10-10-2018 2:00 PM

Development of Digital Control Systems for Wearable Mechatronic Devices: Applications in Musculoskeletal Rehabilitation of the Upper Limb

Tyler Desplenter
The University of Western Ontario

Supervisor
Trejos, Ana Luisa
The University of Western Ontario

Graduate Program in Electrical and Computer Engineering
A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

Development of Digital Control Systems for Wearable Mechatronic Devices: Applications in Musculoskeletal Rehabilitation of the Upper Limb

Tyler Desplenter
Ph.D. Thesis, 2018
Department of Electrical and Computer Engineering
The University of Western Ontario

Abstract

The potential for wearable mechatronic systems to assist with musculoskeletal rehabilitation of the upper limb has grown with the technology. One limiting factor to realizing the benefits of these devices as motion therapy tools is within the development of digital control solutions. Despite many device prototypes and research efforts in the surrounding fields, there are a lack of requirements, details, assessments, and comparisons of control system characteristics, components, and architectures in the literature. Pairing this with the complexity of humans, the devices, and their interactions makes it a difficult task for control system developers to determine the best solution for their desired applications. The objective of this thesis is to develop, evaluate, and compare control system solutions that are capable of tracking motion through the control of wearable mechatronic devices.

Due to the immaturity of these devices, the design, implementation, and testing processes for the control systems is not well established. In order to improve the efficiency and effectiveness of these processes, control system development and evaluation tools have been proposed. The Wearable Mechatronics-Enabled Control Software framework was developed to enable the implementation and comparison of different control software solutions presented in the literature. This framework reduces the amount of restructuring and modification required to complete these development tasks. An integration testing protocol was developed to isolate different aspects of the control systems during testing. A metric suite is proposed that expands on the existing literature and allows for the measurement of more control characteristics. Together, these tools were used

to developed, evaluate, and compare control system solutions.

Using the developed control systems, a series of experiments were performed that involved tracking elbow motion using wearable mechatronic elbow devices. The accuracy and repeatability of the motion tracking performances, the adaptability of the control models, and the resource utilization of the digital systems were measured during these experiments. Statistical analysis was performed on these metrics to compare between experimental factors. The results of the tracking performances show some of the highest accuracies for elbow motion tracking with these devices. The statistical analysis revealed many factors that significantly impact the tracking performance, such as visual feedback, motion training, constrained motion, motion models, motion inputs, actuation components, and control outputs. Furthermore, the completion of the experiments resulted in three first-time studies, such as the comparison of muscle activation models and the quantification of control system task timing and data storage needs. The successes of these experiments highlight that accurate motion tracking, using biological signals of the user, is possible, but that many more efforts are needed to obtain control solutions that are robust to variations in the motion and characteristics of the user.

To guide the future development of these control systems, a national survey was conducted of therapists regarding their patient data collection and analysis methods. From the results of this survey, a series of requirements for software systems, that allow therapists to interact with the control systems of these devices, were collected. Increasing the participation of therapists in the development processes of wearable assistive devices will help to produce better requirements for developers. This will allow the customization of control systems for specific therapies and patient characteristics, which will increase the benefit and adoption rate of these devices within musculoskeletal rehabilitation programs.

Dedicated to Joseph Henry Desplenter, great in virtue, rich in miracles.

Acknowledgements

My foremost appreciation goes to my doctoral thesis supervisor, Dr. Ana Luisa Trejos. Your openness to my research ideas and guidance in pursuing them played a critical role in this work and my academic development. The support and critique of this work by my advisory committee, Dr. Ken McIsaac and Dr. Ilia Polushin, is also greatly appreciated.

During my doctoral research, I was lucky to receive advice from and work with many talented individuals. I thank Dr. Arno Stienen for advising me and allowing me to explore my ideas in the ARMS Lab at the University of Twente. The efforts of Abelardo Escoto and Eugen Porter to help develop and troubleshoot the experimental software and hardware are greatly appreciated. I would like to recognize all of the experts that have helped shaped my ideas along this journey, including Dr. Mark Post, Dr. Petro Feketa, Dr. Andrew Pruszynski, Dr. Roy Eagleson, Dr. Charles Rice, Dr. Isha DeCoito, Shrikant Chinchalkar, Dr. Dave Walton, and Dr. Evan Friedman.

To my colleagues, Anastasiia Kyrylova, Myles Lidka, Raneem Haddara, Yue Zhou, Brandon Edmonds, Jacob Tryon, Taylor Stanbury, Nicole Devos, Emma Farago, Allison Goldman, Álvaro Julián Saldarriaga, José Guillermo Colli Alfaro, Anas Ibrahim, and those of you at the ARMS Lab, thank you all for making this journey more enjoyable and putting up with my question everything mentality.

I must thank my family, John Desplenter and Sharon Prevost, Sherri and Dale Kints, Astrid Desplenter, Charlie and Linda Simmons, Frank and Francis Kints, and the rest of the clan for their love and support during this endeavour.

Lastly, I am eternally grateful for the support that my wife, Sarah Byl, has given me. Her encouragement, love, trust, and understanding has allowed me to flourish. She has also blessed

me with our daughter, Kaelyn, during this endeavour. I have the deepest love for the two of them and they keep me focused on my dreams.

Contents

List of Figures

List of Tables

Nomenclature and Acronyms

Latin Letters

a	Muscle activation signal
A	Normalized muscle force–velocity relationship shaping factor
a_0	Transition point between linear and non-linear portions of $u(t)$–$a(t)$ relationships
A_1, A_2	Optimization parameters
$\bar{a}$	Mean muscle activation
a_{TI}	Total muscle activation, derived from the inverse optimization
a_{TF}	Total muscle activation, derived from the forward optimizaiton
b	Damping coefficient
b_d	Number of bytes representing a data value in computer memory
b_i	Number of bytes representing a processor instruction
B	Matrix of torques
c	Parameter of a muscle activation model
C	Matrix of coefficients
c_d	Normalized muscle force–velocity relationship constant
d	Electromechanical delay
e_a	Total muscle activation error
e_T	Joint torque error
e_{RMS}	Root-mean-square error

e_θ	Joint position error
emg	EMG signal
EMG_{BB}	EMG signal of the biceps brachii muscle
emg_{max}	Maximum EMG signal
emg_{min}	Minimum EMG signal
emg_{norm}	Normalized EMG signal
EMG_{TB}	EMG signal of the triceps brachii muscle
g	coefficient of the Hill shortening equation
G	Function of noise
g_{max}	Maximal normalized eccentric force
G_{na}	Neural activation amplification constant
f	Force–length relationship of the muscle
F	Muscle force
F_{active}, F_{CE}	Active force generated by the contractile element
f_c	Cut-off frequency
F_{fl}	Normalized isometric force–length relationship of the muscle
F_{fv}	Normalized force–velocity relationship of the muscle
F_{max}	Maximum isometric force
F_{RTE}	Rigid tendon element force
F_{TM}	Total muscle force
I	Inertial mass
K_{BB}	Proportional constant of the biceps brachii EMG signal
K_{TB}	Proportional constant of the triceps brachii EMG signal
K_p	Proportionality constant
k_1, k_2	Stiffness coefficients
L_m, L_M	Length of the muscle fiber

$\dot{L_M}$	Contraction velocity of the muscle fiber
$\dot{L}_{M_{max}}$	Maximum muscle fiber shortening velocity
L_{M_o}	Optimal length of the muscle fiber
L_{MT}	Length of the musculotendon unit
L_S	Muscle slack length
m	Parameter of a muscle activation model
M	Moment generated by the muscle about the joint
M_m	Mass matrix
m_G	Gearhead reduction ratio
m_{G1}	Reduction ratio of the first gearhead
m_{G2}	Reduction ratio of the second gearhead
M_g, M_G	Moment due to gravity
M_P	Passive joint torque
M_T, M_{joint}	Total joint moment
M_{TM}	Total muscle torque
N	Number of samples
N_b	Number of bytes of data
N_{dp}	Number of data points
N_m	Number of muscles
N_p	Number of processing cores
P	Function of noise
P_u	Processor usage
p_d	Constants of the polynomial
$p_{1_{BB}}$–$p_{4_{BB}}$	Coefficients of the polynomial representing the biceps brachii muscle torque
$p_{1_{TB}}$–$p_{4_{TB}}$	Coefficients of the polynomial representing the triceps brachii muscle torque
Q	Measurement noise constant

r	Moment arm
R	Process noise constant
$S_{program}$	Size of a program
t	Time
T	Joint torque
$\bar{T}$	Mean joint torque
T_a	Time activation constant
T_d	Derived joint torque
T_{BB}	Torque generated by the biceps brachii muscle
t_e	Task execution time
T_e	Estimated joint torque
t_{end}	The time at which a task has completed its execution
t_{dp}	Data point optimization time
t_{opt}	Execution time of optimization task
$t_{program}$	Time taken to execute a program
t_{start}	The time at which a task begins its execution
T_{TB}	Torque generated by the triceps brachii muscle
t_{total}	Total available processing time
u	Neural activation signal
u_{BB}	Neural activation of the biceps brachii muscle
u_T	Summation of neural activation signals
u_{TB}	Neural activation of the triceps brachii muscle
u_0	Transition point between linear and non-linear portions of $u(t)$–$a(t)$ relationships
$V_{storage}$	Data storage velocity
W	Normalized muscle force–length relationship shaping factor
x	Measured value

$\hat{x}$	Estimated value
x_0	Initial position

Greek Letters

α	Neural activation model coefficient
β	Parameter of a muscle activation model
β_1, β_2	Neural activation model coefficients
Δ	Change in position
θ	Joint angle
$\dot{\theta}$	Joint angular velocity
$\ddot{\theta}$	Joint angular acceleration
$\bar{\theta}$	Mean joint position
θ_c	Corrected joint position
θ_e	Estimated joint position
θ_{GH2}	Position output of the second gearhead
θ_i	Input position
θ_m	Measured joint position
θ_{motor}	Positon output of the motor
θ_o	Output position of the gearhead
θ_p	Predicted joint position
σ_{e_θ}	Standard deviation of joint position error
σ_{e_a}	Standard deviation of muscle activation error
σ_M	Standard deviation of torque error
τ	Summation of torque values
ϕ	Pennation angle
ϕ_o	Pennation angle at the optimal muscle fiber length

Acronyms

ADC	Analog-to-Digital Converter
ADL	Activities of Daily Living
ANN	Artificial Neural Network
BCI	Brain–Computer Interface
CE	Contractile Element
EEG	Electroencephalography
EMD	Electromechanical Delay
EMG	Electromyography
EOG	electrooculography
HTMM	Hill-Type Motion Model
ICA	Independent Component Analysis
IMHA	Institute of Musculoskeletal Health and Arthritis
IMU	Inertial Measurement Unit
KFMM	Kalman Filter Motion Model
MAM	Muscle Activation Model
MFANN	Multilayer Feedforward Artificial Neural Network
MSD	Musculoskeletal Disorder
MVC	Maximum Voluntary Contraction
ND	Not Defined
NL	Not Listed
NPMM	Non-Linear Polynomial Motion Model
NU	Not Used
PI	Proportional-Integral

PID	Proportional-Integral-Derivative
PMM	Proportional Motion Model
PPE	Parallel Passive Element
RMS	Root-Mean-Square
RMSE	Root-Mean-Square Error
RTE	Rigid Tendon Element
SEA	Series Elastic Actuator
sEMG	Surface Electromyography
SPE	Series Passive Element
TDANN	Time Delay Artificial Neural Network
UWB	Ultra-Wide Band
WearME	Wearable Mechatronics-Enabled
WearMECS	Wearable Mechatronics-Enabled Control Software

Chapter 1

Introduction

1.1 Motivation

Musculoskeletal disorders (MSDs) encompass any form of trauma, condition or disease that negatively impacts the ability of the individual to produce and control the motion of their body. According to the Canadian Institute of Musculoskeletal Health and Arthritis (IMHA), 11 million people over the age of 12 in Canada are affected by musculoskeletal disease and injury each year [1]. Based on recent Canadian population estimates, this means that 30% of Canadians are suffering from these diseases and injuries [2]. For a musculoskeletal injury, a medical professional may be required to assist with recovery anywhere from a few weeks up to years. In the case of musculoskeletal diseases, professional care may be a life-long requirement. As a result, MSDs put strain on the patients, their families and the medical system. The IMHA reports musculoskeletal disease and injuries costing $22 billion and $15 billion each year, respectively, where three-quarters of the costs are related to the financial burden from individuals' inability to work [1]. Under the current medical system, much of the expense in dealing with MSDs is not covered, further increasing the financial burden on the patient and, in many cases, forcing the patients to withdraw from professional care. The consequences of removing one's self from care can lead to life-long motion impairment and a lower quality of life.

Over the last few decades, technology has been under development to provide solutions that aid with the diagnosis, assessment, and treatment of MSDs. Brain–Computer Interfaces (BCIs) allow

researchers to study the connections between neurological patterns and corresponding motion, which helps in the diagnosis of diseases. Systems that enable lost or impaired motion may also be controlled through BCIs. Electromyography (EMG) provides methods for analysis of the electrical signals sent to the musculature from the brain. Using EMG, techniques to stimulate motion have been developed as tools to aid with musculoskeletal rehabilitation, such as Functional Electrical Stimulation. Most recently, mechatronic devices have incorporated non-invasive EMG sensors into their design in order to allow individuals to control these devices, driven by the signals acquired from their muscles. With these technologies, the problem of providing motion assistance to those suffering from MSDs can be mitigated.

Even in the presence of current technologies, the scope must be first be narrowed in order to solve this problem. The effects of musculoskeletal injuries are felt by large populations. According to Canadian statistics, 3,427,046 cases of injury severe enough to limit daily activities were reported in 2004 [3] with that number growing to 4,272,000 per year by 2010 [4]. Since injury is universal to all people, the problem is greatly exaggerated if injury incidence is looked at worldwide. However, musculoskeletal injuries have the highest chance of being remedied and, therefore, are a good area for concentration. As a result, mechatronic devices have been undergoing development for the intention of being integrated into musculoskeletal rehabilitation protocols. This is being done in an effort to achieve better recovery outcomes and reduce the overall medical, social, and economic burden that injury causes.

Traditional musculoskeletal therapy involves motion of the affected tissues in order to shape their structures as they heal. Over many centuries, these therapies have evolved and proven to be very successful to restoring natural motion production and control. The physical therapy offered today is dependent on the expertise and ability of the therapist to assess, prescribe, and assist motion. Parts of rehabilitation protocols require a large amount of manual labour from the therapist, as they move or assist in moving the affected tissues of their patients. Using mechatronic devices, the manual labour can be employed to the device, instead of performed by the therapist. Therefore, patients are able to receive motion therapy without a physical connection to the therapist, while therapists are able to focus their energy on other aspects of the patient's rehabilitation. However, using currently available devices, the patients must still travel to the clinic to reap the benefits,

as these devices are large, stationary, and expensive. To extend the benefits of these devices to a larger population, the focus must be to make them light, portable, and controllable with little or no supervision.

1.2 Moving Toward Wearable Solutions

The problem of being unable to perform a desired motion is universal to all human beings. As a child, your parents and community help you to learn body control by demonstration and helping you to lift the cup to your mouth. Throughout life, most people will suffer from an injury that impairs their ability to complete a desired motion or at least forces a change in the motions used to complete a specific task. Even healthy people require assistance when moving objects that are too heavy, large or awkward to move alone, such as a couch. As our bodies age and begin to break down, help with motion is again necessary. For centuries, humans have been intuitively developing devices to help with a lack of motion ability, such as crutches, canes, or wheelchairs.

More recently, technology has enabled the development of mechatronic devices, which are the combination of mechanical structures with electrical and computational components. These devices have been developed to enhance the assistive abilities of their purely mechanical predecessors. A person who is unable to walk can now purchase a wheelchair in which they control ground movement in the full upright position and can even maneuver up and down stairs. An individual lacking arm motion as the result of an injury or stroke can now find stationary robotic arms in rehabilitation clinics that assist with motion to enable proper healing of their musculoskeletal tissues. The efficacy of these mechatronic devices in providing motion assistance is currently under investigation, but existing results are positive. However, the new problem lies in making them safe, easy-to-use, and portable.

Many improvements to these mechatronic devices are being explored to increase the wearability and types of assistance that these devices can produce. One part of the solution to preparing these devices for usage in musculoskeletal rehabilitation lies in the development of control systems for these devices. Equipping these control systems with the proper components will enable the control of smaller actuation systems and transmission of data through wireless channels, both of which

increase the wearability of these devices. Better models of human motion can be used within these control systems to enable the use of data collected from sensors mounted on the body, as opposed to sensors mounted in the environment, such as cameras. Determining requirements for the level and type of assistance that is required for a specific rehabilitative therapy will inform the design of the control system. Control components can be implemented to meet these requirements and produce assistive forces similar to those produced by therapists. There are existing assistive control strategies, but many have not been extensively tested or still require improvement. The development of wearable mechatronics devices that can assist with rehabilitative motion tasks could provide a solution to the growing demand for musculoskeletal rehabilitation services.

1.3 General Problem Statement

Rehabilitative mechatronic devices have been under clinical evaluation for the better part of two decades and shown that they can provide motion assistance equivalent to that of a therapist [5, 6]. More recently, commercially available wearable versions of these devices have also shown promise when used in rehabilitation programs for stroke survivors [7, 8]. These devices can also accomplish even higher accuracy and repeatability of motion patterns than expert therapists if programmed and controlled properly. However, the difficulty in formulating the behaviour of these systems, or their *complexity*, makes determining an appropriate control solutions a difficult problem. Essentially, there are three entities, the patient, the therapist, and the device, interacting to complete the desired motion therapy. The patient's motion and the mechanisms that control it are not fully defined and have their own set of complex interactions. Adding to this, each patient has a unique variation of these motion generating mechanisms, making modeling a difficult task. Therapists, and all their expertise, add another layer of complexity, as the prescribed therapy must be individualized to meet, not only, the unique characteristics of each patient, but also, the unique aspects of the injury. Therapist expertise evolves over time and varies based on their clinical environmental conditions. As a result, modern rehabilitation therapy is filled with subjectivity that presents devices with a larger set of requirements for which to account. Mechatronic devices themselves are comprised of complex sub-systems and sub-system interactions. Facilitating their

own safe functionality, as well as integrating all required functionality demanded by the patient and therapist, such that these devices can be used with minimal supervision, creates a difficult scenario for control system engineers.

As with all engineering problems, the solution lies within the decomposition of the problem into sets of smaller problems grounded in many assumptions. Using this concept, hundreds of control designs have been presented in the literature as solutions to using mechatronic devices to aid in motion assistance. Due to the complexity of these devices and their applications, the solutions have been developed in an *ad hoc* fashion. Therefore, if a control engineer wanted to test a control design, as a possible solution to their own control problem, they must spend large amount of effort and resources to reimplement the proposed control technique on their device. This method of determining the viability of alternative solutions is cumbersome and opposite to the principles of engineering. In order to enhance the application of wearable mechatronic devices within rehabilitation processes, effective control systems are required that can model and adapt to the complexities of the interactions between the patient, the therapist, and the device. Knowing whether a control technique or design is effective requires a process of comparison, which currently is a resource-intensive activity given the number of implementation options that are available. Essentially, the control system developer is required to sift through all of the research to determine the answer to the question: what control solutions works best for operating my device within its intended application?

The focus of this thesis is to answer this question, with the scope narrowed to wearable mechatronic devices providing assistance during musculoskeletal rehabilitation therapies. It is possible to control these devices using the motion intention captured through the biological signals of the user. However, it has been stated recently by Gopura *et al.* that "identifying the exact human motion intention is still under [investigation] at [the] research level... Therefore, understanding and optimizing the best control method is difficult." [9]. Given the complexity of human motion and the interactions required during assistive rehabilitation therapies, it is clear as to why developing control systems for these purposes is difficult. The general lack of studies in the literature that involve the performance measurements of these control systems and devices, suggests that these devices are still in their infancy in terms of their readiness for rehabilitation. There is also

a general lack of requirements to guide the development of these devices for motion therapies. Together, these aspects leave many opportunities for exploring and improving upon the existing control solutions.

1.4 Research Objectives

In order to address the problems stated above, the objective of this thesis is to develop, evaluate, and compare control system solutions that are capable of tracking elbow motion through the control of wearable mechatronic devices. Completing this objective will push forward toward the answer to the question, what control solutions works best for operating my device within its intended application? Furthermore, the studies conducted in this thesis which involve development, evaluation, and comparison of these control systems, provide insights into the aspects that affect the performance of motion tracking. Based on the existing literature, there is a need for the quantification and comparison of control system solutions to enable developers to focus their efforts. This need will be addressed through the completion of the research objectives.

The problems that this thesis explores complex. As a result, the following presents a more detailed set of primary objectives:

1. to review the literature to determine control system characteristics, components, and architectures,

2. to create control system development tools to support the development and assessment of control solutions,

3. to use these tools to develop control systems capable of tracking elbow motion,

4. to evaluate and improve upon the existing elbow motion estimation models,

5. to perform comparative studies to determine factors that affect the control system tracking performance,

6. to determine requirements for software systems that will allow integration of these devices into rehabilitative therapies, and

7. to discover limitations and areas for improvement of these control systems.

Completion of these objectives will help inform the development of control solutions that are capable of tracking elbow motion using wearable assistive devices. Highly accurate motion tracking will enable the performance of motion therapies using wearable mechatronic devices. The research completed within this thesis will benefit control system and wearable mechatronic devices developers, therapists, and, most importantly, patients suffering from musculoskeletal injury or disease.

1.5 Overview of the Thesis

The structure of this thesis is summarized as follows:

Chapter 1 Introduction: Introduces the motivation, general problem, and objectives.

Chapter 2 Computation Fundamentals of Wearable Assistive Devices: Reviews the literature surrounding five fundamental areas of computation related to digital control system development. These areas include human–machine interfacing, signal processing, communication, human motion modelling, and controlling device behaviours.

Chapter 3 Control System Development and Evaluation Tools: Proposes tools to aid with the development and evaluation of control system solutions in order to more effectively assess their performance. The contribution from this chapter is a set of tools that are used to design, implement, and test motion tracking control solutions.

Chapter 4 Design and Implementation of Control Solutions: Presents the design of control system components and the aggregation of these components into control system solutions for the purposes of tracking human motion.

Chapter 5 Evaluation of Motion Tracking Control Systems: Outlines seven experiments conducted to evaluate the performance of the developed control solutions during an elbow motion tracking task. Testing the efficacy of the development and evaluation tools, performance improvements compared to existing solutions, identification of

factors that affect the control performance, and comparisons of control solutions are the main contributions of this chapter.

Chapter 6 Applications in Musculoskeletal Rehabilitation: Outlines the opportunities for tailoring control systems within different musculoskeletal rehabilitation therapies. Results from a national survey of therapist data collection and analysis methods and software requirements derived from the survey are the contributions of this chapter.

Chapter 7 Conclusions: Highlights conclusions and contributions of this work. Future directions for the development of digital control systems are also presented.

Chapter 2

Computational Fundamentals of Wearable Assistive Devices

2.1 Introduction

Wearable assistive devices are a realization of an opportunity to enhance the ability of human motion performance. The benefits of successfully facilitating this integration extend beyond the reality of human capabilities. Wearable assistive technology is a relatively new approach to helping humans move and, as a result, will require many efforts from researchers, clinicians, law makers, and end-users to reach its potential. The complexity of these devices, the human body, and their interactions presents a number of avenues for study and improvement. The road pursued within this thesis is the one leading towards understanding the computational aspects required to enable useful human–machine interactions. The computer systems of wearable assistive devices facilitate perception of the environment, communication with the user, analysis of biological signals, modelling of human motion, and production of assistive forces. In this chapter, wearable assistive devices will be defined, the computational knowledge areas related to these devices will be explored, and the state-of-the-art in these knowledge areas will be presented.

2.2 Definitions of Wearable Assistive Devices

According to Gopura *et al.*, humans have been working on creating exoskeleton-based systems for the purpose of assisting humans with motion since 1883 [9]. Two forms of these devices, the end-effector robot and the robotic exoskeleton, have become popular over the last few decades due to technological advances. End-effector robots produce motion of the user through contact at the end-effector. Typically, the user will be attached to or grip the end-effector in order to cooperatively complete a motion. When the user is weak, the end-effector robot will guide the user through a specific motion. As the user increases his or her strength, the user will guide the robot through the motion while being assisted as needed. The end-effector robot motion is produced through a series of joints and links. Due to their large size, these systems are mounted to rigid surfaces such as a floor or wall. The effectiveness of these robots stems from their large work space that enables various activities of daily living (ADLs) to be practiced, such as brushing teeth or reaching for a cup. Many research projects have led to the development of end-effector robots for rehabilitation purposes, including popular systems such as MIT-MANUS [10], GENTLE/s [11], and Rehabrob [12].

Two major limitations have shifted the focus from end-effector robotic systems to exoskeletons systems. First, the size and cost of the end-effector robots limits where they can be used. Medical centers with large budgets are the only organizations that can afford to purchase these systems. The large size and weight of the systems means that they cannot be taken home by patients. The other major limitation is the inability of the end-effector systems to control the user's joints independently. For example, rehabilitative therapy begins by retraining joint motion separately until they are strong enough to complete more complex multi-joint tasks. Therefore, end-effector robots, in general, can only provide multi-joint assistive, resistive or functional motion therapy.

Robotic exoskeletons have become the most common form of assistive devices due to an increased level of assistability compared to end-effector robots. The popularity of these systems stems from their ability to control the motion of each joint independently. The purpose of exoskeletons, in rehabilitation scenarios, is to complement the ability of the human limb and restore the handicapped function [13]. Exoskeletons achieve their purpose by carefully mapping their motions to

those of the underlying human structures. A huge area of research has arisen from the initial conception of exoskeleton systems. Today, exoskeleton systems are being designed and fabricated in research facilities around the world. upper limb exoskeletons aiding with musculoskeletal rehabilitation tasks have shown a similar level of effectiveness compared to the same amount of exercise performed by trained therapists [5, 6]. The opportunities for these systems to help mitigate the growing rehabilitation costs and improve the outcome of rehabilitation have manifested themselves into commercially available devices [14, 15]. While robotic exoskeletons, and end-effector robots in some cases, are worn by the user, their large size, weight, and technical requirements make them stationary objects at clinics, which the user must visit in order to gain their benefit. In order to enhance the potential for these devices to provide a larger societal benefit, research and industrial sectors are moving towards development of wearable assistive devices.

In this thesis, wearable assistive devices are defined as mechatronic systems that provide assistance to humans during motion, much like exoskeleton or end-effector robots. Wearable and assistive are the two main qualities that separate these types of devices from other mechatronic systems. The term *wearable* is defined as having the base of the manipulator or system originate from one or more attachment points on the human body. Many of the characteristics of wearable assistive devices have stemmed from the development of end-effector and exoskeleton manipulators. Although these devices are attached to the human body, the bases of these systems are fixed to points in the environment, not on the human. Therefore, under this definition, these devices are not wearable, even though they may be portable. *Assistive* is defined as the production of motion that assists the user to accomplish their desired goal. Humans exhibit both desired and undesired motion of their bodies. An assistive device produces forces that assist the user in producing a desired motion, such as in rehabilitation of injured tissues. However, a device that produces forces to suppress undesired motion may also be considered assistive if the goal of the user is to reduce or eliminate the undesired motion. Understanding these two qualities is important to identifying devices that fit into the scope of this thesis.

There are many factors that contribute to a device being wearable and assistive. One can view the wearable and assistive characteristics as a spectrum, instead of binary qualities, by considering them as assistability and wearability. In terms of robotic manipulators, assistability has been

defined as the ability of one or more manipulator joints to assist another joint with performing a desired motion [16]. However, a modification to this definition is required since, in the case of wearable assistive devices, there are two interacting systems each with one or more joints. Instead, it is proposed to look at assistance from the view of the actuation systems performing the motion. With this in mind, assistability is defined as the ability of one or more systems to assist a primary system with the production of motion. This definition does not restrict the configuration of any of the interacting systems. Extending this to assistive devices, *assistability* is defined as the ability of the assistive device to assist the human with the production of motion to meet their desired motion goal.

In general, *wearability* is a characteristic of an object that quantifies the level to which it can be worn on the human body. If a device is tethered to a location or is portable but requires you to drag a bunch of equipment around with you, it would be seen to have low wearability. Wearability is often used with the intent that it encompasses many other characteristics. Park *et al.* define wearability as taxonomy of characteristics [17]. An example taxonomy of a wearable system is shown in Fig. 2.1. The ISO/IEC 25010 Software Product Quality provides a similar template to build customized software product quality taxonomies [18]. However, no standard exists for product wearability to date. Borrowing from software quality engineering, one can apply the principles governing quantification of software qualities to construct a customized hierarchy of wearability-related characteristics that can be quantified and compared between devices. Although this is an important aspect for improving wearable assistive devices, the detail lies outside of the scope of this thesis.

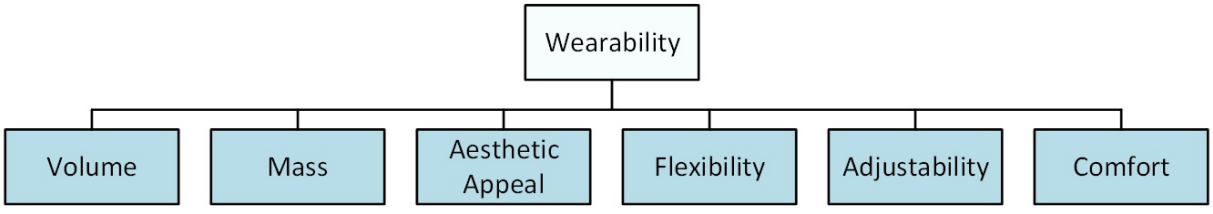


Figure 2.1: A wearability quality taxonomy to determine the level of wearability of assistive devices.

2.2.1 Implementing Assistability and Wearability

In the previous section, the principles that separate wearable assistive devices from other mechatronic systems were defined. Now, the focus shifts to the question of *how are assistability and wearability implemented?* In general, wearable assistive devices are developed using a mechatronics perspective of device design. A mechatronic system consists of mechanical, electrical, computer, and control systems (Fig. 2.2). These systems must interact in an intelligent manner in order to perceive the environment, including the user and their intentions, and provide some benefit to the user or environment. In the context presented, the benefit is simply assisting humans with motion production, though the realization of this benefit is rather complex.

The prescription and delivery of assistance may vary based on the user size, presence of MSDs, affected location, therapist opinion, therapy requirements, and many other factors. However, there is a general procedure that can be abstracted from motion assistance scenarios that all mechatronic devices can build upon to provide motion assistance functionality (see Fig. 2.3). First, the device must perceive the motion intention of the user, which is typically accomplished through sensing of biological signals. Second, the device must use internal models to estimate the intended human motion behaviour. Third, calculation and decisions must be made as to how to react to the user, such that the device provides the appropriate assistance while maintaining safety of both the user and itself. Finally, the decisions are translated into motion commands that are sent to the device's actuation systems to produce the desired motion of the device and user, simultaneously. In order to accomplish this procedure, many constraints are placed on the mechatronic system. Sensors that are able to perceive biological signals require high resolutions and accuracy. Well-established motion models are needed in order to provide accuracy estimates of human motion and drive decision making algorithms, and high-power low-weight actuation systems are needed to ensure sufficient motion generation forces.

Implementing a high level of wearability also poses many constraints on the sub-systems of these devices. The weight of the entire system is crucial, especially in cases where motion production abilities are severely impaired. If muscles are unable to generate appropriate forces to move body segments, adding load to these segments may cause negative impacts to the user. The mechanical

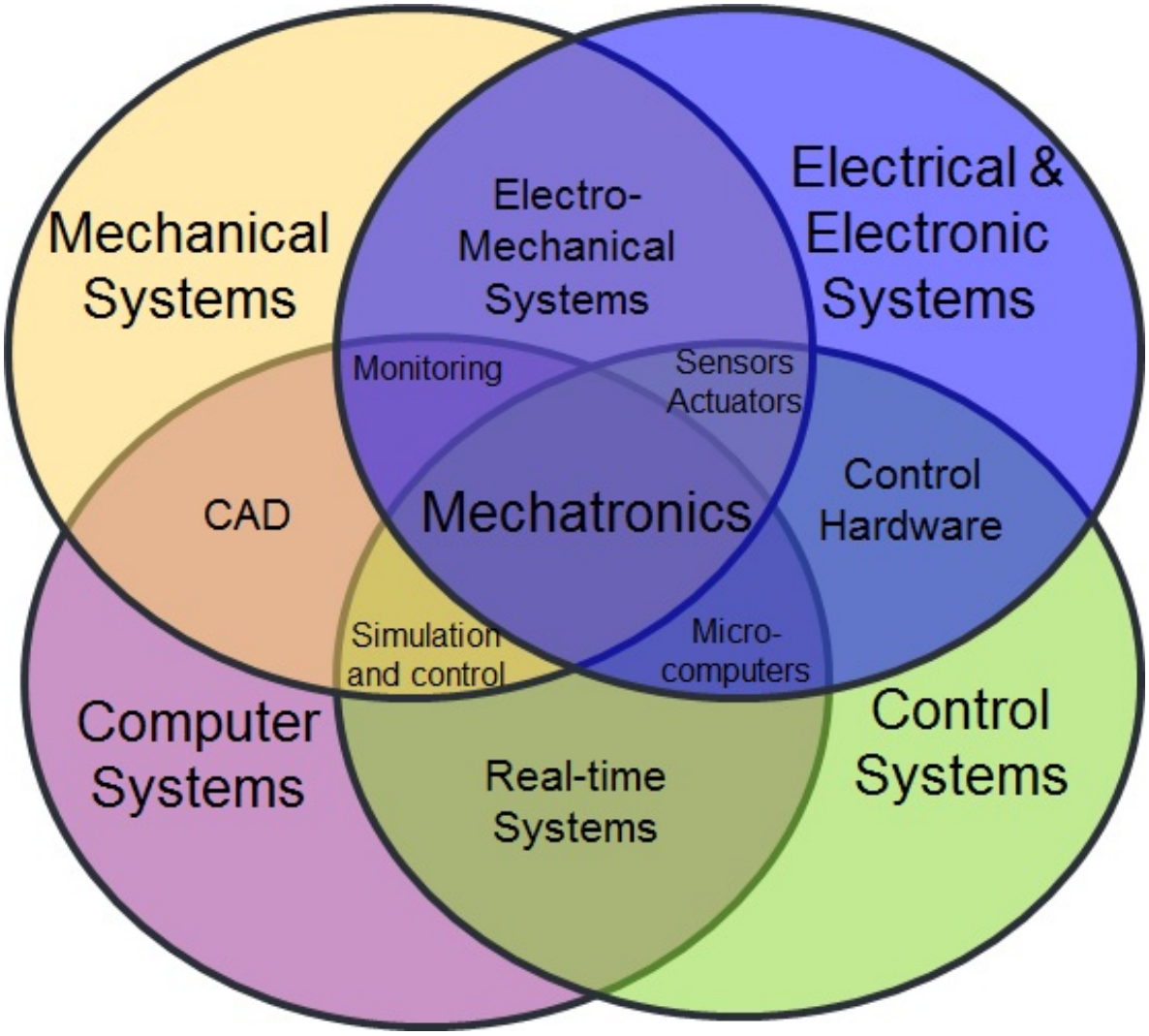


Figure 2.2: Definition of Mechatronic Systems, represented as a combination of four other types of systems. Adapted with permission from [19].

interface between the human skin and the device must consider the contact forces between these two bodies. If the force is not distributed appropriately, generating large shearing forces across a section of the skin, the interface may cause discomfort or even damage to the soft tissues of the user. The volume of the device is crucial to minimize in order to cause the least disruption to the user's normal motion patterns. For example, there is little space in general between the torso and the medial side of the upper limb within which a device can exist. Furthermore, increasing wearability by allowing the user to walk while wearing the device requires careful planning on placement of components and minimization of component volumes. Another reason to consider

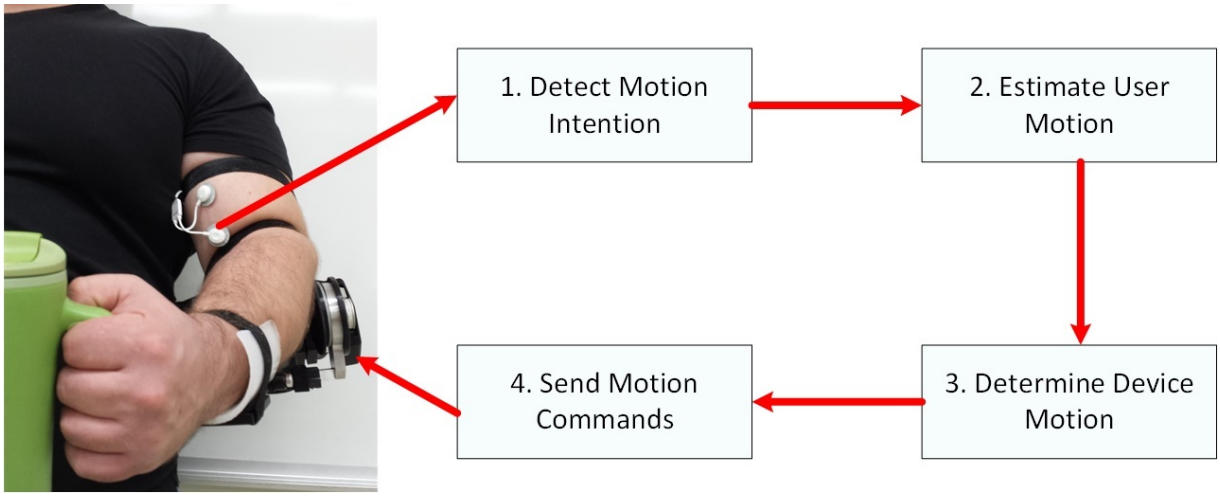


Figure 2.3: The general steps for implementing assistive behaviour in the control of wearable assistive devices.

volume as a design requirement is due to the change in dynamic properties as seen by the human motor control system. Systems that produce a large change in the inertial mass of the upper limb will affect the user's motion control signals and, therefore, the produced motion. A device with this characteristic is still wearable but may not rank high in wearability if maintenance of natural motion patterns is considered to be an important design specification.

Sensors, actuators, and control systems all influence wearability factors such as weight and volume. The larger the set of sensors, the larger the size or number of actuators or the more complex the control method, the more computational resources will be required to interface with and control the behaviour of these sub-systems. As will be explored in more detail, the computational demand of the control methods alone is currently limiting the ability of these devices to become more wearable. Furthermore, although circuitry is relatively light in weight compared to other components of these devices, the batteries required to power these devices can increase the total weight bearing on the user substantially. The volume required to house both the circuitry and the batteries is quite large relative to the available locations on the body. Typically, harnesses, straps, belts, or bags are the solution to house these components as their volume and mass added directly to the device would decrease its wearability.

Currently, it is difficult to achieve high levels of assistability and wearability given the state-of-the-art of relevant technologies. However, the curious can look for trends in order to abstract

guidelines that aid development and future improvements. As with any mechatronic device, the rule to follow is to design the system with the specifications of all sub-systems considered simultaneously to provide the best results. The following section provides the reader with some examples of wearable assistive devices in order to highlight existing solutions.

2.2.2 Existing Wearable Assistive Devices

Wearable assistive devices for the upper limb are still an emerging field with a large amount of research remaining to be conducted. End-effector robots were able to provide a moderate level of assistability to gross movements, but exhibit low wearability. Exoskeletons allow for increases in assistability due to the concept of providing assistance for each joint individually but also rank low in terms of wearability as they still remain as stationary devices. The portable, wireless assistive devices presented by the research community are the results of generating devices that attempt to increase both assistability and wearability or, at least, attempt to find a balance between these two characteristics.

Although it has been discussed that exoskeleton-based assistive devices are not specifically wearable, it would be fruitful to review the existing exoskeleton technology as wearable assistive devices borrow heavily from these forebearers. Reviews of existing exoskeleton and wearable assistive devices can be found in the literature. In 2011, Gopura *et al.* provided a brief examination of upper limb exoskeletons, categorized by actuation type [20]. In 2012, Lo and Xie investigated existing upper limb exoskeleton systems with a summary of the key challenges faced by these systems [21]. In 2014, Maciejasz *et al.* compared over 120 rehabilitation systems, revealing the shift towards wearable devices [22]. Gopura *et al.* reviewed the mechanisms, actuators, and transmission systems of existing upper limb exoskeletons in 2016 [9].

The human elbow is a popular choice for researchers developing wearable assistive systems due to the relative simplicity of the joints, large area for placement of components, and availability of the major muscles that move the elbow. As a result, many devices are aimed at assisting with elbow motion. Even for this simple case, there are still many limitations within these devices. The REHAROB Therapeutic Systems [23], the NEUROExos [24], the WearME elbow brace 2.0 [25], and, finally, the MyoPro [26] will be used to highlight the evolution towards wearable devices. First,

the REHAROB Therapeutic System is a dual end-effector robotic system that was developed for passive movement therapy [23]. Although the device is worn by the user, this system has low wearability due to its large size and immobility and low assistability, as it was designed to assist with completion of passive exercise tasks. The NEUROExos is an elbow exoskeleton developed with the ergonomics of human–machine interfacing in mind [24]. Although this device is also stationary, the developers improved upon the interfacing material between the device and the user's skin and added a passive degree of freedom to account for the natural variability in elbow motion. Next, the WearME elbow brace 2.0 prototype was developed in order to enhance aspects of both wearability and assistability [25]. Adjustable links and 3D-printable cuffs were included in this design to allow for variation in user dimensions, while a dual-sided actuation system, providing evenly distributed torque to both sides of the elbow, and improvements to predictive motion models were added to increase the potential in assistance accuracy. Finally, the MyoPro is currently the only commercially available elbow orthosis designed to maximize wearability. The small volume, single-sided actuation system and self-contained computer and power systems give it a high level of wearability. However, limited torque and simplistic control methods leave much room for improvement in terms of the assistance provided to the user. Multiple clinical studies, using the MyoPro to assist with rehabilitation activities for individuals suffering from stroke, conclude that rehabilitation using this device is as effective as traditional manual therapy and improves performance during functional tasks when worn [7,8,27,28]. The successes of the MyoPro support further inquiry into these technologies and show a promising future for wearable assistive devices.

Apart from the MyoPro elbow device, the state of wearable assistive devices lies within the research realm. A total of 52 wearable assistive mechatronic upper limb devices have been identified from the literature. The complexity of the technology and the human body has caused a huge expansion of research into these devices over the last decade. In fact, 54% of the articles reviewed were published within the last 5 years. The multidisciplinary nature of these devices coupled with the immaturity of this field had lead to a large diversity of the technologies used. An overview of the major technologies used in these devices can be found in Table 2.1. Devices in this table have been categorized by the joints they actuate. Brief details about their supported motions, sensed quantities, actuation systems, control quantities, and application are listed.

Table 2.1: An overview of existing wearable assistive mechatronic devices.

Author [Reference]	Supported Movements	Sensed Quantities	Actuation	Controlled Quantities	Type of Assistance
Shoulder, Elbow, and Wrist					
Sugar [29]	Shoulder-F Elbow-F, S Wrist-E	Position, pressure	Pneumatic actuator	Torque	Support
Shoulder and Elbow					
Brackbill [30]	Shoulder-FE, AbAd, IE Elbow-FE	Not listed	Electric motor	Position	Support
Lessard [31]	Shoulder-Ab Elbow-FE, PS	Position, heart rate	Electric motor	Torque	Support
Elbow and Wrist					
Rocon [32]	Elbow-FE, PS Wrist-FE	Velocity, force	Electric motor	Torque, velocity	Suppression
Ueda [33]	Elbow-FE, PS Wrist-FE, RU	Position, EMG	Pneumatic actuator	Force	Support
Xiao [34]	Elbow-FE, PS Wrist-FE, RU	EEG	Electric motor	Position	Support
Elbow					
Ando [35]	Elbow-FE	EMG	Electric motor	Torque	Suppression
Benitez [36]	Elbow-FE	EMG, position, torque	Electric motor	Torque	Support
Desplenter [37]	Elbow-FE	EMG	Electric motor	Velocity	Support
Herrnstadt [38]	Elbow-FE	Position, velocity	Electromagnetic brake	Torque	Suppression
Kim [39]	Elbow-FE	EMG, pressure	Pneumatic actuator	Torque	Support
KleinJan [40]	Elbow-FE	Position	SMA	Torque	Support
Kyrylova [25]	Elbow-FE	EMG, position	Electric motor	Position, velocity	Support

Elbow					
Looned [41]	Elbow-FE, PS	EEG, position	Electric motor	Torque	Support
Pylatiuk [42]	Elbow-FE	EMG, position, pressure	Hydraulic actuator	Force	Support
Ren [43]	Elbow-FE	Position	Electric motor	Torque	Support
Stein [26,27]	Elbow-FE	EMG	Electric motor	Position	Support
Tang [44]	Elbow-FE	Position, EMG	Pneumatic actuators	Position	Support
Vanderniepen [45]	Elbow-FE	Not listed	Electric motor	Torque	Support
Wang [46]	Elbow-FE	Position, EMG	Electric motors	Position	Support
Wrist					
Andrikopoulos [47]	Wrist-FE, RU	Position	Pneumatic actuators	Position	Support
Higuma [48]	Wrist-FE, RU	Not listed	Electric motor	Torque	Support
Kazi [49]	Wrist-FE	Acceleration	Piezoelectric actuator	Acceleration	Suppression
Loureiro [50]	Wrist-FE	Position	MRF actuator	Position	Suppression
Taheri [51]	Wrist-FE, RU	Not listed	Pneumatic actuator	Torque	Suppression
Xiao [52]	Wrist-FE, RU	EMG, torque	Electric motor	Torque, position	Support
Hand					
Al-Fahaam [53]	Fingers/Thumb-FE	Force, position, EMG	Pneumatic actuator	Force	Support
Allotta [54–56]	Fingers-FE	Not listed	Electric motor	Velocity	Support
Arata [57]	Fingers-FE	EMG, position	LEA	Force	Support
Aubin [58]	Thumb-FE, AbAd	Position	Electric motor	Position	Support

Hand					
Burton [59]	Fingers-FE Thumb-FE	Position	Pneumatic actuator	Position	Support
Cao [60]	Fingers-FE Thumb-FE	EMG	Electric motor	Velocity	Support
Cempini [61]	Finger-FE	Not listed	Electric motor	Torque	Support
Chiri [62, 63]	Finger-E	Position	Electric motor	Position	Support
Delph [64]	Fingers-FE Thumb-FE	EMG	Electric motor	Position, force	Support
Fok [65]	Fingers-FE	EEG	LEA	Position	Support
Goutam [66]	Finger-F	Force	LEA	Force	Support
Hadi [67]	Fingers-F, Thumb-F	Force	SMA	Force	Support
In [68, 69]	Fingers-FE Thumb-FE	EMG, force, angle	Electric motor	Force	Support
Iqbal [70–72]	Finger-FE Thumb-FE	Position, force	Electric motor	Position	Support
Iqbal [73]	Fingers-FE Thumb-FE	Position, force	Electric motor	Not listed	Support
Kang [74]	Fingers-FE Thumb-FE	Not listed	Electric motor	Force	Support
Matheson [75]	Fingers-FE	Force	Pneumatic actuators	Force	Support
Mulas [76]	Fingers-F Thumb-F	EMG	Electric motor	Velocity	Support
Nycz [77]	Fingers-FE	Position	LEA	Position	Support
Polygerinos [78]	Fingers-FE Thumb-FE	Position	Hydraulic actuator	Force	Support
Saharan [79]	Fingers-FE Thumb-FE	Position	TCA	Position	Support
Sandoval-Gonzalez [80]	Finger-FE Thumb-FE	Position, force	Electric motor	Position, force	Support

Hand					
Tong [81, 82]	Fingers-FE Thumb-FE	EMG	LEA	Velocity	Support
Xing [83–85]	Finger-FE	Position	Pneumatic actuator	Position	Support
Yap [86, 87]	Fingers-FE Thumb-FE	EMG, gesture	Pneumatic actuator	Force	Support
Yun [88]	Fingers-FE Thumb-FE	Not listed	Pneumatic actuator	Force	Support

F–Flexion, E–Extension, P–Pronation, S–Supination, I–Internal rotation, E–External rotation, Ab–Abduction, Ad-Adduction, R-Radial deviation, U-Ulnar deviation, EMG–Electromyography, EEG–Electroencephalography, MRF–Magnetorheological Fluid, SMA–Smart material actuator, TCA–Twisted coiled actuator, LEA–Linear electric actuator.

Overall, there are a few trends worth noting about these devices. First, the bulk of these devices are wearable but not fully portable. In many cases, either a portion or all of the actuation system, power supply, or communication and control cables are tethering these devices to a location. Second, the majority of the devices in this review focus on the motion of a single arm segment (88% of devices) with opening and/or closing of the hand and fingers being the largest focus (50% of devices). This is likely due to the complexity of human motion and related biological signals, which increase in complexity when neuromuscular disorders have occurred to the user. However, improving the movement of even one segment of the upper limb can translate into better quality of life for those suffering from these disorders. The single-segment approach to the design of these devices has a higher likelihood of finding solutions and, therefore, getting to market and creating social benefits. Lastly, it should be noted that few clinical studies have been performed on any of the reviewed devices. The devices are still in development and require vigorous testing to ensure human safety, which makes it difficult to attain ethics approval for studies with human volunteers. Therefore, the research areas surrounding wearable assistive technologies have many questions still to be answered.

Although wearable assistive devices have experienced many successes, further research and technological advancements must be made before the full benefits can be exploited. From the above examples, it is clear that wearability has become an important characteristics in the design of these devices. However, aspects of the assistability still need to be explored and studied. Sensing

and actuation technologies will continue to improve but these devices will not be adopted if their computer systems are unable to keep up and perform accurate and safe behaviours. Before moving onto the computational aspects attributable to these devices, a list of general guidelines for the reader is provided to enhance the identification of design specification for wearable assistive devices, as follows:

- the structures of the system must not prevent the user from completing natural motions,
- the weight must be minimized in order to prevent excess strain on already weak tissues,
- the volume must be minimized to reduce the change in dynamic parameters of human motor control,
- the power consumption must be minimized due to size constraints on power sources,
- wireless communication with control and data storage devices is preferred, in order to increase wearability,
- proper safety features must be in place to ensure no harm comes to the user or device,
- easy to use, replaceable, and comfortable human–machine interfaces are essential,
- aesthetically pleasing components should be incorporated to promote user adoption,
- interactive control systems are preferred to encourage user participation in motion tasks,
- and sensitive biological data must be handled appropriately to ensure that the confidentiality of the user's data is upheld.

2.2.3 Computer Systems Review

Modern digital computer systems are the foundation for implementing a variety of aspects of sensing, actuation, and control of wearable assistive devices. The complexity of these devices creates a huge demand for processing, analysis, and storage of information, which is being supplied by computer systems and electronic components. Although, computer systems facilitate the interaction of all of the mechatronic system components, some authors have neglected to include important

details relating to the computer systems used in the design, development, and testing of their devices. From the reviewed devices, no description of the computer architectures or systems were given for 21% of the devices [33, 40, 45, 49, 50, 53, 59, 61, 66, 73, 75]. Of the devices that provided computer system information, the general computer hardware and software specifications have been extracted.

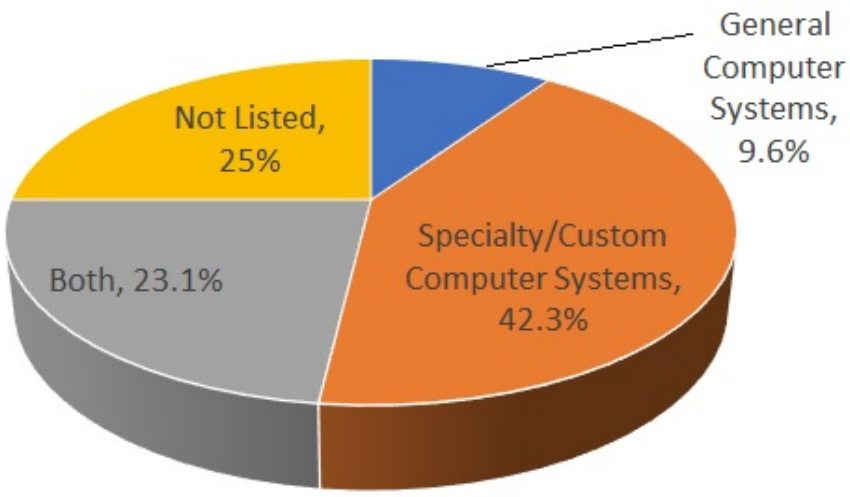
2.2.3.1 Computer Hardware

Computer system hardware is described in 79% of the reviewed devices, although only briefly by some. The descriptions range from as vague as "a standard data acquisition board" to listing hardware components, their connections, and their communication protocols. General purpose computer systems, such as a laptop or personal computer, were listed as an integral component of the computer architecture in 33% of the devices [25, 29, 30, 34, 37–39, 41, 46, 47, 52, 63, 65, 70, 71, 76, 77, 79, 83, 84], while 31% of devices listed the use of off-the-shelf microcontroller boards [27,48,53,54,58,60,64,67,74,76–78,81,83,87,88]. Where model numbers are listed, basic processor information can be looked up, but only 12% of the devices listed basic eprocessor and memory information [25, 31, 35, 41, 47, 76]. To complement these computer systems, 27% devices are using other off-the-shelf electronics, such as data acquisition boards and sensing platforms [25,30,32,37, 39, 46, 47, 51, 52, 63, 68, 70–72, 79, 83–85]. In order to increase the portability of these devices, the computer system must be embedded within the devices or stored on the user's body. This means that computer hardware analysis is important to ensure the computational requirements can be factored in when designing these devices. A breakdown of the type of computer hardware systems is presented in Fig. 2.4.

2.2.3.2 Computer Software

In terms of computer system software, a similar trend can be seen towards researchers using off-the-shelf software systems. The review reveals that 50% of the studies reported the use of off-the-shelf software with MATLAB (23% of devices) [25, 29, 32, 36, 39, 40, 44, 57, 59, 63, 66, 76], including the Simulink tool kit [39, 74, 76], and LabVIEW (15% of devices) [34, 38, 39, 47, 52, 62, 65, 68] being the

Figure 2.4: Distribution of computer hardware systems of wearable assistive devices. General computer systems include laptops, desktops and other personal computers, while the specialty/custom computer systems include microcontrollers, custom circuits, specialty computer systems, sensing electronics and data acquisition boards.



most popular. Other software systems mentioned were OpenSim [31], SolidWorks [25,29,37,85,86], SimMechanics [29, 54–56], ControlDesk [30], Emotiv Cognitiv Suite [34], OpenSignals [25, 37], Presentation [36], Datalog [35], DAFUL [48,57], FEMAP [48], MPLAB [76], BCI2000 Framework [41, 65] and XVR [80]. Although authors are listing their software systems, only 6% of devices describe the operating system, in which these software systems are executed [25,35,47]. The type of operating system that is used can have major effects on the execution of control software and, therefore, should be reported. Programming languages, namely C and C++, were used in 10% of the devices as a primary development tool for software [25,27,32,37,76]. Very few descriptions of information regarding the software structure, complexity or timing can be found in the literature. Some of the software structure can be inferred from the control system descriptions, but it may not be possible to derive the entire software architecture from the control architecture, as they need not be mapped one-to-one.

2.2.3.3 Computer System Trends and Limitations

One of the successes seen with the computer systems is in power supplies. Both lithium-ion [74,77] and lithium-polymer [31, 78, 88] batteries are used to power the motors and electronics of these

devices, making it possible to increase portability. These power supplies remove one of the aspects that tether these devices to specific locations. However, much work still remains to reduce power consumption of the electronics, as well as to decrease the weight of the battery, while extending the amount of power provided. Power supplies are a crucial research area that supports the vision of these devices being used in a continuous all-day manner.

The literature shows that many of the devices used desktop or laptop computer systems, off-the-shelf microcontrollers and other self-contained electronic systems for sensing or actuation, supporting the idea that developers are focusing on proof-of-concept development. This development strategy reduces the amount of time and resources needed to create a functioning prototype. By using off-the-shelf software systems, mechatronics engineers are able to prototype devices more rapidly.

The popularity of the Simulink tool for MATLAB, and LabVIEW emphasizes the view that engineers in this field may be more comfortable using visual-based control system development tools. It could be fruitful to increase the education of mechatronics engineers with software engineering principles or add software engineers to development teams. By making custom software, developers have more freedom of implementation and the opportunity to reduce the computational demand. Existing embedded computer systems are likely to be unable to meet the processing requirements using software systems developed for desktop computer systems. These software systems will not typically operate on these embedded systems. In case where it is possible to execute them on embedded systems, the embedded systems do not have enough resources to complete the tasks required of the wearable assistive devices in similar time periods as their desktop counterparts. The lack of development on computer systems for these devices suggests that development teams may not have the expertise in all aspects of these wearable assistive devices. Reconfigurable computer hardware, such as field-programmable gate arrays, may be a potential solution, but have yet to be implemented and reported on by this research community.

Improvements to this research area can be made by development teams weighing in on what they consider important aspects of the software and hardware components that are required to replicate or evaluate results. The vagueness and lack of computer hardware and software details make it difficult to understand and reproduce the designs and experiments with wearable assistive

devices. Even basic details of the computer systems, such as software versions, libraries, operating system software, model numbers of physical components, and processing resources, would help to alleviate this issue. One of the most important aspects for researchers and developers alike is to standardize information about the computer systems of these devices. A lack of basic computational information hinders the evolution of this field. Many of the computer systems included in this review operate the device as expected because the computational environment has more resources than required. However, a limitation will be realized as the complexity of the computation grows, while the requirements for power and space on the devices or body limit the resources available in such embedded computer systems.

2.3 Computational Foundations of Wearable Assistive Devices

Computer systems are the enabling technology for wearable assistive devices. Without computer systems, facilitating the complex human–machine interactions would not be possible. The focus of the remaining chapter will be on aspects of computer systems that support these interactions. As will be discussed, the complexity of the computer systems alone must be decomposed in detail to provide a full understanding of their functionality. These computer systems can be decomposed into the five fundamental knowledge areas as shown in Fig. 2.5.

Human–machine interfacing encompasses how the computer system facilitates interaction between the human and the physical components of the devices. Processing digital and analog signals, generated from both human and machine, is crucial to ensuring that the data are interpretable and useful. Communication aspects define how the channels, between components and between systems, provide flow of data. Modelling relationships, both internal and external to the system, provide the ability to make accurate estimates of events and inform decision-making algorithms. Finally, controlling the device behaviour enables useful and safe interactions for all systems involved. Each of these five knowledge areas of computation constitute large sets of knowledge that are dependent on and influence aspects of these wearable mechatronic devices. The following sections will examine these five foundational areas in order to highlight the relationships, the scientific fundamentals, and the existing solutions contained within their scope.

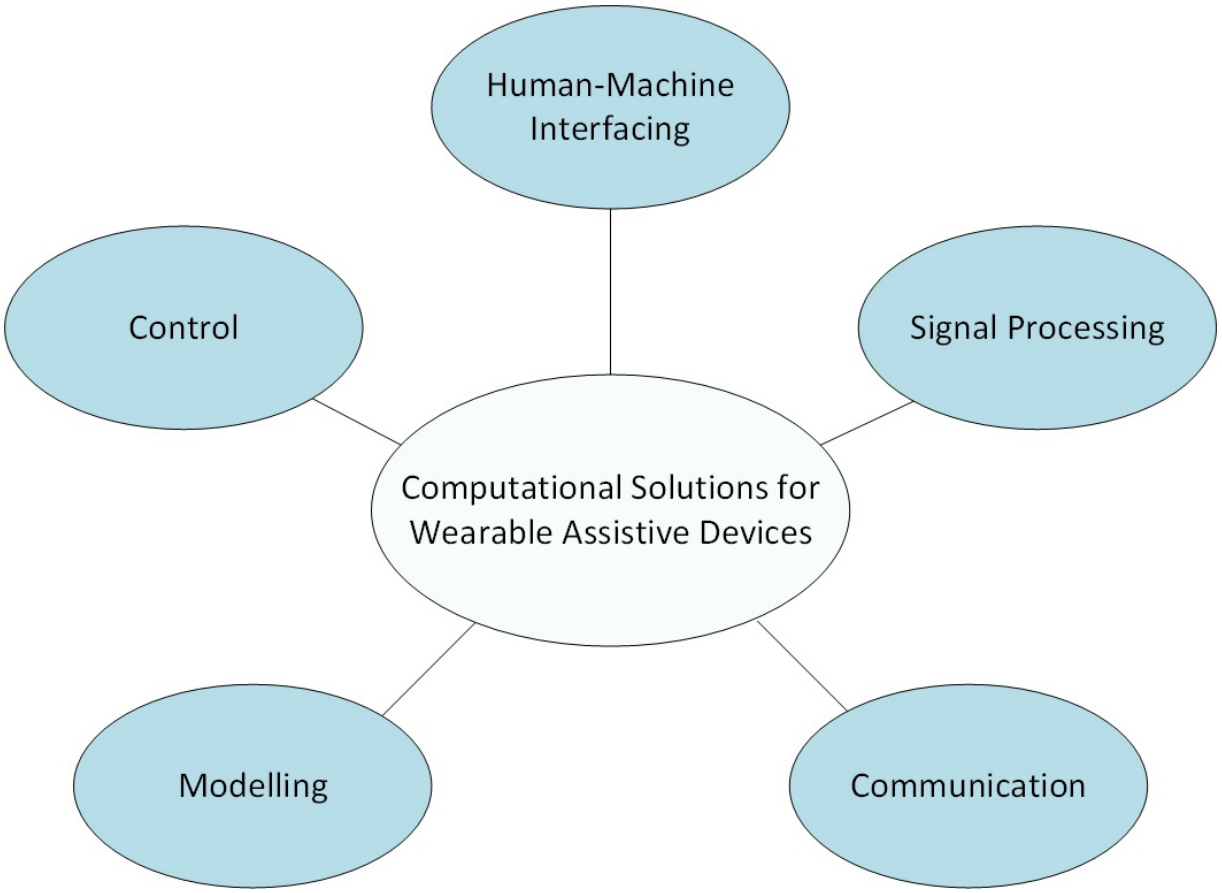


Figure 2.5: The computational systems of wearable assistive devices consist of five key knowledge areas.

2.4 Human–Machine Interfacing

The first fundamental knowledge area is human–machine interfacing. Computer systems are physical devices capable of perceiving physical interactions or phenomena through changes in digital signals. In this sense, humans interact with wearable assistive devices in both the physical and digital realms. Wearable assistive devices provide physical interaction points in two general forms: sensors and actuators. Sensors located on the human and the device allow each of these system to perceive each other's action in order to aid in the formulation of decisions and new actions. Actuators, such as electric motors and muscle tissue, allow both the device and the human to produce changes in the physical world, such as motion. These interfaces are the initial input and final output of the wearable assistive device to provide the appropriate amount of assistive force or

torque about a particular human joint. Many forms of sensing and actuation technology have been developed, each of which influences the requirements on the control and computational systems that interface with them.

2.4.1 Sensors

Changes in energy caused by physical phenomenon, such as human motion, can be detected by sensors. For computers to detect these changes, sensors must transform the changes in energy into changes in digital voltages. Whether the sensor is detecting a change in skin temperature, that a button was pressed by the user, or changes in an electric motor's position, the computer system inteprets this through a change in voltage. Computer programs require models to make use of these voltage changes but, if programmed appropriately, can develop a digital representation of physical phenomena. Therefore, using sensors, the computer system can create a digital representation of both the physical and digital environments in which it is interacting.

The type of sensors used for control inputs vary depending on many factors, including environmental conditions, actuation technology, abilities of the users, and the intended application for the device. At the current stage of wearable assistive device development, sensor data come from either the user or the device. Quantities sensed from the user form the main set of control system inputs and can be referred to as control interfaces. Lobo-Prat *et al.* have done extensive research in control interfacing, including a review of non-invasive control interfaces that provide motion intention data and comparisons of EMG, force, and joystick control interfaces for arm support systems [89–91]. Device sensors are used to determine motion outputs of the user, such as the force they exert onto the device, or properties of the device that are important to regulating its behaviour. Overall, a variety of sensors are required to facilitate human–machine interactions. Fig. 2.6 shows the sensors used to facilitate elbow motion tracking [37].

Currently, many non-invasive control interfaces have been explored as primary control interfaces for wearable assistive devices. In order to allow the user to control the device, their physiological signals are required, as they provide motion intention information. These control interfaces are supported by three general types of sensed signals (bioelectrical, position, and force signals) that are required to collect input data from the control system. These signals represent inputs to or

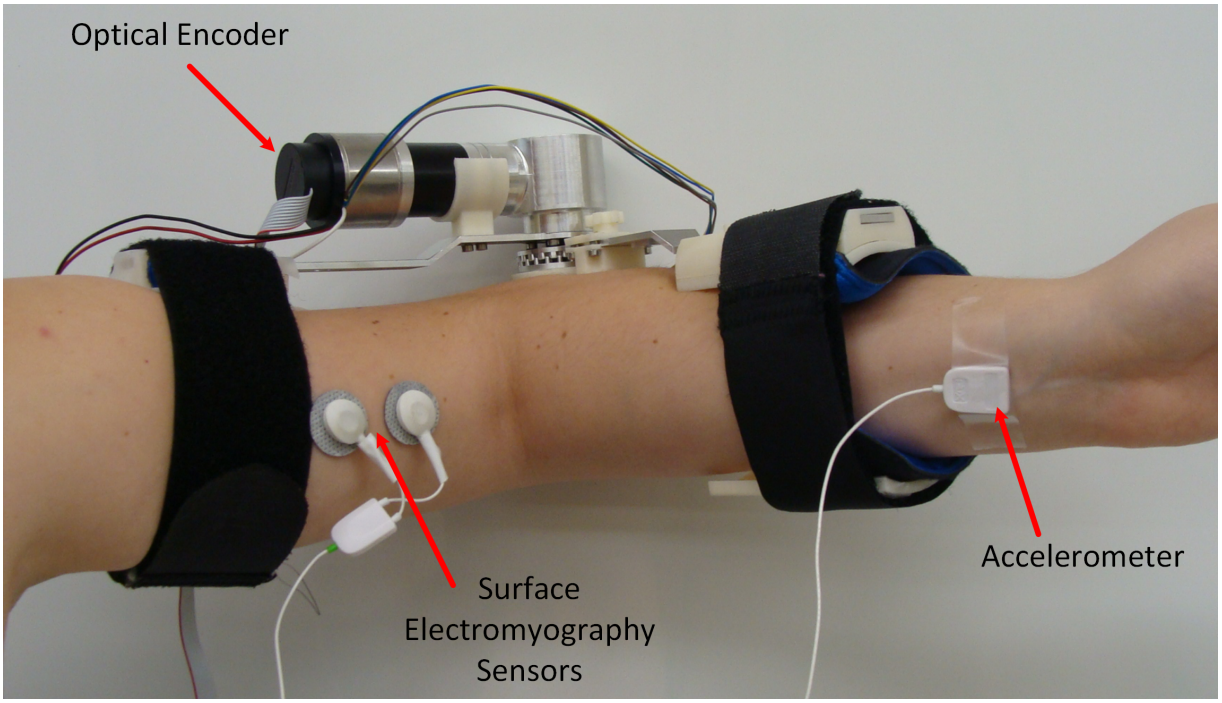


Figure 2.6: Overview of sensors that were incorporated in to the WearME elbow brace 1.0. [37]

outputs of the musculoskeletal system. The most common type of control interfaces use surface EMG (sEMG) sensors [89]. sEMG sensors provide the representations of the bioelectrical impulses of the nervous system used to control force production of muscles and have been used extensively for both wearable [25, 27, 35–37, 44, 46, 52, 60, 64, 68, 69, 76, 81, 82] and stationary assistive devices [92–95]. Popular interfaces based on position data include inertial measurement units (IMUs) [25, 29, 31, 32, 38, 41] and cameras [86, 87, 96]. These types of sensors operate on motion of the body segments. However, motions of other parts of the body have been used as control inputs, such as the eyes and tongue [89]. Control interfaces employing force-based sensors operate on an action–reaction paradigm, in which the user produces a desired force and the device must react accordingly. This type of sensing has also been used by many wearable [36, 53, 62, 63, 66, 70–73, 75, 80, 83–85] and stationary mechatronic systems [24, 93, 94, 97–100].

Sensing the user's motion intention is only part of the sensing architecture, the other being sensing specific properties of the device. The most important data to collect about the device are related to its own motion. More specifically, position, velocity, acceleration, force, and torque are the quantities that need to be sensed, as these allow for comparisons to be made between the

user's desired motion and the resultant motion of the device. Based on the devices in Table 2.1, the most commonly sensed motion parameter for wearable assistive devices is position. In some cases, it makes more sense to include sensors that collect information related to the motion of the device, but are not a direct measurement of the motion, such as pressure [29, 39, 42]. Sensors responsible for characterizing actuation are usually used in control feedback loops (see Section 2.8.1.3) as a means to correct actuation errors. This makes them an integral part of facilitating human–machine interactions. Table 2.3 provides a brief comparison of different sensor types used in stationary exoskeleton-type assistive devices.

Table 2.2: The number of wearable assistive devices employing each type of motion parameter generated through device actuation.

Number of Devices	**Actuation Parameters**				
	Position	**Velocity**	**Acceleration**	**Force**	**Torque**
27	2	1	9	2	13

2.4.2 Actuators

Wearable assistive devices require actuators to provide some forces through a mechanical interaction point between the device and the user's skin. Computer systems facilitate this interaction through the inverse energy transformation processes as those used with sensors. A voltage, commanded from a computer system output is transmitted to the actuation system, interpreted, and used to produced a change in energy to induce motion, such as a change in air pressure in pneumatic artificial muscles [29, 33, 39, 44, 47, 53, 59, 75, 83]. In most actuation systems, there are more than one transformation that must occur to induce motion, nevertheless, the initial signaling for the motion begins with the computer system. Based on this transformative energy principle, computer systems are able to cause events to occur in the physical environment using actuation systems. Actuation systems provide the means for assistive devices to produce meaningful human–machine interactions.

An actuation system is a group of components that produce motion between the articulating surfaces of these devices. In the musculoskeletal system, the muscles are the active producers

Table 2.3: A comparison of electrical-based, position-based, and force-based sensors used for control of stationary exoskeleton-type assistive devices. Sensor types are those listed by the authors.

Author [Reference]	**Bioelectrical-Based Sensors**	**Position-based Sensors**	**Force-based Sensors**
Culmer [97]	NU	Rotary position sensor String-potentiometer	NU
Frisoli [101]	NU	Potentiometer Optical encoder	NU
Gopura [93]	sEMG	Potentiometer	Strain gauge
Gupta [102]	NU	Encoder	NU
Kiguchi [94]	sEMG	Potentiometer	3 DOF pico force sensor
Klein [103]	NU	Potentiometer	Pressure sensor
Li [95]	sEMG	Optical Encoder	NU
Martinez [104]	NU	Encoder Linear position sensor	Pressure sensor
Moubarak [100]	NU	Magnetic encoder	NL (Force)
Nef [105]	NU	Optical encoder Potentiometer	NU
Park [92]	NL (EMG)	NL (Position)	6 DOF force/torque sensor
Perry [106]	NU	Potentiometer Encoder	NU
Sanchez [107]	NU	Linear potentiometer Rotary position sensors MEMS accelerometer	Pressure sensor Air flow sensor
Tsagarakis [98]	NU	NL(Position)	Strain gauges NL (Pressure)
Vertechy [99]	NU	Optical encoder	6 DOF force/torque sensor Strain gauge 3 DOF force sensor
Vitiello [24]	NU	Optical encoder	Piezoresistive strain gauges

NU - Sensing modality was not used, NL (quantity) - the quantity was collected but sensor information was not listed, DOF - degree-of-freedom.

of motion. The main functionality of the assistive devices is to orchestrate the behaviour of the actuation systems to meet two general goals: motion tracking and motion assistance. A major focus of the actuation system research is on developing actuation systems that produce higher forces at lower weights and volumes, as the carrying capacity of the user is limited, especially in the presence of a musculoskeletal disorder. Due to the complexities of human–machine interaction and current technologies, there does not exist an actuation system that is equivalent to or improves upon all properties of human muscle. Furthermore, the asisstive forces provided to patients during traditional assistive therapies have yet to be quantified. This leaves control system researchers and developers focused on evaluating existing actuation systems to track human motion or pre-defined trajectories.

Control of the actuation system varies based on the type of actuator and the type of transmission system that are used. Considering the devices in both Table 2.1 and Table 2.4, electric motors are the most popular choice to produce motion of these devices [25, 27, 30–32, 34–37, 41, 43, 45, 46, 48, 52, 54, 57, 58, 60–66, 69–71, 73, 74, 76, 77, 80, 81, 92–95, 99–102, 104–106]. This is likely due to the large variation in electric motor specifications that are available to developers, making designs involving them easy to adapt to different assistive scenarios. Apart from electric motor actuators, pneumatic actuators [29, 33, 39, 44, 47, 51, 53, 59, 75, 83, 86–88, 97, 98, 103, 104, 107] and hydraulic actuators [24, 42, 78] are the next most common actuator types, as they are typically lighter than electric motors. However, pneumatic and hydraulic actuators suffer from the need of infrastructure, such as pumps, valves, and hoses, to operate and are harder to control than electric motors. Researchers have also explored electomagnetic friction brakes [38], magnetorheological fluid actuators [50], shape memory alloy actuators [40, 67] and twisted coiled actuators [79] as possible actuation solutions.

Many of these actuators cannot directly drive the joints of the device due to their power, weight, or volume limitations. As a result, transmission systems are used to either scale the power output or transmit motion to the joint from a location away from the joint. Wearable assistive devices, whose actuators are not coupled directly to the joint, have used gears [25, 32, 35–38, 43, 80], cables [58, 75, 76, 79, 83], linkages [65, 81], or a combination of these transmission methods [34, 45, 46, 52, 54, 59–61, 63, 64, 66, 68, 70, 73, 74]. The design of the transmission system affects the actuator

requirements as there are power losses due to inefficient transmission. Both power losses in the actuator and within the transmission must be accounted for by the control system. A comparison of the actuation system parameters of stationary exoskeleton-type assistive devices is presented in Table 2.4.

2.4.2.1 Increasing Actuator Safety

Increasing the safety of assistive devices is accomplished mainly through the actuation and control systems. In this section, the discussion will surround two aspects of actuation systems that can increase safety, while aspects of safety concerning the control system are discussed in Section 2.8.2. Safe human–machine interaction can be increased through introducing backdrivability and compliance into the actuation systems. Backdrivability is a measure of the degree to which the actuation system can be moved due only to the user's force [108]. From a safety perspective, backdrivability provides a way to ensure that, even in the event of power failure, a user can move the device with minimal effort. Therefore, even a person with limited motor function can manipulate themselves and the device into a position in which they can disconnect from the system. Backdrivability has been reported as a design requirement of many assistive devices [102, 105, 107, 109–112] and requires friction and gearing to be minimized in order for it to be achieved.

Compliance of the actuation system is the level to which the system will resist forces that are applied to it. The human body exhibits compliance to external forces, making it desirable to mimic this characteristic within the device. One method for introducing physical compliance into the actuation systems is through a type of actuator known as series elastic actuators (SEAs). The fundamental concept of SEAs is to transfer power from the source to the destination through an elastic element (see [112–114] for further information). The common elastic element used in SEAs is a spring. Using Hooke's Law, a spring can be introduced into the power transmission train and the stiffness can be modulated simply by varying the spring displacement (Fig. 2.7). Compliance can be achieved using passive elastic elements [114] or by controlling the spring's displacement by a separate actuation unit [46]. However, two major problems plague SEAs: their control bandwidth and the achievable stiffness. The control bandwidth of the actuator is relatively

Table 2.4: Actuator parameter comparison of stationary exoskeleton-type assistive devices.

Author [Reference]	**Actuator Type**	**Maximum Output Torque (Nm)***
Culmer [97]	Pneumatic cylinder actuator	15 - shoulder flexion 6 - shoulder abduction 5.5 - elbow flexion
Frisoli [101]	Electric motor	2
Gopura [93]	Electric motor	98 - shoulder flexion 28.4 - shoulder adduction 28.4 - elbow flexion 8.1 - forearm pronation 1.38 - wrist flexion 1.38 - wrist abduction
Gupta [102]	Electric motor	5.46 - elbow flexion 5.08 - forearm pronation 0.4 - wrist flexion 0.4 - wrist abduction
Kiguchi [94]	Electric motor	NS
Klein [103]	Pneumatic cylinder actuator	108
Li [95]	Electric motor	NS
Martinez [104]	Electric motor Pneumatic artificial muscle	NS
Moubarak [100]	Electric motor	52.5 - shoulder flexion 63 - shoulder abduction 17.1 - shoulder internal rotation 13.5 - elbow flexion
Nef [105]	Electric motor	37.76 - shoulder flexion 38.5 - shoulder internal rotation 32.0 - elbow flexion
Park [92]	Electric motor	200 - shoulder flexion 85.3 - shoulder abduction 20 - shoulder internal rotation 32 - elbow flexion 11 - forearm pronation 9 - wrist flexion
Perry [106]	Electric motor	6.2 - shoulder 6.2 elbow 1 - wrist
Sanchez [107]	Pneumatic cyclinder actuator	NS

Tsagarakis [98]	Pneumatic artificial muscle	30 - shoulder flexion 27 - shoulder abduction 6 - shoulder internal rotation 6 - elbow flexion 5 - forearm pronation 4 - wrist flexion 4 - wrist abduction
Vertechy [99]	Electric motor	151 - shoulder flexion 151 - shoulder abduction 80 - shoulder internal rotation 151 - elbow flexion
Vitiello [24]	Linear hydraulic actuator	15 - elbow flexion

* - Where joint torques were given, the parameters are associated with body segment motions. Otherwise, the torque is assumed to be the output torque of the motor.

NS - quantity was not stated by the authors.

low and is limited due to the elastic element. Although SEAs offer the ability to vary the stiffness, the limits are again set by the elastic element. Changing the stiffness range would require changing the springs. SEAs offer an interesting approach to increasing compliance of assistive devices but require further research to reduce the limitations, while meeting other actuation specifications, such as power output and weight.

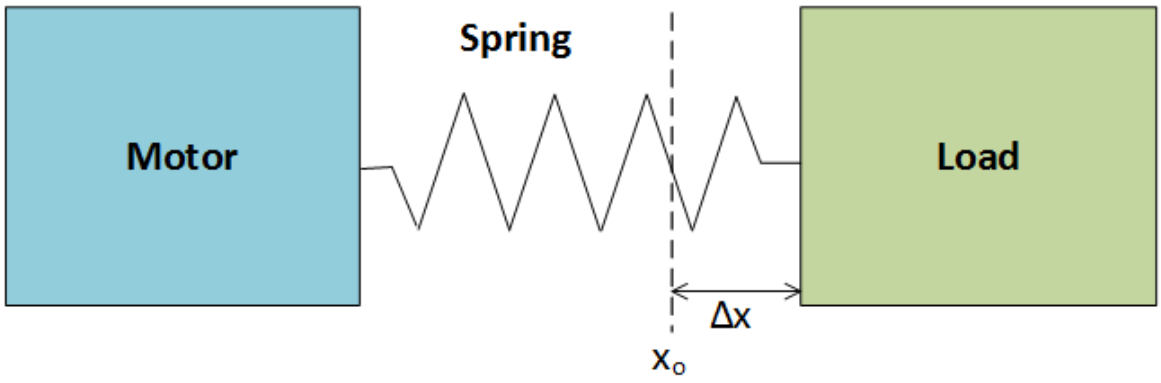


Figure 2.7: The general model of a series elastic actuator, using a motor as the actuator. x_0 is the initial position and Δx is the change in position.

2.5 Signal Processing for Biological and Mechatronic Systems

The second fundamental knowledge area is signal processing. In this section, the fundamentals of digital signal processing will be discussed, as these concepts help to form important parts of the control systems developed for this thesis (Chapter 4). In order to control the behaviour of a wearable assistive device, the computer system must be able to manipulate and parse signals originating from the physical environment. The human and device both produce a variety of signals related to their current state as shown in Fig. 2.8. From a non-invasive biological perspective, skin temperature, electrical activity from the muscles and brain, segment motion and force, sound, heart rate and oxygen consumption can all be measured from the human body. Sensors embedded within wearable assistive devices provide measured quantities, such as motion, orientation, applied forces, deformation, power consumption, communication and temperature. Each of these signals can provide important information if transformed into a signaling pattern that can be interpreted by the computer system. The processes and methods by which this transformation occurs is known as *signal processing*.

In mechatronic devices, the theories and discussions about signal processing generally concern only the inputs to the system. However, it is useful to note that producing appropriate output signals also requires processing. Fig. 2.9 shows an example of an abstract signal processing pipeline that can be found in mechatronic systems. More specifically, the signal processing domain emphasizes the processing that occurs within the digital realm, known as digital signal processing. On the input side, this encompasses the transformations between the electrical signals generated by the sensor and the digital signals received by the input pins of the processing unit. Considering the output signal processing, this typically includes the processing required between the processing unit and the inputs of another component within the system, such as an actuation system. For wearable assistive devices, signal processing can vary in scope from the transformative processes within the sensors themselves to all transformations required for the computer processors to execute tasks. However, to stay within the established realm of computer system signal processing, discussions will be constrained to those involving only the electrical channels connecting mechatronic system components and processing units.

