceleration. The calculation was repeated 10 times and was averaged to obtain the latency.

Comparison to ResNet

To validate the design choice of LIR-Net architecture, we compared the performance of ours with ResNet18 [117]. We used ResNet pre-trained on ImageNet to perform a fair comparison due to our scarcity of the number of data samples compared to the complexity of ResNet [118]. An identical training and evaluation procedure for LIR-Net was employed to quantify the performance of ResNet. A one-way ANOVA was used to measure the statistical difference between the two architectures.

5.4.4 Effect of Sensor Modalities and Locations on LIR-Net

To determine the effect of sensor types and locations, we trained LIR-Net classifiers on various subsets of the data and compared their performance. The subsets were determined by dividing the sensor data into four modality (EMG, GONIO, IMU, all) and three laterality groups (contralateral, ipsilateral, bilateral). The LIR-Nets were trained on each laterality group with a subset of the four modalities, and their error rates were recorded. The remaining sensor data were withheld during training. We divided all subsets of data into testing and training sets, where the division was randomized by user-dependent 10-fold cross validation.

We conducted statistical analyses of LIR-Nets by using a two-way ANOVA with the overall error rate as the dependent variable, modality and laterality as independent variables, and subject as a random factor. We performed a *post hoc* comparison test using Tukey's Criterion to determine the statistical difference between the pairs of interest ($\alpha = 0.05$).

5.4.5 Visualizing Activations of LIR-Net

Activation (*i.e.* output of the convolutional operations) visualization is a technique that can provide greater understanding of the internal operations of CNNs [119]. To this end, we visualized the activation of the trained LIR-Net after the first convolutional layer, given one sample of spectrogram. This was accomplished by first localizing the maximum value in each output channel (total 128 channels) and mapping each pixel from the first convolutional layer back to the input space (*i.e.* receptive field), where a highly activated pixel is likely important to the CNN. After identifying the mapping between the input and the activations, the receptive field of the input was weighted by the magnitude of its according maximal activation of each channel, and the max activations of all channels were summed

Table 5.2: Error Rates of the Classifiers

	User-Dependent				User-Independent			
	Random	LDA	SVM	LIR-Net	Random	LDA	SVM	LIR-Net
Overall (%)								
Generic	47.99 [1.18]*	6.43 [0.53]*	4.30 [0.61]*	**1.11 [0.26]**	47.99 [2.65]*	12.21 [4.84]*	16.17 [7.27]*	**7.75 [3.84]**
Mode-specific	18.48 [1.03]*	1.85 [0.35]*	2.19 [0.20]*	**1.52 [0.42]**	18.52 [1.72]*	10.52 [4.55]*	9.16 [1.87]*	**7.26 [3.42]**
Steady-State (%)								
Generic	47.42 [1.55]*	1.03 [0.20]	2.54 [0.52]*	**0.54 [0.18]**	47.46 [3.18]*	6.99 [6.21]	14.18 [8.69]*	**5.45 [4.41]**
Mode-specific	**0.00 [0.00]***	1.32 [0.31]	1.19 [0.15]*	0.61 [0.23]	**0.00 [0.00]***	10.08 [6.01]	7.39 [3.03]*	5.14 [3.95]
Transitional (%)								
Generic	50.54 [2.00]*	30.26 [1.77]*	12.05 [1.76]*	**3.64 [0.98]**	50.46 [0.81]*	35.41 [7.67]*	25.08 [4.86]*	**18.08 [6.43]**
Mode-specific	100.00 [0.00]*	**4.18 [1.17]***	6.60 [0.99]*	5.54 [1.94]	100 [0.00]*	**12.37 [4.13]***	17.06 [5.46]*	17.03 [7.96]

Error rates (mean, [standard deviation]) of the generic and the mode-specific classifiers using the bilateral sensors with all modalities. Asterisks under random, LDA and SVM classifiers denotes statistically significant differences between the according classifiers and LIR-Net. The difference between the generic and the mode-specific configurations were all significant across all error types regardless of user-dependencies. Bold numbers represent the classifier with the lowest error rate for each type of error rate and configuration.

and then normalized to create an averaged activation in the input space.

5.5 Results

5.5.1 Performance Evaluation

We evaluated the performances of the classifiers on all classifier configurations and user-dependencies (Fig. 5.4). The interaction between all pairs of classifier types, configurations, and user-dependencies were all significant except the pair of configuration and user-dependency ($p = 0.63$).

Effect of Configuration

Error rates of the classifiers on both classifier configurations were compared (Tab. 5.2). The generic and mode-specific configurations were statistically different across all classifiers and types of error rates. In general, the mode-specific configuration lowered the overall error rates of the random and heuristic-based classifiers, while the change in the error rate of LIR-Net was minimal.

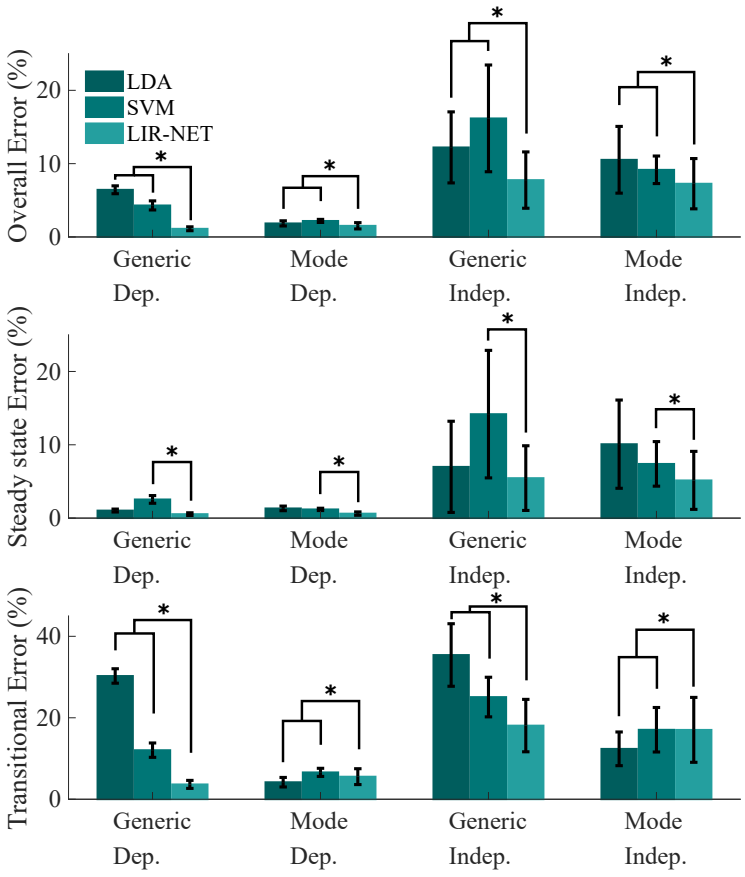

Figure 5.4: Error rates of the classifiers on Generic and Mode-specific (Mode) configurations with user-independent (Indep.) and user-dependent (Dep.) conditions.

Effect of Classifiers

Error rates of the random and heuristic classifiers were compared to that of LIR-Net (Tab. 5.2) for each configuration and user-dependency. In general, the error rates were statistically different from LIR-Net across all error types, classifier configurations and user-dependencies, except the steady-state error of the generic and mode-specific LDA classifiers. For overall errors, our proposed system achieved the lowest error rate (Dependent: [Generic: 1.1%, Mode-specific: 1.5%], Independent: [Generic: 7.7%, Mode-specific: 7.2%]) on both configurations and user-dependencies; whereas for the steady-state er-

ror, LIR-Net obtained the lowest error-rate on the generic configuration across all user-dependencies (Dependent: 0.5%, Independent: 5.4%) and the random classifier had the lowest error rate (0.0%) on the mode-specific configuration for both user-dependencies. In practice, the mode-specific random classifier simply predicted the current activity as the upcoming activity, which produced 0% error rates in steady-states. Since there were more steady-state than transitional cases in the dataset, the overall error rate of the mode-specific random classifier was lower than that of the generic classifier (Generic: 47.9 %, Mode-specific: 18.4%). The generic LIR-Net reached the lowest transitional error rates, while LDA had lowest transitional error within the mode-specific configurations across all user-dependencies.

Effect of User-Dependencies

We compared the error rates of the classifiers in the presence of different user-dependencies. The user-independent condition was statistically different from the user-dependent condition. For all classifiers, the error rates increased with the user-independent condition except the random classifier, the performance of which was governed by the distribution of the data (Tab. 5.1).

Classification Latency of LIR-Net

The averaged latency was 136.07 ± 3.86 ms.

Comparison to ResNet

The overall error rate of the ResNet was compared with that of LIR-Net. The overall error rate of the pre-trained ResNet ($1.29 \pm 0.20\%$) was not statistically different ($p = 0.13$) to that of LIR-Net ($1.11 \pm 0.26\%$), which validated the choice of our network design.

Table 5.3: LIR-Net performance on different sensor modalities and laterality groups

# of Sensor	Sensor Type	Ipsi (%)	Contra (%)	Bi (%)
1	I	4.68 [0.51]	5.99 [0.70]	2.58 [0.38]†
	G	3.85 [0.43]	4.00 [0.57]	**1.49 [0.29]**†
	E	7.66 [0.58]	8.00[0.51]	3.08 [0.46]†
2	I & G	3.05 [0.42]	3.33 [0.55]	1.29 [0.44]
	I & E	3.96 [0.55]	4.42 [0.52]	2.17 [0.71]†
	E & G	3.05 [0.55]	3.11 [0.32]	**1.15 [0.26]**
3	ALL	2.56 [0.50]*	2.89 [0.38]*	**1.11 [0.26]**

Overall error rates (mean, [standard deviation]) of LIR-Net using all possible combinations of laterality groups and sensor modalities: IMU (I), EMG (E), GONIO (G). The lowest error rates on each number of modalities are bolded.

