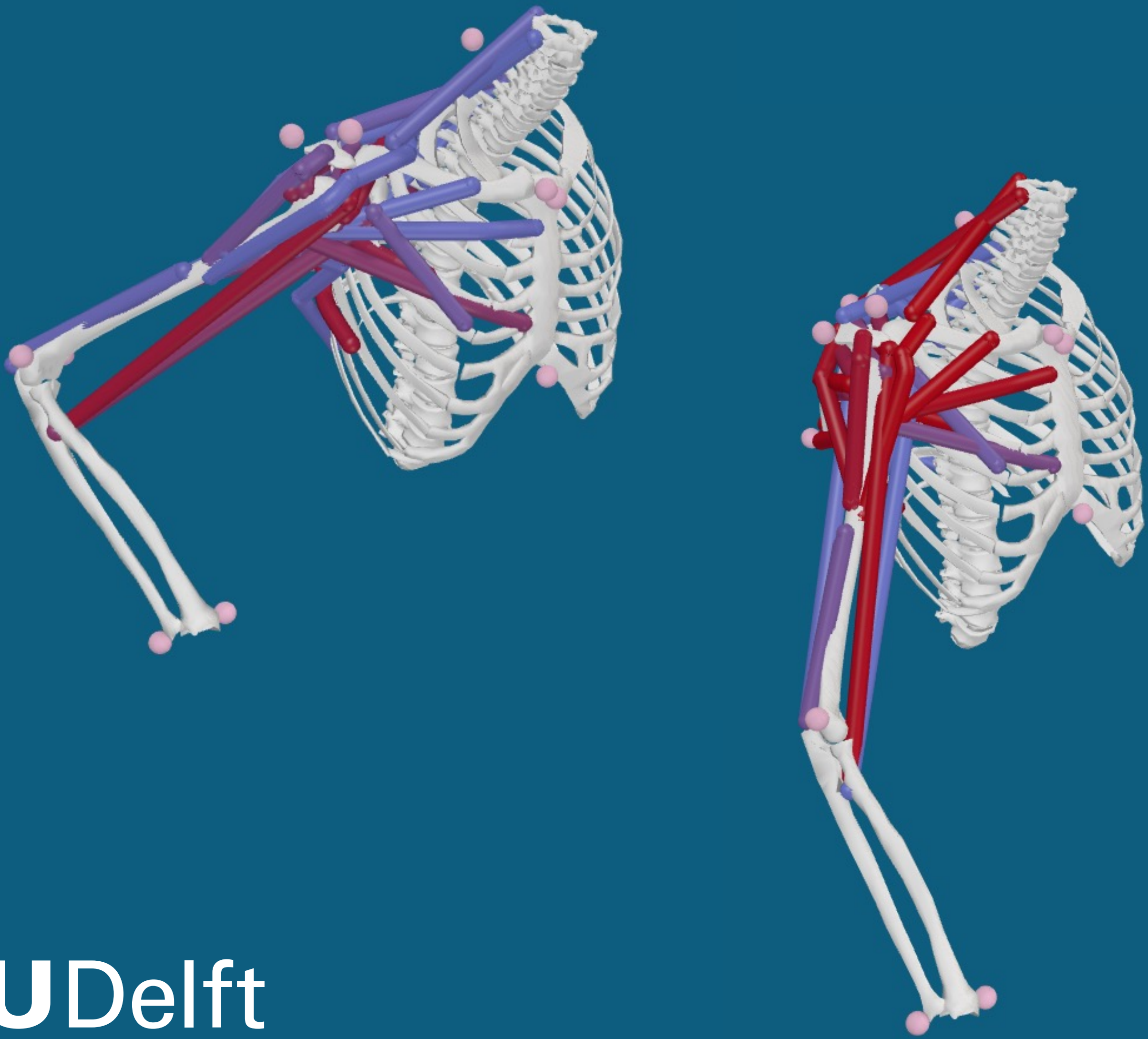
Muscle contributions to work during manual wheelchair propulsion

Master Thesis Biomedical Engineering

Ana Guiomar Cudell Santos Carvalho

Delft University of Technology



TU Delft

Muscle contributions to work during manual wheelchair propulsion

by

Ana Guiomar Cudell Santos Carvalho

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in Biomedical Engineering

Track Neuromusculoskeletal Biomechanics
at the Delft University of Technology,
to be defended publicly on Friday April 19, 2024 at 14:00.

Student number:	5605571	
Supervisors:	Dr. A. Seth,	TU Delft,
	I. Belli,	TU Delft,
	W. de Vries,	Swiss Paraplegic Research

Preface

I am happy and proud to present this master's thesis, which marks the last and most challenging chapter of my journey as a biomedical engineering student. I was lucky to be surrounded by extraordinary people who inspired me both academically and personally throughout this process. I would like to thank you all.

First and foremost, I am grateful to my supervisors, Ajay Seth and Italo Belli. Ajay, thank you for sharing your vast expertise and enthusiasm during this research, and for continuously challenging me to get the most out of this project. Italo, thank you for your invaluable assistance during the simulations and for always being ready to help.

Furthermore, I want to thank the team at the Swiss Paraplegic Research, especially Wiebe de Vries, for providing all the data, but also for always being open to discussions and questions. I am grateful for your enthusiasm for research and continuous support.

Last but not least, I had an incredible support system without which I would have never made it this far. I am grateful to my family who has celebrated my victories and helped with my struggles; even from afar, you have always been present. I am also extremely thankful to my friends, both near and far. You have always been there, listening to my never-ending monologues, giving me advice and encouragement, and ready to create fun memories. My life in Delft would not have been the same heart-warming experience without my incredible housemates, the parties at Jumbo, the 'bombing' dinners, and anniversary brunches. All of you mean a lot to me, *olelelele*!

Ana Guiomar Cudell Santos Carvalho
Delft, April 2024

Abstract

Manual wheelchair users, especially those with spinal cord injuries, often suffer from shoulder overuse and pain. Understanding the specific role of the muscles involved in wheelchair propulsion is essential to prevent these complications. Yet, there is uncertainty about which muscles are primarily recruited, and their contribution to propulsion power. For example, some studies have suggested that the rotator cuff muscles are the primary contributors to wheelchair propulsion. Several studies have explored muscle activation patterns using musculoskeletal models, but often omit glenohumeral stability and key muscles like the trapezius, serratus anterior, and rhomboideus. Investigating muscle work provides insights into the actual mechanical output, offering a more accurate assessment of efficiency by considering factors like muscle force and movement distance.

The aim of this study was to examine individual muscle contributions to mechanical work during the push and recovery phases of wheelchair propulsion using a musculoskeletal model and to identify the role of rotator cuff muscles. The thoracoscapular shoulder model featuring accurate scapula kinematics, inclusive of the rhomboideus, serratus anterior, and trapezius muscles, was employed. In a previous experiment electromyography, kinematic and pushrim forces of 5 persons with paraplegia were collected. The filtered pushrim forces and the kinematic data as well as the scaled musculoskeletal model were inputs to the rapid muscle redundancy solver, that enforced glenohumeral joint stability while estimating muscle activations and muscle power. Muscle work was integrated from the estimated muscle power and normalized by the total mechanical work. During the push phase, significant contributions came from the pectoralis major, anterior deltoid, infraspinatus, serratus anterior, triceps brachii, and biceps brachii. In contrast, the recovery phase primarily involved the teres major, subscapularis, trapezius, posterior deltoid, middle deltoid, and rhomboideus. The contribution of rotator cuff muscles to propulsion was noted but to a lesser extent than previous reports, with no contribution from the supraspinatus. Surprisingly, the teres major showed high work values, possibly due to insufficient activation of the latissimus dorsi. Our findings support the importance of incorporating muscles like the serratus anterior, trapezius, and rhomboids into musculoskeletal models. Furthermore, the contribution of reserve actuators to total work generation was below the threshold of 5%. The comparison between estimated muscle activations and measured electromyographic activations generally showed excellent to good magnitude matching. This study's outcomes can aid in designing ergonomic wheelchairs and guide the development of tailored rehabilitation and training methods to decrease upper extremity stress in wheelchair users.

Contents

List of Figures

List of Tables

Abbreviations

DSEM	Delft Shoulder and Elbow Model
EMG	Electromyography
MAE	Mean absolute error
MVC	Maximum voluntary contraction
RMR	Rapid muscle redundancy

1 Introduction

Studying wheelchair propulsion significantly impacts the quality of life of those who depend on wheelchairs for mobility. Efficient propulsion enables greater independence and access to various activities and environments [1]. Moreover, understanding the biomechanics of wheelchair propulsion is key to preventing injuries related to long-term wheelchair use.

1.1. Importance of understanding muscle contribution

The most common complaint amongst individuals relying on manual wheelchairs for daily mobility is shoulder pain, at 76% of manual wheelchair users [2]. This pain arises from the significant demands placed on the upper extremities by the repetitive motion of wheelchair propulsion, as well as other activities of daily living, such as transferring and weight relief tasks [3]. Consequently, the shoulder complex is at a significant risk of overuse and injury [3, 4]. Further, the pain may prevent users from being physically active and limit their independence and quality of life [4]. Due to their lower limb paralysis, individuals with spinal cord injury are common wheelchair users who suffer from the aforementioned shoulder overuse and its implications [5–7]. Various factors, including the severity and completeness of the injury [8], gender [8], wheelchair type and setup [5], and propulsion biomechanics [5, 9], contribute to the risk of shoulder pain. Injuries at the wrist and shoulder have been linked to higher forces applied on the push rim [10], and individuals who endure higher shoulder forces and moments during wheelchair propulsion exhibit more signs of shoulder pathology [11]. Therefore, understanding which shoulder muscles contribute to manual wheelchair propulsion is crucial to prevent shoulder injuries and overuse.

Identifying the muscles most engaged in propulsion not only aids in designing ergonomic wheelchairs but also in shaping rehabilitation and training programs. Recognizing the individual muscle contributions to propulsion can lead to wheelchair designs that minimize strain on specific muscle groups, thereby preventing overuse, injuries and enhancing propulsion efficiency [12, 13]. Furthermore, identifying muscles prone to overuse allows for the development of targeted rehabilitation and training programs for wheelchair users. Therapists can tailor exercise programs aimed at strengthening these muscles and improving propulsion technique, ultimately reducing the demands on the shoulder [14, 15].

1.1.1. Push and Recovery Phases

A wheelchair propulsion stroke cycle, as illustrated in Fig. 1.1, consists of two distinct phases: the push phase, where the hands are in contact with the pushrim arc, while delivering mechanical power to the wheel, and the recovery phase, where the hands are repositioned in preparation for the upcoming stroke cycle [16].

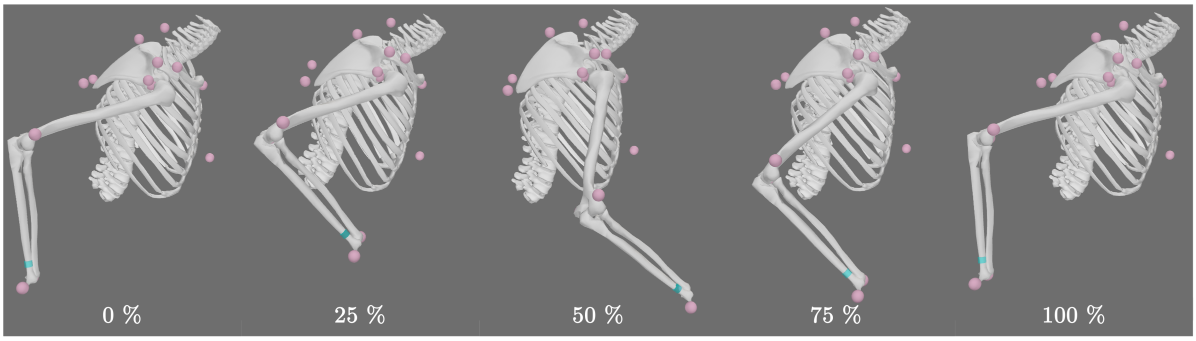


Figure 1.1: The stages of a propulsion cycle: the first half depicts the push phase, and the second half, the recovery phase.

1.2. Previous studies analysing muscle contributions to wheelchair propulsion

The biomechanical contributions of individual muscles to manual wheelchair propulsion on flat terrain have been explored in various studies, although with differing methodologies, participant populations, and diverse measures of muscle contributions (muscle activation, power or work). In the following, an overview of the state of the art is presented.

1.2.1. Push phase

In the push phase, studies utilizing fine-wire electromyography (EMG), such as those by Mulroy et al. [14, 17], and musculoskeletal modeling [15, 18–20] consistently identified the deltoideus anterior and pectoralis major as primary contributors. Early models of wheelchair propulsion [20–24], however, lacked individual specificity, which restricted their application in tailored treatment evaluations [25]. These models, notably the Delft Shoulder and Elbow Model (DSEM), were pivotal in assessing glenohumeral contact forces and muscle forces, thereby highlighting the crucial role of the rotator cuff muscles. Their dual function in stabilizing the glenohumeral joint and potentially generating compensatory moments for the deltoideus muscles was underscored [17, 22–24, 26]. Yet, the rotator cuff muscle's forces were not entirely distinguished between their stabilizing and propulsion roles. Lin et al. employed a computer graphics based musculoskeletal model for muscle force analysis, underlining the rotator cuff muscles' significant contribution to propulsion [19]. Rankin et al. adopted the open-source SIMM Stanford VA Model to investigate upper limb demand and shoulder muscle contributions during wheelchair propulsion [15, 18]. This marked the first analysis using muscle power as a criterion for contribution, revealing substantial power generation by the rotator cuff muscles, in addition to that of the deltoideus anterior and posterior muscles. Similarly, Slowik et al. found significant power generated by the rotator cuff muscles using the SIMM model [27]. However, these models did not account for glenohumeral stability constraints and overlooked muscles, such as the serratus anterior, rhomboideus major, and the upper and middle trapezius [15, 18, 19, 27]. Mulroy et al. previously demonstrated the significance of these deep thoracohumeral muscles in wheelchair propulsion via fine-wire EMG, arguing against their exclusion from models [14, 17]. Addressing this gap, Odle et al. developed a personalized upper body model within OpenSim that included these muscles [7]. Contrary to previous findings, this study identified the rotator cuff muscles and the serratus anterior as the main contributors, rather than the deltoideus and pectoralis major.

The scapula has typically been modeled using one of two kinematic approaches: either by kinematically constraining the scapula to glide along the thoracic surface [15, 18, 20–24], or by defining the orientation of the scapula and clavicle through regression equations based on the angles of the humerus [7, 28]. However, Seth et al. have demonstrated that neither of these two kinematic approaches has adequately modeled scapular kinematics [29].

Further, there is uncertainty about the recruitment of the triceps brachii and biceps brachii. Dubowsky et al. were the first to construct and validate a patient-specific model of wheelchair propulsion. They utilized the AnyBody software and investigated minimizing shoulder joint forces, highlighting the intricacies of the force distribution across the shoulder during propulsion [30]. In their study, they identified the deltoideus and pectoralis major as prime contributors to muscle activity in the push phase as well as the biceps brachii. Similarly, Lin et. al analysed muscle force and measured significant high values of the biceps brachii. The studies from van Drongelen et al. [23] and Vegter et al. [24], both using data from able-bodied participants with the DSEM, analysed high triceps brachii muscle forces. Controversially, Rankin et al. measured high muscle power of the triceps brachii and biceps brachii in the recovery phase of regular wheelchair users [15, 18].

1.2.2. Recovery phase

Rankin et al. identified both the deltoideus posterior and the subscapularis as crucial to the recovery phase [15, 18], a finding paralleled in the work of Lin et al. [19] and Veeger et al. [22], who similarly spotlighted the deltoideus posterior and the supraspinatus as key muscles during the recovery phase, based on their analysis of muscle forces. These findings underscore the pivotal role of rotator cuff muscles not just in the push phase but also throughout the recovery phase, as evidenced in studies assessing both muscle forces [7, 19, 22] and power [15, 18, 27]. Remarkably, Lin et al. are one of the few studies employing musculoskeletal modeling that mention the deltoideus middle as primary contributor [19]. In contrast, a broad spectrum of studies utilizing either fine-wire or surface EMG, or kinetic methodologies, has consistently observed high activation levels in the deltoideus middle and the trapezius during the recovery phase [14, 17, 31, 32].

1.2.3. Muscle power and mechanical work

Muscle activation studies offer insight into the timing and intensity of muscle engagement, but fall short in specifying how this engagement translates into movement and mechanical work. While high levels of muscle activation may suggest extensive muscle utilization, this does not necessarily translate to its effectiveness or efficiency in contributing to the propulsion of the wheelchair [18, 24]. In contrast, analyzing muscle power and mechanical work contributions identifies which muscles actively contribute to propulsion by considering both force generation and movement velocity [18, 24]. Thus, highlighting the efficiency of muscle contributions in terms of mechanical output, thereby providing insight into the dynamics of wheelchair propulsion [18].

1.3. Problem statement

The contributions of individual muscles to manual wheelchair propulsion have been analyzed using various methodologies, yet there remains no consensus on the specific roles of individual muscles, particularly the rotator cuff muscles. This gap in knowledge leads to the following research questions: What are the contributions of individual muscles to the mechanical work during both the push and recovery phases of wheelchair propulsion? What roles do the rotator cuff muscles play? This study aims to address these questions through the use of a musculoskeletal model that incorporated accurate scapula kinematics and included the rhomboideus, serratus anterior, and trapezius muscles. EMG and kinematic data, along with pushrim forces, were collected from five individuals with paraplegia during manual wheelchair propulsion. The model, combined with kinematic and kinetic data, were processed using a rapid muscle redundancy (RMR) solver, which included a glenohumeral joint stability constrain, to estimate individual muscle activity and power. By integrating muscle power over time, the work performed by individual muscles was analyzed.

2 Methods

In a previous experiment EMG, kinematic and kinetic data were collected (Fig. 2.1 left). The filtered force data from the Smartwheel and the kinematic data as well as the scaled musculoskeletal model were set as inputs to the RMR solver. The RMR solver enforced a glenohumeral joint constraint and estimated muscle activations and muscle power. Muscle work was integrated from the estimated muscle power and normalized by the total mechanical work. The results of the RMR solver were verified by looking at the estimated coordinate accelerations and the contributions of the reserve actuators. Lastly, the model was validated by calculating the mean absolute error between measured and estimated muscle activations.