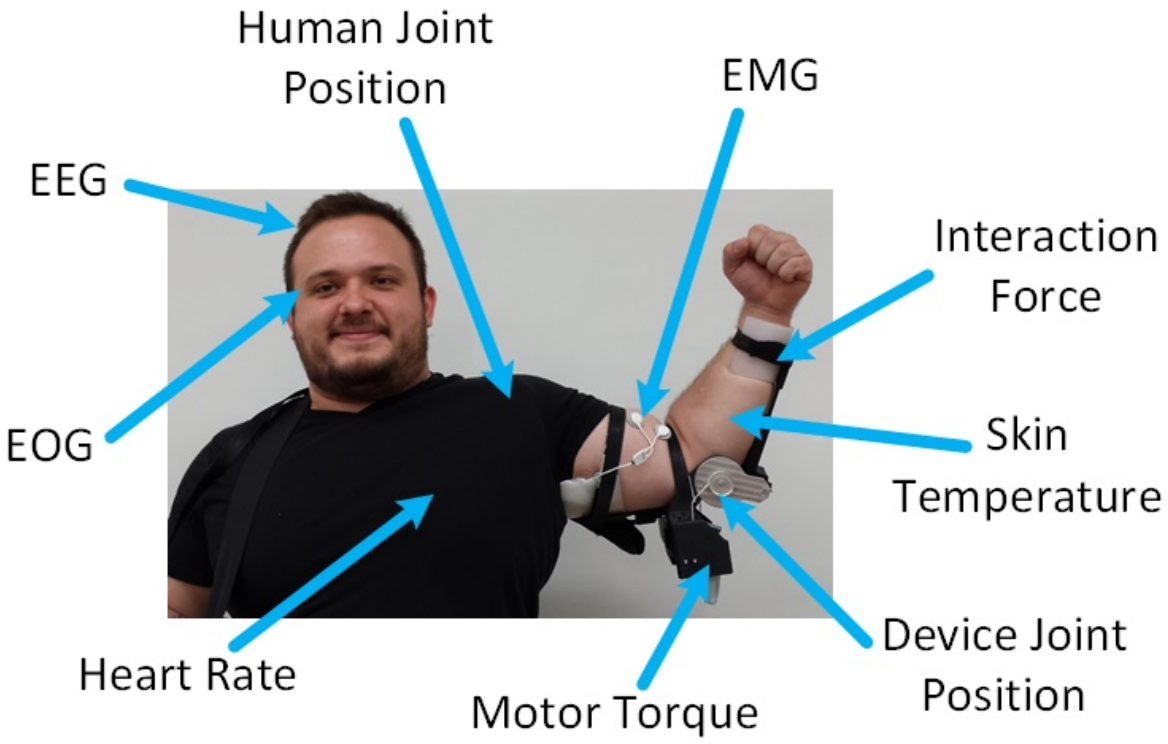


Figure 2.8: Biological and mechatronic system signals that can be used to facilitate assistive motion tasks, including EMG, electrooculography (EOG), and electroencephalography (EEG) signals.

Generally, signal processing is composed of discretization, scaling, rate conversion, and filtering. Discretization is the process of taking samples of a continuous signal at a specified rate in order to produce a discrete representation of that signal. Since no processing systems, digital or biological, process information in continuous form, discrete forms of signals are required to produce some meaningful behaviour. It is common that the range of interpretation of the processing unit is different from the range of values at which the signal was captured. In this case, scaling is required to map one range of values to another. Another issue arises in interpretation when two signals are sampled at different rates. Rate conversion refers to the methods by which number of samples representing a signal in a given time period are either increased or decreased. Filtering methods are used to remove the undesired portions of a signal. Typically, filters are used to remove electrical noise or motion artifacts from the electrical signals. These four basic processes encompass most of the signal processing required to use signals for control a wearable assistive devices. The following

sections will describe these four basic processes and how they are used to process elbow motion data for motion estimation purposes.

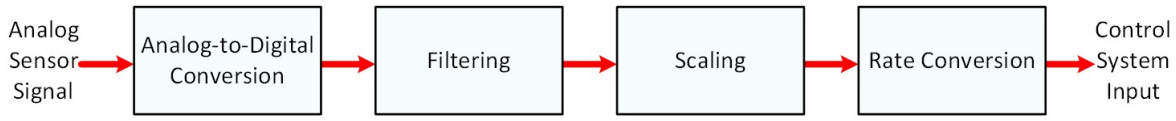


Figure 2.9: An example of an abstract signal processing pipeline used in wearable mechatronic systems.

2.5.1 Discretization

The first basic process of signal processing is the discretization of continuous signals. Discretization is the process of representing a continuous signal with a set of discrete values. There are two key components to determing whether a digital signal is a discrete or continuous representation of an analog signal. First, a discrete representation is one whose smallest meaningful difference is larger than that of the continuous signal. Continuous signals operate within real or complex number systems, in which the smallest meaningful change is essentially infinitely small. In contrast, discrete representations have a bounded smallest meaningful difference. Second, a discrete representation is generated by sampling a continuous signal at a frequency that is less than twice its largest meaningful frequency component. The Nyquist rate is a sampling rate equal to twice the largest frequency component of interest in a signal [115]. Sampling this signal at a rate greater than or equal to the Nyquist rate guarantees that all of the desired information will be captured.

Analog sensors convert continuous phenomenon occurring in the environment into bounded continuous electrical representations. Analog-to-digital converters (ADCs) are used to change these bounded continuous signals into bounded discrete signals. Digital electrical sensors are essentially a combination of an analog sensor and an ADC. This process of converting analog electrical signals into digital electrical signals is called *digitization*. Digitization is a specific form of discretization to convert continuous electrical signals into discrete electrical signals. Since computer system are a bounded binary number system, digitization, and, therefore, discretization are required by the computer system to interpret phenomena.

Human biological signals contain information occurring at a variety of frequencies. Voluntary

elbow motion is typically captured within the 0–3 Hz frequency spectrum of the signal [116]. Using the Nyquist rate, one could determine that sampling this signal at a frequency of 6 Hz or higher would capture this signal such that the sampled signal could be considered to contain all meaningful data. Commonly, the models that use EMG signals to control device behaviour consider the main frequency content of the signal around the range of 10–500 Hz [117, 118]. If the only information of interest lies in within the 10–500 Hz spectrum, then a 1000 Hz sampling frequency can be considered a continuous signal with respect to the desired frequency content, but still a discrete representation of the true EMG signal, as it has components operating at frequencies higher than 1000 Hz. It is important to note that even a digital system that can sample signals at or above the Nyquist rate is still a discrete representation of the continuous signal, since digitization is still required to interpret the signal.

The discretization of signals influences the accuracy of the control system to determine device behaviour. A perfect model of the system may still not produce accurate results if the information it relies on was not fully encompassed when sampling the signal. It is important to know the frequency content of each signal type in order to analyze the trade-off between sampling at higher frequencies and spending that time executing other computational tasks. Furthermore, using discretization systems, such as digital sensors, with the smallest possible resolution will help to detect more meaningful changes and improve the accuracy of the system. A review of signals that are commonly sampled to control wearable assistive devices was shown in Table 2.1. Since the goal of these devices is to produce motion, it is most helpful to measure aspects of the human body that might help to predict the desired motion. EMG signals are the most cited for this purpose. Recently, electroencephalography (EEG) signals have been introduced as a supplement to EMG-based motion estimation [119, 120]. These signal types represent the inputs to the computer system, while the position, velocity and acceleration are outputs of the human and mechatronic systems. Regardless of the signal types or combinations, the goal is to acheive the least amount of discretization possible. This amount can

2.5.2 Scaling

The second basic process of signal processing is to scale the signals between two systems or components. Scaling is a general method for converting the output of one system into a range of values that can be interpreted as input to another system. With bioelectrical signals, the meaningful changes are generally measured in the range of microvolts or millivolts [117]. These signals require amplification to be interpreted appropriately by digital systems. Amplification is an example of scaling, as the amplitude of the signal coming into the amplifier has been increased at the output in order to scale the signal to the input requirements of the next signal processing component. Amplification is used by many devices where voltage changes are small, such as those captured through EMG [64, 117, 121]. For example, Ruiz *et al.* amplify the collected EMG signal with a 1000 gain value as the first step in their signal processing method [96]. Scaling, by means of amplification, not only makes signals interpretable by other signal processing systems, but also makes signals easier for humans to interpret.

Control systems can perform more predictably if inputs are bounded to some known range. One method to scale digitized biological signals is to use normalization. Normalization is the scaling of an input signal into a bounded range using the maximum and minimum values of a specific signal. For example, EMG signals recorded from the muscle can be bounded between the signal collected at the resting state of the muscle and during maximal voluntary contraction (MVC) [39, 122]:

$$emg_{norm}(t) = \frac{emg(t) - emg_{min}}{emg_{max} - emg_{min}} \tag{2.1}$$

where emg_{norm} is the normalized EMG signal, emg is the EMG signal to be normalized, emg_{min} and emg_{max} are the minimum and maximum EMG values of the MVC, respectively, and t is time. Using Eq. (2.1), the collected EMG signal is scaled to a range between 0 (minimal muscle activation) and 1 (maximal contraction). One advantage of normalization, and scaling in general, is that control errors can be reduced if the signal is guaranteed within a specified range of values. This makes it possible to reuse and mix various signal processing and control components without having to redesigning the entire control architecture each time a new components is added with a different input and output ranges. Due to the flexibility of microprocessors, this advantage is

typically only experienced between the electronic components and the microprocessor. When two electronic components, with non-configurable input and output ranges, are connected in series, redesign of the electrical circuitry must still occur. Scaling is an important part of the control architecture design and must be considered to account for all scaling scenarios.

2.5.3 Rate Conversion

The third basic process of signal processing is to convert signals from one sampling rate to another. Sampling a signal is conducted at a specified rate, known as the *sampling rate*. The sampling rate is defined as the number of samples taken of a signal in a given time period. As discussed in Section 2.5.1, different signals operate at different frequencies, which leads to variation in sampling rates. A problem occurs when a system requires signals to be represented at a specific sampling rate at which they were not sampled. *Rate conversion* is the process of representing a signal with a different number of samples than are in its current representation and is the general solution to this problem. For example, a control system may require both EMG signals from muscles controlling a joint and the position of the joint itself as input, such as for elbow motion estimation [25,96,122–129]. EMG sampling rates, for control purposes, are 500 Hz or above [25,35,117,118,122,130], while voluntary elbow motion requires a minimum sampling frequency of 6 Hz to capture all of the data [116]. If the motion estimation model requires both signals to be at the same sampling rate, then the signals need to be captured at the same sampling rate or one of them has to undergo rate conversion to meet this criteria. Using one sensing systems to sample both signals, means that this criteria could be fulfilled by sampling them at the same rate. However, this comes at the expense of computation to collect 1000 samples per second from the position signal when only 6 per second are needed to fully represent the signal. Since computational resources are limited in wearable assistive device, it is a good practice to collect only what is necessary, unless the rate conversion takes more resources to operate than sampling at a higher frequency. For wearable assistive devices, it is common that more than one sampling system is used, where each one is customized to a particular signal and operates at a different sampling rate. As a result, rate conversion will be necessary.

Representing a sampled signal with a different number of samples is a difficult but common task. The difficulty of this task is correlated with the dynamic components of the signal. Mathe-

matical analysis software systems have developed standard functions, such as MATLAB's *interp* (MathWorks, USA), to enable users to complete rate conversions of their data without understanding of the algorithms. Standard algorithms and methods can be found in the literature, but are beyond the scope of this thesis.

2.5.4 Filtering

The fourth basic process of signal processing is filtering. Filtering is a signal processing method involving the removal of portions of a signal to shape it into a desired signal. Often, filtering is used to make a signal more interpretable or ensure it only contains characteristics accounted for by the systems that will use this signal. With regards to wearable devices, filtering is used to remove electrical noise, motion artifacts, and other unwanted data. Filtering creates a signal which contains only the data that are useful to the system.

The ability for a signal to be interpreted plays significantly into the level of filtering that is conducted on a given signal. For example, an assistive device, which requires joint position and EMG signals in order to interact with the user, employs different filtering options for each of these signals. Since voluntary human motion occurs at low frequencies, low-pass filtering of joint position with cut-off frequencies ranging from 1–5 Hz are commonly used [122, 124, 125, 130, 131]. EMG, on the other hand, requires band-pass filtering to remove high- and low-frequency noise from the signal, additional low-pass filtering to create a reasonable control signal and, in some cases, independent component analysis (ICA)-based filtering to remove cross-talk that occurs when measuring biological signals. For band-pass EMG filtering, high-pass cut-off frequencies are between 5–100 Hz, while the low-pass cut-off frequencies are between 300–10,000 Hz [46, 96, 121, 131–134]. Next, a low-pass filter is applied as the final step in the signal processing of EMG to determine the *linear envelope*. The linear envelope of an EMG signal is a more interpretable and smooth signal than the band-pass filtered signal alone and obtained through a low-pass fitler with a cut-off frequency in the range of 2–10 Hz [44, 121, 135–139]. Finally, ICA-based filtering can be used to reduce cross-talk through decoding of a single channel into independent components [133]. Ruiz *et al.* use ICA-based filtering to remove electrocardiography signal artifacts from their EMG recordings [96, 140].

Filtering plays a significant roles in enabling human motion estimation and control of wearable assistive devices. The amount or specifics of the filters must change to meet the characteristics of the signal, which can vary between humans and devices. Filters are necessary to create useful signals, but come at a cost of computational resources. It is common to implement filters using electrical circuitry. However, this limits the flexibility of the filter and increases the space required to house the computer circuitry. Careful considerations are required to ensure that filters generate signals that meet system requirements.

2.5.5 Processing Elbow Motion Data

Enabling accurate and smooth control requires processing the collected elbow motion data prior to its usage with other components of the developed control systems. In this thesis, both the position and the EMG data are used for control purposes and need to be processed. Position data go through a two-stage signal processing pipeline involving filtering, to smooth the input signal, and rate conversion, to match the sampling frequencies of the EMG signals. Discretization can either be handled by stand-alone sensing systems or implemented within the customized control system circuitry. Many control models require the sampling frequency of all input signals to be the same, which is the reason rate conversion is used. The EMG data are transformed using a three-stage signal processing pipeline, which includes filtering and scaling of the signals. Fig. 2.10 shows the generalized signals processing pipelines for the position and EMG data. However, the specific design of these signal processing pipelines depends on the properties of other control system components and should vary accordingly.

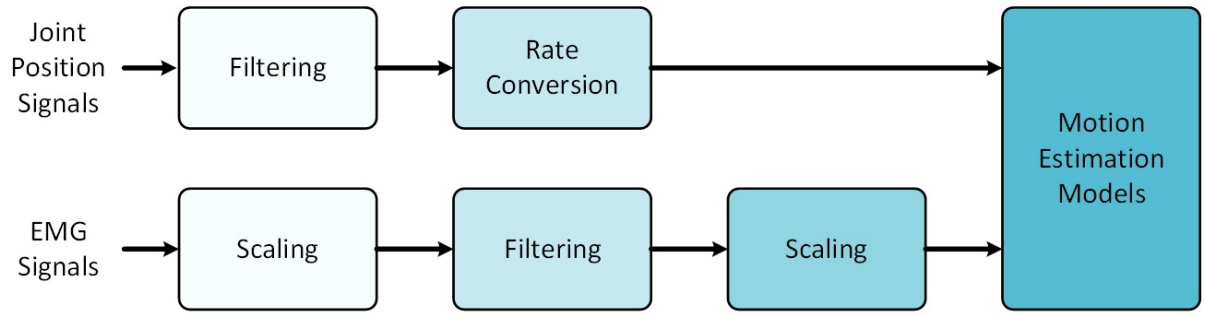


Figure 2.10: Generalized signal processing pipelines for transforming the joint position and EMG signals used in this thesis.

2.5.5.1 Position Signal Processing

The first step in processing the position signal and its derivatives, such as velocity and acceleration, is to filter the noise and smooth the signals. Voluntary elbow motion occurs at frequencies of less than 3 Hz [116], and, typically, far less for assisted elbow motions. Low-pass Butterworth filters have been applied to position-based signals in order to remove the noise and smooth the signals [25, 97, 116, 141–143]. The cut-off frequency can be derived from analysis of the collected position signals before and after filtering. A comparison of unfiltered and filtered position, velocity, and acceleration signals is presented in Fig. 2.11.

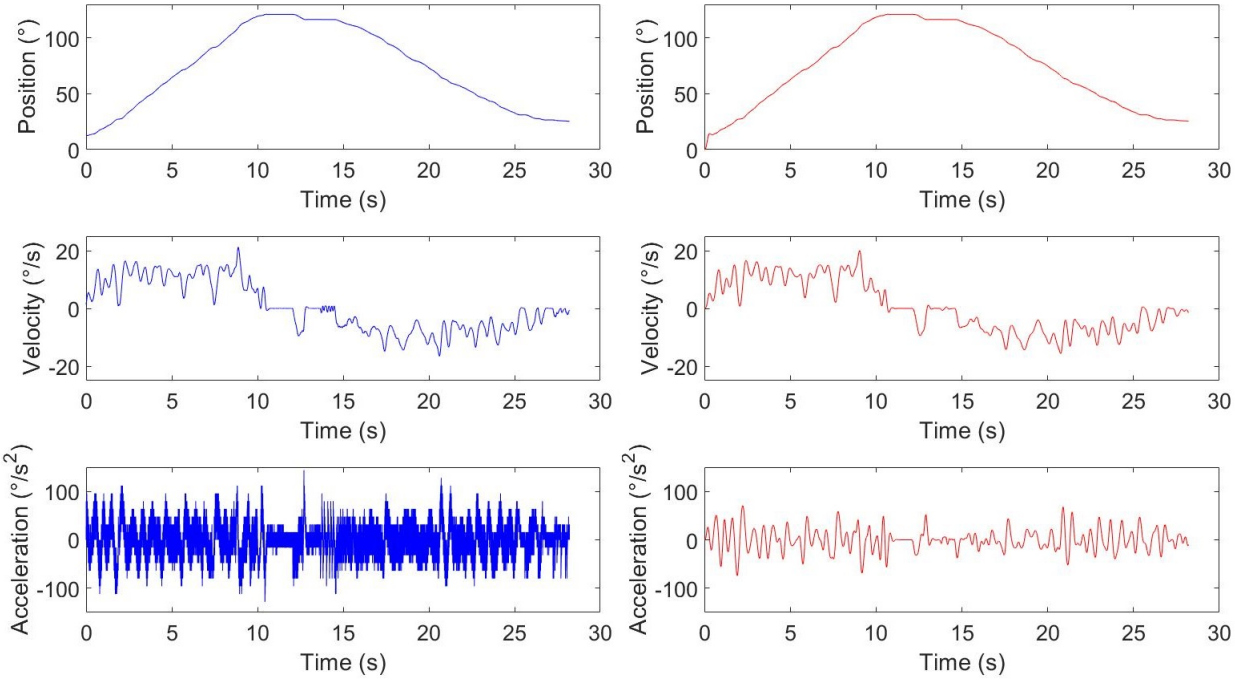


Figure 2.11: A fourth-order low-pass Butterworth filter with a cut-off frequency of 3 Hz is applied to the raw (left) position (top), velocity (middle), and acceleration (bottom) data to remove noise from these signals, producing filtered (right) position, velocity, and acceleration data.

Position data are usually sampled at much lower frequencies than EMG data, due to the information in position signals occurring at lower frequencies. To address this issue, sampling rate conversion algorithms are used. There are no restrictions for choosing the sampling rate at which all the signals will be synchronized. However, a higher sampling rate means more computational resources are required per unit of time.

Choosing the correct sampling rate and rate conversion procedures requires looking at the demands for both the position and EMG signals. Considering that voluntary elbow motion occurs at frequencies lower than 3 Hz, a sampling rate of 6 Hz would be the minimum requirement for the position-based signals. In terms of using EMG data for control, a large portion of the community uses a 500 Hz upper frequency cut-off for determining motion intention information [95, 117, 144–147]. As a result, EMG signals must be sampled at a frequency of 1000 Hz or higher to capture all of the desired data. During real-time control, each window of data to be analyzed will need to be put through the rate conversion process. Limiting the amount of rate conversion is important as the computational resources of wearable assistive devices are limited.

2.5.5.2 EMG Signal Processing

A general EMG signal processing procedure may consist of DC offset removal, band-pass filtering, rectification, and normalization. DC offset removal involves shifting the entire signal, such that the EMG signal oscillates around 0 volts. This can be accomplished by analyzing the signal to determine the oscillation point and subtracting this offset from the collected EMG signal. The band-pass filter is used to remove high-frequency noise and low-frequency motion artifacts from the EMG signal. Due to variations in the frequencies, the community is not unified on the cut-off frequencies to use for the band-pass filters. De Luca states that the main power of the signal is situated in the 0–500 Hz band and that filtering the signal in the 20–500 Hz band increases the signal to noise ratio [134]. In the literature, high-pass cut off frequencies range between 5–100 Hz and low-pass frequencies between 300-10 000 Hz [46, 96, 121, 131–134]. Fig. 2.12 shows a comparison of collected EMG signals before and after the various band-pass filters have been applied.

Once the EMG signals have been filtered to the desired frequency band, the next step is to scale the signal. Many of the models for estimating human motion rely on a signal representing the muscle activation as a quantity within the bounds of 0 to 1 [121]. However, EMG signals are a voltage signal that is both positive and negative. Scaling the EMG signal involves a two-step process of rectification and normalization. Rectification is a process that transforms a signal oscillating about a point to be bounded by only oscillating above or below that point. To remove

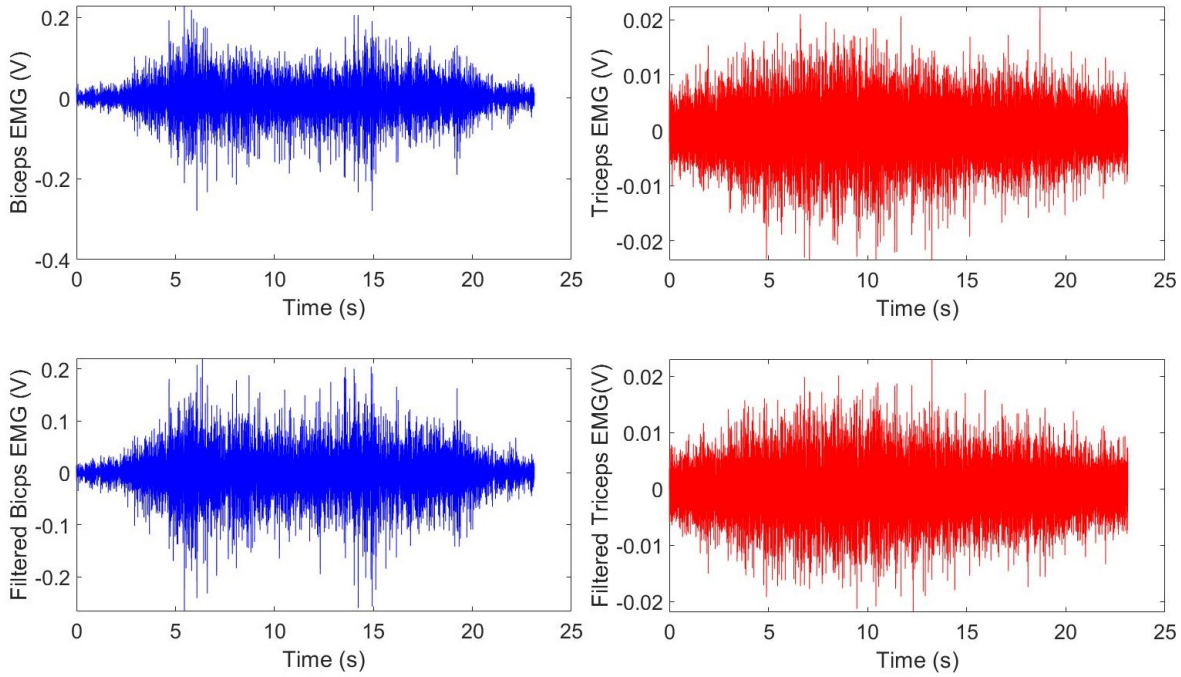


Figure 2.12: Band-pass filters, such as the fourth-order Butterworth with 20–300 Hz bandwidth used here, is applied to EMG signals to remove unwanted noise. The differences between the raw EMG signals (top) of the biceps brachii (blue) and triceps brachii (red) muscles and the filtered EMG signals (bottom) are subtle and hard to detect by eye.

the negative voltages of the collected EMG signal, an absolute value function can be applied to the signal. This will transform the signal such that is oscillates only between 0 and some positive maximum voltage. The rectification process makes the EMG signal easier to interpret and extract features. It has been used extensively in the processing of EMG signals for control and motion estimation [25, 36, 89, 121–125, 133, 144, 148–155]. After rectification, the signal has a lower bound of 0, but is bounded at the top based on the maximum voltage detected from the subject. Normalization is an attempt to provide an upper bound of 1 for the signal and has been used by many researchers when processing EMG signals [25, 39, 89, 122–125, 128, 150, 155–161]. Commonly, the subject or user would perform an MVC of the desired muscle and the maximum voltage of that signal would be used as the value from which to scale the collected EMG signals. However, this does not always ensure the signal remains within the bounds due to variability of the EMG signals. Fig. 2.13 shows the transformation of the filtered EMG into a rectified and normalized signal. The end result of this signal processing pipeline is a smooth signal bounded between 0 and

1, which can be used as input to the motion estimation models.

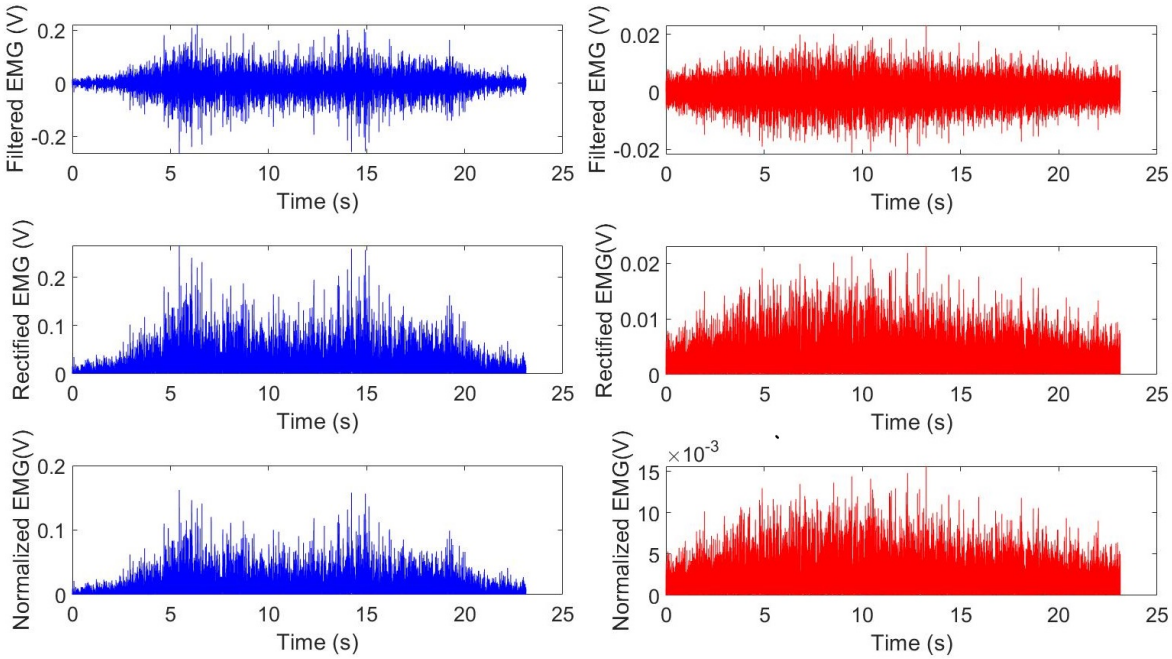


Figure 2.13: A comparison of the filtered (top), rectified (middle), and normalized (bottom) EMG signals of the biceps brachii (blue) and triceps brachii (red) muscles.

2.6 Communication of Digital Data

The third fundamental knowledge area surrounds communication of digital data. Design of communication channels becomes an important factor in the development of computer systems that enable wearable assistive devices. Modern digital communication consists of many complex interacting entities, which require thoughtful planning in order to consistently relay information between them. Standard protocols and processes exist within computer systems that allow engineers to customize communication systems. However, humans and computers process information differently, creating a need for translational technologies in order to facilitate useful interactions. Over the years, technological advancements have allowed for humans and computer systems to have meaningful communication through interaction devices, such as keyboards or biological sensors. These standards and technologies become the fundamental components in the communication architectures of wearable assistive devices.

The uniqueness and complexity of each wearable assistive device and its intended application result in a large variety of custom communication architectures. Designing these communication architectures begins by considering the formatting, sequencing, and time requirements of the desired communications. The representation or format of data is crucial for one system to be able to understand and act upon the data they receive from another system. Humans are more robust to variations in communication, however, digital systems are not. The format of the digital data, including the number of the bytes, data types, and physical-world representation must be defined to ensure proper communication is achieved. Even the binary-to-text encoding format, such as the ASCII and UTF-8 formats, needs to be used by two communication systems to ensure proper exchanges of information. In addition, data that are inconsistently sequenced will require additional functionality, such as data retransmission behaviours, to complete communications, even if the data are formatted properly. The order of the bytes in a data packet, the data packets in a message, and the message within the communication protocol must occur in sequences that are known to all systems involved in the communication. The timing of data exchanges will vary significantly based on the requirements of the communicating system. Since the computational resources are typically shared between communication channels and other systems operations, it is important that proper timing designs are made to reduce or eliminate scenarios in which communications are not delivered on time. One byte of data improperly formatted, out of sequence, or mistimed can have negative consequences on the functionality of digital systems and, therefore, impacts the operation of wearable assistive devices.

2.6.1 Communication Architectures

Wearable assistive devices require many communication channels to relay information between the human and device, as well as between components of the device. The culmination of these communication pathways constitute the *communication architecture*. In traditional engineering fashion, the complexity of these communication architectures can be decomposed into individual communication channels and designed in isolation. For example, some digital control systems may require the user or therapist to program the anatomical parameters [27, 81, 122], identify the beginning or end of a motion [44], and send motion data to the device [29,38,44,64,81,82,124,125],

as forms of communication.

Varying complexities of communications protocols are also required between the internal components of the digital control systems. Between microcontrollers and peripheral components binary digital signals are commonly used to dictate actions. For example, a microcontroller may enable a sampling operation of an ADC using binary digital signals to gather data about servomotor currents, such as in the digital controller implemented by Delph *et al.* [64]. In software, motion estimation functions require sets of motion data, collected from the sensors, to estimate intended motion. The EMG-driven motion estimation models developed by Kyrylova *et al.* and Desplenter *et al.* require sets of EMG and joint position data, represented as floating-point numbers, to generate estimations [123–125]. In order to relay information back to the patient and therapist, control systems need to have communication pathways to digital displays. Xing *et al.* [83], Tong *et al.* [81] and Aubin *et al.* [58] have all incorporated LCD screens into their devices as a means for the control system to relay information back to the user. Facilitating all of these communications, together with other functionalities, to meet real-time control constraints, becomes a difficult task. As a result, part of the design process for communication architectures involves the integration of the individual communication pathway designs into a system-level design.

To reiterate, the communication architecture must take into account the formatting, sequencing, and timing of the data communications. The formatting and sequencing of data are mostly addressed during the design of the individual communication pathways. It is in the system-level integration of these pathways that the timing of the communication is considered. The simplest design would consider one computer system to facilitate each communication pathway. However, this minimization of complexity comes at a cost of size, power consumption, and computational resources. On the other end of the spectrum, the entire communication structure would be implemented on one computer system. Some wearable assistive devices can require up to Gigabytes of data to be transmitted per second. Facilitating this requirement on a single computer system, embedded in the devices or on the user, may not feasible with current computer technology. Therefore, the timing of the communication and the required computational resources to ensure proper communication are tightly coupled.

In order to meet timing requirements, engineers must estimate, or measure, and account for the

delays present in both the human and computer systems. The nervous system of the human body experiences delays in transmission of electrical signals of between 23 and 131 ms [135]. Computer systems also experience delays in the transmission of electrical signals. For any fundamental logic gate, there exists a time delay required for the gate to change states once a new input signal is given. This is known as the gate delay of components in digital logic systems. The reader is forwarded to Mano and Ciletti's *Digital Design* book for more details on this topic [162]. For microprocessor-based systems, the summation of state transition delays of all of the logic gates required to execute one processor instruction represents the instruction execution delay. Therefore, for a given series of instructions that must be executed to produce a communication, the timing constraints can be determined for each computer system that is proposed to house the communication architecture.

2.6.2 Wireless Communication

Removing aspects that tether mechatronic systems to one location in space is one of the key steps to increasing their wearability. Part of the solution lies in the incoporation of wireless communication technologies into these devices. Having a wireless system means that the user can move their entire body while interacting with the device, increasing the potential applications for the devices.

Wireless communication has been implemented in existing wearable assistive devices [25, 31, 37, 42, 43, 81, 163, 164]. These implementations have been completed using one of either the Zigbee, Bluetooth, or ultra-wide band (UWB) wireless protocols. The devices created by Tong *et al.* and Nycz *et al.* include components that provided wireless communication over the Zigbee protocol. The Bluetooth wireless protocol is the most commonly used in wearable assistive devices. Kyrylova *et al.* and Desplenter *et al.* used the Biosignalsplux sensing platform (PLUX, Portugal) to receive sensor data over its Bluetooth communication channel [25, 37, 124]. Pylatiuk *et al.*, Rocon *et al.* and Ren *et al.* enabled their wireless communication through electrical circuitry containing Bluetooth modules [42, 43, 164]. Finally, a UWB wireless network was developed by Lessard *et al.* to enable communication between IMU sensors and the main controller [31]. Although wireless communication is a key component to the success of wearable mechatronic devices, many of the existing devices still employ wired communication methods. This is likely due to these devices being in an early stage of research and development but should be expected to shift towards

wireless communication systems as the devices evolve.

The wireless communication protocol should be chosen based on the specifications of the application and the constraints of the components of the device. The number and complexity of communications will dictate the required data transfer rate of each communication pathway. When considering wireless communication alternatives, power consumption and electronic circuitry volume are two major design quantities that are directly correlated with the data rate, both of which are constrained in embedded computer systems. As the data transfer rate increases, the number of processor operations increases as well. This directly correlates to a higher power consumption per unit time. For example, for a data transfer rate of 5 Mb/s, the system would require at least 5 Bluetooth modules transfering in parallel, according to the comparison conducted by Pothuganti and Chitneni [165]. One UWB or Wi-Fi module would satisfy this data transfer rate requirement but the normalized power consumption of these protocols is about 6–7 times that of the Bluetooth protocol (see Fig. 1 in [165]). As the data transfer rate increases, the demands on the processing units also increase. At some point, multiple processing units will be required to handle the data transfer rate. This leads to extra circuitry and components, which increases power consumption as well, in order to enable co-ordination of the data transfer between processing units. Data transmission rates is discussed based on the results is presented in Chapter 5 Section 5.4.6.

2.7 Modeling Human Motion

The fourth fundamental knowledge area involves developing and implementing human motion models. Motion gives us the means to learn, to adapt, and to survive. To most, human movement may seem trivial, since the ability is inherent to the observer. To some, the complexity of human motion is intriguing and the study of it provides information that both directly and indirectly benefits society. In engineering, models are developed to help understand and characterize a system. Therefore, the task becomes to design models of the human motion system in order to understand how the various properties of the human body relate to its motion. Models that fall into this category are called *motion models*.

Decomposing the human motion system into one or more models is dependent on the level to

which the observer considers that the motion is being produced. At the highest level, motion can be considered a product of electrical stimulus of the musculature and, at the lowest levels, motion can be viewed as interactions between atoms. Regardless of the perspective, the study of human motion and the generation of motion models is fundamental to facilitating interactions between humans and wearable assistive devices. Only motion models that describe the motion system from a perspective where movement of the joints is a function of the soft tissues, such as muscles, will be considered in this chapter. This is due to the requirement for non-invasive sensing modalities. These motion models will become primary estimation-level components in the developed control systems of wearable assistive elbow devices.

The scope of this thesis will include modelling and motion estimation related to the upper limb. The goal is to examine and develop motion estimation models that can produce accurate estimates to guide the control systems of wearable assistive devices. In order to produce accurate estimates, the dynamic properties contributing to joint motion must be modelled. Many models have been proposed in the literature, including the formulation of general musculoskeletal motion models. In the following sections, a review of the existing upper limb motion models is presented.

2.7.1 Modelling Musculoskeletal System Dynamics

Motion is produced through a systematic interaction of musculoskeletal tissues. Initially, electrical signals are produced by the brain and delivered to the muscle through the nervous system. The signals are interpreted by receptors in the muscles, which cause them to contract and produce forces. The forces are transferred to the bones, which cause motion of the joints. In order to provide stability of the motion, the bones and passive soft tissues restrict the movement and provide resistive forces that help prevent injury. This scenario constitutes the major interactions of components of the human motion system.

Each component of this system has a set of properties that can be described through one or more sets of equations, using one or more parameters. Some parameters, such as limb segment lengths, mass, or height, can be measured from the individual. Other properties of soft tissues, which can vary significantly from person to person, cannot be measured easily and must be estimated from cadaver studies [166]. Properties regarding limb segment mass or inertia can be estimated

or calculated from formulas or tables found in the literature [167, 168]. Parameters related to the forces produced by musculoskeletal tissues cannot be measured using non-invasive techniques and must be estimated, typically through interactions with sensing systems. Measurement and estimation of these parameters are essential to developing effective models but these processes require a substantial amount of time and resources.

The level of accuracy and flexibility of these motion models relies on modelling the functional behaviours of all force sources that contribute to the motion of each joint. In general, internal forces stem from bones, ligaments, cartilage, tendons, and muscles. Each of these tissue types exhibits properties that determine the force production of the tissue. Understanding and modelling the relationships between these properties and the equations and parameters that represent them is essential to producing accurate motion estimates and optimizing the model. A generalized motion model provides a foundation for these relationships and a place to begin decomposing the complexity of the human motion systems.

2.7.2 A Generalized Motion Model

Motion models should take into account all force sources in order to generate realistic motion estimates. As it will be shown in Section 2.7.3, existing models do not always provide mathematical descriptions for every force generating component of the musculoskeletal system. Therefore, it is important to compare between these components in order to determine variations, classify the models, and assess their function and limitations. In order to aid in these tasks, a generalized motion model is extended from those proposed in the literature.

The generalized motion model is based on a *tissue-level decomposition* perspective. Tissue-level decomposition defines the soft tissues, such as muscles, as uniform components with uniform dynamic properties, as opposed to a set of smaller components, such as muscle fibers, each with its own unique properties. This assumption reduces the complexity of the system and limits the number of parameters that need to be determined. However, it does not fully encompass the reality of the human motion system, since muscle are made of muscle fibers that each have their own properties. Based on this premise, the forces generated by muscles, tendons, ligaments, cartilage, and bones, can be described using relationships of the overall tissue properties, as opposed to

the properties of the components that make up these tissues. Tissue-level of decomposition of the dynamic properties of the human motion system lends itself well to studies where data are collected non-invasively.

The usefulness of a generalized motion model is evident in the literature, as multiple researchers have proposed generalized tissue-level motion models [121,128,129,169]. However, these generalized models describe the system using only components related to the generation of muscle forces, as shown in Fig. 2.14. These versions of a generalized muscle model operate using some basic assumptions that enable other force generating components to be excluded from the model. First, bone is treated as infinitely rigid material, meaning no deformation dynamics are accounted for in the transfer of forces from one joint to another. Second, ligament forces are not modelled, which is likely due to the fact that they cannot be measured non-invasively and are not actively controlled by the nervous system. Third, cartilage between articulating surfaces of the joint are not included in these models, under the assumption that the joint is near frictionless. Lastly, muscles and tendons are commonly modelled as musculotendon units, which means a tendon model is not explicitly defined in these generalized motion models. Therefore, these models decompose motion into components that represent only the muscle and the skeleton.

The existing generalized motion models do not always include graphical representations of all musculoskeletal components. However, it does not mean that these forces have not been included in the mathematical representation of the models. One example of this is the inclusion of tendon behaviour into musculotendon models [96,121,128,129,169–172]. In Section 2.7.3.3, existing solutions that account for these non-muscular forces, such as bone, ligament, and cartilage forces, will be explored. For now, an extension of the generalized motion model proposed by Buchanan *et al.* is presented in Fig. 2.15 [121]. In this model, a component is added to account for the neural activation dynamics. Buchanan *et al.* describe this component as one that captures the linear transformations of the raw or processed EMG signals into input signals for the muscle activation dynamics component, even though they did not include it in their generalized motion model. Looking across the literature, many researchers have developed mathematical transforms that fit this criteria, which makes it a good candidate to be added to the generalized motion model [96,121,127,131,144,157,172,173]. This model will be used as a basis for comparison of the

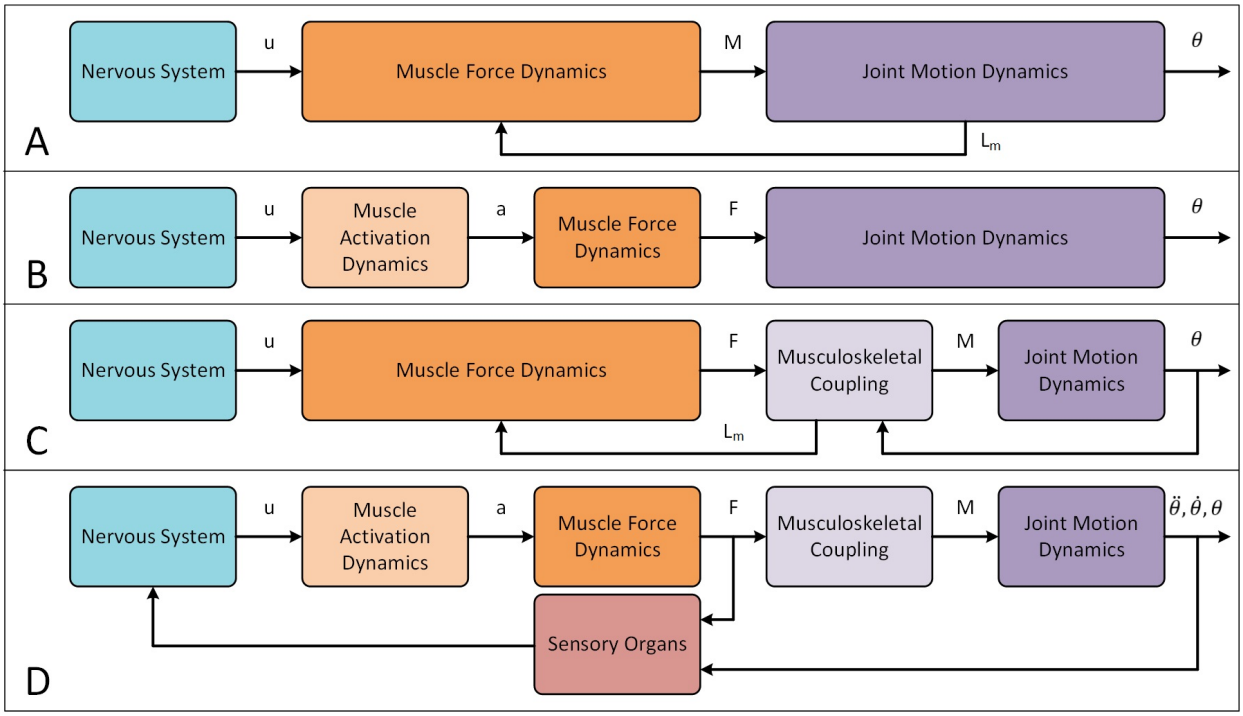


Figure 2.14: Comparison of generalized motion models proposed by Gao *et al.* [129] (A), Bai *et al.* [128] (B), Chadwick *et al.* [169] (C), and Buchanan *et al.* [121] (D). u is the neural activation signal, a is the muscle activation signal, F is the muscle force, L_m is the muscle length, M is the moment caused by the muscle, and θ, $\dot{\theta}$, and $\ddot{\theta}$ are the joint position, velocity, and acceleration, respectively.

existing upper limb motion models.

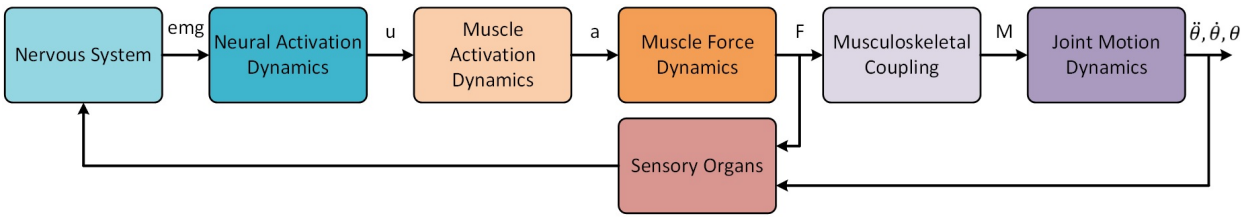


Figure 2.15: An extension of the generalized motion model presented by Buchanan *et al.* to include a component for the neural activation dynamics [121]. *emg* is the processed EMG signal, u is the neural activation signal, a is the muscle activation signal, F is the muscle force, M is the moment caused by the muscle, and θ, $\dot{\theta}$, and $\ddot{\theta}$ are the joint position, velocity, and acceleration, respectively.

2.7.3 Review of Existing Motion Models

Many models for estimation of upper limb motion parameters have been proposed in the literature. Determining which motion models to implement for control of wearable assistive devices begins with a review of the existing models. A literature survey was conducted on the ACM Digital Library, IEEE Xplore, Scopus, and Inspec and Compex databases. Combinations of the terms 'dynamic', 'arm', 'motion', 'model', 'elbow', 'joint', 'stiffness', 'EMG', 'feature', and 'relationship' were used in this survey. The search resulted in 26 upper limb motion models, which were examined with two main objectives. First, the existing motion models were examined using the generalized motion model (Section 2.7.2) to determine which dynamic components of the human motion system were modelled. This aided in understanding the existing models and identifying areas for improvement or evaluation. Second, the experimental methods and results were compared to determine the successes of the motion model evaluations. The results of existing motion model experiments will give developers targets for improvement.

To meet the first objective, the existing models were reviewed and decomposed using the generalized motion model (Fig. 2.15). Upon review, it was discovered that most of the models do not explicitly include components that correspond to those in the generalized motion model. Moody *et al.* were the only model developers to include bone deformation dynamics into their model [170]. All other existing models treat bone as beams made of infinitely rigid material. Chadwick *et al* defined a ligament model for the conoid ligament to restrict axial rotation of the clavicle in their model [169]. This was the only explicitly defined ligament model among the reviewed motion models. None of the existing models defined components to represent the cartilage present on the articulating surfaces of the joint. However, some motion models did include passive joint torques that aim to account for some of the forces generated by the ligaments and cartilage. Across the reviewed models, the only constant was the definition of a joint motion model, as the functionality of these models is to produce estimates of joint motion. The variation in these models stems mostly in the modelling of forces generated through the control of the musculotendon units and the combination of joint forces in the joint motion models. In fact, only three of the existing musculotendon models contained components for each of the neural activation dynamics,

muscle activation dynamics, and muscle contraction dynamics. Based on these observations, Table 2.5 provides a comparison of components included in the musculotendon models of the reviewed motion models. Further discussion of the musculotendon models, joint motion models, and bone, ligament and cartilage forces is presented in Section 2.7.3.1, Section 2.7.3.2, and Section 2.7.3.3, respectively.

The second objective was met by examining aspects of the experimental protocol and results. These aspects included the methods for generating input signals, the dynamic nature of the input data, the parameter determination techniques, and the estimation errors. It is important that these models produce accurate estimates based on data gathered from human subjects. This is reflected in the fact that 21 of the experiments involved the collection of human motion data to use as input for the model. However, given the variability in human subjects, using simulated motion inputs may help to control for this variability. Both static and dynamic motion inputs were used to evaluate these models, with 22 experiments using dynamic motion inputs, 4 using static inputs, and 2 using both types of inputs. In order to determine models parameters, the experimental evaluations involved a number of methods, including gathering values from existing studies, measuring aspects of the subject from which the data were recorded, and tuning the models using optimization algorithms. The most common parameter determination methods were using values from existing studies [96, 144, 169, 170, 174, 175], variations of the least-squares algorithms [126,127,171,173,176,177], and constrained nonlinear minimizations [96,169,178]. The estimation error is the most important characteristic in evaluating the abilities of these models but was only reported in 11 of the reviewed articles, while only 8 of these errors reported a standard deviation measurement. A comparison of these experimental parameters and results is presented in Table 2.5.

2.7.3.1 Components of Musculotendon Force Generation

A musculotendon model describes the dynamic force response of both the muscle and tendon. These two soft tissue types are lumped together to match their biological arrangement. In the body, muscle forces are transfered to bone segments through their connections to the tendons. The inability to measure the musculotendon units and dissect them in living human bodies means

Table 2.5: Decomposition of existing musculotendon models into components of the generalized motion model presented in Fig. 2.15. A '✓' indicates that the authors have included the component in their musculoskeletal motion model, while an '-' indicates that the component was not included in the model.

Author [Reference]	**Musculotendon Model Components**		
	Neural Activation Dynamics	Muscle Activation Dynamics	Musculotendon Contraction Dynamics
Abdullah [179]	-	-	-
Bai [128]	-	✓	✓
Baiqing [166]	-	-	-
Bayati [180]	-	✓	-
Chadwick [174]	-	✓	✓
Chadwick [169]	-	✓	✓
Clancy [127]	✓	-	-
Ding [172]	✓	✓	✓
Fu [178]	-	-	-
Gao [129]	-	✓	✓
Kang [175]	-	-	-
Kashima [126]	✓	-	✓
Katsiaris [181]	-	-	-
Kiguchi [119]	-	-	-
Konyk [182]	-	-	-
Lakatos [176]	-	-	✓
Liu [173]	✓	-	-
Moody [170]	-	✓	✓
Nagarsheth [167]	-	-	-
Peng [144]	✓	-	-

Author [Reference]	**Musculotendon Model Components**		
	Neural Activation Dynamics	Muscle Activation Dynamics	Musculotendon Contraction Dynamics
Rosen [157]	✓	-	✓
Ruiz [96]	✓	✓	✓
Song [131]	✓	-	-
Venture [177]	-	-	-
Venture [171]	-	✓	✓
Wang [183]	-	-	✓

that musculotendon models are still being explored. In general, musculotendon models are driven through activation signals that are either simulated or derived from EMG signals.

Deriving activation signals from the EMG signals is difficult. This is due to the lack of understanding surrounding the mechanisms that generate EMG signals and the specific information that is encoded within them. The presence of both linear and nonlinear isometric EMG–force relationships has been identified for different muscles [184, 185]. As a result, a neural activation model precedes the use of a musculotendon model to account for some linear dynamics of this relationship. Some models, such as the Hill-type models, incorporate the nonlinear dynamics into a sub-model known as a muscle activation model. Other models do not explicitly account for these dynamics and lump them together with the musculotendon contraction model.

Considering the neural activation, muscle activation, and musculotendon contraction models, there are four main variations found across the reviewed devices. First, motion models that are trying to derive muscle forces from joint motion use completely lumped musculotendon models, such as the Kiguchi and Hayashi's neuro-fuzzy musculotendon model [119]. In these types of musculotendon models, typically one equation is used to describe the relationship between the model inputs and muscle force outputs. Second, neural activation models are used to smooth the EMG inputs but the muscle activation and muscle contraction dynamics are lumped together. For example, Clancy *et al.* used a neural activation model to transform EMG data for input to variations of polynomials and linear time invariant systems that represent the muscle activation and

Table 2.6: Experimental factors of the existing upper limb motion models studies.

Author [Reference]	**Input Data**		**Parameter Determination Methods**
	Simulated or Collected	Static or Dynamic	
Abdullah [179]	Collected	Dynamic	Subject measurements
Bai [128]	Collected	Dynamic	Direct search algorithm Conjugate search algorithm
Baiqing [166]	Simulated	Static	Lagrange's multipliers optimization method
Bayati [180]	Simulated	Dynamic	Gradient project algorithm
Chadwick [174]	Simulated	Dynamic	Existing studies First-order Euler solver Fourth-order Runge–Kutte solver
Chadwick [169]	Simulated	Dynamic	Existing studies Constrained nonlinear minimization
Clancy [127]	Collected	Static	Linear least squares algorithm Nonlinear least squares algorithm
Ding [172]	Collected	Dynamic	Levenberg-Marquardt algorithm
Fu [178]	Collected	Static	Constrained nonlinear minimization
Gao [129]	Collected	Dynamic	No details
Kang [175]	Simulated	Dynamic	Existing studies Fourth order Runge-Kutta method
Kashima [126]	Collected	Both	Transition matrix algorithm Least-square algorithm
Katsiaris [181]	Collected	Both	Chow-Liu algorithm
Kiguchi [119]	Collected	Dynamic	Error back propogation learning algorithm
Konyk [182]	Simulated	Dynamic	Manual selection
Lakatos [176]	Collected	Dynamic	Least squares algorithm
Liu [173]	Collected	Static	Linear least-squares algorithm with pseudoinverse Nonlinear least squares algorithm
Moody [170]	Simulated	Dynamic	Existing studies

Author [Reference]	Input Data		Parameter Determination Methods
	Simulated or Collected	Static or Dynamic	
Nagarsheth [167]	Collected	Dynamic	Curve fitting
Peng [144]	Collected	Dynamic	Existing studies Subject measurements Levenberg-Marquard back propogration algorithm
Rosen [157]	Collected	Dynamic	No details
Ruiz [96]	Collected	Dynamic	Existing studies Constrained nonlinear minimization
Song [131]	Collected	Dynamic	Levenberg-Marquardt algorithm
Venture [177]	Collected	Dynamic	Linear least-squares algorithm
Venture [171]	Collected	Dynamic	Nonlinear least squares using a Newton–Gauss algorithm
Wang [183]	Collected	Dynamic	No details

contraction dynamics [127]. Third, no neural activation models are used and the muscle activation and musculotendon contraction dynamics are defined separately. Gao *et al.* fit into this category as the processed EMG signals are used directly as input to a muscle activation function, prior to muscle contraction forces being calculated [129]. Lastly, motion models employing Hill-type musculotendon contraction models are decomposed in neural activation, muscle activation, and musculotendon contraction models. The Hill-type musculotendon model is the most commonly used musculotendon model of the reviewed motion models [96, 157, 169–172, 174]. The following sections will examine the existing neural activation, muscle activation, and musculotendon contraction models.

2.7.3.1.1 Neural Activation Models Neural activation models are used to model the linear dynamics in the conversion of EMG signals into muscle activation signals. Buchanan *et al.* introduced this as an intermediary step in the conversion of EMG signals to muscle activation signals [121]. To determine the neural activation signal, $u(t)$, time delay and historical components of the signal are modelled. Modelling these aspects is accomplished using one of two solutions in the reviewed motion models. The first solution is to use discrete version of the critically damped

linear second-order differential equation developed by Milner-Brown *et al.* [186]:

$$u(t) = \alpha \cdot emg(t-d) - \beta_1 * u(t-1) - \beta_2 * u(t-2) \tag{2.2}$$

where $emg(t)$ is the processed EMG signal, α_1, β_1, and β_2 are coefficients that determine the second-order dynamics, and d is the electromechanical delay [1]. Ding *et al.* employ this neural activation model after processing their collected EMG signals [172].

The second solution involves applying a low-pass filter after the EMG signal has been rectified. This was the most common solution among reviewed models [96, 127, 131, 144, 157, 173]. Four different cut-off frequencies were used in these filters, which included 3 Hz [131,144], 6 Hz [96,157], 16 Hz [173], and 16.4 Hz [127]. Rosen *et al.* implemented fourth-order Butterworth low-pass filters to determine the neural activation signal [157]. No other reviewed articles defined the type or order of the filter that was used. Although this solution is common among motion models, Buchanan *et al.* note that others have experienced difficulties estimating muscle forces from rectified and low-pass filtered signals, which can be mitigated by using solution presented in Eq. 2.2 [121]. No comparative studies of neural activation models could be found in the literature.

2.7.3.1.2 Muscle Activation Models Muscle activation models are another method for transforming the processed EMG signals into muscle activation signals. These models are commonly used as a second stage of transformation of the EMG signals. However, they have also been used to determine a muscle activation signal, $a(t)$, directly from the processed EMG signal [128, 129, 180]. There are three types of muscle activation functions found in the reviewed motion models: a first-order differential function, an exponential function, and a piecewise logarithmic/linear function. First order differential functions are the most commonly used muscle activation model [96, 169–171, 174, 180]. These functions describe the muscle activation in relationship to time constants that represent the time it takes to activate and deactivate the muscle fibers. Since these models rely on the previous value of the signal, one challenge is choosing an appropriate initial value since muscle activation is never zero. Ding *et al.* implemented a function using exponentials with a single parameter that determines the degree of nonlinearity of the

[1] See [121] for details on Eq. 2.2

model [172]. Bai *et al.* and Gao *et al.* used a piecewise function to represent the relationship between the neural activation and the muscle activation signals. The function consists of a logarithmic function to describe lower muscle activations (<30% activation) and a linear function to describe higher muscle activations (>30% activation) with four parameters. As it stands, further study is required to reach any conclusions about which muscle activation model produces the best estimation results.

2.7.3.1.3 Hill-Type Musculotendon Models Throughout the decades of studying human muscle, the Hill-type musculotendon models, also referred to the Hill muscle model or Hill-type model, have been established and used extensively in upper limb motion models, including those examined in the literature review [96, 157, 169–172, 174]. Here, the Hill-type model is defined as a musculotendon model due to the fact that the model commonly includes components representing both the muscle and tendon. The general Hill-type musculotendon model consists of a series passive element (SPE), parallel passive element (PPE), and contractile element (CE) (Fig. 2.16 A) [96, 121, 169, 171, 187].

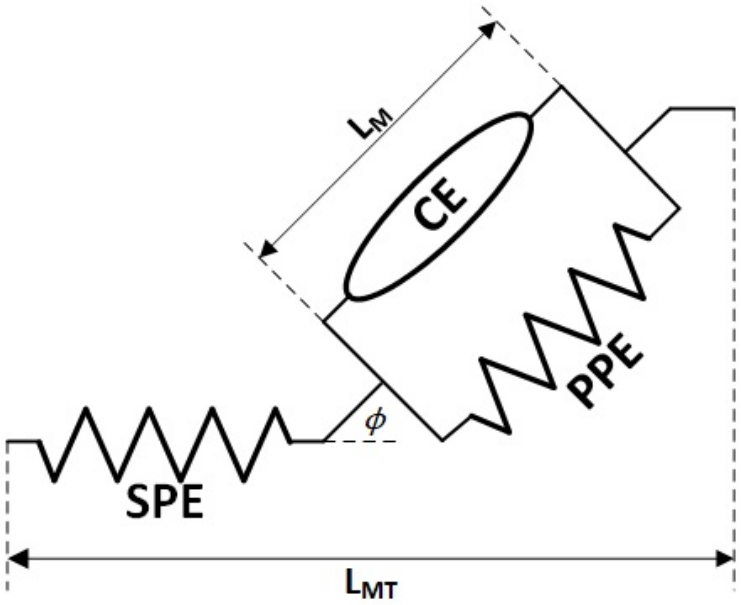


Figure 2.16: The Hill-type musculotendon model consists of a series passive element, a parallel passive element, and a contractile element. Together these components describe the passive force generated by the tendon, the passive and active forces generated by the muscle, and the relationship between these components.

In the general Hill-type model, the SPE and the PPE are used to represent the viscoelastic response of the tendon and muscle, respectively, while the CE defines the active contractile force generated by the muscle. Here, these elements are described abstractly since there is significant variation in how each of the elements are implemented across the reviewed motion models. The variations of the SPE and PPE elements of the reviewed models are shown in Fig. 2.17 and Fig. 2.18. In these models, the SPEs were modelled as either a spring [96, 169–171, 174], a rigid tendon element [172], or an exponential function [157]. Modelling the tendon as a rigid element is essentially treating it as if there is no elastic response in this element. However, Millard *et al.* showed that there was an increase in simulation speeds and similar estimation accuracy when using a rigid tendon element compared to an elastic element if the models are sub-maximally activated [188]. Four variations of PPEs were defined in the literature, including a spring [96, 169, 171, 172], a spring–damper system [174], a mass–spring–damper system [170], and an exponential function [157]. The spring model considers the passive muscle forces as a purely elastic response, while the spring–damper system captures elastic and viscous dynamics. Moody *et al.* were the only researchers to include a mass component in their Hill-type model [170]. This is contrary to most other models that factor in the effects of the mass into the gravitational components of the joint motion model. No comparisons have been conducted to determine the effectiveness of either method.

One area of commonality between these Hill-type models is the definition of the CE. The output of the CE represents the active portion of the muscular force, generated through either conscious or unconscious control of the muscle. The most common representation of this force, F_{active}, is defined as a multiplication of three normalized values and the maximum isometric force of the muscle as follows [96, 157, 169–172]:

$$F_{active} = a(t)F_{max}F_{fl}(L_M)F_{fv}(\dot{L}_m) \tag{2.3}$$

where F_{max} is the maximum isometric force, $F_{fl}(L_M)$ is the normalized isometric force–length relationship, $F_{fv}(\dot{L}_M)$ is the normalized force–velocity relationship of the muscle, L_M is the muscle fiber length, and $\dot{L}_M$ is the muscle fiber shortening velocity. Chadwick *et al.* were the only

researchers to propose a variation of this model involving the multiplication of three functions, as follows:

$$F_{active} = a(t)f(L_M)g(\dot{L}_m, a(t)) \tag{2.4}$$

where f defines the force–length relationship of the muscle and g defines the "Hill shortening equation" [174]. Regardless of the model form, it is important to note that the research community agrees that the active component of the muscle force exhibits relationships with the muscle length, the muscle velocity, and the activation signal stemming from the nervous system.

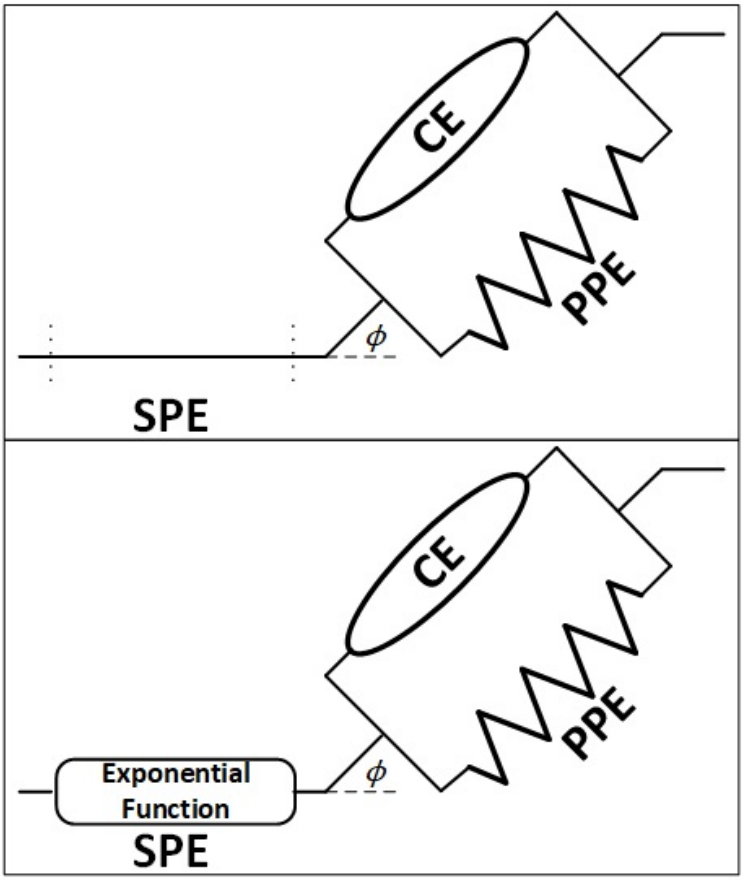


Figure 2.17: Variants of the SPE of Hill-type musculotendon models include rigid tendon models (top) and exponential functions (bottom).

As shown in Fig. 2.17 and Fig. 2.18, there are multiple variations of the Hill-type models used in the reviewed motion models. This is due to the fact that a perfect model of the musculotendon unit has not yet been determined. Fung notes, in his 1982 text *Biomechanics: Mechanical*

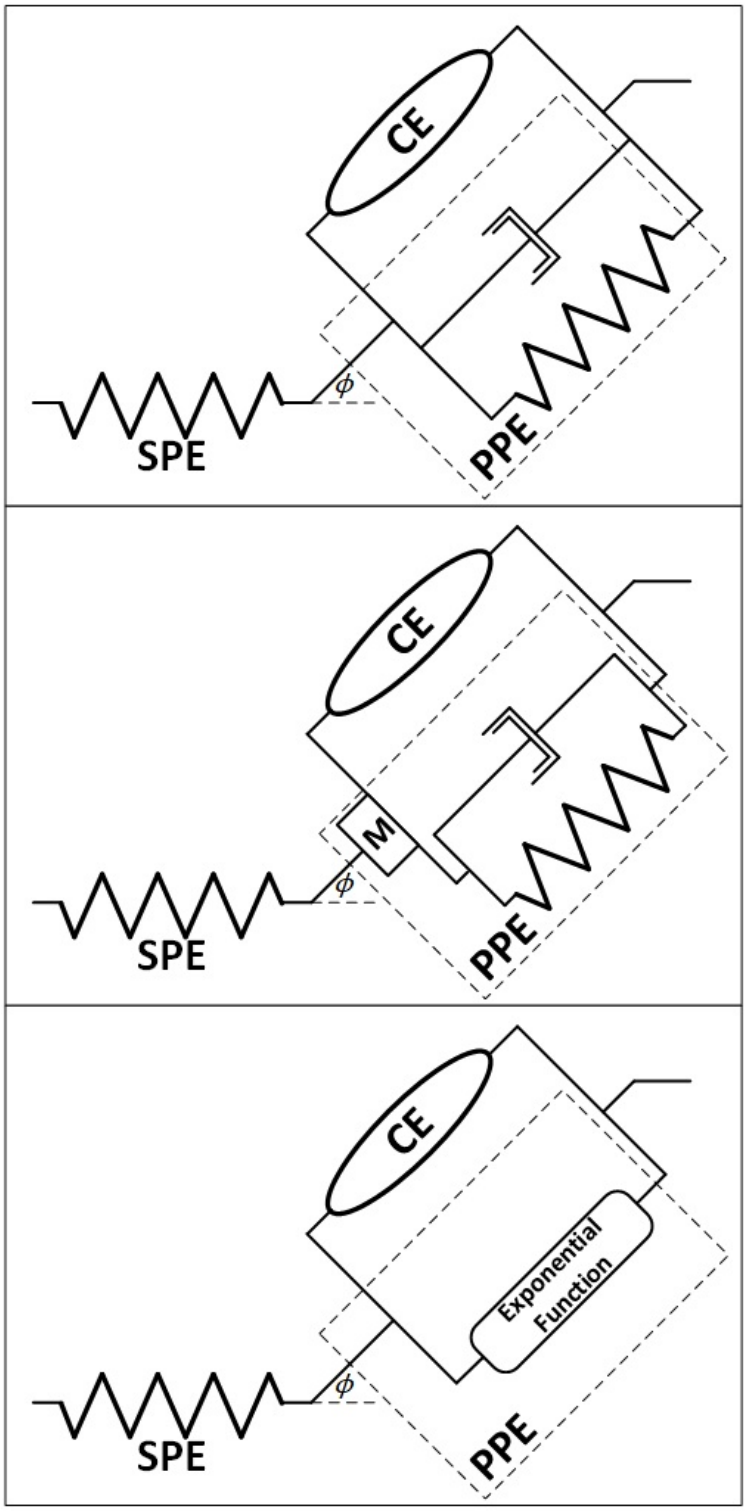


Figure 2.18: Variants of the PPE of Hill-type musculotendon models include spring–damper (top), mass–spring–damper (middle), and exponential (bottom) models.

Properties of Living Tissues, that the debate over the Hill-type model had already caused much controversy by that time [189]. Decades later there still exists no definitive consensus on the matter. However, this has allowed researchers to keep exploring and adapting the Hill-type model to match the assumptions and constraints of their specific problem. In 2013, Millard *et al.* conducted a comparative study of variations of the Hill-type model and their effects on computational resources [188]. However, in many cases they found no statistically significant differences, leaving the topic open for further debate.

The lack of a definitive Hill-type model is likely due to the simplicity of the model. The model does not account for individual characteristics of muscle bundles, differences in muscle fiber types, variations caused by muscle wrapping or displacements, or depletion of energy resources causing fatigue. Furthermore, the fact that most of the parameters of these models vary as a function of multiple parameters, such as time, but are chosen as constants, makes the estimates only valid for specific cases. Further study, improvement, and comparison of Hill-type models are essential to decomposing the musculotendon dynamics and developing more accurate and adaptable models.

2.7.3.1.4 Artificial Neural Network Musculotendon Models Artificial neural networks (ANNs) have become a popular modelling tool due their ability to produce relationships between inputs and outputs without explicitly describing the properties of the phenomena that is being modelled. ANNs were used as musculotendon models in three of the reviewed works [131,144,157]. All three of these models included a neural activation model in order to transform the processed EMG signal before being used as input to the ANN. Song *et al.* developed two three-layer recurrent ANN to estimate elbow torque [131]. One of the ANNs was developed to use EMG and kinematic inputs, while the other used only EMG inputs. Song *et al.* concluded that the ANN developed to use EMG and kinematic data exhibited higher accuracy estimations than the ANN that used only EMG inputs. Peng *et al.* developed and compared the ability of a multilayer feedforward ANN (MFANN) and a time delay ANN (TDANN) to estimate elbow torque, based on EMG and kinematic inputs [144]. Both the MFANN and TDANN used the current values of two EMG signals, angular displacement, and angular velocity as inputs, while the TDANN also used the previous four values of the two EMG signals as additional inputs. Peng *et al.* determined that

the MFANN could not effectively represent the EMG–torque relationships and concluded that the TDANN was a more suitable model.

Rosen *et al.* developed an three-layer ANN and compared it to a Hill-type musculotendon model [157]. In terms of torque estimation, the ANN was superior to the Hill-type model. However, Rosen *et al.* discusses that there is a trade-off between the two extreme cases of the model. The Hill-type model was not as accurate as the ANN, but can be easily adapted to many motion tasks, while the ANN's predictive power is specific to the task for which it has been trained. Kiguchi and Hayashi developed two variations of ANNs, both combined a neural network with fuzzy logic layers to produce estimates based on joint angles, EMG, and EEG inputs [119]. The output of these models were weighting factors that were then multiplied by the EMG and EEG signals in order to determine joint torques and hand velocities. The results of Kiguchi and Hayashi's study showed that EMG only model, performed better than one EMG + EEG model in estimating hand velocity. Although these ANNs have produced accurate estimates, further analysis is required to compare amongst variations of the ANN solutions and between ANN musculotendon models and other types of musculotendon models.

2.7.3.1.5 Alternative Musculotendon Models Both Hill-type models and ANN represent two extremes for representation of the musculotendon dynamics. These dynamics are most decomposed in Hill-type models and most aggregated using ANNs, which act as black-box models. However, other models have been proposed that lie somewhere along this spectrum. Gao *et al.* and Bai *et al.* both used a similar model, which implemented a linear dimensionless musculotendon actuator as described by Zajac [187], to estimate joint torque [128, 129]. Bayati *et al.* state that the muscle acts a nonlinear low-pass filter but did not provide a mathematical description of their musculotendon model [180]. Kashima *et al* developed a first-order differential equation relating the neural activation to the isometric muscle force using a time constant [126]. Wang *et al.* proposed a mass–spring–friction model of each muscle [183]. Clancy *et al.* compared 4 different models comprised of variations of linear time invariant systems and nonlinear polynomial functions that provided a direct relationship between EMG and torque [127]. Liu *et al.* developed three different nonlinear polynomial models for evaluating the effects of joint angle on constant posture

EMG–torque relationship [173]. The variance of the proposed musculotendon models supports the need for continual study of human motion in order to improve upon the successes of existing musculotendon models.

2.7.3.2 Joint Motion Models

Joint motion models describe the resultant joint motion caused by the various forces or torques acting upon the system. Most of the reviewed studies employ a joint motion model that is the summation of the joint torques acting on a particular joint [126,129,166,169–172,174–177,180,183]. In general, these models include position-, velocity-, and acceleration-dependent torques making them second-order differential equations. For example, Chadwick *et al.* define their multi-joint motion model as follows [169]:

$$M_m(\theta) \cdot \ddot{\theta} + B(\theta, \dot{\theta}) + C(\theta) \cdot \tau = 0 \tag{2.5}$$

where M_m is a mass matrix, B includes torques caused by centrifugal and coriolis forces, gravity, ligaments, and contact forces, C is a coefficient matrix, and τ is the summation of passive joint torques, muscle torques, conoid ligament torque, and scapulo-thorax contact torque. This model is one of the more thorough of the reviewed devices, accounting for many torque sources. However, some joint motion models simply are a summation of the muscle torques, such as the one used by Rosen *et al.* [157], but are less accurate as they do not account for all of the dynamic properties that contribute to joint motion.

One other joint motion model has been proposed. Katsiaris *et al.* have created a Dynamic Bayesian Network to represent static postures and define transition states between them [181]. The Dynamic Bayesian Network uses probabilities to determine state transitions, where joint angles can only be based on their previous angle or the angles of other joints. The second-order differential equation seems to be the front runner in terms of tried and tested joint motion models. In this model, each dynamic torque source can be accounted for individually and offers the ability to find both forward and inverse solutions. The more components that are added to these joint motion models, the closer the model matches reality, but this comes at a cost of computational resources,

especially when trying to find inverse dynamic solutions to these equations.

2.7.3.3 Bone, Ligament, and Cartilage Models

Increasing the accuracy and adaptability of these motion models comes from modelling more of the dynamic components of the musculoskeletal system. Bone, ligament, and cartilage are currently three of these components that are typically not included into these models. This is likely due to the set of assumptions under which existing motion models are tested. First, bone dynamics are not modelled, since under small loads the effects of their deformations might be neglible in relation to other forces acting on the system. Second, ligaments produce a level of force that creates stability of the joint to move passively but not so great that it impedes the free motion of the joint. Third, cartilage, and the synovial fluid, provide a level of friction that is relatively negligible compared to other forces and, therefore, is treated as essentially frictionless. The limitations in accurate motion estimates of the existing models may be, in part, due to these assumptions, as they do not match reality for most of the scenarios involving natural human motion.

The research community has recognized the limitations of these assumptions and proposed methods to deal with them. Moody *et al.* developed a mathematical motion model that allows for the definition of motion components that have elastic deformation characteristics, such as bone [170]. One issue remaining for the use of bone deformation dynamics is that it is currently not possible to measure the deformations in living humans performing motion tasks. Studies on cadaver bones can be used to determine parameters of the deformation mechanics but these parameters may still vary significantly in live humans. Chadwick *et al.* were the only research group to explicitly provide a ligament model of the conoid ligament for their motion model [169]. Much like the passive force models for tendons, Chadwick modelled the conoid ligament as a nonlinear spring.

Aside from explicit models of bone, ligaments, and cartilage, many researchers account for their forces through lumped force models. The passive forces have been captured through the use of stiffness models [169, 175], damping models [169, 171, 175, 177], and friction models [171, 180]. Stiffness models provide a force that represents the joint's overall resistance to changes in position, while damping and friction models define forces that represent resistance to changes in velocity.

The benefit of using these lumped models is that they are easier to determine than the individual passive forces of the individual soft tissue components. However, the reality is that each of the motion components can vary independently of each other and this is not accounted for in the lumped models. In many cases, determination of the parameters is done through an optimization procedure, as even the measurement of the lumped parameters is difficult or impossible. New techniques and technologies are required to address these limitations, especially when the assumptions stated above are made for healthy individuals, not those suffering from MSDs.

2.7.3.4 Performance of Existing Motion Models

Examining the performance of the existing motion models provides data to enable comparisons. As discussed in Chapter 3 Section 3.4.2, there are four qualities of interest when assessing these motion models. Measuring the accuracy, repeatability, adaptability, and resource demand will help in determining which model is best suited for a given application or set of constraints. Arguably, accuracy is the most important characteristics to measure as it defines the ability of the model to meet its primary objective of producing motion estimates that match the measured motion. Accuracy is the most commonly measured quality across the reviewed models. Eleven of these models provide a measurement of accuracy, usually represented as the error between an estimated and a measured motion parameter [119, 127–129, 131, 144, 157, 166, 172, 173, 178]. Comparing the estimation error across trajectories is typically accomplished by calculating the root-mean-square of the error, e_{RMS}, across the entire motion trajectory, as follow:

$$e_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{x} - x)^2} \tag{2.6}$$

where N is the number of samples, $\hat{x}$ is the estimated value, and x is the measured value. Accuracies of the existing models are represented as mean errors in Table 2.7. During static posture estimation, Fu *et al.* produced the lowest root-mean-square error (RMSE) of 0.0024 m for linear hand position [178]. A RMSE of 4.15° was the lowest recorded value in terms of angular position accuracy for dynamic elbow flexion–extension motions [129]. Ding *et al.* achieved the best angular velocity RMSE of 8.94°/s [172], while the best linear velocity RMSE of 0.014 m/s came from Kiguchi and

Hayashi's work [119]. Kiguchi and Hayashi's model also produced the lowest torque estimation error, ranging from 0.019 Nm for shoulder abduction–adduction to $8\times10^{-}6$ Nm for ulnar–radial deviation of the wrist.

Repeatability measurements provide evidence for the variation in producing desired results over large sets of input data. One measurement of repeatability is the standard deviation of the estimation errors produced by the models. As shown in Table 2.7, the standard deviation was calculated during seven of the reviewed motion model experiments [127,131,144,166,172,173,178]. Song *et al.* produced the best standard deviation in torque error at 0.06 Nm using their ANN [131]. The only angular position and velocity standard deviations were presented by Ding *et al.* at 1.53° and 1.41°/s, respectively [172]. None of the reviewed motion model studies evaluated the adaptability or resource demand of their motion models.

Three of the reviewed studies developed two or more motion models and performed comparative evaluations of their estimation accuracy [127, 157, 173]. Rosen *et al.* developed a Hill-type model and an ANN to compare their ability to estimate elbow joint torque. The results showed average estimation errors of 4.2 Nm and 0.012 Nm for the Hill-type model and ANN, respectively. However, Rosen's experiment only used one subject, making it difficult to say anything conclusive about the results. Clancy *et al.* conducted a comparison of four models developed for estimating the EMG–torque relationship during static postures [127]. Their experiment concluded that the nonlinear polynomial model performed better than either the linear time invariant system or the two models combined with a torque error of 4.65 ± 3.6% of MVC flexion torque at 90° of elbow flexion. However, the authors do not present the numerical results fully. Liu *et al.* developed three different nonlinear polynomial models for providing EMG–torque relationships during constant posture, torque varying contractions [173]. The results show that an angle-invariant model was not statistically significant compared to an angle-specific model, which produced torque estimation errors of 4.06 ± 1.2% and 4.01 ± 1.2% MVC flexion torque at 90° of elbow flexion, respectively. The lack of comparisons leaves many opportunities for comparative studies and exploration of motion model improvements.

Table 2.7: Experimental results of the existing upper limb motion models studies.

Author [Reference]	**Estimation Error Results (Mean ± Standard Deviation (if available))**
Bai [128]	Elbow position: 5.0571°
Baiqing [166]	Simulated elbow muscle forces: 3.871 ± 2.033 N
Clancy [127]	Elbow torque: 4.65 ± 3.6% MVC flexion torque
Ding [172]	Elbow velocity: 8.94 ± 1.41°/s Elbow position: 8.14 ± 1.53°
Fu [178]	Linear hand position: 2.39 ± 2.73 mm
Gao [129]	Elbow position: 4.16°
Kiguchi [119]	Shoulder flexion–extension torque: 0.0002385 Nm Shoulder abduction–adduction torque: 0.01965 Nm Shoulder internal–external rotation torque: 0.006853 Nm Elbow flexion–extension torque: 0.0005253 Nm Forearm pronation–supination torque: 0.0002876 Nm Wrist flexion–extension torque: 0.000008522 Nm Wrist radial–ulnar deviation torque: 0.000008198 Nm Hand velocity: 0.0144 m/s
Liu [173]	Elbow torque: 4.06 ± 1.2% MVC torque at 90° flexion
Peng [144]	Elbow torque: 0.99 ± 0.31 Nm
Rosen [157]	Elbow torque: 0.012 Nm
Song [131]	Elbow torque: 0.35 ± 0.06 Nm

2.8 Controlling Device Behaviour

The fifth fundamental knowledge area is controlling behaviour of wearable assistive devices. Controlling the behaviour of wearable assistive devices involves executing a complex series of tasks, within proper timing constraints, to acheive a desired motion. Vahid and Givargis propose that, in general, a control system "seeks to make a physical system's output track a desired reference input, by setting physical system inputs" [190]. This definition does constitute the major goal for control systems of wearable assistive devices but can be defined more specifically. Further, modern control systems are almost exclusively implemented on digital logic systems, such as binary-based computer systems. To be specific to the scope of this thesis, a *digital control system* is a digital

logic systems designed to make a device's motion output track a desired motion trajectory. In order to meet this goal, digital control systems are responsible for orchestrating a large number of computational tasks that extend far beyond the classical control lens. These control systems are responsible for estimating user motion, communicating internally and externally, commanding actuators, gathering sensor data, processing and transforming data, and ensuring user safety, to name a few major tasks.