* Asterisks under ipsilateral (Ipsi) and contralateral (Contra) classifiers denote the statistically significant differences between the according laterality and the bilateral (Bi) sensor set when all modalities are used.

† Daggers under sensor modalities denote the statistically significant differences between the according modality and all combined sensor (IMU & EMG & GONIO) with the bilateral sensors.

5.5.2 Effect of Sensor Locations and Modalities on LIR-Net

The overall error rate of LIR-Net was statistically compared across all combinations of sensor laterality groups and modalities. The interaction between the modalities and laterality groups was significant. The effect of laterality groups was observed by comparing the classifier's performance with all sensor modalities combined (Tab. 5.3). The error rate of the bilateral sensor set was statistically less (1.11%) than either ipsilateral or contralateral set. Similarly, the effect of sensor modalities was tested by comparing the performances of the classifier using the bilateral sensor set. The statistical significance was measured between all combined sensor modalities and individuals or combinations of two different sensor modalities. As a result, the error rate of the all combined sensor sets was significantly less than all individual sensor modalities and a pair of IMU and EMG sensors. For the single

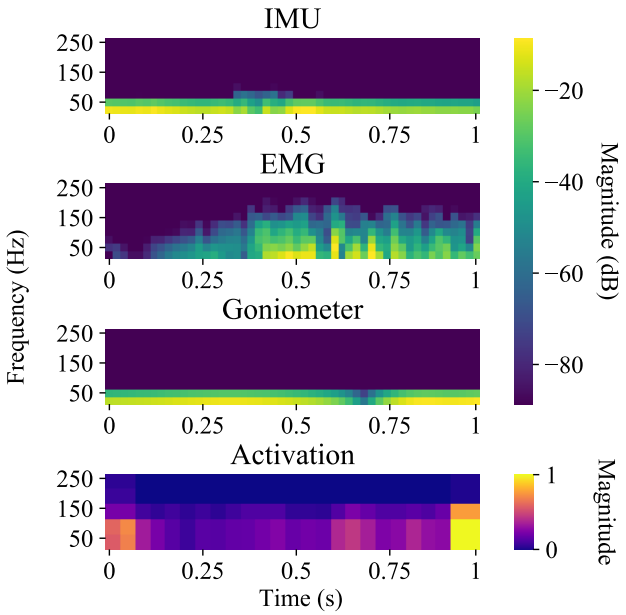

Figure 5.5: A representative sample of the right shank IMU, right VL EMG, and right knee GONIO signals (top three), and activation visualization of LIR-Net (bottom).

modalities with the bilateral sensor set, GONIO achieved the best performance, and for the two modalities, EMG and GONIO combination attained the lowest error rate.

5.5.3 Visualizing Activations of LIR-Net

The network showed greater activations in the lower frequencies, where much of the information in the signals was localized (Fig. 5.5). Additionally, the greatest activations in the low frequency area occurred near to the gait events.

5.6 Discussion

In this paper, we proposed a CNN-based intent recognition system that utilized the spectrogram to represent the frequency content of the input data. To this end, we studied the effect of sensor modalities and laterality groups on the proposed system, visualized the

activation of CNNs, and compared our system to the state-of-the-art (SOTA) intent recognition classifiers [44, 110]. The overall error rate of our proposed system was 1.11% which exceeded the performance of the existing work in a generic classification scheme. The motivation of this paper is to improve the prediction capabilities of lower-limb locomotor intent recognition systems; ideally, providing a framework for autonomous wearable robots which can assist wearers with a diverse range of activities encountered in the real-world.

5.6.1 Limitations

The error rates of the user-independent classifiers were statistically higher than that of the user-dependent classifier. This result showed the classifiers, including our proposed system, were not able to generalize well to novel subjects. The performance reduction of our system was due to an increase in transitional error (LDA: +5%, SVM: +12%, LIR-Net: +14%) across user-dependencies, compared to the increase in steady-state error (LDA: +5%, SVM: +12%, LIR-Net: +5%). This relatively weak generalizability of LIR-Net to novel users in transitional states is likely due to the unbalanced number of activity samples in the ENABL3S dataset; specifically, the number of transitions were less than that of steady states (Tab. 5.1). This stems from the fact that the data collection was conducted in a circuit that consisted of each activity, which is a convenient protocol, but may lead to sparseness in transition data [44]. Although our system obtained the lowest transitional error rates among all classifiers in the generic condition, the network had been trained and biased to lowering the overall error. In addition, deep learning (DL) generally performs better with more data, which may lead to greater improvements in performance when data are added, when compared to other classical machine learning algorithms. Thus, collecting more subject data with balanced number of samples, and techniques, such as data augmentation, can mitigate this limitation and improve the DL-based classifier [103].

5.6.2 Comparison to Past Works

Our proposed system was compared to the SOTA intent recognition systems using the ENABL3S dataset [44, 110]. For the combined mode-specific and user-dependent condition, our system performed comparably (1.52%) to the previous work of (1.43% [44]); but most importantly, in the generic classifier configuration, our work outperformed (1.11%) the heuristic-feature based classifiers (LDA: 6.43%, SVM: 4.30%) and CNNs with heuristic features as an input (3.7% [110]). Although mode-specific configurations could improve system performance, in real-world scenarios, relying on accurate knowledge of the previous step's activities (*i.e.* ground-truth) may be untenable. The performance of our intent recognition system in generic configuration demonstrates that our approach can be generalized across different environment conditions with various sequences of activities.

5.6.3 Effect of Sensor Locations and Modalities on LIR-Net

In general, as we fused more senor modalities and lateralites, the performance of the classifiers improved. This result was in accordance with the prior work using ENABL3S [44, 110]. Our findings showed GONIO had the best single modality performance with the bilateral sensor set; whereas IMU sensor data had the lowest error rate in previous works [44, 110]. For two modality sensors, EMG and GONIO combinations gave the best performance agreeing with the prior work [110]. Interestingly, the IMU and GONIO, EMG and GONIO combinations were not statistically different from all combined sensors, which suggests near optimal performance may be obtained from limited sensor selections.

5.6.4 Visualizing Activations of LIR-Net

Our technique of visualizing activations allowed a simple, but intuitive understanding of which features were learned by the network (Fig 5.5). The network had high activations nearby the gait events under the lower frequency region (<100 Hz). This shows that the information that dictates activity transitions are concentrated on the signals close to toe-off or heel-contact. To our knowledge, this is the first time that the activations of CNNs were qualitatively analyzed within a gait cycle, which is critical for identifying the intent using lower-limb neuromechanical signals. Although there was previous work visualizing the features of a CNN in lower limb sensor signals, the visualization was less intuitive and features were indistinguishable [109].

5.6.5 Application to Control of Wearable Robotics

Intent recognition is a control strategy which enables a wearable lower-limb robot to autonomously switch between controllers responsible for a specific task by inferring the wearer's locomotor intent. Typically, intent recognition is used as a high-level controller in a hierarchical control structure, where a mid-level controller encodes the activity-specific instructions for how to provide mechanical effort (*e.g.* via impedance or position control), and a low-level controller tracks the desired reference trajectories (*e.g.* feedback controller) [19]. A representative use case of the hierarchical controller is intent recognition in conjunction with mid-level finite-state controllers, where the gait cycle is divided by distinctive phases, and the transitions between these phases are based on heuristic rules. Since it is assumed that signals are stationary within each phase (*i.e.* identical activity), recognition-based classification strategies mitigate the time-varying characteristics of signals during the gait cycle [42].

The latency of our system was below what users may perceive (300 ms [120]) and

within the time window required to ensure smooth transitions between activities following the gait events [121]. The latency can be further reduced by exploiting optimal sensor selection, and microcomputers with higher processing capabilities. Thus, this work demonstrates the usability of these techniques in real-time control of wearable robots.

Chapter 6

Conclusion

6.1 Concluding Remark

In this disseration, I discuss key elements of wearable robots that will potentially enable these technology to be applicable in the real-world. I first describe the actuation element, which is a driving force of robotic applications. To this end, in Chapter 2, I presented a detailed analysis on high torque density brushless DC motor, where having a lightweight actuators is a major factor reducing the overall weight of the robotic system. In Chapter 3, I discuss the common limitations and misunderstanding when we are interpreting these high performance motors, despite the popularity of these motors in use of robotic applications. We also provide a detailed instructions of how to correctly model and understand the specifications of the motors. In Chapter 4, I discuss the second element, which is personalization or customization of controller based on users' preference. The assistance is provided from the actuator, and it is important for shaping the actuation profile to meet individual's needs. In this work, I presented a system which can automatically optimize users preferred controller setting in real-time. In Chapter 5, I touch on the last element, which is controlling these wearable devices. Despite the advancement in hardware technology, how to provide seamless and environment-aware control is still a challenge. I

provide a framework which uses machine learning to predict users' intent for classifying activities ahead of time. This information will be passed to an activity-specific controller according to the activity that the user is currently in. I hope my contributions on these key elements can push wearable systems outside of the research laboratories, and ultimately translating these robots for use in daily life.

6.2 Discussion of Contributions

The following describes the contribution of each chapter in the dissertation.

6.2.1 Chapter 2

This work (published in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems [18]) provided accurate and detailed specifications of a common ER-BLDC motor, the parameters of which are often unreported or not generalizable beyond drone applications. We characterized the motor's transfer function, efficiency across torque-speed regimes, as well as thermal properties. Accurate and detailed motor specifications are increasingly important, especially as ER-BLDC motors developed for the drone industry are more commonly used in general robotic applications.