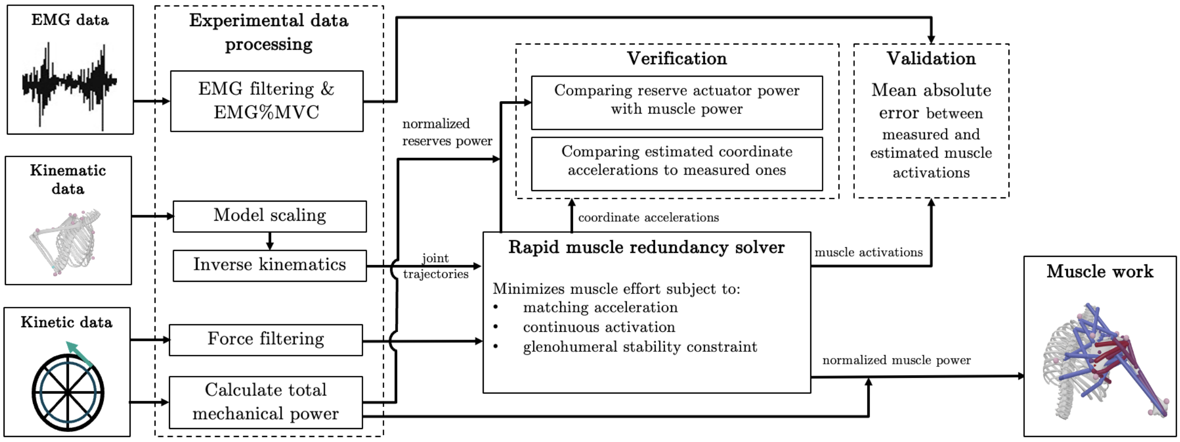

EMG data
Experimental data processing
EMG filtering & EMG%MVC
Kinematic data
Model scaling
Inverse kinematics
Kinetic data
Force filtering
Calculate total mechanical power
normalized reserves power
joint trajectories
Verification
Comparing reserve actuator power with muscle power
Comparing estimated coordinate accelerations to measured ones
coordinate accelerations
Validation
Mean absolute error between measured and estimated muscle activations
Rapid muscle redundancy solver
Minimizes muscle effort subject to:
• matching acceleration
• continuous activation
• glenohumeral stability constraint
muscle activations
normalized muscle power
Muscle work

Figure 2.1: Overview of the methodology to estimate muscle work. EMG, kinematic and kinetic data were collected in a previous experiment. The filtered force data from the Smartwheel and the kinematic data as well as the scaled musculoskeletal model were set as inputs to the RMR solver. The RMR solver enforced a glenohumeral joint constraint and estimated muscle activations and muscle power. The resulting powers were normalized by the output power from the experiment and muscle work was calculated. The results of the RMR solver were verified by looking at the estimated coordinate accelerations and the contributions of the reserve actuators. Lastly, the model was validated by calculating the mean absolute error between measured and estimated muscle activations.

2.1. Experimental data collection

A prior study analyzed a population-based sample of wheelchair users with spinal cord injuries, comprising 34 individuals (average age: 50.8 ± 9.7 years, 82% male) [9]. Neuromuscular activation and propulsion biomechanics were assessed on a treadmill at fixed power outputs of 25 W and 45 W, both before and after a fatigue-inducing protocol [9]. For the purposes of this paper, we utilized data from five subjects performing exercises at 45 W prior to fatigue induction. Detailed data collection methodologies are extensively outlined in previous publications [9] and summarized below.

Upon receiving informed consent, participants propelled their personal wheelchairs on a treadmill at a speed of 1.11 m/s for a duration of 40 seconds, targeting a predetermined power output. Kinematic and EMG data were captured during the final 30 seconds of propulsion. Handrim kinetics, including forces and moments across six degrees of freedom, were measured at 240 Hz using a SmartWheel (24 inches; Three Rivers Holdings, Inc., Mesa, AZ) affixed to the non-dominant side of each participant's wheelchair. Upper body kinematics were recorded at 100 Hz using an eight-camera marker-based motion capture system (Oqus, Qualisys AB, Gothenburg, Sweden), adhering to the protocol established by Wu et al. [33]. EMG signals for the biceps brachii, pectoralis major pars sternalis, deltoideus pars acromialis, and both the lower and upper trapezius were collected using a wireless system (Telemyo 2400T DTS; Noraxon, Inc., Scottsdale, AZ) equipped with surface electrodes.

To assess maximum voluntary contractions (MVCs), participants performed four predefined tests, securely strapped in a seated position to reduce variability arising from limited trunk control. The sequence of MVC tests was randomized to prevent bias.

2.2. Musculoskeletal model

The thoracoscapular shoulder model [29], utilized within OpenSim [34], features a scapula with 4 degrees of freedom relative to the thorax and models the glenohumeral joint as a gimbal joint with 3 degrees of freedom. This results in a comprehensive model featuring 7 degrees of freedom, actuated by 35 muscle-tendon elements derived from the DSEM [21] and producing moment arms bounded by measurements from cadaver experiments [35]. Given the model's initial configuration with a fixed elbow, this coordinate has been unlocked and the muscle properties of both the triceps brachii and biceps brachii were fine-tuned. Notably, the medial and lateral heads of the triceps brachii were incorporated into this model. Adjustments to the attachment points for the triceps brachii at the ulna and biceps brachii at the radius were made to align with Holzbauer et al. [36], by presenting similar moment arms, ensuring the representation of accurate joint moments. The model's muscle properties, including maximal isometric force, optimal fiber length, tendon slack length, and pennation angle, were aligned with those in the DSEM [21]. This model was used as a base model (Fig. 2.2) and scaled individually for each participant.

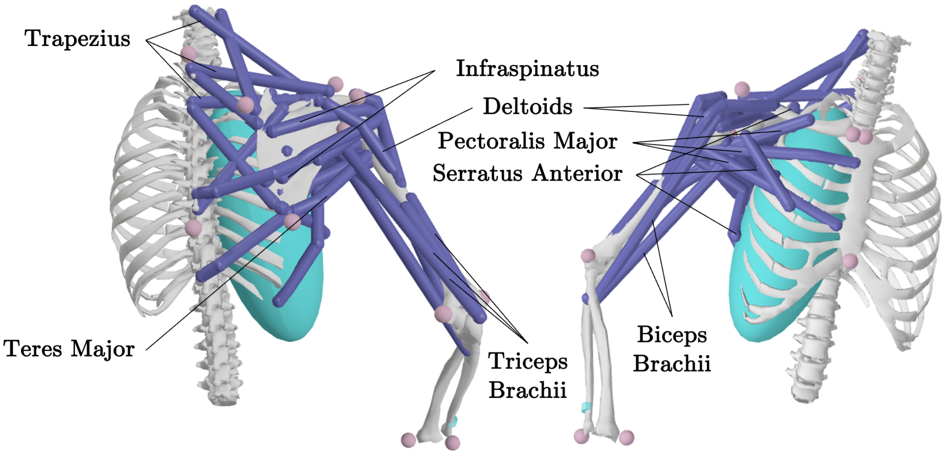


Figure 2.2: Musculoskeletal model in the initial position with muscles mentioned in the report.

Using OpenSim's scaling tool, the model's rigid bodies and related points were scaled linearly based on marker-based distances from corresponding anatomical landmarks [34]. This process continued until the average marker error was reduced to approximately 2 cm. Optimal muscle fibre and tendon slack lengths were also scaled to preserve their proportional relationship to muscle path length. The scapulothoracic joint's ellipsoid surface was refined by optimizing its tilt and radii to reduce marker tracking errors [29]. The scale factors and ellipsoid parameters for the scapulothoracic joint were optimized to enhance the fit to experimental marker data during inverse kinematics, aiming for a root-mean-squared error below 1 cm. In cases where musculotendon properties exceeded optimal ranges post-scaling, appropriate adjustments were made to the optimal fiber length and/or tendon slack length [37]. This entailed calculating normalized fiber lengths and modifying the optimal fiber length and/or tendon slack length until the normalized fiber lengths approached 1 throughout the motion.

Muscle power and activation were estimated using a RMR solver, which incorporates constraints on joint reaction forces from a musculoskeletal model [38]. Given the scaled model, experimental motion and external forces, the RMR solver determines muscle activations and joint forces by minimising the weighted sum of square activations while matching the experimental motion [38]. In addition, the RMR solver enforces glenohumeral joint stability, accounting for muscle contributions to both propulsive motion and glenohumeral joint stabilization.

Forces exerted on the pushrim during the push phase were processed using through a low-pass 15 Hz fourth-order Butterworth filter, preparing them as input for the RMR solver. Moreover, three markers were placed on the model at the glenoid cavity and the centre of the humeral head, to enable the glenohumeral joint constraint. Reserve actuators were integrated at each joint coordinate, guaranteeing precise adherence to experimental joint trajectories [38]. To minimize the influence of reserve actuators, their weights were finely adjusted for each subject. As the RMR solver originally only estimated muscle and reserve actuator activations, it was expanded to also compute muscle and reserve actuator power. This was implemented by first setting the optimized muscle and reserve activations, and then updating the model accelerations. Subsequently, muscle and reserve powers were extracted from the model and designated as outputs (see appendix A for the full code). Consequently, the RMR solver was able to calculate the most efficient muscle activations and corresponding powers, along with the powers exerted by the reserve actuators.

2.3. Data analysis

Individual subject stroke data, including muscle activation, muscle power, and power exerted by the reserve actuators, were normalized to 101 time points of the propulsion cycle, with the first 50 points allocated to the push phase and the latter 51 points to the recovery phase. Each point represented one percent (0 to 100 %) of the entire stroke, achieved through cubic-spline interpolation. For the analysis the mid 10 stroke cycles of each participant were selected. This data was averaged across the 10 strokes and standard deviations were computed to assess intra-subject variability.

2.3.1. Muscle work

For comparative analysis, the power generated by the muscles and reserve actuators was normalized to the participant's total power output. Total power output was calculated by multiplying angular velocity and torque measured by the SmartWheel for each time step. Muscle activations and normalized powers were then averaged across participants, and standard deviations were computed to assess inter-subject variability. To analyse the positive and negative work performed by each muscle, the area under the original muscle power curve was computed. Likewise, the total mechanical work was derived from the total power output. The total mechanical work represents the external work derived from the SmartWheel measurements. Subsequently, individual muscle work was normalized by the total mechanical work for each participant, and the mean with its standard deviation across all participants was reported. This offers a visual representation of each muscle's contribution during the push and recovery phases. Moreover, the sum of negative and positive muscle work during the push phase was compiled for subsequent comparison with the total mechanical work.

2.3.2. Verification

To verify the solutions produced by the RMR solver, both measured and simulated coordinate accelerations were analyzed. Additionally, the contribution of the reserve actuators to propulsion was quantified. The positive and negative work of the reserve actuators was determined by calculating the area under the original power curve of the reserve actuators. These work estimates were then divided by the total work from the model, which consisted of the sum of work from both the reserve actuators and the muscles. The reserve actuator's positive and negative contribution, excluding those of thorax and forearm, to the push and recovery phase were tabulated. Additionally, the sum of negative and positive of the negative and positive work for reserve actuators during the push phase was compiled for subsequent comparison with the total mechanical work.

2.3.3. Validation

Moreover, EMG signals were offset corrected, rectified, and filtered using both a high-pass (20 Hz) and low-pass (3 Hz) third-order Butterworth filter [9]. EMG readings were presented as a percentage of the maximum voluntary contraction (EMG%MVC). The maximum voluntary contractions for each muscle were established by averaging the peak force from each repetition and selecting the highest force recorded across the four distinct MVC tests [9].

The mean absolute error (MAE), previously used to quantitatively validate computational musculoskeletal models [7, 30, 39], was calculated for each muscle's activity by using the following equation:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |MA_i - EA_i|$$

where n presents the number of frames in a propulsion cycle, MA_i is the measured EMG muscle activity as a percentage of maximum voluntary contraction at frame i, and EA_i is the computed muscle activity in frame i. For biceps brachii, the average activation of the two muscle bundles in the model was used in the comparison.

3 Results

To estimate the work of individual muscles during wheelchair propulsion, the model was scaled to match the participant's dimensions. Following inverse kinematics, the root-mean-squared error between the virtual markers on the model and the experimental markers was calculated for each frame using inverse kinematics. The root-mean-squared error across all participants for the entire motion was less than 1 cm, aligning with the maximum allowable error for soft tissue artifact, set at 1 cm by Chiari et al. [40]. Adjustments were made to the muscle properties of the triceps brachii and biceps brachii to ensure their normalized fiber lengths were close to 1 throughout the motion. Detailed values of these adjustments are provided in table C.1.

3.1. Muscle work

The RMR solver estimated muscle activations for each subject, from which individual muscle powers were calculated. The normalized muscle work for the push and recovery phases is displayed in Fig. 3.1 and 3.2, ranging from 0 to 1, with 1 representing the total mechanical work.

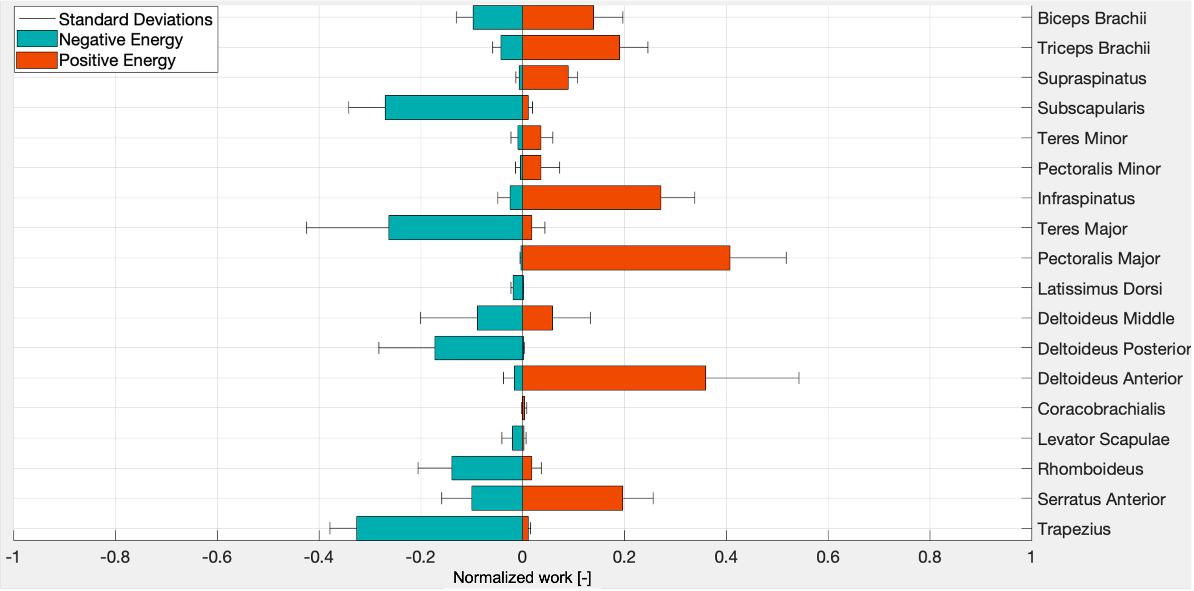


Figure 3.1: Mean normalized muscle work during the push phase with standard deviation. The orange bars represent the positive work, while the blue bars represent the negative muscle work.

During the push phase, the results indicate that certain muscles exhibit higher positive work output compared to others. Excluded were muscles performing under 3% of the total muscle work in the push and recovery phase, respectively [41]. The muscles that generated the highest average normalized work

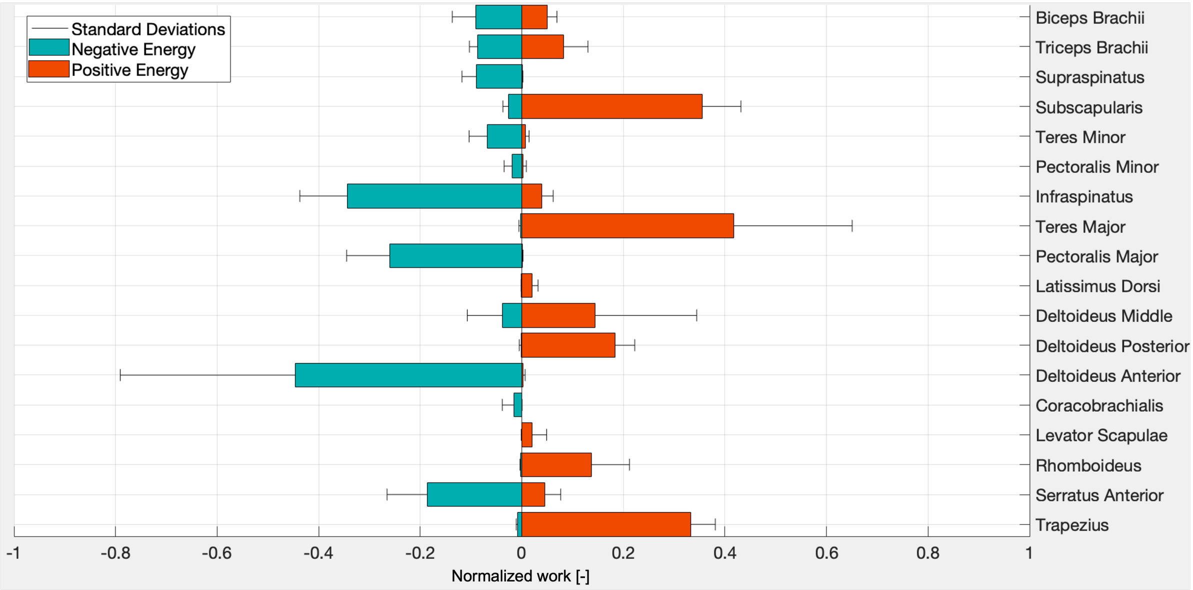


Figure 3.2: Mean normalized muscle work during the push phase with standard deviation. The orange bars represent the positive work, while the blue bars represent the negative muscle work.

during the push phase were the pectoralis major (0.407 ± 0.067), deltoideus anterior (0.361 ± 0.183), infraspinatus (0.272 ± 0.067), serratus anterior (0.197 ± 0.059), triceps brachii (0.191 ± 0.055), biceps brachii (0.139 ± 0.005), supraspinatus (0.090 ± 0.018), and deltoideus middle (0.059 ± 0.075). When examining the mean normalized power outputs (refer to Fig.3.3), it is evident that the biceps brachii long and brevis generate power in the initial phase, followed by the serratus anterior and infraspinatus. The deltoideus clavicle anterior and the pectoralis major contribute towards the end of the push phase.

In contrast, the muscle generating the mostnegative work during the push phase were the trapezius (-0.325 ± 0.053), subscapularis (-0.269 ± 0.072), teres major (-0.263 ± 0.162), deltoideus posterior (-0.172 ± 0.110 J), rhomboideus (-0.139 ± 0.66), serratus anterior (-0.100 ± 0.059), and biceps brachii (-0.097 ± 0.033).

Regarding the recovery phase, the muscles that generated the greatest average normalized work were the teres major (0.417 ± 0.233), subscapularis (0.356 ± 0.076), trapezius (0.332 ± 0.049), deltoideus posterior (0.184 ± 0.038), deltoideus middle (0.145 ± 0.200), rhomboideus (0.137 ± 0.075), and triceps brachii (0.082 ± 0.048). When examining the mean normalized power outputs (refer to Fig. 3.3), it is shown that the deltoideus scapula posterior and trapezius generate power during the initial phase of the recovery, followed by the teres major, subscapularis, rhomboideus and triceps brachii. The deltoideus middle generated power towards the end of the recovery phase.

In contrast, the muscles absorbing larger work during the recovery phase were the deltoideus anterior (-0.046 ± 0.345), infraspinatus (-0.344 ± 0.093), serratus anterior (-0.186 ± 0.079), biceps brachii (-0.091 ± 0.046), triceps brachii (-0.087 ± 0.016), supraspinatus (-0.089 ± 0.029), and teres minor (-0.067 ± 0.036).

The total work of the muscles during the push phase was consistently smaller than the total external work. For example for participant 12 the total push muscle work of 498.1 J fell bellow the total external work of 761.2 J and the total work of the thorax reserve actuators was 389.9 J. Suggesting that the thorax muscles also contribute to propulsion.

3.2. Verification

To verify that all system work is accounted for by muscles, the work by reserve and residual actuators are reported. The muscle force estimates produced accelerations that matched accelerations from experimental kinematics, achieving average accelerations for transitional and rotational coordinates of 0.001 m/s^2 and 0.067 deg/s^2, respectively. These values fell comfortably within one standard deviation of the experimental data (5.297 m/s^2 and 0.350 deg/s^2, Table C.2). Such tracking errors were deemed acceptable, as determined by Hicks et al. [42].