The complexities of the digital logic system, the device, the users, and their interactions, as well as determination of the desired motion trajectory, are numerous and dependent on many factors, such as computational resources, actuation technology, muscle health, and the intended application. As a result, sets of control system requirements may vary significantly from one project to another. These facts make it difficult to compare control systems, which has resulted in only a few robust comparisons of control systems found in the literature. Following the general extension of wearable devices from their stationary counterparts (see Section 2.2), control systems of wearable assistive devices have followed from the successes of stationary assistive devices. However, robust control solutions proposed for stationary robotic manipulators may not be suitable to control wearable systems as there are major differences in their characteristics, such as a non-stationary manipulator base, and computational resources. Due to the aforementioned complexities and lack of comparative studies, it is too early to rule out any control strategies simply because they were not designed specifically for a wearable assistive device. Therefore, this review of digital control systems will consider control strategies proposed for all styles of assistive devices. The following sections will discuss major characteristics of digital control systems and safety strategies used by existing control systems.

2.8.1 Digital Control System Characteristics

The number of control system solutions proposed in the literature have sparked other researchers to examine for commonalities between them. Anam and Al-Jumaily decompose active exoskeleton control systems into one of four control major control strategies [191]. Lo and Xie provide a brief summary of control strategies employed by upper limb exoskeletons and note that adaptive control strategies are under-utilized [21]. Maciejasz *et al.* classify control systems of upper limb

rehabilitation devices into 'high-level' control strategies, such as assistive controllers, and 'low-level' control strategies, such as impedance controllers [22]. Proietti *et al.* break control strategies of upper limb robotics exoskeletons for neurorehabilitation into assistive, corrective, and resistive modes, noting that no devices were found that developed resistive mode controllers [192]. Gopura *et al.* provide a brief summary of control systems of upper limb exoskeletons and provide a classification system that involves considering differences in control system inputs, algorithmic architectures, and control system outputs [9].

To date, none of the control system reviews propose a classification scheme that has encompassed all important aspects of these control systems. Such a classification system may not be feasible as the list of important aspects changes between devices and applications. However, based on the control system reviews, the following review will consider the inputs, strategies, outputs, and feedback paths as the characteristics through which the existing control systems will be analyzed. Control system inputs are the signals from the users, the device, or the environment that have been collected, digitized, and transformed for control system usage. Control strategies encompass the major architectures, designs, or algorithms used to generate useful outputs from the control system inputs. The control system outputs are those signals used to orchestrate the interaction between users and devices, such as to control the motion of actuators. Feedback paths are used to correct errors in the control signals and are extremely useful with imperfect models, which is the case for wearable assistive devices. Through comparison of these four characteristics, the landscape of digital control systems will be presented.

2.8.1.1 Control System Input

The first control system characteristics is the control system input. Many of the control strategies, algorithms, parameters, and outputs are dependent on the inputs to the control system. At minimum, the control system requires inputs that enable it to determine the motion intention of the user, the desired motion trajectory, and actuate the device to facilitate a human–machine motion interaction. Considering that wearable assisitive devices are intended to be used in many environments, the control system may also need input data that describe aspects of the environment as well. As a result, control inputs come from three main sources: the users, the device, and the

environment. Each of these input sources provides specific types of data, each of which shapes the control system design and behaviour.

In most scenarios, these devices have two user types, the patient and the therapist, from which to gather data. Patients can provide physiological, anatomical, and cognitive data as control inputs, while therapists provide physiological data. Physiological data inputs from the patient can include electrical signals, such as EMG or EEG signals, motion signals, such as position or force signals, thermal signals, such as skin temperature, and flow signals, such as blood pressure, heart rate [31], and perspiration. For example, many of the existing devices used EMG data [25, 27, 35–37, 44, 46, 52, 60, 64, 68, 69, 76, 81, 82] or EEG data [34, 41, 65] as inputs to their control systems, as the user's motion intention can be identified, at least in part, from these signals. Anatomical measurements, such as height and limb segment lengths, are taken from the patient and used as parameters in control models to achieve better motion estimates. Sugar *et al.* provide methods for determining the combined center of gravity of their device and the user in order to adjust their control model [29]. Desplenter and Trejos configure their biomechanical control model using data that have been scaled to anatomical data collected from study participants [122]. Cognitive inputs may include attention, reaction time, and multi-stimulus processing abilities, which could be useful for adapting control system behaviour and for analysis of therapeutic interventions. However, these data are currently not being employed in these devices due to lack of objective real-time measurements and limited knowledge of the human brain.

Therapists will use these devices as a tool to help their patients and, therefore, will be responsible to program their prescribed motion protocol into the device. In general, this will involve describing motion patterns or choosing from a set of predefined motions. Both Sugar *et al* and Rahman *et al.* developed a control system for their devices that executes pre-defined trajectories, which are chosen and modified by a therapist or clinician [29,193]. These input data will eventually represent the desired motion trajectory for both the patient and the device.

Devices can be modelled more effectively than humans, which lends itself to a smaller set of control inputs. The major behaviour of the device being regulated is its ability to produce motion. Therefore, control inputs are focused around providing data to aid in this regulation task. Commonly, the control inputs gathered from the devices are limited to signals stemming

from sensors on the device, such as encoders sensing a motor's position [25, 30, 46, 62, 63, 70–73]. Common quantities sensed from devices are position [25, 30, 43, 46, 49, 51, 62, 63, 70–73, 75, 77], speed [25, 30, 43, 46, 62, 63, 70–73], pressure [42, 51, 78, 83–87], force [32, 83–85], and torque [52]. The data collected from these sources provide feedback for comparison between the commanded and resultant motion of the device.

Stationary assistive devices exist in highly controlled environments and, typically, do not include systems that gather information about the environment. This is because the ability to control the environment is far less expensive and complex than the addition of these components to devices. However, wearable assistive devices are meant to be used in less controlled environments, such as in the patient's home. Inputs that detail information about the environment could allow devices to adjust their behaviour. For example, control models for fluidic actuators may need to vary with ambient temperature changes or disturbance estimation and rejection techniques may be tuned based on environmental conditions. To date, no wearable devices have been found that use control system inputs stemming solely from the environment. Due to the novelty of these devices, the inclusion of control inputs from the environment currently limited but likely to grow as the devices mature.

2.8.1.2 Control System Strategies

The second control system characteristic is the strategies used for controlling behaviour. The need for accurate and safe control systems propels the study to find new and effective control strategies. Control strategies consist of a series of variables and processes for facilitating human–machine movement tasks. As a result, the literature surrounding control strategies for assistive scenarios begins under the assumption that the motion task has been chosen, although it could be loosely constrained, and that the task is feasible for both the human and the mechatronic systems. This assumption means the control system is then responsible to produce motion, compare the human motion and the commanded motion with the desired motion task, and adjust accordingly to meet the objective. Facilitating this interaction is complex and many proposals for solutions have been generated. These proposed control strategies are diverse and, typically, address many levels of the problem.

Anam and Al-Jumaily proposed a high-level categorization of exoskeleton control systems, which divided the systems into one of either model-based, physical parameter-based, usage-based or hierarchy-based control strategies [191]. Model-based systems use models of the human, the mechatronic system, the environment, or combinations of these to estimate parameters used to control the system. Control systems built upon motion parameters, such as position or force, fall under the physical parameter-based control system category. Usage-based systems are developed with a focus on the specific interaction that needs to be completed. Finally, the hierarchy-based control systems form a three-level hierarchy made up of the task-level, high-level, and low-level controllers, where the task-level is responsible for task completion, the high-level is responsible for computing the properties of the human–machine interaction, and the low-level is responsible for production of the motion. This classification system is the broadest found in the literature. The dissection of alternative classifications into the one made by Anam and Al-Jumaily is presented in Table 2.8.

Table 2.8: A dissection of control systems classifications into the four classes proposed by Anam and Al-Jumaily [191]

Author [Reference]	**Model-based Control**	**Physical Parameter-based Control**	**Usage-based Control**	**Hierarchy-based Control**
Anam [191]	Dynamic Muscle	Position Force Impedance Admittance	Virtual Reality Tele-operation Gait	Task-level High-level Low-level
Gopura [9]	Dynamic Muscle Fuzzy Neuro-fuzzy	Human biological Non-biological Hybrid	ND	ND
Macijasz [22]	ND	Impedance Admittance	ND	High-level Low-level
Proietti [192]	ND	ND	Assistive Corrective Resistive	ND

ND – Not defined.

Although it is productive to have a simple classification system for these control systems, the

reality is that complexity of the desired control behaviour means that most of the control systems fall into all of these classes. As a result, a better comparison can be made if these categories are considered more as characteristics. All control systems use some type of model, and many times more than one, to transform control system inputs into actuator commands. The goal of every one of these systems is to manage the physical motion parameters of the user and the device, such as the position and forces. The usage, or intended application, of the control systems affects many of requirements, design decisions, structures, and properties of the control system. Finally, although a control system may not have been designed with hierarchy in mind, the basic control system has a two-tier hierarchy of control at minimum, where a central controller is managing the behaviour of the rest of the components. There is no doubt that classification of these control systems is difficult and that many more characteristics must be examined to compare between them. Determining a basic set of characteristics and allowing for that to evolve according to certain factors of the devices, user, environments, and control systems is a fruitful method for facilitating more detailed analyses of control system strategies in the future.

2.8.1.3 Control System Feedback

The third control system characteristics is the use of feedback signals within the control strategy. Control systems are commonly developed with feedback loops that help to decrease errors in controlling behaviour. Considering a control sequence as a pipeline of signals, feedback consists of using measurements of one control signal, located further down the pipeline, to influence the values of other control system variables, located prior to the measurement point. When discussion control feedback, it is classified using a binary state as either having or not having feedback loops. Open-loop control denotes a control system that does not use feedback loops, while closed-loop control identifies the presence of at least one feedback loop. The number of feedback loops, the level of loop nesting, and complexity of feedback loop models may vary significantly between different control systems and between signals within one control system. For example, a control system may use an open-loop controller to command the position of one actuator and use closed-loop control to command the position of another actuator within the same device. The choice of feedback structure depends on the requirements of the control scenarios and comes with trade-off

considerations between control system design factors.

Open-loop control systems have the advantage that they are simpler to implement, requiring less computational resources to operate. This advantage makes them ideal for controller prototyping, device characterization, and in situations of well-known system dynamics. However, this type of control structure offers no concurrent error correction method resulting in lower controller accuracies. Due to wearable assistive devices currently being in a general state of exploration, open-loop control systems have been developed for a number of these devices [29,48,49,54–56,74,88]. Open-loop controllers have been used to characterize device properties, such as the open-loop controller Higuma *et al.* used to characterize their actuation system [48]. Sugar *et al.* experienced positive changes in the motor function test scores of stroke patients when using an open-loop controller to control joint torques [29]. Despite this success, Sugar mentions that there exists a severity level of patient impairment that requires closed-loop feedback control to ensure a higher accuracy of interaction.

Achieving a higher level of accuracy can be accomplished using closed-loop control. Closed-loop feedback has been implemented with different types of control strategies. Binary-style controllers use feedback of a control variable to determine whether the device should be in an active state or inactive state of actuation [38,53,75,77]. A sliding-mode controller with feedback from a pressure sensor used to actuate an assistive glove was developed by Polygerinos *et al.* [78]. Biomechanical-based control models use feedback of joint parameters, such as joint position [25,124,125]. Proportional-integral-derivative (PID)-style controllers inherently are closed-loop controllers and are used for low-level actuation control by many existing devices [25, 30, 31, 37, 43, 47, 58, 63, 84, 85, 124, 125]. Cascaded feedback controllers, which contain control loops nested within other control loops, have also been implemented [25, 32, 36]. The commonality of closed-loop control systems is due to the high demand for accurate control interactions, since accuracy is positively correlated with safety. However, an increased accuracy comes at the cost of increased complexity, size, and computational demand. The choice of open-loop or closed-loop control pathways is an important design consideration that is dependent on multiple factors.

2.8.1.4 Control System Output

The fourth control system characteristic is the control system output. The goal of these control systems is to regulate the output of the wearable devices, which is its motion. Motion of the device is described through the position, velocity, acceleration, force, or torque quantities. One or more of these five quantities become the outputs of the control system and are the focus of Anam and Al-Jumaily's physical parameter-based control systems [191]. The position- and force-based outputs are the physical outputs of the actuators, or of the device. However, the outputs of the digital control systems are voltage or current used to drive the actuators. The classification system proposed by Gopura *et al.* supports this notion by splitting the control system outputs between voltage and current, not between the physical motion parameters [9]. Since the function of these control systems is to regulate physical motion parameters, outputs are commonly discussed using the position-based or force-based parameters.

Based on the wearable assistive devices in Table 2.1, the most common control system output is position, with 37% of devices regulating this motion parameter. This is caused by the fact that position of the human body is more easily measurable non-invasively than torque. However, humans are able to modulate forces and joint stiffness, to some extent, at each joint position. It is important that wearable assistive devices either adapt similar multi-parameter control or, at least, account for it within their design. Currently, 10% of the review wearable assistive devices in Table 2.1 control two motion parameters, though not always simultaneously. The choice of control system output relies heavily on the type of actuators used and the parameters of the interaction chosen by the therapist or clinician. A comparison of control system outputs and a list of devices controlling two motion outputs are shown in Table 2.9.

2.8.2 Safety Strategies for Control Systems

Safety of the user is the number one concern during any human–machine interaction. Wearable assistive devices will not be approved for use as regulated medical device, or even for use in clinical testing, if they do not employ techniques to guarantee the safety of the user. From a control system perspective, the following four methods can increase the safety of the system:

Table 2.9: A comparison of the control system outputs across devices listed in Table 2.1. For control systems regulating two motion parameters, check marks indicate which parameters are being controlled.

Number of Devices	**Control System Outputs**				
	Position	**Velocity**	**Acceleration**	**Force**	**Torque**
19	7	1	14	15	
Author [Reference]	**Two-Parameter Control Systems**				
Delph [64]	✓	x	x	✓	x
Kyrylova [25]	✓	✓	x	x	x
Rocon [32]	x	✓	x	x	✓
Sandoval-Gonzalez [80]	✓	x	x	✓	x
Xiao [52]	✓	x	x	x	✓

maximizing accuracy, monitoring sensors, limiting output, and implementing compliance models. Achieving a 100% accurate motion tracking system is extremely difficult, if not impossible, with current knowledge and technologies. One important aspect that affects system response accuracy is overshoot. If a system produces an overshoot that leads to even a small amount beyond the userss position or force limits, the user can experience intense pain and perhaps further damage the injured tissues. As a result, minimizing the errors between human and device motion is crucial to user safety.

Sensor monitoring and output limiting are the most common methods for increasing the safety of rehabilitation systems [24, 42, 43, 45, 47, 54–56, 58, 63, 92–94, 101, 102, 104–106, 194–198]. For example, Aubin *et al.* perform a calibration process to set safety limits for the range of motion of the carpometacarpal and metacarpophalangeal joints of their wearable thumb device [58]. The healthy human body has natural limits for position, velocity, acceleration, and force production and these limits vary with the level of motion impairment. By monitoring relevant sensors, these devices can ensure that their motion is within the capabilities of the user. Output limits can be used to further enhance sensor monitoring. In the case of a sensor failure, an output limit will restrict the device from producing motion that could be dangerous to the user, even if the control system has commanded motion outside of these limits.

The motion compliance inherent in the musculoskeletal system acts as a safety mechanism for interactions between the limb segments and external objects. By modelling this aspect of motion, the device can mimic the same level of compliance, making interactions between the user and device less rigid in nature. Two popular models for implementing compliant motion characteristics are through impedance and admittance control. An impedance control takes a position as input and produces a force output, while the admittance controller receives a force as input and regulates the position [22, 191]. In either case, the input is measured from the user and the output occurs from a device's actuation system. In the simplest approach, the relationship between the position and force can be represented using a linear spring [199]. The spring constant can be modified to adjust the dynamic relationships between these quantities in order to fulfill the requirements for safe motion. Both impedance controllers [24, 93, 105, 164, 196, 197, 200–207] and admittance controllers [32, 69, 139, 143, 198, 208–212] have been proposed for assistive devices, though most of them for stationary exoskeleton-type devices.

2.9 Summary of Fundamental Concepts

Advances in technologies that support wearable assistive devices are continually moving them towards a future where they can help to provide a higher quality of life for those suffering from musculoskeletal disorders. However, one of the major challenges to realizing this potential is the development of digital control systems. Throughout this chapter, the fundamental concepts that pertain to control of these devices and the existing control solutions have been presented. At the highest level, the desired control functionality is that which supports motion interactions between humans and devices. The difficulty in achieving this functionality is due to the complexity of the human, the device, and the interaction. The non-invasive sensing requirements leads to either control systems that are too simplified or are indetermined, both of which require many assumptions that cannot be verified on a constant basis, if at all. Furthermore, even when signals can be captured and transformed to produce motion, the lack of understanding of the human brain means that these signal transformations are simply best guesses. Improvements can be determined through exploration and comparison of existing motion models. Devices come with

their own complexities, such as signal processing pipelines, that add to the control challenge but can be taken apart or isolated to be characterized. Examination of sensing and actuation strategies can help inform control system designs. The large variation in humans and devices creates a need for high adaptability of control systems, for which no solutions currently exist. Controlling the interactions becomes a difficult task when using adaptable control systems. In essence, both the human and the device have adaptable control systems that are trying to adapt to one another. In this scenario, it is difficult to guarantee stability and accuracy of the system, especially since the human has reflexive properties that cannot be actively controlled.

As is the custom in problem solving, the efforts of researchers have been to decompose these complexities into a series of simpler problems. This has led to a large amount of control solutions proposed in the literature, from traditional robotic manipulator solutions to biomechanical models for estimation of motion. However, comparison of these solutions is extremely difficult as the approach, assumptions, level of decomposition, number of components, and many other factors differ between them. To compound this problem further, implementation of a single control system may take weeks or months to complete. Since there are no standards or tools for guiding implementation of these control systems, the task of comparison relies on any one researcher reimplementing each proposed control solution for their unique experimental setup. Development of these standards and tools are the general solution to examining the existing solutions space and identifying which digital control systems properties are best suited to aid humans in motion tasks using wearable assistive devices. The following chapter will propose control system development tools to address these issues and enable comparison of control system components.

Chapter 3

Control System Development and Evaluation Tools

3.1 Introduction

Comparison is a general purpose tool for determining the best solution to a given problem. By comparing various aspects of a control system, one can discover methods for generating ideal solutions. Identifying the differences between two control structures, theories, or methods provides a set of relationships that can be monitored to inform engineering decisions. A robust method for comparison is required to generate the necessary data that inform these decisions. Such methods should consist of a development framework that provides the flexibility to restructure without remorse and processes for evaluating these control alternatives. It is essential that comparison methods are as easy as possible to reconfigure, since the development of control systems is time consuming, and by extension, expensive.

Currently, there are few tools that are capable of aiding with the comparison of control systems developed for wearable assistive devices. However, with the continued growth of the research domains surrounding these devices, more tools are necessary to support the comparison between existing solutions and the exploration of future solutions. Developing these tools will provide the ability to adapt existing solutions to changing requirements, evolve control systems for new uses, and control more experimental variables. The aim of this chapter is to present a series of tools to

aid with the development, evaluation, and comparison of control solutions.

3.2 Existing Control System Comparisons

Comparing control systems is a difficult and tedious task since there are currently no standards for design and implementation. To some extent, this freedom allows control system engineers to develop without constraints, thereby, maximizing the possibility for creativity. However, a lack of basic structure makes the task of decomposing each control system architecture difficult. Furthermore, the uniqueness and variation of each user, device, interaction, and developer means comparison methods must be flexible enough to account for these variations but constrained enough to ensure modularity. Modularity is a quality that describes the level to which the components that constitute a given functionality are broken up into independent modules. Increasing the modularity of a control system makes it easier to understand, manage, and modify. The goal is to have comparison methods that provide a basic structure for arranging and decomposing control system functionality, while minimizing the restrictions on the creative freedoms of control systems developers.

Another reason comparison can be difficult to perform is that the measurements are dependent on the internal structures and components. At a high level of design, it is relatively easy to determine whether a system used an open-loop or closed-loop control strategy. When other characteristics, such as actuation accuracy, are under analysis, the entirety of the control architecture must be examined. In these cases, developers must decompose every alternative solution that they wish to compare, which requires more effort if no basic structure is common amongst the alternatives. As a result, there are very few comparative studies found in the literature surrounding control systems. Rosen *et al.* compared a Hill-based muscle model to a neural network muscle model and found that neural network-based models can achieve higher estimation accuracies but are task dependent and computationally expensive [157]. Clancy *et al.* compared four different models to determine the EMG–torque relationships during constant-posture tasks [127]. Their results suggest increases in performance due to whitened (filtered) EMG signals, increases in data set length, and an increased number of model parameters. Kiguchi and Hayashi developed two neural

networks to compare, one based on EMG data and the other based on EMG and EEG data [119]. In their study, the estimation model built around only EMG data inputs out-performed the one which combined EMG and EEG data inputs. Millard *et al.* performed a simulation study to compare characteristics of three types of Hill-based muscle models [188]. One important finding was that when these models are sub-maximally activated, the mean estimated errors are very similar between models. These results suggest that there may be some motions that the Hill-type muscle model variations could be used interchangably. However, Millard's results were compared to lower body musculature and may not scale perfectly for upper-body muscles. Lobo-Prat *et al.* compared EMG-based and force-based input methods in the control of a stationary elbow device noting that the EMG-based interface was less fatiguing for the subjects, who suffered from Duchenne Muscular Dystrophy [91]. Desplenter and Trejos developed an elbow motion estimation model and used it to compare seven different muscle activation models [122]. The results show that not one muscle activation model out-performs the other; instead there is a trade-off between estimation accuracy and computational expense. These comparison studies provide useful findings, but just scratch the surface of comparing the hundreds of proposed control solutions.

3.3 Control System Development Framework

In order to continue improving the control systems of wearable assistive devices, comparative studies must continue and this drives the need for comparison platforms or tools. The advantage of developing comparison methods and tools is that modifying the existing solutions or adding new solutions to the comparison becomes more straight-forward and can be done more efficiently. One method to facilitate comparisons is through use of software frameworks. Software frameworks allow developers to capture the shared characteristics of systems and formalize their representation both abstractly, in software designs, and concretely, in software implementations. In addition, software frameworks provide modularity and embody the flexibility inherent to software solutions. Furthermore, decomposing software systems into modules or components increases the efficiency of modification and the ability for developers to understand how the system operates. Due to these advantages, a software framework is chosen as a general solution to facilitate the development and

comparison of existing control solutions.

3.3.1 Control System Frameworks

Before designing a new control system framework, the literature was reviewed to determine if any such structure exists and could be used as-is or require improvement. Papers matching combinations of the search terms 'mechatronic', 'robotic', 'control system', and 'framework' were collected from the ACM Digital Library, IEEE Xplore, Scopus, and Google Scholar databases for the review, resulting in 22 articles. Each article was assigned a framework type: behavioural framework, communication framework, control framework, and design framework. Behavioural frameworks are designed for organization of device behaviours. Communication frameworks provide useful templates for facilitating communications between multiple devices or systems. For designs focused on specific control techniques or characteristics, control frameworks provide a base for implementation. Design frameworks facilitate more effective designs of devices and sub-systems. Table 3.1 provides categorization and brief descriptions of frameworks found in this literature survey.

With respect to Table 3.1, behavioural and control frameworks were developed for specific applications and did not provide structures to support all types motion assistance control systems. The communication frameworks, such as Player 2.0 [222] and ROS [223], facilitate communication between systems and sub-systems of devices but do not specifically provide structures at the control system level. From the design frameworks, TORP offers the most complete breakdown of mechatronic systems into computational, electrical, and mechanical components but did not offer structures specific to control [232]. A control software framework could either be decomposed into the computational components or extend this design framework to include control components. The results of the literature survey showed that no candidate framework exists that is specifically designed to enable the development and comparison of control solutions that facilitate motion assistance tasks using wearable assistive devices. Therefore, a new control software framework, the Wearable Mechatronics-Enabled Control System (WearMECS) framework, is proposed.

Table 3.1: Categorization of frameworks relating to the control of mechatronic or robotic systems

Author [Reference]	**Framework Description**
Behavioural Frameworks	
Cherubini [213]	Formalization of human-robot cooperative tasks for safe interaction using end-effector based robots.
Visinsky [214]	Fault detection and tolerance routines framework for robotic systems using dynamic fault tree analysis.
Stamigioli [215]	Analytical framework for the behaviour of robots employing interactive-based control techniques.
Woolley [216]	A software framework that encapsulates robot behaviours for common interfacing and reuse.
Oubbati [217]	Recurrent neural network framework is proposed for solving problems in robot motion control and behaviour generation.
Przystalka [218]	A rapid control prototyping framework for robots used in air duct inspection.
Gianni [219]	Autonomous robot framework for control of stimulus-response and task selection.
Seiger [220]	A programming framework, built upon the Robot Operating System, for development of small domestic service robots.
Woolley [221]	A framework for defining behaviours and using genetic algorithms to determine optimal behaviours of mobile robots.
Communication Frameworks	
Collett [222] (Player 2.0)	A network architecture that allows communication between mobile robotic systems and sub-systems.
Quigley [223] (ROS)	A communication framework for off-board and on-board computers of robotic systems.
Control Frameworks	
Petrič [224]	Stability framework ensures humanoid robots maintain stability during motion control tasks.
Chen [225]	A software framework aimed at re-usable middle-ware applications for mobile service robots.
Artemiadis [226]	A probabilistic framework for detecting muscle fatigue during EMG-driven control of a robotic arm.
Urgulu [227]	Torque estimation framework for control of wearable EMG-driven upper limb exoskeleton.
Peters [228]	A framework for simplifying the design of nonlinear control laws for robotic systems.
Richter [229]	A modeling and control design framework for regenerative robotic systems.

Author [Reference]	Framework Description
Das [230]	Vision-based cooperative control framework for groups of mobile robots.
Albu-Schäffer [231]	Position, torque, and impedance controllers developed from a passivity-based control framework.
Design Frameworks	
Simões [232] (TORP)	A robotic system design framework that decouples computational, electrical, and mechanical designs.
Tekin [233]	A framework and tool kit for automatic generation of control software for mechatronic systems.
Zou [234]	Architecture of a modular mechatronic systems is defined and a framework is developed for reusable software and hardware components within the given architecture.

3.3.2 The WearMECS Framework

The desired framework must fulfill two main requirements: (1) a modular structure must be used such that it enables more effective comparison of control techniques and (2) all categories of control systems for wearable mechatronic systems must be supported. Four categories for control systems of wearable mechatronic systems have been established: model-based, physical-parameter-based, usage-based, and hierarchy-based [191]. Model-based systems use models of the human, the mechatronic system, the environment, or combinations of these entities to estimate parameters used in the control of the system. Control systems built upon parameters, such as position or force, fall under the physical-parameter-based control system category. Usage-based systems are developed based on the specific task that needs to be completed. Finally, the hierarchy-based control systems form a three-level hierarchy, task-level, high-level, and low-level, where the task-level is responsible for task completion, the high-level is responsible for computing the properties of the human–machine interaction, and the low-level is responsible for production of the motion based on physical parameters. As shown in Fig. 3.1, the usage-based, model-based, and physical-parameter-based control systems can be decomposed into the hierarchy-based structure, but, typically, not perfectly into one level.

Proper functional decomposition of control systems would allow them to be split appropriately into the three-tiered hierarchal structure. This is important as it supports the notion that, regardless of control system type, there are common behaviours shared among the control systems.

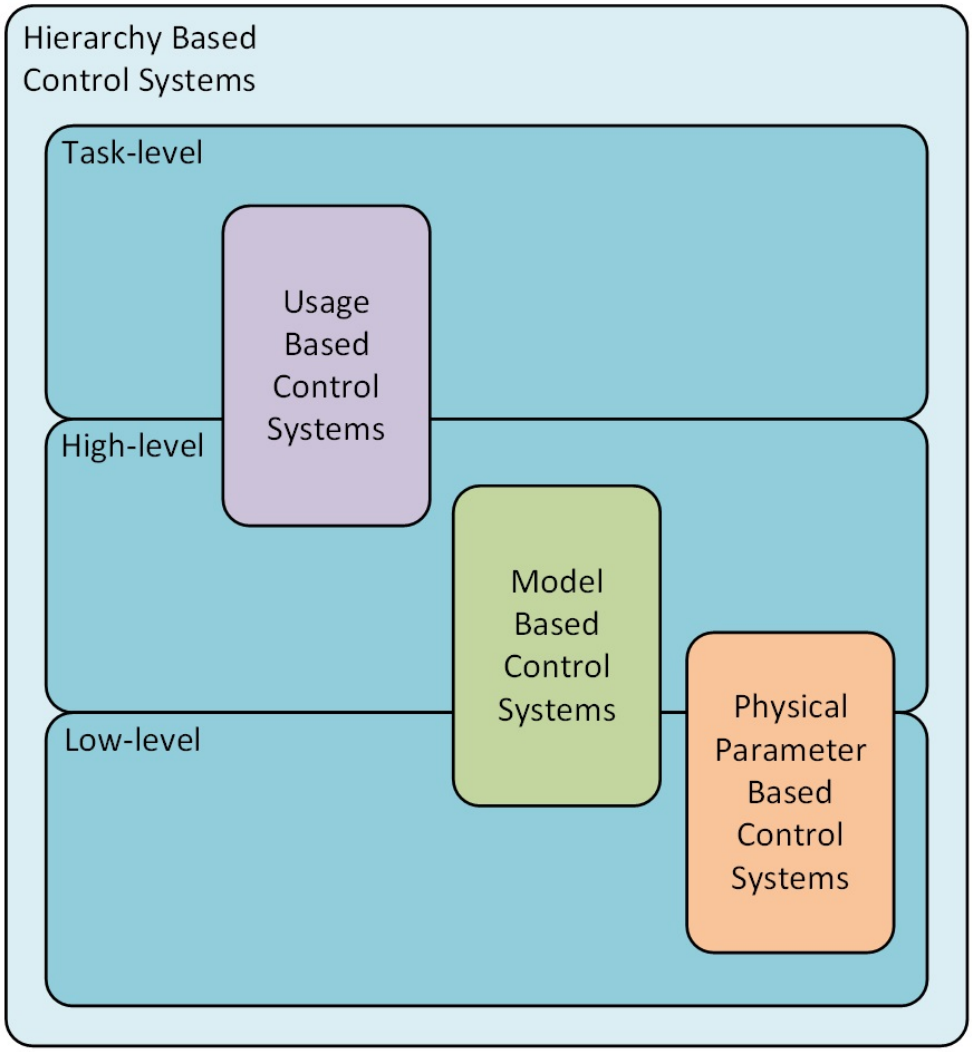


Figure 3.1: Visual representation of Anam and Al-Jumaily's usage-based, model-based, and physical parameter-based control types within the hierarchy-based control type [191].

Although Anam and Al-Jumaily categorize control systems by their focus and structure, every control system that produces the same functionality would share common categories of control behaviour. Therefore, the three-tiered hierarchical control type provides the best structure to represent different levels of control functionality. To reflect the three major control functionalities required to facilitate assistive human–machine interactions, the naming convention was modified from task, high, and low to task, estimation, and actuation, respectively. The task-level comprises control functionality responsible for orchestrating the desired motion task, the estimation-level controls the production of motion estimates required to determine how to control device behaviour,

and the actuation-level facilitates the production of motion through control of the actuation systems. Hierarchical control structures are inherent to both humans and digital control systems, which makes a hierarchical framework desirable. Furthermore, some control systems have been explicitly developed in a hierarchical manner [24, 143, 235]. However, imposing a three-tiered hierarchical structure may needlessly restrict developer freedom. It is important that creative freedom should be maintained. As a result, the framework will only constrain decomposition of functionality into one of the task-level, estimation-level, or actuation-level control functionality groups. Under the assumption that every motion assistance control system must include similar functionality at some level, Fig. 3.2 demonstrates how each of Anam and Al-Jumaily's control types would be composed of these three control functionality groups.

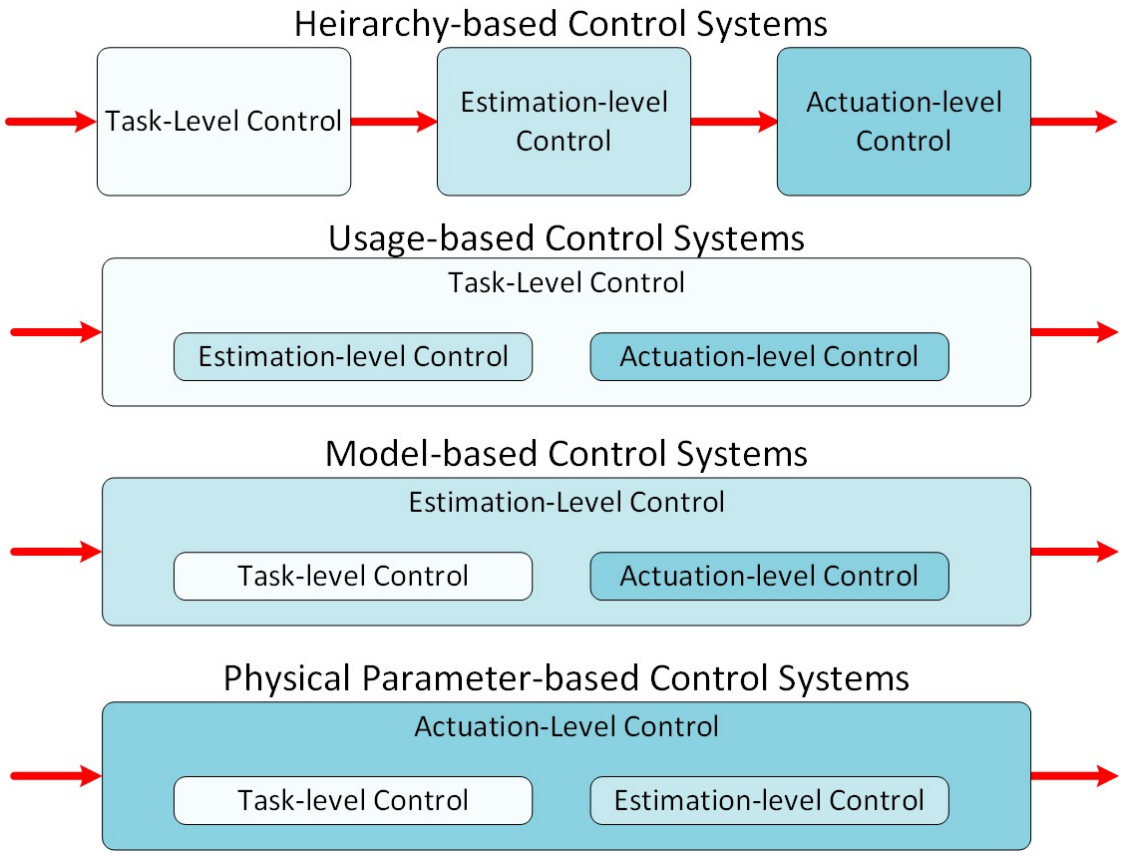


Figure 3.2: Decomposition of control types into the three control functionality groups of the proposed framework. The red arrows show the flow of control through each type of control software.

The framework, presented thus far, is formed abstractly and therefore places no implementation restrictions on control system developers. The abstract framework is useful as a software design tool. However, a concrete framework is a necessary tool to facilitate control system development and comparison. In order to use the framework as a software tool, the following characteristics are required:

- a software library will provide the basic framework structure and allow common functionality to be defined and reused,
- modularization is enabled through the use of object-oriented software principles,
- exchange of data between components should not be limited based on the control functionality group, and
- control should always flow from the human, through the control software, to the mechatronic system.

A software library is the most effective form to capture the framework concepts, provide a basic structure to implement upon, and promote reusability. Furthermore, familiarity with the use of software libraries to software developers means that no new technology is required to use the framework. Adhering to object-oriented principles offers the ability to provide the framework as a template, using interfaces, and encapsulating common functionality into modules that can be swapped without restructuring the entire control system architecture. Allowing data to exchange between components, regardless of their functionality group, leaves an open communication structure for control system engineers. Although the remainder of the research focuses on EMG-driven control, the framework has been designed such that no restriction on input type has been made. In a hierarchy-based control system, the control flows from task-level, through estimation-level, to the actuation-level. In the WearMECS framework, this constraint was removed to allow for other hierarchies and non-hierarchical structures to be developed. The only restriction placed on the framework is that control flows from the human to the mechatronic system, since the intention of the application of wearable assistive devices is for the human to control the device. Data may be given as feedback to the human but the control system should not control the human in assistive

motion tasks. An example of a hierarchy-based control system and where it fits into the interaction between human and device is shown in Fig. 3.3. The WearMECS framework has been designed to meet the requirements and characteristics.

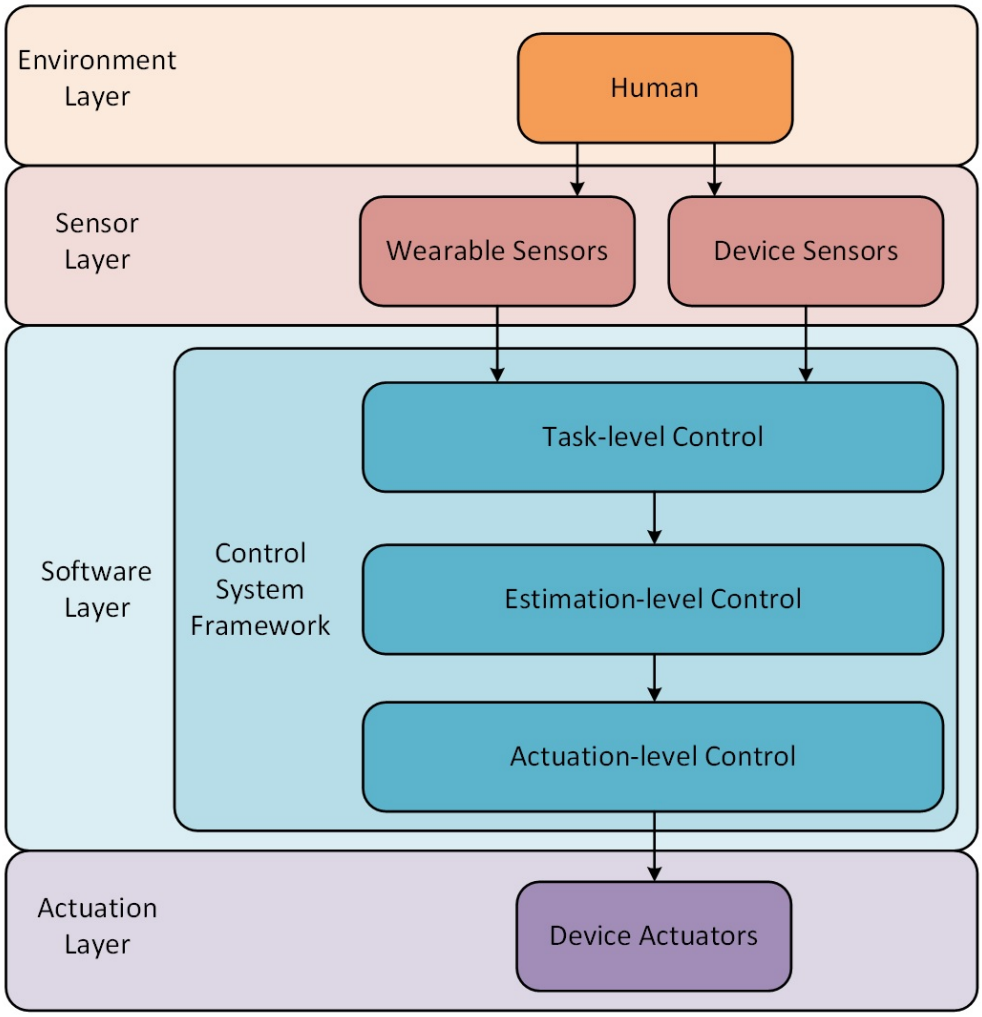


Figure 3.3: The WearMECS framework exists in the software realm to facilitate human control of the wearable assistive devices. Task-level controllers determine the training mode, estimation-level controllers provide estimates of motion parameters, and actuation-level controllers are responsible for commanding motion of the system.

3.3.3 Functional Decomposition Using the WearMECS Framework

The WearMECS framework is a tool to support development and comparison of both software designs and implementations. Control software can be decomposed, by functionality, into the

framework at varying levels of abstraction. At a high-level of abstraction, designs can be decomposed based on their general control strategies. To formalize the designs at a lower-level, formal design languages, such as the Unified Modelling Language [236], can be used to describe the software systems. This flexibility is what drives the use of frameworks as a tool for software development. The abstract formulation and use of the framework will be described in the following sections, while usage of the framework as a software implementation tool will be demonstrated in Chapter 5.

3.3.3.1 Control Strategy Decomposition

The first step to facilitating comparison of control systems is to decompose these systems iteratively, analyzing more details with each iteration. The first iteration should start by decomposing at a control strategy level. As discussed in Chapter 2 Section 2.8, control strategies are general structures or methods for regulating motion behaviour of wearable assistive devices. Using the WearMECS framework, the control system can be decomposed into the control strategies or focus of each functionality group. The task-level strategies are related to the desired motion tasks, the estimation-level strategies are developed to estimate parameters related to the motion tasks, and the actuation-level strategies describe how motion will be produced. Considering wearable assistive elbow devices as an example, Table 3.2 shows a comparison of control strategy decomposition for these devices. Due likely to the novelty of wearable assistive devices and current publication formats, many authors have not included sufficient details even for decomposition at the control strategy level [27, 40, 41, 43, 45, 46]. The devices in Table 3.2 were all developed for rehabilitation or motion assistance scenarios. As a result, the focus of their task-level control strategies are split between production of passive, assistive, or resistive elbow motion tasks. Estimation-level control strategies vary significantly but are all focused on providing estimates of motion properties to guide the motion of the device. Actuation-level control strategies are commonly formed around PID-based controllers, due to their simplicity of implementation and proven record for accurate actuation. Even within this small sample of devices, much diversity can been seen at the control strategy level. To guide developers on the criteria for decomposing controllers, the following sections will present various types of control strategies that fall into each of the functionality

groups.

Table 3.2: Decomposition of control strategies of existing wearable assistive elbow devices.

Author [Reference]	**Task-level Strategy**	**Estimation-level Strategy**	**Actuation-level Strategy**
Benitez [36]	Passive motion tasks	Torque estimation model	PID-based control of a motor
Desplenter [37]	Assistive motion tasks	Velocity estimation model	PID-based control of a motor
Kim [39]	Passive and assistive motion tasks	Force estimation model	PWM of solenoid valves
Kyrylova [25]	Assistive motion tasks	Velocity estimation model	PID control of motors
Pylatiuk [42]	Passive and assistive motion tasks	NL	Proportional control of fluidic actuators
Tang [44]	Assistive motion tasks	Position estimation model	Proportional control of pneumatic actuators

NL - information was not listed by the authors, PID - proportional-integral-derivative, PWM - pulse width modulation.

3.3.3.2 Task-level Control Strategies

Determining the task-level control strategy is a matter of identifying the details of interaction that the control system will facilitate. All of the devices have been developed to provide some form of assistance during motion. However, the type of assistance can vary depending on the intended application, leading to different types of task-level control strategies (see Fig. 3.4). There are two main categories of task-level control strategies: rehabilitation and augmentation. Rehabilitation control strategies consist of musculoskeletal rehabilitation strategies (aimed at restoring tissue properties) and motor control rehabilitation strategies (used to restore effective control of soft tissues). These types of control strategies share the view that it is possible for the individual's motion abilities to improve. Augmentation strategies are used under the assumption that an individual's level of motion ability is permanent or will progressively decrease over time. Assistive

augmentation strategies help maintain higher levels of motor function as the individual's motor function decreases, such as in degenerative muscle diseases. For individuals suffering from spasms, such as tremors, suppressive augmentation strategies are used to suppress the involuntary motion, while allowing the individual to perform the voluntary motion.

For each of the four sub-categories of task-level control strategies, there are a number of different applications related either to existing rehabilitative therapies and augmentation strategies. Both Basteris *et al.* and Proietti *et al.* have performed reviews of motor control rehabilitation (neurorehabilitation) control strategies that employ upper limb assistive devices for stroke patients [192,237]. Basteris decomposes motion control rehabilitation strategies into assistive, active, passive, passive-mirrored, active-assistive, corrective, path guidance, and resistive. However, it is noted that in many cases it is unclear how each of the research groups have implemented these strategies. Proietti decomposes these strategies using a two-tiered classification system, which breaks strategies into assistive, corrective, or resistive modes of neurorehabilitation. It was also found that none of the devices examined in Proietti's review implemented resistive control strategies. Basteris, on the other hand, reviewed a few resistive controllers but noted that no significant improvements to subject motor control abilities were found. This suggests that resistive-based rehabilitation may be more effective when conducted without mechatronic devices. To date, no classification of musculoskeletal rehabilitation strategies can be found in the literature. However, the musculoskeletal and nervous systems cannot be trained in isolation from one another. As a result, many of the motor control rehabilitation strategies are used for musculoskeletal rehabilitation as well.

Assistive augmentation control strategies are used for individuals who will need a greater amount of assistance as their disorder progresses. Currently, the most common application is for those suffering from muscular dystrophy disorders. Control strategies have been proposed for stationary elbow supports [89, 91], planar arm supports [238], and multi-joint arm supports [125,139,239] used to assist with muscular dystrophy. The main type of suppressive augmentation strategies are for tremor suppression. Devices employing these types of control strategies have been developed to suppress tremor at the elbow [32, 35, 38] and wrist [32, 49–51]. To date, no classification systems for either type of augmentation can be found in the literature. As a result,

the possibilities to employ wearable assistive devices have yet to be fully explored. This leaves many research opportunities open for task-level control strategies.

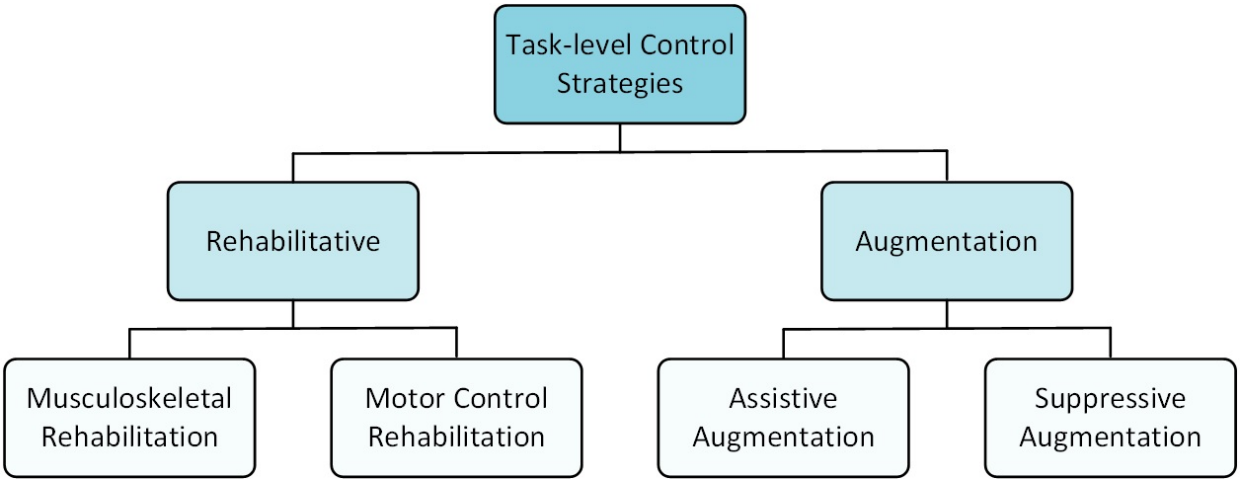


Figure 3.4: Task-level control strategy classification for assistive motion devices.

3.3.3.3 Estimation-level Control Strategies

Estimation-level control strategies depend heavily on the type of input data the developers intend to use in the generation of motion estimates. In general, these strategies are based on the use of either bioelectrical signals, biomechanical signals, or a mixture of both, with the main goal being to provide estimates that are useful for directing actuation systems. These strategies control the interactions between software components that are responsible for estimating properties of the interaction at the current or future time steps. Bioelectrical inputs lend themselves well to control strategies that can predict future motion. Due to the electro-mechanical delay inherent to the musculoskeletal system, the time at which the bioelectrical signals are sensed occurs before the muscles are actuated. Biomechanical signals can be used in a predictive manner as well but only once motion has been produced by the individual. For example, if the system knows the current position and force, a motion model that assumes a constant force between two data points can be used to predict the next position. Mixing these estimation strategies can lead to better estimation quality but comes at the expense of system size and computational resources.

Estimation-level control strategies are centered around one or more models used to produce the

desired estimated motion parameters. Examples of these control system models include neuro-fuzzy models [93, 119], neural network models [131, 144], muscle models [96, 121, 122, 135, 169, 170, 174], polynomial models [127,173], Dynamic Bayesian Network models [181], and musculoskeletal motion models [122, 169, 170, 174, 176]. Some of these models, such as neuro-fuzzy models and neural network models, use only bioelectrical signals to produce estimates of joint torques [144] or hand velocity [119]. Others, such as the one developed by Katsiaris *et al.*, use only biomechanical inputs to predict arm motions [181]. Control strategies using both bioelectrical and biomechanical inputs combine muscle models with a musculoskeletal motion models to predict motion parameters, such as joint torque [122, 135][1].

3.3.3.4 Actuation-level Control Strategies

The actuation-level control strategies fulfill two general functionalities: determining behaviour of the actuation system and commanding the actuator to produce motion. In order to determine the appropriate behaviour of the actuation system, strategies for transforming motion parameters, modelling transmission systems, decision-making, and ensuring safety must be incorporated. The desired actuation behaviour can then be produced using control strategies specific to each type of actuator. Actuation-level controllers often need input from various components of the control software to produce the desired motion. For example, a task-level controller may dictate when the motion should be commanded, estimation-level controllers will supply motion estimates of the user, and sensor drivers will provide data from the actuation system sensors as feedback.

To determine the behaviour of the actuation system, the actuation-level controllers must take the estimated human motion and determine the appropriate motion of the device. This involves the usage of control models of the device and decision algorthms. The actuation-level controllers include both kinematic models and dynamic models [54, 62, 66, 71, 80, 163]. These models allow the actuation-level controllers to determine the relationships between joint motion and actuator motion of each device. Other common controllers use impedance models [24, 93, 105, 164, 196, 197, 200–207] and admittance models [32, 69, 139, 143, 198, 208–212]. Impedance and admittance models allow the motion compliance to be addressed at the control level. Using these control models, the

[1] For more information on estimation-level control strategies, the following reviews are suggested [148, 240–242].

actuation-level control strategies must implement some form of decision algorithm. Estimates of the user motion and the device motion are combined with safety checks and other criteria, such as the device state, to determine what actuation command to send and when to send it. Binary threshold control strategies are one of the most common and simple decision algorithms to be implemented in wearable assistive devices [34, 37, 60, 64, 65, 76, 81, 82, 87]. This style of algorithm determines whether the actuation system is in an active or an inactive state, based on whether certain control parameters exceed pre-set thresholds. Fig. 3.5 shows an example of a decision algorithm for determining the velocity to be commanded of the actuation system.

The final step in actuation-level control strategies is to command the actuation system. This is accomplished through either open-loop or closed-loop controllers. Open-loop controllers are commonly used when motions are pre-programmed or actuator outputs are static [48, 64, 88]. However, the inaccuracy of open-loop controllers means that almost all wearable assistive devices use closed-loop actuation-level control strategies. PID-based control strategies are used almost exclusively to regulate actuator behaviour based on feedback from the actuator sensors [25, 30–32, 36, 37, 43, 47, 52, 56, 58, 63, 84, 85, 124, 125]. The extensive theory and implementation of PID-based controllers makes them an favourite for actuation-level control but other control strategies still need to be explored.

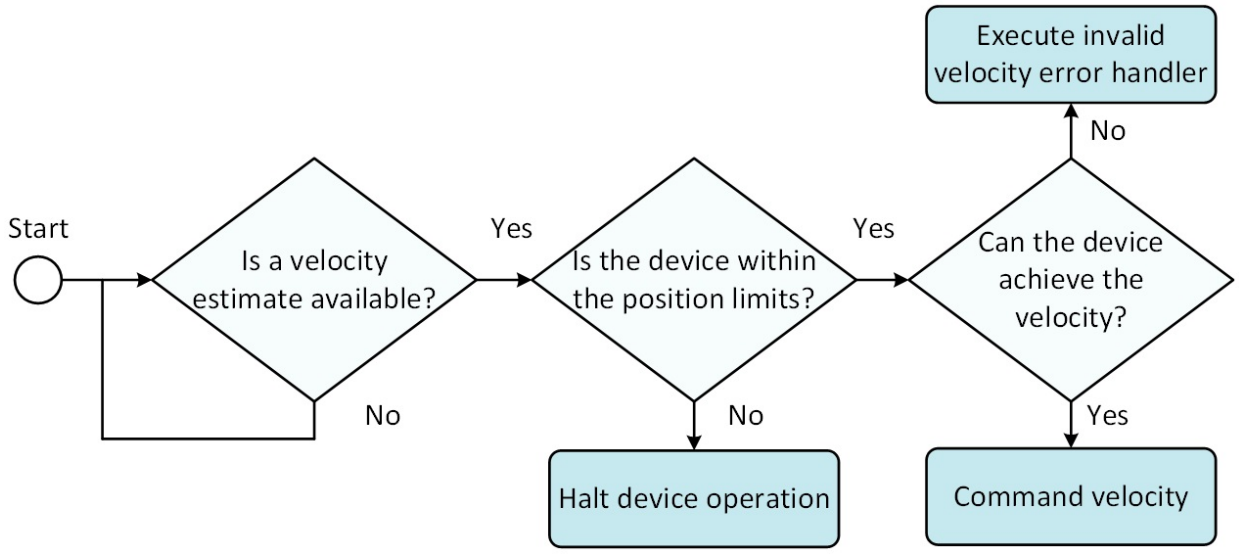


Figure 3.5: Example of an actuation-level decision algorithm.

3.3.4 Formal Design of the WearMECS Framework

Functional decomposition of the control strategies into task-level, estimation-level, and actuation-level functionality categories is only the first step in the design process. The next step is to define the software functions within these groups that will compose each of the desired control functionalities. Thus far, the WearMECS framework has been presented and discussed only as a control strategy design tool. However, the major benefit of this framework comes as a software design and implementation tool. Fig. 3.6 depicts the class diagram of the framework in its most basic form. The framework consist of three classes, where each one is intended to facilitate control of one of the three functionality groups discussed in Section 3.3.3. Each of these classes is developed as a interface, which is a type of class that forces the user to implement each of the inherited methods. In the framework, none of the classes have relationships between each other. This is due partly because of the interface class type, but mainly this was chosen to not constrain developers to specific relationships. The benefit of this choice is that control software developers can have multiple task-level, estimation-level, and actuation-level controllers and are able to define their relationships in ways that meet the requirements of their devices and applications.

Using this framework structure, the requirement to allow comparison of Anam and Al-jumaily's four types of control systems is met, since developers are not restricted by pre-defined class relationships. Furthermore, developers are not required to implement all of the task-level, estimation-level, and actuation-level controllers. For example, if the task-level controller was implemented as a separate software system, the developer need only implement the estimation-level and actuation-level functionalities. However, for each type of controller class in Fig. 3.6, the set of functions listed for that class must be implemented.

The framework facilitates decomposing control into three functional groups but it is important to note that this does not mean that the software design is limited to only three classes. For each functionality groups, software objects (classes) that are vital to fulfill the control requirements of that functionality groups should be added as needed. Fig. 3.7 (A) shows an example of a hierarchical software design for control of a wearable mechatronic elbow brace using the WearMECS framework. In this example, each of the task-level, estimation-level, and actuation-level function-

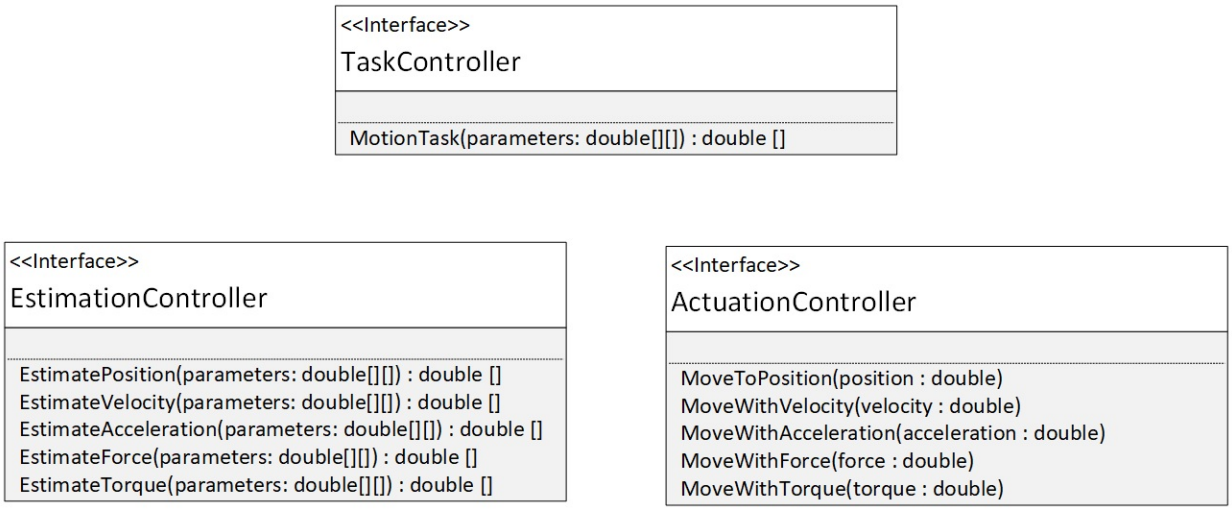


Figure 3.6: The WearMECS framework is founded on a set of three interfaces, one for each of the functionality groups required to facilitate assistive human–machine interactions. It is assumed that the system will have at least one of each of the task-level, estimation-level, and actuation-level controllers, though it is not necessary. There are no relationships established between these interfaces to ensure that the developer has freedom to define object relationships as needed.

alities are broken into multiple software objects. The task-level SensorInput object interacts with sensor drivers to gather control inputs from the human and device. Estimation-level software objects, such as the BiomechanicalModel, are used to determine the user's motion estimates. The MotorOutput class of the actuation-level facilitates the output of the desired commands of the actuation-level controller to the actuation system. An advantage to structuring and designing control software with the WearMECS framework is that modification of the software to compare between two motion estimation models (Fig. 3.7 (B)) or to extend the control system from one device to another (Fig. 3.7 (C)) is completed by swapping components. This reduces the effort required to modify the software and allows developers to constantly build upon their existing control software instead of restructuring entire architectures or starting from scratch. The WearMECS framework has been developed as a software library to allow easy usage for control software developers. Implementation of control systems using the framework is discussed in detail in Chapter 5.

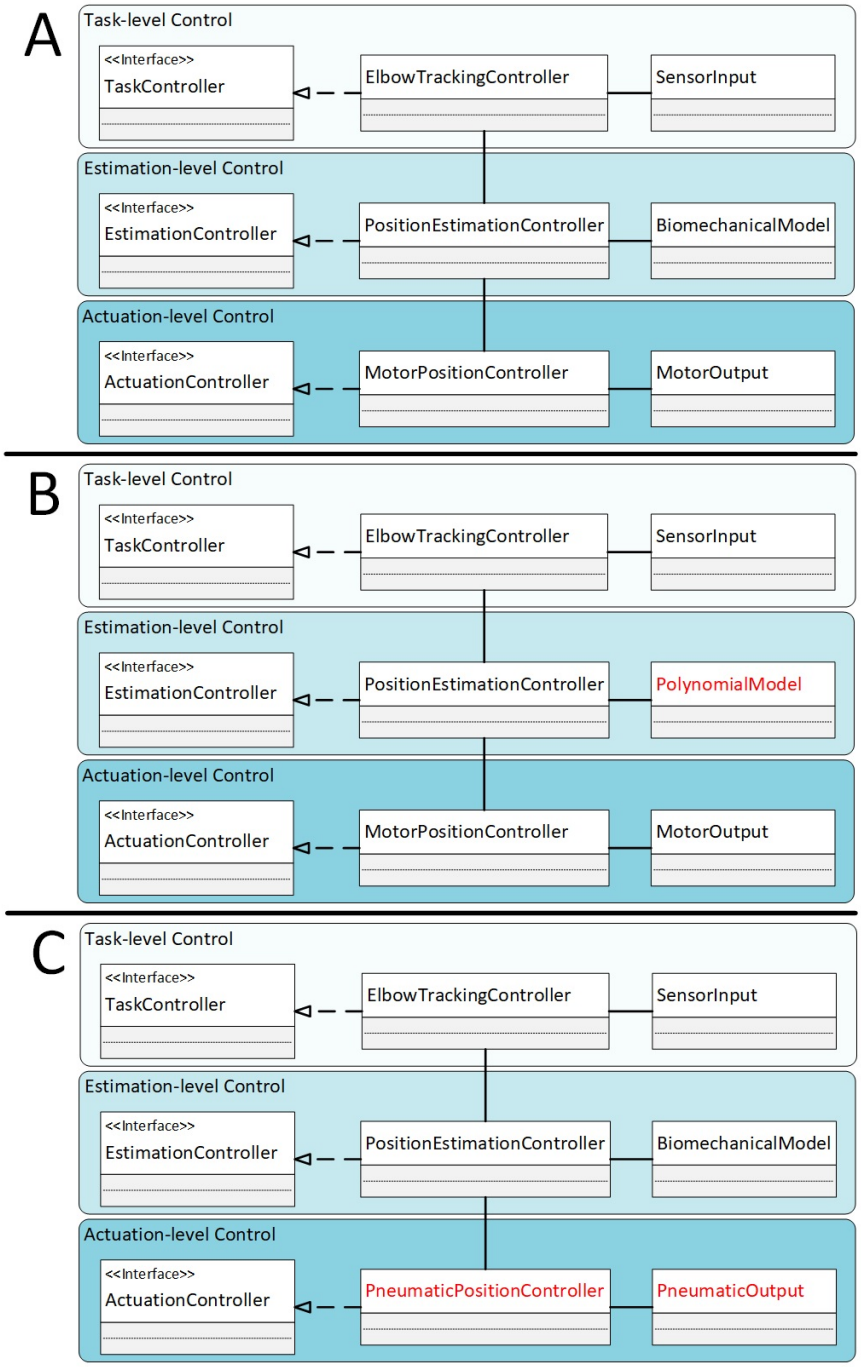


Figure 3.7: Class diagrams depicting the high-level structure of a control software solution (A), modification of that system for a different motion model (B), and modification of that system for a different, pneumatically-driven, device (C). Classes that have been changes between variations of this control software solution are marked with red text.

3.4 Control System Evaluation Tools

Control system performance evaluations are the key to determining the successes and limitations of the existing solutions and finding new avenues to explore. The WearMECS framework provides a platform for decomposing control functionality in such a way that makes modification easier to complete. However, more tools are required to assess different aspects of these functionalities. Tools to help with integrating the humans and devices in a safe and controlled manner, provide an overall assessment of the control systems quality, and produce important quantitative data about the control system performance are especially useful. Three control system evaluation tools have been developed to address these areas. In the following sections, an integration testing process, a control system quality model, and a metric suite are presented.

3.4.1 Human–Machine Integration Testing

The experimental evaluation of the implemented control systems was conducted in a specific multi-phase human–machine integration protocol. To date, no integration testing protocols between humans and wearable assistive devices have been found in the literature. As a result, a new integration testing protocol is proposed. This protocol is broken into four general phases, as shown in Fig. 3.8. First, the software simulation phase involves collecting human motion data and simulating interactions between the control system and the human to determine the validity of the software outputs. Second, in the offline remote control phase, the control system is connected to the device and the device is controlled using motion data that were previously collected. Third, the online remote control phase involves attaching the user to sensors, mounting the device remotely, and allowing the user to directly control the device, under real-time control deadlines. Lastly, the user wears the device to test and refine the human–device interaction in what is denoted the worn phase.

Each of the phases of the integration testing protocol provide a unique opportunity to study various aspects of the control system or human–machine interaction. Control systems developed for the software simulation phase only produce motion estimates or other software metrics. During this phase, motion estimation functionality is isolated from the actuation of the device. The software

simulation phase supports the evaluation and comparison of motion estimation models, which are an important component of these control systems. Offline remote control testing allows for one to study the actuation components and the control response to input signals. The major advantage of this phase is that the input signals are extremely repeatable. This is possible since the signals are being gathered from digital sources instead of being streamed from sensors on the human or device. Much like the offline remote control phase, the online remote control phase ensures the safety of the human and device by remotely locating the device. However, in online remote control testing, the control performance can be studied under real-time control conditions. These constraints often limit the performance and, therefore, it is important to study them before mounting the device onto the user. The worn phase of testing also operates under real-time control conditions. The main difference between the worn phase and the online remote control phase is that the interaction effects between the device and the user are present. Understanding these effects and adapting the control system for them is important to meeting the both the engineering and clinical requirements for these devices.

Based on the reviewed devices, it is shown that there are still many opportunities for integration testing studies. Of the wearable devices that were designed for rehabilitative elbow motion, only five studies involve integration testing of their digital control systems in which quantitative data are presented. Furthermore, regardless of integration testing phases, only Kyrylova and Tang *et al.* provided any quantification regarding the error between the subject and the device motion. Ueda *et al.* developed a muscle-force-based control system and tested it during both software simulation and worn scenarios [33]. They quantified the changes in muscle forces required to hold various loads in a static posture both with and without their device. Kim *et al.* performed worn integration tests to determine the differences in the peak torque during isokinetic movements [39]. Looned *et al.* used healthy subjects to simulate stroke subjects during a worn scenario and measured their time to complete a drinking task, using only EEG signals for control [41]. Kyrylova performed both software simulation and offline remote control phase tests in which the RMSE between the subject and the position estimates and between the subject and the device, respectively, were measured [25]. Average RMSEs ranged from 1.82–3.38° and from 2.65–5.62° for the software simulation and offline remote control scenarios, respectively. Tang *et al.* conducted experiments

during the worn phase of integration testing with their device, achieving an RMSE between the subject and the device position of 9.67° [44].

In the literature, no integration testing of wearable mechatronic elbow devices in the online remote control phase were found. With only two studies measuring the position accuracy of the control system to perform elbow motion tracking tasks and none performed during the online remote control phase, there are a lot of opportunities to investigate. Using the integration testing protocol, these investigations can occur such that there is more control over experimental variables and the safety of the human and the device are maintained. This testing protocol was used during the motion tracking experiments described in Chapter 5.

3.4.2 Control System Quality Model

In order to determine differences between control system components and experimental factors, metrics must be used to quantify the performance of the motion tracking abilities. Metrics allow for the comparison of expected and measured outputs of the components of the control system. In the case of motion models, metrics are computed based on differences between the estimated motion and measured human motion input. The differences between the measured device motion and the measured human motion are calculated to measure the overall tracking performance. The most common metric in the upper limb model literature is accuracy [96, 119, 127–129, 131, 144, 166, 167, 172]. Accuracy is a measurement of the similarity between an estimated motion parameter and a measured elbow motion parameter or between measured motion parameters of the human and device. This metric is the most important for assessment as it determines how well the system can complete its primary objective: to produce motion. However, it is not the only performance metric that needs to be analyzed.

The number and type of assessment metrics may vary depending on multiple factors, including the input, the model, the output, and the application of the model. In most of the control system literature, accuracy is the only metric that is used for assessment. This is likely due to the state of the research surrounding the control of motion tracking. As will be shown in Chapter 5, the level of accuracy of existing motion models still needs further improvement and experimental validation. This would help improve the overall tracking as these devices rely heavily on accurate motion

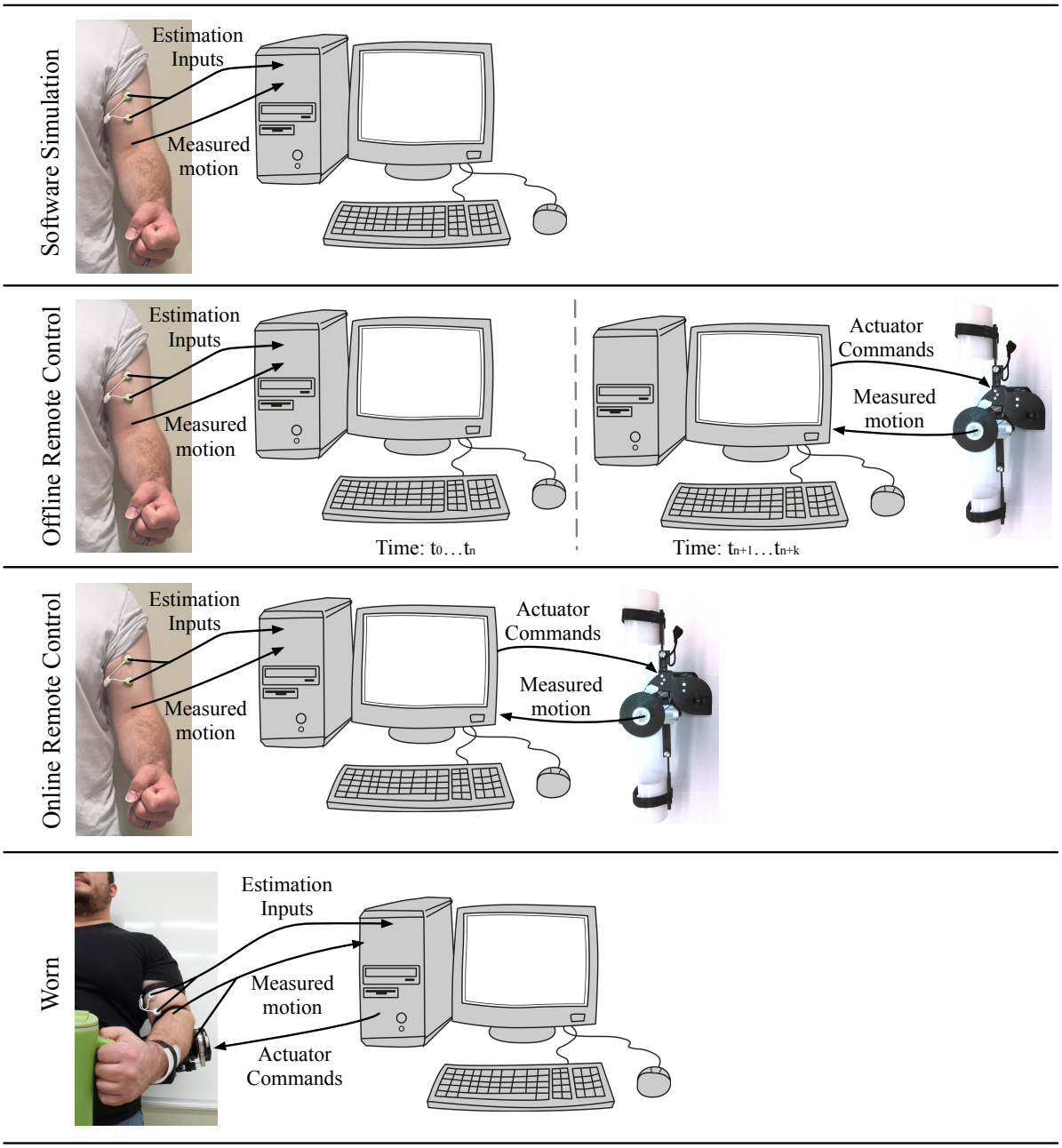


Figure 3.8: Integration testing is divided into the software simulation, offline remote control, online remote control, and worn phases. It is intended that devices traverse this protocol from top to bottom.

estimates. However, the limitations on the accuracy does not mean that other metrics should not be analyzed. Analyzing suites of metrics will allow developers to examine the trade-offs between the metrics and study methods for improving multiple aspects of these control systems.

Alongside accuracy, it is important to analyze the repeatability, adaptability, and resource utilization of motion models. Repeatability defines the degree to which results deviate given the same inputs. Accurate motion must be generated on a repeatable basis in order for the model to be beneficial for human–machine interactions. The degree to which a system can be adapted for variations in input sources is known as the adaptability of the system. Repeatable results are only beneficial if they are adaptable, within a reasonable amount of time, to variations in the human and their motion. The reasonable amount of time depends on the set of circumstances surrounding the use of the control system. The utilization of resources defines the levels of each resource type that are required to adapt the control system to new circumstances or execute the control functionality to produce accurate and repeatable motion.

Accuracy, repeatability, adaptability, and resource utilization constitute the main factors that will be used to analyze these control systems. These factors will provide points of quantitative comparison and evaluation of the control systems. The control systems will be implemented, mainly, as software solutions and therefore a software product quality model is used as a basis for the assessment. Fig. 3.9 presents a software product quality model based on the ISO 25010 software product quality standard [18]. The next step in assessing the quality of the control software is to determine metrics that will be used to represent each of the four quality factors.

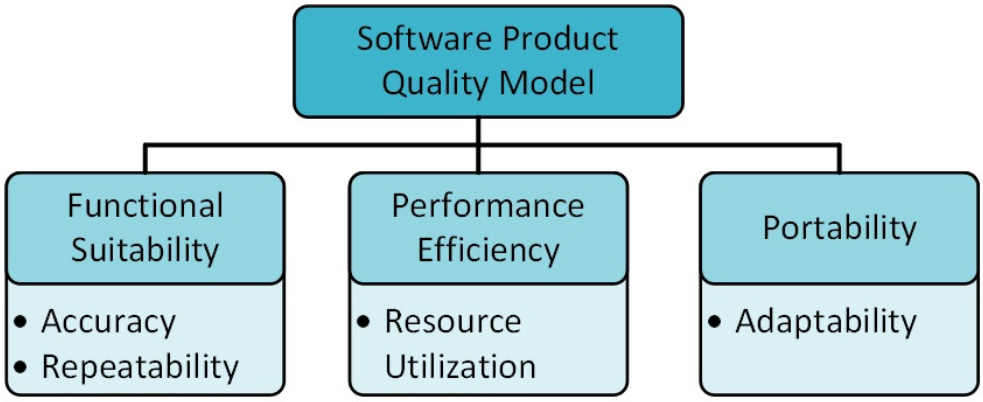


Figure 3.9: A quality model for the determining the quality of the motion tracking control software.

3.4.3 Control System Metrics

The lack of quantitative performance evaluations of control systems and components in the literature creates many opportunities to further the study of these solutions. As discussed in the previous section, the accuracy of the control behaviour is the most commonly quantified control characteristic. Commonly, the accuracy is quantified using the error between an expected value and measured value. In order to assess the correctness of the control functionality and compare with existing studies, these types of errors must be measured. In most cases, the important errors to measure are those between desired motion and the motion generated by the device. However, intermediary quantities such as the error between the desired motion and estimated motion are also of useful to measure.

It is important that the number and type of measured quantities evolves in order to begin exploring other factors that affect control performance. The control system quality model presented in Section 3.4.2 is a step in this evolution. However, the quality of the system needs to be determine quantitatively using more metrics. Repeatability is important to understand the variation in control behaviour but has not been proposed as an important control characteristic in the literature.

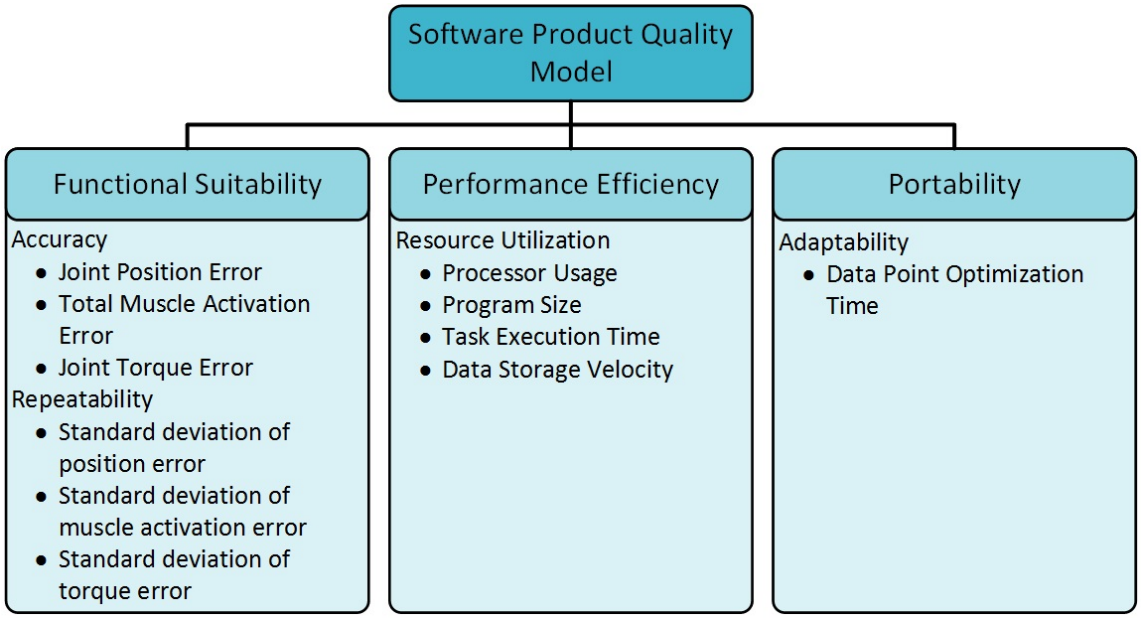


Figure 3.10: Depiction of the software product quality model used to describe the quality of the control software developed in this thesis.

One possible method for representing repeatability is through the standard deviation of a series of measurements. Standard deviations of motion model estimation errors have been measured in the literature, which makes it a good candidate as a repeatability metric [127,131,144,166,172,173,178].

In the context of these control systems, adaptability refers to the amount of effort it takes to adapt control components between user, applications, or other sets of criteria. One way to determine this effort is to measure the time. Each of the control solutions has one or more parameters that must be adapted to each user or for each motion task. The process of determining these parameters is typically completed through mathematical optimization [96, 122, 169, 178]. Measuring the amount of time it takes to perform these optimizations provides a basic insight into the adaptability of the solution. Although time is important, it is not the only resource being consumed to optimize the models or perform control tasks. Control systems for wearable assistive devices operate on digital computer systems and, therefore, require computational resources to execute these tasks. Currently, there are no studies on computational resources required to provide motion assistance with wearable assistive devices. To address this, resource utilization metrics have been developed to quantify computational resources during execution of control tasks. A metric suite containing 11 metrics has been developed to quantify the performance of the control solutions developed in this thesis. The description and mathematical definitions of each of the metrics are listed in Table 3.3.

3.5 Summary

Determining an ideal control solution for assistive scenarios using wearable devices is a large and difficult task. The complexity and variation of the users, the devices, and the applications, coupled with the limited knowledge of human motion, have created a large amount of control system research. Very few comparisons are available in the literature, leaving many unknowns for researchers and developers. Comparison of existing control solutions will help to identify the current opportunities and challenges of these solutions. However, comparison is extremely expensive given that no platforms or tools have been developed to assist with this task.

As a result, a series of control system development and evaluation tools are proposed to aid

Table 3.3: Mathematical definitions of the metrics used in the evaluation of the control systems.

Metric	**Description**	**Equation**
	Accuracy Metrics	
Joint position error	The root-mean-square of the difference between the estimated joint position (θ_e) and the measured joint position (θ_m). N is the number of samples.	$e_\theta = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\theta_{e_i} - \theta_{m_i)^2}}$
Total muscle activation error	The root-mean-square of the squared difference between the forward-derived muscle activation (a_{TF}) and the inversely-derived muscle activation (a_{TI}).	$e_a = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left((a_{TI_i} - a_{TF_i})^2\right)^2}$
Joint torque error	The root-mean-square of the difference between the estimated joint torque (T_e) and the derived joint torque (T_d).	$e_T = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(T_{e_i} - T_{d_i)^2}}$
	Repeatability Metrics	
Standard deviation of position error	Standard deviation of joint position error (θ). $\bar{\theta}$ is the mean joint position.	$\sigma_{e_\theta} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(e_{\theta_i} - \bar{e_\theta})^2}$
Standard deviation of muscle activation error	Standard deviation of muscle activation (a). $\bar{a}$ is the mean muscle activation.	$\sigma_{e_a} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(a_i - \bar{a})^2}$
Standard deviation of torque error	Standard deviation of joint torque (T). $\bar{T}$ is the mean joint torque.	$\sigma_{e_T} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(T_i - \bar{T})^2}$

Adaptability Metrics		
Data point optimization time	Amount of time required to perform an optimization task (t_{opt}) divided by the number of data points (N_{dp}). This metric allows for comparison given different data set lengths.	$t_{dp} = \frac{t_{opt}}{N_{dp}}$
Resource Utilization Metrics		
Processor usage	The ratio of the time spent executing a program ($t_{program}$) to the total available processing time (t_{total}) expressed as a percentage. In multi-core processors, the summation of the percentage used by each processing core (i) is taken. N_p is the number of processing cores.	$P_u = \sum_{i=1}^{N_p} \left(\frac{t_{program_i}}{t_{total_i}} \cdot 100\% \right)$
Program size	The summation of the number of bytes (b_i) per instruction (k) and bytes (b_d) per data value (j) of computer memory that a program requires to execute its behaviour. I and D are the total number of instructions and data values, respectively.	$S_{program} = \sum_{k=1}^{I} b_{i_k} + \sum_{j=1}^{D} b_{d_j}$
Task execution time	The duration of time between the beginning (t_{start}) and the end (t_{end}) of the execution of a control system task.	$t_e = t_{end} - t_{start}$
Data storage velocity	The number of bytes of data (N_b) that are generated by a control system task and need to be stored. t_e is the amount of time taken to execute the task.	$‘ = \frac{N_b}{t_e}$

with this problem. The WearMECS framework has been developed as a tool for both design and implementation of control software. The framework uses object-oriented software design principles to facilitate modularization of functionality and promotes developers to build upon their existing solutions instead of starting over. The WearMECS framework is the first tool developed to aid in design and implementation of control software for wearable assistive devices. Three evaluation tools, the human–machine integration testing protocol, the control system quality model, and the metric suite, provide methods for evaluating these control systems more effectively. The integration testing protocol provides a platform for conducting motion experiments with wearable devices, such that characteristics of the system can be studied in isolation and the safety of the users and the devices is promoted. The control system quality model formalizes system quality as the aggregation of four important control characteristics. Using this model, alongside the metric suite, control system developers can quantify various aspects of the system and verify that quality of the system meets the requirements. Together, these development and evaluation tools are the contributions of this chapter and can be used to aid with testing and comparison of control systems.