6.2.2 Chapter 3

In this work (which is currently under review), we present the underlying mathematical modeling of brushed and brushless DC motors, as well as electrical commutation and winding configurations of BLDC motors. Most importantly, we described common sources of error in BLDC motor modeling, which often stem from inconsistencies and misinterpretation of manufacturer datasheets. To address these errors, we provide explana-

tions and conversions to the direct-quadrature reference frame, which facilitates a convenient DC representation while conserving key motor properties (*e.g.* resistive power loss). This work will contribute to the robotics community by 1. guiding engineers and robot designers who utilize BLDC motors in their robotic application to more accurately model and select a motor that is optimal for their application; and 2. highlighting the need for greater standardization and details to be provided in manufacturer datasheets (*e.g.* unit clarity, winding configuration, *etc.*).

6.2.3 Chapter 4

Our novelty is in two fold (this work is currently in prepartion for a publication): 1. Algorithmic contribution. As far as we know, the work presents a new algorithmic method of learning and optimizing user preference. Specifically, we employ an active learning strategy using a combination of neural networks and evolutionary algorithms, which differs from previous works in the field of preference learning. In our investigation, we validated the proposed method reduces the amount of queries required for identifying the optimal preferred setting. 2. Contribution to an application. We developed a framework which can be extended for daily use for identifying users preference. In addition, we used a lightweight, high torque output lower-limb exoskeleton system, which demonstrates physiological benefits (*e.g.* metabolic cost) for able-bodied people. Preference-based learning on these systems are not yet explored to the best of our knowledge.

6.2.4 Chapter 5

In this work (published in 2020 IEEE Robotics and Automation Letters [122]), we developed an intent recognition framework implementing convolutional neural networks with image encoding (*i.e.* spectrogram) that enables prediction of the upcoming locomotor ac-

tivity of the wearer's next walking step. In this aim, we describe our intent recognition system, comprised of a mel-spectrogram and subsequent neural network architecture. In addition, we analyzed the effect of sensor locations and modalities on the recognition system, and compared our proposed system to state-of-the-art locomotor intent recognition strategies.

Appendix

6.3 BLDC motor modelling cheat sheet

This appendix provides a condensed set of equations that are intended to be more easily referenced when analyzing BLDC motors in the design process. As a reminder, these equations are intended for brushless motors with a sinusoidal back-EMF profile. The cheat sheet includes governing equations of brushed and BLDC motors, and conversions for representing BLDC motors as a single phase “brushed" motor analogue, which includes a single current, voltage resistance, inductance, and torque / velocity constants, which are more convenient to analyze during modeling and design specification. We advocate for the DC representation (*i.e.* q-axis) of BLDC motors, which would provide convenient and accurate analysis while conserving critical properties, such as resistive power loss and torque production. For full derivation of the equations, please refer to the main sections of the paper, where the equations included in this cheat sheet are denoted with boxes. Equation numbers follow the numbering in the main sections.

6.3.1 Prerequisites

To convert to the q-axis DC representation of a selected BLDC motor, the designer should identify at least one frame of reference (q-axis, phase, line) for each electrical quantity. These quantities include R, K_t, K_v, which typically can be obtained from manufacturer datasheets; either (desired) current / voltage or torque / velocity in a specified reference

frame;as well as the motor's winding type (wye or delta). In addition, we provide a procedure for identifying the winding type of the motor in case they are not included in the manufacturers' specification (Appendix 6.4).

6.3.2 Governing motor equations

Brushed DC motor

$$J\frac{d^2\theta_m}{d^2t} = K_t I^a - b\frac{d\theta_m}{dt} - \tau_L \tag{3.38}$$

$$V = RI^a + K_b\frac{d\theta_m}{dt} + L\frac{dI^a}{dt} \tag{3.2}$$

Brushless DC motor

$$J\frac{d^2\theta_m}{d^2t} = K_t^q I^q - b\frac{d\theta_m}{dt} - \tau_L \tag{6.1}$$

$$V^q = R^\phi \cdot I^q + K_b^q\frac{d\theta_m}{dt} + L^e\frac{dI_q}{dt} \tag{3.39}$$

$$\begin{bmatrix} \bullet^d \\ \bullet^q \end{bmatrix} = \boldsymbol{PC} \begin{bmatrix} \bullet_A^\phi \\ \bullet_B^\phi \\ \bullet_C^\phi \end{bmatrix}$$
$$\boldsymbol{PC} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2}{3}\pi\right) & \cos\left(\theta + \frac{2}{3}\pi\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2}{3}\pi\right) & -\sin\left(\theta + \frac{2}{3}\pi\right) \end{bmatrix} \tag{3.31}$$

where $\bullet$ denotes an arbitrary electrical quantity (*e.g.* current or voltage).

6.3.3 Conversion to q-axis quantities

$$\text{Wye:} \qquad \bar{V}^{ll} = \sqrt{3}\bar{V}^{\phi} \tag{3.43}$$

$$\bar{I}^{l} = \bar{I}^{\phi} \tag{3.44}$$

$$\text{Delta:} \qquad \bar{V}^{ll} = \bar{V}^{\phi} \tag{3.45}$$

$$\bar{I}^{l} = \sqrt{3}\bar{I}^{\phi} \tag{3.46}$$

Resistance, resistive power loss, & inductance

$$\text{Wye:} \qquad R^{\phi} = \frac{1}{2}R^{ll} \tag{3.50}$$

$$\text{Delta:} \qquad R^{\phi} = \frac{3}{2}R^{ll} \tag{3.51}$$

$$P = I^{q2}R^{\phi} \tag{3.49}$$

$$\text{Wye:} \qquad L^{q} = \frac{3}{2}L^{ll} \tag{3.40}$$

$$\text{Delta:} \qquad L^{q} = \frac{1}{2}L^{ll} \tag{3.41}$$

Current & torque

$$\text{Wye:} \qquad \sqrt{\frac{3}{2}}\bar{I}^{l} = \sqrt{\frac{3}{2}}\bar{I^{\phi}} = I^{q} \tag{3.57}$$

$$\text{Delta:} \qquad \sqrt{\frac{1}{2}}\bar{I}^{l} = \sqrt{\frac{3}{2}}\bar{I^{\phi}} = I^{q} \tag{3.58}$$

Table 7.1: Conversion to phase / q-axis from line / terminal quantities

Parameter	Equivalencies	
	Wye	Delta
$\bar{V}^{\phi}$	$\frac{1}{\sqrt{3}}\bar{V}^{ll}$	$\bar{V}^{ll}$
$\bar{I}^{\phi}$	$\bar{I}^{l}$	$\frac{1}{\sqrt{3}}\bar{I}^{l}$
R^{ϕ}	$\frac{1}{2}R^{ll}$	$\frac{3}{2}R^{ll}$
L^{q}	$\frac{3}{2}L^{ll}$	$\frac{1}{2}L^{ll}$
V^{q}	$\frac{1}{\sqrt{2}}\bar{V}^{ll}$	$\sqrt{\frac{3}{2}}\bar{V}^{ll}$
I^{q}	$\sqrt{\frac{3}{2}}\bar{I}^{l}$	$\sqrt{\frac{1}{2}}\bar{I}^{l}$
K_t^{q}	$\frac{1}{\sqrt{2}}\bar{K_b^{ll}}$	$\sqrt{\frac{3}{2}}\bar{K_b^{ll}}$
τ	$K_t^{q}I^{q}$ or $\frac{\sqrt{3}}{2}\bar{K_b^{ll}}\bar{I}^{l}$	
P	$I^{q2}R^{\phi}$	

Tabulated description of Section 6.3.3. Note, if converting from a manufacturer's reported K_v number $(1/K_b^{ll})$, see Section 3.4.1.

$$\text{Wye:} \qquad K_t^{q} = \sqrt{\frac{3}{2}}\bar{K_t^{\phi}} \tag{3.55}$$

$$\text{Delta:} \qquad K_t^{q} = \sqrt{\frac{3}{2}}\bar{K_t^{\phi}} \tag{3.56}$$

$$\text{Wye:} \qquad K_t^{q} = \frac{1}{\sqrt{2}}\bar{K_b^{ll}} \tag{3.52}$$

$$\text{Delta:} \qquad K_t^{q} = \sqrt{\frac{3}{2}}\bar{K_b^{ll}} \tag{3.53}$$

$$\tau = K_t^q I^q \tag{3.59}$$

$$\tau = \frac{\sqrt{3}}{2} K_b^{\bar{ll}} \bar{I}^l \tag{3.54}$$

Voltage & angular velocity

$$\text{Wye:} \qquad \bar{V}^{ll} = \sqrt{2} V^q = \sqrt{2} K_b^q \frac{d\theta_m}{dt} \tag{3.60}$$

$$\text{Delta:} \qquad \bar{V}^{ll} = \sqrt{\frac{2}{3}} V^q = \sqrt{\frac{2}{3}} K_b^q \frac{d\theta_m}{dt} \tag{3.61}$$

$$\text{Wye:} \qquad \frac{d\theta_m}{dt}_{max} = \sqrt{\frac{1}{2}} \frac{V^{bus}}{K_b^q} \tag{3.60}$$

$$\text{Delta:} \qquad \frac{d\theta_m}{dt}_{max} = \sqrt{\frac{3}{2}} \frac{V^{bus}}{K_b^q} \tag{3.61}$$

6.4 Winding Type identification procedure

In this tutorial, one of the key attributes needed is the BLDC motor's winding type. Often, this information is not included in motor specifications or datasheets. In this case, we provide a procedure for identifying the winding type as described below.