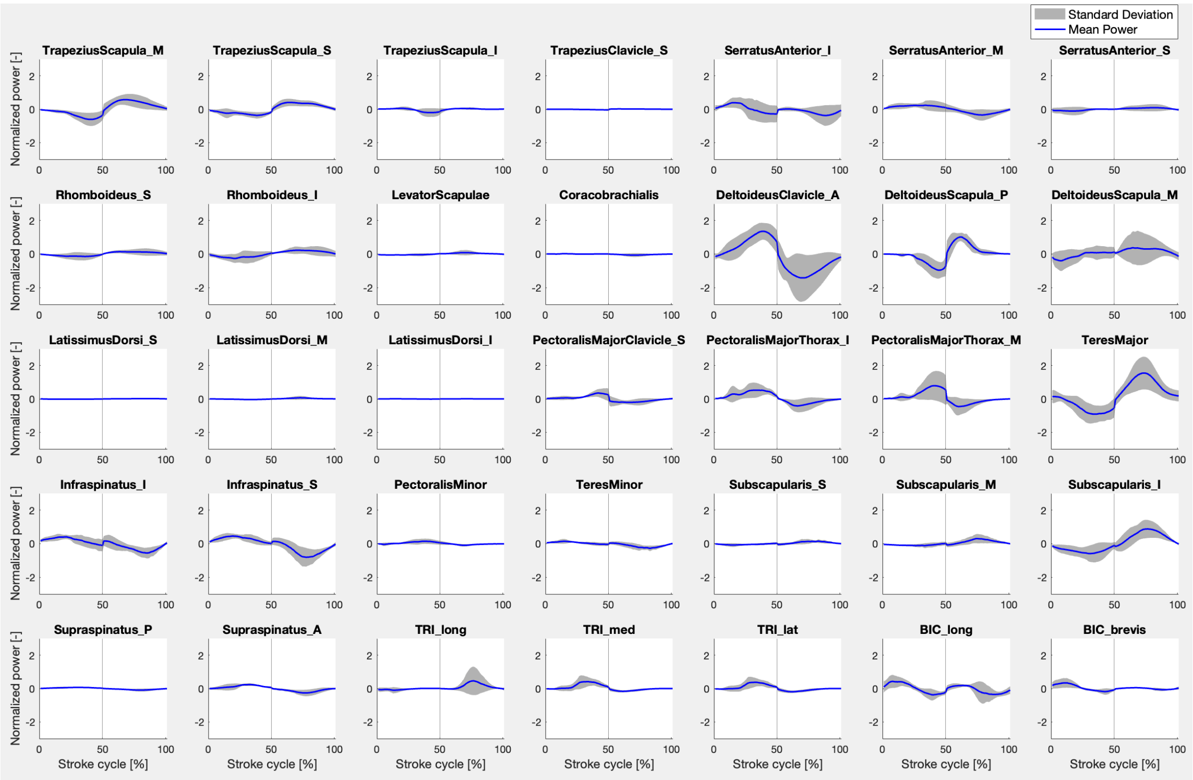


Figure 3.3: Averaged normalized muscle power across participants throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

Table 3.1: The negative and positive contribution of the reserve actuators, excluding the thorax and forearm actuators, in percentage to the push and recovery phase for each participant. Participant order comes from the experiment [9].

		Push Phase	**Recovery Phase**
Positive Contribution [%]	**07**	0.04	0.34
	08	0.15	0.05
	10	11.47	3.89
	12	0.14	0.11
	25	3.41	3.41
Negative Contribution [%]	**07**	0.03	0.04
	08	0.01	0.01
	10	5.64	5.16
	12	0.13	0.13
	25	0.08	0.34

Furthermore, it was crucial for the optimizations of the RMR solver to ensure that the muscles, rather than the reserve actuators, primarily drove the model's actions. The reserve actuators linked with the thorax could exhibit higher values, as they prevent the model from collapsing, essentially providing support for the lower body. Notably, the model did not encompass the brachioradialis and brachialis muscles, which aid in elbow flexion, nor did it include the forearm muscles. Thus, these coordinates were expectedly actuated by the reserve actuators. The positive and negative contributions of the reserve actuators, excluding the thorax and forearm coordinates, to both the push and recovery phases are detailed in table 3.1. Generally, the contributions remained significantly below 5%. These reserve actuator contributions were deemed acceptable, falling below the threshold of 5% established by Hicks et al. [42]. However, it is worth noting that the values for participant 10 exceeded this threshold.

3.3. Validation

The MAE was calculated for each muscle during each stroke cycle, and Table 3.2 reports the average MAE across all cycles. An average MAE below 0.10 indicated excellent magnitude matching, between 0.10 and 0.20 suggested good magnitude matching, and above 0.20 indicated poor magnitude matching [7, 30]. Muscles exhibiting an excellent magnitude matching, with a mean MAE across all participants below 0.01, included the upper trapezius and biceps brachii. The deltoideus, lower trapezius, and pectoralis major demonstrated good magnitude matching (shown in Table 3.2 as light grey shaded). However, for participant 07 the magnitude matching for the lower trapezius and pectoralis major was poor, with MAE values of 0.227 and 0.212, respectively. Participant 25 also showed poor magnitude matching for the lower trapezius (0.210) and notably poor for the pectoralis major (0.437) (shown in Table 3.2 as dark grey shaded).

Table 3.2: Average mean absolute errors and their standard deviations between computed and measured muscle activations across all stroke cycles for each participant. The final row presents the mean of each muscle's absolute error across all participants, along with their standard deviations. The light grey shaded areas show good magnitude matching of the mean MAE, and the dark grey shaded areas show bad magnitude matching.

	Deltoideus	Upper Trapezius	Lower Trapezius	Biceps Brachii	Pectoralis Major
07	0.061 ± 0.007	0.169 ± 0.019	0.227 ± 0.025	0.039 ± 0.008	0.212 ± 0.047
08	0.036 ± 0.004	0.060 ± 0.007	0.023 ± 0.002	0.086 ± 0.012	0.044 ± 0.005
10	0.101 ± 0.018	0.102 ± 0.003	0.048 ± 0.005	0.053 ± 0.014	0.116 ± 0.026
12	0.144 ± 0.018	0.020 ± 0.003	0.040 ± 0.008	0.022 ± 0.002	0.170 ± 0.017
25	0.179 ± 0.028	0.043 ± 0.006	0.210 ± 0.145	0.060 ± 0.010	0.437 ± 0.054
Mean	0.104 ± 0.058	0.078 ± 0.059	0.110 ± 0.100	0.052 ± 0.024	0.196 ± 0.149

When comparing the computed muscle activations with the EMG activations, it is evident that the upper trapezius and biceps brachii exhibit similar patterns. For instance, the magnitude matching for participant 08 were excellent and their EMG%MVC activations and computed muscle activations are illustrated in Fig. 3.4. In contrast, the deltoideus (posterior), lower trapezius, and especially the pectoralis major (thorax middle) exhibit rather divergent patterns. This divergence is particularly noticeable when comparing the EMG%MVC activations and computed muscle activations of participant 25, who displayed poor magnitude matching (Fig. 3.5). The experimental data indicated that the pectoralis major (thorax middle) activation peaked during the initial push phase and then decreased drastically in the recovery phase. In contrast, the modeled activation peaked towards the end of the push phase and remained high at the beginning of the recovery phase. Regarding the deltoideus (posterior), the measured activation peaked at the end of the push phase or the beginning of the recovery phase and remained high, whereas the computed muscle activation peaked during the recovery phase.

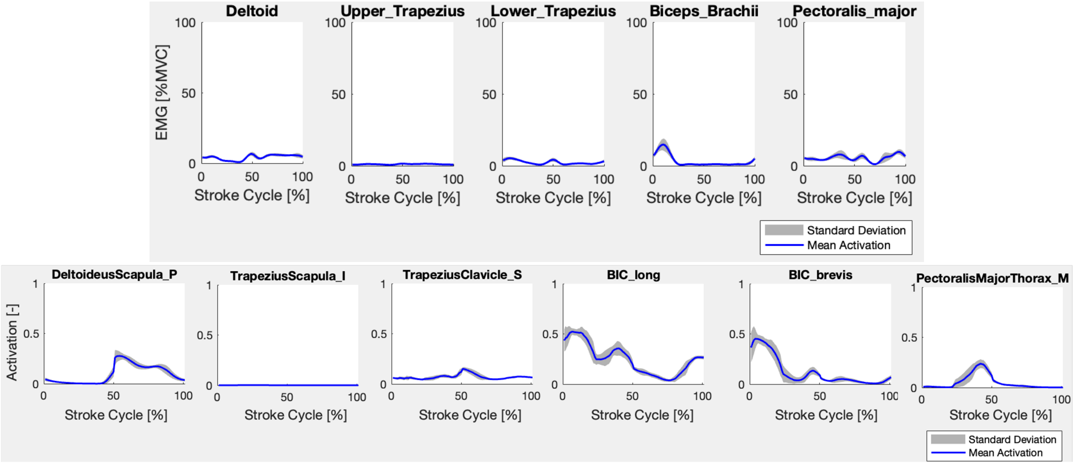


Figure 3.4: Average EMG%MVC (top) and muscle activations from the model of participant 08 (bottom), throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

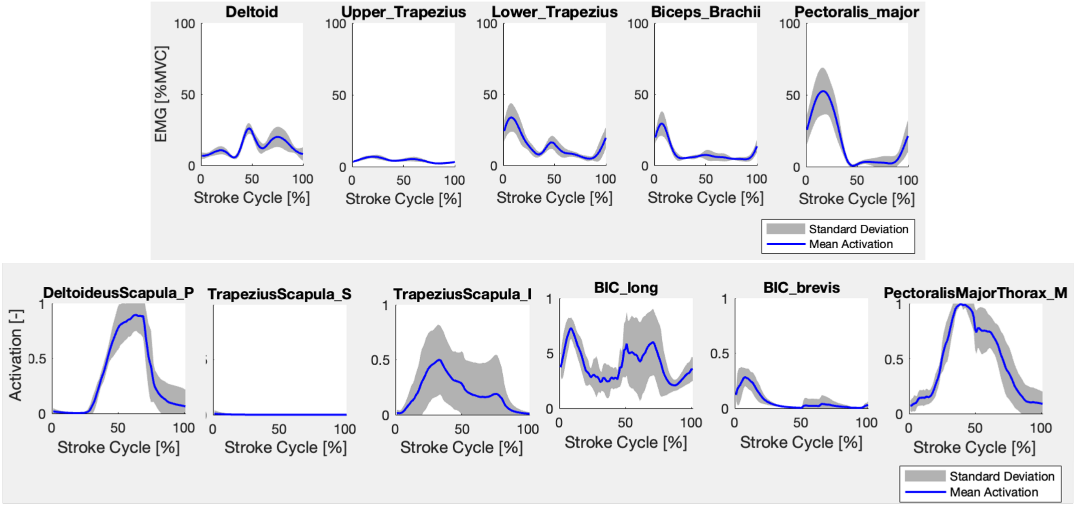


Figure 3.5: Average EMG%MVC (top) and muscle activations from the model of participant 25 (bottom), throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

4 Discussion

Two research questions were posed; What are the contributions of individual muscles to the mechanical work during both the push and recovery phases of wheelchair propulsion? What roles do the rotator cuff muscles play? The aim of this study was to answer these questions through the use of the thoracoscapular shoulder model and experimentally collected data. Both the model and the data served as inputs to the modified RMR solver to estimated individual muscle work for the push and recovery phase.

4.1. Muscle work during the push phase

The pectoralis major and deltoideus anterior are clearly contributing to manual wheelchair propulsion generating high work to shoulder flexion in the push. The pectoralis major also contributes to shoulder adduction. These results are comparable to previous studies utilizing various methodologies including musculoskeletal modeling for muscle power measurement [15, 18], studies using the DSEM [20] and studies using fine-wire EMG for muscle activation measurement [14]. The infraspinatus also emerged as a significant contributor by yielding external rotation. As the arm extends and pushes against the handrim, the infraspinatus works to maintain this external rotation. Studies utilizing OpenSim for muscle force measurement [7, 19], muscle power analyses [15, 18], and fine-wire EMG [14] came to similar findings. The serratus anterior contributed significantly to mechanical work, as seen in studies incorporating fine-wire EMG [14] and one study using a musculoskeletal model [7]. The serratus anterior contributes to the propulsive efforts by protracting the scapula, enabling the shoulder to reach forward effectively. This finding underlines the necessity of incorporating this muscle into musculoskeletal models to fully capture the mechanics of wheelchair propulsion. The biceps brachii contributes to the initial push to elbow flexion and sequentially triceps brachii generates work for elbow extension at the end of the push. Synergies between these muscles are recorded in studies employing fine-wire EMG [14] and modeling [19, 30]. These muscles contributed less to mechanical work than the other prime movers, thus it is understandable that they generate confusion between studies to whether they contribute to propulsion or not.

4.2. Muscle work during the recovery phase

The teres major contributed the most to mechanical work during the recovery phase, a muscle not commonly mentioned in the context of wheelchair propulsion. It produces internal rotation and arm adduction. This high contribution to work possibly originates from the lack of activation and work generation observed in another muscle with a similar function, the latissimus dorsi, hinting at a compensatory mechanism. As mentioned in previous studies, the recovery muscles perform two critical functions: decelerating the arm during follow-through and lifting the arm during its return [14]. The subscapularis, along with the teres major, provides internal rotation, while the trapezius—due to its high work production—and the rhomboideus function to decelerate the scapula during follow-through and to retract the scapula during arm return. This underscores the importance of including these muscles in musculoskeletal modeling analyses of wheelchair propulsion. Additionally, the middle and posterior deltoids are considered prime contributors to the recovery phase by facilitating necessary arm elevation. Studies employing musculoskeletal modeling have observed high muscle power and forces in the posterior deltoid [15, 18, 19], and sometimes even throughout the entire cycle [22]. Conversely, the middle deltoid is predominantly mentioned in studies analyzing EMG data [14].

4.3. Role of the rotator cuff muscles

Previous studies have identified the crucial role of the rotator cuff muscles in manual wheelchair propulsion [20, 21]. These muscles not only stabilize the glenohumeral joint but also contribute to wheelchair propulsion. However, only studies using the DSEM have analyzed the forces generated by this muscle group separately [21–23]. In this study, we enforce a glenohumeral joint constraint, focusing on the work generated by the rotator cuff specifically for propelling the wheelchair.

As previously mentioned, the infraspinatus plays a crucial role during the push phase, and the subscapularis during the recovery phase. Studies analyzing muscle force [7], muscle force relative to maximum capacity [22, 23], or muscle power [15, 18] have reported significantly higher values for the infraspinatus compared to the pectoralis major during the push phase. Nevertheless, these studies report higher values. Thus, the infraspinatus can be seen as a prime contributor to propulsion during the push phase, yet its contribution is less than previously reported.

The infraspinatus, responsible for external rotation of the arm, works in conjunction with the subscapularis, which provides internal rotation, aiding in returning the arm to the starting position. The subscapularis contributes significantly to work during the recovery phase, surpassed only by the teres major, a finding supported by other studies analyzing muscle power [15, 18] and muscle forces [7].

Despite previous research highlighting the supraspinatus as a significant contributor to both the push and recovery phases [14, 19, 22], this study found that this muscle did not produce high muscle work, possibly due to the glenohumeral joint constraint imposed. This observation underscores the supraspinatus' role as a joint stabilizer but raises questions about its direct contribution to propulsion.

4.4. Latissimus dorsi's recruitment

Although other studies do not consider the latissimus dorsi a prime contributor, it is still activated during wheelchair propulsion, assisting in arm adduction and internal rotation, and is expected to be recruited during the recovery phase. An examination of its moment arms during shoulder elevation reveals deviations from the expected values, as determined by Ackland et al. [35]. Specifically, when plotting the moment arm against shoulder elevation for shoulder abduction (see Fig. 4.1a), the middle section of the latissimus dorsi in our generic model exhibits a curve shape similar to that in Ackland et al. [35], with the lowest point at approximately -33 mm at a shoulder elevation angle of 65 deg. In contrast, the scaled model's lowest point occurs at -38 mm at a shoulder elevation of 26 deg. More critical to wheelchair propulsion, for shoulder elevation values between 18 and 45 deg, the moment arms are larger than those in the generic model and Ackland et al. [35]. The scaled model of participant 12 is used as an example. With the model positioned at the beginning of the recovery phase, with an extended arm (see Fig. 4.1b), the scaled model exhibits larger depression moment arms than the generic model for shoulder elevations between 18 and 45 deg, indicating inaccuracies in the scaled model's moment arms. The significant increase in depression moment arms during both shoulder abduction and the recovery phase position indicates that the muscle may not be accurately scaled; however, an increase in the moment arm should signify that its recruitment is favored. Previous studies using the thoracoscapular shoulder model, also reported inaccurate latissimus dorsi recruitment [38, 41]. This suggests that there might be a general issue with this model regarding the latissimus dorsi.

4.5. Verification

For all participants, the reserve actuators contribute minimally to propulsion, with the exception of participant 10. For this participant, the reserve actuator showing the highest values is the scapula abduction actuator, closely followed by the scapula elevation actuator, as depicted in Fig. D.3. These actuators, along with other scapula actuators, reach their peak at the end of the push phase or at the beginning of the recovery phase, indicating the challenges in stabilizing the scapula when it is most abducted. Thus, replicating the scapula's movement through muscle recruitment alone may present difficulties for this subject. Nonetheless, the average contribution of the reserve actuators across all participants remained significantly below the threshold of 5 %.

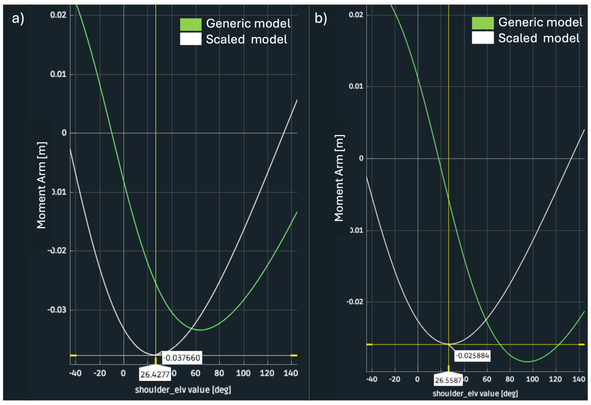


Figure 4.1: Moment Arm of the latissimus dorsi middle against shoulder elevation of the generic model and the scaled model of participant 12 a) for shoulder abduction; and b) with the model positioned at the beginning of the recovery phase, with an extended arm. Critical to wheelchair propulsion are values for shoulder elevation between 18 and 45 deg.

4.6. Validation

Most participants demonstrated excellent to good levels of magnitude matching. However, while participant 25 showed excellent and good magnitude matching for the deltoid, lower trapezius, and biceps brachii, the upper trapezius and, notably, the pectoralis major, with a MAE of 0.4, exhibited poor magnitude matching. The measured and estimated muscle activations not only peak at different times but also show variation in their timing, as illustrated in Fig. 3.5. Several hypotheses could explain this discrepancy. One possibility is the misplacement of EMG electrodes, particularly challenging for accurately targeting the individual pectoralis major pars sternalis, especially in female participants. Additionally, the electrodes might have shifted during the experiment. It is also important to consider that the muscle strength in the model is not customized to the individual, meaning a weaker muscle could appear overactivated in someone with stronger muscles.