Chapter 4

Design and Implementation of Control Solutions

4.1 Introduction

Assistive rehabilitation tasks are defined as those in which the patient cannot complete the desired motion voluntarily and requires assistance from the therapist. If a wearable assistive device is designed appropriately, it can facilitate these tasks by supplying assistive forces equivalent to those of the therapist. Active-assistive motion tasks involving an assistive device are tasks in which the device is actively tracking the user's motions and intentions and supplying assistance only as needed [237]. This type of rehabilitative motion task forms the focus of many existing assistive devices, due to the potential benefits of reducing therapist labour [63]. However, it is extremely difficult to achieve since the human body is much more capable of detecting when assistance is needed and adjusting to maintain a smooth motion trajectory than a mechatronic device.

The elbow joint is the second-most common upper limb joint for which wearable assistive devices have been developed (see Chapter 2 Section 2.2.2). This is due to many factors, such as possibilities for larger and heavier devices, highly available EMG signals from primary flexor and extensor muscles, and one active degree-of-freedom of motion. The research presented here focuses exclusively on elbow joint motion and its usage for control of assistive elbow devices. In particular,

this chapter will describe the design and implementation of control systems whose purpose is to track human elbow joint motion. The functionality of these control systems is decomposed into the three major functionality groups of the WearMECS framework. After the designs are shown and discussed, the aggregation of these components into control software implementations is presented.

4.2 Task-level Control System Design

The goal of active-assistive motion tasks is to provide a minimum amount of resistance to complete the task, forcing the patient's tissues and motor control systems to adapt to the demand without regressing in their function. Early in the rehabilitation process, the patient may not be able to complete any portion of the motion. This requires the therapist to supply the entire force trajectory of the desired motion task. The therapist will decrease the level of assistance as the patient regains strength and control of their elbow. It is important to note that the degree of assistance (force) will vary with many factors including the position within the motion trajectory, the tightness of the patient's tissues, the level of fatigue, and the patient's mental focus. During active-assistive therapies, the patient and therapist will work together in a complex interaction to complete the prescribed elbow motions. The elbow joint contains one active degree-of-freedom and facilitates elbow flexion and elbow extension movements. Fig. 4.1 provides an example of active-assistive elbow flexion–extension in both a seated and standing position. The therapist will read physical cues from the patients body, converse with the patient, and sense the motion through their own body to gather information that allows them to adjust the level of assistance. Active-assistive interactions are essentially a simultaneous optimization of multiple objective functions. The therapist is trying to minimize their own effort, maximize patient effort, minimize error in the movement, minimize patient discomfort, and maximize task completion, in order to move synchronously with the patient.

Providing active-assistive therapies is accomplished through two general activities: tracking and disturbing. The therapist will track the motion of the patient's elbow using their hands. At any point that the patient cannot produce the desired motion, they will compensate with disturbance forces. To complete these two tasks, the human body is required to simultaneously

Figure 4.1: The patient and the therapist participate together to achieve a desired motion trajectory, such that the therapists only assists as needed. These interactions may occur with the patient in various postures, such as seated (left) or standing (right).

regulate position and force outputs. The system-level view of the therapist control scheme is shown in Fig. 4.2 (A). The therapist controls the muscles to actuate movement of the limbs, such that the position and force requirements for the task are met. Active-assistive motion tasks can be replicated with the appropriate device and control system. In this case, the control system regulates the actuation system to produce movement of the mechanical segments of the device (Fig. 4.2 B).

Replicating therapists' motion abilities is the primary objective for these wearable assistive devices. However, the existing knowledge regarding production and regulation of human motion is too limited to provide control solutions that are as accurate and flexible as the therapist. No studies involving the quantification of patient–therapist interactions during active-assistive elbow motion tasks were found in the literature. This means that no targets are available for the research community to aim for in terms of tracking or disturbing. Wearable assistive devices could be a potential solution to quantify these interactions, as discussed in Chapter 6. Due to the lack of requirements, developers have generally assumed that there is a need to work toward perfect tracking abilities (zero error) until the research community can determine these targets. The difficulty of determining these targets comes from the facts that measurement is difficult, each

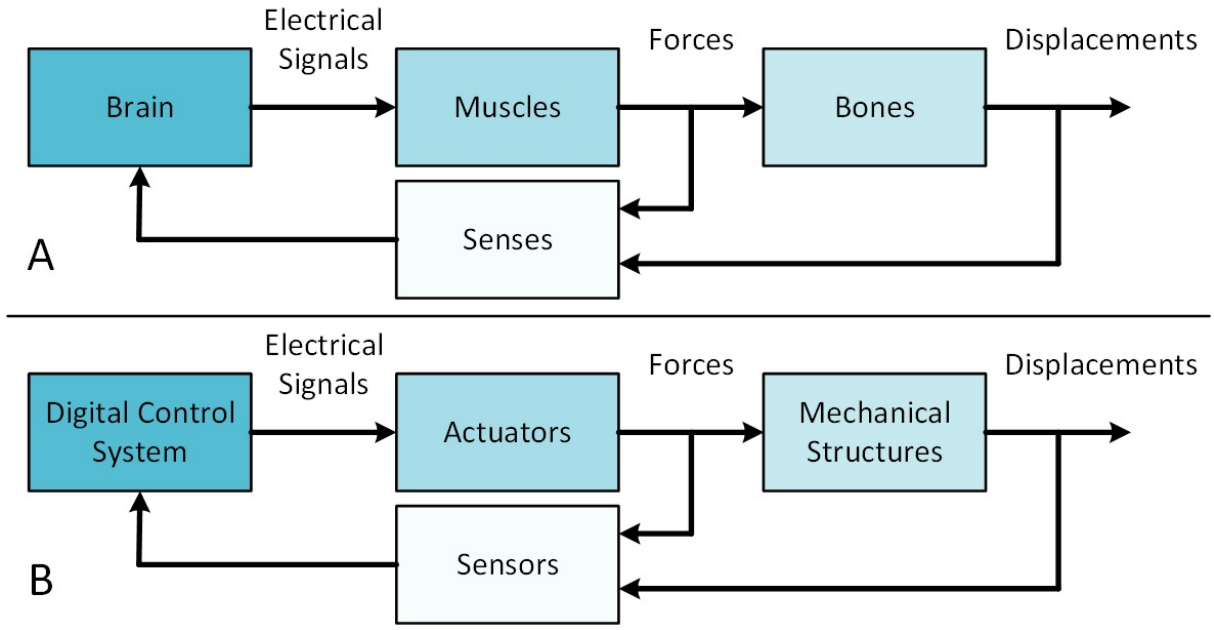


Figure 4.2: A system-level comparison between the therapist and the wearable assistive device. The therapist regulates forces of the muscles to produce displacements between bone segments. This can be mimicked with a wearable assistive device, which uses the actuators to produce displacements between the mechanical structures.

patient and their case is different, motion is highly variable, and assessment is, in part, subjective. Once accurate and flexible motion tracking is achieved, wearable assistive devices can be easily made to produce disturbance forces as desired. However, this will require large measurement studies of the forces applied by therapists to patients and the relationships between those forces and aspects of the patient, the therapist, the environment, and the interaction.

From the control perspective, the assumption of an ideal actuator means that the problem can be decomposed further into one of accurate and flexible motion estimation. Considering only tracking abilities, these models must estimate the motion of the patient, so that the device's motion and the patient's motion can be synchronized. Detailed development, analysis, and discussion of these motion models is presented in Chapter 5. The remained of this thesis focus on improving motion tracking ability of wearable mechatronic elbow devices, leaving quantification of disturbance forces as future work.

4.2.1 Elbow Motion Tracking Task Controller

With the scope narrowed to elbow motion tracking, a task-level controller can be designed. The goal of this controller is to orchestrate the rest of the control system to generate motion of the device that tracks human elbow motion input as accurately as possible. The motion tracking task controller is developed as a single control loop. There are four general tasks that the motion tracking task controller must complete with each iteration of the loop, as shown in Fig. 4.3. This controller facilitates the gathering of data recorded from the user, parsing of the data into appropriate formats, estimation of the user's desired motion, and commanding of the actuation system to produce motion. It is not necessary that all of these functionalities are implemented within this controller or even at the task-level of the control system. The only requirement is that the motion tracking task controller ensures that each of these major functionalities are executed in the appropriate order and at the appropriate times. However, the gathering and formatting of user data are good candidates to be implemented into the task-level control, since the motion tracking task controller will need to communicate the formatted data to other control system components. For the developed control systems, estimation of human motion is implemented at the estimation-level and commanding of device motion is implemented at the actuation-level. With the order of operations established, the next important step in the design of this controller is to determine the control loop timing.

4.2.2 Control Loop Timing

The response time of the control system should occur at a similar rate to that of the natural motion response time of the human motor control system. The EMD of the musculoskeletal system is the parameter that determine the amount of time between activation of the musculature and the resultant motion of the joint (see Chapter 2 Section D.0.2.2). It was found that EMDs for elbow flexor and extensor muscle can be as small as 25 ms and that humans start to sense a delay in a system if it is larger than 300 ms [44, 243]. From the upper limb motion model literature, researchers use elbow flexion and extensor activation time constants that are in the range of 5–40 ms [244, 245]. Based on these studies, it is clear that the EMD is variable but can be matched if

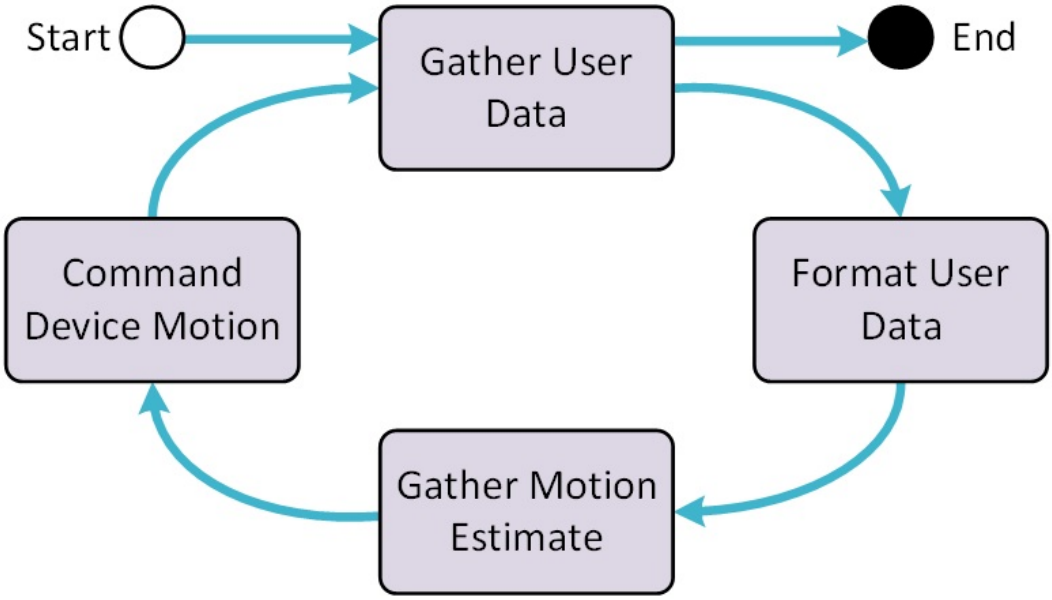


Figure 4.3: The motion tracking task controller must interact with the rest of the control system components in order to complete four basic tasks, which include gathering user data, formatting the data, collecting motion estimates, and commanding motion of the actuation system. This controller is developed as a single control loop that iteratively completes these processes as long as data are available and motion of the device is desired.

the task control loop is executed every 5–45 ms. Estimating and producing motion at a faster rate than this would cause the system to move toward the target before the human system attempts to do so.

Tang *et al.* observed that EMDs of less than 300 ms were undetectable. As a result, it is possible to execute motion commands at a rate slower than that of the human EMD and not be detected by the device user. This means that the control loop execution deadline must be between 5 and 300 ms. During early testing of the control hardware, the response delays ranged from 50–200 ms. Therefore, the control loop execution deadline was set for every 250 ms. This resulted in a motion command being generated at a frequency of 4 Hz, which was significantly higher than the frequency of the motions that were completed during testing. Based on improvements to the control system, the actuation system hardware response time improved to an average of 6.76 $\pm$ 0.90 ms for sending motion commands. However, the tracking loop execution deadline remained at 250 ms. This decision was made to keep consistency in the control command frequency across

the variations of the control systems developed in this thesis.

4.3 Estimation-level Control System Design

The elbow joint is the most common upper limb joint that is modelled in the literature. It is also one of the most common joints for which wearable assistive devices are developed (Chapter 2 Table 2.1). As a result, the focus of this research is on the development and evaluation of elbow motion models. Four different elbow motion models were developed for characterizing elbow motion, including a Kalman filter motion model (KFMM), a Hill-type motion model (HTMM), a proportional motion model (PMM), and a nonlinear polynomial motion model (NPMM). These models were developed to represent the spectrum of the motion models that are proposed in the literature. Fig. 4.4 shows a comparison of the dynamic components that are modelled explicitly by each of these motion models. The KFMM and the HTMM represent the least and most decomposed in terms of modelled dynamics, respectively. The PMM and the NPMM are decomposed within the generalized motion model to the same extent, but they each model the muscle contraction dynamics differently, as either a proportional constant or a nonlinear polynomial function, respectively. The following sections describe the mathematical details of these four elbow motion models.

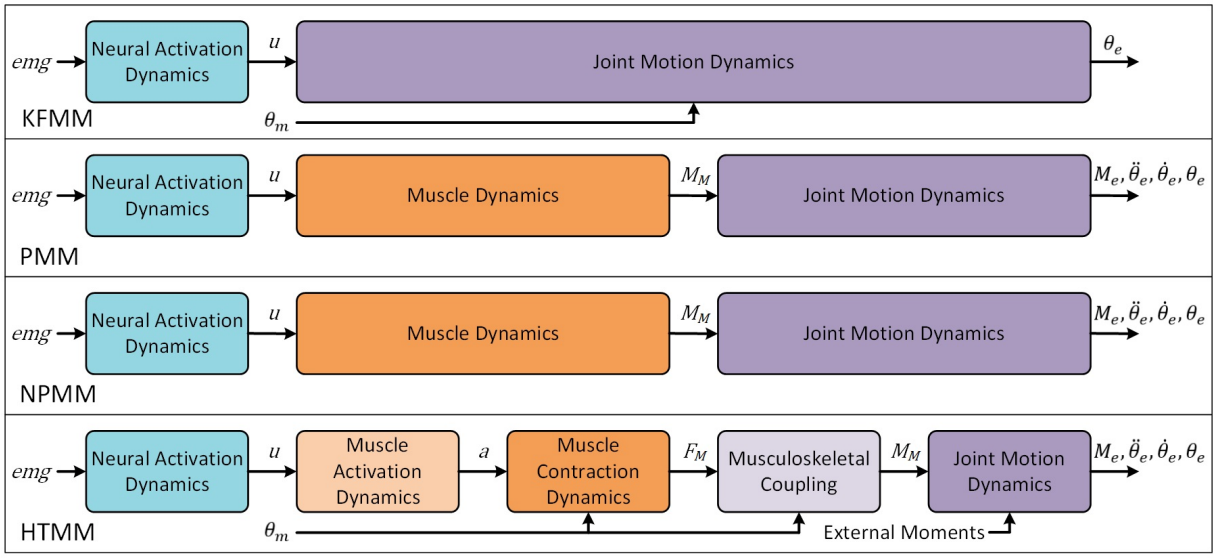


Figure 4.4: A comparison of the developed motion models in terms of the generalized motion model components that were implemented.

4.3.1 Kalman Filter Motion Model

Kyrylova *et al.* first proposed the KFMM as an alternative solution to the existing elbow motion models. The mean position estimation error of the original KFMM (1.82°–3.38) was lower than any other existing model that reported elbow position error measurements. As a result, variations of this model have been implemented and it has been extensively used throughout this work. The muscle activation, muscle contraction, and joint motion dynamics are fully aggregated within the KFMM. The KFMM provides an estimation of elbow joint angle based on the history of the joint position or torque and the EMG signals from the biceps brachii and triceps brachii muscles. Fig. 4.4 (KFMM) presents the abstract representation of the KFMM and the transformation of data types between components. After EMG signals have been filtered, rectified, and normalized to the MVC of the user, the KFMM employs a neural activation model identical to Eq. 2.2, described in Section 2.7.3.1.1, for each of the biceps, u_{BB}, and triceps, u_{TB}, signals. Next, the total neural activation, u_T, is calculated as the difference between the biceps and triceps neural activation signals amplified by a gain, G_{na}, as follows:

$$u_T = G_{na}(u_{BB} - u_{TB}) \tag{4.1}$$

The final component of the KFMM is a simplified Kalman filter, which is decomposed into a prediction function and four correction functions. The predicted joint angle, θ_p, is determined as a linear combination of the total neural activation from the previous time step and the joint angle, θ, from two time steps prior, as follows:

$$\theta_p(t) = \frac{u_T(t-1) + \theta(t-2)}{2} \tag{4.2}$$

where t is time. This Kalman filter has two parameters, Q and R, that determine the error correction and represent the measurement noise and process noise, respectively. The correction model is made up of four equations that use these parameters to reduce the error in the signal, as follows:

$$P(t) = P'(t-1) + Q \tag{4.3}$$

$$G(t) = \frac{P(t)}{(R + P(t))} \tag{4.4}$$

$$P'(t) = P(t)(1 - G(t)) \tag{4.5}$$

$$\theta_c(t) = \theta_p(t) + G(t)(u_T(t) - \theta_p(t)) \tag{4.6}$$

where $G(t)$, $P(t)$, and $P'(t)$ are functions of noise and θ_c is the corrected joint angle estimate. In the original model, the output of the corrective functions was the estimated joint torque, not the estimated joint position, and was used to derive the estimated joint position. After the large errors produced by this version of the model and further analysis, the torque model was removed and the output of the corrective functions is as shown in Eq. 4.6. The KFMMs developed in this thesis are variations of the work by Kyrylova *et al.* [25, 123]. Two studies using these KFMM variations have been published in the literature [124, 125].

4.3.2 Hill-Type Motion Model

The HTMM was developed in order to decompose the dynamics of elbow motion to match that of those motion models found in the literature. The mathematical modelling of the HTMM model was influenced heavily by the work of Buchanan *et al.* and Chadwick *et al.* [121,169]. In this model, a neural activation, muscle activation, musculotendon contraction, and joint motion dynamics are all explicitly defined as separate components. Fig. 4.5 depicts the components of the HTMM and the data flow of the system. In general, the EMG signals of the biceps brachii and triceps brachii muscles and the joint position are measured from the subject and used to determine the resultant joint moment. In addition to the muscle torque dynamics, the joint motion model includes gravitational torques and passive joint torque. The following sections describe the components of the model in more detail.

4.3.2.1 Neural Activation and Muscle Activation Models

Both a neural activation and several muscle activation models were developed as part of the HTMM. The neural activation model used in the HTMM was developed as a low-pass filter that was applied to the processed EMG signals. Cut-off frequencies for these filters range from 2–10 Hz

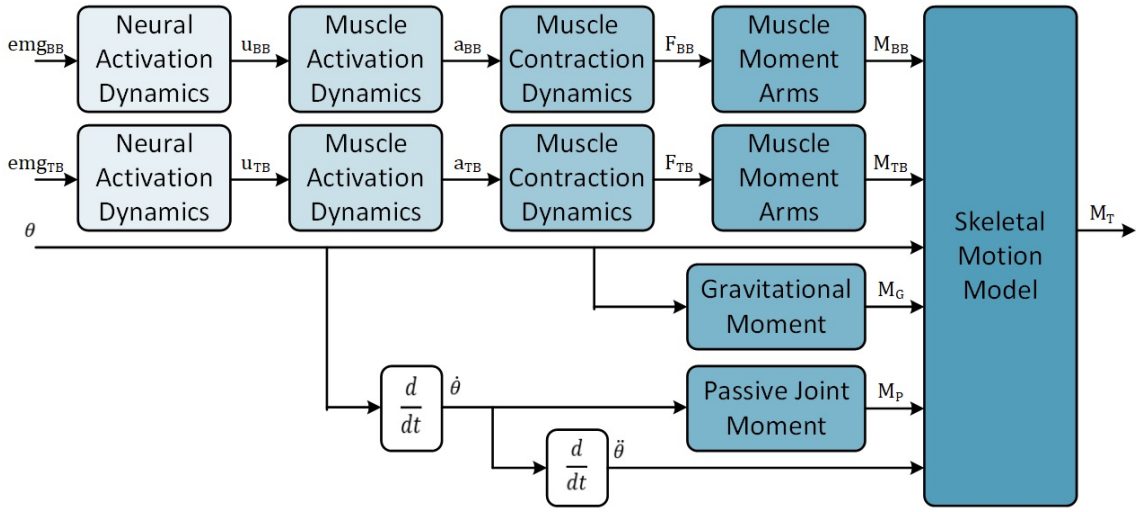


Figure 4.5: The developed elbow motion model uses EMG signals of the biceps brachii, EMG_{BB}, and the triceps brachii muscles, EMG_{TB}, combined with the current elbow angle to estimate the total joint moment of the elbow, M_T. The two muscle torques are configured antagonistically within the skeletal motion model. M_G is the moment caused by gravitational forces; M_P is the passive joint torque; and θ, $\dot{\theta}$ and $\ddot{\theta}$ are the joint position, angular velocity and angular acceleration, respectively.

in the literature [44, 121, 135–139]. Based on frequency spectrum analysis of the collected EMG signals, the implemented neural activation model was a fourth-order low-pass Butterworth filter with a cut-off frequency, f_c, of 3 Hz. The smoothed neural activation signal generated by this filter became the input to the muscle activation models. After examining the literature, it is clear that many muscle activation models have been proposed but no comparative studies of these models have been conducted. A study was conducted using the HTMM to evaluate and compare the various muscle activation models proposed in the literature (see Chapter 5 Section 5.4.1.2).

The muscle activation models can be divided into two categories based on how the models describe the muscle activation dynamics. First, the muscle activation model is developed such that there is a linear and nonlinear component determined by a single parameter [135, 136, 246]. Manal *et al.* and Cavallaro *et al.* both define variations of a single-parameter exponential model [135, 246]. The difference between these two models is whether the parameter is the base of the exponential component or part of the exponent. Manal and Buchanan developed a single-parameter piecewise curvilinear model, using a natural logarithm function [136]. One problem with these muscle activation models is that they have not been optimized separately from the elbow motion

model in which they reside. This makes it difficult to determine the specific performance of these muscle activation models.

The other avenue for designing muscle activation models is based on the concept that there is an EMD for both muscle activation and muscle deactivation, making them two-parameter models [169, 174, 247, 248]. The output of these models varies based on whether the muscle is in a state of muscle activation or muscle deactivation. Rengifo *et al.* and Thelen incorporate this property of muscular control through piecewise differential functions in which the regions are based on the ratio of neural activation to muscle activation [247, 248]. Chadwick *et al.* propose two different functions in which the time delay relationship is incorporated into a non-piecewise first-order differential equation [169, 174]. Commonly, the activation and deactivation time constants are chosen at fixed values based on previous studies of control of human muscle [169, 174, 247, 248]. These seven muscle activation models are described in Table 4.1.

4.3.2.2 Musculotendon Contraction Model

The musculotendon contraction model (Fig. 2.17 (top)) defines both the active and the passive forces generated by the musculotendon unit. To describe this dynamic behaviour, a Hill-type muscle model is commonly used [121,135,136,169,170,174,247,249]. The SPE, typically modelled as an elastic element, is modelled as a rigid tendon element (RTE) in this musculotendon contraction model. This modification was made as Millard *et al.* have shown that an RTE offers similar force estimation errors when compared to other elastic tendon element models used with sub-maximally-activated Hill-type models [188]. The output of the muscle CE, F_{CE} (previously defined as F_{active}), is derived from the mechanical activation, the maximum isometric force (F_{max}), the isometric force-length relationship and the force-velocity relationship of the muscle, as described in Eq. 2.3.

The active force potential of the muscle is dependent on the length of the muscle as described by the following force-length relationship:

$$F_{fl}(L_M) = e^{\left(-\frac{L_M - L_{M_o}}{W L_{M_o}}\right)^2} \tag{4.7}$$

where L_{M_o} is the muscle length at which the most force is expressed (optimal length) and W is

Table 4.1: Functional definitions for the seven muscle activation models developed for the HTMM. Each of the models is assigned an identifier of MAM, which stands for muscle activation model, followed by a number.

Model ID	**First Author [Reference]**	**Model Equations**
MAM 1	Manal [246]	$a(t) = \dfrac{e^{A_1 u(t)} - 1}{e^{A_1} - 1}$
MAM 2	Cavallaro [135]	$a(t) = \dfrac{A_1^{u(t)} - 1}{A_1 - 1}$
MAM 3	Manal [136]	$u_0 = 0.3085 - A_1 \cos(45^\circ)$ $a_0 = 0.3085 + A_1 \sin(45^\circ)$ $m = \dfrac{a_0 - 1}{u_0 - 1}$ $c = 1 - m$ $\beta = \dfrac{e^{a_0/A_2} - 1}{u_0}$ $a(t) = \begin{cases} A_2 \ln(\beta u(t) + 1), & 0 \le u(t) \le u_0 \\ m u(t) + c, & u_0 \le u(t) \le 1 \end{cases}$
MAM 4	Chadwick [169]	$\dot{a}(t) = \left(\dfrac{u(t)}{A_1} + \dfrac{1 - u(t)}{A_2}\right)(u(t) - a(t))$
MAM 5	Rengifo [247]	$\dot{a}(t) = \begin{cases} -\frac{a(t)}{A_1} + \frac{u(t)}{A_1}, & u(t) \ge a(t) \\ -\frac{a(t)}{A_2} + \frac{u(t)}{A_2}, & u(t) < a(t) \end{cases}$
MAM 6	Chadwick [174]	$\dot{a}(t) = (A_1 u(t) + A_2)(u(t) - a(t))$

a shaping factor. The force-velocity relationship describes the difference in force potential as a function of velocity, which varies depending on whether the muscle is concentrically ($\dot{L}_M \le 0$) or

Model ID	First Author [Reference]	Model Equations
MAM 7	Thelen [248]	$\dot{a}(t) = \frac{u(t) - a(t)}{T_a}$ $T_a = \begin{cases} A_1(0.5 + 1.5a(t)), & u(t) > a(t) \\ A_2/(0.5 + 1.5a(t)), & u(t) \leq a(t) \end{cases}$

Optimization parameters of the muscle activation models are denoted A_1 and A_2; $u(t)$ is the neural activation signal; $a(t)$ is the muscle activation signal; u_0 and a_0 are the transition points between the linear and nonlinear portions of the $u(t)$–$a(t)$ relationship; t is time, β, m and c are constants; and T_a is an activation time constant. MAM - muscle activation model.

eccentrically ($\dot{L}_M > 0$) contracting [169], as follows:

$$F_{fv}(\dot{L}_M) = \begin{cases} \frac{\dot{L}_{M_{max}} + \dot{L}_M}{\dot{L}_{M_{max}} - \frac{\dot{L}_M}{A}}, & \dot{L}_M \leq 0 \\ \\ \frac{g_{max} * \dot{L}_M + c_d}{\dot{L}_M + c_d}, & \dot{L}_M > 0 \end{cases} \tag{4.8}$$

where $\dot{L}_{M_{max}}$ is the maximum shortening velocity (chosen as $10 \cdot \dot{L}_{M_o}$ m/s), g_{max} is the maximal normalized eccentric force and A is the Hill curve shaping parameter. Chadwick *et al.* define c_d as a constant used to ensure a continuous first derivative at $\dot{L}_M = 0$ [169]:

$$c_d = \frac{\dot{L}_{M_{max}} * A * (g_{max} - 1)}{A + 1} \tag{4.9}$$

The PPE represents the passive force output of the muscle fibres. This structure exhibits elastic properties that are dependent on muscle length. In this Hill-type model, the PPE is modelled as

a nonlinear spring [169], as follows:

$$F_{PPE}(L_M) = \begin{cases} k_1(L_M - L_S), & L_M \leq L_S \\ k_1(L_M - L_S) & \\ +k_2(L_M - L_S)^2, & L_M > L_S \end{cases} \tag{4.10}$$

where k_1 and k_2 are stiffness coefficients and L_S is the muscle slack length. L_S is chosen to be the L_{M_o} except in situations resulting in high passive forces [169].

In this elbow model, the musculotendon contraction is modelled as an equilibrium musculotendon model [188]. Using this formulation of the musculotendon unit, the forces generated by the muscle and the tendon are in equilibrium, as follows:

$$F_{RTE} = (F_{CE} + F_{PPE}) \cos(\phi) \tag{4.11}$$

where ϕ is the pennation angle of the muscle fibres. Pennation angle is calculated using a constant volume assumption [169], as follows:

$$\phi = \frac{L_{M_o} \sin(\phi_o)}{L_M} \tag{4.12}$$

where ϕ_o is the pennation angle at the optimal muscle length. Combining Equations (4.11) and (4.12), F_{RTE} becomes the output of the muscle contraction model.

In order to customize the muscle contraction model to each subject, OpenSim (National Center for Simulation in Rehabilitation Research, California, U.S.A.) was used to derive estimates of the model parameters. First, the OpenSim Upper Lower Body Model was scaled based on limb lengths measured from each subject [250]. Next, optimal muscle fibre length (L_{M_o}) and pennation angle at optimal fibre length (ϕ_o) values were taken for the biceps brachii long head and triceps brachii long head muscles. Since this elbow model is developed as a two-muscle system, the maximum isometric force (F_{max}) of the major flexor muscle units (biceps brachii short head, biceps brachii

long head, brachialis, brachioradialis and pronator teres) and of the major extensor muscle units (triceps brachii lateral head, triceps brachii long head, triceps brachii medial head and anconeus) were summed to represent the maximum isometric force of each of the flexor and extensor muscle model, respectively. This design choice was made to allow for a simpler two-muscle elbow model that could be driven by two EMG signals, while maintaining approximately similar total muscle forces, as would be generated by the separate muscles. Constant values taken from OpenSim are listed in Table 5.2. Finally, data defining the relationship between muscle unit length (L_M) and elbow joint angle (θ) and between musculotendon unit length (L_{MT}) and elbow joint angle were exported to MATLAB (MathWorks, USA) in order to define muscle length as a function of joint angle using the curve fitting function *fit*.

4.3.2.3 Joint Motion Model

The skeletal motion model describes the relationship between all sources of joint motion. To determine moments generated by the muscles, the moment arms of the musculotendon units must be determined. These moment arms, r, are determined by differentiating the muscle length with respect to the joint angle for each muscle [169], as follows:

$$r = \frac{d}{d\theta} L_{MT}(\theta) \tag{4.13}$$

Height and weight measurements combined with the proportionality equations defined by Winter provide estimates of the gravitational forces acting on the lower arm [251]. A simple cylindrical model of the lower arm provides an inertia estimate for this skeletal model. The passive joint torque equation is modified from the equation proposed by Chadwick *al.* [169] to include only damping, as follows:

$$M_p = -b\dot{\theta} \tag{4.14}$$

where b is a damping coefficient. Chadwick's equation includes a stiffness model that defines that stiffness forces push towards the middle of the joint's range of motion. However, this chosen position and stiffness force distribution may vary based on the individual, the joint position limits,

and the joint structure. The passive joint torque equation has been simplified due to the fact that the joint stiffness was not determined, participants had healthy elbow joints, and motions were restricted within their joint limits.

Finally, these quantities are combined into an equation of motion by summation of joint moments:

$$M_{joint} = I\ddot{\theta} + M_p + M_g + \sum_{i=1}^{N_m} r_i F_{RTE_i} \tag{4.15}$$

where M_{joint} is the total resultant joint torque, I is the inertial mass, M_g is the moment due to gravitational forces, i denotes the i-th muscle, and N_m is the number of muscles contributing to joint motion. The resulting output of the model is the total torque about the elbow joint.

4.3.3 Proportional Motion Model

One of the simplest way of representing the relationship between the EMG signals and the muscle torque is through a proportional relationship. This is based on the idea that the amplitude of the EMG signal is proportional to the muscle force or torque that is provided. After decades of study, it is clear that the relationship is more complex than this initial conception. Hence, to quantify the limitations of this concept, the PMM was developed (Fig. 4.4 (PMM)). The PMM consists of a neural activation model, a muscle torque model, and a joint motion model. The neural activation model consists of a fourth-order Butterworth low-pass filter with a cut-off frequency of 3 Hz. The input to this neural activation model is the processed EMG signals from the biceps brachii and triceps brachii muscles. In order to determine the torque generated by each muscle, the following equation is used:

$$T_m = K_p u \tag{4.16}$$

where T_m is the torque generated by the muscle and K_p is the proportionality constant. Each of the muscle torques can be generated using the relationship in Eq. 4.16 and combined into the joint motion, as follows:

$$M_{joint} = I\ddot{\theta} + T_{BB} - T_{TB} \tag{4.17}$$

where T_{BB} and T_{TB} are the torques generated by the biceps brachii and triceps brachii muscles, respectively. From Eq. 4.17, the acceleration can be determined and used to calculate the estimated velocity and position through integration. The estimated position becomes the output of the PMM.

4.3.4 Nonlinear Polynomial Motion Model

Both Clancy *et al.* and Liu *et al.* have used a NPMM to describe the relationship between EMG signals and the torque of the joint during static postures [127, 173]. Their results are normalized as percentages of MVC flexion torque at 90° elbow flexion. Quantifying results in this manner removes some of the variability of the difference in the strength levels of the participants, but makes it difficult to compare against non-normalized data sets. Furthermore, maintaining static postures simplifies the input signals and may not be representative of the abilities of a nonlinear polynomial to represent the dynamic elbow EMG–torque relationship. As a result, this is the first time a NPMM is used to generate estimates from dynamic elbow motion inputs. The developed NPMM is shown in Fig. 4.4 (NPMM) and consists of three components. To maintain similarities between the NPMM and the PMM, the neural activation model of the NPMM is the same as the one used for the PMM. The relationship between the neural activation of the muscle and the muscle torque is defined as follows:

$$T_m = \sum_{d=0}^{D} p_d \cdot {u_m}^d \tag{4.18}$$

where p_d are the constants of the polynomial and D is the degree of the polynomial. The muscle torques stemming from the biceps brachii and triceps brachii muscles are integrated using Eq. 4.17. Just like the PMM, the output of the NPMM is the estimated joint position.

4.4 Actuation-level Control System Design

The final steps in facilitating human–machine interactions are to appropriately generate motion of the device. The task-level control components determine what to do and when to act and the estimation-level components provide important estimations. The actuation-level control components are responsible for determining how to actuate the device to meet the desired targets, while maintaining motion within the system boundaries. The desired targets may include moving to a

specific position or maintaining a certain torque output, and vary based on the application of the device. Measurement of the degree to which the targets are met is an indication of the control system performance, which relies heavily on the actuation-level control components. Failure to meet the targets could result in non-effective devices or cause injury to users.

There are many important aspects to generating the appropriate motion of the device. The functionality to support motion generation must consider both digital software and hardware designs to maximize performance. Software–hardware partitioning is vital to increasing both the wearability and assistability of wearable assistive devices. Wearability is improved through the use of embedded computer systems to facilitate all of the desired functionality. One issue with these systems is their limited computational resources leading to the need for multiple or more powerful embedded computer systems. However, increasing either their number or power, increases the volume needed to house them, as well as the need for larger batteries. As a result, there is some level where the wearability may regress due to demand for computational resources. Furthermore, the assistability is affected by the available computer hardware as it plays a role into the amount of information that can be processed per unit of time. If a system can process more information, one might expect the accuracy of its behaviour to increase, which leads to an improvement in the device's ability to assist with motion. Due to the novelty of wearable assistive devices, no studies have been done to determine optimal software–hardware partitioning solutions for their digital control systems.

The novelty of wearable assistive devices means that the actuation-level control systems are, in general, in the prototype and exploration stage. This means that software is generally designed without consideration of software engineering principles, implemented under a procedural programming paradigm, and, in many cases, not described in the literature. Hardware is either implemented on general purpose processing systems, such as desktop computers, or using off-the-shelf specialized circuity, such as digital signal processing boards (see Chapter 2 Section 2.2.3). The downside of these approaches is that non-functional qualities of the system are not maximized. However, this is to be expected as the current state of these devices is focused on getting them to achieve their basic functions related to generating desired motions.

Based on the existing literature, the research presented in the following sections deviates in

terms of software development of the actuation-level controller, but follows a similar trend in terms of hardware choices. As with the task-level control and estimation-level control, the actuation-level control software is developed using an object-oriented software development paradigm. Actuation-level control software components were developed to actuate the wearable mechatronic elbow devices used in the experiments defined in Chapter 5. The developed software systems include device transmission models, safety features, and decision algorithms. The control system hardware used to actuate the devices were combinations of general-purpose processing systems and off-the-shelf actuation and sensing systems. Together these software and hardware systems facilitate the motion generation of the devices.

4.4.1 General Functionality of Actuation-level Controllers

Before diving into the details, it is helpful to abstract a set of general functionalities that should be taken into consideration when developing the actuation-level control components. Many factors influence the details of these general functionalities and also may affect the number of them that are required for a specific application of motion production. This set of general functionalities was gathered from the examination of the literature, discussions with experts, and development of digital control systems for wearable assistive elbow devices. Fig. 4.6 presents seven general functionalities required to control the actuation systems of these devices. It should be noted that these aspects may be designed and implemented into other non-actuation-level components. However, for the purposes of this research, these groups of functionalities were implemented under the domain of actuation-level control.

The most important consideration of the actuation-level controller is the timing of task execution. It is vital that motion is produced by appropriate deadlines, which requires control tasks to be timed, optimized, and accounted for when designing execution pathways. Decision algorithms must determine the command to send to the actuation systems. To make these decisions, data must be collected to determine the current state of the user and the device, motion parameters may need to be transformed, and safety checks are made to reduce the probability of entering dangerous user or device states. Once motion commands have been determined, the actuation parameters must be tuned or optimized for specific ranges before the controller is able to regulate motion with

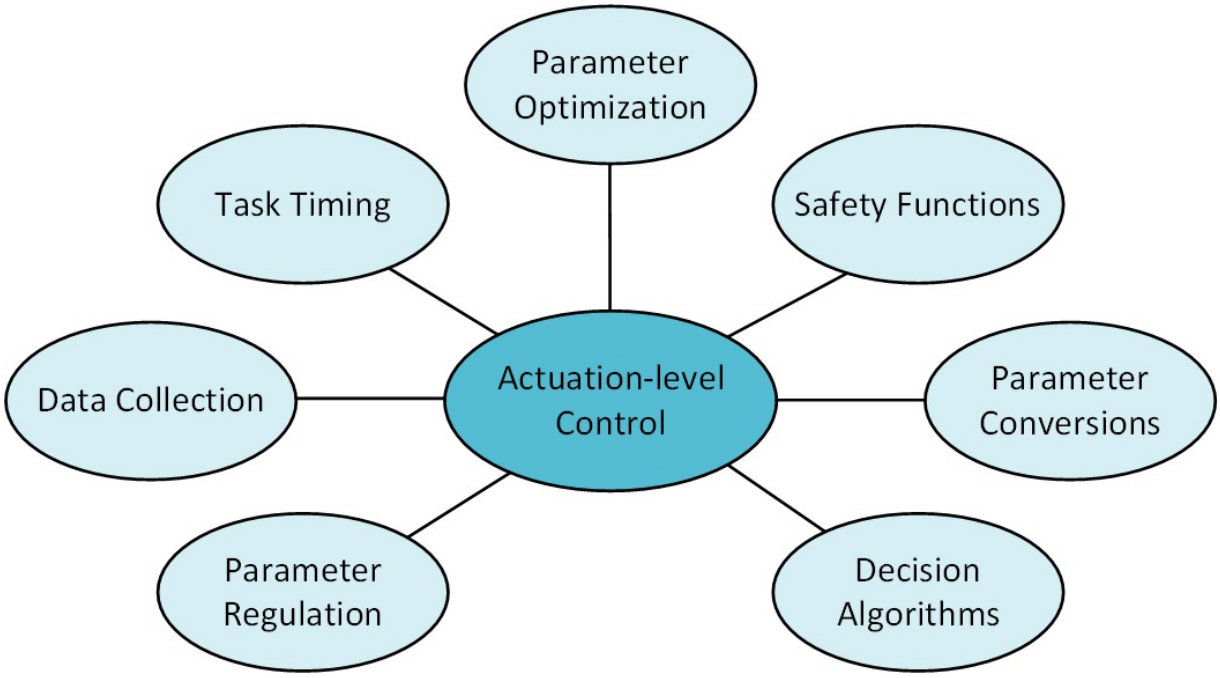


Figure 4.6: Seven general functionalities to consider when developing the actuation-level control components.

a high level of accuracy. These are the general actuation-level functionalities of existing wearable assistive devices. However, the list of possible actuation-level functionalities will vary, based on the intended application.

4.4.2 Multi-threaded Control Software

Wearable assistive devices fall under the category of safety critical systems. Safety critical systems have deadlines on the execution of specific tasks that cannot be missed in order to ensure the safety of the users. For example, tracking elbow motion with a wearable devices requires the generation of motion to occur at very specific times. Furthermore, many important tasks must be executed to generate the appropriate behaviour of the devices. This means that any one function cannot monopolize computational resources that are required to execute multiple functions, such as the case with general-purpose processors. Ideally, all functionality would be decomposed in such a manner that it can be executed in parallel. Although, parallel execution condenses the time to execute tasks, it comes at the expense of processing resources and may not always be possible.

For general-purpose computer systems, one solution is to use an operating system that enables

multi-threaded execution of software. In software, a thread is a container and execution trajectory for a specific series of code. In most cases, there is one thread for an entire software application. When more than one thread is defined, the software application is said to be a multi-threaded application. By sectioning important functionality into independent threads, the operating system will share the processing resources among these threads allowing them to be executed in a more parallel fashion, as opposed to sequentially. This means that safety critical functions do not need to wait for other functions to complete their execution before being executed themselves.

For wearable assistive devices, there are many opportunities to incorporate a multi-threaded software architecture. Defining software threads has a great benefit for functions that are essential to proper behaviour or that may hold the system up for long periods of time. Collecting data from sensing systems and commanding motion of actuation systems are two ideal candidates in multi-threaded control software. The more systems involved in performing a task, such as interacting with sensing and actuation systems, the longer it will take to execute the desired functionality, due to the inherent delays in each system. Furthermore, without sensor data and generation of motion, wearable assistive devices cannot fulfill their goal of assisting with human motion.

The development of the elbow motion tracking control systems in this thesis has shown the need for multi-threaded software systems. Early versions of the control systems were executed as one sequential program (one thread). Although the elbow motion estimation errors were small (Chapter 5, Section 5.4.1.1), there were large errors in the actuation of the wearable mechatronic elbow device controlled by these versions of the control software. Much of this error was due to large delays, relative to the control loop timing, that are caused by interacting with the control system hardware to sense data and actuate the device. To address these issues, a multi-threaded architecture (Fig. 4.7) was used in the control architecture, where separate threads were used for any sensing and actuation tasks that required interacting with the control hardware. Across these software systems, threads were implemented at the actuation-level for commanding position, commanding velocity, and collecting motor position. The main adjustment to the task-level controller was to start execution of these actuation-level threads instead of executing these functionalities within the main control thread. This multi-threaded architecture was a major contributor to the improvement in elbow motion tracking performance of the control systems.

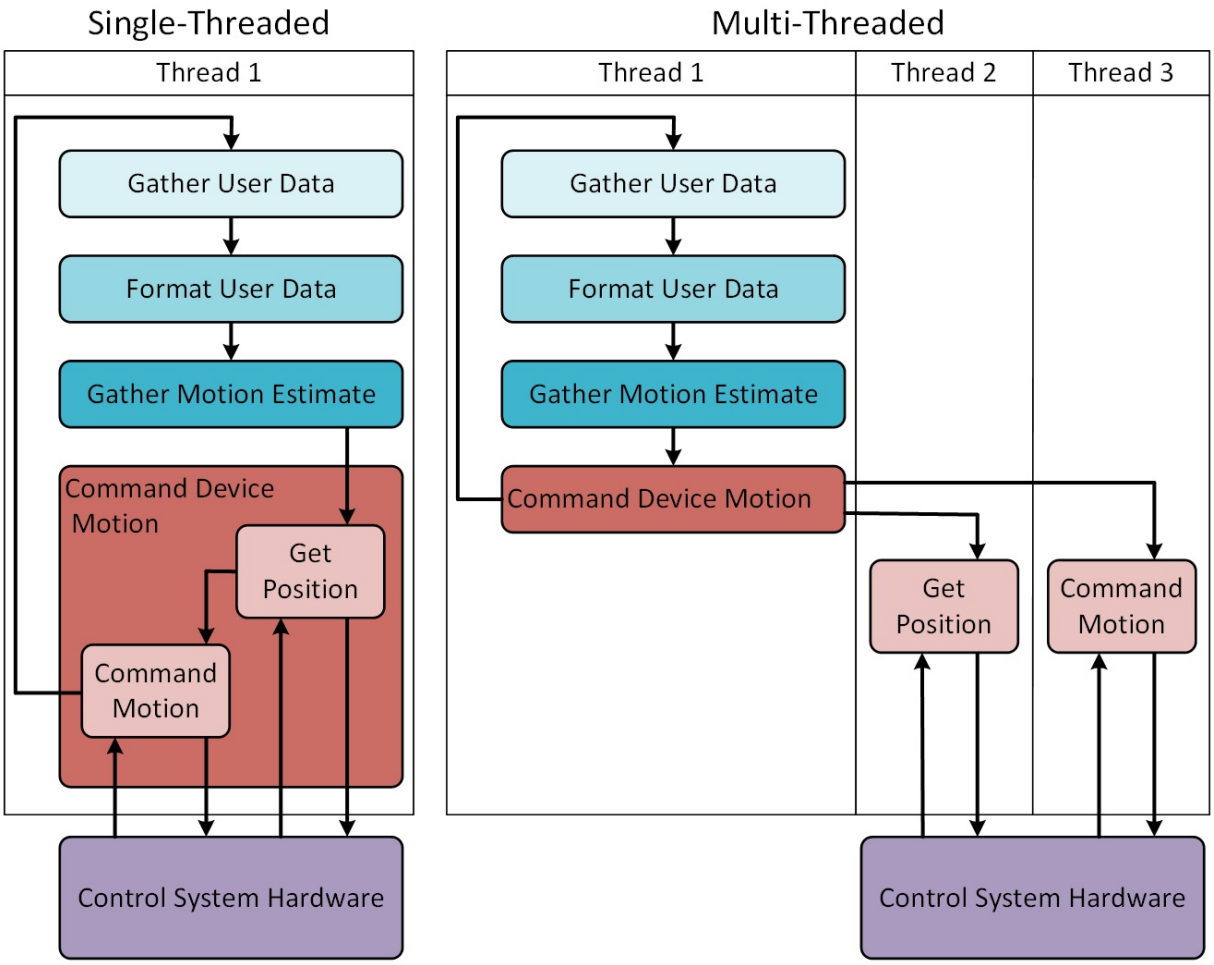


Figure 4.7: Differences in execution of the actuation-level control functionality between the single-threaded and multi-threaded variations of the control system.

4.4.3 Transmission Models

Many of the existing wearable assistive devices require a transmission system to either amplify the actuator outputs or transmit them from one location to another. Actuation systems that produce enough force or torque are often too large to be connected directly to the mechanical joints of the device and require transmission of their outputs from a location removed from the joint. Smaller actuators are used so that they can be placed much closer to the joint. However, the reduced power outputs of these actuators requires transmission systems that amplify the outputs to some desired range. Transmission modelling is used to determine the relationship between the output of the actuator and the output at a specific point of motion within the device.

Two different wearable elbow devices, both with similar transmission systems, were controlled using the developed control systems. Both the WearME elbow brace and the Active A-Gear devices were developed with a motor connected in series with two gearheads, as shown in Fig. 4.8. The WearME elbow brace had a brushless motor (EC-i 40, Maxon Motors AG, Switzerland) connected with a planetary gearhead (GP 32 C, Maxon Motors AG, Switzerland) and a right angle gearhead (Maxon Motors AG, Switzerland). The Active A-Gear used a flat brushless motor (EC 20, Maxon Motors AG, Switzerland) attached to a planetary gearhead (GP 22 C, Maxon Motors AG, Switzerland) in series with a bevel gear (SDI/SI, New York, U.S.A.).

In general, the angular position change across a gearhead can be described by the following equation [252]:

$$\theta_o = \frac{\theta_i}{m_G} \tag{4.19}$$

where θ_o is the output position of the gearhead, θ_i is the input position, and m_G is the gearhead reduction ratio. Using Eq. (4.19) for each of the gearheads in Fig. 4.8, the output position of the second gearhead can be described as a function of the motor input using the following equation:

$$\theta_{GH2} = \frac{\theta_{motor}}{m_{G1} m_{G2}} \tag{4.20}$$

where θ_{GH2} is the position output of the second gearhead, θ_{motor} is the position output of the motor, m_{G1} is the gear ratio of the first gearhead, and m_{G2} is the gear ratio of the second gearhead. The output motion of the second gearhead is the motion that is experienced at the joint of the elbow devices. As a result, Eq. 4.20 is the transmission model used to describe the motion of the joint as a function of position changes of the devices' motors. The WearME brace had gear reduction ratios of 31:1 and 23:1 for the planetary gearhead and the right angle gearhead, respectively. Reduction ratios of the Active A-Gear's planetary gearhead and bevel gear were 128:1 and 3:1, respectively. The transmission models of these two devices were implemented into the control software using the gear ratios together with Eq. 4.20. These transmission models are used bi-directionally within the control system. When a desired joint position for the device is determined, the transmission model can be used inversely to determine the appropriate motor position to command. Conversely, when the position of the motor is gathered from the motor sensors, the transmission model can be

used to derive the joint position.

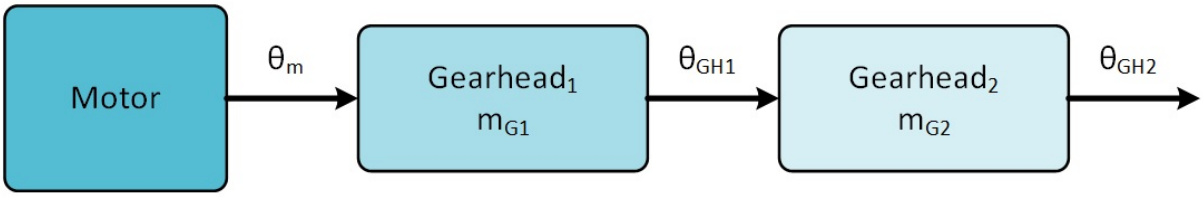


Figure 4.8: A generalized transmission model of the wearable mechatronic elbow devices. In this model, changes in motor position are reflected in changes in position of the two gearheads. θ_m, θ_{GH1}, θ_{GH2} are the position output of the motor, first gearhead, and second gearhead, respectively, m_{G1} is the gear ratio of the first gearhead, and m_{G2} is the gear ratio of the second gearhead.

4.4.4 Decision Algorithms

Decision algorithms are part of the functionality that makes these wearable mechatronic systems intelligent devices. The algorithms enable the control system to make a decision about how to interact with the user. As a prerequisite to a decision, the control system must gather information about the state of the user, the device, and the interaction. Important data from the user include their current and desired motion. The current motion of the user can be gathered from sensing systems, while the desired motion is derived from the estimation-level control components. Sensors incorporated into the devices relay information about its current motion or other important state information. The interaction information includes the goals, such as a desired motion trajectory, and the constraints placed on those goals, such as a maximum acceptable motion error. Decision algorithms must take these data into consideration and decide whether or not to actuate the device and the specific motion command to be executed.

The decision algorithms developed for the control systems evolved in their criteria through iterative development and testing. In general, these decision algorithms were developed around safety checks to determine whether or not an estimated motion command would be sent to the actuators. One assumption for these decision algorithms was that estimated motion parameters were the exact motion that the user desired to complete. This meant that no error correction was conducted with these decision algorithms and the criteria driving the decision were to ensure that

safety of the user was maintained. The velocity decision algorithm for determining the appropriate velocity to command to the motor is as follows:

```
double MakeMotionDecision(double current_position, double desired_velocity)
{
    double time_remaining = 0;
    double space_remaining = 0;

    //Positive velocities represent elbow flexion
    if (desired_velocity > 0)
        space_remaining = upper_position_limit - current_position;
    //Negative velocities represent elbow extension
    else if (desired_velocity < 0)
        space_remaining = current_position- lower_position_limit;
    else
        return 0;

    time_remaining = spaceremaining / desired_velocity;

    if (time_remaining > control_loop_timer)
        return desired_velocity;
    else
        return 0;
}
```

The first step is to determine whether the desired velocity is zero or not. If it is zero, a velocity of zero is commanded to the motor. If the desired velocity is a non-zero value, the algorithm then calculates the amount of space remaining between the current position of the joint and either of the two position limits. From this value, the amount of time required for the device to reach the position limit, if the desired velocity was commanded, is determined. This time value is then compared to the control loop timer. If the position limit would be reached before the next decision is planned to be made, then a velocity of zero is commanded, otherwise the desired velocity would be sent to the motor. This decision algorithm ensures that motions will not go beyond position

limits, even if the processor is busy executing other control functionality. However, it was decided that output limiting should be added to algorithm as an additional safety check, as follows:

```
double MakeMotionDecision(double current_position, double desired_velocity)
{
    double time_remaining = 0;
    double space_remaining = 0;

    //Positive velocities represent elbow flexion
    if (desired_velocity > 0)
        space_remaining = upper_position_limit - current_position;
    //Negative velocities represent elbow extension
    else if (desired_velocity < 0)
        space_remaining = current_position - lower_position_limit;
    else
        return 0;

    time_remaining = spaceremaining / desired_velocity;

    if (time_remaining > control_loop_timer)
        if (desired_velocity > maximum_velocity)
            return maximum_velocity;
        else if (desired_velocity < minimum_velocity)
            return minimum_velocity;
    else
        return 0;
}
```

Some of the developed control systems required position output to be commanded to the motor. Using position output, the decision algorithm is simplified significantly. This is due to the fact that even the execution of other control functions do not interrupt the device from staying at a position that is commanded. Essentially, the desired position needs only be compared to the position limits. The decision algorithm can be simplified to a single safety check, as follows:

```
double MakeMotionDecision(double current_position, double desired_position)
{
    if (desired_position <= upper_position_limit &&
        desired_position >= lower_position_limit)
        return desired_position;
    else
        return current_position;
}
```

The outputs of these decision algorithms are the motion commands to be sent to the controller. The final step before sending these commands is to convert the desired velocities or positions into the appropriate units of each device's control hardware.

4.5 Control Software Implementations

Implementation of the control software requires the integration of the WearMECS framework, software designs, algorithms, software development tools, and software libraries. During this research, many different software applications for either testing software libraries, interacting with sensing systems, commanding actuation systems, developing user interfaces, manipulating subject and device data, and optimizing system parameters were developed to inform the designs of the control system software. From these efforts, four versions of control system software were developed to facilitate either software simulations, control of wearable mechatronic elbow devices, or both of these activities. All of these software systems were developed and executed using either the Windows 7, Windows 8.1, or Windows 10 operating systems. Visual Studio versions 2012–2015 (Microsoft Corporation, U.S.A.) and MATLAB versions R2013b–R2018a were the main development environments used for implementation. Applications involving the development and testing of software libraries, sensing systems, actuation systems, and user interfaces were implemented in Visual Studio. MATLAB was used mainly for developing code to manipulate data, analyze the processed data, and perform optimization of the motion models. Combining all of these aspects into cohesive control software systems was accomplishing using Visual Studio.

Three programming language, C++, C#, and MATLAB script, were used to implement all

of the software systems. Specifically, two of the implementations were developed using C++, one was developed using C#, and one was developed solely as a software simulation, using MATLAB script. These control system implementations relied on the standard .NET software libraries (Microsoft Corporation, U.S.A.), as well as custom software libraries to interface with the sensing and actuation systems. The EPOSCmd library was used to interface with control hardware in order to provide motion commands for the actuators and gather sensor data from sensors located on the device. Standard .NET libraries were used for collecting subject data and digital file manipulation. To support the remainder of the functionality, a custom library was developed to gather EMG data and the System library of the .NET framework provided position data. The following sections will describe four control system implementations in more detail.

4.5.1 Control System Implementation I

Implementation I was developed with the purpose of learning how to control wearable mechatronic elbow devices and improving upon the motion tracking results of studies on both existing motion models and wearable assistive elbow devices. This implementation was used for both software simulations and control of a wearable mechatronic elbow device. Fig. 4.9 presents the implementation, including both software and hardware components. In this implementation, the Tracking Task Controller gathers subject data from digital files, using the File Parser object. The Tracking Task Controller houses the main control loop that iteratively gathers estimates from the KFMM Estimation Controller object and triggers the Motor Controller object to make an actuation decision and command a motion, when it is appropriate to do so.

The KFMM Estimation Controller is developed around the KFMM to predict joint position of the elbow. The estimates from the KFMM Estimation Controller are in the form of joint velocities, derived from the joint position estimates. These estimates are transmitted from the Tracking Task Controller to the Motor Controller at a specific rate, as discussed in Section 4.2.2. Motor positions and velocities are commanded by the Motor Controller object to produce device motion. When the Motor Controller object is requested to command a motion, it executes a decision algorithm, as described in Section 4.4.4, including safety checks, to determine the appropriate motion command. Completion of this decision process leads to a motion command being issued to control hardware

through software functionality developed in the EPOS Driver. The electronic motion controllers can be programmed to operate as either a PI or PID controller to regulate the velocity or position of the motors, respectively. Feedback data for the PID-based controller is gathered from the sensors attached to the motor by the motion controller.

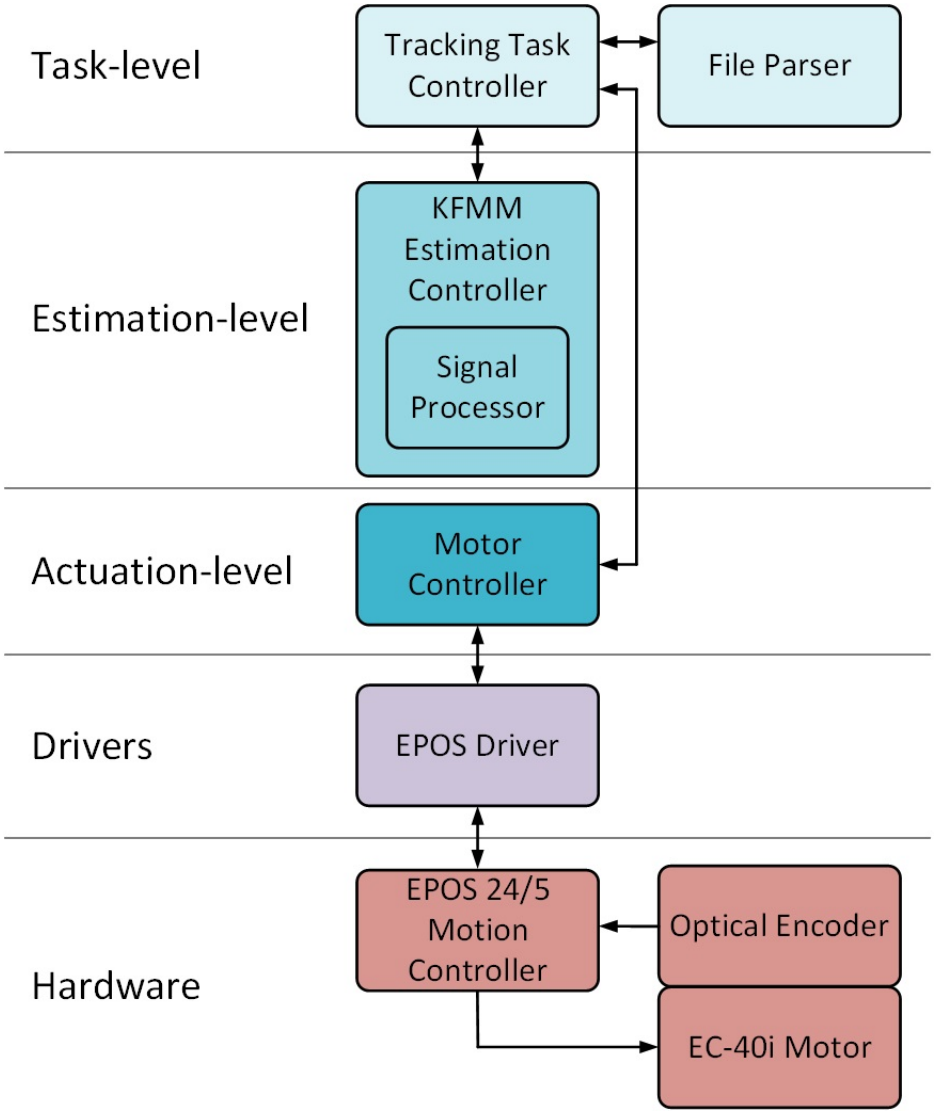


Figure 4.9: Component diagram of the Implementation I control system. This implementation was developed to control the WearME elbow brace (Chapter 5) during elbow motion tracking tasks.

4.5.2 Control System Implementation II

The lessons learned during the development of Implementation I guided the development of Implementation II. The goal for Implementation II was to modify Implementation I to be used with a different wearable mechatronic device. Due to the modular software architecture provided by the WearMECS framework, modification of Implementation I was straight-forward. A component diagram of this implementation is presented in Fig. 4.10. The elbow joint motor of this device was driven using an electronic motion controller, similar to the one used in Implementation I. The EPOSCmd software libraries used to interface with these two control units are similar, resulting in minor modifications of the Motor Controller component. Since the transmission models of the two devices are the same, only modifications of the gear ratios, within the Motor Controller, was made to address the differences. The cut-off frequencies of the filters in the Signal Processor component were changed and modifications were made to the File Parser object to account for the data formatting specifications of a different sensing system. The remainder of control system Implementation II matched Implementation I. This implementation was successfully modified from controlling one device to another, thereby meeting the goal for this system.

4.5.3 Control System Implementation III

Implementation III was developed to conduct a comparison of muscle activation models. The main functionality was designed to facilitate the optimization task described in Section 5.3.2. The component diagram of Implementation III is shown in Fig. 4.11. First, an inverse optimization of the HTMM was performed to generate muscle activation trajectory. Next, a forward optimization of each muscle activation model was performed to determine the optimal parameters for each model. Finally, the HTMM was used, with each optimized muscle activation model, to estimate the joint torque. The performance of the muscle activation models was determined by comparing these estimates to optimized joint torques derived during the inverse optimization procedure. This task was implemented within the Optimization Controller object.

At the task-level, two other objects, the File Parser and the Signal Processor, were implemented to handle the parsing of subject data from digital files and the processing of these data for use

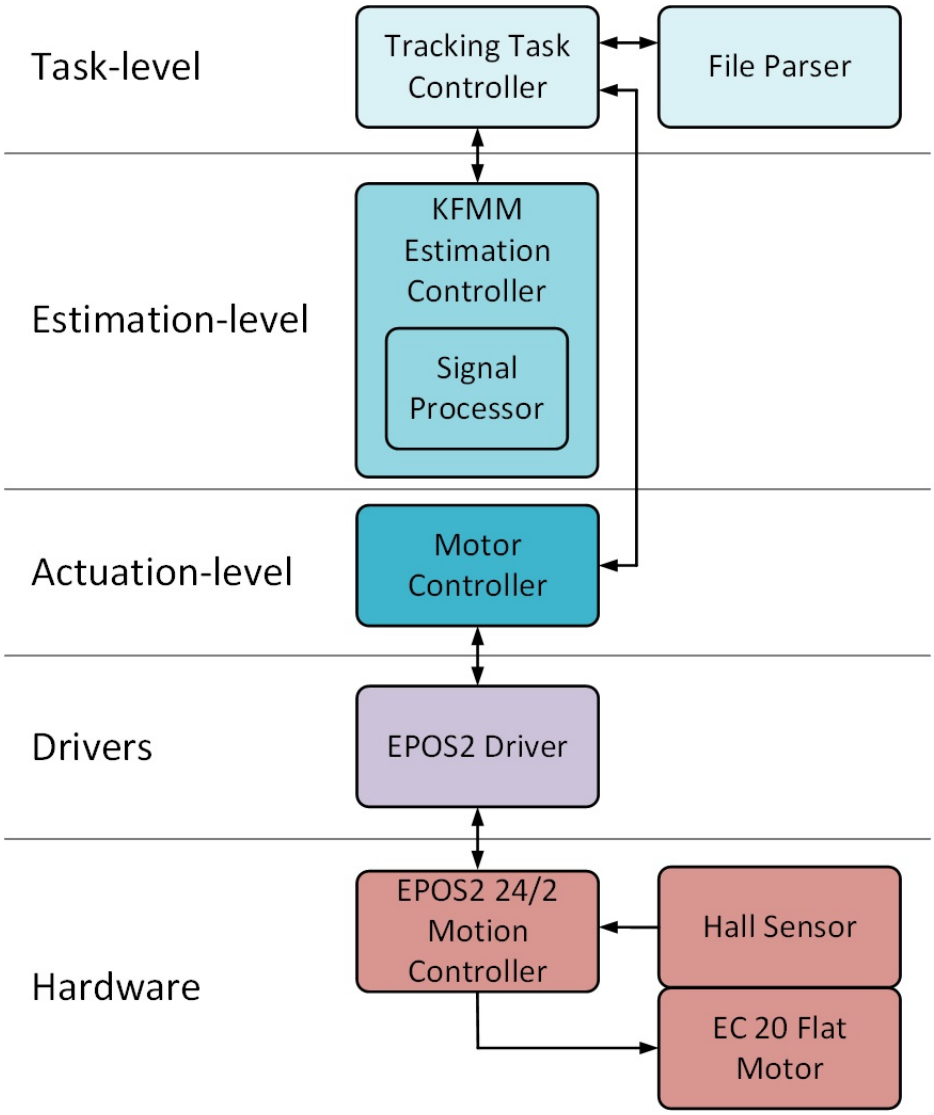


Figure 4.10: Component diagram of the Implementation II control system. This implementation was developed to control the Active A-Gear arm support during elbow motion tracking tasks.

in the optimization task, respectively. The HTMM was developed as an estimation-level component for this control task, as the HTMM Estimation Controller. This component contained both forward and inverse representations of the HTMM described in Section 4.3.2. Implementation III successfully facilitated the comparison of muscle activation models, which was the first comparison of muscle activation models to be found in the literature and was published in [122].

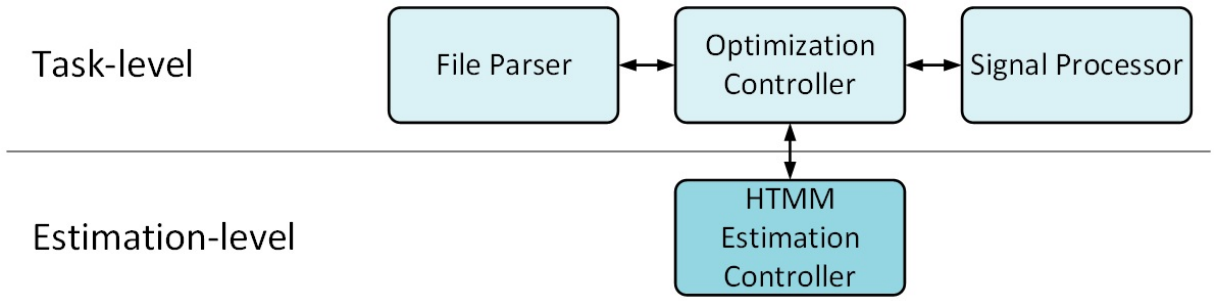


Figure 4.11: Component diagram of the optimization task software developed in Implementation III.

4.5.4 Control System Implementation IV

Building upon the previous work, Implementation IV was developed with the purpose of comparing different elbow motion models used to control a wearable mechatronic elbow brace for a remote-controlled scenario. This implementation is based off of Implementation I, with the major modifications occurring to the task-level and estimation-level components. The Tracking Task Controller was modified to gather data directly from the sensing systems, since the data collection and motion tracking are conducted simultaneously. This resulted in the incorporation of the data formatting functionality into the Tracking Task Controller components. The software objects, Collection Arm Driver and Trigno Driver, were implemented to act as interfaces between the Tracking Task Controller and the sensing systems.

Compared to Implementation I, the PMM Estimation Controller, NPMM Estimation Controller, and Signal Processor components were added to the estimation-level of this control software implementation. In the previous implementation, the functionality of the Signal Processor was included with in the KFMM Estimation Controller. The Signal Processor functionality was implemented as a separate software object in Implement IV, as the functionality was common to and required by all three of the estimation controllers. The PMM Estimation Controller and the NPMM Estimation Controller components implement the PMM and NPMM control models that are defined in Sections 4.3.3 and 4.3.4, respectively. The output of all three estimation controllers is the estimated joint position, which is sent to the Motor Controller to generate motion of the device. The actuation-level control components are the same as those developed in Implementation I. Fig. 4.12 presents a component diagram of this implementation.

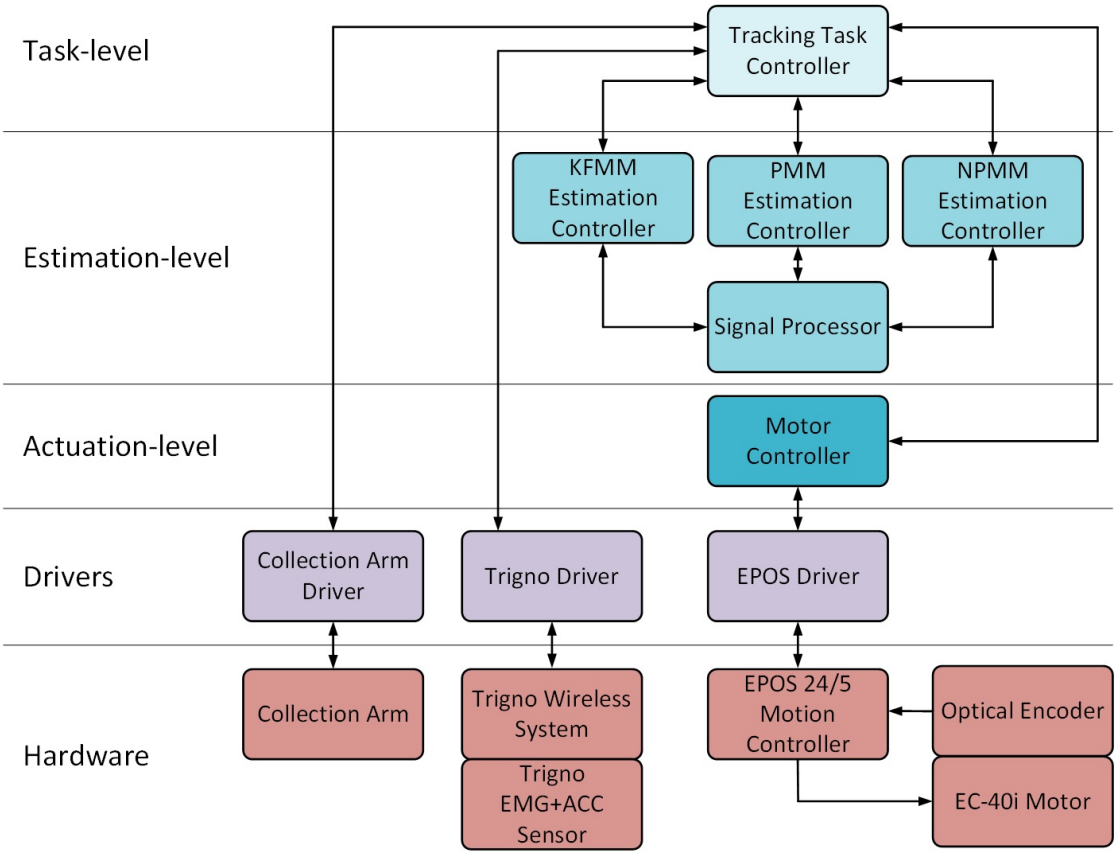


Figure 4.12: Component diagram of the control system developed in Implementation IV. This system was used to perform remote-controlled elbow motion tracking tasks, by controlling the position of the WearME elbow brace.

4.6 Summary

The development of control systems for motion tracking with wearable assistive devices involves the orchestration of many components. It is important that control functionality is proper decomposed to ensure the requirements of the system are met. Using the WearMECS framework, the design of control system functionality can be broken into one of either the task-level, estimation-level, or actuation-level functionality groups. The control systems developed for tracking human elbow motion with wearable mechatronic devices were decomposed in this manner and presented within this chapter. The intended application for these devices was to perform motion tracking tasks that mimic those performed during active-assistive rehabilitation scenarios for the elbow. In order to determine the user's motion intention, four different motion estimation models were developed

to produce either elbow joint position or torque estimates. Tracking motion was accomplished through the control of either the joint position or velocity of the wearable mechatronic devices. Based on the details of the designs, four different control systems were implemented for the purpose of being used in elbow motion tracking tasks. The usage of these implementations and the results of their performances are detailed in Chapter 5.

Chapter 5

Evaluation of Motion Tracking Control Systems

5.1 Introduction

One of the most important functionalities for wearable assistive devices is their ability to track the intended motion of their users. Accurate and repeatable motion tracking leads these systems to be able to provide various assistive forces at any point during this motion. Currently, the motion tracking abilities of the wearable mechatronic elbow devices need improvement. In order to address this problem, a series of seven experiments were conducted in order to evaluate existing control solutions and improve upon their ability to both estimate elbow motion and track elbow motion using these devices. Appendix A contains a map of the experimental variations. The control system evaluation tools (Chapter 3) and implementations (Chapter 4) form the foundations for these experiments. The experiments required healthy individuals to perform various elbow motions that fit within the realm of active-assistive rehabilitation therapies for the elbow. The data collected from these motions were used to estimate the intended motion of the subjects and control the motion of the devices.

Based on their intent, the seven experiments have been split into three groups for the analyses presented in this chapter. The first five experiments were conducted to evaluate and improve upon the motion estimation ability of the KFMM. The sixth experiment was performed to compare

between muscle activation models and determine their contribution to joint torque estimations within a HTMM. In the final experiment, a comparison of three motion estimation models, the KFMM, the PMM, and the NPMM, was conducted during real-time control of a wearable mechatronic elbow brace. In total, six analyses were performed to assess various aspects of the control system performance during motion tracking tasks. The first three analyses focus on the first three phases of integration testing, software simulation, offline remote control, and online remote control, respectively. During these analyses, the effects of different motions and motion estimation models are examined. The fourth analysis focuses on the effects that visual feedback, motion training, and constraining motion have on the motion tracking performance of the control system. The fifth analysis compares between the usage of position or velocity control of the devices' motors. The final analysis explores control task execution times and data storage needs of these control systems. The following sections will present the experiments, optimization procedures, analyses, and discussions. Portions of this chapter have been published in [122, 124, 125].

5.2 Evaluation of Elbow Motion Tracking Control Software

Evaluation of the developed control system was conducted during a series of experiments. The primary objective of the experiments was to quantify the motion tracking abilities of these control systems. In total, seven experiments were completed, in which elbow motions were generated by healthy individuals and the control systems were responsible for tracking the human motion. The focus was to collect various elbow motions that could be experienced in active-assistive phases of elbow rehabilitation, use the data to control wearable mechatronic elbow devices, and measure the motion tracking abilities of the control system implementations. The elbow motions were restricted to a range of -10–120° and a maximum velocity of 40°/s to fit within the realm of active-assistive rehabilitation of the elbow and the capabilities of the devices used in the experiments.

Elbow motion tracking tasks were performed during each of the software simulation, offline remote controlled, and online remote controlled phases of integration testing. No testing was performed during the worn phase, due to inconsistent accuracies achieved during the online remote controlled testing. Across these experiments, there were many variations in experimental factors.

These factors include various motions, models, data collection systems, control outputs, control software implementations, integration testing phases, and experimental hardware. Ethics approval was granted for these experiments from the Human Research Ethics Board at Western University (Reference Numbers: 105717 and 110957) and are shown in Appendix C.

In the experiments, the rate conversion processes involved only upsampling of the position-based data to match the sampling frequency of the EMG data sets. Experiments that used the Biosignalsplux platform captures both EMG and acceleration data at the same rate of 1000 Hz, which means no rate conversion was conducted. Acceleration data captured in Experiment 3 using the Trigno Wireless System was upsampled from 148 Hz to 2000 Hz in order to match the EMG signals captured by this system. Elbow joint positions collected using the Collection Arm device were upsampled from 10 Hz to 4000 Hz or 2000 Hz to match EMG data generated from either the Intronix 2024F Isolated Amplified or Trigno Wireless System, respectively. Positions captured from the WearME elbow brace were obtained at 10 Hz and upsampled to 2000 Hz to match the EMG data collected by the Trigno Wireless System during Experiment 7. The Polaris Optical Tracking System captures the position of the devices at 18 Hz and was upsampled to 1000 Hz to be used with EMG collected from the Biosignalsplux. Either the MATLAB *interp* function or a custom rate conversion algorithm (see Appendix B) were used to facilitate the upsampling of position-based data.

The control system hardware used to facilitate the elbow motion tracking experiments consisted of off-the-shelf computer and electronic systems. For all experiments except Experiment 3, the optimization processes and motion tracking trials were completed using a desktop computer system with an Intel i7-4770 quad-core processor and 16 GB of DDR3 RAM operating at frequencies of 3.4 GHz and 1600 MHz, respectively. Windows-based operating systems provided the development and control platform for these experiments, which included Windows 7, Windows 8.1, and Windows 10 operation system versions. Experiment 3 was conducted using a laptop computer containing an HP Hexa-Core processor and 12 GB of DDR3 RAM with operating frequencies of 2.0 GHz and 1333 MHz, respectively, and was designed using the Windows 8.1 operating system.

Experiments involving the WearME elbow brace used an EPOS 24/5 Positioning Controller to facilitate generating of motor motion, while motion of the Active A-Gear's elbow joint was

generated using the EPOS2 24/2 Positioning Controller. Fig. 5.1 provides an overview of the experimental hardware used with each of these two devices. In the case of the WearME elbow brace, motor commands are sent over a USB-to-RS-232 communication channel that connect the desktop computer to the EPOS 24/5 controller. The EPOS 24/5 controller translates the digital commands into motor currents and transmits these currents to the motor of the brace. The EPOS 24/5 gathers position data from an optical encoder attached to the motor and uses these data as feedback to internally regulate the motor current signals using PID-based control laws. The current motor position, determined from the optical encoder data, is sent back to the desktop computer system, over the USB-to-RS232 communication channel, as desired by the control software. The experimental hardware setup for control of the Active A-Gear is very similar to that of the WearME elbow brace. The differences are that a USB-to-USB communication channel is used between the laptop and the EPOS2 24/2 controller, and that hall sensors are used, instead of an optical encoder, to determine the position of the elbow joint motor. The EPOS2 24/2 controller also internally regulates the motion of the motor using PID-based control laws. The actuation-level control components of the experimental control software were designed to interface with both C++ and C# versions of the EPOSCmd software library (Maxon Motors AG, Switzerland), which enables communication and command of the EPOS 24/5 and EPOS2 24/2 controllers.