Contact the manufacturer

The simplest method of identifying the winding type of a BLDC motor is to contact the motor manufacturer. Importantly, wye wound motor types may be called “star" or “star-serial".

Visual Inspection

Another convenient method for identifying the motor's winding type when the windings are visible is to visually inspect them. For wye wound motors, the leads of the three phases are connected to form a neutral point (see Fig. 3.4) with the other ends of the phases being the motor's leads. Sometimes, this neutral point can be observed (*i.e.* under shrink wrap). Alternatively, delta wound motors have no neutral points and the combination of each of the phases are wired together to form the ends (see Fig. 3.4).

Thermal Imaging

Thermal imaging techniques provide a reasonably convenient and economical approach for measuring the surface temperature of objects. The following describes the procedure for identifying the winding configuration of a BLDC motor:

1. Connect two of the leads of the BLDC motor to a DC power supply. We commonly use power supply currents from 4A to 8A.

2. Direct a thermal imaging camera to the motor.

3. Allow current to flow through the leads until some windings become visible in the thermal camera, while not overheating the motor excessively.

In the example image (Fig. 7.1), only one set of windings is producing the majority of the heat while two sets are producing less (4x less resistive loss), indicating that the motor is delta wound. Wye wound motors would have two sets of windings producing more heat with one set of windings unpowered.

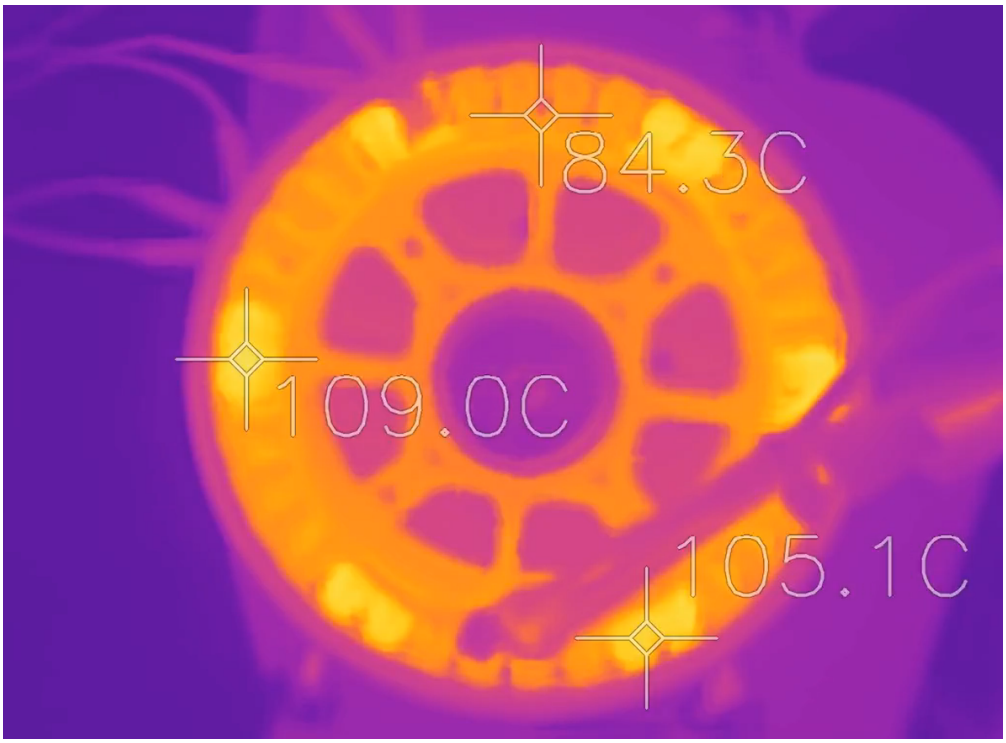

Figure 7.1: A delta wound motor with current flowing through two leads with the majority of the heat produced through one phase. This method can be used to identify winding type by visual inspection.

6.5 Motor Physics

Here, we briefly review the fundamentals of brushed DC motor operation. Permanent magnet electric motors create force via interaction between the magnetic field produced from permanent magnets and an electromagnetic field produced by the windings. This force is governed by the Lorentz Force Law. If we consider a one squared-loop of wire in two magnets, the Lorentz Law can be expressed as:

$$\boldsymbol{F} = I(\boldsymbol{\ell} \times \boldsymbol{B}) = I\|\boldsymbol{B}\|\|\boldsymbol{\ell}\|\boldsymbol{n} \;\; (\boldsymbol{\ell} \perp \boldsymbol{B}) \tag{6.2}$$

where $\boldsymbol{F}$ is the force created on one side of the wire, $\boldsymbol{B}$ is the magnetic field created by the magnet, I is current flowing through the wire, and $\boldsymbol{\ell}$ is the length vector of the wire

parallel to the rotational axis, and $\boldsymbol{n}$ is a unit vector perpendicular to both wire and the magnetic flux. Then, the torque acting on a whole loop becomes:

$$\boldsymbol{\tau} = 2\boldsymbol{F} \times \boldsymbol{D} = 2\|\boldsymbol{D}\|\|\boldsymbol{F}\| \sin(\theta_m)\, \boldsymbol{j} \tag{6.3}$$

$$= 2\, \|\boldsymbol{D}\|\|\boldsymbol{\ell}\|\|\boldsymbol{B}\| I \sin(\theta_m)\, \boldsymbol{j} \tag{6.4}$$

where D is a half of the width of the loop (Fig. 28-18 of [123]), and θ is the angle of the loop, and $\boldsymbol{j}$ is a unit vector perpendicular to $\boldsymbol{F}$ and $\boldsymbol{D}$. Please refer to Figure 28-18 of [123] for depiction of the motor.

The relative movement of the coil caused by the Lorentz law in the magnetic field induces a voltage (*i.e.* the back-EMF). The back-EMF is generated by the Faraday's Law:

$$\begin{aligned} V_e &= \oint (\boldsymbol{v} \times \boldsymbol{B}) \cdot \boldsymbol{d}\ell = 2\|\boldsymbol{v}\|\|\boldsymbol{B}\|\|\boldsymbol{\ell}\| \\ &= 2\|r\|\frac{d\theta_m}{dt}\|\boldsymbol{B}\|\|\boldsymbol{\ell}\| \quad (\boldsymbol{v} \times \boldsymbol{B} \parallel \boldsymbol{\ell}) \end{aligned}$$

where V_e, is the induced electromotive force voltage, and v is the tangential velocity of the wire.

Torque & Back-EMF production

The torque acting on the rotor composed of N number of coils is calculated by the Lorentz Law:

$$\boldsymbol{\tau} = 2(IBN\ell)D \sin(\theta_m)\, \boldsymbol{j} \tag{6.5}$$

For the rotor rotate in a single direction ($\theta > 0$), the current must reverse at $\pm 90 \times n$ ($n = 1, 2, 3...$). In a brushed motor, this is enabled by the brushes and physical commutator.

Furthermore, by adding the loops of coils and offsetting from each other, a motor can be developed which creates a nearly constant torque (*i.e.* no torque ripple) [124]. By reflecting these changes, the torque exerted on the rotor can be expressed as:

$$\boldsymbol{\tau} = 2(IBN\ell)D\boldsymbol{j} = K_t I\boldsymbol{j} \tag{6.6}$$

where the torque constant is defined as:

$$K_t = 2BN\ell D \tag{6.7}$$

Similarly, the back-EMF of a brushed motor can be calculated:

$$V_e = 2(D\frac{d\theta_m}{dt})BN\ell = K_b\frac{d\theta_m}{dt} \tag{6.8}$$

where K_b is the back-EMF constant defined as:

$$K_b = 2BN\ell D \tag{6.9}$$

These two equations (6.6), (6.8) demonstrate that the torque constant and back-EMF constants are identical:

$$K_b = K_t \tag{6.10}$$

Note that this relationship is only valid when using SI units. For different units, adequate conversion should be included in the modeling analysis. The above property stems from

the power conversion:

$$\tau \frac{d\theta_m}{dt} = IV_e \tag{6.11}$$

$$(K_t I)\frac{d\theta_m}{dt} = I(K_b \frac{d\theta_m}{dt}) \tag{6.12}$$

$$K_t \cancel{I} \cancel{\frac{d\theta_m}{dt}} = K_b \cancel{I} \cancel{\frac{d\theta_m}{dt}} \tag{6.13}$$

where it represents the conversion from mechanical to electrical power in an ideal motor.