4.7. Application

Analyzing muscle activations does not fully capture the effort muscles exert during wheelchair propulsion. Investigating muscle work provides insights into the actual mechanical output, offering a more accurate assessment of efficiency by considering factors like muscle force and movement distance. Understanding the contributions of specific muscles in regular wheelchair users can lead to optimized techniques, enhancing efficiency and performance. Proper technique not only reduces fatigue but also prevents overuse injuries by distributing the workload evenly across muscles, minimizing injury risks. Our findings suggest targeted rehabilitation strategies could be developed to strengthen the muscles most engaged during the push and recovery phases of wheelchair propulsion, specifically tailored for individuals with paraplegia. By focusing on enhancing the strength and endurance of the pectoralis major, anterior deltoid, infraspinatus, but also of the serratus anterior, rehabilitation programs can potentially reduce the risk of shoulder pain and improve propulsion efficiency. Similarly, understanding the significant roles of the teres major and subscapularis in the recovery phase can guide the development of recovery-specific exercises that balance muscle function and support joint health. Additionally, wheelchair design and setup can be tailored to meet the biomechanical demands of various muscle groups during propulsion, enhancing comfort and efficiency. Adjustments to wheelchair features such as wheel placement, seat height, and back support could be made to reduce the demand on heavily utilized muscles and to facilitate the engagement of underutilized muscles. This study establishes a baseline of the primary muscles used by regular wheelchair users, serving as a reference for future research investigating how muscle contributions change due to factors such as neural drive loss from spinal cord injuries or muscle weakness from aging.

4.8. Limitations

While the results are promising, this study has several limitations. Firstly, the small number of participants, consisting only of regular manual wheelchair users, restricts our ability to generalize the findings to the broader population. Secondly, the omission of the brachioradialis and brachialis muscles in the model limits a full understanding of elbow flexion. Slowik et al. [28] identified the brachioradialis and brachialis as prime contributors, thus including these muscles in the model might impact the contribution of the biceps brachii, another elbow flexor. Nonetheless, the contribution of these muscles is accounted for by the reserve actuator at the elbow. Additionally, the latissimus dorsi muscle is not activated, and its moment arms are inaccurately modeled. Previous studies using the thoracoscapular model also reported inaccurate recruitment of the latissimus dorsi [38, 41]. Future research should explore the muscle path or architecture of the latissimus dorsi to improve its recruitment and consider collecting EMG data to verify its activation during manual wheelchair propulsion. Moreover, the triceps brachii muscles have not been validated. While the long head of the triceps brachii was included, changes to its attachment point on the ulna and muscle properties, as well as the full addition of the lateral and medial heads, necessitate validation through comparison with EMG data, similar to the process for the biceps brachii. Lastly, scaling the model posed challenges, as it is not a commonly reported procedure in studies, depends heavily on the user's experience, and is time-consuming. For example, the optimal fiber length and tendon slack length of the triceps brachii and biceps brachii muscles had to be individually adjusted. Although the scaling process incorporated an automated step by optimizing scale factors and ellipsoid parameters of the scapulothoracic joint, a previously scaled model from the OpenSim GUI with average marker errors below 2 cm must be inputted, and the output of these optimizations must be meticulously reviewed for physiological accuracy. Further work is needed to fully automate the scaling of the scapulothoracic joint, including the adjustment of associated muscle paths and parameters.

5 Conclusion

Individual muscle work was estimated for manual wheelchair propulsion through a musculoskeletal model by analyzing experimental movement data from individuals with paraplegia. During the push phase, the largest contributions came from the pectoralis major, anterior deltoid, infraspinatus, serratus anterior, triceps brachii, and biceps brachii. In contrast, the recovery phase primarily involved the teres major, subscapularis, trapezius, posterior deltoid, middle deltoid, and rhomboideus. Our findings highlight the role the serratus anterior, trapezius, and rhomboids, previously excluded in modeling studies of manual wheelchair propulsion. The contribution of rotator cuff muscles to propulsion was lower than reported in previous studies, with no contribution from the supraspinatus. Surprisingly, the teres major contributed significantly to mechanical work, possibly due to insufficient activation of the latissimus dorsi. We recognize significant limitations, such as a small sample size which limit the generalizability of the findings and inaccurately capturing the latissimus dorsi.

Our research highlights the potential for developing targeted rehabilitation strategies to strengthen key muscles used during the push and recovery phases of wheelchair propulsion for individuals with paraplegia. By improving the strength and endurance of muscles like the pectoralis major, anterior deltoid, infraspinatus, and serratus anterior, these programs could reduce shoulder pain and enhance propulsion efficiency. Additionally, our findings support the design of recovery-phase specific exercises focusing on the teres major, subscapularis and trapezius to balance muscle function. Regarding wheelchair design, our study suggests ergonomic optimizations could help distribute the muscular load more effectively. Modifications to wheel placement, seat height, and back support could alleviate strain on overused muscles and help engage underutilized ones. Furthermore, our study provides a baseline regarding muscle usage in regular wheelchair users, offering a reference point for future research into how muscle contributions are affected by factors like neural drive loss or aging.

References

[1] O. Fliess-Douer, Y. C. Vanlandewijck, and L. H. Van Der Woude. “Most essential wheeled mobility skills for daily life: an international survey among paralympic wheelchair athletes with spinal cord injury”. In: *Archives of physical medicine and rehabilitation* 93.4 (2012), pp. 629–635. ISSN: 0003-9993.

[2] O. W. Heyward et al. “Shoulder complaints in wheelchair athletes: A systematic review”. In: *PloS one* 12.11 (2017), e0188410. ISSN: 1932-6203.

[3] B. Mason et al. “Managing shoulder pain in manual wheelchair users: a scoping review of conservative treatment interventions”. In: *Clinical Rehabilitation* 34.6 (2020), pp. 741–753. ISSN: 0269-2155.

[4] D. D. Gutierrez et al. “The relationship of shoulder pain intensity to quality of life, physical activity, and community participation in persons with paraplegia”. In: *The journal of spinal cord medicine* 30.3 (2007), pp. 251–255. ISSN: 1079-0268.

[5] T. A. Dyson-Hudson and S. C. Kirshblum. “Shoulder pain in chronic spinal cord injury, part 1: epidemiology, etiology, and pathomechanics”. In: *The journal of spinal cord medicine* 27.1 (2004), pp. 4–17. ISSN: 1079-0268.

[6] K. A. Curtis et al. “Shoulder pain in wheelchair users with tetraplegia and paraplegia”. In: *Archives of Physical Medicine and Rehabilitation* 80.4 (1999), pp. 453–457. ISSN: 0003-9993.

[7] B. Odle et al. “Construction and evaluation of a model for wheelchair propulsion in an individual with tetraplegia”. In: *Medical and Biological Engineering and Computing* 57.2 (2019), pp. 519–532. DOI: 10.1007/s11517-018-1895-z.

[8] F. M. Bossuyt et al. “Shoulder pain in the Swiss spinal cord injury community: prevalence and associated factors”. In: *Disability and rehabilitation* 40.7 (2018), pp. 798–805. ISSN: 0963-8288.

[9] F. M. Bossuyt et al. “Compensation Strategies in Response to Fatiguing Propulsion in Wheelchair Users: Implications for Shoulder Injury Risk”. In: *Am J Phys Med Rehabil* 99.2 (2020), pp. 91–98. ISSN: 1537-7385 (Electronic) 0894-9115 (Linking). DOI: 10.1097/phm.0000000000001267.

[10] M. L. Boninger et al. “Wheelchair pushrim kinetics: body weight and median nerve function”. In: *Archives of physical medicine and rehabilitation* 80.8 (1999), pp. 910–915. ISSN: 0003-9993.

[11] J. L. Mercer et al. “Shoulder joint kinetics and pathology in manual wheelchair users”. In: *Clinical biomechanics* 21.8 (2006), pp. 781–789. ISSN: 0268-0033.

[12] L. H. van der Woude et al. “Alternative modes of manual wheelchair ambulation: an overview”. In: *American journal of physical medicine and rehabilitation* 80.10 (2001), pp. 765–777. ISSN: 0894-9115.

[13] S. J. Mulroy et al. “Effect of fore-aft seat position on shoulder demands during wheelchair propulsion: part 1. A kinetic analysis”. In: *The journal of spinal cord medicine* 28.3 (2005), pp. 214–221.

[14] S. J. Mulroy et al. “Effects of spinal cord injury level on the activity of shoulder muscles during wheelchair propulsion: An electromyographic study”. In: *Archives of Physical Medicine and Rehabilitation* 85.6 (2004), pp. 925–934. DOI: 10.1016/j.apmr.2003.08.090.

[15] J. W. Rankin et al. “The influence of wheelchair propulsion technique on upper extremity muscle demand: A simulation study”. In: *Clinical Biomechanics* 27.9 (2012), pp. 879–886. DOI: 10.1016/j.clinbiomech.2012.07.002.

[16] A. M. Kwarciak et al. “Comparing handrim biomechanics for treadmill and overground wheelchair propulsion”. In: *Spinal cord* 49.3 (2011), pp. 457–462. ISSN: 1476-5624.

[17] S. J. Mulroy et al. “Electromyographic activity of shoulder muscles during wheelchair propulsion by paraplegic persons”. In: *Archives of Physical Medicine and Rehabilitation* 77.2 (1996), pp. 187–193. DOI: 10.1016/s0003-9993(96)90166-5.

[18] J. W. Rankin, W. M. Richter, and R. R. Neptune. "Individual muscle contributions to push and recovery subtasks during wheelchair propulsion". In: *Journal of Biomechanical Engineering* 44.7 (2011), pp. 1246–1252. doi: 10.1016/j.jbiomech.2011.02.073.

[19] H. T. Lin et al. "Muscle forces analysis in the shoulder mechanism during wheelchair propulsion". In: *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* 218.H4 (2004), pp. 213–221. doi: 10.1243/0954411041561027.

[20] H. E. J. Veeger, L. H. V. Vanderwoude, and R. H. Rozendal. "LOAD ON THE UPPER EXTREMITY IN MANUAL WHEELCHAIR PROPULSION". In: *Journal of Electromyography and Kinesiology* 1.4 (1991), pp. 270–280. doi: 10.1016/1050-6411(91)90014-v.

[21] F. C. Van der Helm. "A finite element musculoskeletal model of the shoulder mechanism". In: *Journal of biomechanics* 27.5 (1994), pp. 551–569. issn: 0021-9290.

[22] H. E. J. Veeger, L. A. Rozendaal, and F. C. T. van der Helm. "Load on the shoulder in low intensity wheelchair propulsion". In: *Clinical Biomechanics* 17.3 (2002), pp. 211–218. doi: 10.1016/s0268-0033(02)00008-6.

[23] S. van Drongelen et al. "Glenohumeral contact forces and muscle forces evaluated in wheelchair-related activities of daily living in able-bodied subjects versus subjects with paraplegia and tetraplegia". In: *Archives of Physical Medicine and Rehabilitation* 86.7 (2005), pp. 1434–40. doi: 10.1016/j.apmr.2005.03.014.

[24] R. J. K. Vegter et al. "Early motor learning changes in upper-limb dynamics and shoulder complex loading during handrim wheelchair propulsion". In: *Journal of NeuroEngineering and Rehabilitation* 12 (2015). doi: 10.1186/s12984-015-0017-5.

[25] B. J. Fregly, M. L. Boninger, and D. J. Reinkensmeyer. "Personalized neuromusculoskeletal modeling to improve treatment of mobility impairments: a perspective from European research sites". In: *Journal of NeuroEngineering and Rehabilitation* 9.1 (2012), p. 18. issn: 1743-0003. doi: 10.1186/1743-0003-9-18. url: https://doi.org/10.1186/1743-0003-9-18.

[26] F. C. T. van der Helm and H. E. J. Veeger. "Quasi-static analysis of muscle forces in the shoulder mechanism during wheelchair propulsion". In: *Journal of Biomechanical Engineering* 29.1 (1996), pp. 39–52. doi: 10.1016/0021-9290(95)00026-7.

[27] J. S. Slowik et al. "Compensatory strategies during manual wheelchair propulsion in response to weakness in individual muscle groups: A simulation study". In: *Clinical Biomechanics* 33 (2016), pp. 34–41. doi: 10.1016/j.clinbiomech.2016.02.003.

[28] J. S. Slowik et al. "The influence of wheelchair propulsion hand pattern on upper extremity muscle power and stress". In: *Journal of Biomechanics* 49.9 (2016), pp. 1554–1561. doi: 10.1016/j.jbiomech.2016.03.031.

[29] A. Seth et al. "A biomechanical model of the scapulothoracic joint to accurately capture scapular kinematics during shoulder movements". In: *PloS one* 11.1 (2016), e0141028. issn: 1932-6203.

[30] S. R. Dubowsky et al. "Validation of a musculoskeletal model of wheelchair propulsion and its application to minimizing shoulder joint forces". In: *Journal of Biomechanical Engineering* 41.14 (2008), pp. 2981–2988. doi: 10.1016/j.jbiomech.2008.07.032.

[31] A. J. Spaepen, Y. C. Vanlandewijck, and R. J. Lysens. "Relationship between energy expenditure and muscular activity patterns in handrim wheelchair propulsion". In: *International Journal of Industrial Ergonomics* 17.2 (1996), pp. 163–173. doi: 10.1016/0169-8141(95)00047-x.

[32] L. Qi, M. Ferguson-Pell, and Y. Lu. "The Effect of Manual Wheelchair Propulsion Speed on Users' Shoulder Muscle Coordination Patterns in Time-Frequency and Principal Component Analysis". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.1 (2019), pp. 60–65. doi: 10.1109/tnsre.2018.2886826.

[33] G. Wu et al. "ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion—Part II: shoulder, elbow, wrist and hand". In: *Journal of biomechanics* 38.5 (2005), pp. 981–992. issn: 0021-9290.

[34] S. L. Delp et al. "OpenSim: open-source software to create and analyze dynamic simulations of movement". In: *IEEE transactions on biomedical engineering* 54.11 (2007), pp. 1940–1950.

[35] D. C. Ackland et al. “Moment arms of the muscles crossing the anatomical shoulder”. In: *Journal of anatomy* 213.4 (2008), pp. 383–390. ISSN: 0021-8782.

[36] K. R. Holzbaur, W. M. Murray, and S. L. Delp. “A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control”. In: *Annals of biomedical engineering* 33 (2005), pp. 829–840. ISSN: 0090-6964.

[37] L. Modenese et al. “Estimation of musculotendon parameters for scaled and subject specific musculoskeletal models using an optimization technique”. In: *Journal of biomechanics* 49.2 (2016), pp. 141–148. ISSN: 0021-9290.

[38] I. Belli et al. “Does enforcing glenohumeral joint stability matter? A new rapid muscle redundancy solver highlights the importance of non-superficial shoulder muscles”. In: *Plos one* 18.11 (2023), e0295003. ISSN: 1932-6203.

[39] M. De Zee et al. “Validation of a musculo-skeletal model of the mandible and its application to mandibular distraction osteogenesis”. In: *Journal of biomechanics* 40.6 (2007), pp. 1192–1201. ISSN: 0021-9290.

[40] L. Chiari et al. “Human movement analysis using stereophotogrammetry: Part 2: Instrumental errors”. In: *Gait and posture* 21.2 (2005), pp. 197–211. ISSN: 0966-6362.

[41] A. Seth et al. “Muscle contributions to upper-extremity movement and work from a musculoskeletal model of the human shoulder”. In: *Frontiers in neurorobotics* 13 (2019), p. 90. ISSN: 1662-5218.

[42] J. L. Hicks et al. “Is my model good enough? Best practices for verification and validation of musculoskeletal models and simulations of movement”. In: *Journal of biomechanical engineering* 137.2 (2015), p. 020905. ISSN: 0148-0731.

A

Adjustments to the RMR Solver

A.1. main_analyse_dataset.m

```
% Script to run the Rapid Muscle Redundancy (RMR) solver on user-selected experiments.
% The user is prompted with the selection of the tasks to analyze.
% Within the script, it is possible to adjust the downsampling to be
% applied, and whether the analysis should include the glenohumeral
% constraint or not.
%
% Author: Italo Belli  2023

close all; clear; clc; beep off;

% Import the OpenSim libraries.
import org.opensim.modeling.*;

% set the path current folder to be the one where this script is contained
mfile_name          = mfilename('fullpath');
[pathstr,~,~]  = fileparts(mfile_name);
cd(pathstr);

% getting path to other folders in this repo
addpath(pathstr)
cd ../../../
path_to_repo = pwd;
addpath(path_to_repo)
addpath(fullfile(path_to_repo, 'Code/Data␣Processing/'))

%choose participant & Trial
participant = 'UEFS10';
dataset_considered = 'Trial2_45W';

%path to OpenSim folder
path_to_opensim = ['/Users/guiomarsantoscarvalho/OpenSim/Thesis/', participant,'/'];

% Select model
modelFile = append(path_to_opensim, 'TSM_', participant,'_scaled.osim');
model = Model(modelFile);

% where you have the experimental files (.trc)
trc_path = fullfile(path_to_opensim, 'TRC');
[files,path] = uigetfile('*.trc', 'Select the .trc files to analyse', trc_path, 'MultiSelect'
    ,'on');
experiment = append(path,files);
% experiment = 0;         %No TRC file c

% where to save the results
saving_path = fullfile(path_to_opensim, 'RMR/');

% get the motion file from Scaling
```

```matlab
% motion_file = fullfile([path_to_opensim, '/RMR/'] , ['IK_', participant, '_ExpTrial_2_4kmh_45W.mot']);
% motion_file = fullfile(path_to_opensim, 'IK', 'IK_ExpTrial_2_4kmh_45W.mot');
motion_file = 0; % No motionfile

% Downsampling
time_interval = 1;

% Set the weight for the various scapula coordinates in IK
% This is to achieve a good agreement between scapula upward rotation and
% shoulder elevation (as reported in the paper)
weight_abd = 0.0001;
weight_elev = 0.0002;
weight_up_rot = 0.0002;
weigth_wing = 0.0001;
weight_coord = [weight_abd, weight_elev, weight_up_rot, weigth_wing];

% Flags (Select whether to enforce constraints)
dynamic_bounds = true;               % enforcing continuity of the activations from one timestep to the next, to respect first-order dynamics
enforce_GH_constraint = true;       % enforcing directional constraint on the glenohumeral joint force
apply_external_force = 1;

%Check if 3 extra markers were added to the model (G_center,HH_center,G_edge)
numMarkers = model.getNumMarkers();
assert(numMarkers == 16, 'Add␣3␣markers:G_center,HH_center,G_edge');

%% Generate the external force and add it to the model
force_params =[];
force_params.apply_external_force = apply_external_force;
force_1 = [];
% Create an empty torque identifier
torque_identifier = '';
ground_force_p = '␣';

if apply_external_force

    external_force_filename = 'ExternalForces_Trial2_45W.xml';             % name of the filename in which the force is going to be stored

    % Create Storage object
    output_file_path = fullfile(path_to_opensim, 'ExternalLoads', [participant '_ExpTrial_2_4kmh_45W_ExternalForce_filt.sto']);
    data_storage = Storage(output_file_path);
    data_storage.setName([participant '_ExpTrial_2_4kmh_45W_ExternalForce_filt.sto']);

    % Create ExternalForce object
    external_force = ExternalForce(data_storage, "ground_force_v", ground_force_p, torque_identifier, "hand", "ground", "ground");
    external_force.print(fullfile(path_to_opensim, 'ExternalLoads', external_force_filename));

    % Save external force parameters in structure
    force_1.ef_filename = external_force_filename;
    force_1.ef_storage = data_storage;
    force_1.ef = external_force;

    % Update force_params structure
    force_params.num_forces = 1;
    force_params.forces{1} = force_1;
end

%% Run Rapid Muscle Redundancy (RMR) solver
disp('Running␣RMR')

[optimization_status, unfeasibility_flags, tOptim, result_file] = RMR_analysis(participant, model, experiment, motion_file, weight_coord, time_interval, dynamic_bounds, enforce_GH_constraint, force_params, saving_path);

fprintf('\n␣Solved␣with␣%i␣unfeasible␣solutions␣\n␣\n␣\n', sum(unfeasibility_flags));
```