Optimal gains for the PID-based controllers are important to achieve the best possible performance of the actuation-level control system. Both the EPOS 24/5 and EPOS2 24/2 controllers were tuned using the EPOS Studio (Maxon Motors AG, Switzerland) software. The EPOS 24/5 controller was implemented as a velocity-driven PI controller with a desired velocity of 20°/s, to match the average velocities of the motions completed during Experiment 1 and Experiment 2. The velocity-driven controller of the Active A-Gear's elbow joint was also implemented as a PI controller. The EPOS2 24/2 controller was tuned for an joint velocity of 25°/s as this was the average velocity of the flexion–extension movements completed by the subjects during Experiment 3. For position-driven control of the WearME elbow brace during Experiment 4–7, the EPOS 24/5 was tuned as a PID controller for a position steps that had a maximum velocity of 40°. One draw-back of the PID-based controllers is that they are highly tuned for a specific velocity or position step, while human motion is highly variable. This issue can be mitigated by constraining

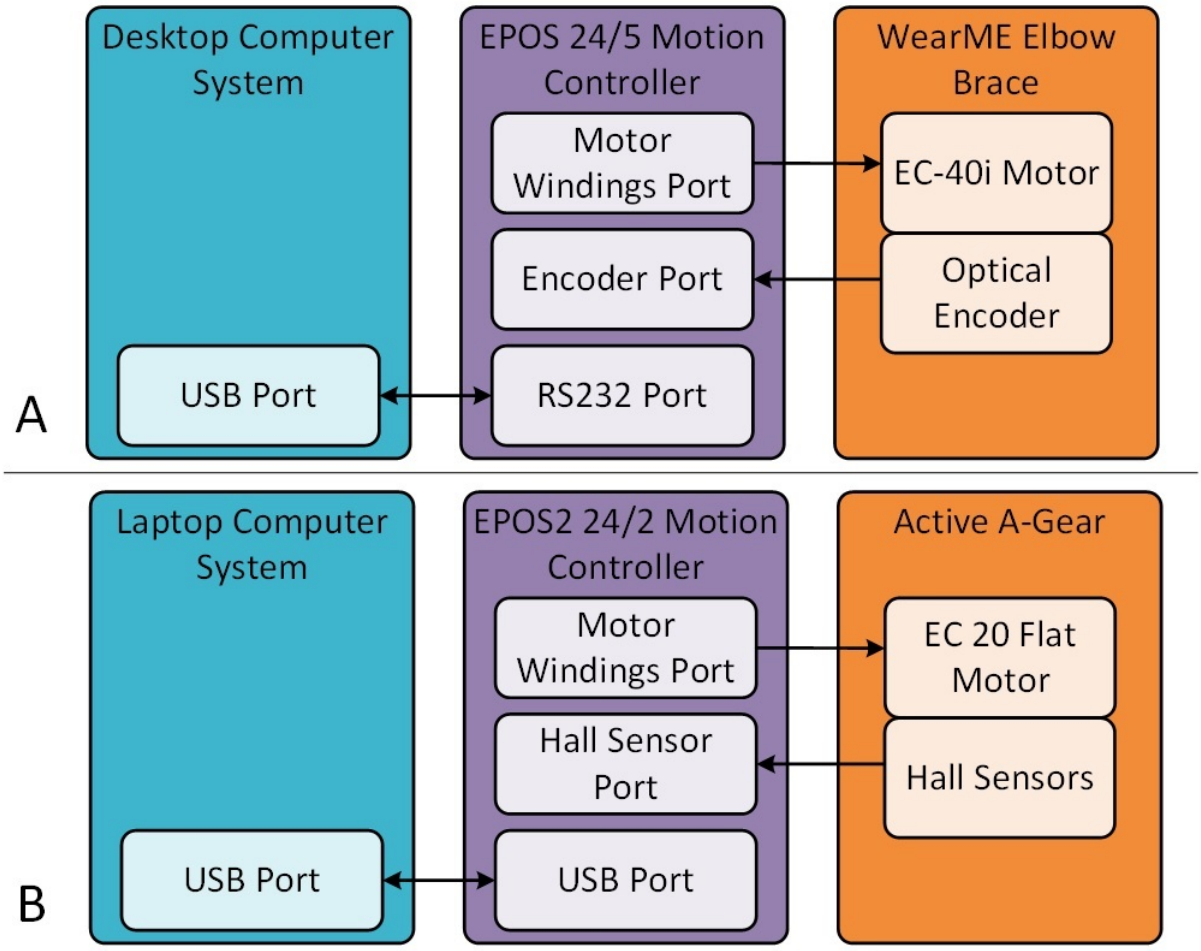


Figure 5.1: Communication diagram of the two experimental hardware setups. The WearME elbow brace was controlled via the EPOS 24/5 Motion Controller, which communicates with a desktop computer system that housed the developed control systems. The control system of the Active A-Gear runs on a laptop computer system and commands motion via the EPOS2 24/2 Motion Controller.

the motion or training the user to perform the motion as accurately as possible. However, this may not translate well to individuals suffering from MSDs, since their motions may inherently be more variable. Future studies should be conducted to determine actuation-level controllers that are more robust to the variability in human motion. A map of the experimental variations is presented in Appendix A. The following sections describe the specifics of the experimental factors for each of the seven experiments.

5.2.1 Experiment 1

Subjects: Five healthy subjects participated in this experiment. .

Motions: Two motions were performed consisting of elbow flexion. Each motion was performed for five repetitions. Both motions were restricted in their range of motion, one from 0–45° and the other from 0–90°. The subjects were not given visual feedback of their elbow position, their elbows were not constrained, and they were not given time to train the motions prior to performing them.

Data collection systems: The Biosignalsplux (PLUX – Wireless Systems, Portugal) sensing system was used to capture both EMG data (Fig. 5.2 and Fig. 5.3) and acceleration data (Fig. 5.3) at a sampling frequency of 1000 Hz.

Integration testing phase: This experiment consisted of both software simulation and offline remote control integration testing.

Implementation: Implementation I was used to facilitate this experiment.

Models: The KFMM was the elbow motion estimation model used in this experiment. The output of this model was the estimated joint position.

Control output: The estimated joint positions were used to calculate an estimated joint velocity. If the conditions of the decision algorithm were met, this estimated velocity was commanded to the device.

Experimental hardware: The WearME elbow brace (Fig. 5.4) was used to track elbow velocity in this experiment. The EPOS 24/5 Positioning Controller (Maxon Motors AG, Switzerland) receives commands from a desktop computer and translates those commands into motor currents that produce motion of the brace's motor.

5.2.2 Experiment 2

Subjects: Five healthy subjects participated in this experiment.

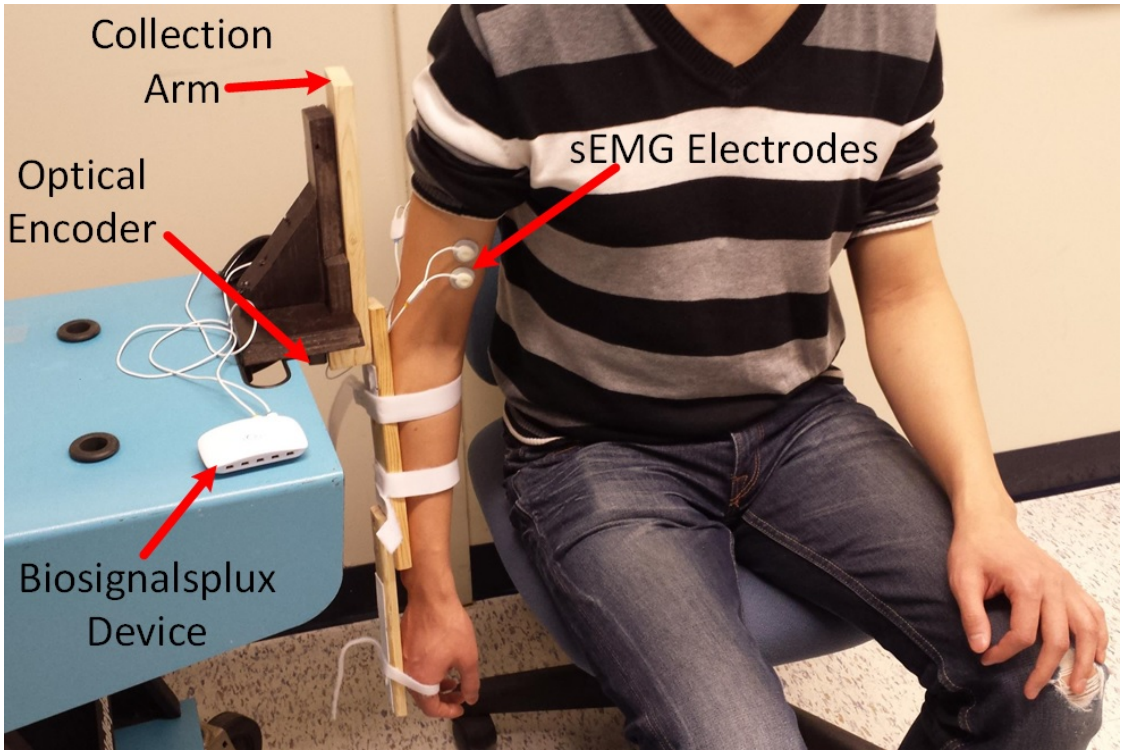


Figure 5.2: The Biosignalsplux device was used to EMG signals from the subjects, while the Collection Arm was used to determine the elbow joint position.

Motions: Three motions were performed consisting of elbow flexion. Each motion was performed for five repetitions. These motions were restricted in their range of motion in the following ranges: 0–45°, 0–90°, and 0–105°. The subjects were not given visual feedback of their elbow position, their elbows were not constrained, and they were not given time to train the motions prior to performing them.

Data collection systems: The Biosignalsplux sensing system was used to capture both EMG data and acceleration data at a sampling frequency of 1000 Hz.

Integration testing phase: This experiment consisted of both software simulation and offline remote control integration testing.

Implementation: Implementation I was used to facilitate this experiment.

Models: The KFMM was the elbow motion estimation model used in this experiment. The output of this model was the estimated joint position.

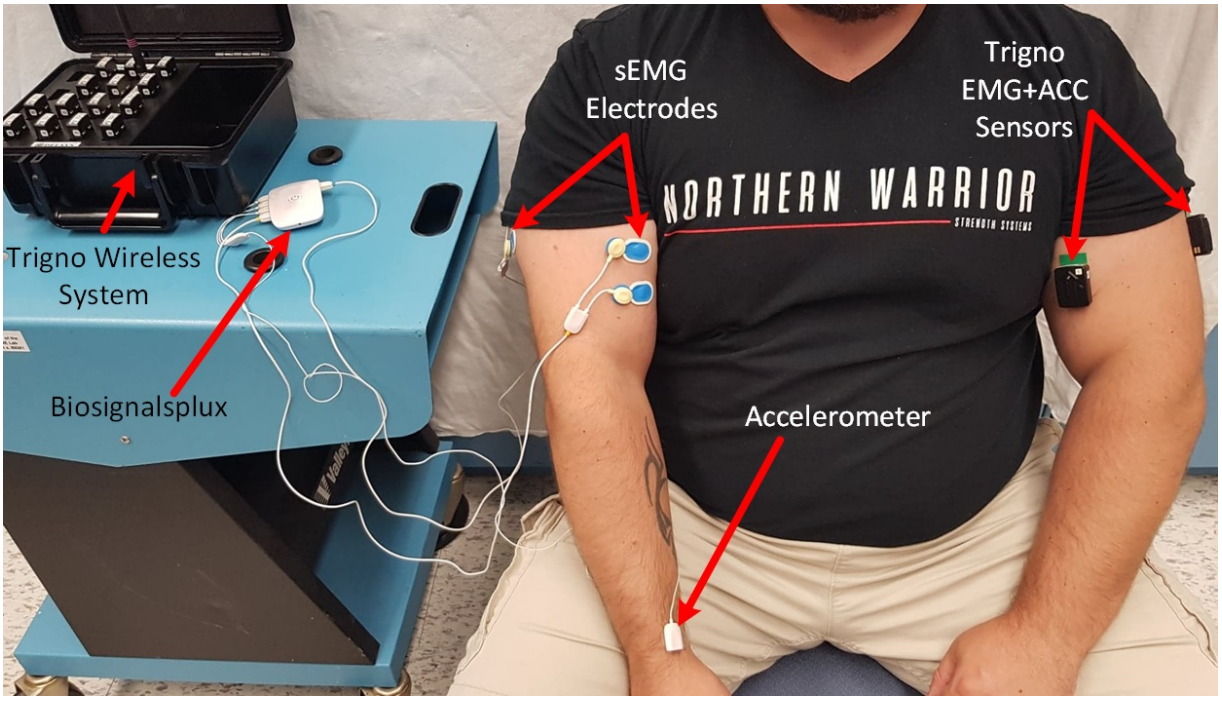


Figure 5.3: Both the Trigno Wireless System and the Biosignalsplux were used to collect EMG and acceleration signals.

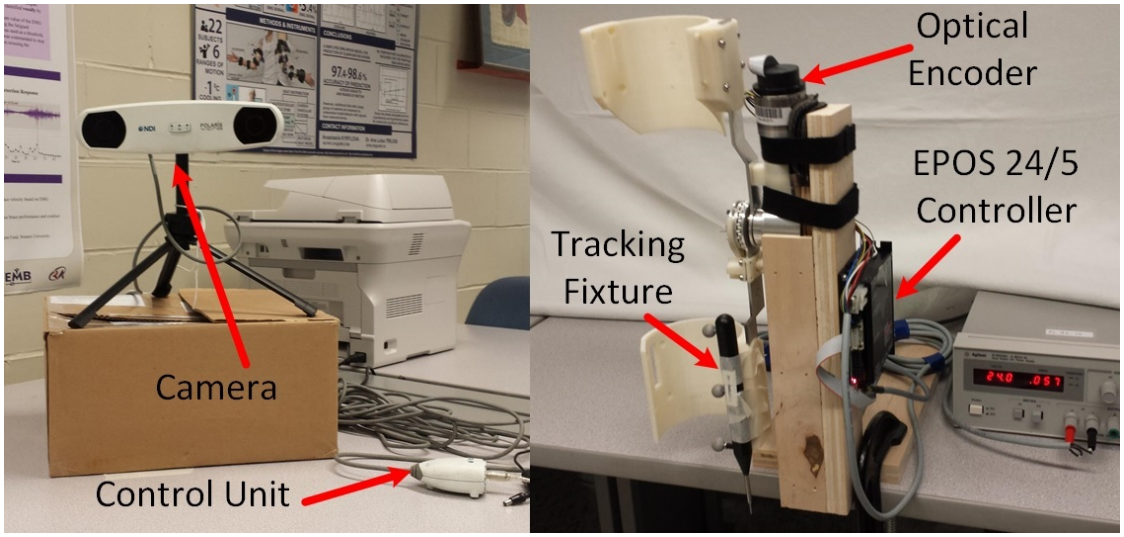


Figure 5.4: The Polaris Vicra optical tracking system (left) was used to measured the position of the WearME elbow brace (right). The tracking fixture contains silver coated globes that are detected by the optical tracking system. The WearME elbow brace has an optical encoder attached to the motor shaft that determines the motor position and can be collected using the EPOS 24/5 Positioning Controller.

Control output: The estimated joint positions were used to calculate an estimated joint velocity. If the conditions of the decision algorithm were met, this estimated velocity was commanded to the device.

Experimental hardware: The WearME elbow brace was used to track elbow velocity in this experiment. The EPOS 24/5 Positioning Controller receives commands from a desktop computer and translates those commands into motor currents that produce motion of the brace's motor.

5.2.3 Experiment 3

Subjects: Five healthy subjects participated in this experiment.

Motions: Three motions were performed consisting of elbow flexion–extension. Each motion was performed for three repetitions.These motions were restricted in their range of motion in the following ranges: 0–45°, 0–90°, and 0–120°. The subjects were not given visual feedback of their elbow position and their elbow was not constrained. However, they were given time to train the motions prior to performing them.

Data collection systems: The Trigno Wireless System (Fig. 5.3 and Fig. 5.5) was used to capture both EMG data and acceleration data at a sampling frequency of 2000 Hz and 148 Hz, respectively.

Integration testing phase: This experiment consisted of both software simulation and offline remote control integration testing.

Implementation: Implementation II was used to facilitate this experiment.

Models: The KFMM was the motion estimation model used in this experiment. The output of this model was the estimated joint position.

Control output: The estimated joint positions were used to calculate an estimated joint velocity. If the conditions of the decision algorithm were met, this estimated velocity was commanded to the device.

Experimental hardware: The elbow joint of Active A-Gear (Fig. 5.6) was used to track elbow velocity in this experiment. The EPOS2 24/2 Positioning Controller (Maxon Motors AG, Switzerland) receives commands from a desktop computer and translates those commands into motor currents that produce motion of the Active A-Gear's motor.

5.2.4 Experiment 4

Subjects: Five healthy subjects participated in this experiment.

Motions: Three motions were performed consisting of elbow flexion–extension. Each motion was performed for three repetitions. These motions were restricted in their range of motion in the following ranges: 0–45°, 0–90°, and 0–120°. The subjects were not given visual feedback of their elbow position but their elbows were constrained to the Collection Arm and they were given time to train the motions prior to performing them.

Data collection systems: The Biosignalsplux sensing system was used to capture EMG data from the subjects, while the Collection Arm (Fig. 5.2 and Fig. 5.7) was used to capture the elbow position. During the offline remote controlled portion of this experiment, the Polaris Vicra optical tracking system (Northern Digital, Canada) (Fig. 5.4) was used to capture the position of the WearME elbow brace.

Integration testing phase: This experiment consisted of both software simulation and offline remote control integration testing.

Implementation: Implementation I was used to facilitate this experiment.

Models: The KFMM was the elbow motion estimation model used in this experiment. The output of this model was the estimated joint position.

Control output: The estimated position of the KFMM was commanded to the motor of the WearME elbow brace if it fell within the limits of safe joint positions.

Experimental hardware: The WearME elbow brace was used to track elbow position in this experiment. The EPOS 24/5 Positioning Controller receives commands from a desktop

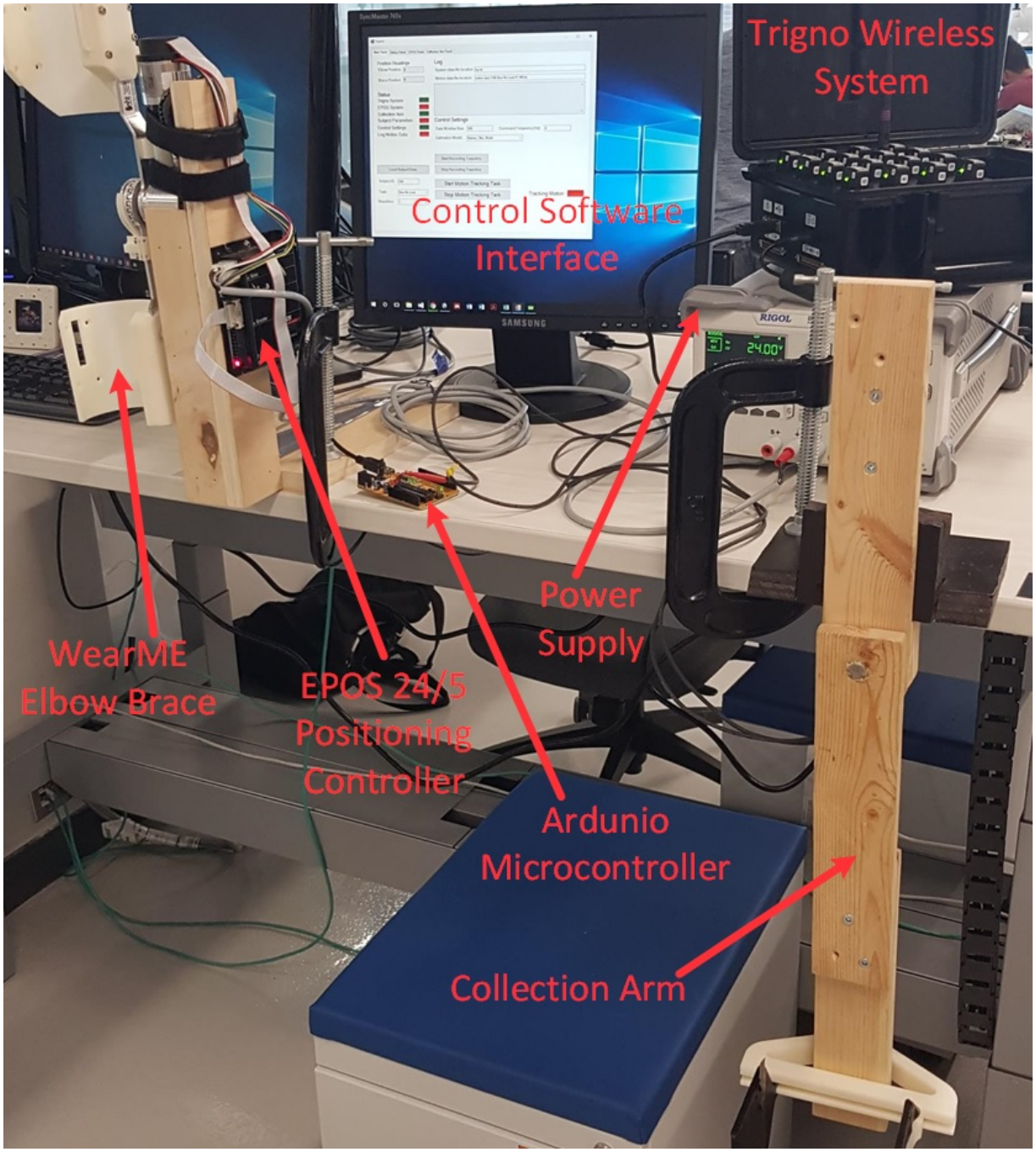


Figure 5.5: The Trigno Wireless System, Collection Arm, and EPOS 24/5 Positioning Controller were used to gather the EMG signals, the subject's elbow position, and the brace's position, respectively.

computer and translates those commands into motor currents that produce motion of the brace's motor.

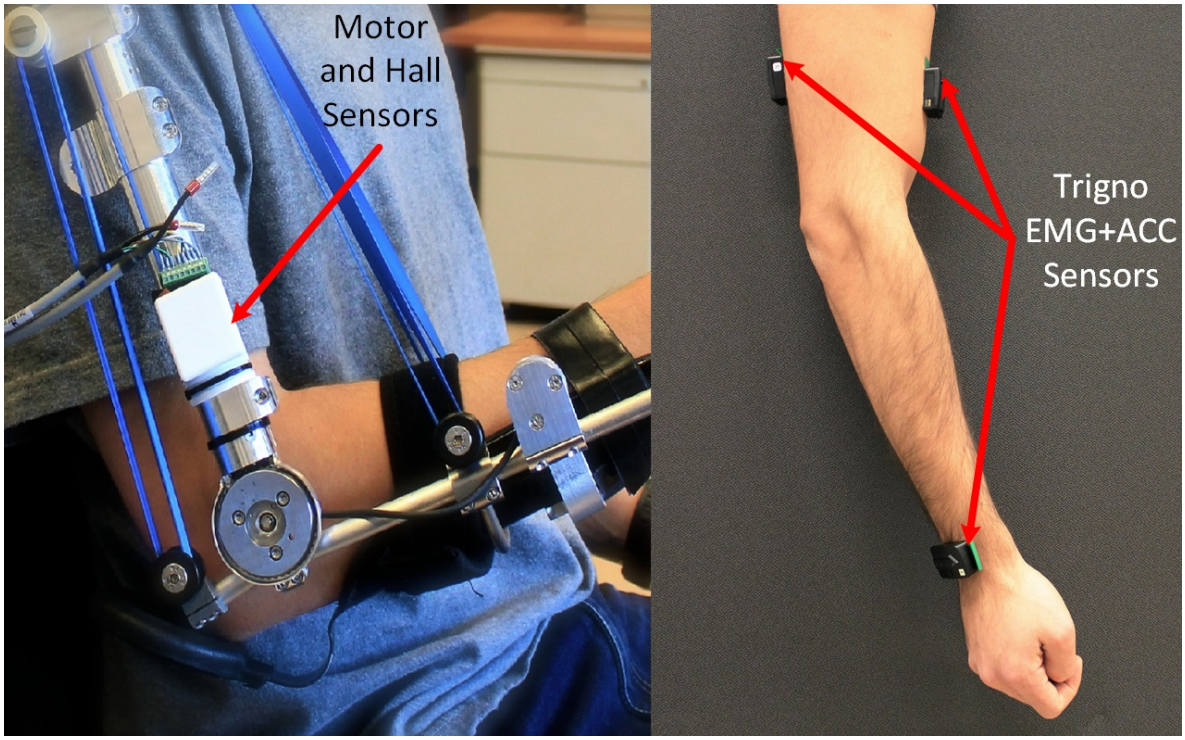


Figure 5.6: The elbow joint position of the Active A-Gear arm support is determined using Hall sensors attached to the motor that drives this joint.

5.2.5 Experiment 5

Subjects: Three healthy subjects participated in this experiment.

Motions: Five motions were performed in total during this experiment. The first motion was performed for one repetition, while the other four motions were performed for three repetitions. The first motion was a smooth continuous flexion–extension motion limited only in range of motion. The second motion was a smooth continuous extension–flexion motion limited only in range of motion. The third motion involved three flexion–extension repetitions with the starting flexion angle being 20, 60, and 100°, respectively. The fourth motion was performed as a flexion–extension movement with two pauses. One pause during flexion and one during extension. The placement pauses were chosen at random by the subject and were held for 1–3 s. The fifth movement involved a smooth continuous flexion–extension motion while holding a 1 kg mass in the hand. The subjects were given visual feedback of their elbow position, their elbow was constrained to the Collection Arm, and they were given time to

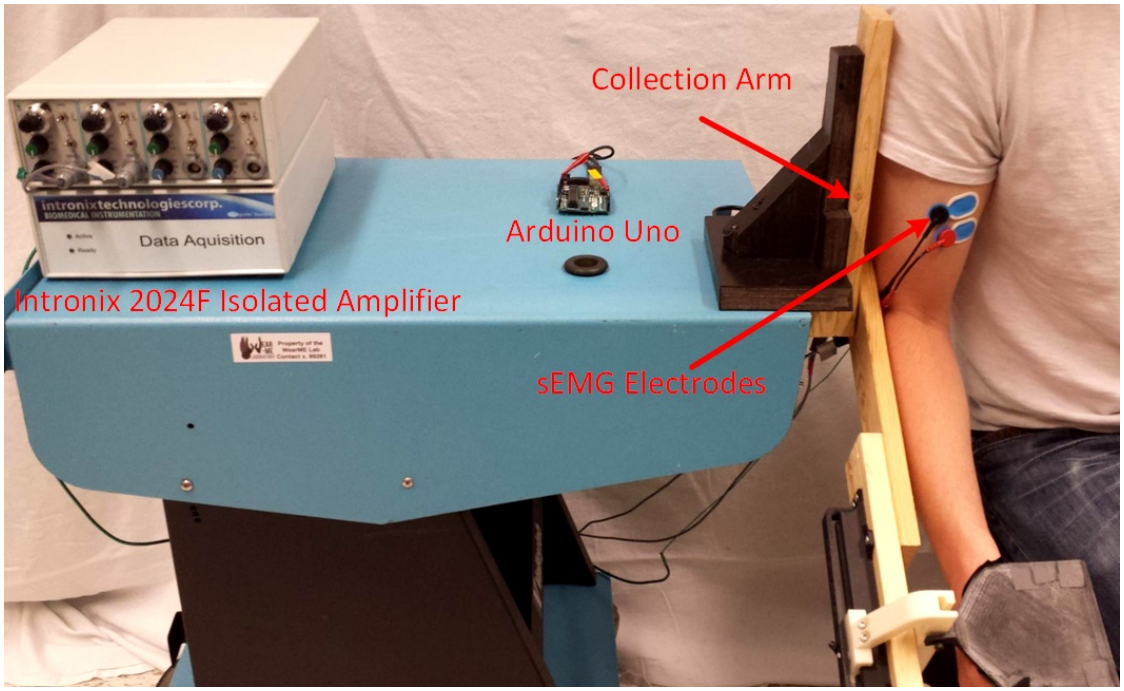


Figure 5.7: The Intronix 2024F Isolated Amplifier collects EMG signals from the subjects. These signals can be combined with joint position signals, such as those collected using the Collection Arm device.

train the motions prior to performing them.

Data collection systems: The Intronix 2024F Isolated Amplifier (Intronix Technologies Corporation, Canada) (Fig. 5.7) was used to capture EMG data from the subjects, while the Collection Arm was used to capture the elbow position. During the offline remote controlled portion of this experiment, the Polaris Vicra optical tracking system was used to capture the position of the WearME elbow brace.

Integration testing phase: This experiment consisted of both software simulation and offline remote control integration testing.

Implementation: Implementation I was used to facilitate this experiment.

Models: The KFMM was the elbow motion estimation model used in this experiment. The output of this model was the estimated joint position.

Control output: The estimated position of the KFMM was commanded to the motor of the

WearME elbow brace if it fell within the limits of safe joint positions.

Experimental hardware: The WearME elbow brace was used to track elbow position in this experiment. The EPOS 24/5 Positioning Controller receives commands from a desktop computer and translates those commands into motor currents that produce motion of the brace's motor.

5.2.6 Experiment 6

Subjects: Six healthy subjects participated in this experiment.

Motions: Five motions were performed in total during this experiment. The first motion was performed for one repetition, while the other four motions were performed for three repetitions. The first motion was a smooth continuous flexion–extension motion limited only in range of motion. The second motion was a smooth continuous extension–flexion motion limited only in range of motion. The third motion involved three flexion–extension repetitions with the starting flexion angle being 20, 60, and 100°, respectively. The fourth motion was performed as a flexion–extension movement with two pauses. One pause during flexion and one during extension. The placement pauses were chosen at random by the subject and were held for 1–3 s. The fifth movement involved a smooth continuous flexion–extension motion while holding a 1 kg mass in the hand. The subjects were given visual feedback of their elbow position and their elbow was constrained to the Collection Arm. However, they were not given time to train the motions prior to performing them.

Data collection systems: The Intronix 2024F Isolated Amplifier (Intronix Technologies Corporation, Canada) (Fig. 5.7) was used to capture EMG data from the subjects, while the Collection Arm was used to capture the elbow position.

Integration testing phase: This experiment consisted of the software simulation integration testing phase.

Implementation: Implementation III was used to facilitate this experiment.

Models: The HTMM was the elbow motion estimation model used in this experiment. The HTMM was combined with seven different muscle activation models in order to evaluate the performance differences between them. The output of the HTMM was the estimated joint torque.

Control output: The output of this control system is the estimated joint torque. No device was being controlled during this experiment as the intent was to evaluate the estimation abilities of the model.

Experimental hardware: Only a desktop computer system was used to run the experiment.

5.2.7 Experiment 7

Subjects: Eight healthy subjects participated in this experiment.

Motions: Six motions were performed in total during this experiment. Each of the motions was performed for three repetitions, one repetition for each motion model used. All of the motions consisted of smooth continuous elbow flexion–extension motions. The six motions were grouped into pairs where one of the paired motions involved an additional 2 kg mass held in the hand and the other involved no additional mass. The first pair of motions required that the subjects performed a single repetition with a maximum velocity of 10°/s. The second pair of motions were performed as a single repetition with a maximum velocity of 40°. The third pair of motions involved 10 consecutive repetitions with a maximum velocity of 40°/s. The subjects were given visual feedback of their elbow position, their elbow was constrained to the Collection Arm, and they were given time to train the motions prior to performing them.

Data collection systems: The Trigno Wireless System was used to capture EMG data from the subjects, while the Collection Arm was used to capture the elbow position. The position of the WearME elbow brace was gathered using the EPOS 24/5 Positioning Controller, which reads the position of an optical encoder attached to the motor of the brace.

Integration testing phase: This experiment consisted of both software simulation and online remote controlled integration testing phases.

Implementation: Implementation IV was used to facilitate this experiment.

Models: Three elbow motion estimation models, the KFMM, the PMM, and the NPMM, were used in this experiment. The output of all three models was the estimated elbow joint position.

Control output: The output of this control system is the commanded joint position of the WearME elbow brace. If the estimated position lies within the safe joint positions, then it is used as the control output.

Experimental hardware: The WearME elbow brace was used to track elbow position in this experiment. The EPOS 24/5 Positioning Controller receives commands from a desktop computer and translates those commands into motor currents that produce motion of the brace's motor.

5.3 Optimizing Motion Models

Each of the four elbow motion models have multiple parameters that must be tuned in order to minimize the motion estimation errors. Optimization algorithms provide methods for tuning the parameters in order to meet some criteria. The main criteria of the optimization procedures used in the experiments were to minimize the error between estimated motion and measured motion, such as elbow joint position. The optimization processes vary based on the model that was used and the experiment in which it was used. The processes used for the experiments are grouped into three groups, Experiments 1–5, Experiment 6, and Experiment 7, based on these differences. The details of these processes are described in the following sections.

5.3.1 Model Optimization: Experiments 1–5

The KFMM requires the optimization of parameters belonging to neural activation model (Eq. 2.2), the summation of neural activation signals (Eq. 4.1) and the Kalman filter model (Eq.

4.2–4.6). Kyrylova *et al.* manually tuned the neural activation parameters using MATLAB (Mathworks, USA) to the following values: α = 0.0021, β_1 = -1.78, β_2 = 0.7821 and G_{na} =10. These values were used in Experiment 1–5 and remain constant across all data sets. Lastly, R and Q were determined using the calibration data set collected from the subject, along with the optimized neural activation parameters. The calibration data set for each experiment was taken from the largest range of motion flexion-only or flexion–extension movement. For Experiment 1, Experiment 2, and Experiment 3–5, the calibration movements involved 0–90° flexion, 0–105° flexion–extension, and 0–120° flexion–extension movements, respectively. In Experiment 1, the values of R and Q were tuned manually based on analysis of the estimated and measured trajectories, such that the same R and Q values could be used across all data sets. A constrained nonlinear minimization was completed to determine the R and Q values for Experiment 2–5, using the MATLAB *fmincon* function. The goal of this optimization was to minimize the error between the estimated joint angle and the measured joint angle. Based on the poor results of tuning from Experiment 1, the optimizations of Experiments 2–5 were conducted for each subject, instead of optimized across all subjects. The optimized parameters of the KFMM for each of the experiments is presented in Table 5.1.

Table 5.1: R and Q values determined through the optimization procedures conducted during Experiments 1–5.

Experiment	R	Q
1	570	0.15
2	1.97–5.94$\times10^{16}$	8.76–25.2$\times10^{7}$
3	3.31–1695.50$\times10^{13}$	8.12–1639.80$\times10^{5}$
4	1.74–5.24$\times10^{15}$	1.65–5.54$\times10^{7}$
5	2.59–14.54$\times10^{15}$	1.28–4.13$\times$‘10^{6}

5.3.2 Model Optimization: Experiment 6

Optimizing the muscle activation models was accomplished using both an inverse and forward optimization for each recorded motion. These procedures were completed using a desktop computer

system with an Intel i7-4770 quad-core processor and 16 GB of DDR3 RAM running the Windows 10 operating system. The inverse optimization began by finding the total muscle torque that minimizes Eq. 4.15 at each position along the trajectory of the recorded motion. Next, the muscle activations were determined from the resulting total muscle torque. However, the relationship between the activations of the two muscles are dynamic and unknown, leading to an indeterminant system. To work around this limitation, a single musculotendon unit, sharing the properties of the biceps brachii muscle, was used. The muscle activation was inversely derived from the optimized muscle moment as shown in Figure 5.8.

The forward optimization was completed in order to optimize the parameters of each muscle activation model. Each EMG channel, two in this case, has an associated muscle activation model where the difference of the activation between these two channels forms the total muscle activation (Figure 5.8). A similar minimization process as the one described by Buchanan *et al.* is used [121]. However, the procedure used here minimizes the squared muscle activation error (Chapter 3, Table 3.3 second row) as opposed to minimizing of the squared joint moment error. Both optimizations were completed using a constrained minimization implemented in MATLAB using the *fmincon* function. For the inverse optimization, only a single parameter, the total muscle torque, was used as an optimization variable. The forward optimization procedure required optimization of either one parameter (MAM 1 and MAM 2) or two parameters (MAM 3–7), per EMG channel, depending on the muscle model used. Parameter ranges for MAM 1 (A_1:-3–0) and MAM 2 (A_1:0.05–1) were taken from the literature [135,246]. For MAM 3, it was decided to optimize two model parameters simultaneously instead of the nested minimization used by Manal *et al.* [136]. The range of the A_1 parameter was listed in [136] (0.0001–0.12), and the A_2 parameter range was chosen by experimenting with the limits of the range (0.01–10^{11}). MAMs 4–7 did not optimize activation and deactivation parameters in their studies [169, 174, 247, 248]. Time constants of the elbow muscles have been listed as 5–40 ms and 20–70 ms in the literature for muscle activation and deactivation, respectively [244, 245]. As a result, the optimization range for these values was chosen to be 0–70 ms. The parameter ranges for both forward and inverse optimizations are shown in Table 5.2. Following each forward optimization, the optimized parameters were used in conjunction with the HTMM to estimate the elbow joint moment.

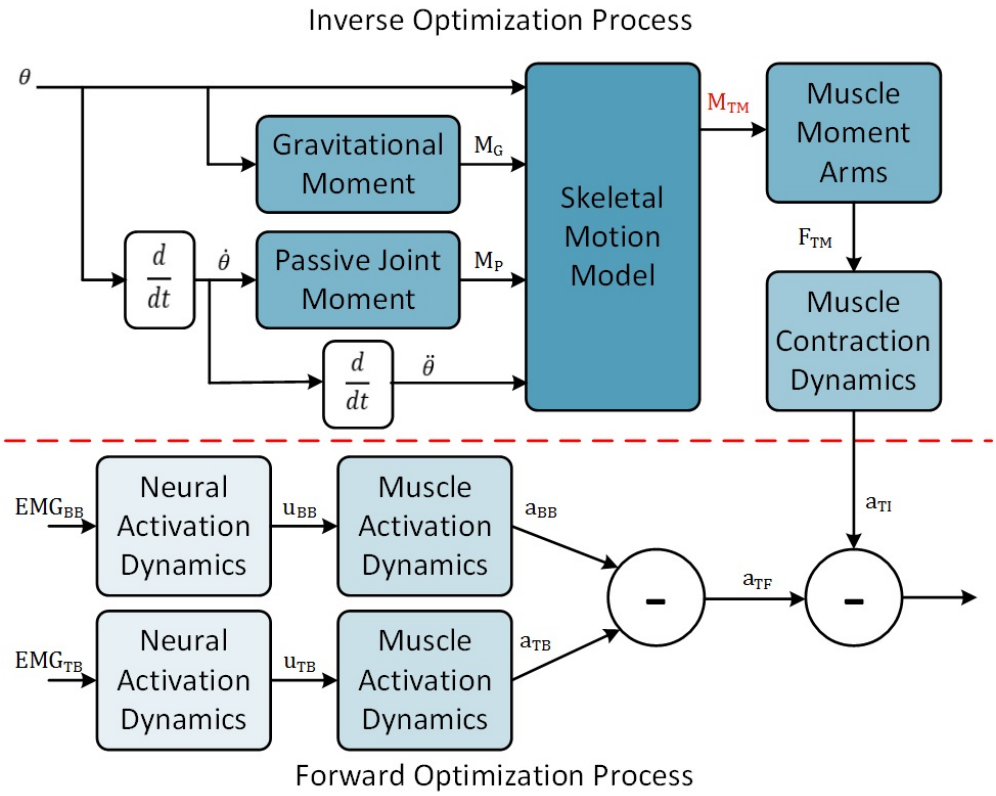


Figure 5.8: The inverse optimization procedure determines the optimal total muscle torque, M_{TM}, to minimize the elbow joint moment. Then, the total muscle activation, a_{TI}, is derived from the total muscle torque. The forward optimization process minimizes the error in muscle activation signals by determining the optimal parameters values for each muscle activation model. These two processes (separated by the red dotted line) are completed sequentially for each dataset. F_{TM} is the total muscle force; a_{TI} is the inversely-derived total muscle activation; and a_{TF} is the forward-derived total muscle activation.

5.3.3 Model Optimization: Experiment 7

Three motion models, the KFMM, PMM, and NPMM, were used to estimate elbow motion during Experiment 7. The first portion of this experiment involved the optimization of these motion models. Custom software was developed in MATLAB to perform the optimization of these models, prior to being used in the motion tracking portion of the experiment. The optimization began by recording a MVC of the biceps brachii and triceps brachii muscles of the subject. A single repetition of elbow flexion–extension motion was then recorded to be used as the trajectory for which the models would be optimized.

Table 5.2: Description and source information for elbow model constants.

Parameter Symbol	Parameter Source	Parameter Values (Units)	
EMG Processing and Neural Activation Model			
f_c	Analysis, [91, 138]	3 Hz	
Muscle Contraction Model			
		Biceps Brachii (Long Head)	Triceps Brachii (Long Head)
F_{max}	OpenSim	2874.67 N	2397.12 N
L_{M_o}	OpenSim	0.1106–0.1361 m	0.1236–0.1681 m
ϕ_o	OpenSim	0 rad	0.2094 rad
W	[169]	0.56	0.56
$\dot{L}_{M_{max}}$	[169]	1.106–1.361 m/s	1.236–1.681 m/s
A	[169]	0.25	0.25
g_{max}	[169]	1.5	1.5
L_S	[169, 174]	0.1659–0.2042 m	0.1854–0.2522 m
Skeletal Motion Model			
b	[169]	1 Nms/rad	

In this experiment, three parameters of the KFMM were used in the optimization process, R, Q and G_{na}. In the previous optimizations, G_{na} was held constant resulting in large values for R and Q. The PMM had two parameters to be optimized. These were the proportionality parameters, K_{BB} and K_{TB}, for each EMG channel. Eight optimization parameters were required for the NPMM model. In this model, the relationship between the EMG signal and the muscle torque was represented by a cubic polynomial, resulting in four coefficients, p_1–p_4, per EMG channel. An unconstrained nonlinear minimization was performed on the collected data, using the MATLAB function *fmincon*. The objective was to minimize the RMSE between the subject's elbow position and the estimated elbow position of the model. A new calibration data set was collected and the optimization process was performed repeatedly until a RMSE between the subject's position and the estimated position of less than 20° was found for all three models. Ranges for each of the

optimization parameters of these three models are listed in Table 5.3.

Table 5.3: Optimized model parameter ranges derived during the optimization procedure of Experiment 7.

Parameter	Minimum Value	Maximum Value
	KFMM	
G_{na}	-71972.544	10181.262
R	2.000	83624648.760
Q	0.000	16054.634
	PMM	
K_{BB}	-1344.410	-18.178
K_{TB}	-2295.161	-38.595
	NPMM	
$p1_{BB}$	0.958	244095.144
$p2_{BB}$	-775709.654	312777.473
$p3_{BB}$	-4672.625	2843.332
$p4_{BB}$	-1.374	10.308
$p1_{TB}$	-52482.947	1766.949
$p2_{TB}$	-43877.545	75982.391
$p3_{TB}$	-2596.170	4663.827
$p4_{TB}$	-10.306	1.377

BB and TB represent parameters for each of the biceps brachii and triceps brachii, respectively.

5.4 Control System Evaluation

During the experiments, metrics were collected to support the analysis of the control system qualities, which included accuracy, repeatability, adaptability, and resource utilization (Chapter 3 Section 3.4.3). Statistical analysis of these metrics were conducted within each of the testing phases to identify factors that may impact the control systems and their ability to track elbow motion. The

control systems were evaluated with two main intents. First, control system analyses were grouped together based on the phase of integration testing they embodied. In this thesis, the experiments were conducted under either one or two of the software simulation, offline remote control, or online remote control integration testing phases. During these analyses, the effects of the motions and models were examined. Second, three analyses were conducted to explore experimental factors that affect the motion tracking performance. An analysis of the subjects' motion performance showed that there are three factors which significantly affect the motion tracking performance. A comparison between position and velocity control of the WearME elbow brace was conducted. Finally, an analysis of control task timing and data storage needs was performed. The results of these analyses are presented in the following sections.

5.4.1 Software Simulation Evaluation

All of the seven experiments involved the tracking of the motion estimation values, which is denoted as the software simulation portion of these experiments. In Experiment 1–5, Implementations I and II were used to estimate either the desired position or the velocity of the subject. Implementation III was developed purely to facilitate a software simulation involving the optimization of muscle activation models, which was performed during Experiment 6. In this experiment, forward and inverse optimizations were performed to determine the optimal muscle activation model parameters that would minimize the joint torque errors. Finally, the position estimates of three different elbow motion models were collected from Experiment 7, which used Implementation IV to perform this tracking task.

One benefit for performing software simulation experiments is that the portion of the control system implementations responsible for regulating the device's motion are removed from the analysis. In the case of developed control systems, this means that the performance of this testing phase is based solely on the task-level and estimation-level components. As a result, it is much easier to isolate and compare between different motion estimation models and their relationships with the motion estimates. When comparing across Experiments 1–5, it was found that there were statistically significant differences between the motion inputs used with the KFMM model. The analysis revealed that the KFMM performed better for motions that were more similar to the

motion for which it was developed. This suggests that motion models could be designed for specific motion tasks or that they should be improved to be more robust to variations in motion inputs. The results of the software simulation tracking completed during Experiment 6 show that the developed HTMM is able to produce torque estimation errors that are consistent with the ranges found in the literature. However, this came at a massive expense of computational resources and time. These results highlighted that there is a trade-off between accuracy and resource utilization and that more studies are required with HTMMs to improve the understanding of human motion dynamics.

The software simulation results of Experiment 7 emphasized that it may not be possible, especially with the existing motion models, to have one motion model that works for all humans. Three different elbow motion models were developed within Implementation IV, as part of this comparative study. Not only were there differences found between the models overall, but it was also found that models performed better for some individuals than for others. Due to the small sample size, it was not possible to control for all of the individual differences between the subjects. However, this is still an important finding, which suggests further studies should be conducted to determine how factors of the human affect the motion estimation abilities of these models. Full descriptions of the results and analyses of these software simulation tracking experiments are presented in the following sections.

5.4.1.1 Evaluating the Kalman Filter Motion Model

The KFMM was implemented and evaluated through six elbow motion tracking experiments, Experiments 1–5 and 7. However, the analysis here will only consider Experiments 1–5, as there were major differences in the experimental protocol and objectives between these experiments and Experiment 7. The analysis of Experiment 7 is presented in Section 5.4.1.3. During each of these experiments, comparisons between the input motion and the estimated motion were conducted to determine accuracy and repeatability measurements. Accuracy was measured using the joint position error metric, while the repeatability was assessed using the standard deviation of the position error metric. These measurements represent the performance of the control software to estimate motion and represent the software simulation case of these experiments. A summary of

these metrics across all data sets for each experiment is presented in Table 5.4 and for each motion is presented in Table 5.5.

A statistical analysis was performed to determine any significant differences between both the experiments and the motions. The position errors collected from both the software simulation motion tracking portion of Experiments 1–5 were used as the dependent variables. The Kruskal–Wallis H test was used to determine these differences, as the data exhibited non-normal distributions and the factors contained more than two independent groups. In total, 10 pair-wise tests were performed between the 5 experiment groups and 45 pair-wise tests were performed between the 10 motion groups. Since results from multiple experiments were compared, a Bonferonni correction was used to adjust the significance. The analysis was performed using IBM SPSS Software (Version 24, IBM Corporation, U.S.A.) and statistical significance was determined using an α value of 0.05.

Looking across the experiments, the lowest mean position estimation error was $0.008 \pm 0.002°$, which was achieved in Experiment 4. The largest position estimation errors were captured during Experiment 1. However, this estimation error ($0.280 \pm 0.186°$) is still a large improvement on the estimation error achieved by Kyrylova *et al.* (1.82–3.38°) during the initial evaluation of the KFMM [123]. In terms of motions, the 0–120° flexion–extension motion produced the lowest position estimation RMSE ($0.031 \pm 0.047°$), while the largest error was obtained from the 0–90° flexion motion ($0.242 \pm 0.220°$). One could reasonably expect this result for the best performing motion estimation to occur when using the 0–120° flexion–extension motion as input, as this is the motion for which the model was developed [123].

The statistical analysis revealed significant differences among the position estimation errors between both experiments and motions. Experiment 1 produced the largest estimation errors but was not statistically different than all other experiments ($0.280 \pm 0.191°$), while Experiment 4 produced significantly lower errors than all other experiments ($0.008 \pm 0.003°$ vs. 0.054 ± 0.012–$0.280 \pm 0.191°$, $p \leq 0.001$–0.016). Aside from Experiment 4, Experiment 2 was significantly different from Experiment 1 ($0.054 \pm 0.012°$ vs. $0.076 \pm 0.020°$, $p = 0.005$). Statistically significant differences were found in only 1 of the 45 tests performed between motion groups. The 0–90° flexion motion was significantly different from the 0–120° flexion–extension ($0.242 \pm 0.220°$ vs. $0.031 \pm$

0.047°, $p = 0.028$).

Table 5.4: Accuracy and repeatability measurements generated by the KFMM during Experiments 1–5.

Experiment	Mean Position Error ± Standard Deviation (°)
1	**0.280 ± 0.191***
2	**0.054 ± 0.012***
3	**0.105 ± 0.103***
4	**0.008 ± 0.003**
5	**0.076 ± 0.020***

Bolded values indicate statistically significant differences between the levels of the factors. Bolded cells which include an * indicate results where not all pairwise tests were statistically significant.

Looking across the experiments in Table 5.4, the results do not show a consistent improvement even though the experiments were conducted sequentially and aimed at building upon the successes of the previous experiment. This is due to variations in optimization procedures, elbow motion sensing systems, and elbow motions conducted by the subjects. The results of Experiment 2 were an improvement on those in Experiment 1, due largely to the fact that the KFMM parameters, R and Q, were optimized for each subject and the torque model was removed from the KFMM. In Experiment 1, KFMM parameters were held as constant values across all subjects. Experiment 3 shows a worse overall accuracy than in Experiment 2, due to the introduction of both flexion and extension movements, a different elbow position sensing system, and an optimization variation. Experiment 1 and Experiment 2 only considered the flexion movement of the elbow and used the Biosignalsplux system to collect elbow motion, while Experiment 3 consisted of multiple repetitions of flexion–extension motions using the Trigno Wireless System to collect elbow motion. Experiment 3 also performed optimization of the KFMM parameters on one repetition of elbow motion and then was used to track three repetitions sequentially. This produced larger errors due to the cumulative errors experienced across three repetitions versus those of the single repetitions used in the previous two experiments. Due to these factors, it was decided that a better elbow motion

Table 5.5: Comparison of motions completed during the software simulation portion of Experiments 1–5.

Range of Motion (°)	**Number of Repetitions**	**Additional Mass (kg)**	**Mean Position Error ± Standard Deviation (°)**
Flexion			
0–45	1	0	0.097 ± 0.065
0–90	1	0	**0.242 ± 0.220***
0–105	1	0	0.063 ± 0.084
Flexion–Extension			
0–45	1	0	0.064 ± 0.084
0–90	1	0	0.074 ± 0.119
0–120	1	0	**0.031 ± 0.047***
0–120	1	1	0.070 ± 0.014
Elbow Flexion–Extension with Varied Starting Angles			
0–120	3	0	0.075 ± 0.014
Flexion–Extension with Pauses			
0–120	1	0	0.057 ± 0.014
Extension–Flexion			
0–120	1	0	0.101 ± 0.013

Bolded values indicate statistically significant differences between the levels of the factors. Bolded cells which include an * indicate results where not all pairwise tests were statistically significant.

sensing system was needed and that optimization and tracking would both be conducted on single repetitions of elbow motions.

For Experiment 4 and Experiment 5, the Collection Arm sensing system was used to provide a direct measurement of elbow position as opposed to the previous method of deriving joint position from linear accelerations. This, combined with other experiment improvements, resulted in a significant decrease in position estimation errors compared to the previous experiments. The KFMM was originally developed using slow simple flexion–extension movements with no additional

load held by the subjects. The results of Experiment 4 were measured under these conditions. In order to assess limitations of the KFMM, subjects participating in Experiment 5 performed more dynamic elbow movements. The results of Experiment 5 show larger errors than those seen in Experiment 4. However, this was expected given that motions were used for which the KFMM was not initially designed. Even with these motion input variations, the KFMM produced a position estimation error much lower than both the original KFMM developed by Kyrylova *et al.* and any other elbow motion model in which position estimation errors were reported.

5.4.1.2 Comparing Muscle Activation Models

The main purpose of Experiment 6 was to evaluate and compare the performance of the muscle activation models using EMG inputs collected during various elbow motions. To accomplish this, the experiment was developed purely as an optimization task. The specific goal of the optimization was to produce the highest accuracy in estimation for each muscle activation model while quantifying the computational resources required to achieve these results.

During the experiment, 78 elbow motion datasets were collected (13 per subject). These datasets were used in the inverse optimization procedure to generate 78 muscle activation trajectories used as the controls. For each of these control trajectories, seven muscle activation trajectories, one for each muscle activation model, were optimized using the forward optimization procedure. This resulted in 546 muscle activation trajectories to be used in the evaluation. During the forward optimization procedure, the total muscle activation and standard deviation, joint torque error and standard deviation, data point optimization time, processor usage, and program size metrics, described in Chapter 3, Section 3.4.3, were measured. Average results for total muscle activation error, joint torque error and data point optimization time grouped by muscle activation model, subject motion, and number of optimization parameters are listed in Table 5.6. It important to note that the number of optimization parameters in analysis is for a single muscle activation model. However, two of these muscle activation models in the HTMM, meaning that ultimately, the developed HTMM is optimized as either a two-parameter and four-parameter model.

A statistical analysis was performed using Kruskal–Wallis H tests since the collected data exhibited non-normal distributions. The Kruskal–Wallis H tests were conducted with model,

motion and number of optimization parameters as fixed factors and the total muscle activation error, the total muscle torque error and the data point optimization time metrics as the dependent variables. Thirty-one pair-wise Kruskal–Wallis H tests were conducted using IBM SPSS Software (Version 24) based on 7 muscle activation model groups, 5 motion groups and 2 optimization parameter groups. Statistical significance was determined using an α value of 0.05.

Table 5.6: Comparison of the total muscle activation error, joint torque error, and data point optimization time metrics.

Model ID	**Total Muscle Activation Error ($\times 10^{-3}$)**	**Total Muscle Torque Error (Nm)**	**Data Point Optimization Time (μs)**
	Model Averages		
MAM 1	**0.706 ± 0.947** *	2.15 ± 1.42	**99.75 ± 17.69**
MAM 2	**0.706 ± 0.947** *	2.15 ± 1.42	**156.09 ± 23.42**
MAM 3	0.835 ± 1.086	2.10 ± 1.49	**255.56 ± 212.66**
MAM 4	0.901 ± 1.151	2.17 ± 1.50	**766.40 ± 803.18**
MAM 5	**0.946 ± 1.128** *	**2.19 ± 1.47** *	**301.88 ± 329.06** *
MAM 6	0.702 ± 1.064	**1.67 ± 0.74** *	**266.61 ± 116.58** *
MAM 7	**0.974 ± 1.179** *	**2.19 ± 1.50** *	**375.38 ± 356.94**
Motions	**Motion Averages**		
Flexion–extension	0.684 ± 0.516	1.83 ± 0.83	392.17 ± 621.58
Extension–flexion	**0.718 ± 0.654** *	1.88 ± 0.87	326.37 ± 430.08
Flexion–extension with varied starting angles	**0.489 ± 0.358** *	**1.71 ± 0.87** *	281.55 ± 288.17
Flextion–extension with pauses	**0.622 ± 0.450** *	**2.02 ± 1.03** *	350.04 ± 511.41
Flexion–extension with additional mass	**1.533 ± 1.875** *	**2.84 ± 2.21** *	295.84 ± 342.42
Number of Parameters	**Averages according to the Number of Parameters**		
1	**0.706 ± 0.944**	2.15 ± 1.42	**127.92 ± 35.02**
2	**0.871 ± 1.121**	2.06 ± 1.38	**393.17 ± 471.87**
Total	0.824 ± 1.075	2.09 ± 1.39	317.38 ± 416.73

Bolded cells indicate statistically significant differences in the results. Bolded cells that include an * indicate results where not all pairwise tests were statistically significant.

An average total muscle activation error of 0.00082 ± 0.00108, across datasets, was determined from the experiment. Considering that the total muscle activation is a normalized value of two opposing muscles, whose summed value could range between -1 and 1, this average error represents a percentage of 0.041% of this range. Joint torque errors, derived from the optimized muscle activation model trajectories, resulted in a 2.09 ± 1.39 Nm error across all datasets. An average torque error of 3.4–4.2 Nm, shown in the motion model developed by Cavallaro *et al.*, suggests that the torque errors determined in this experiment are a small improvement [135]. Figure 5.9 shows examples of the flexion–extension (top graph) and the flexion–extension with an additional 1 kg load (bottom graph) for the control and optimized trajectories of each model.

The data point optimization time of the forward optimization procedure was 317 ± 416 μs on average. The longest optimization time during the forward optimization procedure was 616 s. In terms of processing requirements, the average processor usage was measured to be 100.0013% of a possible 400% available quad-core processing time, while requiring an average of 94.91 MB of program space to execute the forward optimization trials in the MATLAB environment. It should be noted that the algorithms were not implemented as a solution that fully utilizes multi-core processing. The duration of one inverse optimization process ranged approximately from 1–4 h.

The statistical analysis of these results indicated many significant differences between models, motions, and the number of optimization parameters. Total muscle activation errors showed statistically-significant differences between models, motions, and number of parameters. Both MAM 1 and MAM 2 showed statistically smaller muscle activation errors than MAM 5 (0.706 ± 0.947×10^{-3} vs. 0.946 ± 1.128×10^{-3}, $p = 0.027$) and MAM 7 (0.706 ± 0.947×10^{-3} vs. 0.974 ± 1.179×10^{-3}, $p = 0.022$). Regarding motions, the flexion–extension with varied starting angles produced statistically smaller errors than the extension–flexion (0.489 ± 0.358×10^{-3} vs. 0.718 ± 0.654×10^{-3}, $p = 0.013$), the flexion–extension with additional mass (0.489 ± 0.358×10^{-3} vs. 1.533 ± 1.875×10^{-3}, $p = 0.003$), and the flexion–extension with pauses (0.489 ± 0.358×10^{-3} vs. 0.622 ± 0.450×10^{-3}, $p = 0.012$) motions. The number of optimization parameters showed a statistical difference (0.706 ± 0.944×10^{-3} vs. 0.871 ± 1.121×10^{-3}, $p = 0.008$) between one-parameter models (MAM 1 and MAM 2) and two-parameter models (MAMs 3–7), with one-parameter models exhibiting smaller errors. However, this statistical significance between the number of parameters

did not carry through to the total muscle torque error.

Statistical differences were observed between the models and the motions for the joint torque error metric. MAM 6 showed a statistically-significant lower torque error compared to MAM 5 (1.67 ± 0.74 Nm vs. 2.19 ± 1.47 Nm, $p = 0.039$) and MAM 7 (1.67 ± 0.74 Nm vs.2.19 ± 1.50 Nm, $p = 0.049$), but was not statistically different from the other models in this metric. The torque error for the flexion–extension with varied starting angles motion was statistically lower than the flexion–extension with additional mass (1.71 ± 0.87 Nm vs. 2.84 ± 2.21 Nm, $p = 0.004$) and the flexion–extension with pauses (1.71 ± 0.87 Nm vs. 2.02 ± 1.03 Nm, $p = 0.007$), similar to the total muscle activation error, but was not statistically different than the flexion–extension or extension–flexion motions. Data point optimization time exhibited differences between the models and the number of optimization parameters, but not between motions. All models showed statically-significant differences in data point optimization time ($p < 0.001$), except between MAM 5 and MAM 6. There was also a statistically-significant difference in data point optimization time between muscle activation models with one- and two-optimization parameters (127.92 ± 35.02 μs vs. 393.17 ± 471.87 μs, $p < 0.001$), as shown in Figure 5.10.

5.4.1.3 Evaluation of Elbow Motion Models

The main objective for Experiment 7 was to facilitate the comparison of three elbow motion models, the KFMM, PMM, and NPMM, during an elbow motion tracking task. In this experiment, the tracking performance was measured for both the motion estimation (software simulation case) and the motion generation (online remote-controlled case). Here, the analysis and discussion surrounds the software simulation case, while the online remote-controlled portion of Experiment 7 is presented in Section 5.4.3. During this experiment, subjects performed a series of elbow motion tasks, which were used as input to the developed control system. First, a MVC and optimization motion were performed. The calibration motion was used for optimization of the models, while the MVC motion was used for both optimization and estimation of elbow position during the tracking portion of the experiment. Next, the optimization procedure was performed on each of the three motion models to determine optimal model parameters. Finally, the optimized model parameters were imported to the control software and the subjects performed a series of elbow motion tasks, in

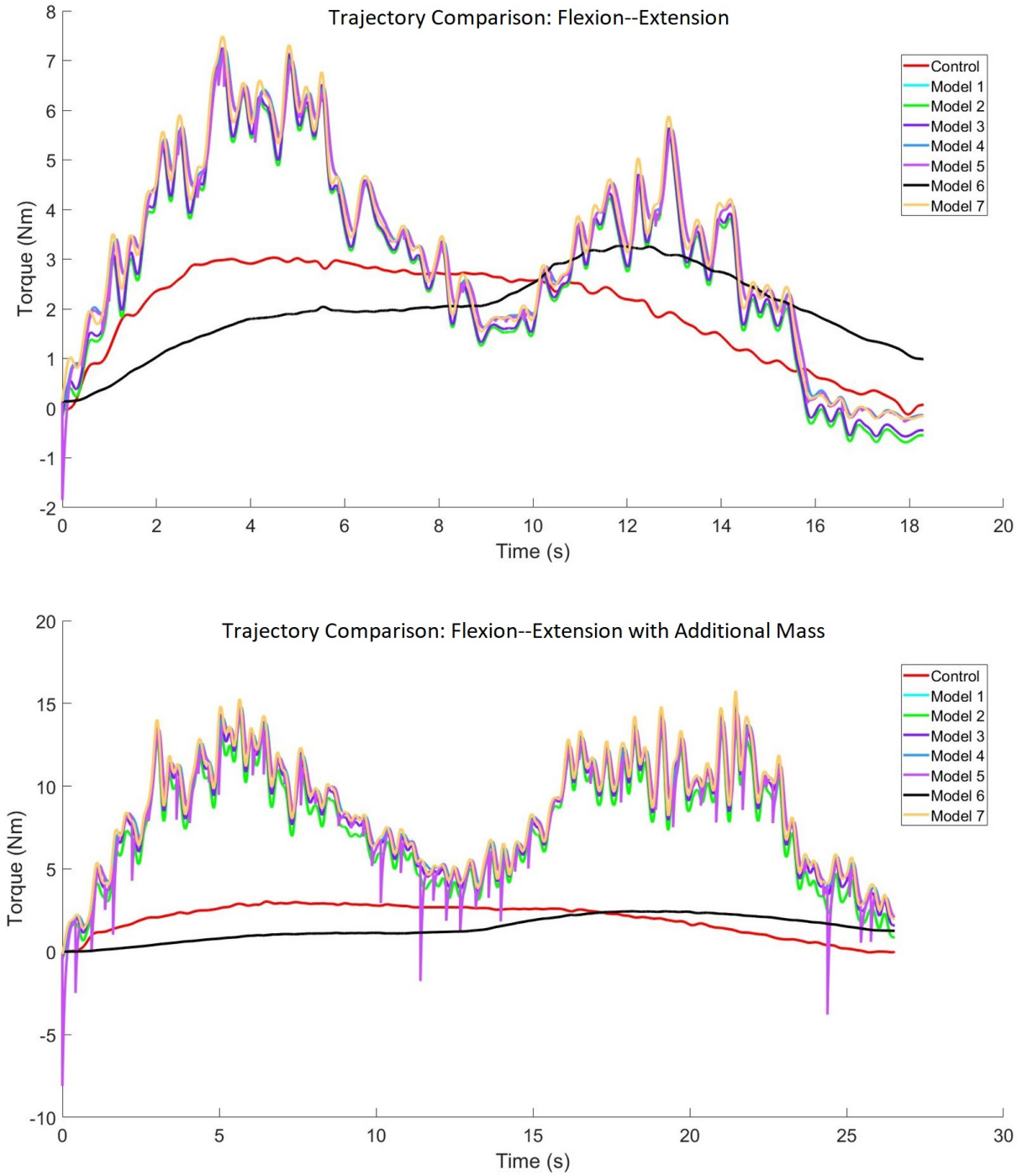


Figure 5.9: The variance in torque estimates with respect to the optimized torque is shown for both the flexion–extension (top graph) and the flexion–extension with an additional 1 kg load (bottom graph) motions from a representative subject.

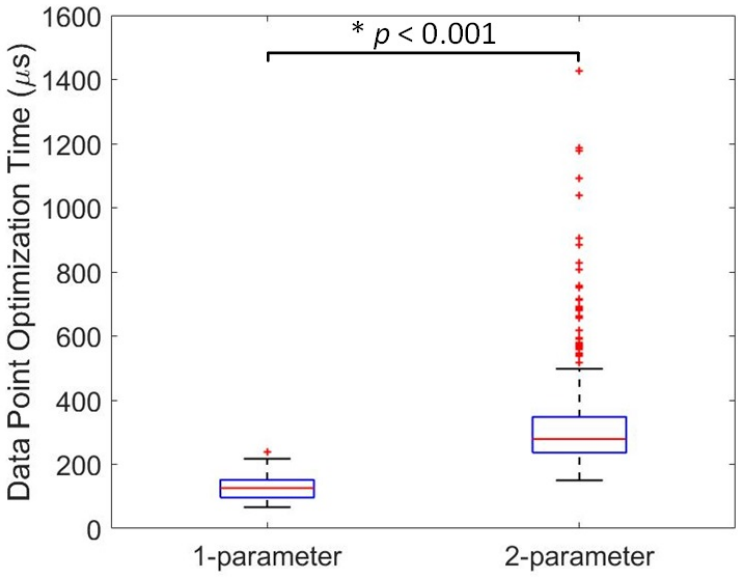


Figure 5.10: Data point optimization times vary with a statistically-significant difference with respect to the number of optimization parameters of each muscle activation model.

which the WearME elbow brace attempted to follow their motion under real-time constraints. All motions completed in Experiment 7 were elbow flexion–extension motions and subjects attempted to stay within the 0–120° range of motion. For this experiment, slow motion denotes a maximum elbow velocity of 10°/s, while fast motion considers a maximum velocity of 40°/s.

For the software simulation case, two metrics were measured during each of the optimization and estimation portions of the experiment. For the optimization procedure, the position estimation RMSE of the optimized model and the data point optimization time were collected to determine the accuracy and adaptability of these models during this procedure. Table 5.7 presents these metrics for each of the models. During the tracking portion, the mean and standard deviation of the position estimation RMSEs were taken to represent the tracking accuracy and repeatability. These metrics are presented in Table 5.8 and Table 5.9 to provide a comparison between the implemented motions and models, respectively.

A statistical analysis was performed to determine any differences between the models during the optimization portion of the experiment and between the motions and models during the tracking portion of the experiment. The position errors were used as the dependent variables for both the optimization and tracking analysis, while the data point optimization time was used a second

dependent variable for the optimization analysis. The Kruskal–Wallis H test was used to determine statistical differences, as the data sets exhibited non-normal distributions and the factors contained more than two independent groups. Overall, 6 pair-wise tests were performed between the 3 models used in the optimization and 18 pair-wise tests were performed between the 6 motions and 3 models used for motion tracking. The analysis was performed using IBM SPSS Software (Version 24) and statistical significance was determined using an α value of 0.05.

The analysis of the optimization portion of Experiment 7 shows that differences exist between the models for both the position estimation RMSEs and the data point optimization time metrics. All three of the models have significantly different position estimation errors from each other ($p \leq 0.016$). This means that the models are ranked as KFMM, NPMM, and PMM, in order of lowest to highest position estimation errors. As shown in Table 5.7, the KFMM was able to produce the lowest position estimation errors on average (0.535°) but had the largest variance (6.157°) compared to the other two models. For data point optimization times, it was shown that the NPMM produced times that were significantly longer from those produced by the KFMM (626.092 ± 203.082 μs vs. 165.400 ± 26.048 μs, $p = 0.002$) and the PMM (626.092 ± 203.082 μs vs. 108.318 ± 217.810 μs,$p = 0.001$). Comparing these models in terms of the number of optimization parameters, this means that the two-parameter model (PMM) and the three-parameter model (KFMM) produced significantly faster optimization times than the eight-parameter model (NPMM). Fig. 5.11 provides an example of the motion trajectories generated from the optimized models. For comparison, the trajectories generated, for the same motion as the calibration motion, during the tracking portion of the experiment are presented in Fig. 5.12.

Statistical analysis of the motion tracking tasks for the software simulation cases revealed significant differences between the motions used in this experiment. The slow flexion–extension motion was significantly different, in terms of position estimation error, than the other five motions (4.841 ± 4.924° vs. 5.282 ± 3.637–30.375 ± 86.315°, $p \leq 0.029$), with the exception of the fast flexion–extension motion. This motion also had the lowest position estimation RMSE (4.841 ± 4.924°) of all of the motions. Two other tests produced statistically significant differences between motions, which included those between the fast flexion–extension motion and both of the 10-repetition flexion–extension (17.526 ± 42.824° vs. 5.816 ± 4.817°, $p = 0.027$) and 10-repetition

Table 5.7: The RMSE of the estimated position and the data point optimization time were measured during the optimization procedure of Experiment 7 to represent the accuracy and adaptability of the models during the optimization procedure. Both a lower RMSE and data point optimization time indicate a better optimization performance.

Model	Mean Value ± Standard Deviation
Position Estimation Error (°)	
KFMM	**0.535 ± 6.157**
PMM	**12.123 ± 3.341**
NPMM	**5.918 ± 2.221**
Data Point Optimization Time (μs)	
KFMM	**165.400 ± 26.048***
PMM	**108.318 ± 217.810***
NPMM	**626.092 ± 203.082***

Bolded values indicate statistically significant differences between the levels of the factors. Bolded cells which include an * indicate results where not all pairwise tests were statistically significant.

flexion–extension with additional load ($17.526 \pm 42.824°$ vs. $30.375 \pm 86.315°$, $p = 0.019$) motions. Analyzing between models, it was found that all three models are significantly different to each other ($p \leq 0.001$) in terms of position estimation errors. As a result, the ranking of these models would be PMM, KFMM, and NPMM in order from lowest to highest RMSE. Interestingly, this ranking does not match the one determined in the optimization portion of the experiment. Aside from the statistical differences, the overall position estimation errors were much larger than those determined during Experiments 1–5.

5.4.2 Offline Remote Control Evaluation

Offline remote control testing was performed as part of Experiments 1–5. During these experiments, data were collected from subjects performing elbow motions and were saved in digital files. Data collection was performed separately from the motion tracking in these experiments. For each of the experiments, one calibration motion was used to optimize the KFMM model of Implementations I and II. Once the optimized model parameters were determined, they were imported into

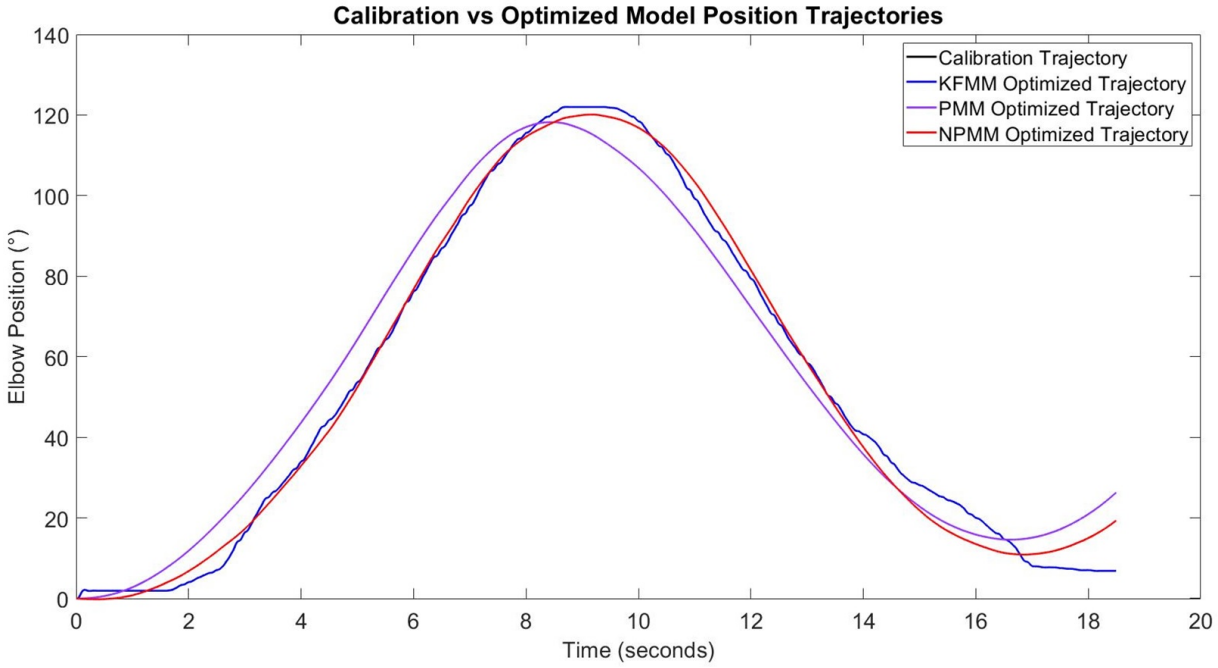


Figure 5.11: An example of the motion trajectories generated by the optimized motion models based on the calibration motion collected during Experiment 7. A single repetition of fast flexion–extension with no load motion was recorded as the calibration motion. The difference between the calibration motion and the KFMM optimized trajectory is so small that the calibration trajectory cannot be seen on the figure.

the control software implementations and used, together with the data stored in the digital files, to generate motion of either the WearME elbow brace (Experiments 1, 2, 4, and 5) or the Active A-Gear elbow joint (Experiment 3). The goal of these control systems was to track the subjects' elbow motion through the regulation of either the position or the velocity of the devices' elbow joint. The advantage of conducting this tracking task in an offline remote controlled scenario is that the motion input data sets can be reused. This allows for isolation of the control system behaviours from the inputs, since the exact same motion inputs can be given to the control system as many times as desired. Based on the results from the software simulation phase testing, it was clear that the motion inputs had an effect on the tracking performance. Therefore, the offline remote control testing style can be used to control for some factors that affect the motion inputs.

Using data collected during the offline remote control scenarios, a statistical analysis was performed to determine any effects that the motion variations had on the tracking performance. The Kruskal–Wallis H test was used to determine these differences, given that the data exhibited

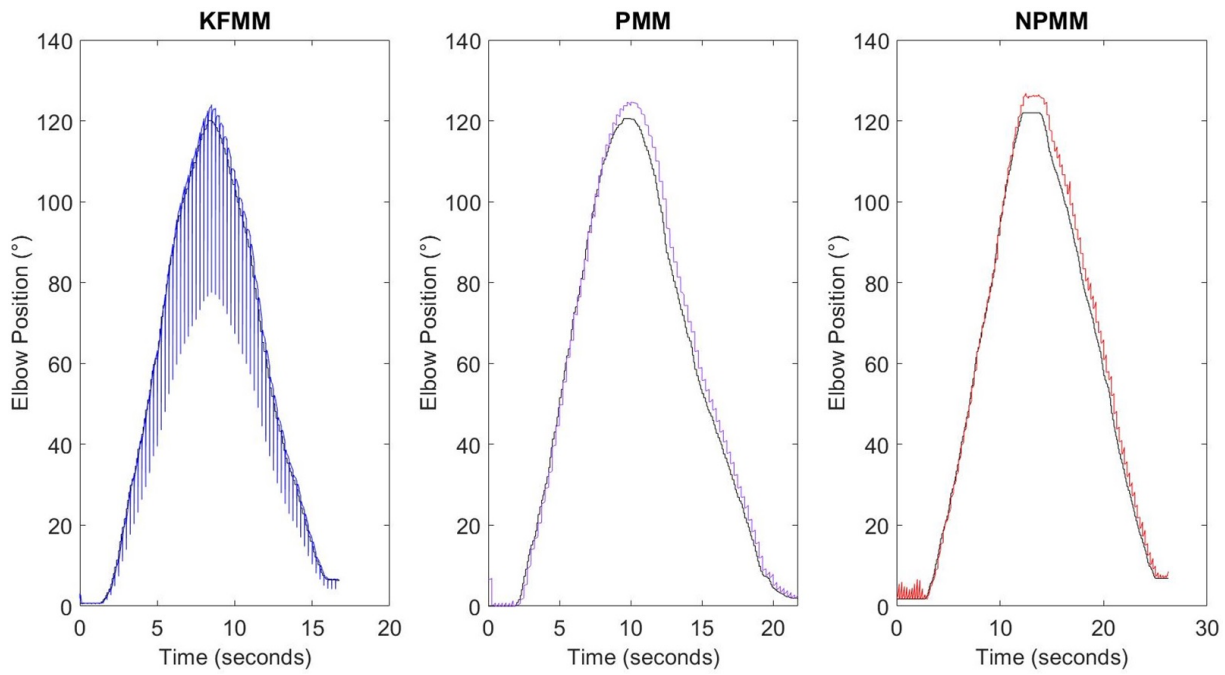


Figure 5.12: An example of the motion trajectories generated by the optimized motion models during the motion tracking portion of Experiment 7. These data sets are generated based on the subject completing one fast flexion–extension motion with no additional load.

non-normal distributions. For this analysis, the motions were used as the fixed factors and the error between the subject and brace positions was used as the dependent variables. In total, 45 pair-wise tests were performed between the 10 motions that were used across Experiments 1–5. A Bonferonni correction was used to adjust the significance due to the comparison across experiments. The IBM SPSS Software (Version 24) was used to perform the analysis, basing significance on an α value of 0.05. The results of this analysis are presented in Table 5.10.

Compared to these same analyses of these data for the software simulation phase, many more statistically significant differences were found between the motions performed during Experiments 1–5. In total, 13 of 45 pair-wise tests revealed significant differences. However, no single motion was statistically different than all other motions. In general, there was one major difference that was found between the motions in the analysis. All statistical differences occurred between pairs of motions in which one performed only elbow flexion and the other performed both flexion and extension. The 0–45° flexion–extension, 0–120° flexion–extension with additional mass, 0–120° flexion–extension with varied starting angles, and the 0–120° flexion–extension with pauses

Table 5.8: Comparison of the accuracy and repeatability across the different motions conducted during the software simulation tracking portion of Experiment 7. Slow motion considers a maximum elbow velocity of 10°/s, while fast motion denotes a maximum velocity of 40°/s.

Additional Mass (kg)	**Mean Position Error ± Standard Deviation (°)**
Slow Flexion–Extension	
0	**4.841 ± 4.924***
2	**5.282 ± 3.637***
Fast Flexion–Extension	
0	**17.526 ± 42.824***
2	**21.103 ± 55.880***
10-Repetition Flexion–Extension	
0	**5.816 ± 4.817***
2	**30.375 ± 86.315***

Bolded values indicate statistically significant differences between the levels of the factors. Bolded cells which include an * indicate results where not all pairwise tests were statistically significant.

Table 5.9: Comparison of position RMSEs for each of the models used in the software simulation of Experiment 7.

Model	**Mean Position Error ± Standard Deviation (°)**
KFMM	**8.068 ± 14.471**
PMM	**4.317 ± 1.094**
NPMM	**30.086 ± 76.022**

Bolded values indicate statistically significant differences between the levels of the factors.

motions produced statistically lower estimation errors than all of three of the flexion-only motions (p 0.005–0.031). The 0–120° extension–flexion motion was also statistically different from the 0–120° flexion motion (0.716 ± 0.088 vs. 10.892 ± 3.061, $p = 0.032$)

It is important to remember that, for the offline remote controlled scenarios, it was found that

the use of visual feedback, allowing motion training, constraining subject motion, and regulating position of the device all produce statistically better accuracies. The results of the current analysis suggest that the motions play a factor into the tracking performance, as one may expect. However, the likelihood of the experimental protocol and control output variations affecting these results is high. For example, Experiments 1 and 2 involved no visual feedback, no training, unconstrained motion, and velocity control, which were all shown to produce statistically higher position errors. The motions of these two experiments were different than the ones for which the KFMM was designed and support the conclusion that the motions themselves also affect the accuracy. However, the Experiment 5 motions were more different from the motion for which the KFMM was designed than the flexion-only motions. However, these more dynamic motions produced the best accuracies among the motion groups. Experiment 5 was also completed with visual feedback, motion training, constrained motion, and regulation of position. Therefore, it is hard to distinguish the exact interaction of these factors with the motion factor and the role that each factor plays in the elbow motion tracking performance.

Although the results of the statistical analysis are not conclusive, they support the idea that the generation of device motion during elbow motion tracking tasks is dependent on many factors. In addition, the average position errors determined through the offline remote control testing were lower than those found during the software simulation motion tracking studies performed by Bai *et al.* (5.057°) and Gao *et al.* (4.16°) [128, 129]. The results were also similar to those found by Kyrylova in each of the software simulation (1.82–3.38°) and offline remote controlled (2.65–5.62°) studies that used the KFMM [25]. Considering only the results of Experiment 5, the tracking performance accuracy was better than any of the reviewed studies, regardless of integration testing phase. The main goal of Experiments 1–5 was to improve upon the tracking accuracy of existing elbow motion models and that goal was met, as supported through these results.

5.4.3 Online Remote Control Evaluation

Online remote control testing is important for measuring the performance of wearable assistive devices and their control systems under real-time motion inputs. Experiment 7 was the only one of the experiments in which online remote controlled motion tracking was performed. This involved

Table 5.10: Comparison of motions completed during the offline remote controlled portion of Experiments 1–5.

Range of Motion (°)	**Number of Repetitions**	**Additional Mass (kg)**	**Mean Position Error ± Standard Deviation (°)**
Flexion			
0–45	1	0	**8.504 ± 3.556***
0–90	1	0	**8.054 ± 2.551***
0–105	1	0	**10.892 ± 3.061***
Flexion–Extension			
0–45	1	0	**1.625 ± 0.882***
0–90	1	0	3.032 ± 2.178
0–120	1	0	2.670 ± 1.871
0–120	1	1	**0.560 ± 0.157***
Elbow Flexion–Extension with Varied Starting Angles			
0–120	3	0	**0.437 ± 0.094***
Flexion–Extension with Pauses			
0–120	1	0	**0.527 ± 0.123***
Extension–Flexion			
0–120	1	0	**0.716 ± 0.088***

Bolded values indicate statistically significant differences between the levels of the factors. Bolded cells which include an * indicate results where not all pairwise tests were statistically significant.

the collection of subject position and EMG signals that were immediately used to estimate the user's desired motion, command the actuation systems, and generate motion of the device's elbow joint. Essentially, the subject is controlling the motion of the device under real-time reactions to their motion signals. The experiment involved the performance of elbow motion tracking tasks, while using one of the KFMM, PMM, and NPMM components at a time. Based on the reviewed literature, no studies involving motion tracking in an online remote controlled scenario have been conducted. Furthermore, none of these studies used a NPMM for dynamics muscle torque estima-

tion or proposed PMMs to represent muscle dynamics. The two main objectives of this experiment were to 1) quantify the motion tracking abilities of a wearable assistive elbow device during an online remote controlled tracking scenario, and 2) to compare the abilities of three different elbow motion models to perform these tracking tasks. Through the successful completion of this experiment, both of the objectives were met.