6.6 Conversion from brushless to single phase "brushed" motor quantities (*i.e.* q-axis)

The following section describes a full derivation of the voltage equation that governs BLDC motors. We start from voltage equations of each phase and convert those to the d-q axes. We first introduce the concept of total flux-linkage, which is the total magnetic flux through the windings. The name 'flux-linkage' stems from the fact that the windings are linked by the shared magnetic flux [123]. The total flux-linkage is composed of self and mutual flux-linkages between the stator windings, and the flux-linkages between the permanent magnet and windings (*i.e.* rotor-stator flux-linkage):

$$\boldsymbol{\Psi}^{\phi} = \boldsymbol{L}^{\phi}\boldsymbol{I}^{\phi} + \boldsymbol{\Psi}^{\boldsymbol{R}} \tag{6.14}$$

where, $\boldsymbol{L}^{\phi}$ and $\boldsymbol{I}^{\phi}$ denotes inductances and currents of all phases, and flux-linkages of windings are denoted $\boldsymbol{L}^{\phi}\boldsymbol{I}^{\phi}$ and flux-linkages between the magnet and windings are

denoted $\boldsymbol{\Psi^R}$. Each element represents phase quantities of the flux-linkages:

$$\boldsymbol{\Psi}^{\phi} = \begin{bmatrix} \Psi_A^{\phi} \\ \Psi_B^{\phi} \\ \Psi_C^{\phi} \end{bmatrix}, \boldsymbol{\Psi}^{R} = \begin{bmatrix} \Psi_A^{R} \\ \Psi_B^{R} \\ \Psi_C^{R} \end{bmatrix} \tag{6.15}$$

$$\boldsymbol{I}^{\phi} = \begin{bmatrix} I_A^{\phi} \\ I_B^{\phi} \\ I_C^{\phi} \end{bmatrix} \tag{6.16}$$

$$\boldsymbol{L}^{\phi} = \begin{bmatrix} L_s & L_m & L_m \\ L_m & L_s & L_m \\ L_m & L_m & L_s \end{bmatrix} \tag{6.17}$$

In balanced BLDC motors, the self inductances of each phase and mutual inductances of each pair of phases are identical [125]. To derive the voltage equation of the motor, Kirchoff's Voltage Law can be expressed in matrix form as:

$$\boldsymbol{V^{\phi}} = \boldsymbol{R^{\phi}} \boldsymbol{I^{\phi}} + \frac{d\boldsymbol{\Psi}^{\phi}}{dt} \tag{6.18}$$

where the equation of one representative phase is:

$$V_A^{\phi} = R^{\phi} I_A^{\phi} + \frac{d}{dt}(L_s I_A^{\phi} + L_m I_B^{\phi} + L_m I_C^{\phi}) + \frac{d\Psi_A^R}{dt} \tag{6.19}$$

Since we assumed phase currents are sinusoidal and 120° out of phase, the phase currents have following property:

$$I_A^{\phi} + I_B^{\phi} + I_C^{\phi} = 0 \tag{6.20}$$

The above property simplifies the voltage equation (6.19) of a single phase to:

$$V_A^\phi = R^\phi I_A^\phi + (L_s - L_m)\frac{dI_A^\phi}{dt} + \frac{d\Psi_A^R}{dt} \tag{6.21}$$

$$= R^\phi I_A^\phi + L^e \frac{dI_A^\phi}{dt} + \frac{d\Psi_A^R}{dt} \tag{6.22}$$

where, we define an effective inductance as:

$$L^e = L_s - L_m \tag{6.23}$$

and back-EMF of a phase as:

$$V_{e,A} = \frac{d\Psi_A^R}{dt} = K_{b,A}^\phi \frac{d\theta_m}{dt} \tag{6.24}$$

Therefore, applying the same logic on all phases results in following expression:

$$\begin{bmatrix} V_A \\ V_B \\ V_C \end{bmatrix} = \begin{bmatrix} R^\phi & 0 & 0 \\ 0 & R^\phi & 0 \\ 0 & 0 & R^\phi \end{bmatrix} \cdot \begin{bmatrix} I_A \\ I_B \\ I_C \end{bmatrix} + \begin{bmatrix} L^e & 0 & 0 \\ 0 & L^e & 0 \\ 0 & 0 & L^e \end{bmatrix} \cdot \frac{d}{dt} \begin{bmatrix} I_A \\ I_B \\ I_C \end{bmatrix} + \begin{bmatrix} K_{b,A}^\phi \\ K_{b,B}^\phi \\ K_{b,C}^\phi \end{bmatrix} \frac{d\theta_m}{dt} \tag{6.25}$$

where the winding voltage and back-EMF constant of each phase is:

$$V_A = \bar{V}^\phi \sin(\theta) \tag{6.26}$$

$$V_B = \bar{V}^\phi \sin\left(\theta - \frac{2}{3}\pi\right) \tag{6.27}$$

$$V_C = \bar{V}^\phi \sin\left(\theta + \frac{2}{3}\pi\right) \tag{6.28}$$

$$K_{b,A}^{\phi} = \bar{K}_b^{\phi} \sin(\theta) \tag{6.29}$$

$$K_{b,B}^{\phi} = \bar{K}_b^{\phi} \sin\left(\theta - \frac{2}{3}\pi\right) \tag{6.30}$$

$$K_{b,C}^{\phi} = \bar{K}_b^{\phi} \sin\left(\theta + \frac{2}{3}\pi\right) \tag{6.31}$$

where $V_x, K_{b,x}^{\phi}$ are phase voltages and phase back-EMF constants, respectively and $\bar{V}^{\phi}$, $\bar{K}_b^{\phi}$ are amplitudes of each phase, which amplitudes are all identical.

By applying power invariant transformation (3.31) to the time derivative of the total flux linkage [126]:

$$\boldsymbol{Q}\frac{d}{dt}\boldsymbol{\Psi}^{\boldsymbol{\phi}} = \boldsymbol{Q}\frac{d}{dt}(\boldsymbol{Q}^{\dagger}\boldsymbol{\Psi}^{\boldsymbol{dq}}) \tag{6.32}$$

$$= \boldsymbol{Q}\frac{d}{dt}\boldsymbol{Q}^{\dagger}\boldsymbol{\Psi}^{\boldsymbol{dq}} + \frac{d\boldsymbol{\Psi}^{\boldsymbol{\phi}}}{dt} \tag{6.33}$$

In here, we denote † as the right inverse ($\boldsymbol{Q} \cdot \boldsymbol{Q}^{\dagger} = \boldsymbol{I}$). and $\boldsymbol{Q} = \boldsymbol{PC}$ is the d-q transformation (3.29), (3.30). By converting all elements in the voltage equation to the d-q axis (6.18):

$$\boldsymbol{V}^{\boldsymbol{dq}} = \begin{bmatrix} V_d \\ V_q \end{bmatrix}, \boldsymbol{\Psi}^{\boldsymbol{dq}} = \begin{bmatrix} \Psi_d \\ \Psi_q \end{bmatrix} \tag{6.34}$$

$$\boldsymbol{Q}\frac{d}{dt}\boldsymbol{Q}^{\dagger} = \begin{bmatrix} 0 & -\frac{d\theta}{dt} \\ \frac{d\theta}{dt} & 0 \end{bmatrix} \tag{6.35}$$

Therefore by substituting above d-q quantities in the BLDC electrical equation, the d-q

representation of the voltage equation becomes:

$$V^d = R^\phi I^d + \frac{d\Psi^d}{dt} - \frac{d\theta}{dt}\Psi^q \tag{6.36}$$

$$V^q = R^\phi I^q + \frac{d\Psi^q}{dt} + \frac{d\theta}{dt}\Psi^d \tag{6.37}$$

By further expanding the above equation using the flux-linkages:

$$V^d = R^\phi I^d + L_d\frac{dI^d}{dt} + \frac{d\bar{\Psi^R}}{dt} - \frac{d\theta}{dt}L^q I^q \tag{6.38}$$

where,

$$\Psi_q = L^q I^q \tag{6.39}$$

$$V^q = R^\phi I^q + L_q\frac{dI^q}{dt} + \frac{d\theta}{dt}(L^d I^d + \bar{\Psi^R}) \tag{6.40}$$

where,

$$\Psi_d = L^d I^d + \bar{\Psi^R} \tag{6.41}$$

which demonstrates the magnetic flux-linkage is only in the d-axis. Since the amplitude of the magnetic flux-linkage $(\bar{\Psi^R})$ is constant and the d-q transformation is in phase with the phase currents $(I^d = 0)$ (refer to the full list of assumptions in 3.3.2) the d-q axis voltage equation reduces to:

$$V^d = R^\phi I^d - \frac{d\theta}{dt}L^e I^q \tag{6.42}$$

$$V^q = R^\phi I^q + L^e\frac{dI_q}{dt} + \frac{d\theta_m}{dt}K_b^q \tag{3.39}$$

where $\bar{\Psi^R} = pK_b^q$, and $L^d = L^q = L^e$ from the d-q transformation as follows [126]:

$$\boldsymbol{L^{dq}} = \boldsymbol{Q}\boldsymbol{L}^{\phi}\boldsymbol{Q}^{\dagger} = \begin{bmatrix} L^d & 0 \\ 0 & L^q \end{bmatrix} = \begin{bmatrix} L^e & 0 \\ 0 & L^e \end{bmatrix} \tag{6.43}$$

The last equality is due to the non-saliency of the rotor. This demonstrates the q-axis voltage equation reduces to a DC representation, and agrees with the characteristics of the surface permanent synchronous motor where reluctance is almost equal in d and q-axis direction (*i.e.* saliency ratio of 1) [127, 128].

6.7 Accurate Power Loss of BLDC Motors

In the paper, we provided accurate calculation of electrical power consumption of BLDC motors' power loss of the motors in phase and q-axis quantities, which are agnostic to the winding configuration (Section 3.3.6). However, typically electrical resistance is provided as terminal resistance, where conversion from terminal to phase resistance requires a knowledge of the motor's winding configuration. In this appendix, we demonstrate how using terminal resistance inappropriately without the adequate use of conversion factor can lead to imprecise power loss of the motors.

The following describe the power loss with respect to line-to-line (terminal) quantities on each configuration of the motors. Using (3.49), for wye-wound motors:

$$R^{\phi} = \frac{1}{2}R^{ll} \tag{3.50}$$

$$P = I^{q2}R^{\phi} = \frac{1}{2}I^{q2}R^{ll} \tag{6.44}$$

For delta-wound motors:

$$R^{\phi} = \frac{3}{2} R^{ll} \tag{3.51}$$

$$P = I^{q2} R^{\phi} = \frac{3}{2} I^{q2} R^{ll} \tag{6.45}$$

where ll denotes line-to-line quantities. The power loss (eq. 6.44, 6.45) shows that if one considers the line-to-line quantities as brushed equivalent quantities (*i.e.* $P = I^{q2} R^{ll}$), the power loss would be two times higher for wye-wound and 2/3 times lower for delta-wound than the actual power consumption. The susceptibility of making an error is compounded by the fact that typically manufacturers' datasheets report resistances as terminal resistances. This demonstrates that inaccurate usage of electrical quantities can lead to over or under-prediction of the power loss.