A.2. RMR_analysis.m

```matlab
function [optimizationStatus, unfeasibility_flags, tOptim, file_results] = RMR_analysis(
    subject_considered, model_original, trc_file, motion_file, weight_coord, time_interval,
    dynamic_activation_bounds, flag_JRC_enforced, force_params, saving_path)
% Rapid Muscle Redundancy (RMR) solver, leveraging OpenSim API.
% Starting from experimental marker data (in .trc format) the optimal
% muscle activations are found that can reproduce the motion, solving:
%
%   min   sum (w_i * a_i^2)+ sum (w_j * c_j^2)
%   a,c    i                  j
%
%   s.t.        a_min<=a_i<=a_max    for muscle activations
%              -1<=c_j<=1            for coordinateActuators controls (if present)
%        acc_{j,FD} = acc_{j,data}   constraint on accelerations
%               F_{GH} \in Cone      glenohumeral constraint
%
%
% The code is written specifically to consider a thoracoscapular shoulder
% model that has been already scaled to the biometrics of the subject of
% interest However, this script can be generalized to consider other models
% and data without changing its main structure.
%
% INPUTS:
% * subject_considered: string defining the name of the subject analyzed
%                       (used to save results)
% * model_original: model to be used for the analysis
%                  (valid TSM model with GH markers and coordinate actuators)
% * trc_file : path to the file - and file name - from which to retrieve
%              marker data for IK and subsequent RMR analysis (set to 0 if the
%              input is actually the motion file)
% * motion_file: path to the file - and file name - that carries
%                information on the coordinates (set to 0 if trc file is
%                used)
% * weight_coord : 4x1 vector indicating the weight of each scapula DoF in
%                  the IK tracking
%                  (order: abduction, elevation, upward rotation, winging)
% * time_interval : downsampling of the original data, to reduce
%                   computation effort for the RMR. For example, if set to 10,
%                   every 10th time point is selected.
% *dynamic_activation_bounds : flag to indicate whether dynamic bounds must
%                              be used to limit the activation values during
%                              RMR solution
% * flag_JRC_enforced: true or false, if a joint reaction constraint is
%                      considered or not
% * force_params: parameters of the external force(s) applied
% * saving_path: path to where the results of the redundancy solver are
%                saved
%
% OUTPUT:
% * optimizationStatus: struct containing the status of the optimization at
%                       each of the timestep in which RMR was performed.
% * unfeasibility_flags: an array of the same length as the time instants
%                        considered, containing 0 if the problem is solved and
%                        1 if it is unfeasible
% * tOptim : time required to perform the complete optimization
% * file_results: path and name of the file where the activation results
%                 are saved
% The function also saves plots of the analysis performed, and the muscle
% activation variables (together with coordinate actuators controls) in a
% .mat file

%% Import the OpenSim libraries.
import org.opensim.modeling.*;

%% General settings
% if these are set to true, results are printed but the code will be slower
print_flag = true;
withviz = false;

%% Set the correct paths
```

```
% set the path current folder to be the one where this script is contained
mfile_name          = mfilename('fullpath');
[pathstr,~,~]  = fileparts(mfile_name);
cd(pathstr);

% getting path to other folders in this repo
addpath(pathstr)
cd ../../../
path_to_repo = pwd;
addpath(path_to_repo)
addpath(fullfile(path_to_repo, 'Code/Data␣Processing/'))

%path to OpenSim folder
path_to_opensim = ['/Users/guiomarsantoscarvalho/OpenSim/Thesis/',subject_considered,'/'];

% cd to Personal Results to have all the results saved there
cd([path_to_opensim, 'RMR']);

% create a temporary copy of the model, to be used in the function. In this
% way, the model can be modified freely here without interfering with its
% state/properties outside this function
model_temp = model_original.clone();

%% Getting quantities about GlenoHumeral joint
% get the glenohumeral joint
alljoints = model_temp.getJointSet;
glen = alljoints.get('GlenoHumeral');

state = model_temp.initSystem();
[maxAngle, ~] = get_glenoid_status(model_temp, state); % the value for maxAngle can also be given directly by the user

%% Load the trc file to be considered, if the input is a trc file, and perform IK
if trc_file
    [~, experiment_name] = fileparts(trc_file);
    [markersExp, timesExp, ~, unitsExp] = readTRC(trc_file);
    start_time = timesExp(1);
    end_time =  timesExp(end);

    if strcmp(unitsExp, 'mm')
        markersExp = markersExp/1000;
        unitsExp = 'm';
    end

    frequency_trc_data = 1/(timesExp(2)-timesExp(1));

    % getting the values of default scapula coordinate
    % we get the values of the coordinates describing the scapula position from
    % the general model in default pose
    scapula_abd = model_temp.getJointSet().get(2).get_coordinates(0);
    scapula_ele = model_temp.getJointSet().get(2).get_coordinates(1);
    scapula_urt = model_temp.getJointSet().get(2).get_coordinates(2);
    scapula_wng = model_temp.getJointSet().get(2).get_coordinates(3);

    default_sa = scapula_abd.get_default_value();
    default_se = scapula_ele.get_default_value();
    default_su = scapula_urt.get_default_value();
    default_sw = scapula_wng.get_default_value();

    % Performing IK
    % perform IK on the basis of marker data to retrieve the motion file for
    % the coordinates of the model

    motion_file_name = append('IK_', experiment_name, '.mot');

    ikSetupFile = [path_to_opensim,'' ...
            'IK/IK_Setup_Trial2_45W.xml'];

    ikTool = InverseKinematicsTool(ikSetupFile);
    ikTool.setMarkerDataFileName(trc_file);
    ikTool.setOutputMotionFileName([path_to_opensim, 'RMR/', motion_file_name]);
    ikTool.set_report_marker_locations(1);
    ikTool.setStartTime(start_time);
    ikTool.setEndTime(end_time); %CHANGEEEE only considers 2 sec
    ikTool.setModel(model_temp);

    % set the reference values for the scapula coordinates (last 4 tasks)
    num_IK_tasks = ikTool.getIKTaskSet.getSize();

    % set the weight of each coordinate in the tracking tasks
    ikTool.getIKTaskSet.get(num_IK_tasks-4).setWeight(weight_coord(1));
    ikTool.getIKTaskSet.get(num_IK_tasks-3).setWeight(weight_coord(2));
    ikTool.getIKTaskSet.get(num_IK_tasks-2).setWeight(weight_coord(3));
    ikTool.getIKTaskSet.get(num_IK_tasks-1).setWeight(weight_coord(4));

    % set also the values here
    IKCoordinateTask.safeDownCast(ikTool.getIKTaskSet.get(num_IK_tasks-4)).setValue(
        default_sa);
    IKCoordinateTask.safeDownCast(ikTool.getIKTaskSet.get(num_IK_tasks-3)).setValue(
        default_se);
    IKCoordinateTask.safeDownCast(ikTool.getIKTaskSet.get(num_IK_tasks-2)).setValue(
        default_su);
    IKCoordinateTask.safeDownCast(ikTool.getIKTaskSet.get(num_IK_tasks-1)).setValue(
        default_sw);
    ikTool.print('RMR_autogenerated_IK_setup.xml');

    ikTool.run();

else
    [~, experiment_name] = fileparts(motion_file);
    motion_file_name = motion_file;
    q = read_motionFile(motion_file_name);
    time = q.data(:,1);
    start_time = time(1);
    end_time = time(end);
    frequency_trc_data = 1/(time(2)-time(1));
end

%% getting the kinematic data that we need
% Use the loadFilterCropArray() function provided by OpenSim Tutorial to load the
% coordinate kinematic and generalized force data into MATLAB arrays. This
% function also filters and crops the loaded array based on its two input
% arguments (more details in loadFilterCropArray.m).
lowpassFreq = 3.0; % Hz
timeRange = [start_time end_time];

% get the coordinates from the output of the IK in rad for the rotational
% joints
[coordinates, coordNames, timesExp] = loadFilterCropArray(motion_file_name, lowpassFreq,
    timeRange);
coordinates(:, 1:3) = deg2rad(coordinates(:, 1:3));
coordinates(:, 7:end) = deg2rad(coordinates(:, 7:end));

%Do not use clavicle coordinates for acc matching in the optimization
% coordNames = coordNames([1:6,9:end], 1);
% coordinates = coordinates(:,[1:6,9:end]);

% get the velocities for each joint in rad/s
time_step_data = timesExp(2)-timesExp(1);
speeds = zeros(size(coordinates));
for i=1:size(coordNames,1)
    speeds(:,i) = gradient(coordinates(:,i), time_step_data);
end
speedNames = coordNames;

% get the accelerations for each coordinate in rad/s^2
accelerations = zeros(size(speeds));
for i=1:size(coordNames,1)
    accelerations(:,i) = gradient(speeds(:,i), time_step_data);
end
accNames = speedNames;