Quantifying and comparing motion tracking performance was completed using position error data, which were computed between the subject position and both of the commanded position and the generated brace position. Based on the results of the previous experiments, it was decided that subjects would be provided visual feedback of their position, have time to train the motions, and have their elbow motion constrained during the experiment. Furthermore, the control system was developed to regulate the position of the device's elbow joint. All of these factors were shown to be associated with the production of lower position errors. Using the command and generated position errors, a statistical analysis was performed to compare between the motions and between the models used in this experiment. Due to non-normal distributions, the Kruskal–Wallis H test was used to determine statistically significant differences between these factors. The analysis was performed in the IBM SPSS Software (Version 24), using an α value of 0.05 to determine significance.

The statistical analysis reveals significant differences across both the commanded position error and the brace position error. Based on the commanded position error, it was found that the slow flexion–extension motion produced significantly lower errors than the 10-repetition flexion–extension motion ($3.621 \pm 4.673^\circ$ vs. $3.912 \pm 4.496^\circ$, $p = 0.013$). It was found that all three motion models were statistically different from each other, in terms of commanded position error ($p \leq 0.001$). The commanded position was determined as the average estimated position per data window. Therefore, these results follow naturally from the software simulation results, in which all three models were also significantly different from each other in terms of estimated position errors.

Considering the brace position error metric, statistically significant differences were found between the motions. The slow flexion–extension motion was statistically different from all other motions ($8.377 \pm 2.823^\circ$ vs. 15.119 ± 4.448–$17.729 \pm 12.407^\circ$, $p < 0.001$), except the fast flexion–

extension motion. Aside from the slow flexion–extension motion, the fast flexion–extension motion was also statistically different than the other motions ($11.676 \pm 13.706°$ vs. 15.119 ± 4.448–$17.729 \pm 12.407°$, $p < 0.001$). This same trend was found during the software simulation analysis, though the statistical differences were not as pronounced as they are here. Lastly, there were no statistically significant differences found between the models, based on the brace position error metric. The results of this analysis are presented in Tables 5.11 and 5.12.

An interesting trend can be seen through this analysis when looking at the differences between motions. For both the commanded and brace position errors, there are no significant differences between the slow flexion–extension and the fast flexion–extension motions in which there was no additional load. This was also true for the software simulation analysis of Experiment 7. Combining these results, the trend shows that either the slow flexion–extension or both of these motions are also significantly different from the other motions, when looking across the estimated, commanded, and brace position errors. This suggests that the motion velocity may not have an impact on performance, when the velocity is less than $40°/s$. It also supports the claim that, in general, additional load and number of repetitions affect these models.

Comparing the differences in the models between the commanded and brace position errors raises two points of interest. First, statistical differences are found between the models for the commanded position error metric, but not for the brace position error metric. This difference suggests that either the actuation-level control components, the device's actuation system, or both affected the tracking performance. On average, these components produced higher tracking errors for both the KFMM and PMM and lower tracking errors for the NPMM. It was expected that the actuation software and hardware would affect the tracking performance, but not such that the statistical difference between the models would follow the trend found in the estimated and commanded position errors. Second, ranking the models based on average position errors, from lowest to highest, did not match the same ranking performed during the optimization portion of Experiment 7. This was the same result as exhibited in the software simulation tracking analysis. Together, these results emphasize the need for better understanding of humans and their motion.

Table 5.11: Comparison of position errors for each of the motions used in the offline remote-controlled portion of Experiment 7. During this tracking task, the postion error between the command motion and the subject motion and between the resultant brace motion and the subject motion is presented. Slow motion considers a maximum elbow velocity of 10°/s, while fast motion denotes a maximum velocity of 40°/s.

Additional Mass (kg)	**Mean Position Error ± Standard Deviation (°)**
Commanded Position Error	
Slow Flexion–Extension	
0	**3.621 ± 4.673***
2	3.605 ± 3.74
Fast Flexion–Extension	
0	12.981 ± 31.691
2	15.049 ± 39.803
10-Repetition Flexion–Extension	
0	**3.912 ± 4.496***
2	21.644 ± 61.924
Brace Position Error	
Slow Flexion–Extension	
0	**8.377 ± 2.823***
2	**15.119 ± 4.448***
Fast Flexion–Extension	
0	**11.676 ± 13.706***
2	**15.536 ± 11.280***
10-Repetition Flexion–Extension	
0	**15.128 ± 4.275***
2	**17.729 ± 12.407***

Bolded values indicate statistically significant differences between the levels of the factors. Bolded cells which include an * indicate results where not all pairwise tests were statistically significant.

Table 5.12: Comparison of position errors for each of the models used in the offline remote-controlled portion of Experiment 7. During this tracking task, the position error between the command motion and the subject motion and between the resultant brace motion and the subject motion is presented.

Model	**Mean Position Error ± Standard Deviation (°)**
Commanded Position Error	
KFMM	**8.068 ± 14.471**
PMM	**2.635 ± 0.380**
NPMM	**21.054 ± 54.044**
Brace Position Error	
KFMM	12.462 ± 6.788
PMM	12.640 ± 4.985
NPMM	16.681 ± 14.070

Bolded values indicate statistically significant differences between the levels of the factors.

5.4.4 Effects of Experimental Factors on Motion Tracking

During performance of the elbow motions by subjects, it became apparent that aspects of the experimental protocol could have effects on the tracking accuracy. Three factors were varied between these experiments, which included visual feedback, motion training, and motion constraints. These factors all affect the motion inputs to the system, as they change how the individual performs the intended motion. As a result, they will have effects on the tracking performance of the system as well.

Measured joint position errors for both software simulation and offline remote-controlled motion tracking cases were used to indicate the possible effects these factors have on the accuracy and repeatability of the control systems. Only data collected from Experiment 1–5 were used in the analysis of the experimental factors. This is due to the fact that these experimental protocols and the control system implementations differed significantly from those used in Experiment 6 and 7. A statistical analysis was performed to determine any statistically significant differences between the experimental factors. The position errors collected from both the software simulation and offline remote-controlled portions of Experiments 1–5 were used as the dependent variables. The

Mann-Whitney U test was used as the data exhibited non-normal distributions and the factors contained two independent groups. A Bonferroni correction was applied to account for comparisons across different experiments. One pair-wise test were performed for each of the three experimental factors using IBM SPSS Software (Version 24). Statistical significance was determined using an α value of 0.05. The results of this analysis are presented in Table 5.13.

Table 5.13: Comparison of the effects of visual feedback, training, and motion constraint factors on the motion tracking accuracy and repeatability. These values are based on results from Experiments1–5.

Experiment Factor	**Mean Position Error ± Standard Deviation (°)**	**Number of Data Sets**
Software Simulation Case		
Visual Feedback	0.076 ± 0.021	12
No Visual Feedback	0.096 ± 0.134	55
Training	**0.062 ± 0.075**	42
No Training	**0.144 ± 0.163**	25
Contrained Motion	**0.038 ± 0.037**	27
Unconstrained Motion	**0.130 ± 0.143**	40
Offline Remote Controlled Case		
Visual Feedback	**0.570 ± 0.147**	12
No Visual Feedback	**5.338 ± 4.037**	55
Training	**1.914 ± 1.729**	42
No Training	**8.802 ± 3.150**	25
Contrained Motion	**0.930 ± 0.401**	27
Unconstrained Motion	**6.884 ± 3.682**	40

Bolded values indicate statistically significant differences between the levels of the factors.

In general, the results of this analysis reveal that providing visual feedback of the elbow position, allowing for the user to practice the movement, and constraining their motion all resulted in better motion accuracy and repeatability. In this analysis, the mean and standard deviation of the position error are used to indicate the level of accuracy and repeatability, respectively. As shown

in Table 5.13, the position errors are lower when visual feedback, motion training, and constrained motion were used in both the software simulation and offline remote-controlled tracking of elbow motion. There were also statistically significant differences between these experimental factors. First, using visual feedback produced statistically lower position tracking errors than no visual feedback for the offline remote-controlled ($0.570 \pm 0.147^\circ$ vs. $5.338 \pm 4.037^\circ$, $p < 0.001$) case, but was not significantly different for the software simulation case. Second, giving subjects time to train the motion velocities and range of motion resulted in significantly lower position errors than performing them without training for both the software simulation ($0.062 \pm 0.075^\circ$ vs. $0.144 \pm 0.163^\circ$, $p = 0.006$) and offline remote-controlled ($1.914 \pm 1.729^\circ$ vs. $8.802 \pm 3.150^\circ$, $p < 0.001$) cases. Lastly, constraining the elbow motion, using the Collection Arm device, to a two-dimensional plane resulted in position errors that were significantly different than allowing the subjects to perform the elbow motions freely. This was true for both the software simulation case ($0.038 \pm 0.037^\circ$ vs. $0.130 \pm 0.143^\circ$, $p = 0.001$) and the offline remote-controlled case ($0.930 \pm 0.401^\circ$ vs. $6.884 \pm 3.682^\circ$, $p < 0.001$).

The statistically significant differences are consistent with the generally accepted nature of human motion. As humans go through life, most will come to learn from trial and error that visual feedback, training specific motions, and constraining their motion all have an impact on their performance. Therefore, the results of this analysis seems to agree with our intuitive nature. However, there are a few limitations to these conclusions based on factors that were not controlled for across the experiments included in the analysis. First, the lack of statistically significant differences between the visual feedback levels for the software simulation case, compared to the offline remote-controlled case, suggest that the actuation-level control components and the device may have played a large role in the error. Experiment 3 was performed with a different device and the one used for the other experiments and there are differences in the actuation-level components between the different implementations used for Experiments 1–5. Second, the control output mode, position or velocity, differed across these experiments. There were statistically significant differences between these two control output modes (see Section 5.4.5) that could have factored into the differences seen here. Finally, the motions performed during each of these five experiments were not consistent with each other. In Experiments 1 and 2, elbow flexion motions were performed,

while in Experiments 3–5 elbow flexion–extension movements were completed by the subjects. This may factor into the statistically significant differences since the model used in these experiments was developed specifically for elbow flexion–extension motions.

In conclusion, the statistical analysis has shown that 5 of the 6 tests exhibit statistically significant differences that align with the general assumptions about human motion performance. Furthermore, all cases where either visual feedback, motion training, and constrained motions were used, a better accuracy and repeatability was shown. It should be noted that experimental factor consistency is important when comparing across experiments. Future studies of human motion should design methods for controlling as many of these factors as possible.

5.4.5 Comparing Control Outputs

Using the control hardware and elbow devices, six experiments (Experiments 1–5 and 7) were conducted that involved the control of either the velocity or the position of the actuators. The choice of control outputs is an important aspect of these digital control systems, as the goal is to regulate physical-world variables, such as motion variables. These variables often take the form of one of five primary motion variables: position, velocity, acceleration, force, and torque. Using these five variable types, control systems can be compared between their ability to regulate one or more of these control output types. It is possible to isolate portions of the control system to determine their role in the generation of motion. However, it is important to note that, in most cases, the entirety of the control system affects the production of motion. As a result, Experiment 7 will not be included in this analysis, due to major differences in the experimental protocol and control software implementation compared to those of Experiments 1–5. Within the control software developed for Experiments 1–3, the KFMM was used to estimate elbow velocity, while in Experiments 4 and 5 it was used to estimate elbow position. The WearME elbow brace was used to track subject motion in Experiments 1, 2, 4, and 5, while the Active A-Gear was used to track subject motion in Experiment 3. Other differences between these experiments include experimental factors that affect the motion inputs, such as visual feedback, training, and motion constraints, as discussed in Section 5.4.4. Both a software simulation and offline remote-controlled tracking task were performed during each of these five experiments.

To determine the differences between these two control outputs, a statistical analysis was performed on the data collected from Experiments 1–5. For this analysis, the estimated and the generated brace control outputs were measured for each of the software simulation and offline remote-controlled portions of these experiments. Since the data exhibited non-normal distributions, the Kruskal–Wallis H test was used to discover if differences existed between the velocity-driven and position-driven control system variations. The analysis was performed using IBM SPSS Software (Version 24), with statistical significance determined using an α value of 0.05. A Bonferroni correction was applied to account for comparisons across different experiments. This analysis was also repeated for the data from only Experiments 3 and 4, since the same motions were performed by the subjects of both experiments. Table 5.14 contains the results of the comparison between position and velocity control outputs.

These results indicate that there are statistically significant differences between the velocity-driven and position-driven control system variations. Overall, controlling position, compared to velocity, resulted in significantly lower position estimation RMSEs across data from both Experiments 1–5 ($0.038 \pm 0.037°$ vs. $0.130 \pm 0.143°$, $p = 0.001$) and Experiments 3 and 4 ($0.008 \pm 0.003°$ vs. $0.105 \pm 0.103°$, $p = 0.001$). Position control also had significantly lower errors for both Experiments 1–5 ($0.930 \pm 0.401°$ vs. $6.884 \pm 3.682°$, $p < 0.001$) and Experiments 3 and 4 ($1.218 \pm 0.290°$ vs. $3.687 \pm 1.780°$, $p < 0.001$). This was true for both the software simulation and offline remote controlled tracking tasks of these experiments. Position control was not only more accurate, but also more repeatable, as indicated by the standard deviations of the position RMSEs. The similarities between these two analyses suggest that position control is a better option when paired with the KFMM for motion estimation and production. Discovering these differences between Experiment 3 and 4 was of particular interest, since these two experiments differed only by constraint level, device, and control output.

There are other factors that were not all accounted for or tested in all combinations across these experiments, despite the discovery of significant differences. Experiments 1–5 varied in many factors, which include visual feedback, training, motion constraint, device, control output, and motion inputs. The analysis presented in Section 5.4.4 showed that there were statistical differences between the visual feedback, training, and motion constraint factors. Furthermore,

the studies of the software simulation tracking tasks show differences in estimation performance based on variations in motion inputs. These experiments did not account for factors of the subject performing the motions. However, the analysis was conducted across subjects, which increases probability that the statistical differences are due to either the device or the control output. Two different devices were used in the experiments, but the mass of the lower arm cuffs were similar and the motors and motor controllers were made by the same company. Both devices even exhibited a similar amount of play between parts of their transmission systems. More detailed studies are needed to confirm or refute any generalization of these results. However, the analysis does show that there are significant differences between the control output types for the developed control systems during elbow motion tracking experiments.

Table 5.14: Comparison of position and velocity control outputs for both the software simulation and offline remote controlled tracking modes.

Control Output	**Mean Position Error ± Standard Deviation (°)**	**Number of Data Sets**
Experiments 1–5 Software Simulation		
Position	**0.038 ± 0.037**	27
Velocity	**0.130 ± 0.143**	40
Experiments 1–5 Offline Remote Controlled		
Position	**0.930 ± 0.401**	27
Velocity	**6.884 ± 3.682**	40
Experiments 3 and 4 Software Simulation		
Position	**0.008 ± 0.003**	15
Velocity	**0.105 ± 0.103**	15
Experiments 3 and 4 Offline Remote Controlled		
Position	**1.218 ± 0.290**	15
Velocity	**3.687 ± 1.780**	15

Bolded values indicate statistically significant differences between the levels of the factors.

5.4.6 Control Software Timing and Data Storage Needs

The literature surrounding elbow motion models and control systems that employ them to produce motion of wearable assistive devices is lacking in the analysis of the control system implementations. One could argue that the most important quality of these control systems is the accuracy, as it relates to how well the system fulfills its general goal of producing motions. However, as discussed in Section 2.7.3.4, only 42% of the motion model studies quantified accuracy (mean error), while only 27% of the studies quantified repeatability (standard deviation of error). Furthermore, of the reviewed wearable assistive devices, only Kyrylova and Tang *et al.* provided any quantification of accuracy of their devices' motion performance [25, 44]. It is clear that the general analysis of these control systems is severely lacking. The experiments conducted within this thesis served to address this, through quantification of accuracy, repeatability, adaptability, and resource utilization. However, two important metrics for the development of digital control systems have yet to be discussed.

During Experiment 7, the task execution time and data storage velocity were measured during the execution of the elbow motion tracking tasks. The task execution time metric is important to control system developers as it provides data relevant to the scheduling of control system tasks. The task execution time of five important control system tasks were measured during this experiment. The control software implementation, Implementation IV, was developed as a multi-threaded application, wherein each of these five tasks were executed on separate threads. This reduced some of the interaction effects between these tasks and provided more accurate measurements of their independent execution times. The data storage velocity metric was used to provide control system developers with an idea of the data rates at which critical data are generated and need to be stored. This is important as it informs decisions about how to store these data, such as storing it on the device or transferring it to another digital system for storage. No studies involving the quantification of control system tasks and data storage needs for wearable assistive devices could be found in the literature.

The task execution time was measured during the Track Motion, Log Data, Gather Subject Position, Gather Brace Position, and Command Position tasks of Implementation IV. The Track

Motion task constituted the main motion tracking control loop, in which data were gathered from the subject, estimates were calculated, and actuation of the brace and data logging were initiated. Writing the critical data to file was performed under the Log Data task. Gathering the subject position from the Collection Arm and the WearME elbow brace position from the EPOS 24/5 Position Controller were denoted as the Gather Subject Position and Gather Brace Position control system tasks, respectively. Finally, the task which sent position commands to the EPOS 24/5 controller in order to generate motion of the brace, was denoted the Command Position task. The data logged as a result of executing the Log Data task, were the data for which the data storage velocity was calculated. In this dataset, five critical floating-point numbers were recorded to digital files, which included two EMG signals, one from each of the biceps brachii and triceps brachii muscles, the subject's position, the estimated position, and the brace's position. These data were either collected at a sampling frequency of 2000 Hz or collected at a lower frequency and upsampled to 2000 Hz.

A statistical analysis was performed, based on the task execution time data, to determine if either motions or the models used in Experiment 7 would produce any effects. Using the IBM SPSS Software (Version 24), the Kruskal–Wallis H test was performed on these data to determine statistical differences. Comparisons were made with motions and models as the fixed factors, with the task execution time, measured in milliseconds, as the dependent variable, and an α value of 0.05 indicating significance. The results of the experiment and this analysis are presented in Tables 5.15 and 5.16.

Statistically significant differences were found between the motions and models in this analysis. In terms of motions, the differences were discovered for the Log Data and Command Position task execution times. The execution time of the slow flexion–extension motion was statistically shorter than the 10-repetition flexion–extension (1.748 ± 0.151 ms vs. 1.807 ± 0.085 ms, $p = 0.046$) and the 10-repetition flexion–extension with additional load (1.748 ± 0.151 ms vs. 1.829 ± 0.101 ms, $p = 0.017$) motions, for the Log Data task execution time metric. Although the same amount of data were generated per second, the latter two motions took much longer to complete than the former, which increases the chance of errors or delays that could lead to increases in the execution time. The Command Position task execution times of the slow flexion–extension and fast flexion–

extension motions were significantly shorter than all other motions (6.611 ± 0.227–6.551 ± 0.150 ms vs. 6.795 ± 0.480–6.915 ± 0.646 ms, $p \leq 0.03$), but not statistically different from each other. This exact same statistical result was found between the motions when comparing the brace position errors, as discussed in Section 5.4.3.

The task execution times also revealed statistical differences between the three motion models used in the elbow motion tracking experiment. The models were all statistically different from each other for both the Track Motion task ($p \leq 0.005$) and the Log Data task ($p \leq 0.012$). For the Gather Subject Position Task, the KFMM was significantly different than the NPMM (2.442 ± 0.239 ms vs. 2.638 ± 0.400 ms, $p = 0.01$). The KFMM was also significantly different from both the PMM (3.587 ± 0.103 ms vs. 3.738 ± 0.130 ms, $p < 0.001$) and the NPMM (3.587 ± 0.103 ms vs. 3.773 ± 0.113 ms, $p < 0.001$) for the Gather Brace Position task execution times. The KFMM produced the shortest task execution times for all tasks, with the exception of the Log Data task in which it produced the longest times. No statistically significant differences were found between the models for the Command Position task execution times.

The statistical differences between motions of the Command Position task execution times match those of the brace position errors. One might consider this an expected result since the changes in the brace positions are being commanded during the measurement of the task execution time. The results of Experiment 6 revealed that there is a relationship between accuracy of the estimation (mean torque error) and the resource utilization (data point optimization time). More specifically, that there is a trade-off between these two control system qualities during model optimization tasks. The similarities in the statistical differences between motion for both the brace position errors and the Command Position task execution times also support that there is a relationship between the accuracy and resource utilization during motion tracking tasks. However, these results suggest that these two metrics are proportionally related, which is the opposite of the case for the optimization task. This means that a shorter Command Position task execution time would result in a lower position error. One reason for these results could be due to the inherent delays in the control system components, relative to the time at which the subject performed the motion.

The only case in which the models were expected to affect the task execution times significant-

ly, was during the Track Motion task. This is due to the fact that the number of lines of code that constitute each model are different. Ranking these models in order from shortest to longest task execution time results in the following order: KFMM, PMM, and NPMM. Although the results were not statistically significant, comparing the brace position errors between the models produces this same ranking. Combining this observation with the differences found between motions supports the claim that the task execution time and position tracking error are proportionally related.

It was not expected that the data storage velocity would be affected by different experimental factors, since the number of floating-point values and the sampling frequency remained constant across these factors. As a result no statistical analysis was performed on the data collected during Experiment 7. However, the mean and standard deviation of the data storage velocity were generated across all data sets of this experiment. Overall, an average data storage velocity of 176.646 $\pm$ 0.681 KB/s or 1413.168 $\pm$ 5.448 Kb/s was determined from these data. At this data storage velocity, 10.6 MB of data will be generated and need to be stored every minute. It is important to keep in mind that this result was determined based on the generation and storage of only five floating-point numbers. In reality, users of the control system may be interested in collecting many more critical control system variables than ones collected during Experiment 7.

5.5 Discussion: Estimating Human Elbow Motion

The study of human motion through modelling is one of the most important aspects to being able to facilitate interactions between humans and wearable assistive devices. It is clear that developing accurate and adaptable motion models has been the interest of a large research community and is still on ongoing effort. Through the experiments conducted in this work, the knowledge regarding elbow motion models has been advanced. The success of this research is shown in the improved estimation abilities, first-time comparisons, and novel measurements. Through Experiments 1–5, the KFMM estimation has been improved from its initial conception. The HTMM developed for Experiment 6 has also shown torque estimation errors comparable to those of other HTMMs found in the literature. The first comparison of muscle activation models was conducted in Experiment 6

Table 5.15: Comparison of the task execution times for each of the five critical control system tasks.

Additional Load (kg)	**Mean Task Execution Time ± Standard Deviation (ms)**
Track Motion Task	
Slow Flexion–Extension	
0	9.112 ± 1.260
2	9.250 ± 1.341
Fast Flexion–Extension	
0	9.036 ± 1.207
2	8.894 ± 1.057
10-Repetition Flexion–Extension	
0	9.170 ± 1.164
2	9.135 ± 1.109
Log Data Task	
Slow Flexion–Extension	
0	**1.748 ± 0.151***
2	1.800 ± 0.178
Fast Flexion–Extension	
0	1.798 ± 0.149
2	1.836 ± 0.249
10-Repetition Flexion–Extension	
0	**1.807 ± 0.085***
2	**1.829 ± 0.101***
Gather Subject Position Task	
Slow Flexion–Extension	
0	2.526 ± 0.347
2	2.540 ± 0.365

Additional Load (kg)	**Mean Task Execution Time ± Standard Deviation (ms)**
Gather Subject Position Task	
Fast Flexion–Extension	
0	2.541 ± 0.383
2	2.586 ± 0.372
10-Repetition Flexion–Extension	
0	2.525 ± 0.306
2	2.604 ± 0.311
Gather Brace Position Task	
Slow Flexion–Extension	
0	3.671 ± 0.140
2	3.690 ± 0.175
Fast Flexion–Extension	
0	3.702 ± 0.159
2	3.687 ± 0.160
10-Repetition Flexion–Extension	
0	3.714 ± 0.094
2	3.733 ± 0.105
Command Position Task	
Slow Flexion–Extension	
0	**6.611 ± 0.227***
2	**6.915 ± 0.646***
Fast Flexion–Extension	
0	**6.551 ± 0.150***
2	**6.826 ± 0.492***

Additional Load (kg)	Mean Task Execution Time ± Standard Deviation (ms)
Command Position Task	
10-Repetition Flexion–Extension	
0	**6.854 ± 0.405***
2	**6.795 ± 0.480***

Bolded values indicate statistically significant differences between the levels of the factors. Bolded cells which include an * indicate results where not all pairwise tests were statistically significant.

as well. Experiment 7 is the first comparison of motion models that are neither HTMMs or ANNs. The first use of a NPMM for dynamic motion estimation was also accomplished through this experiments. Finally, new metrics for measuring adaptability and resource demand were proposed and measured in Experiments 6 and 7, expanding the possible analysis and understanding of these motion models.

5.5.1 Motion Estimation with the Kalman Filter Model

The goal of the development and improvement of the KFMM was to achieve better accuracy and repeatability compared to existing elbow motion models whose output was the elbow joint position. In its initial conception by Kyrylova *et al.*, the KFMM produced an mean joint position error in the range of 1.82–3.38° [123]. This result was already an improvement on other elbow motion models. Gao *et al.*, Bai *et al.*, and Ding *et al.* produced mean joint position errors of 4.16°, 5.06°, and 8.14 ± 1.53°, respectively. Throughout the experiments conducted in this research, the mean joint position errors ranged from 0.008 ± 0.003° to 0.280 ± 0.186°. *These results make the KFMM the best joint position estimation model found in the literature to date using motions that fall under the realm of active-assistive therapy.*

Statistical differences between the experiments and motions that used the KFMM were found for Experiments 1–5. In terms of experiments, the position estimation errors of Experiment 4 were significantly different from all other experiments and represented the lowest errors. However, there were many variables that were not consistent between the experiments, such as the experimental factors discussed in Section 5.4.4, that affected the motion inputs, and, therefore, the estimation

Table 5.16: Comparison of the task execution times between models, as measured during each of the five critical control system tasks.

Model ID	**Mean Task Execution Time ± Standard Deviation (ms)**
Track Motion Task	
KFMM	**7.532 ± 0.313**
PMM	**9.780 ± 0.436**
NPMM	**9.70 ± 0.391**
Log Data Task	
KFMM	**1.867 ± 0.133**
PMM	**1.810 ± 0.182**
NPMM	**1.730 ± 0.135**
Gather Subject Position Task	
KFMM	**2.442 ± 0.239***
PMM	2.580 ± 0.350
NPMM	**2.638 ± 0.400***
Gather Brace Position Task	
KFMM	**3.587 ± 0.103**
PMM	**3.738 ± 0.130***
NPMM	**3.773 ± 0.113***
Command Position Task	
KFMM	6.703 ± 0.275
PMM	6.735 ± 0.273
NPMM	6.839 ± 0.667

Bolded values indicate statistically significant differences between the levels of the factors.

performance. Considering the motions, the major difference was between flexion-only and flexion–extension motions. In general, the flexion–extension motions produced lowest motion estimation errors than the flexion-only motions. This difference is likely due to the fact that the KFMM was developed for flexion–extension motions. *Therefore, one could reasonable expect that using input*

motions that deviate from the motion template for which the model was developed or optimized will result in higher estimation errors.

Despite the successes of the KFMM, there are many limitations that still remain to be explored. First, the weighting of the previous joint position with the summation of the neural activation signals are of equal portions. Variations in the weighting may change the accuracy of the estimations but studies to determine the optimal weighting have yet to be conducted. Second, the values of the coefficients R and Q were left unbounded during optimizations and this resulted in relatively large positive integers. In some computer systems, integer ranges may be significantly smaller than the values determined in the experiment. For example, an unsigned 32-bit integer has a largest possible value that is roughly 4.29×10^9, while the smallest R values were determined in the range of 3.31–1695.50×10^{13} (excluding Experiment 1 values). Bounding these values to small ranges is likely to affect the accuracy of the estimations. The KFMM was designed for slow ($< 40°$/s) flexion–extension elbow motions, as would commonly be conducted in active-assisted rehabilitation therapies. In the experiments, the KFMM was tested under similar circumstances since the devices controlled with the recorded data were also developed with maximum attainable angular velocities of 40°/s. Furthermore, the sampling frequencies of elbow motion were significantly higher than required, due to the rate conversion needed to operate at the same frequency as the recorded EMG signals. As a result, the previous joint position from two time steps prior, used to estimated the current joint position, would be very similar to the current joint position, due to the slow movement and high sampling frequency. Therefore, the estimation accuracy is likely to decrease given faster motion, lower sampling frequencies, or a combination of both, but this has yet to be determined through experimental evaluation. Another investigation into the limitations would involve deriving motion estimates based on previous joint angles that occurred further back in time. This would help to establish a relationship between how far into the future a prediction could be made given the current position and EMG signals. As long as the collected elbow motion remains within the active-assistive realm, it seems the KFMM is capable of accurately estimating joint position.

5.5.2 Muscle Activation Models

The main contribution of the muscle activation model comparison is the evaluation of muscle activation models regarding their ability to estimate elbow motion. The results reveal that the torque estimation error generated using MAM 6 is on average lower (1.67 ± 0.74 Nm) than all other muscle activation models considered in the experiment, but only statistically different from MAM 5 (2.19 ± 1.47) and MAM 7 (2.19 ± 1.50). This result is emphasized in the torque trajectories of Figure 5.9, where Model 6 can be distinguished from the other models. However, it can also be seen that none of the models follow the shape of the control trajectory particularly well. The discrepancy between the control and optimized trajectory shapes may be due to modelling errors in either the muscle activation models, the other components of the model, or unmodelled dynamics, but the origin of these errors is still being investigated.

Compared to elbow joint torque estimation error of the models proposed by Cavallaro *et al.* (average error: 3.4–4.2 Nm) and Rosen *et al.* (average error: 4.2 Nm), an improvement in torque estimation was seen, regardless of which muscle activation model was used (average error: 1.67–2.19 Nm) [135, 157]. None of the other HTMMs considered in this study or in the reviewed literature measured torque estimation errors. Although an improvement in estimation error of this elbow model has been shown over other models, even a torque error of a few Nm could cause excess strain on soft tissues of the human body or cause the user to strain excessively thinking the device will provide appropriate assistance. Both of these scenarios can lead to further injury of the user and require improvements to the estimation accuracy of the models.

In general, the results show that none of the models may be acceptable for accurate motion and torque control of a wearable assistive device, as even the lowest errors may be too high to ensure safety. These errors are likely a result of unmodelled dynamic properties. For example, variation in muscle stiffness has not been considered in any of these models. Furthermore, an assumption was made that the biceps and the triceps contained all of the force potential, including that of all major flexors and extensors of the elbow, which we know not to be true. Furthermore, the fact that the tasks are being performed at very slow speed likely has an effect on the smoothness of the motion. Finally, it is possible that incorrect assumptions have been made, or that there are

components of the model between the neural and the muscle activation dynamics that have not been considered. EMG activation is simply not fully understood yet, so further work in this area can lead to improvements in the results.

On the other hand, there are other sources of errors that should not necessarily be considered for the control of wearable devices. For example, electrode placement will have an effect on the signals collected, and it is possible to develop new techniques to find muscle locations more accurately in order to decrease variability. However, doing this will not properly reflect real scenarios in which the therapist or the patient is putting on the device, justifying the need for more robust models that account for this type of variability. Similarly, significant subject variability was observed in the collected data. The variability was more pronounced in patterns of muscle co-contraction, especially at points where the motion changed direction, when starting, or when stopping. These variations may be caused by factors such as stress, level of motivation, muscle tone, level of fatigue, and experience with performing similar motions. All of these factors would be very difficult to measure in an experimental scenario and even more difficult to determine their effect on the collected data.

One method to improve accuracy is to use a larger number of optimization parameters as this increases the number of possible solutions that can fit the data. However, increasing accuracy using this method leads to increased computational demand. Buchanan *et al.* describe that another problem with this approach is the overfitting of the data, which leads to a reduction in the predictive power of the model [121]. However, one solution to increasing accuracy while decreasing computational demand comes from increasing the complexity of biological models. By studying and understanding the mechanisms that generate movement more thoroughly, these biomechanical models can evolve to include components that account for more of the dynamic properties of the human motor control system. Increasing the complexity of the models will generate more parameters to optimize, which as stated above, could increase computational demand. However, if the complexity is modelled such that the parameters can be measured from the subject, or derived from these measurements, the computational demand for the optimization of the model could be reduced.

The statistical analysis revealed other significant findings, namely that torque and muscle

activation errors differ by motion and that data point optimization time differs by model and number of optimization parameters. In terms of muscle torque and muscle activation estimation errors, the differences found between motions suggest that the muscle activation models may be suited better for certain motions than others. For example, the flexion–extension with varied starting angles motion differs significantly from the flexion–extension with additional load and the flexion–extension with pauses motions with respect to torque estimation error due to these latter ones having unmodelled dynamics, such as the added mass or discontinuities in trajectory smoothness, respectively. Both added masses and pauses in motion are common in rehabilitation scenarios and activities of daily living and, therefore, must be accounted for within these models. Another possibility for flexion–extension with varied starting angles performing better than the other motions is that the starting torque is higher due to the motion starting at an increased elbow angle. This behaviour matches models that reflect a torque pattern that begins at a larger positive value, hence matching the torque estimation observed from most models (except MAM 6), as shown in Figure 5.9.

Differences in data point optimization time emphasize that some models may be better choices when optimization is required and computational resources are limited. Based on the results, the models can be ranked from smallest to largest data point optimization time as follows: MAM 1, MAM 2, MAM 3, MAM 5 or 6, MAM 7 and MAM 4. Therefore, MAM 1 would be ideal if optimization time is the highest priority, as it produces a torque error not statistically different from the other models while exhibiting the lowest data point optimization time. Finally, models with two optimization parameters had a significantly longer data point optimization time and larger total muscle activation error than models with one optimization parameter. However, there does not exist a significant difference in torque error between one-parameter and two-parameter models.

Ultimately, this study highlights the trade-off between the accuracy of the model and the computational expense. While estimation accuracy provides a means to determine theoretical feasibility, computational demand provides a quantification of concrete feasibility regarding the use of these models in wearable assistive devices. It is common to develop and study these motion models using computer systems that can execute more than one million instructions per second

and in situations where optimization time is infinite. In the presented experiment, the forward optimization took up to 10.3 min on such a computer system. However, even this length of optimization may be too long to consider using this model in a wearable device that may need to execute only a few movements at a time. Furthermore, moving this computational task to an embedded computer system, where the instructional velocity may be far less than the one used in this experiment, would only increase the optimization time to prohibitive levels. *Accompanying this with the fact that optimizations may need to be conducted at each usage, or for each motion, due to variability in biological signals and environmental factors, a major limitation for adoption of these models and the devices that require them lies in the computational expense.*

5.5.3 Elbow Motion Model Comparison

The successful completion of Experiment 7 has entered into new research territory, shown the capabilities of the WearMECS framework, added to the list of control system comparisons, and highlighted the limitations of existing motion models. This experiment was the first software simulation motion tracking experiment to compare the KFMM, PMM, and NPMM in the same control system implementation. Based on the reviewed motion models, the first use of a nonlinear polynomial to represent dynamic elbow muscle torques was also accomplished during this experiment. Clancy *et al.* and Liu *et al.* both used nonlinear polynomial models to estimate muscle torque but only did so for static postures [127, 173].

This comparative study was made possible through the use of the WearMECS framework. Each of the three motion models were implemented into the control software, as estimation-level components. From the digital interface of the control system, two mouse clicks were all that was needed to select a different motion model and have the control system reconfigured to use that model for estimation of the subject's motion. Using the framework, the total time of the motion tracking experiments was reduced from approximately 2 hours for Experiments 1–5 down to 45 minutes for Experiment 7. Some of the reduction is due to the data collection and tracking occurring simultaneously. However, only one model was being tested during Experiment 1–5, while three were being tested in Experiment 7. *This confirms that tools like the WearMECS framework would be advantageous to increasing the efficiency of motion tracking experiments.* Increasing the

efficiency of control system experiments increases the rate at which studies can be performed, which improves the understanding of human motion at an increased rate.

In addition to these aspects, the results of the statistical analysis revealed significant differences between the models during the optimization portion of the experiment and between the motions during the motion tracking portion of the experiment. During the optimization, the three optimized models produced position estimation RMSEs that were all significantly different from each other, meaning that the KFMM produced a lower error across all subjects. Furthermore, the data point optimization time for the NPMM was significantly longer than those produced by the PMM and the KFMM. This result was expected since the NPMM required the optimization of eight parameters, while the PMM and KFMM only needed to optimize two and three parameters, respectively. It is of interest to note that there were statistically significant differences in the data point optimization times between one-parameter and two-parameter models in Experiment 6, but not between the two-parameter and three-parameter models used in Experiment 7. However, many differences between the experiment exist, including the number of data points for each model and the type of motion models. Further study would help to identify the trade-off between the accuracy and adaptability of these motion models.

Comparing between the motions used in Experiment 7, two main statistically significant differences were found. *The first statistical discovery was that the best motion tracking performance across all subjects and models was when tracking the slow flexion–extension motion.* This motion was significantly different than all others, except for fast flexion–extension motion, which was the motion used for optimization of the models. One explanation for this is that these two motions differ only slightly in terms of position inputs. During the experiment, it was clear that some subjects performed the fast flexion–extension motions at rates that were not much faster than others performing the slow flexion–extension motions. Averaging across subjects could account for the lack of significant difference between these two motions. *The other conclusion to draw from these results is that the number of repetitions and the additional mass affect the position estimation.* First, when performing multiple repetitions, fatigue will become more of a dominant dynamic within the EMG signals. Second, the additional mass will also stimulate the muscle to produce more force, which results in variation in the EMG signals. The fact that the 10-repetition

flexion–extension with additional mass motion has the highest overall errors supports these ideas. The second statistical discovery was that the fast flexion–extension motion produced statistically different position estimation errors than both of the 10-repetition flexion–extension motions. This result suggests that, again, both the number of repetitions and the additional mass play a factor into the tracking performance.

Looking between the models used in the experiment, it was found that all three models were statistically different from each other based on the position estimation RMSEs collected during the tracking portion of the experiment. However, the ranking of lowest to highest RMSE from the tracking task is not the same ranking that was derived during the optimization phase, despite statistical differences between all models in both portions of the experiment. This difference was not expected and warrants further analysis. The first point of analysis is based on the values of the optimization parameters, which are presented in Table 5.3. The large variation and mixture of positive and negative values for each of the parameters suggests that the motion input signals contain large variations. Since the subjects performed the motions with visual feedback, were given training time, and had their motion constrained to a two-dimensional plane, the position signals collected during calibration and performance of the motion tracking tasks should not have differed significantly. These factors were used in order to increase the accuracy and repeatability of the position trajectories, leading to the conclusion that much of the variability in the input signals stems from the EMG signals themselves. The statistical results found between the motions also support this claim.

To further support this conclusion, the repeatability and the ranking of the motion model tracking performances were explored. The standard deviations of the position estimation RMSEs for the KFMM and NPMM are large, with respect to the mean values for both of these models. These standard deviations reflect that these models performed better for some subjects than for others. However, analyzing across subjects is what leads to these large standard deviations. Furthermore, this becomes even more evident if models are ranked on their performance for both the optimization and tracking portions of Experiment 7. During the optimization, the models produced position estimation errors from lowest to highest in the following order: KFMM, NPMM, and PMM. This ranking trend was consistent across all subjects for the optimization. When

performing this model ranking for each of the subjects, the same trend was not found across all subjects. For one subject, the KFMM produced the highest position estimation errors, despite producing the lowest errors across all subjects during the optimization. Based on this evidence, the data suggest that motion models may perform better for some individuals, than for others. However, the results presented here are not conclusive regarding this claim. It is likely that many aspects of the subject factor into the ability to optimize these models and use them to perform motion tracking tasks. The number of characteristics of the subject that affect this is likely to be large, with many of them also being difficult to measure objectively. *The results of this experiment support the exploration of individualization of motion models and the effects of motion factors on the EMG signals.*

There is one major difference between the optimization and tracking software that might attribute for some of the position estimation RMSEs of the motion tracking. During the optimization, the model parameters were determined by performing an optimization procedure using the entirety of the position and EMG signals that constituted the calibration motion. Essentially, the models were optimized over the entire trajectory of a single repetition of elbow flexion–extension motion, which contained tens of thousands of data points. However, during the tracking portion of the experiment, the models produced estimates based on windows of data that were 500 data points in length. This difference may account for some portion of the position estimation RMSEs, since the models were only receiving portions of a trajectory when they were optimized over an entire flexion–extension motion trajectory. Furthermore, the position and EMG signals collected for each motion task also differed from those for which the model was optimized. Although, one may expect that the variations in these signals to be small, the differences may have resulted in large errors. The combination of these effects accounts for at least a portion of the large differences in position estimation RMSEs seen between the optimization and tracking results. However, it is not as clear as to whether or not these effects account for the differences in the ranking of models observed between subjects, or between the optimization and tracking portions of the experiment. Further studies are required to determine the relationships between the individual and the motion estimation abilities of these models.

5.6 Discussion: Controlling Wearable Mechatronic Elbow Devices

The successful implementation and testing of control systems for the purpose of tracking elbow motion with wearable assistive device is due, at least in part, to the control system development tools that were used. *First, the WearMECS framework supported the implementation of four control systems.* The framework reduced the number of modifications required to adapt Implementation I, which controlled the WearME elbow brace, for the control of the Active A-Gear during elbow motion tracking, as seen through Implementation II. Implementation III enabled the comparison of muscle activation models and showed that the framework could support the development of different task-level controllers, such as the optimization task of Experiment 6. The WearMECS framework made it possible to conduct an elbow motion model comparison, as shown through the use of Implementation IV during Experiment 7. Using the software engineering principles that support the framework, it was possible to change the active motion model with two mouse clicks, enabling the subject to perform the motion tracking tasks at a much faster pace than in the previous experiments.

Second, the development of the integration testing protocol allowed for the isolation of control system components during experimental testing. By splitting the testing into progressively higher levels of integration, it is easier to characterize the different types of control components, while maintaining safety of both the subject and the device. Software simulations allow the developer to analyze aspects of the task-level and estimation-level control components. The offline remote control phase enables evaluation of system responses to the exact same inputs, which can be repeated as many times as required. The response of the control system and device can be assessed under real-time control deadlines during the online remote control phase. In this phase, developers are also still able to measure the motion differences between the subject and the device. Once the control system has been sufficiently validated and safety ensured, the worn phase integration testing allows the wearable assistive device to be evaluated under the scenarios for which it was designed. Through the use of these two control system development tools, four control systems were developed, seven experiments were performed across three integration testing phases, and

many insights into the digital control systems of wearable assistive devices have been discovered.

The first set of insights comes from the software simulation portions of the elbow motion tracking experiments. *During Experiments 1–5, it was shown that experimental factors, such as motion training, and control outputs, such as position output, significantly affect the performance of the motion estimation.* This result was also true for the offline remote control testing. These results also showed that different motion input variations also affected the motion estimation performance. The results of Experiment 6 and 7 also supported this claim. The comparison of muscle activation models during the software simulation of Experiment 6 showed that the estimation errors are inversely proportional to the time taken to optimize the motion model. These results also showed that a larger number of optimization parameters, produced significantly longer optimization time. Finally, the results of the Experiment 7 software simulation suggest that the number of repetitions and additional load affect the estimation performance, the models all produce statistically different optimized estimation errors, and the models performed better for certain individuals than others. In addition, the results also show that eight-parameter motion models took significantly longer to optimize than two-parameter and three-parameter models. During Experiment 6, statistical differences were found between the number of parameters, which showed that the two-parameter variants of the HTMM took less time to optimize than the four-parameter variants. Combining these experimental results, the general trend is that more parameters will require more time to produce optimized models.

The offline remote control phase testing involved comparing between experimental factors, motions, and control output variations across Experiments 1–5. As with the software simulation portions of these experiments, it was found that the same experimental factors and control output differences affect the performance and that giving subjects visual feedback digitally also affected the tracking performance. The results of the statistical analysis showed significant differences between the motions used across these experiments. It was shown that flexion-only and flexion–extension motions produced significantly different errors. These differences may, at least in part, be caused by the differences between the experiments, and, therefore, it is not conclusive that the differences in the tracking performance of these implementations were due only to variations in the motions. The objectives of the offline remote control experiments were to quantify the tracking

ability of a control system that employed the KFMM and improve upon this ability. *The final results of Experiment 5 show that these objectives were met, as the position tracking accuracy is better than all of the reviewed motion models that employed position tracking, previous studies with the KFMM, and position tracking studies of the reviewed wearable assistive elbow devices.*

The objectives of Experiment 7 included the performance of online remote controlled phase integration testing. The success of this experiment means that this was the first study to facilitate this phase of integration testing for wearable assistive elbow devices. The results of the motion tracking revealed three interesting points. First, variations in motion, such as the number of repetitions and additional load, affect the device's tracking performance, which supports the results of the other testing phases. Second, the three motion models used in the experiment performed better for some individuals than for others. However, further investigation is required to determine the relationships between these models and aspects of the individuals. Lastly, the actuation-level control software components and actuation system produced brace position errors, in which the trends varied largely from both estimated and commanded position errors between the models. For both the estimation and commanded position errors, statistically significant differences were found between each of the models. When comparing between models using the brace position errors, no statistically significant differences were found. For the KFMM and PMM, the average position errors increased, while for the NPMM, the average errors decreased. *One conclusion is that these actuation components may be a limiting factor for both good and poor tracking performances.* However, the results are not conclusive as the device was tuned for its maximum velocity output and some of the subjects moved close to this maximum. Further studies should be conducted where the motions of the subjects are not close to the maximum capabilities of the device, in order to eliminate this as a factor.

During the online remote controlled testing, the first study of task execution times and data storage needs, for a control system performing elbow motion tracking tasks, was conducted. This study highlighted three important findings that have not been verified for these types of digital control systems. First, statistical differences were found between the motions for the Command Position task execution time that matched those of the brace position errors found during the motion tracking. For this case, it was seen that shorter task execution times corresponded with

lower position errors. Second, significant differences were found between models for the Track Motion task execution time. The ranking of these model performances using the task execution time matched the same ranking of the models for the brace position errors captured during the motion tracking task. Again, it was shown that shorter task execution times were related to lower position errors. *Combining these first two findings, the evidence supports the notion that task execution times and position errors are proportionally related during motion tracking tasks.* This is of particular interest, as the opposite relationship was established for optimization tasks, based on the results of Experiment 6. *Third, the data storage velocity produces more data than might be possible to communicate wirelessly for storage on an external device.* In this experiment, the average data storage velocity was found to be 1.413 Mb/s. According to Pothuganti and Chitneni's review of wireless protocols, the Bluetooth (IEEE 802.15.1 standard) and the Zigbee (IEEE 802.15.4 standard) have maximum data transfer rates of 1 Mb/s and 0.25 Mb/s respectively [165]. The data storage velocity for five floating-point numbers sampled at 2000 Hz exceeds the maximum data transfer rate of both the Bluetooth and Zigbee protocols. As a result, developers may have to use wireless protocols, such as Wifi, which comes at the cost of higher power consumption, or work with storing the data on the device until it can be uploaded to another device.

Looking across the experiments, one final discussion point should be made, as it affects the potential usage of these devices for applications in musculoskeletal rehabilitation. The evidence from these experiments suggests that additional mass, and, more generally, unaccounted dynamics, affects the performance of motion tracking. In Experiments 6 and 7, the results were significantly different between motions where no additional load was used and motions where either a 1 kg or 2 kg load was used. However, there were no significant differences in Experiments 1–5 between the motions with and without additional masses. One reason for this discrepancy between these sets of experiments could be due to the motion model itself. It is possible that the KFMM was less sensitive to changes in the EMG signals due to additional mass and there is strong evidence to support this idea. The KFMM takes both position and EMG signals as inputs, and outputs the estimated elbow position. While experimenting with this model, there were cases in which the KFMM would not produce non-zero position estimates when given only the EMG signals of a subject. In fact, this was the case for most of the data sets and supports the idea that an

additional mass would have little effect on the signal. However, during Experiment 7, the KFMM was able to track the position trajectory of one subject, holding a 2 kg mass, using only their EMG signals. The same results could not be duplicated when the subject did not hold onto the mass. As a result, it suggests that the optimization of the model may affect the sensitivity to variations in the EMG signals. This anecdotal evidence, together with the statistical evidence of Experiments 6 and 7, supports the claim that motion models are not robust to changes in dynamic properties that have not been accounted for in the model. Further studies should be conducted to determine the relationships between these dynamics and the motion tracking performance. *Much of human motion involves loaded motions, including musculoskeletal rehabilitation, and therefore, models must be improved to respond appropriately to the dynamic properties of objects that are lifted by the individual.*

5.7 Summary

Implementation and testing are two important aspects of digital control system development and enable the use of these control systems to appropriately regulate the behaviour of wearable assistive devices. Through the use of control system development tools, the quality of both the implementations and the testing can be improved. To support the implementation and testing presented in this chapter, a set development tools have been used. First, the WearMECS framework has enabled the development and modification of multiple control systems for variations of elbow motion tracking tasks. Second, a human–machine integration testing protocol for the control systems was proposed and used to separate experimental testing in such a way that it isolated control system components. Lastly, a metrics suite was used to harvest data for the analyses conducted in this chapter. The combined results of the experiments support the efficacy of these tools for the design, implementation, and testing of control systems for wearable assistive devices.

Seven experiments were conducted in which part of their purpose was to quantify the motion tracking performance of control system implementations. In each of these experiments, a software simulation portion of the experiment was conducted that involved collecting motion estimates and comparing them to the human motion inputs. Control software was implemented, using the

WearMECS framework, to facilitate these experiments, such that the motion estimation models were implemented as estimation-level control system components. In six of the experiments, wearable mechatronic devices were controlled to track the elbow motion of the subjects. Across these experiments, various metrics were collected to indicate aspects of the control system quality, such as the accuracy, repeatability, adaptability, and resource utilization. During Experiments 1–5, the main focus was on improving the accuracy and repeatability of estimation and control using the KFMM and assessing these metrics under variations of motion inputs. Overall, the estimation abilities of this model and control performances using this model were improved compared to existing studies with this model. Statistically differences were also found between the motions, which suggests that this model, and perhaps motion models in general, may be sensitive to variations in the motion inputs. Experiment 6 consisted of the development of a HTMM to be used for comparison of seven muscle activation models found in the literature. This was the first comparative study of muscle activation models and the results of the experiment highlight that a trade-off exists between the accuracy of estimation and the resource utilization of motion models. Experiment 7 was used to compare three motions models, which were the KFMM, the PMM, and the NPMM. The results of this experiment revealed many statistical differences and opened up many more questions to be explored. Most noteworthy was the fact that the accuracies determined through optimizing the models was quite different from those measured during the motion tracking tasks. These results also varied between subjects, suggesting that the estimation accuracy is dependent on characteristics of the individual for which the model is used. This claim is fairly straightforward to accept abstractly, but further large-scale studies are required to determine the exact relationship between aspects of the motion models and aspects of the individual from which the inputs are collected. Lastly, data communication and storage needs are crucial to evaluate, as they may become limiting factors in the control system or device designs. The results of data storage analysis showed that some common wireless protocols may not be able to support the storage demand, even for a small number of variables.

Overall, the success of these experiments have shown that it is possible to improve upon existing control system solutions and compare between components or sub-systems of control solutions. Many interesting findings were discovered from these research efforts that help to focus

future efforts. However, many questions still remain to be answered, with new questions arising from the results of these experiments. Future work is required at all levels of the control system development if the benefits of these wearable assistive devices are to be realized for applications in musculoskeletal rehabilitation.

Chapter 6

Applications in Musculoskeletal Rehabilitation

6.1 Introduction

Wearable assistive devices are tools developed to provide benefit to both the patient and therapist during musculoskeletal rehabilitation. Due to the variability inherent to rehabilitation processes, the number of applications for realizing these benefits are bountiful. The benefits are seen through two general functionalities. First, wearable assistive devices produce motion. The motion can be used to assist, resist, guide, or correct, or produced in any meaningful form that is within the capabilities of that device. Second, the states of the motion, the users, and the environment, together with the interactions of these states, can be quantified and recorded. Thus far, the assistance that these devices provide is not only through motion production, but quantification of the motion interaction. This will also assist patients and therapists in assessing and adjusting rehabilitation protocols, and researchers in learning about human motion and rehabilitation.

The specifics of rehabilitation applications are lacking, making it difficult for developers of these devices to direct their efforts. A potential benefit of wearable assistive devices is apparent and drives the development of these devices, regardless of the lack of requirements. As a result, devices continue to be developed for motion assistance scenarios, such as active-assistive motion therapy, with the emphasis placed on improving the engineering characteristics of the devices. Ideally, more

collaboration will occur over time between researchers, engineers, patients, therapists, and other domain experts to determine these requirements and specifications. One solution to speeding up this process is to use the wearable assistive devices to quantify the motion interactions and use the data to determine requirements or better understand the human motion systems. In this chapter, two major opportunities to benefit from these devices within the realm of musculoskeletal rehabilitation are discussed.

6.2 Tools for Motion Therapy

The first major opportunity for wearable assistive devices within rehabilitation are as motion production devices. Wearable assistive devices have the opportunity to lift a massive physical burden placed on therapists or other medical professionals. During rehabilitation, the patient must be moved in a very specific manner, which is determined by the therapist and changes between different interaction phases of rehabilitation. As will be discussed in Section 6.3.2, there is no consistent view about these phases because they depend on many factors. However, there is still a need to categorize motion therapies or tasks in order to discuss them and develop wearable assistive devices to assist with their completion. As a result, two spectrums on which one can view the interaction emphasis of the patient and the therapist are proposed. Motion therapies are commonly described by denoting the participation level of the patient and the role of the therapist in completing the goal of the motion therapy. At the extremes, the patient can either completely relax their body (passive) or put full focus and effort into completing the desired task (active). The therapist will either support (assistive) or suppress (resistive) the completion of a motion task. Fig. 6.1 provides some examples of existing motion tasks that fit within each of the four quadrants of the proposed patient–therapist interaction classification.

While the patient is not strong enough to complete the motion, the therapist must generate the motion trajectories that allow the patient to complete their rehabilitation protocol. This scenario is common during the passive-assistive and active-assistive interaction phases. During these phases, a large physical burden is placed on the therapists in order to generate patient motion and manage the performance of their own motion. A limited number of therapists, relative

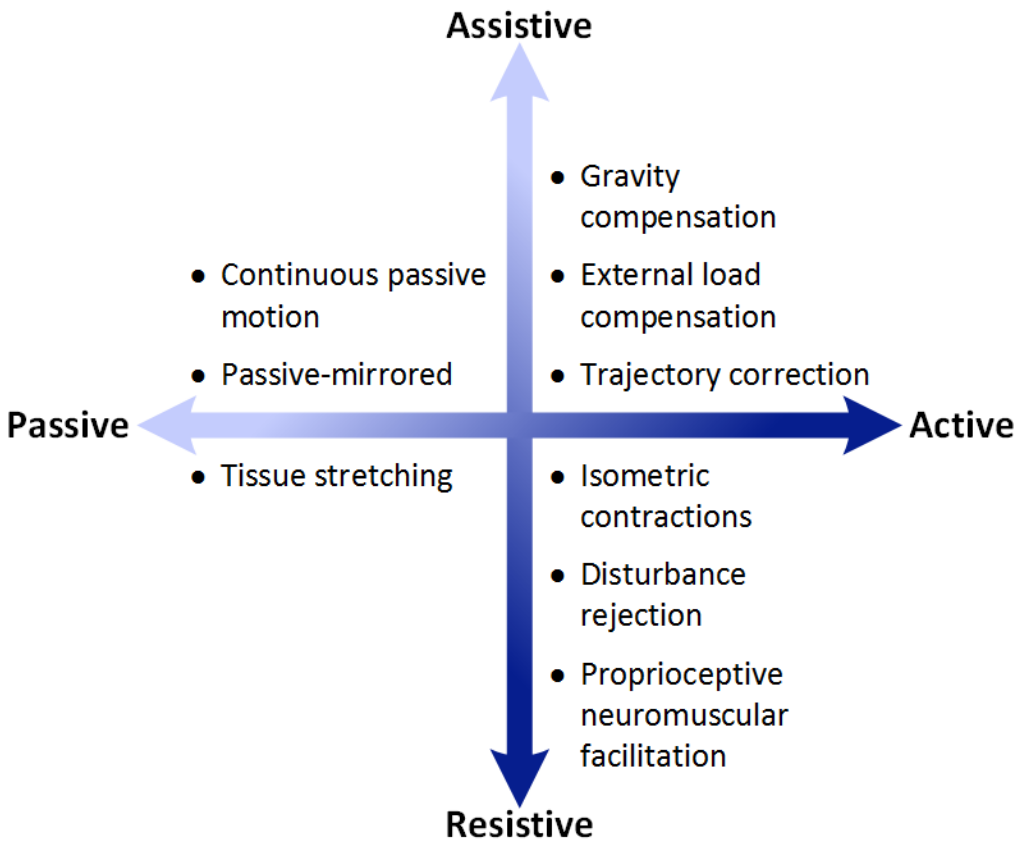


Figure 6.1: Patients either participate in motion therapies passively or actively, while therapists provide either assistive or resistive forces to meet the goal of the therapy. Due to the variability in MSDs, patients, therapists, and rehabilitation plans, the motion therapies are categorized along these spectrums. Examples of motion tasks are provided for each quadrant of this spectrum.

to the growing number of MSDs, means that optimal outcomes are more difficult to achieve for large populations, leaving many individuals with less normal motion abilities. This is one of the reasons that existing devices are oriented to provide solutions for these phases of therapy. There is a large societal benefit to be generated upon the successful integration of wearable assistive devices into the musculoskeletal rehabilitation framework. General requirements for incorporating these devices into each of the four patient–therapist interaction areas are discussed in the following sections.

6.2.1 Passive-Assistive Motion Therapy

The largest physical burden to the therapist is caused during the passive phase of rehabilitation. In this phase, the patient is unable to move their body segments or is instructed not to do so, in order to remove the risk of reinjury. As a result, the full load of the patient's body segments must be supported to produce motion. Examples of passive-assistive motion therapies include continuous passive motion and passive–mirrored motion tasks. Continuous passive motion is performed to help regain joint range of motion following surgery [253,254] or stroke [255]. This therapy requires the joint to be moved throughout a specific range of motion at a slow and constant speed. A controller to facilitate this motion task needs only to generate a constant velocity in one direction until a position limit is reached, then move in the opposite direction until the other position limit is reached. This task is repeated endlessly or until a number of oscillations, determined by the therapist, has been reached.

Passive–mirrored motion tasks are used when a patient is suffering from a MSD that affect them unilaterally [237]. For example, the patient will actively move their unaffected arm and the therapist will mimic this motion by moving the affected arm. Stationary assistive devices have been used to replace the therapist in generating the motions for both passive–mirrored and active–mirrored motion tasks [256]. These devices essentially use two mechatronic systems, one for each arm. The control systems for passive–mirrored motion tasks need only to measure the movement produced by the unaffected arm and replicate the motion with the wearable assistive device, placed over the affected arm. Wearable assistive devices could facilitate this same therapy, but with increased mobility. The measurement of the unaffected arm could be completed with sensing systems with the wearable assistive device being used to replicate the motion on the affected side. This could reduce the cost of the system, compared to the existing stationary solutions. Furthermore, if both the sensing systems and wearable assistive devices are wireless, this therapy could be provided in different locations, instead of forcing the patient to travel to the device.

The passive nature of the patient within these motion therapies means that the digital control systems can be simplified. This is due to the fact that the motion trajectories are not determined through the bioelectric signals of the affected body segment. Simplified control systems usually

lead to reduction in the cost of the assistive devices and makes it easy for the therapist to program these devices. The major benefit of devices targetting passive-assistive rehabilitation therapies is the reduction in total work (motion), that would otherwise be performed by the therapist.

6.2.2 Passive-Resistive Motion Therapy

The passive-resistive motion tasks are those in which the patient is passive in their participation and the therapist is applying resistive forces. One common mode of passive-resistive motion therapy is through passive stretching of soft tissues. In many cases, MSDs will cause stiffness of the soft tissues, which can result in reduced motion abilities and pain. Therapists will stretch the tissues beyond their current active capacity in order to regain some of the natural characteristics of those soft tissues. Essentially, the therapist is resisting the stiffness forces produced by the soft tissues. Ren *et al.* have developed a control algorithm for their device to provide passive stretching motion tasks [43]. Like any other motion task, the range of motion, velocity trajectories, and force trajectories required to perform the passive stretching can be measured from a therapist. These quantities can be used to inform the performance of the passive stretching, such as in the control system developed by Ren *et al.* However, it is difficult to measure discomfort or pain produced by the stretching, which is highly likely during this motion task. A therapist can use their senses to dynamically adjust the amount of stretching force the patient can tolerate. Pain is a complex physiological process that, currently, cannot be readily digitized for wearable assistive devices, meaning it would be difficult to implement this type of motion therapy. Further study is required to determine the relationship between the stiffness of the tissues, the force trajectories applied to stretch the tissues, and the feedback the therapist uses to produce positive effects without further injuring the patient.

6.2.3 Active-Assistive Motion Therapy

Active-assistive forms of musculoskeletal rehabilitation have been the major focus of existing assistive devices, though there are many more opportunities to explore in this area. In these rehabilitation scenarios, the patient actively tries to complete a motion. The therapist will track the motion of the patient and assist them as needed to complete the desired task. Studies employing

either stationary assistive devices or wearable assistive devices to help stroke survivors perform functional active-assistive tasks have shown similar improvements in motion abilities as those seen through manual motion therapies [5–7]. These results suggest that wearable assistive devices are potential solutions to the growing demand for active-assistive rehabilitation.

As discussed in Section 4.2, control systems that support active-assistive rehabilitation scenarios must be able to track the patient's motion and provide disturbance forces that assist with task completion. Currently, the ability of devices to track human motion is limited, as shown in this thesis, and the force trajectories of the therapists have yet to be quantified. This leaves many opportunities to apply these devices for active-assistive rehabilitation under active investigation. However, three general examples of their application in this domain are in gravity compensation, external load compensation, and trajectory correction. Gravity compensation would occur when the patient is trying to move their body segments against gravitational forces. During times when the patient struggles to complete the motion, the therapist will provide assistance to compensate for the gravitational forces, by applying forces in the opposite direction. External load compensation follows a similar scenario, except that the patient is stronger and trying to complete a motion while an external load is being applied to one or more of their body segments. In this case, the therapist compensates for a portion of the dynamic forces introduced by combination of gravity and the external load. During trajectory compensation, the patient will attempt to move in a very specific motion pattern and the therapist will apply assistive forces to correct the motion of one or more joints. This can be used to retrain the fine motor control skills, while strengthening the tissue simultaneously.

Developing control systems for rehabilitation protocols where the patient is participating actively is more difficult than passive participation. This is due to the fact that the device needs to determine what the patient is trying to accomplish and react accordingly. However, the desired motion of the patient varies rapidly, which constrains the control system to execute a series of complicated tasks within a very short period of time. Currently, the therapist has all of the advantage over wearable assistive devices. The therapist's brain and body is fined tuned to move synchronously with other humans, and their professional training only increases these abilities. Furthermore, the human brain has a far larger capacity for processing information than any mi-

croelectronic processor and is much more robust to errors. Duplicating the abilities of the therapist is a difficult task for control system developers, but the benefits for society are large and continue to drive the research efforts within this rehabilitation domain.

6.2.4 Active-Resistive Motion Therapy

Active-resistive therapies are often used to strengthen the patient once they gave regained some of their motion abilities. In these therapies, the patient actively tries to complete a motion task, while the therapist provides resistive forces. Patients who regain a level of strength that surpasses the therapist's ability to resist, would be candidates to return to normal daily activities. Since active-resistive motion tasks are extremely fatiguing for both the patient and the therapist, they are usually short in duration and involve highly specific motion patterns. These therapies are commonly paired with resistance training, such as with dumbells or bands, to alleviate the physical demand on the therapist and enable the patient to improve their motion abilities more freely.

In terms of existing device surveys, Proietti *et al.* found that no resistive devices fit their criteria, Basteris *et al.* discussed how the effectiveness of resistive modes of training with robots was limited, and Maciejasz *et al.* describe devices that produce passive resistive forces [22,192,237]. Wang *et al.* and Ren *et al.* developed passive resistance controllers that are meant to provide a constant level of resistive torque for the user to move against [43, 46]. This type of control system is simple to implement and mimics traditional resistance training tools. However, given the limited effectiveness for resistive training models by Basteris *et al.* and the low cost of traditional resistance training tools, further study is warranted to show the effectiveness of these devices within resistive motion tasks.

One area that active-resistive motion controllers should explore is within motion tasks where the resistance is dynamic. Isometric contractions, disturbance rejection tasks, proprioceptive neuromuscular facilitation therapies are examples of resistive motion tasks in which the therapist applies dynamic resistive forces. Much like active-assistive motion tasks, the devices would be required to track the motion of the patient and provide disturbance forces to the patient. The difference is that these forces are used to make the task more difficult to complete. Since a wearable assistive device produces more repeatable motion than a human, they could also be candidates

for correcting improper motion patterns using resistive forces. The opportunities for developing control systems that enable active-resistive motion therapies are vast.

6.3 Human Motion Quantification

The second major opportunity for wearable assistive devices is to be used as tools for quantification of human motion. The results of the motion tracking experiments suggest that the current understanding of the human motion system and its components is limiting the usefulness of wearable assistive devices. However, the results did show that experimental factors, such as visual feedback, motion training, and constraining motion, all increase the accuracy and repeatability of the motion tracking. This was likely due to the fact that these factors affect the performance of motion, meaning that more specific motion inputs should lead to better tracking outcomes for wearable assistive devices. However, these devices must be able to function appropriately even when motion, and the signals that represent it, do not exhibit a high level of repeatability. Therefore, the improvements to the knowledge surrounding human motion will help to increase the utility of these devices.

One potential solution is to use wearable assistive devices to quantify the motion of healthy individuals to establish normal motion abilities and of patients to determine the changes that occur as one works to regain normal motion abilities. Furthermore, the interaction between the patient and the therapist, as they complete motion therapies together, can also be quantified. The data produced from these endeavours would provide insight into device specifications, control system improvements, effectiveness of therapies, and tissue healing processes. These devices would provide much more dense and rich data sets than could be gathered with traditional mechanical measurement devices.

In the following sections, requirements for the development of wearable assistive devices are explored within two areas. First, a study of human motion repeatability, using data collected with a wearable device, is presented as an attempt to quantify this motion quality. Next, the results of a national survey of therapists will be discussed, with the aim of harvesting requirements for software systems that will enable therapists to communicate with and control the behaviour of wearable assistive devices.

6.3.1 Determining Motion Norms

The variability inherent to humans and their motions makes it difficult to state generalization across groups. For solid generalizations, large populations studies will be needed with as many factors as possible being controlled. However, some evidence can still be gathered from smaller studies. One motion quality that affects the performance requirements of wearable assistive devices is the repeatability. Currently, most control system developers seem to be racing toward the smallest possible position errors and variance of those position errors. This is due to the fact that motion repeatability of human has yet to be quantified for each particular motion task and, therefore, there are no targets for developers, leaving them to continually minimize the errors and variance of errors. Through the analysis of the experiments presented in Chapter 5, it was shown that tracking performance is influenced by motion variations and experimental factors that affect the subject's performance of these motions. Looking across these factors, a large range in repeatability values, as shown through the standard deviation of position error metrics, was found. One hypothesis surrounding these data is that not only is the repeatability affected by various experimental and motion factors, but that humans have large position errors when performing a task repeatedly. This hypothesis is linked to the question, what level of motion repeatability is natural to humans? The answer to this question would allow wearable assistive device developers to match this repeatability level, allowing the devices to be more robust to the natural motion qualities of humans.

To provide insight into this hypothesis, a study of the motion data collected from Experiment 4 was conducted. The experiment involved five healthy subjects performing elbow flexion–extension motions, wearing the Collection Arm device. The motion were completed within three motion ranges: 0–45°, 0–90°, and 0–120°. For each range of motion, subjects were given unlimited training time to learn the motion with visual feedback of their elbow joint position. Upon completion of the training, visual feedback was removed and three repetitions of motion were recorded. The subjects were instructed that their goal was to reach the upper position limit of each motion range with as little error as possible. The standard deviation between repetitions was calculated for the subjects' maximum elbow position and across position trajectories to provide measures of the

end-point and trajectory repeatability. In terms of the trajectory metrics, the mean and maximum standard deviations were calculated to represent the average-case and worst-case scenarios of the elbow motion repeatability.

Overall, the results show that the standard deviation of end-point position, average-case (mean) position trajectories, and worst-case (maximum) position trajectories were 0.828°, 5.012°, and 10.447°, respectively. This shows that, after training and with motion constrained to a two-dimensional plane, the subjects were able to consistently meet end target positions, but the trajectories taken exhibited large relative deviations from each other. To determine if the motions played a role in these results, a statistical analysis was performed between motions using a one-way ANOVA. The analysis was conducted using the IBM SPSS Software (Version 24.0) with an α value of 0.05 indicating significance. A Tukey post hoc test revealed that 0–45° flexion–extension motion was statistically different than the 0–120° flexion–extension motion in both the average-case trajectory standard deviation (3.325° vs 7.102°, $p = 0.009$) and worst-case trajectory standard deviation (7.369° vs 13.658°, $p = 0.048$) metrics. One argument for this result is that the length of the trajectory taken to reach the end-point has a significant effect on the repeatability of the position trajectories. No statistically significant differences were found between motions for the end-point standard deviation. This suggests that, for simple elbow flexion–extension movements, the length of the trajectory may not play a significant factor in reaching the end-point target if the subject has an opportunity to train the motion. The results of the statistical analysis are presented in Table 6.1 and a graphical comparison is shown in Fig. 6.2.

Although the sample size of this study is small, the data were normally distributed and a statistical difference was found. This fact supports the demand for future studies involving larger subject populations. One interesting finding was that some subjects performed better at specific motions than others. Training the motion before performing them was used to mitigate performance differences, but this result still persisted. As shown in the previous experiment, there are many factors that affect the performance and more investigations are required to determine the relationships between these factors and motion qualities, such as repeatability. Looking between subjects, it was also shown that a lower standard deviation of end-point motion, did not always coincide with a lower standard deviation of position trajectories. This likely due to the effort of

completing the goal having a higher weight in the brain's control algorithms than repeatability of the motion trajectory, but the results of this experiment are not conclusive about this idea.

The experimental results suggest that there is a large variability in the ability of a healthy human to control repeated elbow motions. The level of variability that is normal or average is still up for debate. However, the implications of this are quite significant considering that recovery outcomes are determined through comparison to the motion of healthy individuals, which themselves show large variations. Wearable assistive devices are evolving to the point of being able to provide accurate motion patterns for musculoskeletal rehabilitation purposes. However, these devices may never become more effective than traditional therapy methods if the natural motion qualities cannot be characterized and achieved. It may be advantageous, for both the cost and complexity of the device and the user, to have a relaxed requirement for repeatability that matches that of natural human motion. The results collected for this study were done using a wearable device, which supports the potential for wearable devices as systems for studying human motion. In order to increase adoption of these devices by clinicians and therapists, it is important to understand their existing methods of data collection and analysis and how the devices can provide similar or better functionality.

Table 6.1: Comparison of standard deviation metrics across the three different motions performed during Experiment 4.

Motion ID	**End-Point Standard Deviation (°)**	**Average-case Trajectory Standard Deviation (°)**	**Worst-case Trajectory Standard Deviation**
0–45° flexion–extension	0.976	**3.325***	**7.369***
0–90° flexion–extension	0.444	4.610	10.284
0–120° flexion–extension	1.062	**7.102***	**13.658***
Average	0.828	5.012	10.447

Bolded values indicate statistically significant differences between the levels of the factors. Bolded cells which include an * indicate results where not all pairwise tests were statistically significant.

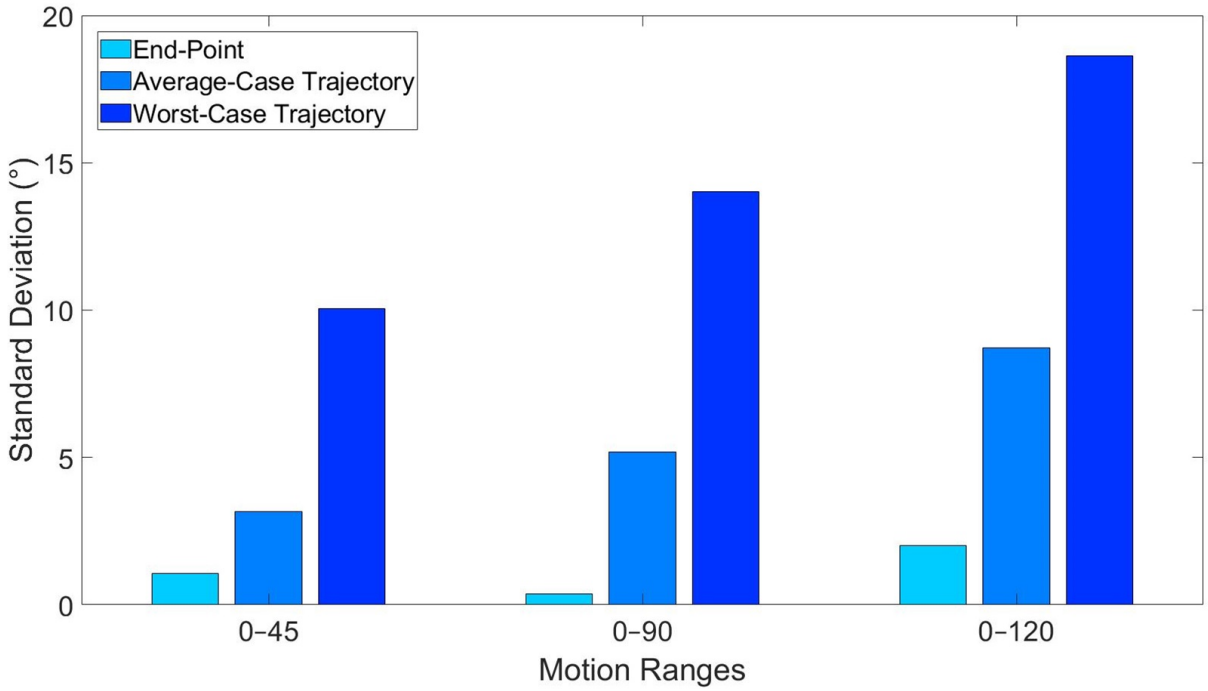


Figure 6.2: Comparison of the standard deviation metrics collected from the data of a single subject.

6.3.2 Collection and Analysis of Patient Data

Increasing the adoption of wearable assistive devices into rehabilitation settings requires the development of software systems that enable the therapist to collect and analyze data related to the patient and their motion. Wearable assistive devices or other mechatronic and sensing systems may be able to provide more data than is currently possible to collect with exiting techniques. In order to determine the data collection and analysis needs of the therapists, a national survey was conducted, which targeted licensed therapists practising in Canada. An online survey was developed to determine the data collection and analysis practises of physiotherapists, occupational therapist, and kinesiologists. It consisted of 18 questions and took approximately 10–15 minutes to complete. The survey was distributed through provincial and national organizations, whose members are one of the three types of therapists. Inclusion criteria required that therapists work with patients who suffer from MSDs affecting the upper limb and that they held a valid license with a provincial or national regulatory organization.

The survey questions were developed with the assistance of an expert upper limb therapist and

were categorized into four categories. The first category of questions provided data relevant to the general assessment techniques, important data, and collection methods. More specifics about the assessment methods and metrics were asked using the questions in the second category, while specifics about the collected data were detailed through the questions in the third category. The fourth category of questions was used to determine how therapists would like to visualize the data for analysis and what data they would like to track over time. In total, 33 therapists participated in the survey, with 11 partial responses and 22 full responses.

Upon analyzing the results of the survey, there were quite a few interesting findings. First, therapists collect both qualitative and quantitative data. The results suggest that rehabilitation processes involve the pairing of quantitative with qualitative data during assessment of the patient and their motion. This is likely due to the subjectivity in some of the data. However, increasing the possibilities for quantifying data may reduce some of the subjectivity inherent in the interaction. Second, the most commonly mentioned data were linked to one of either the patient history, pain, motion, patient activities, and strength. Every injury case is unique, but it seems that looking for data within these five areas provides a good starting point for assessment. Fig. 6.3 shows the responses from participants about the general characteristics of either the patient or motion that they assess. Third, EMG signals were not collected by any of the participants. However, five of the participants responded that they would be interested in tracking the EMG signals over time. This lack of use of EMG signals may be attributed to a lack of tools or a lack of understanding of the relationships between these signals and aspects of the motion or tissue health.

The fourth finding was that the participants are interested in various forms of the visualizations. In some cases, visualizations may make changes in outcomes more clear than analyzing large sets of numerical values and could be easier to demonstrate the changes to patients. Given the appropriate software systems, it may also be possible to provide the patients with these visualization tools. This could help with therapy compliance or provide a better understanding of the therapy to the patients. Fig. 6.4 shows a broad range of visualization methods chosen by the survey participants. Fifth, the participants of the survey showed that they would be interested in tracking many quantities over time. Furthermore, there is some interest in tracking quantities that they might currently not be able to track, such as EMG signals. Fig. 6.5 presents a list of the quantities that

participants wish to track over time.

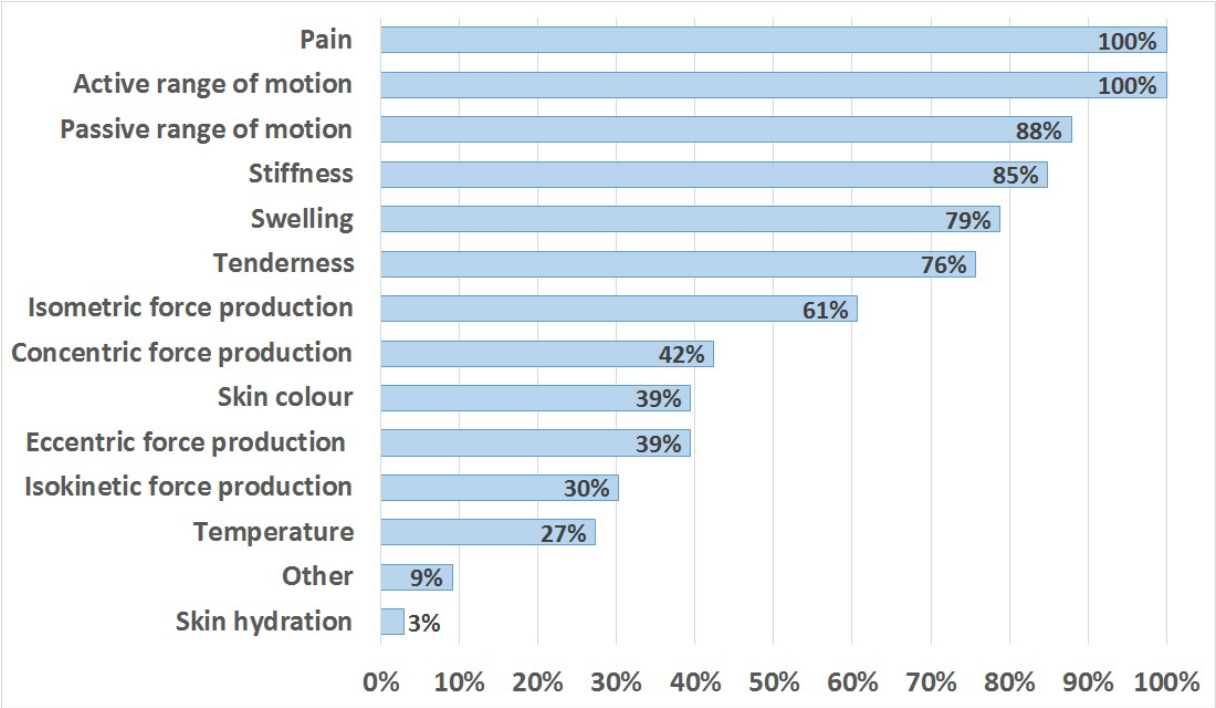


Figure 6.3: Participant responses for Question 3: "What general characteristics of the patient and their motion abilities are useful during your assessment?"