6.8 Accurate Torque Production of BLDC Motors

We describe one of the errors during the process of estimating torque of BLDC motors that users are prone to make. Inaccuracy in applying the formula to a BLDC motor stems from pairing a torque constant K_t with a different current value than the constant is defined in reference to. As described in Sec. 3.4.1, a common error is to consider the back-EMF constant K_b as torque constant K_t which is only true in certain reference frames (*e.g.* phase) for BLDC motors. Specifically, manufacturers of exterior-rotor-type BLDC motors, also known as drone motors, typically report only the velocity constant K_v which is the reciprocal of the back-EMF constant K_b, where the velocity constant is represented in line-to-line frame (3.45):

$$\bar{K_v^{ll}} = 1/\bar{K_b^{ll}} \tag{6.46}$$

We assume a delta configuration in this example, where the line-to-line voltage is identical to phase voltage:

$$\bar{K_b^{ll}} = \bar{K_b^{\phi}} \tag{6.47}$$

Since phase K_b is equal to phase K_t, by using the conversion of K_t described in the main section (3.56), the q-axis torque constant becomes:

$$\bar{K_b^{\phi}} = \bar{K_t^{\phi}} = \sqrt{\frac{2}{3}} K_t^q \tag{6.48}$$

Summing up the equations described above, the accurate form of torque estimation using q-axis current becomes:

$$\tau = K_t^q I^q = \sqrt{\frac{3}{2}} \bar{K_b^{\phi}} I^q = \sqrt{\frac{3}{2}} \bar{K_b^{ll}} I^q = \sqrt{\frac{3}{2}} \frac{1}{\bar{K_v^{ll}}} I^q \tag{6.49}$$

Therefore, incorrectly using the K_v as q-axis representation, this may lead to $\sqrt{\frac{3}{2}}$ smaller torque estimation than actual. Note that there can be variants of inaccurate estimations depending on which pair of torque and current frame is selected, and the demonstrated example is merely one of these potential sources of errors.

Bibliography

[1] Gregory S Sawicki, Owen N Beck, Inseung Kang, and Aaron J Young. The exoskeleton expansion: improving walking and running economy. *Journal of neuroengineering and rehabilitation*, 17(1):1–9, 2020.

[2] Ryan J Farris, Hugo A Quintero, and Michael Goldfarb. Preliminary evaluation of a powered lower limb orthosis to aid walking in paraplegic individuals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(6):652–659, 2011.

[3] Gabi Zeilig, Harold Weingarden, Manuel Zwecker, Israel Dudkiewicz, Ayala Bloch, and Alberto Esquenazi. Safety and tolerance of the rewalkâ¢ exoskeleton suit for ambulation by people with complete spinal cord injury: A pilot study. *The journal of spinal cord medicine*, 35(2):96–101, 2012.

[4] Jeremi Gancet, Michel Ilzkovitz, Elvina Motard, Yashodhan Nevatia, Pierre Letier, David De Weerdt, Guy Cheron, Thomas Hoellinger, Karthik Seetharaman, Mathieu Petieau, et al. Mindwalker: Going one step further with assistive lower limbs exoskeleton for sci condition subjects. In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 1794–1800. IEEE, 2012.

[5] Younbaek Lee, Jongwon Lee, Byungjune Choi, Minhyung Lee, Se-gon Roh, Kyungrock Kim, Keehong Seo, Yong-Jae Kim, and Youngbo Shim. Flexible gait enhancing

mechatronics system for lower limb assistance (gems l-type). *IEEE/ASME Transactions on Mechatronics*, 24(4):1520–1531, 2019.

[6] Aaron J. Young and Daniel P. Ferris. State of the art and future directions for lower limb robotic exoskeletons. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(2):171–182, 2017.

[7] Michael Bernhardt, Martin Frey, Gery Colombo, and Robert Riener. Hybrid force-position control yields cooperative behaviour of the rehabilitation robot lokomat. In *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pages 536–539. IEEE, 2005.

[8] Shiqian Wang, Letian Wang, Cory Meijneke, Edwin Van Asseldonk, Thomas Hoellinger, Guy Cheron, Yuri Ivanenko, Valentina La Scaleia, Francesca Sylos-Labini, Marco Molinari, et al. Design and control of the mindwalker exoskeleton. *IEEE transactions on neural systems and rehabilitation engineering*, 23(2):277–286, 2014.

[9] Klaske Van Kammen, Annemarijke Boonstra, Heleen Reinders-Messelink, and Rob den Otter. The combined effects of body weight support and gait speed on gait related muscle activity: a comparison between walking in the lokomat exoskeleton and regular treadmill walking. *PloS one*, 9(9):e107323, 2014.

[10] Aaron J Young and Daniel P Ferris. State of the art and future directions for lower limb robotic exoskeletons. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(2):171–182, 2016.

[11] Hao Lee, Peter Walker Ferguson, and Jacob Rosen. Lower limb exoskeleton systemsâoverview. In *Wearable Robotics*, pages 207–229. Elsevier, 2020.

[12] Kazunori Koseki, Arito Yozu, Hanako Takano, Atsushi Abe, Kenichi Yoshikawa,

Takayuki Maezawa, Yutaka Kohno, and Hirotaka Mutsuzaki. Gait training using the honda walking assist device® for individuals with transfemoral amputation: a report of two cases. *Journal of back and musculoskeletal rehabilitation*, 33(2):339–344, 2020.

[13] S. Seok, A. Wang, M. Y. (Michael) Chuah, D. J. Hyun, J. Lee, D. M. Otten, J. H. Lang, and S. Kim. Design principles for energy-efficient legged locomotion and implementation on the mit cheetah robot. *IEEE/ASME Transactions on Mechatronics*, 20(3):1117–1129, June 2015.

[14] Gavin Kenneally and Daniel E Koditschek. Leg design for energy management in an electromechanical robot. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5712–5718. IEEE, 2015.

[15] Gavin Kenneally, Avik De, and Daniel E Koditschek. Design principles for a family of direct-drive legged robots. *IEEE Robotics and Automation Letters*, 1(2):900–907, 2016.

[16] Luke M Mooney, Elliott J Rouse, and Hugh M Herr. Autonomous exoskeleton reduces metabolic cost of human walking during load carriage. *Journal of neuroengineering and rehabilitation*, 11(80):1–11, 2014.

[17] Avik De, Abriana Stewart-Height, and Daniel E Koditschek. Task-based control and design of a bldc actuator for robotics. *IEEE Robotics and Automation Letters*, 4(3):2393–2400, 2019.

[18] 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.

[19] Michael R Tucker, Jeremy Olivier, Anna Pagel, Hannes Bleuler, Mohamed Bouri, Olivier Lambercy, José del R Millán, Robert Riener, Heike Vallery, and Roger Gassert. Control strategies for active lower extremity prosthetics and orthotics: a review. *Journal of neuroengineering and rehabilitation*, 12(1):1–29, 2015.

[20] Juanjuan Zhang, Pieter Fiers, Kirby A Witte, Rachel W Jackson, Katherine L Poggensee, Christopher G Atkeson, and Steven H Collins. Human-in-the-loop optimization of exoskeleton assistance during walking. *Science*, 356(6344):1280–1284, 2017.

[21] Xiangyu Peng, Yadrianna Acosta-Sojo, Man I Wu, and Leia Stirling. Perception of powered ankle exoskeleton actuation timing during walking: A pilot study. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 4654–4657. IEEE, 2021.

[22] Siavash Rezazadeh, David Quintero, Nikhil Divekar, Emma Reznick, Leslie Gray, and Robert D Gregg. A phase variable approach for improved rhythmic and non-rhythmic control of a powered knee-ankle prosthesis. *IEEE Access*, 7:109840–109855, 2019.

[23] Ross J Cortino, Edgar Bolívar-Nieto, T Kevin Best, and Robert D Gregg. Stair ascent phase-variable control of a powered knee-ankle prosthesis. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5673–5678. IEEE, 2022.

[24] Yue Wen, Jennie Si, Andrea Brandt, Xiang Gao, and He Helen Huang. Online reinforcement learning control for the personalization of a robotic knee prosthesis. *IEEE transactions on cybernetics*, 50(6):2346–2356, 2019.

[25] Alejandro F Azocar, Luke M Mooney, Levi J Hargrove, and Elliott J Rouse. Design and characterization of an open-source robotic leg prosthesis. In *2018 7th IEEE Inter-*

national Conference on Biomedical Robotics and Biomechatronics (*Biorob*), pages 111–118. IEEE, 2018.

[26] Alejandro F Azocar, Luke M Mooney, Jean-François Duval, Ann M Simon, Levi J Hargrove, and Elliott J Rouse. Design and clinical implementation of an open-source bionic leg. *Nature biomedical engineering*, 4(10):941–953, 2020.

[27] Ruofan Wu, Minhan Li, Zhikai Yao, Wentao Liu, Jennie Si, and He Huang. Reinforcement learning impedance control of a robotic prosthesis to coordinate with human intact knee motion. *IEEE Robotics and Automation Letters*, 7(3):7014–7020, 2022.

[28] Abbie E Ferris, Jennifer M Aldridge, Christopher A Rábago, and Jason M Wilken. Evaluation of a powered ankle-foot prosthetic system during walking. *Archives of physical medicine and rehabilitation*, 93(11):1911–1918, 2012.