```

```
% visually check the values of joint states, speeds and accelerations
if print_flag
    figure
    for i=1:16%size(coordNames,1)
    subplot(4,4,i)
    hold on
    plot(coordinates(:,i))
    plot(speeds(:,i))
    plot(accelerations(:,i))
    title(coordNames{i});
    hold off
    grid on
    end
    legend("coords", "speeds", "accs")
end

%% Store max isometric force values and disable muscle dynamics
muscles = model_temp.getMuscles();
numMuscles = muscles.getSize();
muscles_downcasted = cell(numMuscles,1);
muscleNames = cell(numMuscles,1);

% save here downcasted muscles to a list
if strcmpi(muscles.get(0).getConcreteClassName(), 'Thelen2003Muscle')
    for index_muscle = 1:numMuscles
       % Downcast base muscle to Thelen2003Muscle
       muscles_downcasted(index_muscle) = Thelen2003Muscle.safeDownCast(muscles.get(index_muscle-1));
       muscleNames{index_muscle} = char(muscles_downcasted{index_muscle});
    end
else
    for index_muscle = 1:numMuscles
       % Downcast base muscle to Millard2012EquilibriumMuscle
       muscles_downcasted(index_muscle) = Millard2012EquilibriumMuscle.safeDownCast(muscles.get(index_muscle-1));
       muscleNames{index_muscle} = char(muscles_downcasted{index_muscle});
       muscles_downcasted{index_muscle}.set_ignore_tendon_compliance(true);  % not really relevant as actuation will be overwritten
       muscles_downcasted{index_muscle}.set_ignore_activation_dynamics(true);
    end
end

if (withviz == true)
    model_temp.setUseVisualizer(true);
end

%% Add external force
if force_params.apply_external_force
    % get how many forces we need to apply
    num_forces = force_params.num_forces;

    for force_index = 1:num_forces
        % this part requires to be rewritten to account for the custom external
        % force that the user wants to apply
        file_name = force_params.forces{force_index}.ef_filename;
        storage_file = force_params.forces{force_index}.ef_storage;
        external_force = force_params.forces{force_index}.ef;

        % add the force to the model (it is added as the last element of the
        % force set)
        model_temp.addForce(external_force);
    end

    % ensure that the force is correctly integrated in teh model
    model_temp.finalizeConnections();
end

% Update the system to include any muscle modeling changes
state = model_temp.initSystem();

%% Get coordinate actuators
```

```
allActs = model_temp.getActuators;
num_acts = getSize(allActs);
acts = cell(num_acts,1);

% get all actuators and override actuation for the muscles only
for i = 1:num_acts
    acts(i) = ScalarActuator.safeDownCast(allActs.get(i-1));
    if i<=numMuscles
        acts{i}.overrideActuation(state, true);
        % acts{i}.computeEquilibrium(state);
    end
end

%% Perform optimization
% We use FMINCON to solve the optimization problem at selected time points.
% The 'time_interval' variable selects the time points to be included in the
% optimization. For example, if set to 10, every 10th time point is selected. A
% time interval of 1 will select all available time points.
time_step_RMR = time_step_data * time_interval;

% Update data arrays based on the time_interval.
N = size(coordinates, 1);
coordinates = coordinates(1:time_interval:N, :);
speeds = speeds(1:time_interval:N, :);
accelerations = accelerations(1:time_interval:N, :);
numTimePoints = size(coordinates, 1);
unfeasibility_flags = zeros(size(numTimePoints));
exit2 = zeros(size(numTimePoints));

% Create the FMINCON options structure.
options = optimoptions('fmincon','Display','none', ...
     'TolCon',1e-3,'TolFun',1e-3,'TolX',1e-2,'MaxFunEvals',100000, ...
     'MaxIter',10000,'Algorithm','sqp', 'StepTolerance', 1e-10); %, 'DiffMinChange', 1.0e-2);
%1e-4
% Construct initial guess and bounds arrays
numCoords = length(coordNames);
numCoordActs = num_acts-numMuscles;
k = inf;
t_act = 0.01;           % activation time constant for muscles
t_deact = 0.04;         % deactivation time constant

lb = [zeros(1,numMuscles), -k*ones(1,numCoordActs)];
ub = [ones(1,numMuscles), k*ones(1,numCoordActs)];
x_zero = [0.1* ones(1,numMuscles), zeros(1,numCoordActs)];
x0 = x_zero;                                                %set initial guess to 0 (for fmincon)

% We define the activation squared cost as a MATLAB anonymous function
% It is model specific!
epsilon = 0;
w = [ones(1,numMuscles), epsilon*ones(1,6),10,epsilon,10,10,10,12,10,12,10*ones(1,3)];      % the cost function is written such that it allows the use of coord acts for the underactuated coordinates
cost =@(x) sum(w.*(x.^2));

% Pre-allocate arrays to be filled in the optimization loop
fl = zeros(1, numMuscles);
fv = zeros(1, numMuscles);
fp = zeros(1, numMuscles);
cosPenn = zeros(1, numMuscles);
Fmax = zeros(1, numMuscles);
A_eq_acc = zeros(numCoords,num_acts);
A_eq_force = zeros(3, num_acts);
xsol = zeros(numTimePoints, length(x0));
simulatedAccelerations = zeros(numTimePoints, length(coordNames));
optimizationStatus = cell(numTimePoints,1);
norm_fv_in_ground = zeros(numTimePoints, 3);
norm_fv_rotated = zeros(numTimePoints, 3);
rel_angle = zeros(numTimePoints,1);
MuscVelocity = zeros(numMuscles,numTimePoints);                    %ADDED
MuscPower = zeros(numTimePoints,numMuscles);                       %ADDED
```

```
AMuscForce = zeros(numMuscles,numTimePoints);                    %ADDED
PMuscForce = zeros(numMuscles,numTimePoints);                    %ADDED
ExternalForces = zeros(numTimePoints,3);                         %ADDED
ActuatorPower = zeros(numTimePoints,length(accNames));           %ADDED

% get model quantities we still need
coords = model_temp.getCoordinateSet();

for index_muscle = 1:numMuscles
    Fmax(index_muscle) = muscles_downcasted{index_muscle}.getMaxIsometricForce();
end

% do not track plane_elv and axial_rot during shrugging, as they are poorly
% defined when humerus is vertical. Similar in what done
% by Seth et al. in https://simtk.org/projects/thoracoscapular we lock these
if strcmpi(experiment_name(1:5), 'shrug')
    model_temp.getCoordinateSet().get('plane_elv').set_default_value(-0.433725);
    model_temp.getCoordinateSet().get('plane_elv').set_locked(true);
    model_temp.getCoordinateSet().get('axial_rot').set_default_value(0.8125346);
    model_temp.getCoordinateSet().get('axial_rot').set_locked(true);
end

tic

% enter in the optimization loop
for time_instant = 1:numTimePoints
    if print_flag
        fprintf('.');
        if(mod(time_instant,80) == 0)
            fprintf('\n␣%i', time_instant);
        end
    end

    % set the time of the simulation to be the current one
    % (this is especially important if an external force is present, so
    % that the force value is applied at the right instant of time)
    state.setTime((time_instant-1)*time_interval/frequency_trc_data)

    % Loop through model coordinates to set coordinate values and speeds. We set
    % all coordinates to make sure we have the correct kinematic state when
    % compute muscle multipliers and moment arms.
    for j = 1:length(coordNames)
        coord = coords.get(coordNames{j});
        coord.setValue(state, coordinates(time_instant,j), false); % instead of fals replace so that does the assembly on the last call (j==length)
        coord.setSpeedValue(state, speeds(time_instant,j));
    end

    % realize the system to the velocity stage
    model_temp.realizeVelocity(state);

    % equilibrate the muscles to make them start in the correct state
    model_temp.equilibrateMuscles(state);

    modelControls = model_temp.getControls(state);

    % Populate the muscle multiplier arrays. To do this, we must have realized
    % the system to the velocity stage
    for index_muscle = 1:numMuscles
        fl(index_muscle) = muscles_downcasted{index_muscle}.getActiveForceLengthMultiplier(state);
        fv(index_muscle) = muscles_downcasted{index_muscle}.getForceVelocityMultiplier(state);
        fp(index_muscle) = muscles_downcasted{index_muscle}.getPassiveForceMultiplier(state);
        cosPenn(index_muscle) = muscles_downcasted{index_muscle}.getCosPennationAngle(state);
        MuscVelocity(index_muscle,time_instant) = muscles_downcasted{index_muscle}.getFiberVelocity(state);     %ADDED
        %MuscPower(index_muscle,time_instant) = muscles_downcasted{index_muscle}.getMusclePower(state);          %ADDED
    end

    % get the vector Vec_H2GC between humeral head and the glenoid center
    % (it is expressed in the ground frame)
    [~, Vec_H2GC] = get_glenoid_status(model_temp, state);

    % store the values of active and passive maximum force in the current
    % configuration
    AMuscForce = (fl.*fv.*Fmax.*cosPenn)';
    PMuscForce = (Fmax.*fp.*cosPenn)';

    % create a struct containing relevant information to be passed to the
    % function simulating the accelerations and reaction forces and moments
    % induced in the model
    params.model = model_temp;
    params.state = state;
    params.AMuscForce = AMuscForce;
    params.PMuscForce = PMuscForce;
    params.coords = coords;
    params.coordNames = coordNames;
    params.acts = acts;
    params.muscles = muscles;
    params.numMuscles = numMuscles;
    params.useMuscles = 1;
    params.useControls = 1;
    params.modelControls = modelControls;
    % params.joint_to_constrain = [];
    params.joint_to_constrain = glen;

    [q_ddot_0, F_r0, ~, externalForceValues] = findInducedAccelerationsForceMoments(zeros(1,
        num_acts), params);
    delQ_delX = eye(num_acts);

    for k = 1:num_acts
        [incrementalForceAccel_k, F_rk, ~, externalForceValues] =
            findInducedAccelerationsForceMoments(delQ_delX(k,:),params);
        kthColumn_A_eq_acc =  incrementalForceAccel_k - q_ddot_0;
        A_eq_acc(:,k) = kthColumn_A_eq_acc;
        kthColumn_A_eq_force =  F_rk - F_r0;
        A_eq_force(:,k) = kthColumn_A_eq_force;
    end

    Beq = accelerations(time_instant,:)' - q_ddot_0;

    % do not track the 'clav_prot' (7th) and the 'clav_elev' (8th) coordinates
    A_eq_acc(7, :) = zeros(size(A_eq_acc(7, :)));
    A_eq_acc(8, :) = zeros(size(A_eq_acc(8, :)));
    Beq(7, :) = zeros(size(Beq(7, :)));
    Beq(8, :) = zeros(size(Beq(8, :)));

    % Store values of the external force excerted
    ExternalForces(time_instant,:) = externalForceValues;

    % Call FMINCON to solve the problem
    if flag_JRC_enforced
        [x,~,exitflag,output] = fmincon(cost, x0, [], [], A_eq_acc, Beq, lb, ub, @(x)
            jntrxncon_linForce(x, Vec_H2GC, maxAngle, A_eq_force, F_r0), options);
        if exitflag == 0
            % call the solver again, starting from current x, in case the maximum iterations
                are exceeded
            [x,~,exitflag,output] = fmincon(cost, x, [], [], A_eq_acc, Beq, lb, ub, @(x)
                jntrxncon_linForce(x, Vec_H2GC, maxAngle, A_eq_force, F_r0), options);
        end
        if exitflag<0 && time_instant>1
            if  ~isnan(xsol(time_instant-1, 1))
            % call the solver again, starting from previous optimum found,
            % in case optimization gets stuck in local minimum
            [x,~,exitflag,output] = fmincon(cost, xsol(time_instant-1, :), [], [], A_eq_acc,
                Beq, lb, ub, @(x)jntrxncon_linForce(x, Vec_H2GC, maxAngle, A_eq_force, F_r0),
                 options);
            end
        end
        % if no solution was found by optimizer -> xsol = NaN
        if exitflag < 0 %&& time_instant > 1
           x = NaN(1,length(x));
        end
    else
        [x,~,exitflag,output] = fmincon(cost, x0, [], [], A_eq_acc, Beq, lb, ub, [], options);
        if exitflag ==0
            % call the solver again, starting from current x, in case the maximum iterations are exceeded
            [x,~,exitflag,output] = fmincon(cost, x, [], [], A_eq_acc, Beq, lb, ub, [], options);
        end
        if exitflag<0 && time_instant>1
            % call the solver again, starting from previous optimum found,
            % in case optimization gets stuck in local minimum
            [x,~,exitflag,output] = fmincon(cost, xsol(time_instant-1, :), [], [], A_eq_acc, Beq, lb, ub, [], options);
        end
    end

    optimizationStatus{time_instant} = output;

    if exitflag<1 % was 0 before
        unfeasibility_flags(time_instant) = 1;
    end

    if exitflag==2
        exit2(time_instant) = 1;
    end

    % get best feasible point, if different from what returned by fmincon
    if size(output.bestfeasible,1)>0
        x = output.bestfeasible.x;
    end

    % Store solution
    xsol(time_instant, :) = x;

    % Retrieve muscle power -> Otherwise using passive power
    %1.  Set specific muscle activations
    for index_muscle = 0:length(acts)-1
        if index_muscle <= length(muscleNames)-1 %opensim indexing starts at 0
            muscle = model_temp.getMuscles.get(index_muscle);
            muscle.setActivation(state,x(index_muscle+1));
        else
            ScalarActuator.safeDownCast(allActs.get(index_muscle)).setControls(Vector(1, x(index_muscle+1)), modelControls);
        end
    end

    % 2. Realize dynamics (update model dynamics) -> realize accelerations
    % otherwise the reserve actuators are not included
    model_temp.realizeVelocity(state);
    model_temp.setControls(state, modelControls);

    % model_temp.realizeDynamics(state);
    model_temp.realizeAcceleration(state)

    for i = 0:length(acts) - 1
        if i > 34
        ActuatorPower(time_instant,i-34) = allActs.get(i).getPower(state);
        else
        MuscPower(time_instant,i+1) = muscles.get(i).getMusclePower(state);
        end
    end

    if ~isnan(x(1,1))
        % dynamically update the upper and lower bounds for the activations
        if dynamic_activation_bounds
            for k = 1:numMuscles
                lb(k) = max(x(k) - x(k) * (0.5 + 1.5 * x(k)) * time_step_RMR /t_deact, 0);
                ub(k) = min (x(k) + (1-x(k)) * time_step_RMR / (t_act * (0.5 + 1.5*x(k))), 1);
            end
        end

        % if we want to print suff, we need to compute it now
        if print_flag
            % Retrieve the optimal accelerations
            simulatedAccelerations(time_instant,:) = findInducedAccelerationsForceMoments(x, params);

            if flag_JRC_enforced
                % retrieve the position of the joint reaction force on the approximated
                % glenoid computing the reaction force vector at the given joint
                % The force is expressed in the ground frame
                force_vec = A_eq_force * xsol(time_instant, :)' + F_r0;

                % evaluate the relative angle between the reaction force and Vec_H2GC
                cosTheta = max(min(dot(Vec_H2GC,force_vec)/(norm(Vec_H2GC)*norm(force_vec)),1),-1);
                rel_angle(time_instant) = real(acosd(cosTheta));

                % evaluate the position on the glenoid where reaction force is exerted
                norm_Vec_H2GC = Vec_H2GC/norm(Vec_H2GC);
                norm_fv_in_ground(time_instant,:) = force_vec/norm(force_vec);

                beta_angle = atan(norm_Vec_H2GC(3)/norm_Vec_H2GC(1));
                alpha_angle = atan(norm_Vec_H2GC(3)/(sin(beta_angle)*norm_Vec_H2GC(2)));

                Ry = [cos(beta_angle) 0 sin(beta_angle); 0 1 0; -sin(beta_angle) 0 cos(beta_angle)];
                Rz = [cos(alpha_angle) -sin(alpha_angle) 0; sin(alpha_angle) cos(alpha_angle) 0; 0 0 1];

                norm_fv_rotated(time_instant,:) = Rz*Ry*norm_fv_in_ground(time_instant,:)';
            end
        end

        if (withviz == true)
            model_temp.getVisualizer.show(state);
        end
    end
end

tOptim = toc;

%% Plot results
% According to the value of the 'print_flag'
if print_flag
    % plot muscle activations
    f1 = figure;
    title("Muscle Activations")
    muscleNames = ArrayStr();
    muscles.getNames(muscleNames);
    pgc = linspace(0, 100, size(xsol,1));
    for i = 1:numMuscles
       subplot(5,8,i)
       hold on
       plot(pgc,xsol(:,i),'b-')
       ylim([0 1])
       muscName = muscleNames.get(i-1).toCharArray';
       title(muscName(1:end), 'interpreter', 'none')
       hold off
    end
    legend("muscle activation")
    %f1.WindowState = 'maximized';
    name_fig1 = append(experiment_name, '_MuscleActivations.fig');
    saveas(f1, name_fig1)
    close

    % Plot reserve actuator excitations.
    f2 = figure;
    title("Reserve actuators")
    side = ceil(sqrt(numCoordActs));
    for i = 1:numCoordActs
        subplot(side,side,i)
        hold on
        plot(pgc, xsol(:,numMuscles+i), 'linewidth', 2);
        title(char(acts{numMuscles+i}));
        hold off
    end
    legend("reserve act value")
    %f2.WindowState = 'maximized';
    name_fig2 = append(experiment_name, '_ReserveActuators.fig');
    saveas(f2, name_fig2)
    close

    % plot accelerations
    f3 = figure;
    title("Accelerations")
    side = ceil(sqrt(length(coordNames)));
    for i = 1:length(coordNames)
        subplot(side,side,i)
        hold on
        plot(accelerations(:, i), 'linewidth', 1.5);
        plot(simulatedAccelerations(:, i), 'linewidth', 1);
        xlabel("samples")
        ylabel("[]/s^2")
        grid on
        title(coordNames{i});
        hold off
    end
    legend("measured", "simulated")
    f3.WindowState = 'maximized';
    name_fig3 = append(experiment_name, '_AccelerationsMatching.fig');
    saveas(f3, name_fig3)
    close

    % plot the constraint violation on the accelerations per coordinate
    violation = abs(accelerations-simulatedAccelerations);

    f4 = figure;
    for i = 1:length(coordNames)
        subplot(side,side,i)
        hold on
        plot(violation(:,i), 'linewidth', 1.5);
        xlabel("samples")
        ylabel("[]/s^2")
        grid on
        title(coordNames{i});
        hold off
    end
    legend("acc violation")
    %f4.WindowState = 'maximized';
    name_fig4 = append(experiment_name, '_AccViolation.fig');
    saveas(f4, name_fig4)
    close

    % plot the constraint violation per timestep
    violation_t = sum(violation, 2);
    f5 = figure;
    hold on
    scatter(1:numTimePoints ,violation_t, 'filled')
    plot(1:numTimePoints, violation_t, 'blue')
    xlabel("samples")
    ylabel("const violation")
    grid on
    title("Cumulative constraint violation per time-step")
    hold off
    % f5.WindowState = 'maximized';
    name_fig5 = append(experiment_name, '_CumulativeAccViolation.fig');
    saveas(f5, name_fig5)
    close

    % % plot the position of the center of pressure of the joint reaction force
    % if flag_JRC_enforced
    %     radius = sind(maxAngle);
    %     p=nsidedpoly(1000, 'Center', [0,0], 'Radius', radius);
    %     c = linspace(0,timesExp(end),length(norm_fv_rotated));
    %     f6 = figure;
    %     hold on
    %     plot(p, 'FaceColor', 'r')
    %     for time_instant=1:numTimePoints
    %         scatter(-norm_fv_rotated(time_instant,3), -norm_fv_rotated(time_instant,1), [],
         c(time_instant), 'filled')
    %     end
    %     hcb = colorbar;
    %     h = gca;
    %     set(h, "XTickLabel", [])
    %     set(h, "YTickLabel", [])
    %     xlabel("back
       front")   % corresponding roughly to OpenSim X axis (horizontal pointing forward)
    %     ylabel("down
       up")      % corresponding to OpenSim Y axis (vertical pointing upwards)
    %     colorTitleHandle = get(hcb,'Title');
    %     titleString = 'time [s]';
    %     set(colorTitleHandle ,'String',titleString);
    %     hold off
    %     name_fig6 = append(experiment_name, '_CoPGH.png');
    %     saveas(f6, name_fig6)
end

%% SAVING THE RESULTS TO FILE
name_file = append('muscle_activations_', experiment_name);

muscleNames = ArrayStr();
muscles.getNames(muscleNames);

muscle_order = "";
for i = 1:numMuscles
    muscle_order= [muscle_order, string(muscleNames.get(i-1).toCharArray')];
end

for i=1:length(coordNames)
    muscle_order= [muscle_order, string(coordNames{i})];
end

muscle_order = muscle_order(2:end);

% rescale the frequency of the solution knowing the freq of the data
frequency_solution = frequency_trc_data/time_interval;

%setting to have (timesteps x number of muscles)
AMuscForce = AMuscForce.';
PMuscForce = PMuscForce.';
MuscVelocity = MuscVelocity.';

for i = 1:size(acts)
actNames(i)=string(char(acts{i, 1}));
end

save(name_file, 'xsol', 'muscle_order', 'frequency_solution', 'optimizationStatus', '
    unfeasibility_flags', 'tOptim','AMuscForce', 'PMuscForce', 'MuscVelocity', 'MuscPower', '
    ExternalForces', 'exit2', 'violation_t', 'ActuatorPower', 'actNames', '
    simulatedAccelerations', 'accelerations');

file_results = append(saving_path,'/', name_file, '.mat');
end
```

B

Estimated muscle power, activation and measured activation

Participant 07

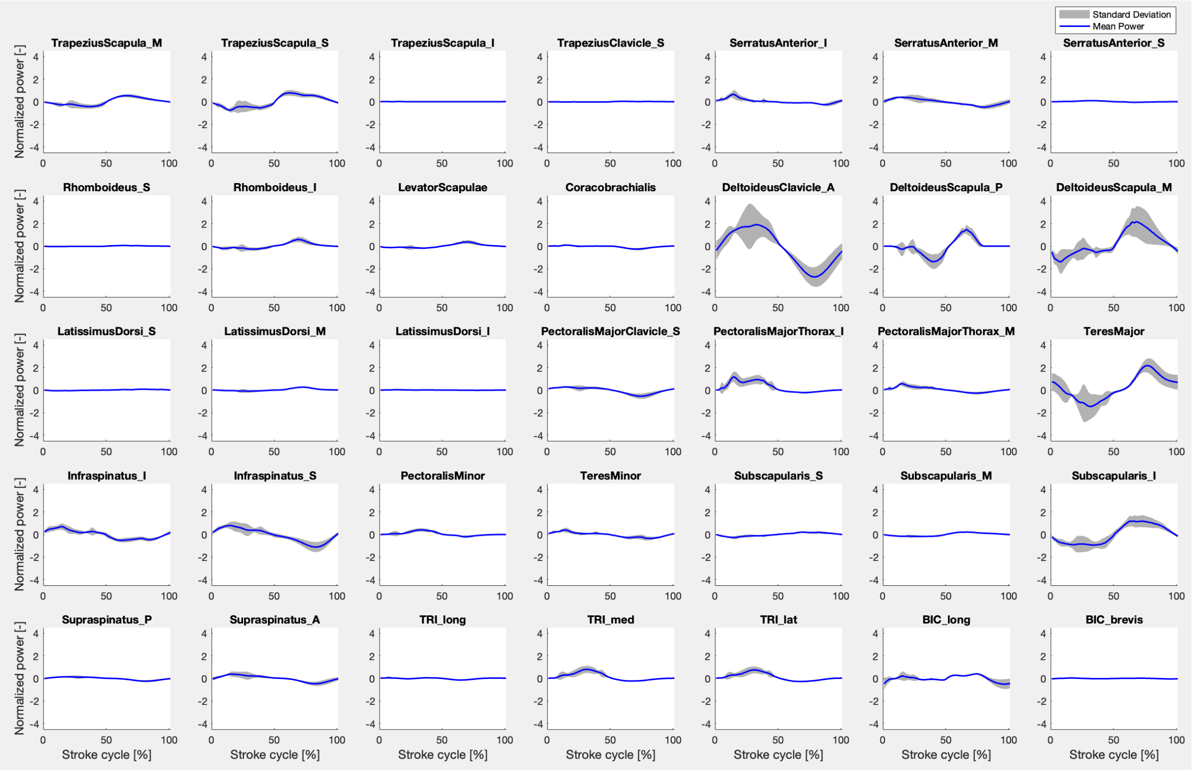


Figure B.1: Average muscle power of participant 07 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

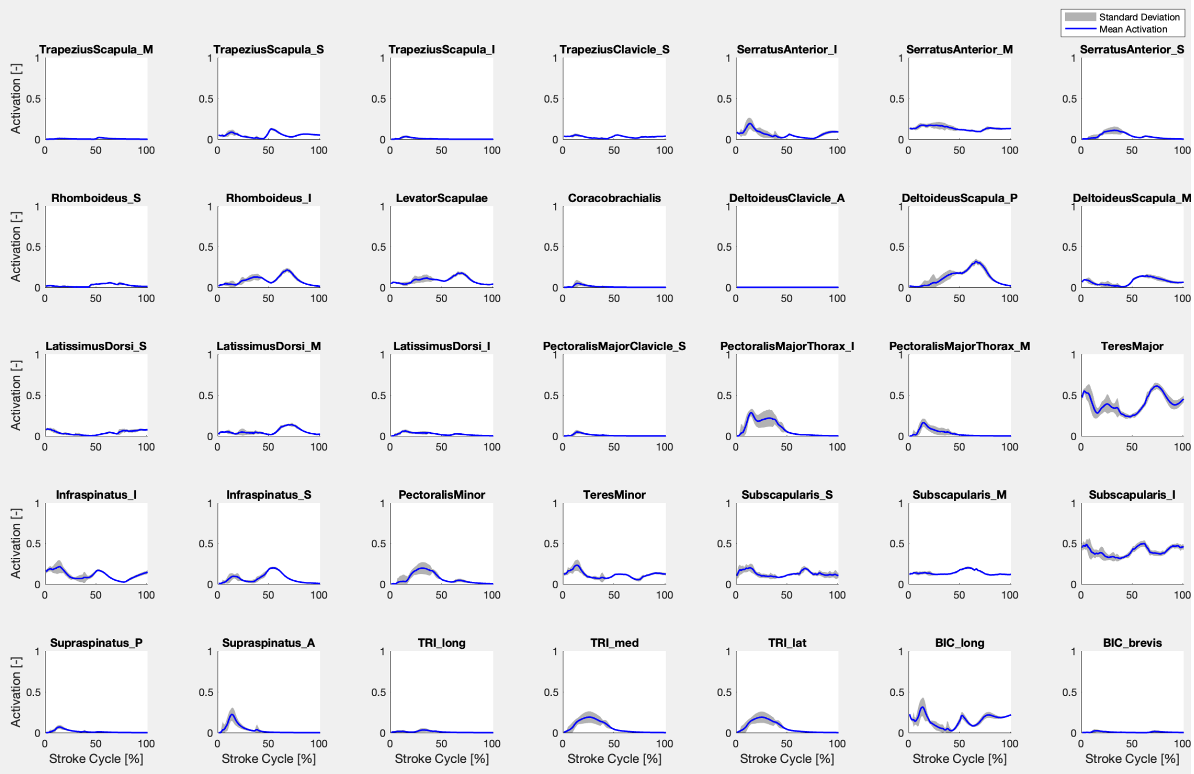


Figure B.2: Average muscle activation of participant 07 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

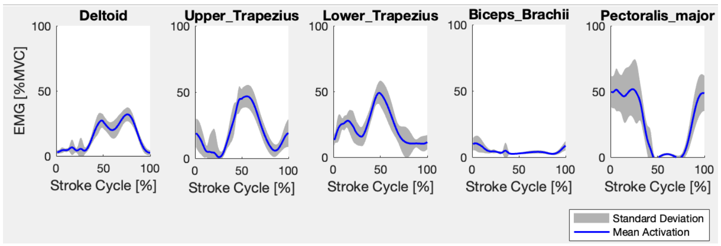


Figure B.3: Average EMG%MVC of participant 07 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

Participant 08

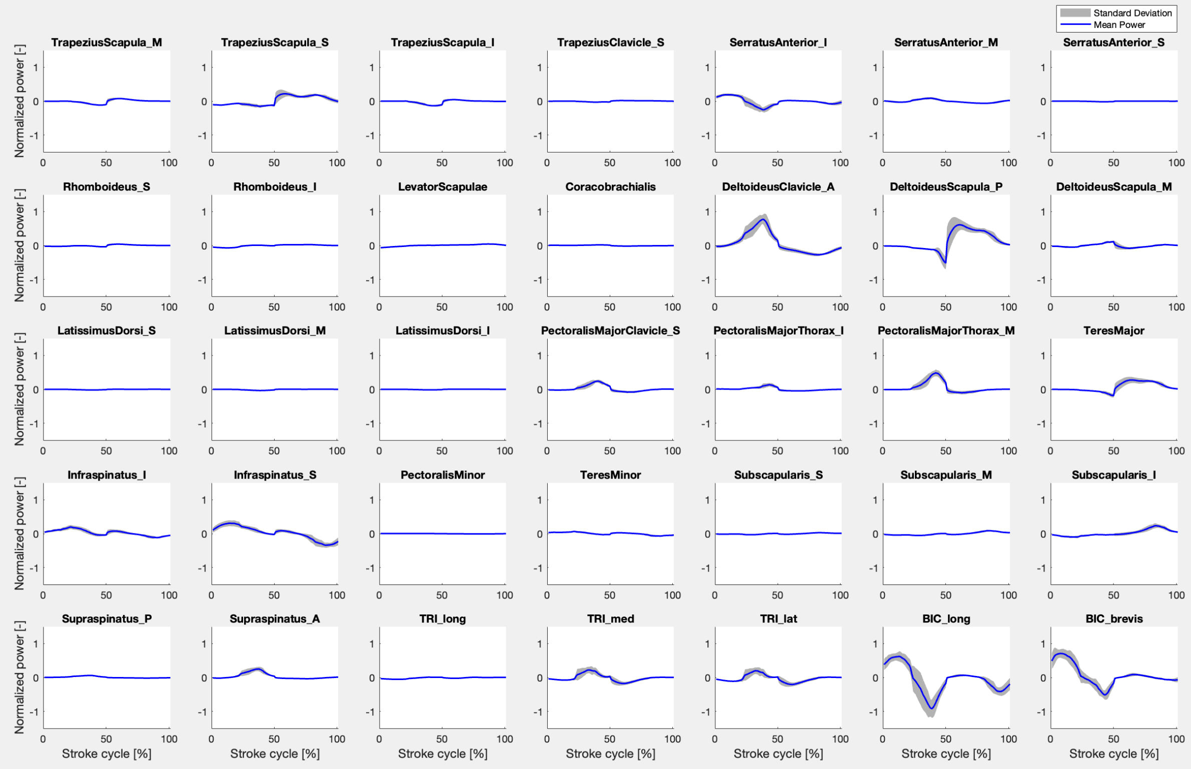


Figure B.4: Average muscle power of participant 08 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