The final and most profound finding from the survey was that the view of the rehabilitation process varies significantly among therapists. Question 6 asked the participants: "What are the major stages that the patient moves through during the rehabilitation process?" After reviewing the responses, the processes described by the therapists have been categorized based on the focus of the process. Each of the rehabilitation process descriptions were categorized into one or more of the following process orientations: goals, tissue healing, tasks, severity, time, mentality, and administrative. Goal-oriented processes are centered around setting goals and working toward achieving them, such as regaining the ability to perform an activity of daily living. Tissue healing-oriented processes are focused on the three major phases of tissue healing: inflammatory, fibroblastic repair, and remodelling. Task-oriented processes aim to improve characteristics that are important to proper function of the human motion systems, such as reduction in pain or increases in muscular strength, through the completion of motion tasks. Severity-oriented processes are focused on the severity of the injury, such that the patient should move from physiotherapy as

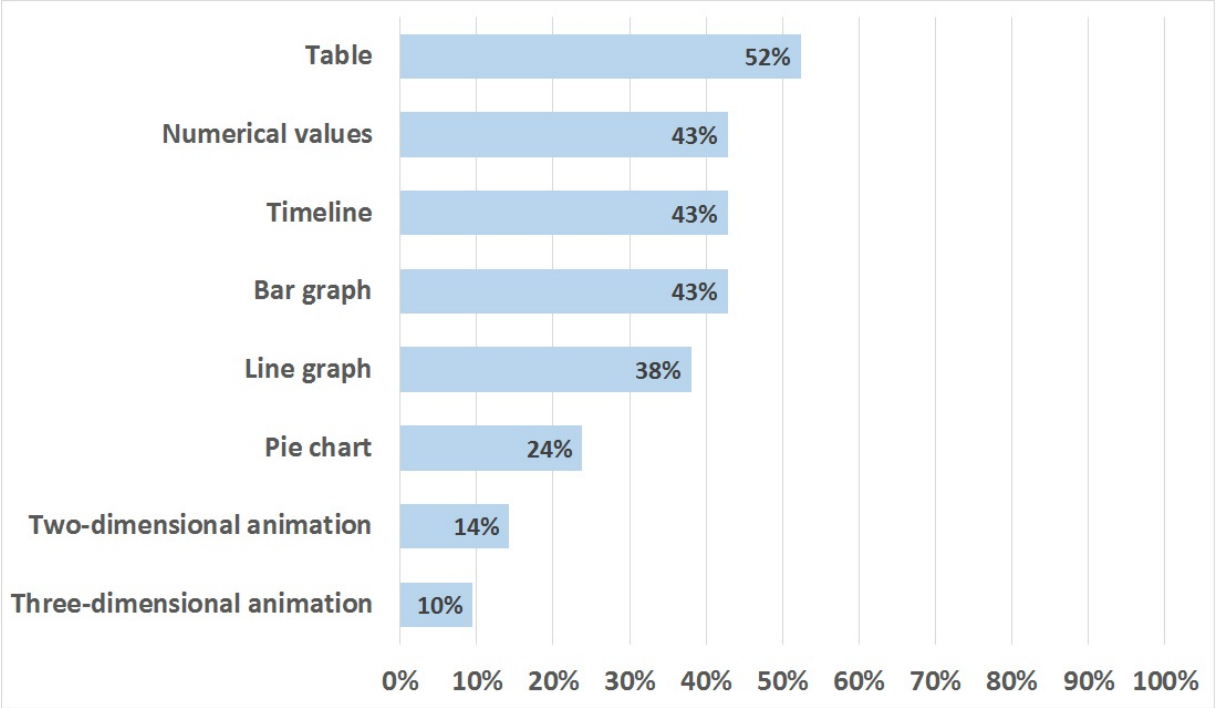


Figure 6.4: Participant responses for Question 17: "Which of the following visual representations would you prefer to see quantitative patient data displayed in?"

first-aid to physiotherapy that is more rehabilitative in nature. Time-oriented processes structure efforts based on the timeline of the injury. Mentality-oriented processes attempt to build trust between the therapist and patient in order to improve rehabilitation outcomes. Administrative-oriented processes are concerned with the high-level tasks that the patient completes from their initial visit with a physician to their eventual release from the rehabilitation program.

Although the participants provided different view points of the major stages of the rehabilitation process, there exists some connection between all of these view points. The therapists may have only been considering one or two of these view points in their description. However, the general rehabilitation process is quite complex and it seems logical that the therapists would attempt to simplify this process when responding to Question 6. Fig. 6.6 presents an attempt to link these rehabilitation process orientations together. This figure shows that there is a general linear administrative process, in which the most time consuming phase is the treatment. The treatment is broken down into three general phases of assessment, planning, and intervention, which are conducted iteratively until the patient is ready to be released from the rehabilitative program. Based

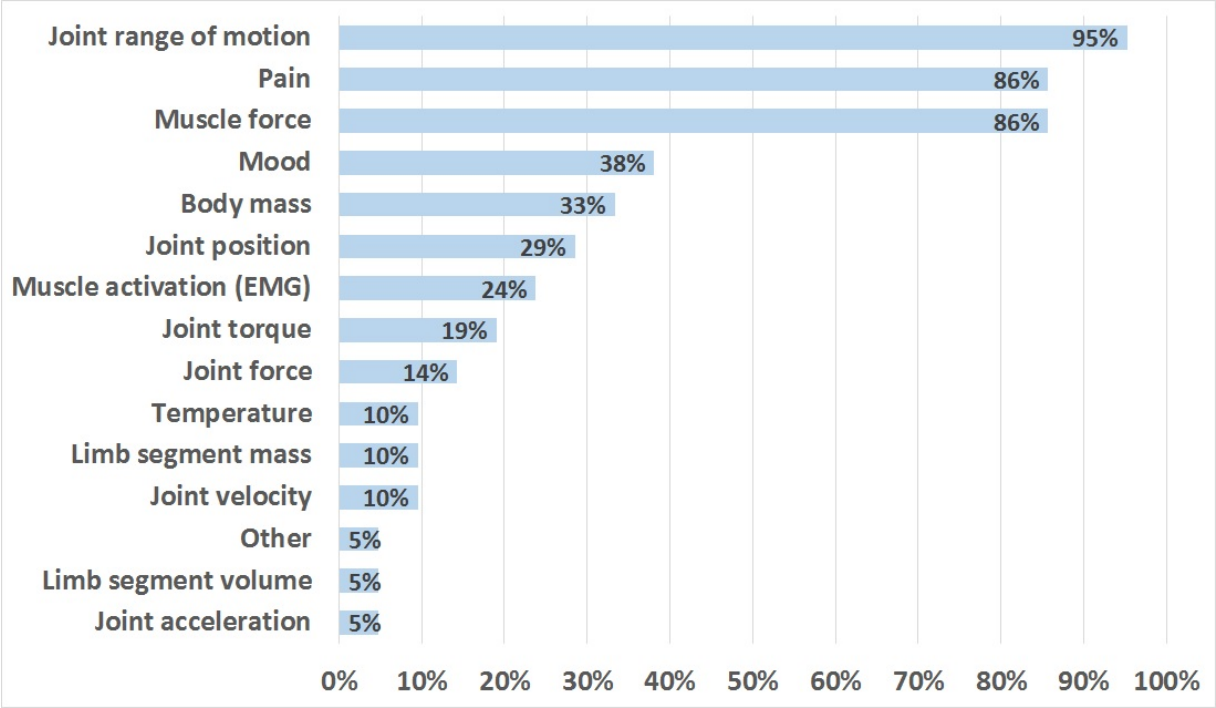


Figure 6.5: .

Participant responses for Question 18: "What patient data would you like to track over time? Please include any quantities that are not available for you to measure with your current set of tools or quantities that may be difficult to measure."

on these ideas, the rehabilitation process is viewed as a multi-layer structure, where the process orientations described above all interconnect. Further studies will help to identify the components and interactions of this complex structure.

In conclusion, the results of the survey have shown that there exists a lot variability in the collected data and assessment techniques used in musculoskeletal rehabilitation. Some of this variability may be due to the uniqueness of the musculoskeletal injury, the patient, the therapist, and the tools that are available to the therapist. Rehabilitation seems to be some part artistic and some part scientific in nature, due to all of the subjectivity. The use of digital data collection and analysis tools could help to alleviate some of this subjectivity, which would help in assessment of patients and tracking their outcomes over time. There are many opportunities to improve musculoskeletal rehabilitation using wearable devices or other digital devices, with the potential benefits being for both the patient and the therapist. One important task to complete is to generate requirements for researchers and engineers to develop these digital systems and provide

these benefits.

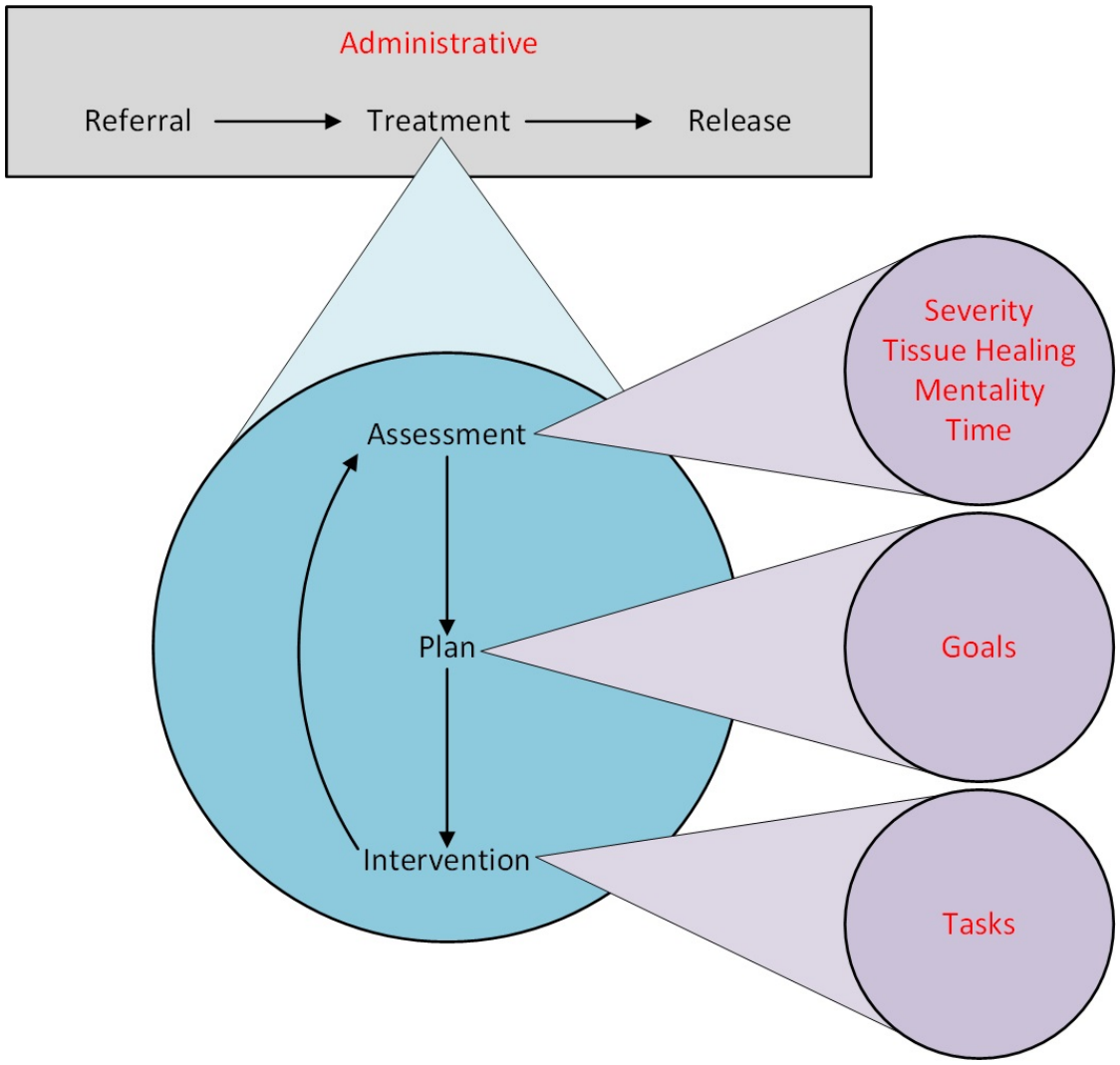


Figure 6.6: A depiction of the multi-layer general rehabilitation process that connects the rehabilitation process orientations identified from the survey responses. The survey responses were elicited from the question: "What are the major stages that the patient moves through during the rehabilitation process?"

6.3.3 Data Collection and Analysis Software Requirements

Wearable assistive devices are operated through digital control systems, which can be communicated with using other digital systems. Whether these systems are to be used to assist with motion, quantify patient motion, or both, software applications will need to be developed that allow users

to interact with them digitally. A primary goal of the national therapist survey was to begin gathering requirements for software systems that would enable collection and analysis of patient data. These software applications should allow for the collection of data from both traditional sources and from digital systems, such as wearable assistive devices. Based on the results of the survey, a software requirement analysis was performed.

One form of software requirement elicitation involves the investigation of natural language descriptions. Essentially, the nouns, verbs, adjectives, and adverbs of natural language descriptions can be used to determine objects, attributes, functions, and relationships of the software system. To perform this analysis on the survey results, the first step was to take the results and write them into paragraph form. The paragraphs were then analyzed for words that would be good candidates for different portions of object-oriented software designs. In cases where this did not make sense to do, word counting was done on the aggregated responses to determine which words were most prevalent between responses. The results of this analysis produced hundreds of potential software requirements, objects, attributes, functions, and relationships.

From the collected results, the next step would be to create some preliminary software use cases and designs and request feedback from end-users. This process would be repeated iteratively until the software designs are robust enough to be implemented and the therapists feel that all of their requirements would be met. Since a further study would be required to gather feedback, the software requirements harvested from the survey have not been verified. In order to highlight the important functionality harvested from the survey, a series of 14 abstract software requirements is presented. Keep in mind, each of these requirements would need to be decomposed into a more detailed sets of requirements. Based on the software requirement analysis, it was found that the system shall meet the following requirements:

1. provide standard data record templates for common rehabilitative tasks,
2. allow the user to design and customize data records to match the needs of the particular rehabilitative task,
3. allow the user to collect both qualitative and quantitative data,
4. provide the option for users to attach notes to recorded data,

5. provide standard rehabilitation process templates for analyzing patient progress,
6. allow the user to design and customize rehabilitation process templates to match the needs of the particular rehabilitation process,
7. collect and record time data with each data entry,
8. import digital data written by the patient into the database of records,
9. collect and store digital data directly from external digital devices,
10. provide standard patient evaluation scales as part of data records,
11. allow the user to design and customize patient evaluation scales,
12. generate visualizations based on the quantitative data,
13. complete numerical analyses of the quantitative data, and
14. generate reports based on the either the collected data, the completed analyses, or both.

As shown in this software requirements list, it is important that the developed software system provides a lot of flexibility. The thought is that this will help the adoption of both this software system and wearable assistive devices or other digital data collection devices. The survey results showed a lot of variability and it is speculated that this is inherent to the practise of musculoskeletal rehabilitation. Therefore, enabling the user to customize part of the system to meet the needs of their patients, clinics, tools, and processes is important. Three main areas of variability are seen in the specific data that are collected, the structure of the rehabilitation process, and the assessment and analysis performed on the patients and their data. Designing the software systems such that the components are customizable increases the potential benefit, as the number of scenarios in which it can be used increases. Further studies surrounding the development of software systems for patient data collection and analysis are needed to refine the requirements. Conducting these studies will help increase the amount of data available to both therapists and researchers, which could ultimately lead to better rehabilitation outcomes for future patients.

6.4 Summary

In general, only a small number of opportunities for wearable assistive devices to aid with musculoskeletal rehabilitation have been investigated. These devices are still in their infancy and require further development to produce both safe and useful behaviours. As discussed in this chapter, there are two main areas to explore once these devices have matured. First, wearable assistive devices can provide a huge reduction of labour to medical professionals working in musculoskeletal rehabilitation. In the future, they may also allow patients to perform their exercises remotely and relay the information back to their therapists. The key to this potential benefit is working with end-users to tailor the control systems in such a way that meets the needs of a particular motion therapy.

The second area to be explored is using these devices to quantify and generate patient data. Control systems could be developed for zero-impedance modes, in which the user can move freely and the device simply records the motion. The benefit of this application is that large amounts of highly accurate data can be generated by the devices. Using these data, motion qualities of natural human motion, such as the repeatability study presented in this chapter, and of patients suffering from MSDs can be investigated and compared more thoroughly. Once these devices are easy to configure and use, the amount of data that can be generated across multiple clinics would be quite large, which is desired from a research perspective.

To realize either of these major usages, the focus lies on increasing the adoption of the wearable assistive devices. The data generated by these devices need to be easy to access and manipulate for analysis. A national survey of Canadian therapist suggests that there is a lot of variability in data that are collected and used for assessment. Therefore, software systems that enable collection of patient data should provide therapists with both standard data collection templates and the option to customize these templates to meet their particular needs. The development of such systems would also enable direct interfacing with the wearable assistive devices, in order to gather the data they produce. Together, these two systems could reduce the physical effort of therapists and provide them with more rich data to enhance their assessments. The net benefit of these efforts could lead to better rehabilitation outcomes for patients.

Chapter 7

Conclusions

7.1 Conclusions

The research presented in this thesis was conducted to better understand the successes of and challenges facing the development of digital control systems for the control of wearable assistive devices. Since these control systems and devices are still immature, there are many opportunities for improving their abilities and assessing their potential as tools for musculoskeletal rehabilitation. Nevertheless, this research has resulted in many successes. Overall, the experiments showed that it is possible to develop digital control systems that can accurately track elbow motion using a wearable assistive device. Some the most accurate tracking performances found in the literature have been seen through the results of these experiments. To facilitate these experiments and isolate control components for testing, the WearMECS framework and integration testing protocol were developed. The results of these experiments show the efficacy of these two systems as control system development tools. Due to these tools, comparisons of various experimental factors, devices, motion inputs, control outputs, motion models, and metrics were more efficient. Statistical analysis of these experiments also revealed the complexity of the interactions between these aspects. Through these comparisons and analyses, relationships and trends were discovered, such as those between the accuracy and resource utilization of the control system. Overall, the research promotes a positive outlook for the potential of wearable assistive devices to assist with musculoskeletal rehabilitation.

The results of the research also posed many questions and highlighted the challenges that must

be solved in order to realize the benefits of these devices. Based on this research and that of others, it is clear that optimal solutions for these digital control systems do not yet exist. One part of this problem is caused by insufficient motion estimation models. In order to simplify these models, parameters are often used or derived that are static. However, this is hardly the case of the human physiology, as it is constantly evolving over time. EMG signals are also a popular choice for these models, despite the general lack of understanding of their dynamics. It is evident that motion intention is embedded within this signal, but the common signal processing techniques do not decode this intention accurately. The results of the experiments show that even small variations in EMG signals, relative to the ones used to optimize the model, can result in extremely large position estimation errors. In this research, four elbow motion models were developed and compared, but this just scratches the surface of the studies that are required to understand human motion and improve upon these models.

Digital technology is also a limiting factor for these control systems. It was shown in the timing and data storage analysis that short task execution durations lead to higher accuracy and that the data storage needs might limit the methods that are possible for storing data on an external device. It also important to keep in mind that the results of this research, and of those found in the literature, are determined using more powerful computer systems than can be integrated easily into wearable assistive devices. Without improving digital technologies, the usefulness of these devices may be limited. Coupling this with the variability inherent to the profession of musculoskeletal rehabilitation, it makes it difficult to develop control solutions that can be generalized. Currently, there is a lack of requirements for both the technology and the intended applications. This thesis provides some insights into these requirements, but further studies are still needed. The difficulty in this task, and the general development of these control systems, is related to the multi-disciplinary nature of the devices and the applications. One solution that can help push these devices toward musculoskeletal rehabilitaiton is the standardization of important aspects, as currently there are no standards for the design, implementation, and testing of these control systems and devices. Developing these standards will help in the evaluation and comparison of future control system solutions.

7.2 Contributions

The contributions of this thesis are seen through the development, evaluation, and comparison of digital control systems for wearable assistive devices and their application to musculoskeletal rehabilitation.

The first contribution is the proposal and design of control system development tools. Comparing across the existing control solutions space was made more efficient through the development of the WearMECS framework, the integration testing protocol, and the metric suite. The WearMECS framework made the task of adapting control software implementation to various situations much less demanding. During the experiments, the framework enabled the modification of the developed control systems for different sensing systems, motion models, devices, and control system tasks, without any control system architectural overhauls. The integration testing protocol allowed for portions of the control system to be isolated during experimental evaluations. Furthermore, it provided a method for increasing the integration of the human and device over time, while enabling assessment of the tracking performance and maintaining the safety of the both the human and the device. Finally, the quality model and metric suite expanded the types of data that are available from which to draw insights about these control solutions.

The second contribution is the development of control systems capable of tracking elbow motion with wearable assistive devices and the findings from their usage in the motion tracking experiments. During this research, four major control system variations were developed that formed the control software implementations used in the experiments. The primary objective of these systems was to regulate the behaviour of wearable assistive devices to track the elbow motion inputs of humans. These types of elbow motion tracking studies do exist in the literature, but the number of researchers who measured aspects of the tracking performance is quite low. Across all seven experiments, the accuracy and repeatability of these systems have been quantified, which adds significantly to the lack of control system assessments. Furthermore, some of the tracking performances achieved accuracy and repeatability levels that exceeded those of existing studies on both motion models estimations and motion tracking with devices. The statistical analyses performed on the experimental results also revealed insights regarding factors

that improve the tracking performance, such as visual feedback, motion training, constrained motion, and position-driven control. Finally, the first online remote controlled experiment for elbow motion tracking tasks with a wearable mechatronic elbow device was performed. Based on the literature, no other studies of this kind had been performed at this level of the integration testing protocol.

The third contribution comes from the review, development, and comparison of motion estimation models. The review of upper limb motion models showed that a huge spectrum of potential solutions have been proposed for estimating human motion. In this research, four elbow motion models were developed, based on these existing solutions. These models were used within the control systems to provide elbow motion estimates for the tracking tasks conducted during the experiments. Using the KFMM, the estimation accuracies were improved significantly, from existing studies with this model, during the software simulation phases of testing of Experiments 1–5. Experiment 7 was also the first time this model has been used for tracking elbow motion under real-time device reaction. The study performed in Experiment 6 was the first comparative study of muscle activation models that could be found in the literature. The results of which established the trade-off between accuracy and optimization time. Based on the reviewed motion models, Experiment 7 provided the first comparison of motion models that were neither HTMMs or ANNs, as well as the first implementation of dynamic elbow motion estimation using nonlinear polynomial equations. Together, these studies have identified the limitations of these models and the need for improvement.

The fourth contribution stems from exploration of the computational constraints. In addition to accuracy and repeatability, the last two experiments provided measurements of adaptability and resource utilization. The first finding was that the optimization time increases as the parameter space grows during an optimization task. Although one may expect this result, it has not yet been quantified in the literature. The second interesting finding was that the position errors were proportional to the tracking task execution time and inversely proportional to the optimization task execution times. This suggests that lower errors can be generated for both a great amount of optimization time and smaller amount of tracking task execution time. During the last experiment, the first quantification of data storage needs was completed. The third finding

was that the results of this endeavour suggest that common wireless communication protocols may not be able to meet the demand for data storage, which could create a problem for these devices to communicate data wirelessly. Most importantly, these results highlighted the need to determine the relationships between the motion tracking performance and digital systems that execute the control behaviours.

The fifth and final contribution is the assessment of requirements needed to tailor wearable assistive devices for specific musculoskeletal rehabilitation applications. Wearable assistive devices will only provide benefit if they are adopted within their intended application domains. Increasing this adoption rate comes from improving the existing devices and tailoring them to the needs of the particular application. Knowing what improvements to make is determined by gathering requirements from the patients and therapists that will use these devices. To begin tackling this problem, a national survey of Canadian therapists was conducted to identify data collection and analysis methods. The results show that these devices require a high amount of flexibility, since the variability of musculoskeletal rehabilitation is also high. A preliminary list of software requirements has been gathered from the results of the survey, but further studies involving feedback from end-users are required to continue developing software that has the potential to increase the amount of data and the number of analysis tools available to the therapists. The development of these systems will also allow for the integration of wearable assistive devices to be used as data collection devices within rehabilitation programs.

7.3 Future Directions

The potential benefit of wearable assistive devices to aid with musculoskeletal rehabilitation is large, but exploration is still required in many directions. The direction that is the simplest and most lacking is in quantifying patient–therapist motion interactions. These devices may never fulfill their intended goals, with regards to musculoskeletal rehabilitation, if there are no goals at which to aim. Even with improved motion tracking, the control systems still need to know what force trajectory to apply and how to evolve those trajectories based on inputs from the users. This direction is tied heavily with a better understanding of human motion signals, characteristics, and

relationships. The most responsive wearable assistive devices will still be limited by the estimates of the user's intended motion. It is clear that many efforts have been made to understand human motion and develop models of dynamic motion properties. However, even in some of the best cases, the errors are still too large to ensure the safety of the user and device or the models are not robust enough to the number of possible variations. There is difficulty in determining these characteristics, as humans are complex systems. Sometimes, it is even unethical to perform the studies, such as producing pain in subjects to study its effects, but the limits of these devices are directly correlated with an understanding of the human body and the motion it produces.

It seems that wearable assistive devices are still in their infancy, relative to where they need to be for usage in musculoskeletal rehabilitation. Advancing the field comes from more measurements, more comparison, more tools, and more participation from application domain experts. The lack of quantification of control system aspects makes it difficult for developers to assess the state-of-the-art and make informed decisions about how to push forward. Even once control system architectures and studies have been decided, it is difficult to perform comparisons, since no standards exist for classifying these systems and components. The control system development tools proposed in this thesis were shown to be particularly useful for development, modification, and comparison of control system components, but the prevalence of these tools is quite small. This likely comes from the fact that to develop these tools requires abstracting aspects of the control systems, which in itself is a laborious task and whose efforts could be spent developing control systems instead. Many of the efforts of control system and wearable assistive device developers could be more focused if there were more participation from domain application experts, such as therapists. Currently, these developers must rely mainly on improving the engineering aspects of these systems, which may be less fruitful efforts if the engineering requirements do not align with the application requirements. The work presented in this thesis investigates each of these directions, but the amount of research needed for these control systems to meet their intended applications is vast. With each future effort, the knowledge of human motion, control systems, and rehabilitation will grow, moving closer to the realization of the benefits these devices can provide.

References

[1] C. I. of Musculoskeletal Health and Arthritis, "STRATEGIC PLAN 2014–2018: enhancing musculoskeletal, skin and oral health," 2014. [Online]. Available: http://www.cihr-irsc.gc.ca/e/48830.html

[2] S. Canada, "Quarterly demographic estimates: January to march 2016," Statistics Canada Catalogue no. 91-002-X. Ottawa, 2016. [Online]. Available: http://www.statcan.gc.ca/pub/91-002-x/91-002-x2016001-eng.pdf

[3] SMARTRISK, "The economic burden of injury in canada," SMARTRISK: Toronto, ON, 2009. [Online]. Available: http://www.parachutecanada.org/downloads/research/reports/EBI2009-Eng-Final.pdf

[4] J.-M. Billette and T. Janz, "Injuries in canada: Insights from the canadian community health survey," Statistics Canada Catalogue no. 82-624-X, 2011. [Online]. Available: http://www.statcan.gc.ca/pub/82-624-x/2011001/article/11506-eng.htm

[5] P. S. Lum, C. G. Burgar, P. C. Shor, M. Majmundar, and M. Van der Loos, "Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke," *Archives of Physical Medicine and Rehabilitation*, vol. 83, no. 7, pp. 952–959, 2002.

[6] B. T. Volpe, D. Lynch, A. Rykman-Berland, M. Ferraro, M. Galgano, N. Hogan, and H. I. Krebs, "Intensive sensorimotor arm training mediated by therapist or robot improves hemiparesis in patients with chronic stroke," *Neurorehabilitation and Neural Repair*, vol. 22, no. 3, pp. 305–310, 2008.

[7] S. J. Page, V. Hill, and S. White, "Portable upper extremity robotics is as efficacious as upper extremity rehabilitative therapy: a randomized controlled pilot trial," *Clinical Rehabilitation*, vol. 27, no. 6, pp. 494–503, 2012.

[8] H. T. Peters, S. J. Page, and A. Persch, "Giving them a hand: wearing a myoelectric elbow-wrist-hand orthosis reduces upper extremity impairment in chronic stroke," *Archives of Physical Medicine and Rehabilitation*, vol. 98, no. 9, pp. 1821–1827, 2017.

[9] R. A. R. C. Gopura, D. S. V. Bandara, K. Kiguchi, and G. K. I. Mann, "Developments in hardware systems of active upper-limb exoskeleton robots: A review," *Robotics and Autonomous Systems*, vol. 75, pp. 203–220, 2015.

[10] N. Hogan, H. I. Krebs, J. Charnnarong, P. Srikrishna, and A. Sharon, "MIT-MANUS: a workstation for manual therapy and training." in *IEEE International Workshop on Robot and Human Communication*, September 1992, pp. 161–165.

[11] R. U. I. Loureiro, F. Amirabdollahian, M. Topping, and B. Driessen, "Upper limb robot mediated stroke therapy — GENTLE / s approach," *Autonomous Robots*, vol. 15, pp. 35–51, 2003.

[12] G. Fazekas, M. Horvath, and A. Toth, "A novel robot training system designed to supplement upper limb physiotherapy of patients with spastic hemiparesis." *International Journal of Rehabilitation Research*, vol. 29, no. 3, pp. 251–4, September 2006.

[13] J. L. Pons, *Wearable Robots: Biomechatronic Exoskeletons.* Chinchester, West Sussex, England: John Wiley & Sons Ltd., 2008.

[14] Hocoma, "Armeo ®power," 2018. [Online]. Available: https://www.hocoma.com/solutions/armeo-power/

[15] J. Orthopedic, "WREX: Wilmington robotic exoskeleton arm," 2018. [Online]. Available: http://jaecoorthopedic.com/products/products/WREX\%3A-Wilmington-Robotic-EXoskeleton-Arm.html

[16] D. Matsuura, N. Iwatsuki, and M. Okada, "Redundancy optimization of hyper redundant robots based on movability and assistability," in *IEEE International Conference on Intelligent Robots and Systems*, San Diego, California, USA, October 2007, pp. 2534–2539.

[17] S. Park, K. Chung, and S. Jayaraman, "Chapter 1.1 - wearables: Fundamentals, advancements, and a roadmap for the future," in *Wearable Sensors*, E. Sazonov and M. R. Neuman, Eds. Oxford: Academic Press, 2014, pp. 1–23.

[18] J. P. Miguel, D. Mauricio, and G. Rodríguez, "A review of software quality models for the evaluation of software products," *International Journal of Software Engineering & Applications*, vol. 5, no. 6, pp. 31–53, 2014.

[19] A. L. Trejos, "Lecture notes on advanced medical mechatronics design," January-April 2014, department of Electrical and Computer Engineering, The University of Western Ontario, London, Ontario, Canada.

[20] R. A. R. C. Gopura, K. Kiguchi, and D. S. V. Bandara, "A brief review on upper extremity robotic exoskeleton systems," in *International Conference on Industrial and Information Systems*, vol. 8502, Kandy, Sri Lanka, August 2011, pp. 346–351.

[21] H. S. Lo and S. Q. Xie, "Exoskeleton robots for upper-limb rehabilitation: state of the art and future prospects." *Medical engineering & physics*, vol. 34, no. 3, pp. 261–8, April 2012.

[22] P. Maciejasz, J. Eschweiler, K. Gerlach-Hahn, A. Jansen-Troy, and S. Leonhardt, "A survey on robotic devices for upper limb rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 11, no. 3, pp. 1–29, 2014.

[23] A. Toth, G. Fazekas, G. Arz, M. Jurak, and M. Horvath, "Passive robotic movement therapy of the spastic hemiparetic arm with REHAROB: Report of the first clinical test and the follow-up system improvement," in *International Conference on Rehabilitation Robotics*, Chicago, IL, USA, June 2005, pp. 127–130.

[24] N. Vitiello, T. Lenzi, S. Roccella, S. D. Rossi, E. Cattin, F. Giovacchini, F. Vecch, and M. Carrozza, "NEUROExos: A powered elbow exoskeleton for physical rehabilitation," *IEEE Transaction on Robotics*, vol. 29, no. 1, pp. 220–235, February 2013.

[25] A. Kyrylova, "Development of a wearable mechatronic elbow brace for postoperative motion rehabilitation," Master's thesis, The University of Western Ontario, August 2015.

[26] J. M. McBean and K. Narendran, "Powered orthotic device and method of using same," 2013.

[27] J. Stein, K. Narendran, J. McBean, K. Krebs, and R. Hughes, "Electromyography-controlled exoskeletal upper-limb-powered orthosis for exercise training after stroke." *American Journal of Physical Medicine and Rehabalitation*, vol. 86, no. 4, pp. 255–261, 2007.

[28] G. J. Kim, L. Rivera, and J. Stein, "Combined clinic-home approach for upper limb robotic therapy after stroke: A pilot study," *Archives of Physical Medicine and Rehabilitation*, vol. 96, no. 12, pp. 2243–2248, 2015.

[29] T. G. Sugar, J. He, E. J. Koeneman, J. B. Koeneman, R. Herman, H. Huang, R. S. Schultz, D. E. Herring, J. Wanberg, S. Balasubramanian, P. Swenson, and J. a. Ward, "Design and control of RUPERT: a device for robotic upper extremity repetitive therapy." *IEEE Transaction on Neural Systems and Rehabilitation*, vol. 15, no. 3, pp. 336–346, September 2007.

[30] E. A. Brackbill, Y. Mao, S. K. Agrawal, M. Annapragada, and V. N. Dubey, "Dynamics and control of a 4-dof wearable cable-driven upper arm exoskeleton," in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009, pp. 2300–2305.

[31] S. Lessard, P. Pansodtee, A. Robbins, L. B. Baltaxe-admony, J. M. Trombadore, M. Teodorescu, A. Agogino, and S. Kurniawan, "CRUX : a Compliant Robotic Upper-extremity eXosuit for lightweight , portable , multi-joint muscular augmentation," in *International Conference on Rehabilitation Robotics*, London, United Kingdom, July 2017, pp. 1633–1638.

[32] E. Rocon, J. M. Belda-Lois, A. F. Ruiz, M. Manto, J. C. Moreno, and J. L. Pons, "Design and validation of a rehabilitation robotic exoskeleton for tremor assessment and suppression," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 3, pp. 367–378, 2007.

[33] J. Ueda, D. Ming, V. Krishnamoorthy, M. Shinohara, and T. Ogasawara, "Individual muscle control using an exoskeleton robot for muscle function testing," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 4, pp. 339–350, 2010.

[34] Z. G. Xiao, A. M. Elnady, J. Webb, and C. Menon, "Towards a brain computer interface driven exoskeleton for upper extremity rehabilitation," in *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, São Paulo, Brazil, August 2014, pp. 432–437.

[35] T. Ando, M. Watanabe, K. Nishimoto, Y. Matsumoto, M. Seki, and M. G. Fujie, "Myoelectric-controlled exoskeletal elbow robot to suppress essential tremor: Extraction of elbow flexion movement using STFTs and TDNN," *Journal of Robotics and Mechatronics*, vol. 24, no. 1, pp. 141–149, 2012.

[36] L. M. Vaca Benitez, M. Tabie, N. Will, S. Schmidt, M. Jordan, and E. A. Kirchner, "Exoskeleton technology in rehabilitation: Towards an EMG-based orthosis system for upper limb neuromotor rehabilitation," *Journal of Robotics*, vol. 2013, 2013.

[37] T. Desplenter, A. Kyrylova, T. K. Stanbury, S. Chinchalkar, A. Escoto, and A. L. Trejos, "A wearable mechatronic brace for arm rehabilitation," in *IEEE International Conference on Biomedical Robotics and Biomechatronics*, Sao Paulo, Brazil, August 2014, pp. 491–496.

[38] G. Herrnstadt and C. Menon, "On-off tremor suppression orthosis with electromagnetic brake," *International Journal of Mechanical Engineering and Mechatronics*, vol. 1, no. 2, pp. 7–14, 2013.

[39] K. Kim, K. J. Hong, N. G. Kim, and T. K. Kwon, "Assistance of the elbow flexion motion on the active elbow orthosis using muscular stiffness force feedback," *Journal of Mechanical Science and Technology*, vol. 25, no. 12, pp. 3195–3203, 2011.

[40] J. KleinJan, "Design of an ultra-compact elbow-joint for an assistive device," Ph.D. dissertation, Delft University of Technology, 2013.

[41] R. Looned, J. Webb, Z. Xiao, and C. Menon, "Assisting drinking with an affordable BCI-controlled wearable robot and electrical stimulation: a preliminary investigation," *Journal of NeuroEngineering and Rehabilitation*, vol. 11, no. 51, 2014.

[42] C. Pylatiuk, A. Kargov, I. Gaiser, T. Werner, S. Schulz, and G. Bretthauer, "Design of a flexible fluidic actuation system for a hybrid elbow orthosis," in *IEEE International Conference on Rehabilitation Robotics*. Kyoto, Japan: IEEE, June 2009, pp. 167–171.

[43] Y. Ren, H. S. Park, Y. Li, L. Wang, and L.-Q. Zhang, "A wearable robot for upper limb rehabilitation of patients with neurological disorders," in *IEEE International Conference on Robotics and Biomimetics*, Tianjin, China, December 2010, pp. 64–68.

[44] Z. Tang, K. Zhang, S. Sun, Z. Gao, L. Zhang, and Z. Yang, "An upper-limb power-assist exoskeleton using proportional myoelectric control," *Sensors*, vol. 14, pp. 6677–6694, 2014.

[45] I. Vanderniepen, R. Van Ham, M. Van Damme, R. Versluys, and D. Lefeber, "Orthopaedic rehabilitation: A powered elbow orthosis using compliant actuation," in *IEEE International Conference on Rehabilitation Robotics*, Kyoto, Japan, June 2009, pp. 172–177.

[46] R.-J. Wang and H.-P. Huang, "AVSER Active variable stiffness exoskeleton robot system: Design and application for safe active-passive elbow rehabilitation," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Kaohsiung, Taiwan: IEEE, July 2012, pp. 220–225.

[47] G. Andrikopoulos, G. Nikolakopoulos, and S. Manesis, "Design and development of an exoskeletal wrist prototype via pneumatic artificial muscles," *Meccanica*, vol. 50, no. 11, pp. 2709–2730, 2015.

[48] T. Higuma, K. Kiguchi, and J. Arata, "Low-profile two-degree-of-freedom wrist exoskeleton device using multiple spring blades," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 305–311, 2018.

[49] S. Kazi, A. As'arry, M. Z. Zain, M. Mailah, and M. Hussein, "Experimental implementation of smart glove incorporating piezoelectric actuator for hand tremor control," *WSEAS Transactions on Systems and Control*, vol. 5, no. 6, pp. 443–453, 2010.

[50] R. C. V. Loureiro, J. M. Belda-Lois, E. R. Lima, J. L. Pons, J. J. Sanchez-Lacuesta, and W. S. Harwin, "Upper limb tremor suppression in ADL via an orthosis incorporating a controllable double viscous beam actuator," in *IEEE International Conference on Rehabilitation Robotics*, Chicago, IL, USA, June–July 2005, pp. 119–122.

[51] B. Taheri, "Real-time pathological tremor identification and suppression in human arm via active orthotic devices," Ph.D. dissertation, Southern Methodist University, 2013.

[52] Z. G. Xiao and C. Menon, "Towards the development of a portable wrist exoskeleton," in *IEEE International Conference on Robotics and Biomimetics*, Phukey, Thailand, December 2011, pp. 1884–1889.

[53] H. Al-Fahaam, S. Davis, and S. Nefti-Meziani, "Power assistive and rehabilitation wearable robot based on pneumatic soft actuators," in *International Conference on Methods and Models in Automation and Robotics*, Miedzyzdroje, Poland, August–September 2016, pp. 472–477.

[54] B. Allotta, R. Conti, L. Governi, E. Meli, A. Ridolfi, and Y. Volpe, "Development and experimental testing of a portable hand exoskeleton," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, Germany, September–October 2015, pp. 5339–5344.

[55] B. Allotta, R. Conti, E. Meli, A. Ridolfi, and L. Governi, "Development and testing of a low cost wearable and portable hand exoskeleton based on a parallel mechanism," in *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Boston, Massachusetts, USA, August 2015, pp. 1–10.

[56] R. Conti, E. Meli, and A. Ridolfi, "A novel kinematic architecture for portable hand exoskeletons," *Mechatronics*, vol. 35, pp. 192–207, 2016.

[57] J. Arata, K. Ohmoto, R. Gassert, O. Lambercy, H. Fujimoto, and I. Wada, "A new hand exoskeleton device for rehabilitation using a three-layered sliding spring mechanism," in *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 3902–3907.

[58] P. M. Aubin, H. Sallum, C. Walsh, L. Stirling, and A. Correia, "A pediatric robotic thumb exoskeleton for at-home rehabilitation: The Isolated Orthosis for Thumb Actuation (IOTA)," *International Journal of Intelligent Computing and Cynbernetics*, vol. 7, no. 3, pp. 233–252, 2014.

[59] T. M. Burton, R. Vaidyanathan, S. C. Burgess, A. J. Turton, and C. Melhuish, "Development of a parametric kinematic model of the human hand and a novel robotic exoskeleton," in

IEEE International Conference on Rehabilitation Robotics, Zurich, Switzerland, June–July 2011.

[60] H. Cao and D. Zhang, "Soft robotic glove with integrated sEMG sensing for disabled people with hand paralysis," in *IEEE International Conference on Robotics and Biomimetics*, Qingdao, China, December 2016, pp. 714–718.

[61] M. Cempini, S. M. M. De Rossi, T. Lenzi, M. Cortese, F. Giovacchini, N. Vitiello, and M. C. Carrozza, "Kinematics and design of a portable and wearable exoskeleton for hand rehabilitation," in *IEEE International Conference on Rehabilitation Robotics*, Seattle, Washington, USA, June 2013.

[62] A. Chiri, F. Giovacchini, N. Vitiello, E. Cattin, S. Roccella, F. Vecchi, and M. C. Carrozza, "HANDEXOS: Towards an exoskeleton device for the rehabilitation of the hand," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, USA, October 2009, pp. 1106–1111.

[63] A. Chiri, N. Vitiello, S. Member, F. Giovacchini, S. Roccella, F. Vecchi, M. C. Carrozza, and A. Member, "Mechatronic design and characterization of the index finger module of a hand exoskeleton for post-stroke rehabilitation," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 5, pp. 884 – 894, 2011.

[64] M. A. Delph, S. A. Fischer, P. W. Gauthier, C. H. M. Luna, E. A. Clancy, and G. S. Fischer, "A soft robotic exomusculature glove with integrated sEMG sensing for hand rehabilitation," in *IEEE International Conference on Rehabilitation Robotics*, Seattle, Washington, USA, June 2013.

[65] S. Fok, R. Schwartz, M. Wronkiewicz, C. Holmes, J. Zhang, T. Somers, D. Bundy, and E. Leuthardt, "An EEG-based brain computer interface for rehabilitation and restoration of hand control following stroke using ipsilateral cortical physiology," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, Massachusetts, USA, August–September 2011, pp. 6277–6280.

[66] S. Goutam and K. C. Aw, "Development of a compliant hand assistive device," in *IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, Senigallia, Italy, August 2014.

[67] A. Hadi, K. Alipour, S. Kazeminasab, A. Amerinatanzi, and M. Elahinia, "Design and prototyping of a wearable assistive tool for hand rehabilitation using shape memory alloys," in *ASME Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, Stowe, VT, USA, September 2016, pp. 1–7.

[68] H. K. In, K. J. Cho, K. R. Kim, and B. S. Lee, "Jointless structure and under-actuation mechanism for compact hand exoskeleton," in *IEEE International Conference on Rehabilitation Robotics*, Zurich, Switzerland, June–July 2011.

[69] H. In, B. B. Kang, M. K. Sin, and K. J. Cho, "Exo-Glove: A wearable robot for the hand with a soft tendon routing system," *IEEE Robotics and Automation Magazine*, vol. 22, no. 1, pp. 97–105, 2015.

[70] J. Iqbal, N. G. Tsagarakis, A. E. Fiorilla, and D. G. Caldwell, "A portable rehabilitation device for the hand," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Buenos Aires, Argentina, August–September 2010, pp. 3694–3697.

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

[72] J. Iqbal, N. G. Tsagarakis, and D. G. Caldwell, "Human hand compatible underactuated exoskeleton robotic system," *Electronics Letters*, vol. 50, no. 7, pp. 494–496, 2014.

[73] J. Iqbal, N. G. Tsagarakis, and D. G. Caldwell, "A multi-DOF robotic exoskeleton interface for hand motion assistance." in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, Massachusetts, U.S.A., August–September 2011, pp. 1575–1578.

[74] B. B. Kang, H. Lee, H. In, U. Jeong, J. Chung, and K. J. Cho, "Development of a polymer-based tendon-driven wearable robotic hand," in *IEEE International Conference on Robotics and Automation*, Stockholm, Sweden, May 2016, pp. 3750–3755.

[75] E. Matheson and G. Brooker, "Assistive rehabilitation robotic glove," in *Proceedings of Australasian Conference on Robotics and Automation*, Melbourne, Australia, December 2011.

[76] M. Mulas, M. Folgheraiter, and G. Gini, "An EMG-controlled exoskeleton for hand rehabilitation," in *IEEE International Conference on Rehabilitation Robotics*, Chicago, IL, USA, June–July 2005, pp. 371–374.

[77] C. J. Nycz, T. Butzer, O. Lambercy, J. Arata, G. S. Fischer, and R. Gassert, "Design and characterization of a lightweight and fully portable remote actuation system for use with a hand exoskeleton," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 976–983, 2016.

[78] P. Polygerinos, Z. Wang, K. C. Galloway, R. J. Wood, and C. J. Walsh, "Soft robotic glove for combined assistance and at-home rehabilitation," *Robotics and Autonomous Systems*, vol. 73, pp. 135–143, 2015.

[79] L. Saharan, A. Sharma, M. Jung de Andrade, R. H. Baughman, and Y. Tadesse, "Design of a 3D printed lightweight orthotic device based on twisted and coiled polymer muscle: iGrab hand orthosis," in *Active and Passive Smart Structures and Integrated Systems*, Portland, Oregon, USA, 2017.

[80] O. Sandoval-Gonzalez, J. Jacinto-Villegas, I. Herrera-Aguilar, O. Portillo-Rodiguez, P. Tripicchio, M. Hernandez-Ramos, A. Flores-Cuautle, and C. Avizzano, "Design and development of a hand exoskeleton robot for active and passive rehabilitation," *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, pp. 1–12, 2016.

[81] K. Y. Tong, S. K. Ho, P. M. Pang, X. L. Hu, W. K. Tam, K. L. Fung, X. J. Wei, P. N. Chen, and M. Chen, "An intention driven hand functions task training robotic system," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Buenos Aires, Argentina, August–September 2010, pp. 3406–3409.

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

[83] K. Xing, Q. Xu, J. He, Y. Wang, Z. Liu, and X. Huang, "A wearable device for repetitive hand therapy," in *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, Scottsdale, AZ, USA, October 2008, pp. 919–923.

[84] J. Wu, J. Huang, Y. Wang, K. Xing, and Q. Xu, "Fuzzy PID control of a wearable rehabilitation robotic hand driven by pneumatic muscles," in *International Symposium on Micro-NanoMechatronics and Human Science*, Nagoya, Japan, November 2009, pp. 408–413.

[85] J. Wu, J. Huang, Y. Wang, and K. Xing, "A wearable rehabilitation robotic hand driven by PM-TS actuators," in *International Conference on Intelligent Robotics and Application*. Berlin, Heidelberg: Springer Berlin Heidelberg, November 2010, pp. 440–450.

[86] H. K. Yap, J. H. Lim, F. Nasrallah, J. C. H. Goh, and R. C. H. Yeow, "A soft exoskeleton for hand assistive and rehabilitation application using pneumatic actuators with variable stiffness," in *IEEE International Conference on Robotics and Automation*, Seattle, Washington, USA, May 2015, pp. 4967–4972.

[87] H. K. Yap, B. W. Ang, J. H. Lim, J. C. Goh, and C. H. Yeow, "A fabric-regulated soft robotic glove with user intent detection using EMG and RFID for hand assistive application," in *IEEE International Conference on Robotics and Automation*, Stockholm, Sweden, May 2016, pp. 3537–3542.

[88] S.-S. Yun, B. B. Kang, and K.-J. Cho, "Exo-Glove PM: An easily customizable modularized pneumatic assistive glove," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1725–1732, 2017.

[89] J. Lobo-Prat, A. Q. L. Keemink, M. I. Paalman, E. E. G. Hekman, P. H. Veltink, A. H. A. Stienen, and B. F. J. M. Koopman, "Design and control of an experimental active elbow support for adult duchenne muscular dystrophy patients," in *IEEE International Conference on Biomedical Robotics and Biomechatronics*, Sao Paulo, Brazil, August 2014, pp. 187–192.

[90] J. Lobo-Prat, P. N. Kooren, A. H. Stienen, J. L. Herder, B. F. Koopman, and P. H. Veltink, "Non-invasive control interfaces for intention detection in active movement-assistive devices." *Journal of NeuroEngineering and Rehabilitation*, vol. 11, no. 1, p. 168, 2014.

[91] J. Lobo-Prat, P. N. Kooren, M. M. H. P. Janssen, A. Q. L. Keemink, P. H. Veltink, A. H. A. Stienen, and B. F. J. M. Koopman, "Implementation of EMG- and force-based control interfaces in active elbow supports for men with duchenne muscular dystrophy: A feasibility study," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 11, pp. 1179–1190, 2016.

[92] H. S. Park, Y. Ren, and L. Q. Zhang, "IntelliArm: An exoskeleton for diagnosis and treatment of patients with neurological impairments," in *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, Scottsdale, AZ, USA, October 2008, pp. 109–114.

[93] R. Gopura, K. Kiguchi, and Y. Li, "SUEFUL-7: A 7DOF upper-limb exoskeleton robot with muscle-model-oriented emg-based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, USA, October 2009, pp. 1126–1131.

[94] K. Kiguchi, M. Rahman, M. Sasaki, and K. Teramoto, "Development of a 3DOF mobile exoskeleton robot for human upper-limb motion assist," *Robotics and Autonomous Systems*, vol. 56, pp. 678–691, 2008.

[95] Q. Li, D. Wang, Z. Du, Y. Song, and L. Sun, "sEMG based control for 5 DOF upper limb rehabilitation robot system," in *IEEE International Conference on Robotics and Biomimetics*. Kunming, China: IEEE, December 2006, pp. 1305–1310.

[96] A. L. C. Ruiz, C. Pontonnier, J. Levy, and G. Dumont, "Motion control via muscle synergies: Application to throwing," in *ACM SIGGRAPH Conference on Motion in Games*, Paris, France, November 2015, pp. 65–72.

[97] P. R. Culmer, a. E. Jackson, S. G. Makower, J. a. Cozens, M. C. Levesley, M. Mon-Williams, and B. Bhakta, "A novel robotic system for quantifying arm kinematics and kinetics: description and evaluation in therapist-assisted passive arm movements post-stroke." *Journal of Neuroscience Methods*, vol. 197, no. 2, pp. 259–69, April 2011.

[98] N. G. Tsagarakis and D. G. Caldwell, "Development and control of a 'soft-actuated' exoskeleton for use in physiotherapy and training," *Autonomous Robots*, vol. 15, pp. 21–33, 2003.

[99] R. Vertechy, A. Frisoli, A. Dettori, M. Solazzi, and M. Bergamasco, "Development of a new exoskeleton for upper limb rehabilitation," in *IEEE/RAS-EMBS Internatioal Conference on Rehabilitation Robots*, Kyoto, Japan, June 2009, pp. 188–193.

[100] S. Moubarak, M. T. Pham, T. Pajdla, and T. Redarce, "Design and modeling of an upper extremity exoskeleton," in *World Congress on Medical Physics and Biomedical Engineering*, O. Dössel and W. C. Schlegel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, September 2009, pp. 476–479.

[101] A. Frisoli, F. Salsedo, M. Bergamasco, B. Rossi, and M. C. Carboncini, "A force-feedback exoskeleton for upper-limb rehabilitation in virtual reality," *Applied Bionics and Biomechanics*, vol. 6, no. 2, pp. 115–126, July 2009.

[102] A. Gupta and M. O'Malley, "Design of a haptic arm exoskeleton for training and rehabilitation," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 3, pp. 280–289, June 2006.

[103] J. Klein, S. Spencer, J. Allington, K. Minakata, E. Wolbrecht, R. Smith, J. Bobrow, and D. Reinkensmeyer, "Biomimetic orthosis for the neurorehabilitation of the elbow and shoulder (BONES)," in *IEEE International Conference on Biomedical Robotics and Biomechatronics*, Scottsdale, USA, 2008, pp. 535–541.

[104] F. Martinez, A. Pujana-Arrese, I. Retolaza, I. Sacristan, J. Basurko, and J. Landaluze, "IKO: a five actuated dof upper limb exoskeleton oriented to workplace assistance," *Applied Bionics and Biomechanics*, vol. 6, no. 2, pp. 143–155, June 2009.

[105] T. Nef, M. Guidal, and R. Riener, "ARMin iii arm therapy exoskeleton with an ergonomic shoulder actuation," *Applied Bionics and Biomechanics*, vol. 6, no. 2, pp. 127–142, June 2009.

[106] J. C. Perry, J. Rosen, and S. Burns, "Upper-limb powered exoskeleton design," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 4, pp. 408–417, August 2007.

[107] R. Sanchez, E. Wolbrecht, R. Smith, J. Liu, S. Rao, S. Cramer, T. Rahman, J. Bobrow, and D. Reinkensmeyer, "A Pneumatic Robot for Re-Training Arm Movement after Stroke: Rationale and Mechanical Design," in *International Conference on Rehabilitation Robotics*. IEEE, June–July 2005, pp. 500–504.

[108] T. Nef and P. Lum, "Improving backdrivability in geared rehabilitation robots." *Medical & Biological Engineering & Computing*, vol. 47, no. 4, pp. 441–7, April 2009.

[109] S. J. Ball, I. E. Brown, and S. H. Scott, "Performance evaluation of a planar 3DOF robotic exoskeleton for motor assessment," *Journal of Medical Devices*, vol. 3, no. 2, pp. 1–12, 2009.

[110] A. Frisoli, F. Rocchi, S. Marcheschi, A. Dettori, F. Salsedo, and M. Bergamasco, "A new force-feedback arm exoskeleton for haptic interaction in Virtual Environments," in *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Pisa, Italy, March 2005.

[111] E. Wolbrecht, D. Reinkensmeyer, and J. Bobrow, "Pneumatic control of robots for rehabilitation," *International Journal of Robotics Research*, vol. 29, no. 1, pp. 23–38, January 2010.

[112] H. Vallery, J. Veneman, E. van Asseldonk, R. Ekkelenkamp, M. Buss, and H. van Der Kooij, "Compliant actuation of rehabilitation robots," *IEEE Robotics and Automation Magazine*, vol. 15, no. 3, pp. 60–69, 2008.

[113] D. W. Robinson, "Design and analysis of series elasticity in closed-loop actuator force control," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[114] K. Kong, J. Bae, and M. Tomizuka, "Control of rotary series elastic actuator for ideal force-mode actuation in human-robot interaction applications," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 1, pp. 105–118, February 2009.

[115] H. J. Landau, "Sampling, data transmission, and the nyquist rate," *Proceedings of the IEEE*, vol. 55, no. 10, pp. 1701–1706, 1967.

[116] D. Bennett, J. Hollerbach, Y. Xu, and I. Hunter, "Time-varying stiffness of human elbow joint during cyclic voluntary movement," *Experimental Brain Research*, vol. 88, pp. 433–442, 1992.

[117] R. Merletti, A. Botter, A. Troiano, E. Merlo, and M. A. Minetto, "Technology and instrumentation for detection and conditioning of the surface electromyographic signal: State of the art," *Clinical Biomechanics*, vol. 24, no. 2, pp. 122–134, February 2009.

[118] E. N. Kamavuako, E. J. Scheme, and K. B. Englehart, "Combined surface and intramuscular EMG for improved real-time myoelectric control performance," *Biomedical Signal Processing and Control*, vol. 10, no. 1, pp. 102–107, 2014.

[119] K. Kiguchi and Y. Hayashi, "Motion estimation based on EMG and EEG signals to control wearable robots," in *IEEE International Conference on Systems, Man, and Cybernetics*, October 2013, pp. 4213–4218.

[120] K. K. Ang and C. Guan, "EEG-based strategies to detect motor imagery for control and rehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 4, pp. 392–401, 2017.

[121] T. S. Buchanan, D. G. Lloyd, K. Manal, and T. F. Besier, "Neuromusculoskeletal modeling: Estimation of muscle forces and joint moments and movements from measurements of neural command," *Journal of Applied Biomechanics*, vol. 20, no. 4, pp. 367–395, 2004.

[122] T. Desplenter and A. L. Trejos, "Evaluating muscle activation models for elbow motion estimation," *Sensors*, vol. 18, no. 4, 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/4/1004

[123] A. Kyrylova, T. Desplenter, A. Escoto, S. Chinchalkar, and A. L. Trejos, "Simplified emg-driven model for active-assisted therapy," in *IEEE International Conference on Intelligent Robots and Systems, Workshop on Rehabilitation and Assistive Robotics*, Chicago, U.S.A., September 2014.

[124] T. Desplenter, A. Kyrylova, and A. L. Trejos, "Development of an emg-driven control system for a wearable mechatronic elbow brace," in *IEEE International Conference on Biomedical and Health Informatics*, Las Vegas, USA, February 2016, pp. 501–504.

[125] T. Desplenter, J. Lobo-Prat, A. H. A. Stienen, and A. L. Trejos, "Extension of the wearme framework for emg-driven control of a wearable arm support," in *IEEE International Conference on Advanced Intelligent Mechatronics*, Banff, Canada, July 2016, pp. 288–293.

[126] T. Kashima, Y. Isurugi, and M. Shima, "Analysis of a muscular control system in human movements." *Biological Cybernetics*, vol. 82, pp. 123–131, 2000.

[127] E. A. Clancy, L. Liu, P. Liu, and D. V. Z. Moyer, "Identification of constant-posture EMG-torque relationship about the elbow using nonlinear dynamic models," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 1, pp. 205–212, 2012.

[128] J. Bai, Y. Gao, S. Wang, and J. Zhao, "An elbow biomechanical model and its coefficients adjustment," in *IEEE International Conference on Software Engineering and Service Science*, Beijing, China, June 2014, pp. 954–957.

[129] Y. Gao, J. Bai, S. Wang, and J. Zhao, "An elbow-biomechanical modeling based on sEMG," in *World Congress on Intelligent Control and Automation*, Shenyang, China, June 2014, pp. 5238–5243.

[130] T. Kawase, T. Sakurada, Y. Koike, and K. Kansaku, "A hybrid BMI-based exoskeleton for paresis: EMG control for assisting arm movements," *Journal of Neural Engineering*, vol. 14, no. 1, pp. 1–12, 2017.

[131] R. Song and K. Tong, "Using recurrent artificial neural network model to estimate voluntary elbow torque in dynamic situations," *Medical and Biological Engineering and Computing*, vol. 43, no. 4, pp. 473–480, 2005.

[132] D. Novak and R. Riener, "A survey of sensor fusion methods in wearable robotics," *Robotics and Autonomous Systems*, vol. 73, pp. 155 – 170, 2015, wearable Robotics.

[133] M. Hakonen, H. Piitulainen, and A. Visala, "Current state of digital signal processing in myoelectric interfaces and related applications," *Biomedical Signal Processing and Control*, vol. 18, pp. 334–359, 2015.

[134] C. J. De Luca, "Surface electromyography: Detection and recording," DelSys Incorporated, Tech. Rep., 2002.

[135] E. E. Cavallaro, J. Rosen, J. C. Perry, and S. Burns, "Real-time myoprocessors for a neural controlled powered exoskeleton arm." *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 11, pp. 2387–96, November 2006.

[136] K. Manal and T. S. Buchanan, "A one-parameter neural activation to muscle activation model: Estimating isometric joint moments from electromyograms," *Journal of Biomechanics*, vol. 36, no. 8, pp. 1197–1202, 2003.

[137] M. Ison and P. Artemiadis, "Proportional myoelectric control of robots: Muscle synergy development drives performance enhancement, retainment, and generalization," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 259–268, 2015.

[138] M. Ison, I. Vujaklija, B. Whitsell, D. Farina, and P. Artemiadis, "High-density electromyography and motor skill learning for robust long-term control of a 7-dof robot arm," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 4, pp. 424–433, 2016.

[139] P. N. Kooren, J. Lobo-prat, A. Q. L. Keemink, M. M. Janssen, A. H. A. Stienen, I. J. M. D. Groot, M. I. Paalman, R. Verdaasdonk, and B. F. J. M. Koopman, "Design and control of the Active A-Gear: a wearable 5 DOF arm exoskeleton for adults with duchenne muscular dystrophy," in *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, UTown, Singapore, June 2016, pp. 1242–1247.

[140] A. L. Cruz Ruiz, C. Pontonnier, and G. Dumont, "Low-dimensional motor control representations in throwing motions," *Applied Bionics and Biomechanics*, vol. 2017, pp. 1–19, 2017.

[141] S. Mazzoleni, L. Buono, P. Dario, and F. Posteraro, "Upper limb robot-assisted therapy in subacute and chronic stroke patients: Preliminary results on initial exposure based on kinematic measures," in *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, São Paulo, Brazil, August 2014, pp. 265–269.

[142] L. Q. Zhang and W. Z. Rymer, "Simultaneous and nonlinear identification of mechanical and reflex properties of human elbow joint muscles," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 12, pp. 1192–1207, 1997.

[143] B. K. Dinh, M. Xiloyannis, C. W. Antuvan, L. Cappello, and L. Masia, "Hierarchical cascade controller for assistance modulation in a soft wearable arm exoskeleton," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1786–1793, 2017.

[144] L. Peng, Z. G. Hou, and W. Wang, "A dynamic EMG-torque model of elbow based on neural networks," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Milan, Italy, August 2015, pp. 2852–2855.

[145] L. Li, K. Y. Tong, X. L. Hu, L. K. Hung, and T. K. K. Koo, "Incorporating ultrasound-measured musculotendon parameters to subject-specific EMG-driven model to simulate voluntary elbow flexion for persons after stroke," *Clinical Biomechanics*, vol. 24, no. 1, pp. 101–109, 2009.

[146] A. Fratini, P. Bifulco, M. Romano, F. Clemente, and M. Cesarelli, "Simulation of surface EMG for the analysis of muscle activity during whole body vibratory stimulation," *Computer Methods and Programs in Biomedicine*, vol. 113, no. 1, pp. 314–322, 2014.

[147] C. Cifuentes, A. Braidot, L. Rodriguez, M. Frisoli, A. Santiago, and A. Frizera, "Development of a wearable ZigBee sensor system for upper limb rehabilitation robotics," in *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, Roma, Italy, June 2012, pp. 1989–1994.

[148] R. Tao, S. Xie, Y. Zhang, and J. W. Pau, "Review of EMG-based neuromuscular interfaces for rehabilitation: elbow joint as an example," *International Journal of Biomechatronics and Biomedical Robotics*, vol. 2, no. 2/3/4, pp. 184–194, 2013.

[149] N. Hogan and R. W. Mann, "Myoelectric signal processing: Optimal estimation applied to electromyographypart i: Derivation of the optimal myoprocessor," *IEEE Transactions on Biomedical Engineering*, vol. BME-27, no. 7, pp. 382–395, 1980.

[150] J. W. L. Pau, S. S. Q. Xie, and A. J. Pullan, "Neuromuscular interfacing: Establishing an EMG-driven model for the human elbow joint," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 9, pp. 2586–2593, 2012.

[151] N. Nazmi, M. A. A. Rahman, S. A. Mazlan, H. Zamzuri, and M. Mizukawa, "Electromyography (EMG) based signal analysis for physiological device application in lower limb rehabilitation," in *International Conference on Biomedical Engineering*, Penang, Malaysia, March 2015.

[152] J. L. A. S. Ramos and M. A. Meggiolaro, "Use of surface electromyography for human amplification using an exoskeleton driven by artificial pneumatic muscles," in *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, São Paulo, Brazil, August 2014, pp. 585–590.

[153] T. S. Buchanan, D. P. Almdale, J. L. Lewis, and W. Z. Rymer, "Characteristics of synergic relations during isometric contractions of human elbow muscles." *Journal of Neurophysiology*, vol. 56, no. 5, pp. 1225–1241, 1986.

[154] D. Blana, J. G. Hincapie, E. K. Chadwick, and R. F. Kirsch, "A musculoskeletal model of the upper extremity for use in the development of neuroprosthetic systems," *Journal of Biomechanics*, vol. 41, no. 8, pp. 1714–1721, 2008.

[155] J. S. Dufour, W. S. Marras, and G. G. Knapik, "An EMG-assisted model calibration technique that does not require MVCs," *Journal of Electromyography and Kinesiology*, vol. 23, no. 3, pp. 608–613, 2013.

[156] W. S. Kim, H. D. Lee, D. H. Lim, J. S. Han, K. S. Shin, and C. S. Han, "Development of a muscle circumference sensor to estimate torque of the human elbow joint," *Sensors and Actuators, A: Physical*, vol. 208, pp. 95–103, 2014.

[157] J. Rosen, M. B. Fuchs, and M. Arcan, "Performances of hill-type and neural network muscle models-toward a myosignal-based exoskeleton." *Computers and Biomedical Research*, vol. 32, no. 5, pp. 415–39, October 1999.

[158] D. G. Thelen, A. B. Schultz, S. D. Fassois, and J. A. Ashton-Miller, "Identification of dynamic myoelectric signal-to-force models during isometric lumbar muscle contractions," *Journal of Biomechanics*, vol. 27, no. 7, pp. 907–919, 1994.

[159] A. A. Nikooyan, H. E. J. Veeger, E. K. J. Chadwick, M. Praagman, and F. C. T. van der Helm, "Development of a comprehensive musculoskeletal model of the shoulder and elbow." *Medical & Biological Engineering & Computing*, vol. 49, no. 12, pp. 1425–35, 2011.

[160] D. A. Funk, K. N. An, B. F. Morrey, and J. R. Daube, "Electromyographic analysis of muscles across the elbow joint," *Journal of Orthopaedic Research*, vol. 5, pp. 529–538, 1987.

[161] K. Uno, T. Oku, P. Phatiwuttipat, K. Koba, Y. Yamashita, K. Murakami, M. Uemura, H. Hirai, and F. Miyazaki, "A novel muscle synergy extraction method to explain the equilibrium-point trajectory and endpoint stiffness during human upper-limb movements on a horizontal plane," in *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, São Paulo, Brazil, August 2014, pp. 621–626.

[162] M. M. Mano and M. D. Ciletti, *Digital Design: With an Introduction to the Verilog HDL*, 5th ed. New Jersey, USA: Pearson Education Inc., 2013.

[163] C. J. Nycz, M. A. Delph, and G. S. Fischer, "Modeling and design of a tendon actuated soft robotic exoskeleton for hemiparetic upper limb rehabilitation," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Milan, Italy, August 2015, pp. 3889–3892.

[164] E. Rocon, A. Ruiz, J. Pons, J. Belda-Lois, and J. Sánchez-Lacuesta, "Rehabilitation robotics: a wearable exo-skeleton for tremor assessment and suppression," in *International Conference on Robotics and Automation*, Barcelona, Spain, April 2005, pp. 2271–2276.

[165] K. Pothuganti and A. Chitneni, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," *Advance in Electronic and Electric Engineering*, vol. 4, no. 6, pp. 655–662, 2014.

[166] S. Baiqing and L. Yanjun, "Dynamics modeling of human elbow joints servicing for rehabilitation robots," in *IEEE Conference on Industrial Electronics and Applications*, Xi'an, China, May 2009, pp. 2566–2569.

[167] H. J. Nagarsheth, P. V. Savsani, and M. a. Patel, "Modeling and dynamics of human arm," in *IEEE International Conference on Automation Science and Engineering*, Washington DC. USA, August 2008, pp. 924–928.

[168] F. Clark, D. Nguyen, G. G. Filho, and M. Huber, "Identification of static and dynamic muscle activation patterns for intuitive human/computer interfaces," in *International Conference on Pervasive Technologies Related to Assistive Environments*, Samos. Greece, June 2010.

[169] E. Chadwick, D. Blana, R. F. Kirsch, and A. J. van den Bogert, "Real-time simulation of three-dimensional shoulder girdle and arm dynamics," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 7, pp. 1947–1956, 2014.

[170] C. B. Moody, A. A. Barhorst, and L. Schovanec, "A neuro-muscular elasto-dynamic model of the human arm part 2: Musculotendon dynamics and related stress effects," *Journal of Bionic Engineering*, vol. 6, no. 2, pp. 108–119, 2009.

[171] G. Venture, K. Yamane, and Y. Nakamura, "Identification of human musculo-tendon subject specific dynamics using musculo-skeletal computations and non linear least square," in *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, Pisa, Italy, February 2006, pp. 211–216.

[172] Q. C. Ding, A. B. Xiong, X. G. Zhao, and J. D. Han, "A novel EMG-driven state space model for the estimation of continuous joint movements," in *IEEE International Conference on Systems, Man and Cybernetics*, Anchorage, U.S.A., October 2011, pp. 2891–2897.

[173] P. Liu, L. Liu, and E. A. Clancy, "Influence of joint angle on EMG-torque model during constant-posture, torque-varying contractions," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 6, pp. 1039–1046, 2015.

[174] E. K. Chadwick, D. Blana, A. J. van den Bogert, and R. F. Kirsch, "A real-time, 3-D musculoskeletal model for dynamic simulation of arm movements," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 4, pp. 941–948, 2009.

[175] S. H. Kang and L. Q. Zhang, "Robust identification of multi-joint human arm impedance based on dynamics decomposition: A modeling study," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, Massachusetts, USA, August-September 2011, pp. 4453–4456.

[176] D. Lakatos, F. Petit, and P. V. D. Smagt, "Conditioning vs. excitation time for estimating impedance parameters of the human arm," in *IEEE-RAS International Conference on Humanoid Robots*, Bled, Slovenia, October 2011, pp. 636–642.

[177] G. Venture, K. Yamane, and Y. Nakamura, "In-vivo estimation of the human elbow joint dynamics during passive movements using musculo-skeletal model computations," in *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, Pisa, Italy, February 2006, pp. 205–210.

[178] M. J. Fu and M. C. Cavusoglu, "Human-arm-and-hand-dynamic model with variability analyses for a stylus-based haptic interface," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 6, pp. 1633–1644, 2012.

[179] H. A. Abdullah, C. Tarry, G. S. Mittal, and M. Abderrahim, "A biomechanical model to aid robot-assisted therapy of upper limb impairment," in *Annual Conference on IEEE Industrial Electronics*, Paris, France, November 2006, pp. 4107–4112.

[180] H. Bayati, S. Vahdat, and B. V. Vahdat, "Investigating the properties of optimal sensory and motor synergies in a nonlinear model of arm dynamics," in *International Joint Conference on Neural Networks*, Atlanta, Georgia, USA, June 2009, pp. 2524–2531.

[181] P. T. Katsiaris, P. K. Artemiadis, and K. J. Kyriakopoulos, "Modeling anthropomorphism in dynamic human arm movements," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 2010, pp. 3507–3512.

[182] S. Konyk Jr., "Dynamic trajectory evolution incorporating modeling of arm movement neural data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, July 1992, pp. 542–547.

[183] Y. Wang, R. Ikeura, H. Sawai, and S. Hayakawa, "Analysis of human arm characteristics in cooperative motion based on musculoskeletal model," in *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, Tokyo, Japan, September 2010, pp. 301–306.

[184] C. W. Heckathorne and D. S. Childress, "Relationships of the surface electromyogram to the force, length, velocity, and contraction rate of the cineplastic human biceps," *American Journal of Physical Medicine*, vol. 60, no. 1, pp. 1–19, 1981.

[185] J. Woods and B. Bigland-Ritchie, "Linear and non-linear surface emg/force relationships in human muscles," *American Journal of Physical Medicine*, vol. 62, no. 6, pp. 287–299, 1983.

[186] H. S. Milner-Brown, R. B. Stein, and R. Yemm, "Changes in firing rate of human motor units during linearly changing voluntary contractions," *The Journal of Physiology*, vol. 230, pp. 371–390, 1973.

[187] F. Zajac, "Muscle and tendon: Properties, models, scaling and application to biomechanics and motor control," *Critical Reviews in Biomedical Engineering*, vol. 11, no. 4, pp. 359–405, 1989.

[188] M. Millard, T. Uchida, A. Seth, and S. L. Delp, "Flexing computational muscle: Modeling and simulation of musculotendon dynamics," *Journal of Biomechanical Engineering*, vol. 135, no. 2, 2013.

[189] Y. cheng Fung, *Biomechanics - Mechanical Properties of Living Tissues.* New York, New York: Springer-Verlag, 1981.

[190] F. Vahid and T. Givargis, *Introduction.* John Wiley & Sons, Inc., 2002, pp. 1–28.

[191] K. Anam and A. Al-Jumaily, "Active exoskeleton control systems: State of the art," *Procedia Engineering*, vol. 41, pp. 988–994, 2012.

[192] T. Proietti, V. Crocher, A. Roby-Brami, and N. Jarrasse, "Upper-limb robotic exoskeletons for neurorehabilitation: A review on control strategies," *IEEE Reviews in Biomedical Engineering*, vol. 9, pp. 4–14, 2016.

[193] M. H. Rahman, M. J. Rahman, O. L. Cristobal, M. Saad, J. P. Kenné, and P. S. Archambault, "Development of a whole arm wearable robotic exoskeleton for rehabilitation and to assist upper limb movements," *Robotica*, vol. 33, pp. 19–39, 2014.

[194] E. T. Wolbrecht, J. Leavitt, D. J. Reinkensmeyer, and J. E. Bobrow, "Control of a pneumatic orthosis for upper extremity stroke rehabilitation." in *IEEE Engineering in Medicine and Biology Society Annual International Conference*, New York City, USA, August-September 2006, pp. 2687–93.

[195] K. Kiguchi, M. Rahman, and M. Sasaki, "Neuro-fuzzy based motion control of a robotic exoskeleton: Considering end-effector force vectors," in *IEEE International Conference on Robotics and Automation*, Orlando, USA, May 2006, pp. 3146–3151.

[196] K. Kiguchi and Y. Hayashi, "An EMG-based control for an upper-limb power-assist exoskeleton robot," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 4, pp. 1064–1071, August 2012.

[197] R. Richardson, A. Jackson, P. Culmer, and B. Bhakta, "Pneumatic impedance control of a 3-d . o . f ." *Advanced Robotics*, vol. 20, no. 12, pp. 1321–1339, 2006.

[198] P. R. Culmer, A. E. Jackson, S. Makower, R. Richardson, J. A. Cozens, M. C. Levesley, and B. B. Bhakta, "A control strategy for upper limb robotic rehabilitation with a dual robot system," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 4, pp. 575–585, August 2010.

[199] N. Hogan, "Impedance control: An approach to manipulation: Part ii – implementation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 8–16, 1985.

[200] C. R. Carignan, M. P. Naylor, and S. N. Roderick, "Controlling shoulder impedance in a rehabilitation arm exoskeleton," in *IEEE International Conference on Robotics and Automation*. Pasadena, USA: IEEE, May 2008, pp. 2453–2458.

[201] Y. Yang, L. Wang, J. Tong, and L. Zhang, "Arm rehabilitation robot impedance control and experimentation," in *IEEE International Conference on Robotics and Biomimetics*, Kunming, China, Decemeber 2006, pp. 914–918.

[202] Y. Morita, M. Nagasaki, H. Ukai, N. Matsui, and M. Uchida, "Development of rehabilitation training support system of upper limb motor function for personalized rehabilitation," in *IEEE International Conference on Robotics and Biomimetics*. Bangkok, Thailand: IEEE, February 2009, pp. 300–305.

[203] D. Andreasen, S. Allen, and D. Backus, "Exoskeleton with EMG based active assistance for rehabilitation," in *International Conference on Rehabilitation Robotics*. Chicago, IL, USA: IEEE, June-July 2005, pp. 333–336.

[204] C. Carignan, J. Tang, S. Roderick, and M. Naylor, "A configuration-space approach to controlling a rehabilitation arm exoskeleton," in *Physical Review Letters*, Noordwijk, The Netherlands, 2007, pp. 179–187.

[205] M. Mihelj, T. Nef, and R. Riener, "A novel paradigm for patient-cooperative control of upper-limb rehabilitation robots," *Advanced Robotics*, vol. 21, no. 8, pp. 843–867, 2007.

[206] J. Oblak, I. Cikajlo, and Z. Matjacic, "Universal haptic drive: A robot for arm and wrist rehabilitation," *IEEE Transactions on Neural and Rehabilitation Engineering*, vol. 18, no. 3, pp. 293–302, 2010.

[207] A. Frisoli, L. Borelli, A. Montagner, S. Marcheschi, C. Procopio, F. Salsedo, M. Bergamasco, M. C. Carboncini, M. Tolaini, and B. Rossi, "Arm rehabilitation with a robotic exoskeleleton in virtual reality," in *International Conference on Rehabilitation Robotics*, Noordwijk, The Netherlands, June 2007, pp. 631–642.

[208] W. Yu, J. Rosen, and X. Li, "PID admittance control for an upper limb exoskeleton," in *American Control Conference*. San Francisco, USA: IEEE, June-July 2011, pp. 1124–1129.

[209] J. Garrido, W. Yu, and X. Li, "Modular design and control of an upper limb exoskeleton," *Journal of Mechanical Science and Technology*, vol. 30, no. 5, pp. 2265–2271, 2016.

[210] J. Huang, W. Huo, W. Xu, S. Mohammed, and Y. Amirat, "Control of upper-limb power-assist exoskeleton using a human-robot interface based on motion intention recognition," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1257–1270, 2015.

[211] L. M. Miller and J. Rosen, "Comparison of multi-sensor admittance control in joint space and task space for a seven degree of freedom upper limb exoskeleton," in *IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*. Tokyo: IEEE, September 2010, pp. 70–75.

[212] C. Carignan, J. Tang, and S. Roderick, "Development of an exoskeleton haptic interface for virtual task training," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. St. Louis, USA: IEEE, October 2009, pp. 3697–3702.

[213] A. Cherubini, R. Passama, P. Fraisse, and A. Crosnier, "A unified multimodal control framework for human-robot interaction," *Robotics and Autonomous Systems*, vol. 70, pp. 106–115, April 2015.

[214] M. Visinskyi, J. Cavallaro, and I. Walker, "Expert system framework for fault detection and fault toleracnce in robotics," *Computers and Electrical Engineering*, vol. 20, no. 8, pp. 421–435, 1994.

[215] S. Stramigioli, E. D. Fasse, and J. Willems, "A rigorous framework for interactive robot control," *International Journal of Control*, vol. 75, no. 18, pp. 1486–1503, 2002.

[216] B. Woolley and G. Peterson, "Unified behavior framework for reactive robot control," *Journal of Intelligent and Robotic Systems*, vol. 55, pp. 155–176, 2009.

[217] M. Oubbati and G. Palm, "A neural framework for adaptive robot control," *Neural Computing and Applications*, vol. 19, pp. 103–114, 2010.

[218] P. Przystałka and M. Adamczyk, *EmAmigo framework for developing behavior-based control systems of inspection robots*. Springer Berlin Heidelberg, 2007.

[219] M. Gianni, G.-J. M. Kruijff, and F. Pirri, "A stimulus-response framework for robot control," *ACM Transactions on Iteractive Intelligent Systems*, vol. 4, no. 4, January 2015.

[220] R. Seiger, C. Seidl, U. Aβmann, and T. Schlegel, "A capability-based framework for programming small domestic service robots," in *MORSE/VAO Workshop on Model-Driven Robot Software Engineering and View-based Software-Engineering*, L'Aquila, Italy, July 2015, pp. 49–54.

[221] B. Woolley and G. Peterson, "Genetic evolution of hierarchical behavior structures," in *Genetic and Evolutionary Computation Conference*, London, England, United Kingdom, July 2007, pp. 1731–1738.

[222] T. Collett, B. MacDonald, and B. Gerkey, "Player 2.0: Toward a practical robot programming framework," in *Australian Conference on Robotics and Automation*, Sydney, Australia, December 2005.

[223] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source robot operating system," in *IEEE International Conference on Robotics and Automation Workshop on Open Source Software*, Kobe, Japan, May 2009.

[224] T. Petrič, A. Gams, J. Babič, and L. Žlajpah, "Reflexive stability control framework for humanoid robots," *Autonomous Robots*, vol. 34, no. 4, pp. 347–361, 2013.

[225] P. Chen and Q. Cao, "A middleware-based simulation and control framework for mobile service robots," *Journal of Intelligent and Robotic Systems*, vol. 79, pp. 489–504, 2014.

[226] P. Artemiadis and K. Kyriakopoulos, "Assessment of muscle fatigue using a probabilistic framework for an emg-based robot control scenario," in *IEEE International Conference on BioInformatics and BioEngineering*, Athens, Greece, October 2008.

[227] B. Urgulu, M. Nishimura, K. Hyodo, M. Kawanishi, and T. Narikiyo, "A framework for sensorless torque estimationand control in wearable exoskeletons," in *IEEE International Workshop on Advanced Motion Control*, Sarajevo, Bosnia and Herzegovina, March 2012.

[228] J. Peters, M. Mistry, F. Udwadia, J. Nakanishi, and S. Schaal, "A unifying framework for robot control with redundant dofs," *Autonomous Robots*, vol. 24, pp. 1–12, 2008.

[229] H. Richter, "A framework for control of robots with energy regeneration," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 9, 2015.

[230] A. Das, R. Fierro, V. Kumar, J. Ostrowski, and J. Spletzer, "A vision-based formation control framework," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 813–825, 2002.

[231] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A unifying passivity-based control framework for position, torque and impedance control of flexible joint robots," *The International Journal of Robotics Research*, vol. 26, no. 1, pp. 23–29, 2007.