[29] Alena M Grabowski and Susan DâAndrea. Effects of a powered ankle-foot prosthesis on kinetic loading of the unaffected leg during level-ground walking. *Journal of neuroengineering and rehabilitation*, 10(1):1–12, 2013.

[30] Ann M Simon, Kimberly A Ingraham, Nicholas P Fey, Suzanne B Finucane, Robert D Lipschutz, Aaron J Young, and Levi J Hargrove. Configuring a powered knee and ankle prosthesis for transfemoral amputees within five specific ambulation modes. *PloS one*, 9(6):e99387, 2014.

[31] Jeffrey R Koller, Deanna H Gates, Daniel P Ferris, and C David Remy. 'body-in-the-loop'optimization of assistive robotic devices: A validation study. In *Robotics: Science and Systems*, volume 2016, pages 1–10, 2016.

[32] Ye Ding, Myunghee Kim, Scott Kuindersma, and Conor J Walsh. Human-in-the-loop optimization of hip assistance with a soft exosuit during walking. *Science robotics*, 3(15):eaar5438, 2018.

[33] Gwendolyn M Bryan, Patrick W Franks, Seungmoon Song, Alexandra S Voloshina, Ricardo Reyes, Meghan P OâDonovan, Karen N Gregorczyk, and Steven H Collins. Optimized hip–knee–ankle exoskeleton assistance at a range of walking speeds. *Journal of neuroengineering and rehabilitation*, 18(1):1–12, 2021.

[34] Max K Shepherd, Ann M Simon, Joey Zisk, and Levi J Hargrove. Patient-preferred prosthetic ankle-foot alignment for ramps and level-ground walking. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29:52–59, 2020.

[35] Tyler R Clites, Max K Shepherd, Kimberly A Ingraham, Leslie Wontorcik, and Elliott J Rouse. Understanding patient preference in prosthetic ankle stiffness. *Journal of neuroengineering and rehabilitation*, 18(1):1–16, 2021.

[36] KA Ingraham, CD Remy, and EJ Rouse. The role of user preference in the customized control of robotic exoskeletons. *Science robotics*, 7(64):eabj3487, 2022.

[37] Maegan Tucker, Myra Cheng, Ellen Novoseller, Richard Cheng, Yisong Yue, Joel W Burdick, and Aaron D Ames. Human preference-based learning for high-dimensional optimization of exoskeleton walking gaits. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3423–3430. IEEE, 2020.

[38] Maegan Tucker, Ellen Novoseller, Claudia Kann, Yanan Sui, Yisong Yue, Joel W Burdick, and Aaron D Ames. Preference-based learning for exoskeleton gait optimization. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2351–2357. IEEE, 2020.

[39] Kejun Li, Maegan Tucker, Erdem Bıyık, Ellen Novoseller, Joel W Burdick, Yanan Sui, Dorsa Sadigh, Yisong Yue, and Aaron D Ames. Roial: Region of interest active learning for characterizing exoskeleton gait preference landscapes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3212–3218. IEEE, 2021.

[40] Aaron J Young and Levi J Hargrove. A classification method for user-independent intent recognition for transfemoral amputees using powered lower limb prostheses. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(2):217–225, 2015.

[41] Huseyin Atakan Varol, Frank Sup, and Michael Goldfarb. Multiclass real-time intent recognition of a powered lower limb prosthesis. *IEEE Transactions on Biomedical Engineering*, 57(3):542–551, 2009.

[42] Aaron J Young, Ann M Simon, Nicholas P Fey, and Levi J Hargrove. Intent recognition in a powered lower limb prosthesis using time history information. *Annals of biomedical engineering*, 42(3):631–641, 2014.

[43] AJ Young, TA Kuiken, and LJ Hargrove. Analysis of using emg and mechanical sensors to enhance intent recognition in powered lower limb prostheses. *Journal of neural engineering*, 11(5):056021, 2014.

[44] Blair Hu, Elliott Rouse, and Levi Hargrove. Fusion of bilateral lower-limb neuromechanical signals improves prediction of locomotor activities. *Frontiers in Robotics and AI*, 5(78):1–16, 2018.

[45] Jonathon W Sensinger, Stephen D Clark, and Jack F Schorsch. Exterior vs. interior rotors in robotic brushless motors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2764–2770. IEEE, 2011.

[46] Maxon Motors. Escon 50/5 servo controller hardware reference, 2013.

[47] Jean-François Duval and Hugh M Herr. Flexsea: Flexible, scalable electronics architecture for wearable robotic applications. In *Biomedical Robotics and Biomechatronics (BioRob), 2016 6th IEEE International Conference on*, pages 1236–1241. IEEE, 2016.

[48] Simon Kalouche. Goat: A legged robot with 3d agility and virtual compliance. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 4110–4117. IEEE, 2017.

[49] Nathaniel Kingry, Logan Towers, Yen-Chen Liu, Yue Zu, Yuchen Wang, Briana Staheli, Yusuke Katagiri, Samuel Cook, and Ran Dai. Design, modeling and control of a solar-powered quadcopter. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1251–1258. IEEE, 2018.

[50] Mohamad Ali Tofigh, Mohamad J Mahjoob, and Moosa Ayati. Dynamic modeling and nonlinear tracking control of a novel modified quadrotor. *International Journal of Robust and Nonlinear Control*, 28(2):552–567, 2018.

[51] Daniel J Blackman, John V Nicholson, Jason L Pusey, Max P Austin, Charles Young, Jason M Brown, and Jonathan E Clark. Leg design for running and jumping dynamics. In *Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference on*, pages 2617–2623. IEEE, 2017.

[52] Robert H Park. Two-reaction theory of synchronous machines generalized method of analysis-part i. *Transactions of the American Institute of Electrical Engineers*, 48(3):716–727, 1929.

[53] Lennart Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998.

[54] Safa O Kasap. *Principles of electronic materials and devices*, volume 2. McGraw-Hill New York, 2006.

[55] Elliott J Rouse, Luke M Mooney, and Hugh M Herr. Clutchable series-elastic actuator: Implications for prosthetic knee design. *The International Journal of Robotics Research*, 33(13):1611–1625, 2014.

[56] Max K Shepherd and Elliott J Rouse. Design and validation of a torque-controllable knee exoskeleton for sit-to-stand assistance. *IEEE/ASME Transactions on Mechatronics*, 22(4):1695–1704, 2017.

[57] S. B. Ozturk, W. C. Alexander, and H. A. Toliyat. Direct torque control of four-switch brushless dc motor with non-sinusoidal back emf. *IEEE Transactions on Power Electronics*, 25(2):263–271, 2010.

[58] U. Neethu and V. R. Jisha. Speed control of brushless dc motor: A comparative study. In *2012 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pages 1–5, 2012.

[59] Bryce Vickmark. *MIT's new mini cheetah robot*. MIT News, Mar 2019.

[60] Joseph Xu. *OPEN-SOURCE BIONIC LEG: FIRST-OF-ITS-KIND PLATFORM AIMS TO RAPIDLY ADVANCE PROSTHETICS*. Michigan Engineering, June 2019.

[61] Sangok Seok, Albert Wang, Meng Yee Chuah, David Otten, Jeffrey Lang, and Sangbae Kim. Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot. In *2013 IEEE International Conference on Robotics and Automation*, pages 3307–3312. IEEE, 2013.

[62] Sangok Seok, Albert Wang, Meng Yee Michael Chuah, Dong Jin Hyun, Jongwoo Lee, David M Otten, Jeffrey H Lang, and Sangbae Kim. Design principles for energy-

efficient legged locomotion and implementation on the mit cheetah robot. *Ieee/asme transactions on mechatronics*, 20(3):1117–1129, 2014.

[63] Yanran Ding and Hae-Won Park. Design and experimental implementation of a quasi-direct-drive leg for optimized jumping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 300–305. IEEE, 2017.

[64] Siavash Rezazadeh, Andy Abate, Ross L Hatton, and Jonathan W Hurst. Robot leg design: A constructive framework. *IEEE Access*, 6:54369–54387, 2018.

[65] Benjamin Katz, Jared Di Carlo, and Sangbae Kim. Mini cheetah: A platform for pushing the limits of dynamic quadruped control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6295–6301. IEEE, 2019.

[66] Brock Laschowski, John McPhee, and Jan Andrysek. Lower-limb prostheses and exoskeletons with energy regeneration: Mechatronic design and optimization review. *Journal of Mechanisms and Robotics*, 11(4), 2019.

[67] Rafael R Torrealba and Edgar D Fonseca-Rojas. Toward the development of knee prostheses: Review of current active devices. *Applied Mechanics Reviews*, 71(3), 2019.

[68] Sergei V Sarkisian, Marshall K Ishmael, Grace R Hunt, and Tommaso Lenzi. Design, development, and validation of a self-aligning mechanism for high-torque powered knee exoskeletons. *IEEE Transactions on Medical Robotics and Bionics*, 2(2):248–259, 2020.

[69] Toby Elery, Siavash Rezazadeh, Christopher Nesler, and Robert D Gregg. Design and validation of a powered knee–ankle prosthesis with high-torque, low-impedance actuators. *IEEE Transactions on Robotics*, 2020.

[70] Shuangyue Yu, Tzu-Hao Huang, Xiaolong Yang, Chunhai Jiao, Jianfu Yang, Yue Chen, Jingang Yi, and Hao Su. Quasi-direct drive actuation for a lightweight hip exoskeleton with high backdrivability and high bandwidth. *IEEE/ASME Transactions on Mechatronics*, 2020.

[71] D. Li and R. Qu. Sinusoidal back-emf of vernier permanent magnet machines. In *2012 15th International Conference on Electrical Machines and Systems (ICEMS)*, pages 1–6, 2012.