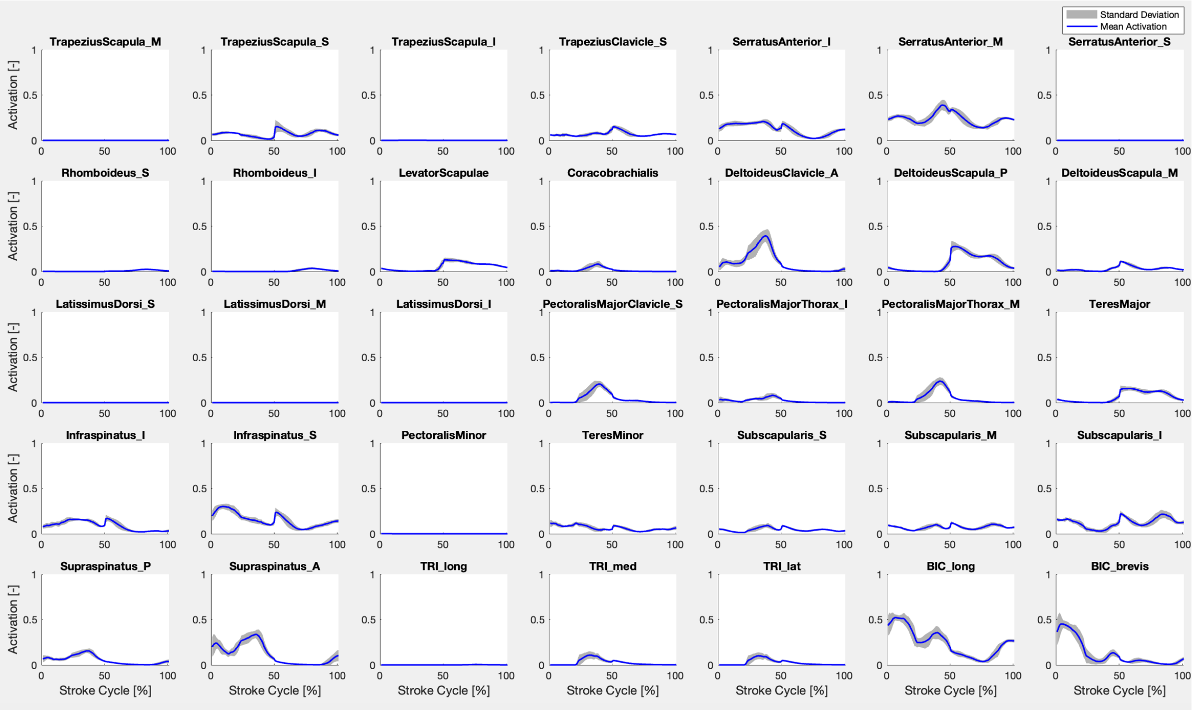


Figure B.5: Average muscle activation of participant 08 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

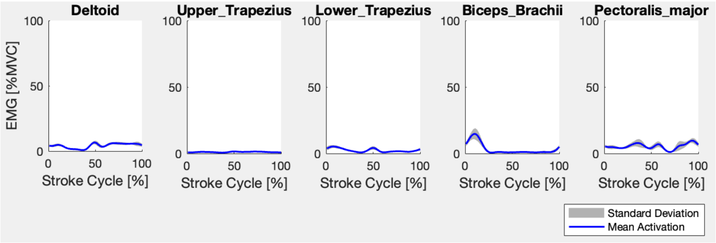


Figure B.6: Average EMG%MVC of participant 08 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

Participant 10

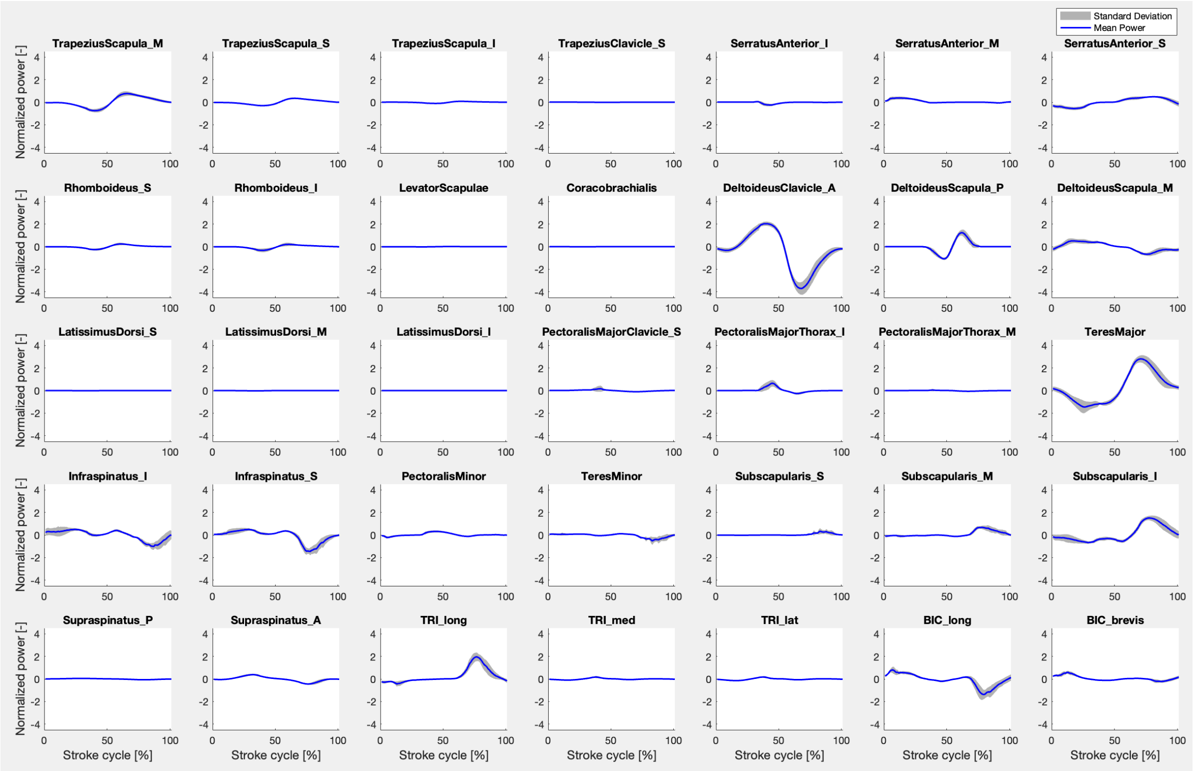


Figure B.7: Average muscle power of participant 10 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

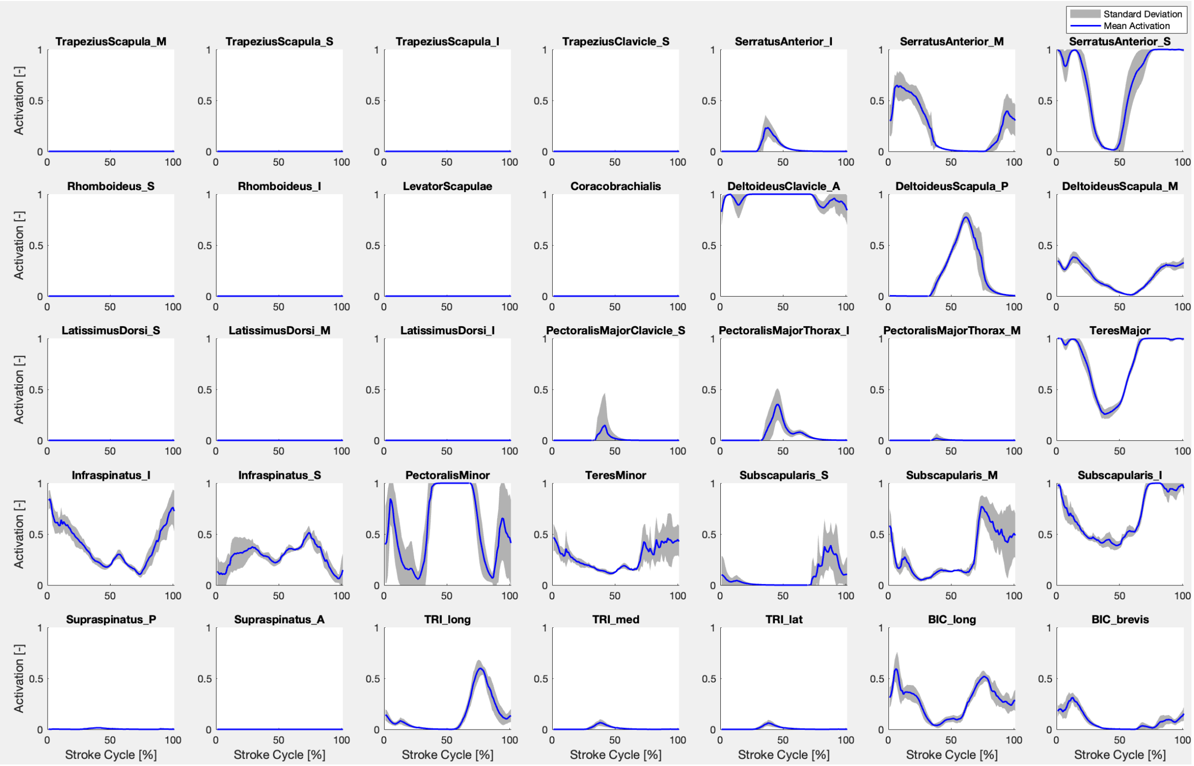


Figure B.8: Average muscle activation of participant 10 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

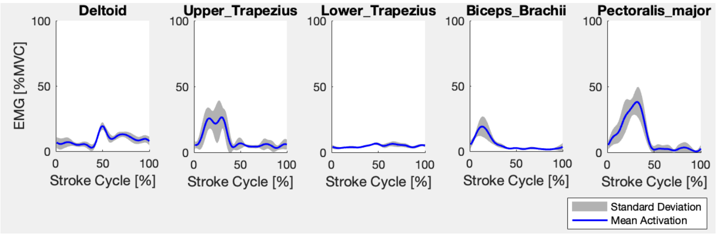


Figure B.9: Average EMG%MVC of participant 10 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

Participant 12

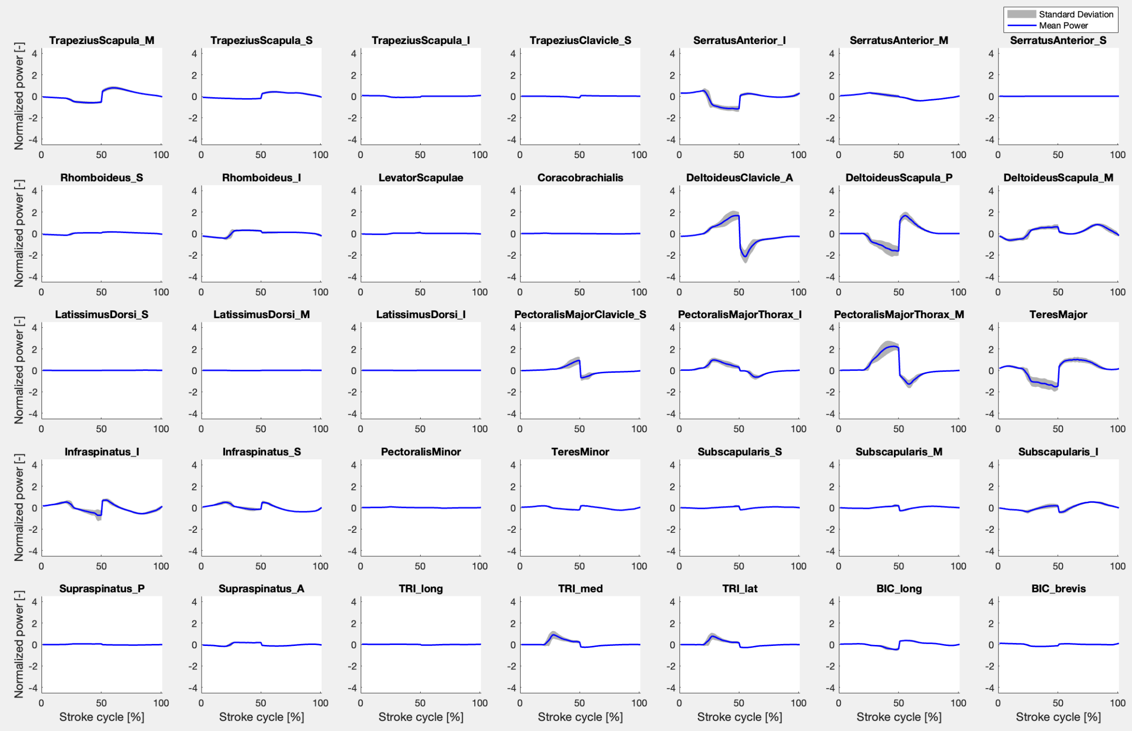


Figure B.10: Average muscle power of participant 12 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

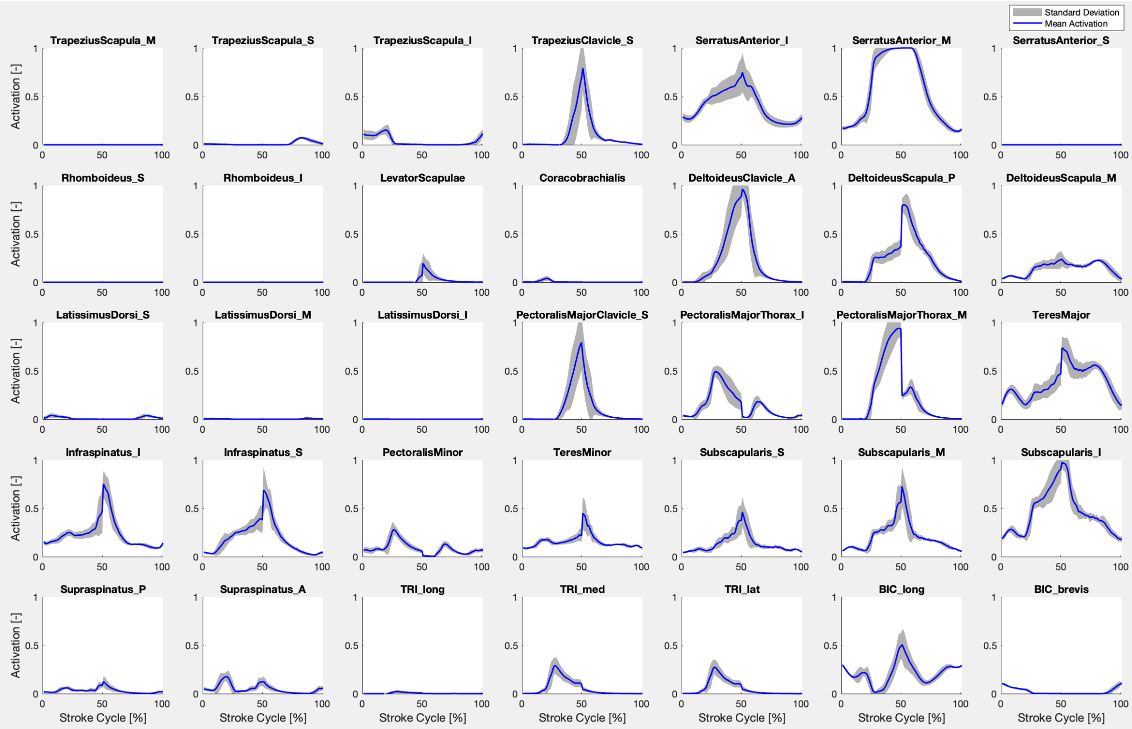


Figure B.11: Average muscle activation of participant 12 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

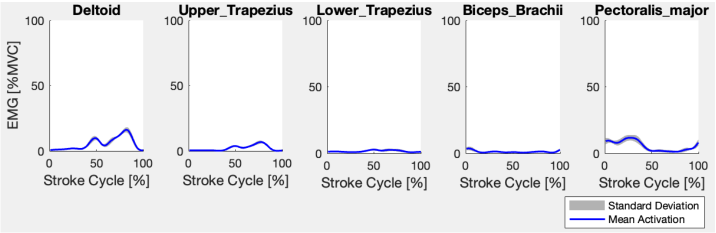


Figure B.12: Average EMG%MVC of participant 12 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

Participant 25

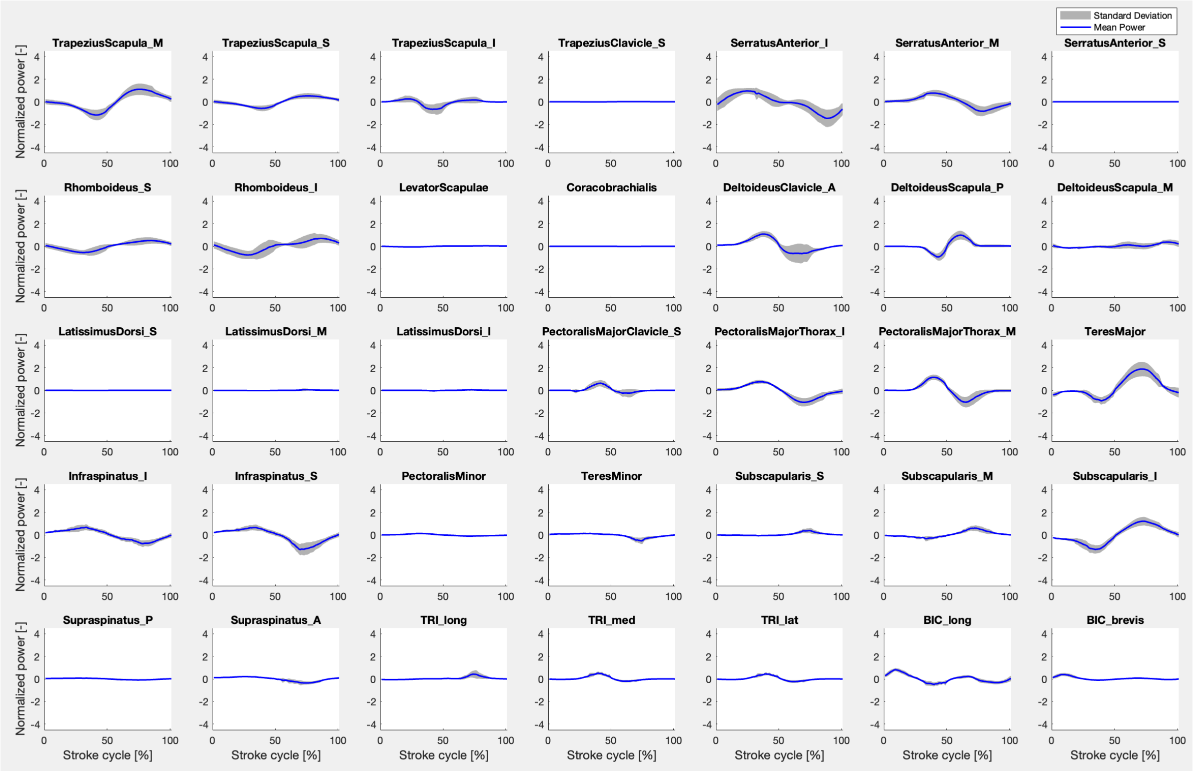


Figure B.13: Average muscle power of participant 25 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