[232] A. Simões, E. Colombini, J. Matsuura, and M. Franchin, "TORP: The open robot project a framework for module-based robots," *Journal of Intelligent and Robotic Systems*, vol. 66, pp. 3–22, 2012.

[233] Ö. A. Tekin, R. Babšuka, T. Tomiyama, and B. D. Schutter, "Toward a flexible control design framework to automatically generate control code for mechatronic systems," in *American Control Conference*, St. Louis, U.S.A., June 2009.

[234] Y. Zou, G. Zhao, and T. Wang, "A general framework of mechatronic modular architecture," *Advances in Mechanical Engineering*, 2013.

[235] C. Fleischer, A. Wege, K. Kondak, and G. Hommel, "Application of EMG signals for controlling exoskeleton robots." *Biomedical Engineering*, vol. 51, no. 5-6, pp. 314–9, December 2006.

[236] B. Bruegge and A. H. Dutoit, *Object-Oriented Software Engineering Using UML, Patterns, and Java*. New Jersey, USA: Pearson Education Inc., 2010.

[237] A. Basteris, S. M. Nijenhuis, A. H. Stienen, J. H. Buurke, G. B. Prange, and F. Amirabdollahian, "Training modalities in robot-mediated upper limb rehabilitation in stroke: a framework for classification based on a systematic review," *Journal of NeuroEngineering and Rehabilitation*, vol. 11, no. 1, p. 111, 2014.

[238] P. T. Straathof, J. Lobo-prat, F. Schilder, P. N. Kooren, M. I. Aalman, A. H. A. Stienen, I. J. M. D. Groot, and B. F. J. M. Koopman, "Design and Control of the A-Arm: an Active Planar Arm Support for Adults with Duchenne Muscular Dystrophy*," in *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, UTown, Singapore, June 2016, pp. 1242–1247.

[239] Y. Hasegawa and S. Oura, "Exoskeletal meal assistance system (EMAS II) for progressive muscle dystrophy patient," in *IEEE International Conference on Rehabilitation Robotics*, June–July 2011, pp. 1–6.

[240] M. Asghari Oskoei and H. Hu, "Myoelectric control systems-A survey," *Biomedical Signal Processing and Control*, vol. 2, no. 4, pp. 275–294, 2007.

[241] M. Ison and P. Artemiadis, "The role of muscle synergies in myoelectric control: trends and challenges for simultaneous multifunction control," *Journal of Neural Engineering*, vol. 11, no. 5, p. 051001, October 2014.

[242] R. A. R. C. Gopura, D. S. V. Bandara, J. M. P. Gunasekara, and T. S., "Recent trends in EMG-based control methods for assistive robots," in *Electrodiagnosis in New Frontiers of Clinical Research*. IntechOpen, 2013, ch. 12, pp. 237–267.

[243] R. Norman and P. Komi, "Electromechanical delay in skeletal muscle under normal movement conditions," *Acta Physiologica Scandinavica*, vol. 106, pp. 241–248, 1979.

[244] J. M. Winters and L. Stark, "Analysis of fundamental human movement patterns through the use of in-depth antagonistic muscle models," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 10, pp. 826–839, 1985.

[245] S. Stroeve, "Impedance characteristics of a neuromusculoskeletal model of the human arm I. Posture control." *Biological Cybernetics*, vol. 81, no. 5-6, pp. 475–494, November 1999.

[246] K. Manal, R. V. Gonzalez, D. G. Lloyd, and T. S. Buchanan, "A real-time EMG-driven virtual arm," *Computers in Biology and Medicine*, vol. 32, no. 1, pp. 25–36, 2002.

[247] C. Rengifo, Y. Aoustin, F. Plestan, and C. Chevallereau, "Distribution of forces between synergistics and antagonistics muscles using an optimization criterion depending on muscle contraction behavior." *ASME Journal of Biomechanical Engineering*, vol. 132, no. 4, p. 11, 2010.

[248] D. G. Thelen, "Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults." *Journal of Biomechanical Engineering*, vol. 125, no. 1, pp. 70–77, 2003.

[249] D. G. Lloyd and T. F. Besier, "An EMG-driven musculoskeletal model to estimate muscle forces and knee joint moments in vivo," *Journal of Biomechanics*, vol. 36, no. 6, pp. 765–776, 2003.

[250] National Center for Simulation in Rehabilitation Research, "OpenSim community," 2010. [Online]. Available: http://opensim.stanford.edu/

[251] D. A. Winter, *Biomechanics and Motor Control of Human Movement*, 2nd ed. Wiley, 1990.

[252] S. Salisbury, "Lecture notes on actuator principles, integration and control," London, Ontario, Canada, September-December 2013.

[253] S. W. O'Driscoll and N. J. Giori, "Continuous passive motion (CPM): theory and principles of clinical application." *Journal of Rehabilitation Research and Development*, vol. 37, no. 2, pp. 179–88, 2000.

[254] S. W. O'Driscoll, *The Treatment of Distal Humerus Fractures.* Springer-Verlag Italia, 2008, ch. 5, pp. 61–69.

[255] D. Lynch, M. Ferraro, J. Krol, C. M. Trudell, P. Christos, and B. T. Volpe, "Continuous passive motion improves shoulder joint integrity following stroke," *Clinical Rehabilitation*, vol. 19, no. 6, pp. 594–599, 2005.

[256] B. Sheng, Y. Zhang, W. Meng, C. Deng, and S. Xie, "Bilateral robots for upper-limb stroke rehabilitation: State of the art and future prospects," *Medical Engineering and Physics*, vol. 38, no. 7, pp. 587–606, 2016.

[257] L. Tan, *Digital Signal Processing: Fundamental and Applications.* Burlington, Massachusetts: Elsevier, 2008, ch. 12, pp. 557–615.

[258] S. V. Vaseghi, *Advanded Digital Signal Processing and Noise Reduction.* Chinchester, West Sussex, United Kingdom: John Wiley & Sons, 2008, ch. 11, pp. 295–320.

[259] C. Rockwood, F. Matsen, M. Wirth, S. Lippitt, E. Fehringer, and J. Sperling, *Rockwood and Matsen's The Shoulder*, 5th ed. New Jersey, USA: Elsevier, 2016.

[260] D. Neumann, *Kinesiology of the Musculoskeletal System: Foundations for Rehabilitation*, 2nd ed. Maryland Heights, MO: Mosby, 2009.

[261] V. Phadke, P. R. Camargo, and P. M. Ludewig, "Scapular and rotator cuff muscle activity during arm elevation: A review of normal function and alterations with shoulder impingement," *Revista Brasileira de Fisioterapia*, vol. 13, no. 1, pp. 1–9, 2009.

[262] R. Nieuwenhuys, J. Voogd, and C. van Huijzen, *Orientation*, 4th ed. Steinkopff-Verlag Darmstadt, 2008, ch. 1, pp. 3–6.

[263] R. Birch, *The Peripheral Nervous System: Gross Anatomy*, 2nd ed. Springer-Verlag London, 2011, ch. 1, pp. 1–41.

[264] A. Farb, R. B. Devereux, and P. Kligfield, "Day-to-day variability of voltage measurements used in electrocardiographic criteria for left ventricular hypertrophy," *Journal of the American College of Cardiology*, vol. 15, no. 3, pp. 618–623, March 1990.

[265] M. Teplan, "Fundamentals of EEG measurement," *Measurement Science Review*, vol. 2, no. 2, pp. 1–11, 2002.

[266] D. Bandara, J. Arata, and K. Kiguchi, "A noninvasive braincomputer interface approach for predicting motion intention of activities of daily living tasks for an upper-limb wearable robot," *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, p. 172988141876731, 2018.

[267] S. Vaid, P. Singh, and C. Kaur, "EEG signal analysis for BCI interface: A review," *International Conference on Advanced Computing and Communication Technologies*, pp. 143–147, February 2015.

[268] M. Mode, E. Journal, and O. F. Scientific, "EMG signal classification for human computer interaction a review EMG signal classification for human computer interaction," *European Journal of Scientific Research*, vol. 3, no. 2, pp. 480–501, 2009.

[269] N. Hogan, "Skeletal muscle impedance in the control of motor actions," *Journal of Mechanics in Medicine and Biology*, vol. 2, no. 3 & 4, pp. 359–373, September 2002.

[270] J. A. Doeringer and N. Hogan, "Intermittency in preplanned elbow movements persists in the absence of visual feedback," *Journal of Neurophysiology*, vol. 80, no. 4, pp. 1787–1799, 1998.

[271] J. M. Winters, "An improved muscle-reflex actuator for use in large-scale neuromusculoskeletal models," *Annals of Biomedical Engineering*, vol. 23, no. 4, pp. 359–374, 1995.

[272] R. J. Downey, M. Merad, E. J. Gonzalez, and W. E. Dixon, "The time-varying nature of electromechanical delay and muscle control effectiveness in response to stimulation-induced fatigue." *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 9, pp. 1397–1408, 2017.

[273] J. Rosen, J. C. Perry, N. Manning, S. Burns, and B. Hannaford, "The human arm kinematics and dynamics during daily activities-toward a 7 dof upper limb powered exoskeleton," in *International Conference on Advanced Robotics.* IEEE, July 2005, pp. 532–539.

[274] J. J. Crisco, W. M. Heard, R. R. Rich, D. J. Paller, and S. W. Wolfe, "The mechanical axes of the wrist are oriented obliquely to the anatomical axes," *The Journal of Bone and Joint Surgery American*, vol. 93, no. 2, pp. 169–177, 2011.

[275] W. M. Murray, S. L. Delp, and T. S. Buchanan, "Variation of muscle moment arms with elbow and forearm position," *Journal of Biomechanics*, vol. 28, no. 5, 1995.

[276] M. D. K. Breteler, C. W. Spoor, and F. C. T. Van der Helm, "Measuring muscle and joint geometry parameters of a shoulder for modeling purposes," *Journal of Biomechanics*, vol. 32, no. 11, pp. 1191–1197, 1999.

[277] L. Zhou, Y. Li, and S. Bai, "A human-centered design optimization approach for robotic exoskeletons through biomechanical simulation," *Robotics and Autonomous Systems*, vol. 91, pp. 337–347, 2017.

[278] J. M. Winters, "Hill-based muscle models: a systems engineering perspective," in *Multiple Muscle Systems: Biomechanics and Movement Organization.* Springer-Verlag, 1990, ch. 5, pp. 69–93.

[279] A. Shklar and Z. Dvir, "Isokinetic strength relationships in shoulder muscles," *Clinical Biomechanics*, vol. 10, no. 7, pp. 369–373, 1995.

[280] S. N. Imrhan and G. D. Jenkins, "Flexion — extension hand torque strengths: applications in maintenance tasks," *International Journal of Industrial Ergonomics*, vol. 23, no. 4, pp. 359–371, 1999.

[281] M. L. Latash and M. Zatsiorsky, "Joint stiffness: Myth or reality?" *Human Movement Science*, vol. 12, pp. 653–692, 1993.

[282] T. Flash, "The control of hand equilibrium trajectories in multi-joint arm movements," *Biological Cybernetics*, vol. 57, no. 4-5, pp. 257–274, 1987.

[283] J. Zawadzki and A. Siemieński, "Maximal frequency, amplitude, kinetic energy and elbow joint stiffness in cyclic movements," *Acta of Bioengineering and Biomechanics*, vol. 12, no. 2, pp. 55–64, 2010.

[284] J. A. Johnson, "BME 9505 class lecture on structure and function of bone," January 2014, wSC 240, The University of Western Ontario, London, Ontario, Canada.

[285] M. Benjamin and J. R. Ralphs, "Tendons and ligaments - An overview," *Histology and Histopathology*, vol. 12, no. 4, pp. 1135–1144, 1997.

[286] J. A. Johnson, "BME 9505 class lecture on structure and function of soft tissues," February 2014, wSC 240, The University of Western Ontario, London, Ontario, Canada.

Appendix A

Experiment Map

Aside from the different motions and individual factors, there were experimental and control output factors that were assessed within this thesis. The results of the experiment showed statistically significant differences between experimental factors, such as the visual feedback, training, and constrained motion, and between the control system parameters, such as motion models and control output. In order to clarify these differences, Table A.1 provides a comparison of these factors across the experiments. Each of the experiments also conducted testing that fell within different phases of the human–machine integration protocol. The different testing phases included in each experiment are presented in Table A.2.

Table A.1: Differences between experimental and control system factors across the experiments.

Experiment	Visual Feedback	Training	Constrained Motion	Motion Models	Control Output
1	No	No	No	KFMM	Velocity
2	No	No	No	KFMM	Velocity
3	No	Yes	No	KFMM	Velocity
4	Yes	Yes	No	KFMM	Position
5	Yes	Yes	Yes	KFMM	Position
6	Yes	No	Yes	HTMM	Torque
7	Yes	Yes	Yes	KFMM PMM NPMM	Position

Table A.2: Comparison of the experimental testing as it fits into the phases of the integration testing protocol.

Experiment	Software Simulation	Offline Remote Control	Online Remote Control
1	Yes	Yes	No
2	Yes	Yes	No
3	Yes	Yes	No
4	Yes	Yes	No
5	Yes	Yes	No
6	Yes	No	No
7	Yes	No	Yes

Appendix B

Rate Conversion Algorithm

The motion models implemented in this thesis work require that the sample frequency of all inputs are the same. As a result, rate conversion algorithms can be used to either upsample or downsample a signal, such that it matches a desired sampling frequency. There are many ways in which rate conversion can be achieved. This is due to the fact that representing a signal with a different number of samples is based on best-guesses. Both linear and nonlinear approximations of the new values can be developed. The choices of which algorithm to use depends on the data and the application of the data. For the interested reader, refer to Chapter 12 of Tan's book for fundamentals of rate conversion [257] and to Chapter 11 of Vaseghi's book for an in-depth look at interpolation (increasing the sampling rate) [258].

The experiments conducted in this research only required increases in sampling rate. Typically, the lowest possible sampling rate was either 1000 or 2000 Hz, as this was the sampling frequency of the EMG signals. All of the developed motion models required elbow position signals which were sampled at a much lower frequency than the EMG signals. In order to upsample these signals, a custom upsampling algorithms was developed as follows:

```
1 //Upsample() takes a list of samples and upsamples them to the desired
     frequency using linear methods.
2 //samples - a list of values to be upsampled
3 //desired_frequency - the desired sampling frequency of the new set of values
4 //duration - the amount of time in seconds that represents the number of
```

```
    values in the variable samples
public List<double> Upsample(ref List<double> samples, double
    desired_frequency, double duration)
{
  double temp = (desired_frequency * duration);
  //the number of new samples
  int desired_samples = Convert.ToInt32(Math.Round(temp));
  //a list to hold all of the new samples
  List<double> samples_us = new List<double>(desired_samples);
  //the number of original samples
  int actual_samples = samples.Count;
  //if there is more than one original sample
  if (actual_samples > 1)
  {
    //determine the distance between the original samples within the new
    samples
    double offset = (double)desired_samples / actual_samples;
    //a list for positions for which to copy the original samples into the
    new samples
    List<double> positions = new List<double>(actual_samples + 1);

    positions.Add(1);
    for (int i = 1; i < actual_samples; i++)
    {
      //fill the list of positions
      positions.Add(positions[i - 1] + offset);
    }
    positions.Add(desired_samples);

    int index = 1;
    //add the first original sample
    samples_us.Add(samples[0]);
```

```
33      for (int i = 1; i < actual_samples; i++)
34      {
35        int inner_index = index;
36        //get the next position
37        int next_position = Convert.ToInt32(Math.Ceiling(positions[i]));
38        //between the next position and the current index
39        for (int j = 0; j < next_position - inner_index; j++)
40        {
41          //add the previous original sample
42          samples_us.Add(samples[i - 1]);
43          index++;
44        }
45      }
46      //the remaining values after the last position
47      int remaining = desired_samples - samples_us.Count;
48      //if there are empty slots in the new samples
49      if (remaining > 0)
50      {
51        for (int i = 0; i < remaining; i++)
52          //fill them with the last value of the original samples
53          samples_us.Add(samples[actual_samples - 1]);
54      }
55      //if there are too many new samples after the last position
56      else if (remaining < 0)
57      {
58        for (int i = remaining; i > 0; i--)
59          //remove them
60          samples_us.RemoveAt(samples_us.Count - 1);
61      }
62    }
63    //if there was only 1 original sample
64    else
```

```
65   {
66     for (int i = 0; i < desired_samples; i++)
67       //fill the new samples with its values
68       samples_us.Add(samples[0]);
69   }
70   return samples_us;
71 }
```

This algorithm takes the original samples and distributes them linearly within the list of new samples, based on a constant offset. Next, the algorithm fills in the slots between two values with the former value. The final step is to either fill in missing values at the end of the new sample list or remove them to match the desired sampling frequency. For sampling frequencies, that are large relative to the original values, this upsampling process will cause steps within the signal. This may be fine for certain applications, but if a smoother signal is desired, then the new samples should be processed using filters or other signal smoothing techniques.

Appendix C

Approvals and Permissions

Three ethics approvals were granted in order to conduct the studies described within this thesis. One permisson was given for adaptation of Fig. D.10.

Figure C.1: Ethics approval for Experiments 1–6

Western Research

Research Ethics

Western University Health Science Research Ethics Board
HSREB Delegated Initial Approval Notice

Principal Investigator: Dr. Ana Luisa Trejos
Department & Institution: Unknown,Western University

HSREB File Number: 105717
Study Title: EMG-driven Model for Motion Prediction
Sponsor: Natural Sciences and Engineering Research Council

HSREB Initial Approval Date: October 06, 2014
HSREB Expiry Date: December 31, 2016

Documents Approved and/or Received for Information:

Document Name	Comments	Version Date
Western University Protocol		2014/09/22
Data Collection Form/Case Report Form		2014/09/22
Letter of Information & Consent		2014/09/22
Instruments	List of Instruments	2014/09/22

The Western University Health Science Research Ethics Board (HSREB) has reviewed and approved the above named study, as of the HSREB Initial Approval Datenoted above.

HSREB approval for this study remains valid until the HSREB Expiry Date noted above, conditional to timely submission and acceptance of HSREB ContinuingEthics Review. If an Updated Approval Notice is required prior to the HSREB Expiry Date, the Principal Investigator is responsible for completing andsubmitting an HSREB Updated Approval Form in a timely fashion.

The Western University HSREB operates in compliance with the Tri-Council Policy Statement Ethical Conduct for Research Involving Humans (TCPS2), theInternational Conference on Harmonization of Technical Requirements for Registration of Pharmaceuticals for Human Use Guideline for Good Clinical PracticePractices (ICH E6 R1), the Ontario Personal Health Information Protection Act (PHIPA, 2004), Part 4 of the Natural Health Product Regulations, Health CanadaMedical Device Regulations and Part C, Division 5, of the Food and Drug Regulations of Health Canada.

Members of the HSREB who are named as Investigators in research studies do not participate in discussions related to, nor vote on such studies when theyare presented to the REB.

The HSREB is registered with the U.S. Department of Health & Human Services under the IRB registration number IRB 00000940.

This is an official document. Please retain the original in your files.

[redacted]

Figure C.2: Ethics approval for therapist survey study.

Western Research

Research Ethics

Western University Health Science Research Ethics Board
HSREB Amendment Approval Notice

Principal Investigator: Dr. Ana Luisa Trejos
Department & Institution: Unknown,London Health Sciences Centre

Review Type: Delegated
HSREB File Number: 109034
Study Title: A Software System for Collection, Analysis and Presentation of Musculoskeletal Rehabilitation Data
Sponsor: Natural Sciences and Engineering Research Council

HSREB Amendment Approval Date: August 21, 2017
HSREB Expiry Date: May 09, 2018

Documents Approved and/or Received for Information:

Document Name	Comments	Version Date
Revised Western University Protocol		2017/07/27
Other	Phase I Questionnaire	2017/07/27
Revised Letter of Information & Consent		2017/07/27
Advertisement		2017/07/27

The Western University Health Science Research Ethics Board (HSREB) has reviewed and approved the amendment to the above named study, as of the HSREB Initial Approval Date noted above.

HSREB approval for this study remains valid until the HSREB Expiry Date noted above, conditional to timely submission and acceptance of HSREB Continuing Ethics Review.

The Western University HSREB operates in compliance with the Tri-Council Policy Statement Ethical Conduct for Research Involving Humans (TCPS2), the International Conference on Harmonization of Technical Requirements for Registration of Pharmaceuticals for Human Use Guideline for Good Clinical Practice Practices (ICH E6 R1), the Ontario Personal Health Information Protection Act (PHIPA, 2004), Part 4 of the Natural Health Product Regulations, Health Canada Medical Device Regulations and Part C, Division 5, of the Food and Drug Regulations of Health Canada.

Members of the HSREB who are named as Investigators in research studies do not participate in discussions related to, nor vote on such studies when they are presented to the REB.

The HSREB is registered with the U.S. Department of Health & Human Services under the IRB registration number IRB 00000940.

[redacted]

Figure C.3: Ethics approval for Experiment 7

Western Research

Date: 11 April 2018

To: Dr. Ana Luisa Trejos

Project ID: 110957

Study Title: Comparing Performance of Control Systems for a Wearable Mechatronic Elbow Brace in Remote-Controlled and Worn Scenarios

Application Type: HSREB Initial Application

Review Type: Delegated

Full Board Reporting Date: 01MAY2018

Date Approval Issued: 11/Apr/2018 09:25

REB Approval Expiry Date: 11/Apr/2019

Dear Dr. Ana Luisa Trejos

The Western University Health Science Research Ethics Board (HSREB) has reviewed and approved the above mentioned study as described in the WREM application form, as of the HSREB Initial Approval Date noted above. This research study is to be conducted by the investigator noted above. All other required institutional approvals must also be obtained prior to the conduct of the study.

Documents Approved:

Document Name	Document Type	Document Date	Document Version
Data Collection Form	Other Data Collection Instruments	08/Feb/2018	1.0
Letter of Information and Consent Form Phase I	Written Consent/Assent	10/Apr/2018	4.0
Letter of Information and Consent Form Phase II	Written Consent/Assent	10/Apr/2018	4.0
Research Protocol	Protocol	08/Feb/2018	1.0
Weekly Email Advertisement Phase I	Email Script	13/Mar/2018	1.0
Weekly Email Advertisement Phase II	Email Script	13/Mar/2018	1.0

Documents Acknowledged:

Document Name	Document Type	Document Date	Document Version
Elbow Position Collection Brace	Device Description	12/Dec/2017	1.0
Electronics Safety Approval for WearME Elbow Brace	Device Description	21/Mar/2018	1.0
Flexible measuring tape	Device Description	23/Jan/2018	1.0
Trigno_wireless_lab_manual	Device Description	12/Dec/2017	1.0
Velcro straps over cuffs	Device Description	23/Jan/2018	1.0
WearME Elbow Brace	Device Description	12/Dec/2017	1.0

No deviations from, or changes to, the protocol or WREM application should be initiated without prior written approval of an appropriate amendment from Western HSREB , except when necessary to eliminate immediate hazard(s) to study participants or when the change(s) involves only administrative or logistical aspects of the trial.

REB members involved in the research project do not participate in the review, discussion or decision.

The Western University HSREB operates in compliance with, and is constituted in accordance with, the requirements of the TriCouncil Policy Statement: Ethical Conduct for Research Involving Humans (TCPS 2); the International Conference on Harmonisation Good Clinical Practice Consolidated Guideline (ICH GCP); Part C, Division 5 of the Food and Drug Regulations; Part 4 of the Natural Health Products Regulations; Part 3 of the Medical Devices Regulations and the provisions of the Ontario Personal Health Information Protection Act (PHIPA 2004) and its applicable regulations. The HSREB is registered with the U.S. Department of Health &

Figure C.4: Permission to use and adapt Fig. D.10

Permission to Use Figure

Date Permission Granted: November 4th, 2014

Email conversation between Tyler Desplenter and Dr. Jim Johnson

Looks great.
Thanks for letting me know. JJ

Sent from my BlackBerry 10 smartphone on the Rogers network.

From: Tyler Desplenter
Sent: Tuesday, November 4, 2014 9:52 AM
To: Jim Johnson
Subject: Re: `~Re: Permission to use figure

Attached is the figure that I have redrawn (adapted). It is not identical to the figure in your lecture notes. I have included it in my literature review which eventually will go into my thesis and therefore that is why I need the reference or permission to use it.

Let me know what you think.
Tyler

On 11/04/14, **Jim Johnson** <[redacted]> wrote:
Hi Tyler,

The figure is not mine and the source is an old manuscript that I do not think I can locate. Do you need it for a school report? Otherwise, that one you describe may be an easy re-draw.

>>> Tyler Desplenter <[redacted]> 11/04/14 8:39 AM >>>
Hello Dr. Johnson,

I would like to adapt one of the figures from your BME 9505 course for my thesis. The lecture was Structure and Function of Soft Tissues and the figure is under section 4.2.2 describing tissue properties during immobilization, remobilization and exercise.

If you have taken this figure from the literature could your provide the reference? If the figure is yours, do I have permission to use/adapt it for my thesis?

Thank you,
Tyler Desplenter

Appendix D

Biomechanical Foundations

The following sections will describe some fundamental concepts, such as anatomy, nervous system signaling, and biomechanical principles, that are important to modelling human motion, while a detailed analysis of upper limb motion estimation models is presented in Chapter 2.

D.0.1 Anatomy and Physiology of the Upper Limb

The first human motion concept is the anatomy of the human body. The anatomy of humans can vary significantly depending on many factors, including genetic inheritance, physical activity, and presence of MSDs. As a species, humans have evolved in the ability to perform specific motion patterns. The universal nature of these patterns leads to a consistent placement of tissue structures, from a macroscopic view. This means that motion parameters can be averaged by looking across different populations, though the estimates will not be as accurate compared to parameters collected from the living tissue. Society has a inherent concept of what constitutes healthy or normal movement abilities. The goal of wearable assistive devices is to return those suffering from MSDs to a more healthy state of motion. By examining across populations of healthy inviduals, general characteristics of healthy motion can be assessed. A brief review of the structural and muscular contributions to upper limb motion is presented in the following sections.

D.0.1.1 The Shoulder

The shoulder is the upper limb structure that joins the arm to the torso. The shoulder provides most of the gross movements associated with the upper limb, while the other structures are used for the finer portions of the arm motions. The shoulder consists of four major joints (Fig. D.1): sternoclavicular, acromioclavicular, scapulothoracic, and glenohumeral. These joints work together to provide the upper limb with 12 gross movements of the shoulder. Flexion–extension, abduction-adduction and internal–external rotation, stemming mostly with the glenohumeral joint, are depicted in Fig. D.2, while elevation–depression, protraction–retraction and upward–downward rotation of the scapula are shown in Fig. D.3. The relationships between joints and the motion they facilitate and between the motions and the major muscle groups used to generate those motions, are listed in Table D.1 and Table D.2, respectively. Due to the complexity of the shoulder, the reader is encouraged to review [259] for a detailed description of the shoulder structure.

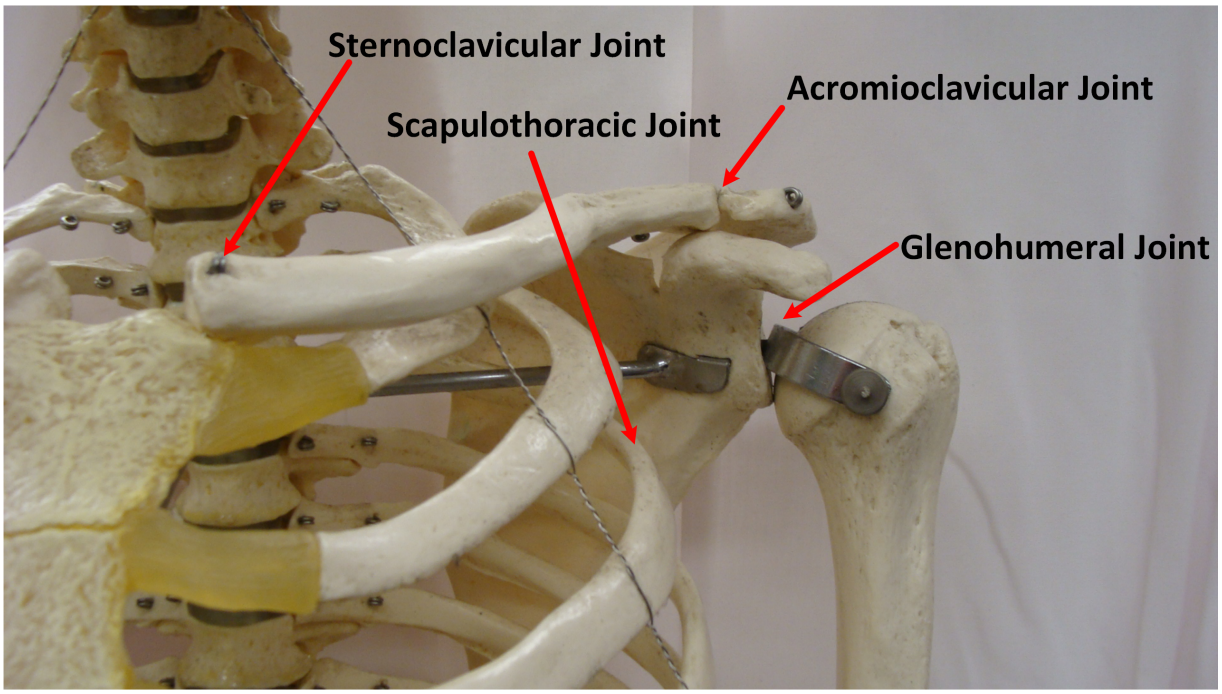


Figure D.1: Joints of the shoulder.

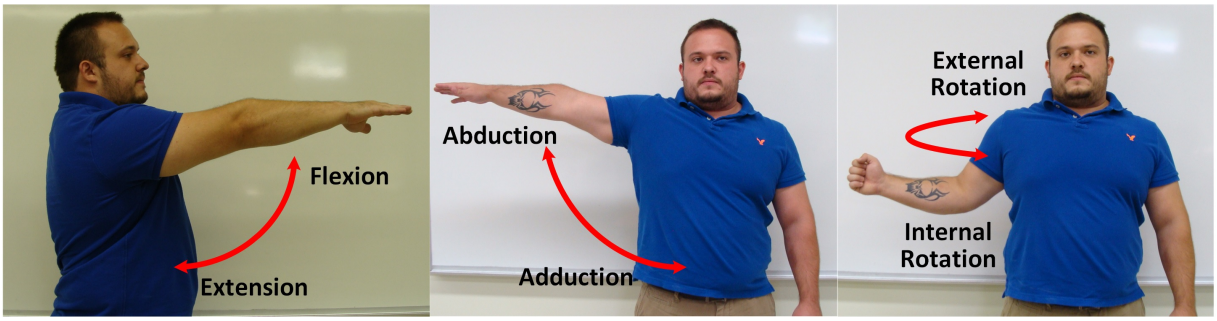


Figure D.2: Shoulder movements related to glenohumeral joint.

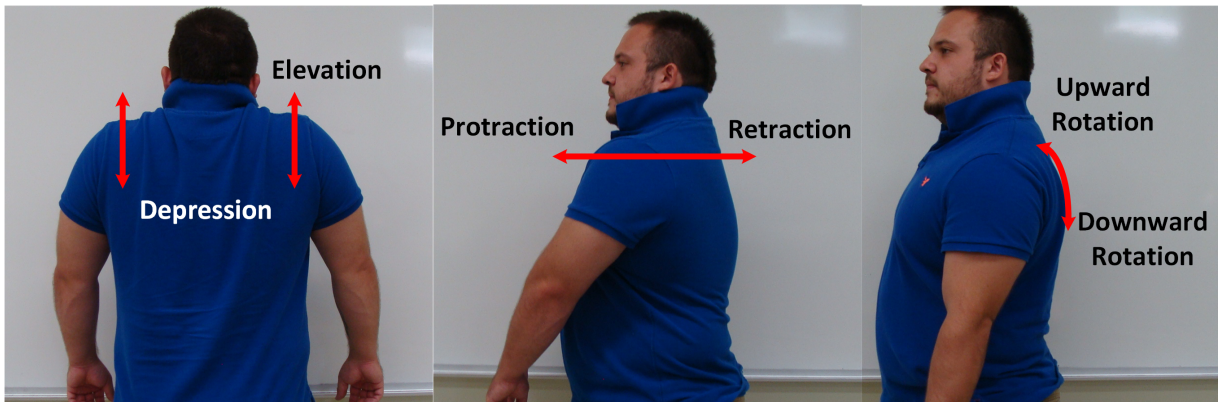


Figure D.3: Shoulder girdle movements.

Table D.1: Joint–motion relationships of the shoulder [260].

Shoulder Joint	**Primary Motions**
Sternoclavicular	Elevation–Depression Protraction–Retraction Upward–Downward Rotation
Acromioclavicular	Upward–Downward Rotation
Scapulothoracic	Elevation–Depression Protraction–Retraction Upward–Downward Rotation
Glenohumeral	Flexion–Extension Abduction–Adduction Internal–External Rotation

D.0.1.2 The Elbow

Moving distally down the upper limb from the shoulder, the next structure is the elbow. The elbow structure is comprised of the humero–radial, humero–ulnar, and proximal radio–ulnar joints.

Table D.2: Motion–muscle relationships of the shoulder [260].

Motion	**Primary Muscles**
Elevation	Upper trapezius Levator scapulae Rhomboids
Depression	Lower trapezius Latissimus dorsi Pectoralis minor Subclavius
Protraction	Serratus anterior
Retraction	Middle trapezius Rhomboids Lower trapezius
Upward Rotation	Serratus anterior Upper trapezius Lower trapezius
Downward Rotation [261]	Rhomboids Levator scapulae Pectoralis minor
Flexion	Anterior deltoid Coracobrachialis Biceps brachii (long head)
Extension	Posterior deltoid Latissimus dorsi Teres major Triceps brachii (long head) Pectoralis major (sterno-costal head)
Abduction	Anterior deltoid Middle deltoid Supraspinatus
Adduction	Posterior deltoid Latissimus dorsi Teres major Triceps brachii (long head) Pectoralis major (sterno-costal head)
Internal Rotation	Subscapularis Anterior deltoid Pectoralis major Latissimus dorsi Teres major
External Rotation	Infraspinatus Teres minor Posterior deltoid

However, the proximal radio–ulnar joint is part of the forearm as well and will be discussed in Section D.0.1.3. Removing the proximal radio–ulnar joint from this discussion, the elbow consists of the other two aforementioned joints that, together, roughly resemble a hinge joint. The term 'loose hinge' is given to these joints as their axis of rotation deviates from the initial position throughout the range of motion. The humero–radial joint is formed between the radius and humerus, while the humero–ulnar joint is formed between the humerus and ulna bones (see Fig. D.4). Although the joint axis of rotation shifts during motion, the shift is not significant enough to be considered another active degree of freedom of elbow motion. Therefore, the elbow only actively provides flexion and extension motions. These motions rotate the lower arm about the elbow joint axis as shown in Fig. D.5. Elbow flexion is produced mainly by the brachialis, biceps brachii, brachioradialis, and pronator teres muscles, while the anconeus and triceps brachii muscles produce the majority of elbow extension motion [260].

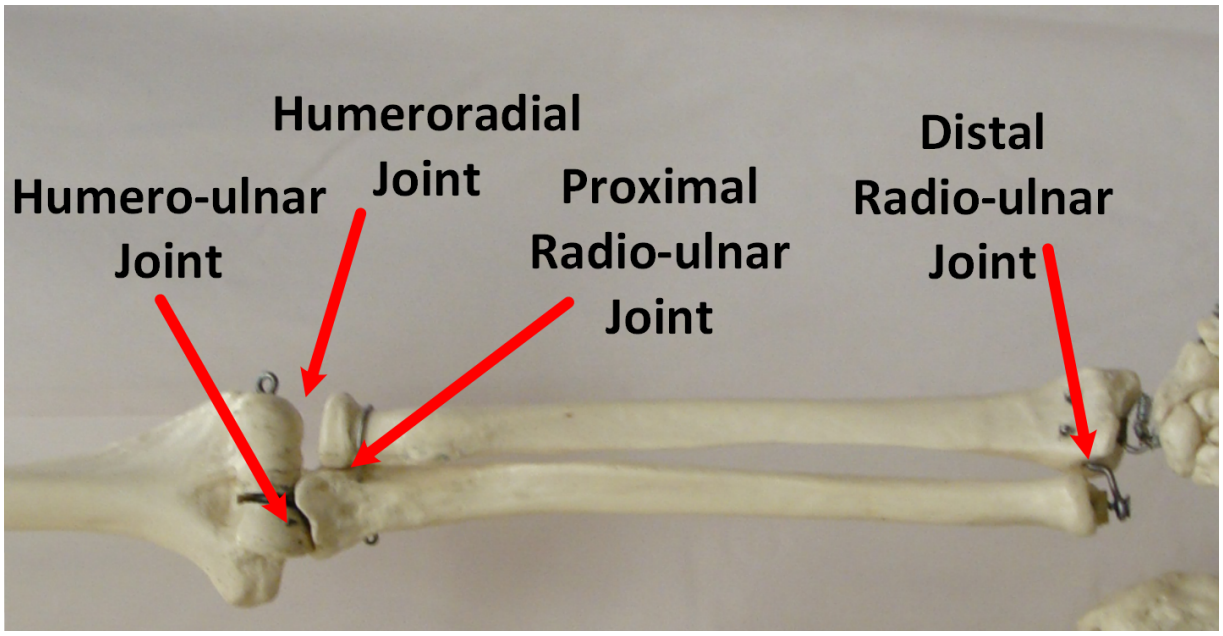


Figure D.4: Joints of the elbow and forearm.

D.0.1.3 The Forearm

The forearm provides the connection between elbow and wrist. The forearm is considered as a separate motion provider in this discussion, where its motion capabilities may be concatenated

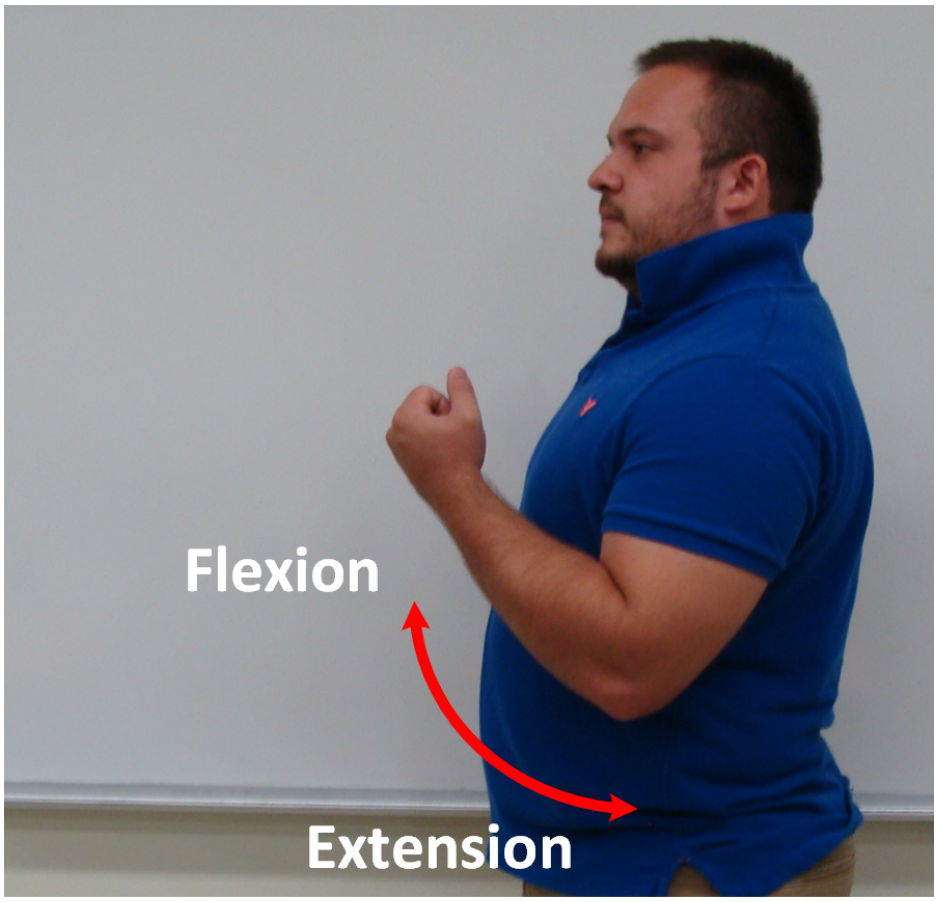


Figure D.5: Motion of the elbow.

with the elbow section of other texts. The forearm provides the ability to rotate the wrist and hand, known as pronation (palm down) and supination (palm up), about the long axis of the forearm. Motions of the forearm (Fig. D.6) are articulated through interaction of the radius and ulna bones. The radius rotates around the ulna at two contact locations, which form two radio–ulnar joints. The proximal radio–ulnar and distal radio–ulnar joints provide forearm motion at the elbow and wrist, respectively (Fig. D.4). The pronator quadratus and pronator teres muscles enable the pronation motion, while supination is enabled primarily by the supinator and biceps brachii muscles [260].

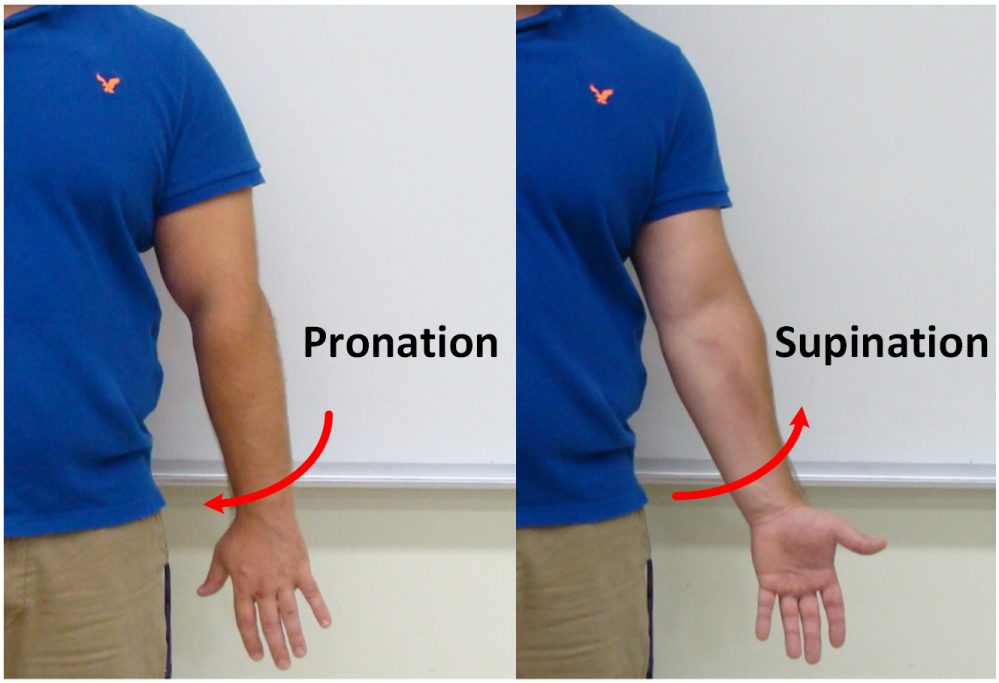


Figure D.6: Pronation and supination motions of the forearm.

D.0.1.4 The Wrist

The wrist structure is a very small structure compared to the rest of the upper limb but plays a crucial role in fine positioning of the hand. Several bone segments constitute the wrist structure. The radiocarpal and midcarpal joints are the main contributors to wrist motions (Fig. D.7). However, each of the bone structures comprising the wrist articulate with each other; known as intercarpal joints. The wrist is able to move in two active degrees of freedom. The radio–ulnar deviation and the flexion–extension motions of the wrist are depicted in Fig. D.8. The motions are provided by a series of muscles, with some of their attachments ranging as far as the elbow. Table D.3 lists the muscles associated with each of the wrist motions.

D.0.1.5 The Hand

The most distal structure in the upper limb chain is the hand. In some discussions, the hand includes the wrist structure, alongside the fingers and thumb. For this discussion, the hand will include only the joints located distally from midcarpal joint of the wrist. Three major types of joints, carpometacarpal, metacarpophalangeal and interphalangeal, allow for hand motion (see

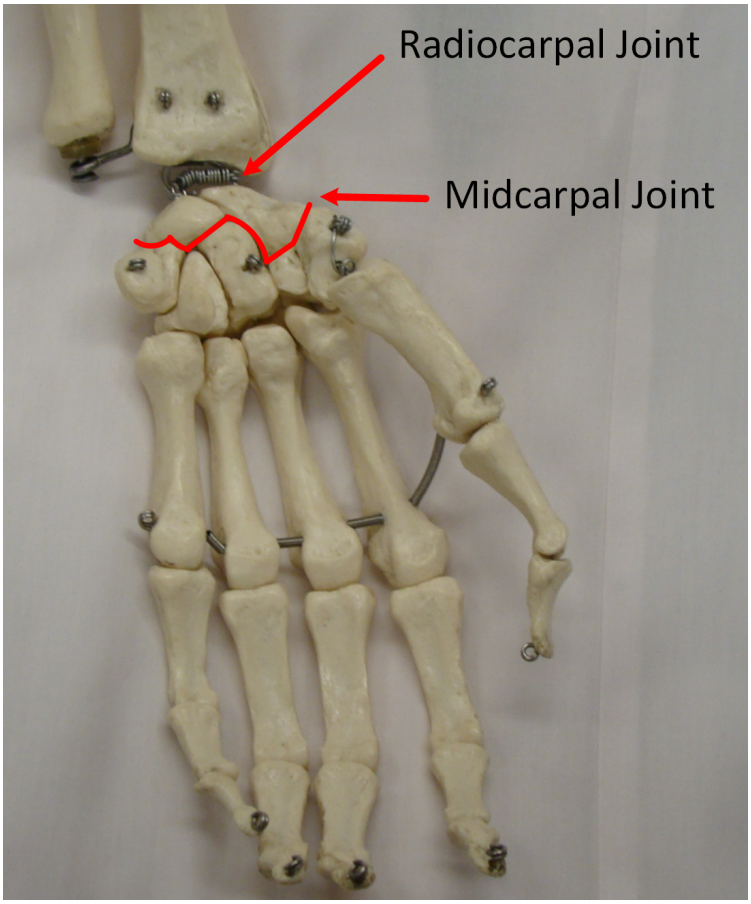


Figure D.7: Bone structure and joints of the wrist.

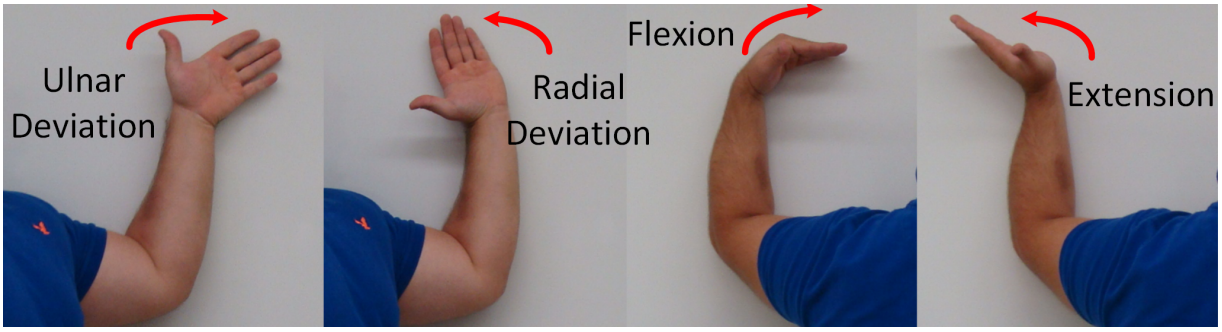


Figure D.8: Major motions of the wrist.

Fig. D.9). The fingers and the thumb can produce individual flexion–extension and abduction–adduction motions. However, the major movements of the hand are opening and closing, with different grasps, which involve synchronization of all hand joints. The primary muscles involved in opening the hand are the extensor digitorum, lumbricals, and interossei [260]. Closing the hand

Table D.3: Motion–muscle relationships of the wrist [260].

Motion	Primary Muscles
Flexion	Flexor carpi radialis Flexor carpi ulnaris Palmaris longus
Extension	Extensor carpi radialis longus Extensor carpi radialis brevis Extensor carpi ulnaris
Radial Deviation	Extensor carpi radialis longus Extensor carpi radialis brevis Extensor pollicis longus Extensor pollicis brevis Flexor carpi radialis Abductor pollicis longus Flexor pollicis longus
Ulnar Deviation	Extensor carpi ulnaris Flexor carpi ulnaris Flexor digitorum profundus Flexor digitorum superficialis

primarily involves the flexor digitorum profundus, flexor digitorum superficialis, and interossei muscles [260].

D.0.2 Nervous System Signalling

The second human motion concept surrounds the nervous system and the signals it produces. The body consists of biological electrical circuitry, which is referred to as the nervous system. The nervous system is a complex electrical system that transmits both communication and control signals between the tissues. This system is split up between the central nervous system and the peripheral nervous systems. The central nervous systems consists of the brain and spinal cord [262]. The peripheral nervous system consists of nerves that connect the central nervous system to all of the organs and musculature of the body [263]. Dissecting the complexity of the nervous system is an ongoing effort but the signals it produces are used for control of wearable assistive devices.

The difficulty of using these signals is that they are hard to interpret by human observation. However, using some basic assumptions, these signals can be transformed to representations that are interpretable and usable for control wearable assistive devices. First, the communications

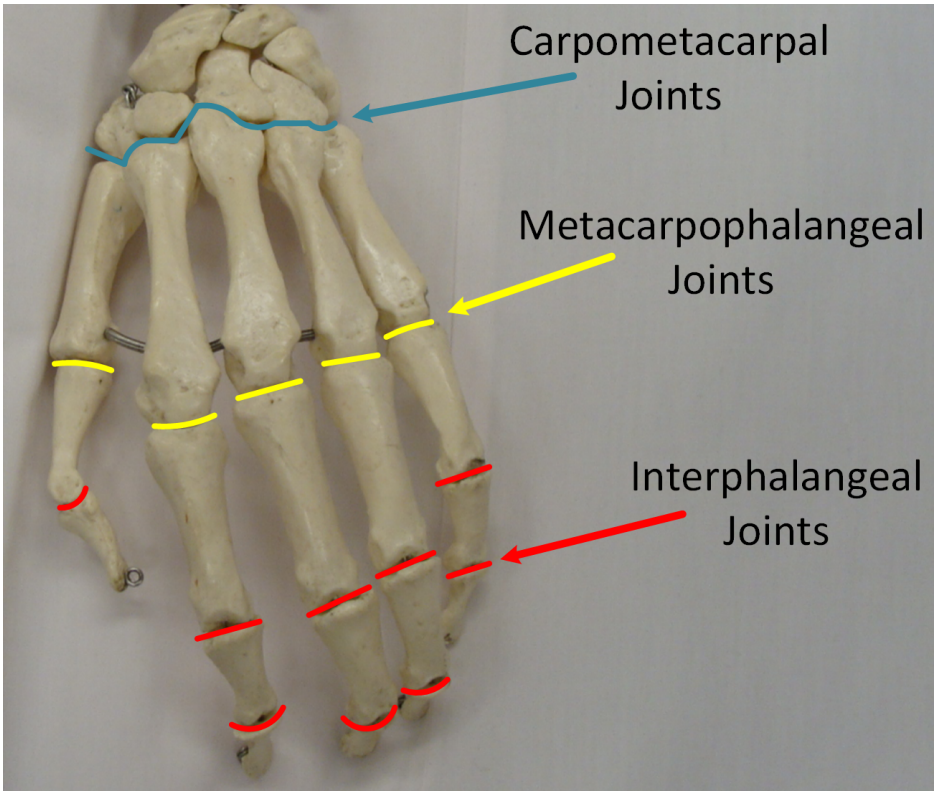


Figure D.9: The carpometacarpal, metacarpophalangeal, and interphalangeal joints of the human hand.

within the nervous systems are conducted using electrical signals. These electrical signals have a very small amplitude, typically, in the μVolt to mVolt range [117,264,265]. Second, the mechanisms that generate these signals have yet to be fully modelled or understood. However, the evidence suggests that the desired motion is encoded into these signals and can be extracted to estimate motion intention. Lastly, as with any communication circuitry, delays are present within the nervous system. By applying modern electrical signal processing techniques, these signals can become inputs into motion estimation models employed by the digital control systems of wearable assistive devices.

D.0.2.1 Extracting Meaningful Signals

Bioelectric signals can be captured from many locations on the surface of the human body. Non-invasive sensors allow for the capture of electrical signals, such as those operating the muscle (electromyography), heart (electrocardiography), eye (electrooculography), or brain (electroen-

cephalography). These sensors are placed on the skin over the area of the body where the signals exist and collect the change in voltage that occurs during control of the tissues. Commonly, these sensors use a location on the body known to have low electrical activity, such as the olecranon for EMG signals collected from the upper limb [122] or the ear lobe for EEG signals collected from the brain [266], as a voltage reference point. The collected voltage signals contain the information desired for estimation of human motion but require further processing to extract a set of data that are meaningful to developers of digital control systems.

For wearable assistive devices, the two most common types of bioelectric signals currently are EEG and EMG signals (Table 2.1). This is due to the fact that the motion commands are generated in the brain and produce motion of the musculature. Since the mechanisms responsible for producing these signals are not fully understood, a series of signal processing techniques, such as those described in Section 2.5, are used to transform the signal into something more meaningful. As a result, the ability to use these signals relies heavily on the ability to produce these transformations accurately and to meet timing constraints of the system.

In general, both EEG and EMG data are amplified, filtered, and converted to a common sampling rate to synchronize with other data sources. However, the details of the processing procedures differ between these two types of data. EEG signals processing includes filtering, scaling, artifact detection, and signal segmentation [267]. Filtering EEG signals is accomplished through high-pass and low-pass filters, whose cut-off frequencies are >0.5 and 40–70 Hz, respectively. Once the EEG signals have been processed, features are extracted from the signals and classification techniques are used to determine states of the individual's motion. The standard processing of EMG signals involves scaling, filtering, and further scaling [25, 122, 124, 125]. Amplification is the first scaling function that increases the signal amplitude in order to increase the resolution of the processing computations. Band-pass filters remove high frequency components, which are difficult to interpret, and low frequency motion artifacts. A rectification and normalization process is applied to this signal, using the muscle's MVC, to bound the signal between 0 and 1. The transformed EEG and EMG signals are suitable to be used as input to the existing control systems. Further information on the processing of EEG and EMG signals can be found in [267] and [133, 268], respectively.

D.0.2.2 Electromechanical Delays

The period of time beginning when a controller sends a command and ending when the system responds to the command is known as the response delay. When such a delay is discussed as it pertains to human motion, it is known as the electromechanical delay (EMD) of the system, due to the type of energy changes that must occur to produce a motion response. A similar energy transformation must occur for mechatronic systems, and, hence, the same term is used. The EMD is a characteristic of the human musculoskeletal system and must be accommodated to achieve accurate estimation and tracking of human motion. The EMD of the musculoskeletal system places response constraints on the control system of mechatronic devices [106,135,207,208,235,269]. If the device responds faster than the human, the system will drag the human through the motion [207]. A slower response time than that of the human means that the assistance provided by the device will be delayed and diminishes the effectiveness of using the system for motion assistance. The small time window, in which the EMD should be, limits the magnitude of the feedback gains that can be used [235,269]. Therefore, the response time of the mechatronic system must correlate with the response of the musculoskeletal system.

Stationary assistive devices can be designed to have a response time much faster than that of the human. It has been suggested that the part of the EMD can be exploited for computations, such as motion predictions, due to the ability of modern computer systems to process data in microseconds [135, 157]. The duration of the EMD of the musculoskeletal system has been listed in the range of 23–131 ms [135]. A study of the elbow listed EMDs for triceps and biceps brachii muscles in the range of 25–45 ms [243]. When human eyes are added to the equation, the visual reaction time is around 150 ms [270]. Tang *et al.* have suggested a maximum 300 ms response time in order to eliminate the sensation of the delay to the user [44]. Therefore, the EMDs is dependent on multiple factors but has an upper bound to ensure appropriate behaviour of the system.

Due to the number of parameters needed to tune the EMD and the short duration of most EMG recordings, the EMD parameter is usually chosen to be a constant value in motion models to reduce complexity. Previous studies of control of human muscle are used to inform parameter values [169, 174, 247, 248]. However, Winters describes the relationship between these delays and

the chemical processes of the muscle tissue [271]. Since the availability of chemicals may vary with respect to time, it is reasonable that EMDs would also vary with time. Recently, Downey *et al.* produced some evidence to support this rationale by showing changes in EMD in response to fatigue induced in the quadriceps femoris muscle [272]. Considering a constant EMD may simplify modelling of human motor control but could contribute to reduced performance if the EMD changes over a series of recordings for which the EMD value is held constant within the model. Further study is required to characterize this motion parameter.

D.0.3 Biomechanic Principles of the Musculoskeletal System

The third human motion concept encompasses the biomechanical properties of the body. Biomechanics is the study of the mechanical properties of biological structures, which are the human tissue structures in this case. Understanding the properties and functionality of these tissues leads to the development of models. Combining the models for the various structures of the human body allows meaningful relationships to be established between biological signals, such as EMG signals, and the motion of the body segments. Motion of the musculoskeletal system is described through the kinematic and dynamic properties. For human bodies, kinematic properties describe relationships between joint positions and joint position derivatives, while dynamic properties form relationships between joint position changes and the forces that cause those changes. Estimating human motion becomes possible through modelling iof the biomechanical properties of the musculoskeletal system.

D.0.3.1 Kinematics of the Musculoskeletal System

Wearable systems form a unique connection between human and machine. In order to have high quality coordination between these two moving systems, the kinematics of the human must be defined. Understanding the capacity of each human joint will allow for the device to mimic or provide motion that is deemed *natural*. The musculoskeletal system can be decomposed into a series of joints connected by bone segments (linkages). Each joint has one or more degrees of freedom, each with a specific range of motion. The range of motion of healthy human joints informs the design of mechanical, sensing, actuation, and control system. Accepted standards for

range of motion of upper limb movements are presented in Table D.4.

Table D.4: Range of motion of upper limb movements of healthy individuals [260].

Upper limb Structure	**Motion**	**Range of Motion (°)**
Shoulder	Depression/Elevation	-10 to 45
	Protraction/Retraction	-30 to 30
	Upward/Downward Rotation	0 to 30
	Adduction/Abduction	0 to 120
	Extension/Flexion	-65 to 180
	Internal/External Rotation	-85 to 70
Elbow	Extension/Flexion	-5 to 145
Forearm	Pronation/Supination	-85 to 75
Wrist	Extension/Flexion	-75 to 85
	Radial/Ulnar Deviation	-20 to 40

Note: 0° is the neutral position of the joint for each movement.

In kinematic analysis, joint velocity and joint accelerations, are also valuable information. These quantities are crucial to picking appropriate actuation technology and setting safety requirements. The device's actuators need to be able to operate at velocity and accelerations that are equivalent to that of the assistance required. The goal of the devices is to help individuals regain the ability to produce normal levels of joint velocity and acceleration, meaning that these normal values are a great starting point for developers of wearable assistive devices to meet. Rosen *et al.* captured joint velocity and acceleration data from one subject performing ADLs [273]. Maximum joint velocity and acceleration data have been gathered from Rosen's work and presented in Table D.5. Although these data were only captured from one individual, it still provides a quantitative insight for joint velocity and acceleration requirements.

Table D.5: Maximum and minimum angular velocity and acceleration values captured during activities of daily living [273]

Movement	**Maximum Angular Velocity (°/s)**	**Minimum Angular Velocity (°/s)**	**Maximum Angular Acceleration (°/s^2)**	**Minimum Angular Acceleration (°/s^2)**
Shoulder Flexion–Extension	95.0	-136.7	924.4	-949.7
Shoulder Abduction–Adduction	171.5	-134.1	1311.7	-1282.7
Shoulder Internal–External Rotation	113.0	-140.7	1020.3	-1049.0
Elbow Flexion–Extension	145.8	-172.8	1214.3	-1266.1
Forearm Pronation–Supination	486.3	-412.8	3715.5	-4343.6
Wrist Flexion–Extension	232.9	-141.2	2116.2	-2790.1
Wrist Radial–Ulnar Deviation	203.9	-180.4	1822.4	-2476.2

D.0.3.2 Dynamics of the Musculoskeletal System

The dynamics of the musculoskeletal system describe the relationships between the forces or torques that cause motion of joints. Each of the musculoskeletal tissues has a variety of properties that influence its force contributions to joint motion and need to be modelled to have accurate motion estimation models. As with neurological signals, motion signals must be captured externally on the skin. This creates problems for modelling the dynamic properties of these tissues as many aspects of their function cannot be measured from the surface of the skin. One solution is to study cadaver tissues to determine these dynamic properties, such as wrist stiffness [274], elbow and

forearm muscle moment arms [275] and muscle length of shoulder muscles [276]. However, there is a large amount of variability in these properties within each musculoskeletal system, between two musculoskeletal systems and between the living and deceased states of each musculoskeletal system. As a result, models of the dynamic properties can become mathematically indeterminant systems, where the number of unknowns is larger than the number of equations describing the system.

To solve this issue, optimization procedures can be performed on motion models to determine the optimal properties to produce a desired or recorded motion [96, 122, 128, 136, 169, 277]. Properties measured from cadavers or living humans may be used as starting conditions for the optimization, where the general goal is to minimize the error between model estimations and some measured or pre-defined movement. However, the complexity and duration of the optimziation is proportional to the number of parameters that must be optimized. Resultantly, many of the model parameters are chosen to be constant (static) values to reduce the optimization expense. In reality, these parameters vary based on many factors, such as the availability of chemical compounds [278].

The musculoskeletal system does exhibit trends in high-level properties that can help guide the development of motion models. These trends can be seen in the torque requirements, viscoelasticity of soft tissues, motion compliance, and immobilization of tissues. The following sections will describe these dynamic properties, while an analysis of musculoskeletal dynamics is presented in Chapter 5.

D.0.3.2.1 Joint Torque Joint torque is the summation of the moments caused by forces applied to a particular joint. The torque required to complete a motion task varies based on many factors, which include position of the joint, presence of tissue damage, mental focus, addition of a load, properties of the performer's body, level of training, task goals, and chosen trajectory. For example, biological differences in maximum joint torque can be seen between men and women performing the same movement [260, 279, 280]. Some of the lower torque differences for women compared to men can be attributed to lower mass and shorter limb-segments, on average. However, biological differences alone will not explain the entirety of these differences as each individual varies in their anatomy, mental abilities, strength, level of training, and efficiency of movement. Table

D.6 shows differences in joint torque captured from men and women performing uni-directional joint movements.

Assisting with human motion means that the human and device together must be capable of applying a joint torque equivalent to that of a healthy individual completing the motion task. Commonly, assistive mechatronic devices are concerned with restoring the ability to perform ADLs. Tasks in this category are any motions that help individuals live with a high quality of life, like brushing teeth, washing the body, feeding, sitting, and climbing stairs. Rosen *et al.* performed a quantification of motion parameters for an individual performing common ADLs [273]. The data from this study demonstrate the differences in maximum joint torque required to complete the following tasks: arm reach to head level, moving an object to waist level, pick up and hang up a phone on the wall, and eating with a spoon with two different grips. Although these data do not represent the population, it gives an insight to torque requirements for actuation and control. Maximum joint torques were taken from Rosen's study and compiled in Table D.7.

Table D.6: Difference in joint torques between men and women for uni-directional movements of the upper limb.

Upper limb Structure	**Motion**	**Male Torque (Nm)**	**Female Torque (Nm)**
Shoulder [279]	Flexion	77.1	47.7
	Extension	113.9	59.8
	Abduction	73.1	41.8
	Adduction	97.5	50.2
	Internal Rotation	31.3	19.6
	External Rotation	45.2	26.3
Elbow [260]	Flexion	71.1	32.9
	Extension	41.3	20.6
Forearm [260]	Pronation	7.2	3.5
	Supination	8.9	4.3
Hand [280]	Flexion	10.5	5.6
	Extension	12.8	6.2

Table D.7: Maximum joint torques required for a healthy male to complete five ADLs [273]. Torque values (Nm) are taken as the maximum absolute values of each movement.

Movement	**Activities of Daily Living**				
	Arm Reach to Head Level	Move Object to Waist Level	Pick Up Phone on Wall/Hang Up	Eat with Spoon	Eat with Spoon (Power–disabled grasp)
Shoulder Flexion–Extension	6.99	9.63	9.03	4.49	4.88
Shoulder Abduction–Adduction	10.06	7.38	2.91	2.58	3.06
Shoulder Internal–External Rotation	2.79	0.61	1.21	0.76	3.10
Elbow Flexion–Extension	3.51	3.76	3.07	1.43	0.52
Forearm Pronation–Supination	0.04	0.04	0.04	0.02	0.02
Wrist Flexion–Extension	0.44	2.79	0.87	0.37	0.25
Wrist Radial–Ulnar Deviation	0.31	0.37	0.28	0.27	0.04

D.0.3.2.2 Viscoelasticity of Soft Tissue Soft tissues of the musculoskeletal systems exhibit a viscoelastic response to a load. This response shows a combination of both viscous and elastic properties. Viscosity of the tissue is heavily affected by hydration level and temperature, which cause the density of these tissues to vary. Measuring both the temperature and hydration level in a specific tissue would require invasive measurement techniques. These techniques are dangerous and cannot be used with wearable assistive devices. One potential solution is to gather data

from the surface of the body and study the relationships between these measures and internal parameters. For example, relationships between skin temperature and muscle temperature or between perspiration and soft tissue hydration levels could be established. The elasticity of the tissues refers to the amount of stresses it can accommodate given variations of strain rate. Strain rate is a measurement of the rate of deformation of a material. The faster the tissue is strained, the higher the maximum stress the tissue can sustain without deforming beyond its elastic region. Strain rate varies depending on the motion task demands and is also difficult to measure in living tissue under normal motion conditions.

Currently, existing motion models do not take into consideration biological viscoelasticity parameters due to the difficulty in measurement and complex dependencies. Instead, assumptions are made and models are simplified. In muscle models, such as Hill-based muscle models, the viscoelastic response has been modelled as a spring [122,169,172,188,248], a spring and damper [170], or an exponential model [121, 135, 249]. The issue with these models is that they are not flexible to changes in the biological parameters that affect the viscoelastic response. For the parameters of these simplified models, assumptions are made that the parameters are constant over time. These assumptions simplify the models, reduce the number of computations and, may hold true for one or more motions completed by healthy individuals. However, these parameters will vary depending on many factors, including the presence of musculoskeletal trauma, which wearable assistive devices must take into account.

D.0.3.2.3 Motion Compliance Humans have evolved a safety mechanism termed motion compliance. Motion compliance is the degree to which a body segment will comply with an external disturbance force when it is applied to that segment. More commonly, stiffness is the term used to describe the resistance of motion and, therefore, is inversely related to compliance. Humans are able to adjust the degree of compliance through activation of muscles. A joint, to which the external force is applied, will have its highest compliance (lowest stiffness) when the muscles are relaxed and its lowest compliance (highest stiffness) when the muscles are maximally contracted. Joint stiffness has been observed as 1 Nm/rad for a relaxed elbow [281], 17.3 Nm/rad on average for cyclical elbow flexion–extension at speeds of 1.6–2.1 rad/s [116], and greater than

40 Nm/rad at the elbow during multi-joint movements [282]. Zawadzki and Siemieński found that elbow joint stiffness was modulated, through increased muscle tension between pairs of antagonistic muscle groups, in the range of 4–130 Nm/rad and was correlated with the frequency of the limb segment motion [283].

These results suggest that joint stiffness is a dynamic motion property that is influenced by all soft tissues involved in a joint's motion. In existing motion models, stiffness is commonly described through contribution of passive muscles and tendon forces. The viscoelastic tissue models described in Section D.0.3.2.3 are essentially models of the passive stiffness of muscle and tendon tissues. These models detail the relationship between the passive tissue force as a function of change in length. Since joint stiffness is at least partially contributed to by muscular torque [244], stiffness contributions in most of these models come solely from the passive forces of the muscle and tendon tissues. However, Stroeve points out that these Hill-based models may not fully describe the intrinsic stiffness [245]. To combat this problem, Chadwick *et al.* include an additional joint stiffness model, alongside their Hill-based muscle models [169]. This model considers the joint stiffness as a piece-wise-continuous nonlinear spring with constant stiffness coefficients. The inclusion of this model may, in some cases, account for some of the intrinsic stiffness not captured by other models but suffers from the fact that the chosen coefficients were constant across time, which is likely to differ from reality.

Joint stiffness is an important characteristics to model accurately in order to provide assistance with wearable devices. Straathof *et al.* developed a joint-stiffness estimation and compensation method for their wheelchair-mounted planar arm support, which estimated the joint-stiffness values over a work area in which reaching movements were performed by a patient suffering from Duchenne Muscular Dystrophy [238]. This joint-stiffness estimation and compensation method outperformed the control of the device without joint-stiffness compensation but was less effective than an EMG-driven control strategy. In general, the literature supports the need to include models of joint stiffness, from all contributing tissue sources, to accurately estimate motion and control devices. However, effective global solutions have yet to be realized. Stiffness or motion compliance requires further research to increase the accuracy of motion estimation models.

D.0.3.2.4 Immobilization and Aging The strength of human soft tissues decreases with both immobilization and aging. Additionally, a loss of mobility, in general, occurs with an increase in age. The ultimate strength of bone decreases by about 2% per decade from age 20 to age 90 [284]. Muscle strength deterioration can occur much more rapidly than that of bone, especially in the presence of disease or injury. Thelen points out that healthy older adults ($\sim$70 years old) experience loss of muscle strength, prolonged twitch contractions, increased passive stiffness, and lower rates of muscle force development, compared to healthy young adults [248]. In terms of modelling, Thelen showed that by adjusting his Hill-based model parameters to account for aging resulted in estimated maximum torques and average power outputs for ankle motion that match quantities measured in older populations [248]. This evidence supports the notion that relationships between model parameters and age are significant in determining accurate motion estimates.

Injury to soft tissues requires some period of immobilization in order to heal. Immobilization can cause a loss of strength of up to 40% (Fig. D.10) compared to pre-injury strength at only 10 days after the onset of the injury [260]. Activation of muscles is affected by both immobilization and age. During the immobilization phase of recovery, neural signals that activate muscles may differ than the signals produced prior to injury. Since tissues need mechanical stresses to gain strength, the result of immobilization is deterioration of the muscle tissues. One side effect of not modulating muscles is that tendon and ligaments can also be significantly weakened [285]. Model parameters will, therefore, dynamically vary based on the health of each tissue. Relationships between tissue parameters and tissue health should be further investigated to better estimate human motion.

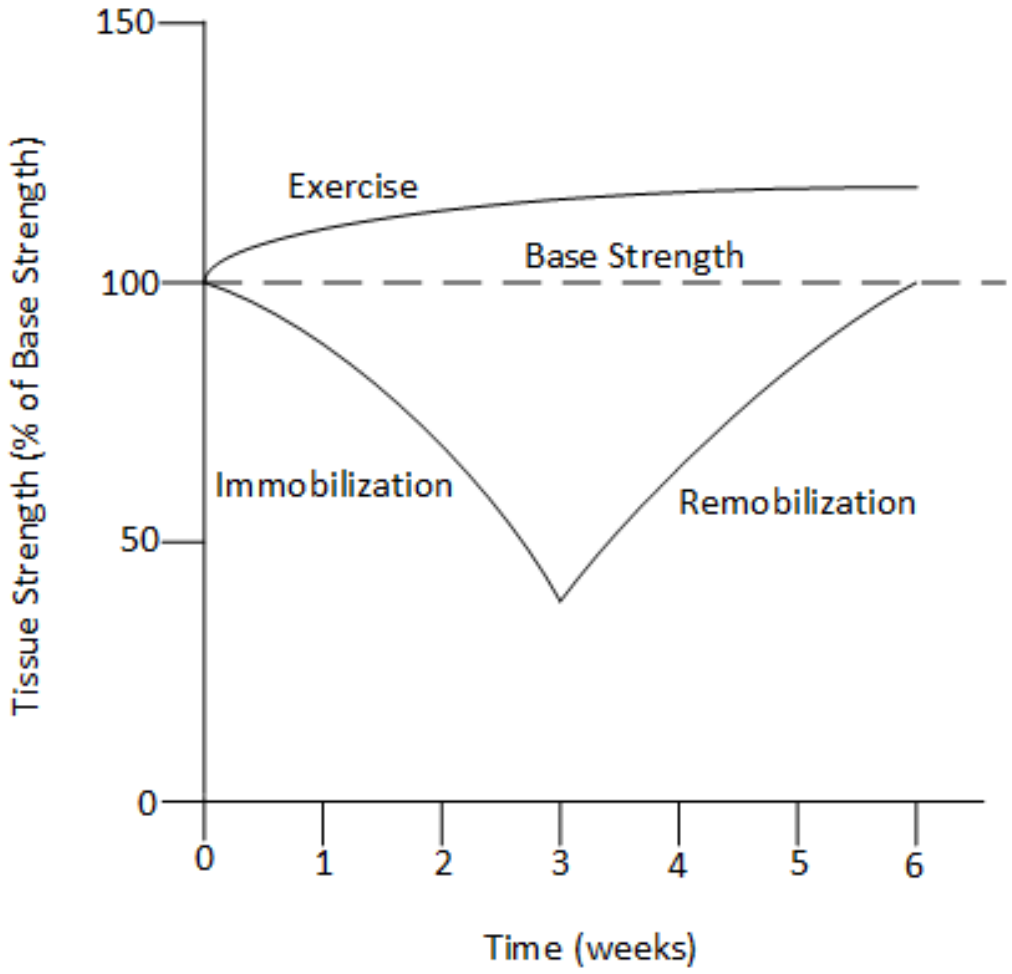


Figure D.10: The effects of exercise and immobilization on the strength of tissues (adapted from [286])

Curriculum Vitae

Tyler Desplenter

Education

Sept. 2013– Present	PhD, Robotics and Control Systems, The University of Western Ontario Thesis: Development of Digital Control Systems for Wearable Mechatronic Devices: Applications in Musculoskeletal Rehabilitation of the Upper Limb Supervisor: Dr. Ana Luisa Trejos
Sept. 2016– May 2017	Higher Education Certificate, The University of Western Ontario Courses: Curriculum Development in Higher Education, Teaching and Learning in Higher Education
Sept. 2008– Aug. 2013	Bachelor of Engineering Science, Software Engineering, Embedded Systems Specialization, The University of Western Ontario

Teaching Interests

Embedded System Design and Development
Software Development, Architectures and Quality
Biomedical Device Development

Teaching Experience

Winter 2018	Teaching Assistant, The University of Western Ontario Course: ECE 3390 Hardware/Software Co-design
Fall 2017	Teaching Assistant, The University of Western Ontario Course: SE 2203 Software Design
Winter 2017	Teaching Assistant, The University of Western Ontario Course: SE 2205 Data Structures and Algorithms
Fall 2016	Teaching Assistant, The University of Western Ontario Course: SE 2203 Software Design
Winter 2016	Teaching Assistant, The University of Western Ontario Course: SE 2250 Software Construction
Fall 2015	Teaching Assistant, The University of Western Ontario Course: ES 1036 Programming Fundamentals for Engineers
Fall 2015	Teaching Assistant, The University of Western Ontario Course: SE 4450 Software Engineering Design II

Winter 2015	Teaching Assistant, The University of Western Ontario
	Course: SE 3350 Software Engineering Design I
Fall 2014	Teaching Assistant, The University of Western Ontario
	Course: SE 3350 Software Engineering Design I
Winter 2014	Teaching Assistant, The University of Western Ontario
	Course: SE 2203 Software Design
Fall 2013	Teaching Assistant, The University of Western Ontario
	Course: SE 3352 Software Requirements and Analysis

Research Interests

Embedded System Development
Control of Wearable Mechatronic Devices
Biological Data Collection and Analysis Platforms
Biomechanics and Motion Modelling

Research Experience

Sept. 2013–	Graduate Research Assistant, The University of Western Ontario
Present	Developed and assessed computer systems for wearable mechatronic devices
	Industry collaboration with Intronix Technologies Corp.
May 2015–	Graduate Research Assistant, The University of Twente
Aug. 2015	Academic exchange to develop control systems for a wearable arm support
Jan. 2013–	Research Assistant, The University of Western Ontario
March 2013	Examined literature and developed a document detailing the state of the art on Software Defined Networking
Sept. 2012–	Research Assistant, The University of Western Ontario
Dec. 2012	Project: Application Extension Facility (AEF) 2.0
	Abstract the functionality of AEF 1.0 to automatically generate a C# API and web service from any JSON-based data structure.
May 2012–	Research Assistant, The University of Western Ontario
Aug. 2012	Project: Application Extension Facility 1.0
	Developed a software application to automatically generate a C# API and web service based on the private JSON-based database structure of www.uniiverse.com

Supervisory Experience

May 2017–	Project Supervisor, The University of Western Ontario
Aug. 2017	Supervision of an undergraduate student project involving the development of an embedded systems for control of a wearable mechatronic elbow brace.
May 2016–	Project Supervisor, The University of Western Ontario
Aug. 2016	Supervision of an undergraduate student project involving the development of an embedded system for driving a DC brushless motor.

Oct. 2015–	Project Supervisor, The University of Western Ontario
June 2016	Supervision of a high school co-op student project involving assessment of various motion tracking technologies.
May 2014–	Project Supervisor, The University of Western Ontario
Aug. 2014	Supervision of a M.Eng. graduate student project on developing a software interface for wearable devices.

Honours and Awards

2016	Graduate Student Travel Award, Department of Electrical and Computer Engineering, The University of Western Ontario
2016	Queen Elizabeth II Graduate Scholarship, The University of Western Ontario
2015	Ontario Graduate Scholarship, The University of Western Ontario
2015	Global Opportunities Award, The University of Western Ontario
2013	Dean's Honour List, Faculty of Engineering, The University of Western Ontario
2012	Summer Student Research Award, Department of Electrical and Computer Engineering, The University of Western Ontario
2012	Dean's Honour List, Faculty of Engineering, The University of Western Ontario
2011	Dean's Honour List, Faculty of Engineering, The University of Western Ontario

Work in Progress

T. Desplenter, A. Kyrylova, A. Escoto, S. Chinchalkar and A.L. Trejos. Development of a Wearable Mechatronics-Enabled Elbow Brace for Motion Rehabilitation after Brachial Plexus Injury. Submitted to *Journal of Rehabilitation and Assistive Technologies Engineering*.

T. Desplenter, Y. Zhou, B. Edmonds, A. Goldman, M. Lidka and A.L. Trejos. Review of Wearable Assistive Mechatronic Devices for the Upper-Limb. Planned submission to *Journal of Rehabilitation and Assistive Technologies Engineering*.

Journal Publications

T. Desplenter and A.L. Trejos. Evaluating Muscle Activation Models for Elbow Motion Estimation. *Sensors*, vol. 18, no. 4, pp. 1–19.

Conference Publications

T. Desplenter, J. Lobo-Prat, A.H.A. Stienen and A.L. Trejos. "Extension of the WearME Framework for EMG-driven Control of a Wearable Assistive Exoskeleton. in *IEEE International Conference on Advanced Intelligent Mechatronics*, Banff, Canada, 2016, pp. 288–293. (Oral)

T. Desplenter, A. Kyrylova and A.L. Trejos. Development of an EMG-Driven Control System for a Wearable Mechatronic Elbow Brace. in *IEEE International Conference on Biomedical and Health Informatics*, Las Vegas, USA, 2016, pp. 501–504. (Oral)

A. Kyrylova, T. Desplenter, A. Escoto, S. Chinchalkar and A.L. Trejos. Simplified EMG-driven Model for Active-Assisted Therapy. in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Rehabilitation & Assistive Robotics*, Chicago, USA, 2014.

T. Desplenter, A. Kyrylova, T.K. Stanbury, S. Chinchalkar, A. Escoto and A.L. Trejos. A Wearable Mechatronic Brace for Arm Rehabilitation, in *IEEE International Conference on Biomedical Robotics and Biomechatronics*, Sao Paulo, Brazil, 2014, pp. 491–496. (Poster)

Conference Presentations

T. Desplenter and A.L. Trejos. Elbow Motion Repeatability: Implications for Musculoskeletal Rehabilitation. *Canadian Bone and Joint Conference*, London, Canada, 2016. (Poster)

T. Desplenter, T.K. Stanbury, A. Escoto, S. Chinchalkar and A.L. Trejos. Design and Control of a Mechatronic Elbow Brace for Rehabilitation. *London Health Research Day*, London, Canada, March 18, 2014. (Poster)

Review Work

S.A.M. Lajimi and J. McPhee. A comprehensive filter to reduce drift from Euler angles, velocity, and position using an IMU. *IEEE Canadian Conference on Electrical and Computer Engineering.* Reviewed March 13, 2017.

Aitziber Mancisidor, *et al.* Enhanced force control using force estimation and nonlinearity compensation for the Universal Haptic Pantograph. *IEEE/RSJ International Conference on Intelligent Robots and Systems.* Reviewed: May 7, 2015.

Community Outreach

July 2018	Volunteer at Ontarios Strongest Man competition
Oct. 2016	Panelist at a powerlifting seminar, Heavy Lifting: A Powerlifting Panel for Everyone
May 2016–	Founder and member of Western Strength a club for strength athletes
Present	at the University of Western Ontario
March 2016	Volunteer at electrical and computer engineering department Intent to Register session
March 2015	Research presentation to first year engineering students
April 2014	Judge at Thames Valley Science and Engineering Fair
March 2014	Research presentation to first year engineering students