[72] Shiyoung Lee, Tom Lemley, and Gene Keohane. A comparison study of the commutation methods for the three-phase permanent magnet brushless dc motor. In *Electrical Manufacturing Technical Conference 2009: Electrical Manufacturing and Coil Winding Expo*, pages 49–55, 2009.

[73] James Robert Mevey. Sensorless field oriented control of brushless permanent magnet synchronous motors. 2009.

[74] James R Hendershot and Timothy John Eastham Miller. *Design of brushless permanent-magnet machines*. Motor Design Books Venice, Florida, USA, 2010.

[75] Richard Parsons. How to estimate the torque of a bldc (pmsm) electric motor using only its kv and current draw, Dec 2018.

[76] Jéröme Cros and Philippe Viarouge. Synthesis of high performance pm motors with concentrated windings. *IEEE transactions on energy conversion*, 17(2):248–253, 2002.

[77] Colm J. O'Rourke, Mohammad M. Qasim, Matthew R. Overlin, and James L. Jr Kirtley. A geometric interpretation of reference frames and transformations: dq0, clarke, and park. *IEEE Transactions on Energy Conversion*, 34(4):2070 – 2083, 2019.

[78] Mohamad Koteich, Thierry Le Moing, Alexandre Janot, and Francois Defay. A real-time observer for uav's brushless motors. In *2013 IEEE 11th International Workshop of Electronics, Control, Measurement, Signals and their application to Mechatronics*, pages 1–5. IEEE, 2013.

[79] Marek Musák and Marek Stulrajter. Novel methods for parameters investigation of pm synchronous motors. *Acta Technica Corviniensis-Bulletin of Engineering*, 6(1):51, 2013.

[80] Tae-Yong Lee, Jun-Young Song, Jaehong Kim, Yong-Jae Kim, Sang-Yong Jung, and Jung-Moon Je. Phase advance control to reduce torque ripple of brush-less dc motor according to winding connection, wye and delta. *Journal of Electrical Engineering and Technology*, 9(6):2201–2208, 2014.

[81] Jing Ye, Yasutaka Nakashima, Bo Zhang, Yo Kobayashi, and Masakatsu G Fujie. Functional electrical stimulation based on a pelvis support robot for gait rehabilitation of hemiplegic patients after stroke. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3098–3101. IEEE, 2014.

[82] David C Johnson, Diane L Damiano, and Mark F Abel. The evolution of gait in childhood and adolescent cerebral palsy. *Journal of Pediatric Orthopaedics*, 17(3):392–396, 1997.

[83] AM Boonstra, V Fidler, and WH Eisma. Walking speed of normal subjects and amputees: aspects of validity of gait analysis. *Prosthetics and Orthotics International*, 17(2):78–82, 1993.

[84] RL Waters, Jacquelin Perry, DANIEL Antonelli, and Helen Hislop. Energy cost of walking of amputees: the influence of level of amputation. *J Bone Joint Surg Am*, 58(1):42–46, 1976.

[85] Natalie Vanicek, Siobhan Strike, Lars McNaughton, and Remco Polman. Gait patterns in transtibial amputee fallers vs. non-fallers: Biomechanical differences during level walking. *Gait & Posture*, 29(3):415–420, 2009.

[86] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.

[87] Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

[88] Christian Igel, Nikolaus Hansen, and Stefan Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary computation*, 15(1):1–28, 2007.

[89] Riad Akrour, Marc Schoenauer, and Michele Sebag. Preference-based policy learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 12–27. Springer, 2011.

[90] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2002.

[91] Chandrayee Basu, Qian Yang, David Hungerman, Mukesh Sinahal, and Anca D Draqan. Do you want your autonomous car to drive like you? In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*, pages 417–425. IEEE, 2017.

[92] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems (TOIS)*, 30(1):1–41, 2012.

[93] Eric Baum and Frank Wilczek. Supervised learning of probability distributions by neural networks. In *Neural information processing systems*, 1987.

[94] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[95] Eytan Bakshy, Lili Dworkin, Brian Karrer, Konstantin Kashin, Benjamin Letham, Ashwin Murthy, and Shaun Singh. Ae: A domain-agnostic platform for adaptive experimentation. In *Conference on Neural Information Processing Systems*, pages 1–8, 2018.

[96] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, pages 1–15, 2015.

[97] Jürgen Wakunda and Andreas Zell. A new selection scheme for steady-state evolution strategies. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, pages 794–801, 2000.

[98] Max K Shepherd, Dean D Molinaro, Gregory S Sawicki, and Aaron J Young. Deep learning enables exoboot control to augment variable-speed walking. *IEEE Robotics and Automation Letters*, 7(2):3571–3577, 2022.

[99] He Huang, Fan Zhang, Levi J Hargrove, Zhi Dou, Daniel R Rogers, and Kevin B Englehart. Continuous locomotion-mode identification for prosthetic legs based on neuromuscular–mechanical fusion. *IEEE Transactions on Biomedical Engineering*, 58(10):2867–2875, 2011.

[100] Thomas Plötz, Nils Y Hammerla, and Patrick L Olivier. Feature learning for activity

recognition in ubiquitous computing. In *Twenty-Second International Joint Conference on Artificial Intelligence*, pages 1729–1734, 2011.

[101] Ming Zeng, Le T Nguyen, Bo Yu, Ole J Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. Convolutional neural networks for human activity recognition using mobile sensors. In *6th International Conference on Mobile Computing, Applications and Services*, pages 197–205. IEEE, 2014.

[102] Taeho Hur, Jaehun Bang, Jongwon Lee, Jee-In Kim, Sungyoung Lee, et al. Iss2image: A novel signal-encoding technique for cnn-based human activity recognition. *Sensors*, 18(3910):1–19, 2018.

[103] B. Hu, A. M. Simon, and L. Hargrove. Deep generative models with data augmentation to learn robust representations of movement intention for powered leg prostheses. *IEEE Transactions on Medical Robotics and Bionics*, 1(4):267–278, Nov 2019.

[104] Yuqing Chen and Yang Xue. A deep learning approach to human activity recognition based on single accelerometer. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1488–1492. IEEE, 2015.

[105] Andrea Zunino, Jacopo Cavazza, Riccardo Volpi, Pietro Morerio, Andrea Cavallo, Cristina Becchio, and Vittorio Murino. Predicting intentions from motion: The subject-adversarial adaptation approach. *International Journal of Computer Vision*, 128(1):220–239, 2020.

[106] Daniele Ravi, Charence Wong, Benny Lo, and Guang-Zhong Yang. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In *2016 IEEE 13th international conference on wearable and implantable body sensor networks (BSN)*, pages 71–76. IEEE, 2016.

[107] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. Data augmentation of wearable sensor data for parkinsonâs disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 216–220, 2017.

[108] Runfeng Zhang and Chunping Li. Motion sequence recognition with multi-sensors using deep convolutional neural network. In *Intelligent Data Analysis and Applications*, pages 13–23. Springer, 2015.

[109] Ben-Yue Su, Jie Wang, Shuang-Qing Liu, Min Sheng, Jing Jiang, and Kui Xiang. A cnn-based method for intent recognition using inertial measurement units and intelligent lower limb prosthesis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(5):1032–1042, 2019.

[110] Kuangen Zhang, Jing Wang, Clarence W de Silva, and Chenglong Fu. Unsupervised cross-subject adaptation for predicting human locomotion intent. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(3):646–657, 2020.

[111] Blair Hu, Elliott Rouse, and Levi Hargrove. Benchmark datasets for bilateral lower-limb neuromechanical signals from wearable sensors during unassisted locomotion in able-bodied individuals. *Frontiers in Robotics and AI*, 5(14):1–5, 2018.

[112] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, pages 437–442, 2013.

[113] K. J. Piczak. Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Sep. 2015.

[114] David A Winter. *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.

[115] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25, 2015.

[116] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[117] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[118] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[119] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[120] Kevin Englehart and Bernard Hudgins. A robust, real-time control scheme for multifunction myoelectric control. *IEEE transactions on biomedical engineering*, 50(7):848–854, 2003.

[121] Fan Zhang, Ming Liu, and He Huang. Investigation of timing to switch control mode in powered knee prostheses during task transitions. *PLOS one*, 10(7), 2015.

[122] Ung Hee Lee, Justin Bi, Rishi Patel, David Fouhey, and Elliott Rouse. Image transformation and cnns: A strategy for encoding human locomotor intent for autonomous wearable robots. *IEEE Robotics and Automation Letters*, 5(4):5440–5447, 2020.

[123] David Halliday, Robert Resnick, and Jearl Walker. *Fundamentals of physics*. John Wiley & Sons, 2013.

[124] Gene F Franklin, J David Powell, Abbas Emami-Naeini, and J David Powell. *Feedback control of dynamic systems*, volume 4. Prentice hall Upper Saddle River, 2002.

[125] SA Niapour, GH Garjan, Mehdi Shafiei, Mohammad Reza Feyzi, Saeed Danyali, and Mojtaba Bahrami Kouhshahi. Review of permanent-magnet brushless dc motor basic drives based on analysis and simulation study. *International Review of Electrical Engineering*, 9(5):930–957, 2014.

[126] James L Kirtley. *Electric power principles: sources, conversion, distribution and use*. John Wiley & Sons, 2020.

[127] Stephan Meier. Theoretical design of surface-mounted permanent magnet motors with field-weakening capability. *Master, Departement of Electrical Engineering, Royal Institute of Technology Stockholm, Stockholm*, 2002.

[128] Viktor Bobek. Pmsm electrical parameters measurement. *Freescale Semiconductor*, 7(8):13, 2013.