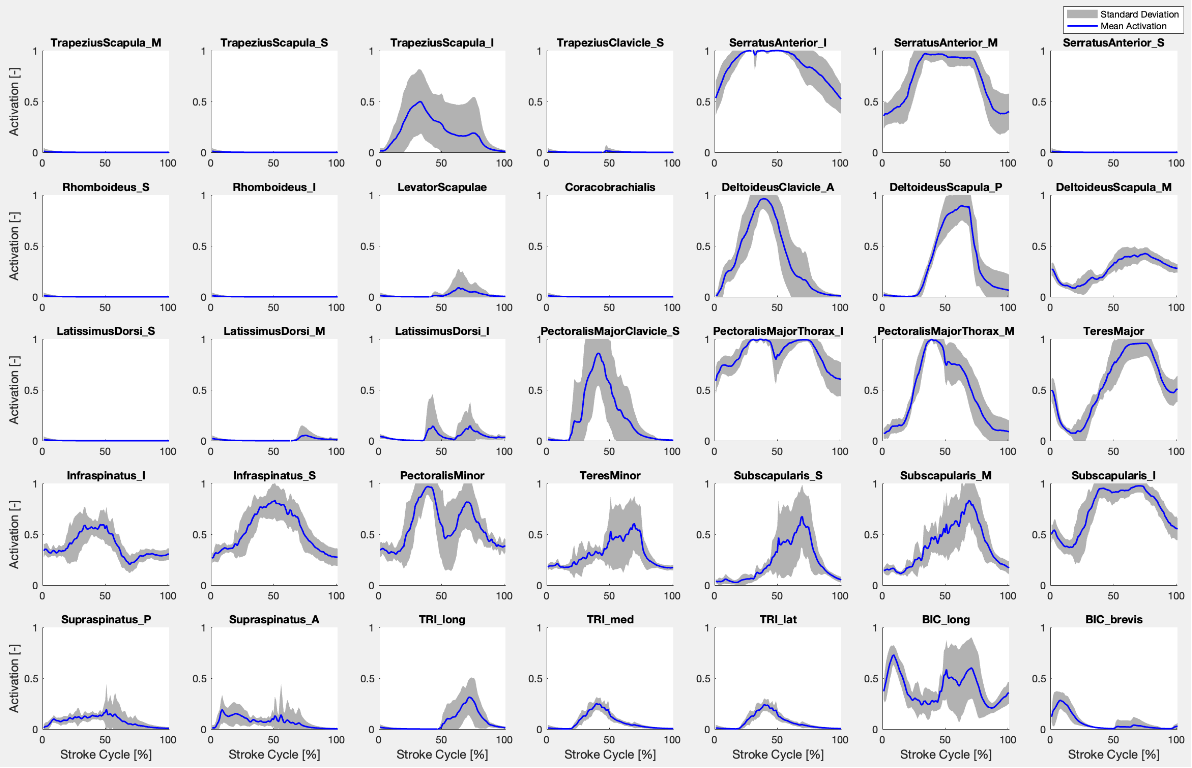


Figure B.14: Average muscle activation of participant 25 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

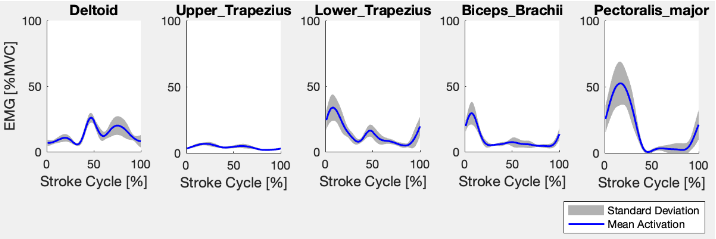


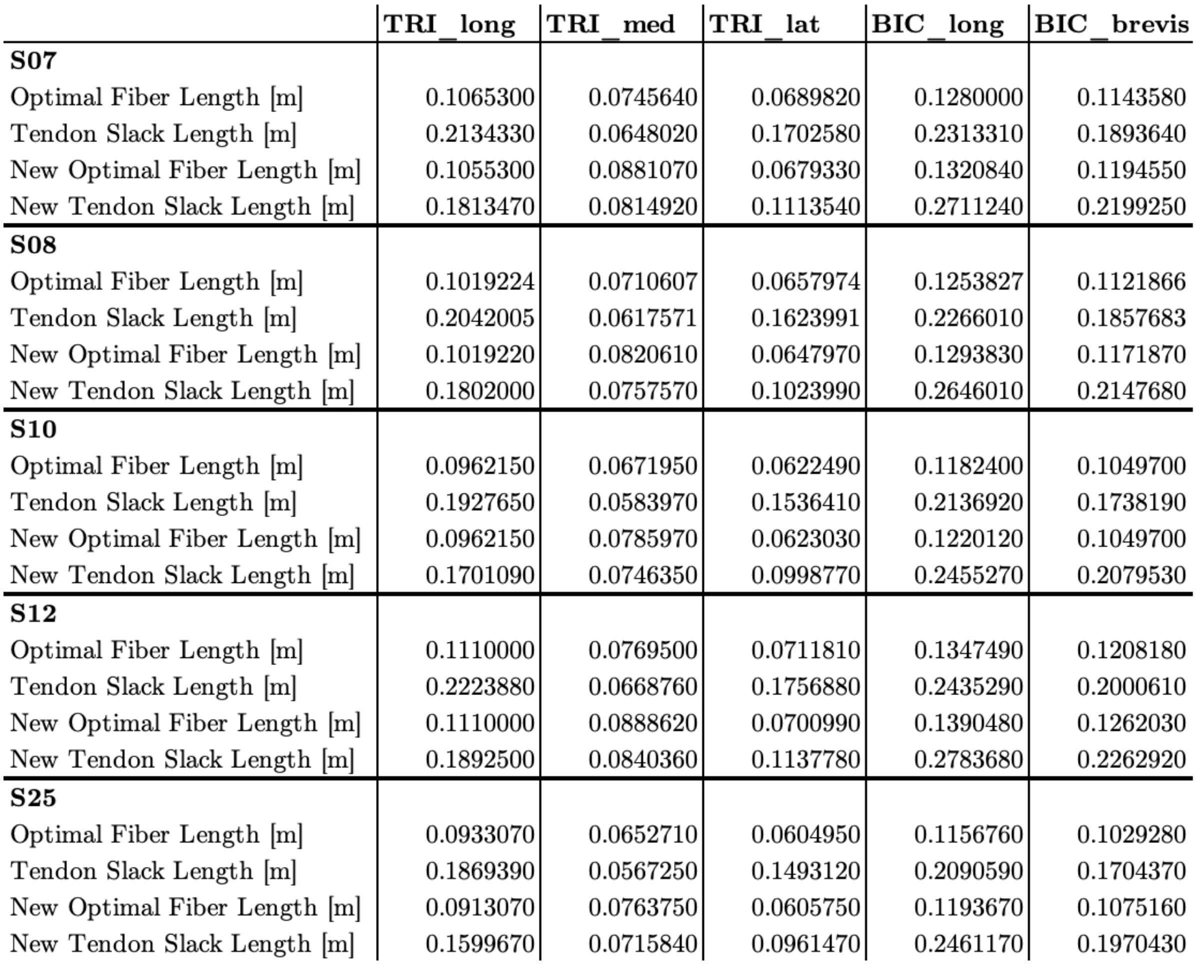
Figure B.15: Average EMG%MVC of participant 25 throughout a stroke cycle, presented with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

C

Muscle Adjustments and Coordinate Accelerations

Muscle Adjustments

Table C.1: Muscle adjustments to the triceps brachii long, medial and lateral, and to the biceps brachii long and brevis are shown for each participant. The old optimal fiber lengths and tendon slack lengths as well as the new adjusted lengths are shown.



	TRI_long	**TRI_med**	**TRI_lat**	**BIC_long**	**BIC_brevis**
S07					
Optimal Fiber Length [m]	0.1065300	0.0745640	0.0689820	0.1280000	0.1143580
Tendon Slack Length [m]	0.2134330	0.0648020	0.1702580	0.2313310	0.1893640
New Optimal Fiber Length [m]	0.1055300	0.0881070	0.0679330	0.1320840	0.1194550
New Tendon Slack Length [m]	0.1813470	0.0814920	0.1113540	0.2711240	0.2199250
S08					
Optimal Fiber Length [m]	0.1019224	0.0710607	0.0657974	0.1253827	0.1121866
Tendon Slack Length [m]	0.2042005	0.0617571	0.1623991	0.2266010	0.1857683
New Optimal Fiber Length [m]	0.1019220	0.0820610	0.0647970	0.1293830	0.1171870
New Tendon Slack Length [m]	0.1802000	0.0757570	0.1023990	0.2646010	0.2147680
S10					
Optimal Fiber Length [m]	0.0962150	0.0671950	0.0622490	0.1182400	0.1049700
Tendon Slack Length [m]	0.1927650	0.0583970	0.1536410	0.2136920	0.1738190
New Optimal Fiber Length [m]	0.0962150	0.0785970	0.0623030	0.1220120	0.1049700
New Tendon Slack Length [m]	0.1701090	0.0746350	0.0998770	0.2455270	0.2079530
S12					
Optimal Fiber Length [m]	0.1110000	0.0769500	0.0711810	0.1347490	0.1208180
Tendon Slack Length [m]	0.2223880	0.0668760	0.1756880	0.2435290	0.2000610
New Optimal Fiber Length [m]	0.1110000	0.0888620	0.0700990	0.1390480	0.1262030
New Tendon Slack Length [m]	0.1892500	0.0840360	0.1137780	0.2783680	0.2262920
S25					
Optimal Fiber Length [m]	0.0933070	0.0652710	0.0604950	0.1156760	0.1029280
Tendon Slack Length [m]	0.1869390	0.0567250	0.1493120	0.2090590	0.1704370
New Optimal Fiber Length [m]	0.0913070	0.0763750	0.0605750	0.1193670	0.1075160
New Tendon Slack Length [m]	0.1599670	0.0715840	0.0961470	0.2461170	0.1970430

Estimated and Simulated Coordinate Accelerations

Table C.2: The estimated coordinate accelerations by the RMR solver (RMR) and one standard deviation of the measured coordinate accelerations (1 std) are shown for each coordinate for all participants. The mean of the accelerations and one standard deviation of the rotational coordinates and translational coordinates are shown at the end of the table.

	S07		**S08**		**S10**		**S12**		**S25**	
	RMR	**1 std**	**RMR**	**1 std**	**RMR**	**1 std**	**RMR**	**1 std**	**RMR**	**1 std**
ground_thorax_rot_x [deg/s^2]	0.000	1.362	0.034	1.036	0.008	1.532	0.029	0.997	0.021	1.583
ground_thorax_rot_y [deg/s^2]	0.000	1.281	0.083	1.712	0.014	1.791	0.012	1.292	0.050	2.982
ground_thorax_rot_z [deg/s^2]	0.000	6.119	0.362	4.401	0.019	4.066	0.029	4.174	0.090	8.798
ground_thorax_tx [m/s^2]	0.000	0.984	0.068	0.699	0.002	0.354	0.009	0.774	0.014	1.022
ground_thorax_ty [m/s^2]	0.000	0.448	0.006	0.242	0.003	0.254	0.003	0.281	0.008	0.535
ground_thorax_tz [m/s^2]	0.000	0.128	0.007	0.160	0.003	0.283	0.001	0.137	0.003	0.227
clav_prot [deg/s^2]	0.207	3.378	0.519	3.704	0.469	1.653	0.315	2.884	0.478	4.271
clav_elev [deg/s^2]	0.344	3.950	2.471	3.036	0.571	1.311	2.364	3.000	0.966	2.163
scapula_abduction [deg/s^2]	0.000	6.632	0.609	7.044	0.059	4.559	0.101	5.529	0.279	8.417
scapula_elevation [deg/s^2]	0.000	2.537	0.276	0.747	0.051	4.406	0.014	1.483	0.542	1.540
scapula_upward_rot [deg/s^2]	0.001	3.908	0.261	3.483	0.045	3.292	0.091	4.126	0.230	5.175
scapula_winging [deg/s^2]	0.295	2.125	1.179	1.971	1.886	2.250	1.667	1.777	0.664	1.969
plane_elv [deg/s^2]	0.001	27.468	2.020	24.036	0.065	16.288	0.219	28.686	0.369	24.806
shoulder_elv [deg/s^2]	0.000	17.537	0.765	9.347	0.033	4.109	0.210	10.275	0.213	6.009
axial_rot [deg/s^2]	0.001	33.320	1.823	25.687	0.095	14.936	0.335	23.211	0.498	29.151
elbow_flexion [deg/s^2]	0.000	28.166	2.911	35.438	0.251	26.114	0.281	31.315	0.597	36.260
pro_sup [deg/s^2]	0.000	14.171	0.517	9.516	0.103	13.251	0.084	10.808	0.304	9.506
Rotational Coordinates [deg/s^2]	**0.001**	**6.213**	**0.559**	**5.004**	**0.083**	**4.463**	**0.134**	**5.021**	**0.262**	**5.986**
Translational Coordinates [m/s^2]	**0.000**	**0.384**	**0.014**	**0.300**	**0.002**	**0.294**	**0.004**	**0.310**	**0.007**	**0.499**

D

Estimated reserve actuator power

Participant 07

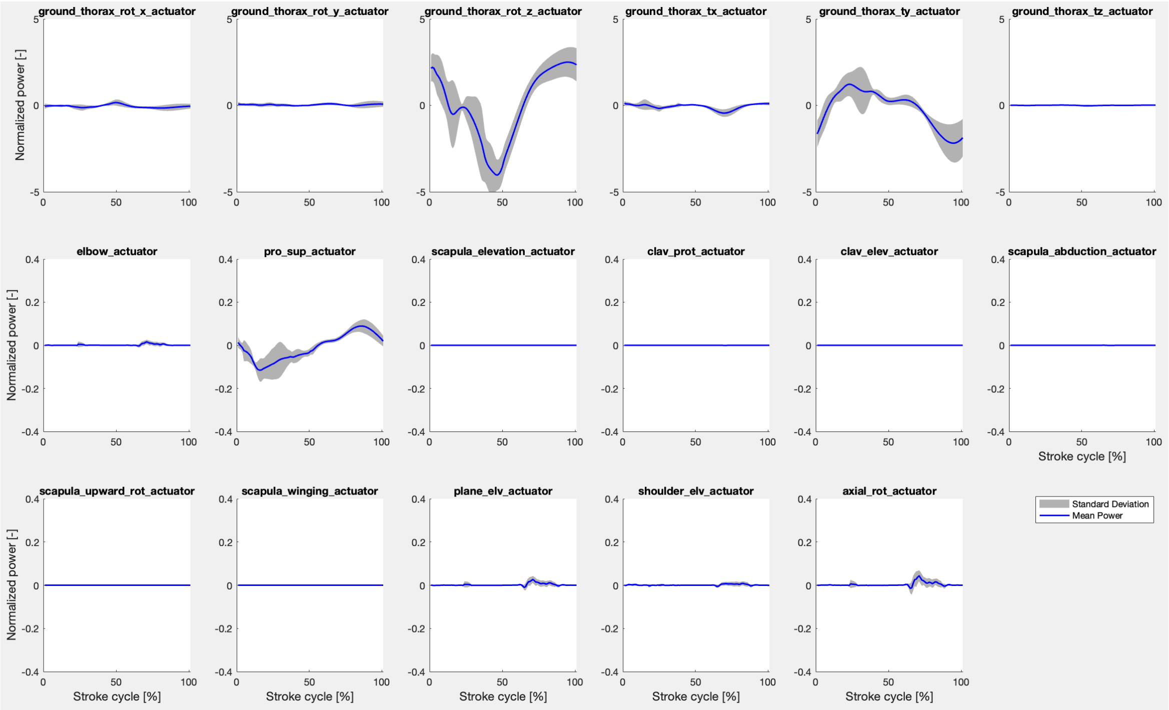


Figure D.1: Mean normalized reserve actuator's power of participant 07, with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

Participant 08

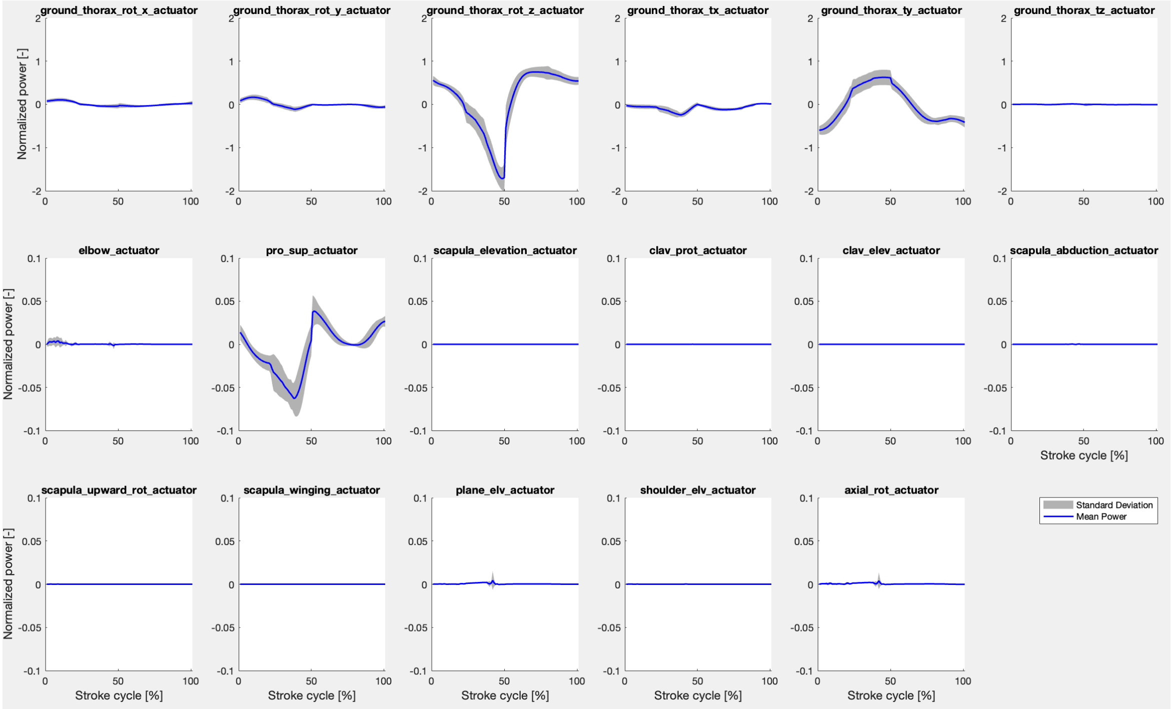


Figure D.2: Mean normalized reserve actuator's power of participant 08, with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

Participant 10

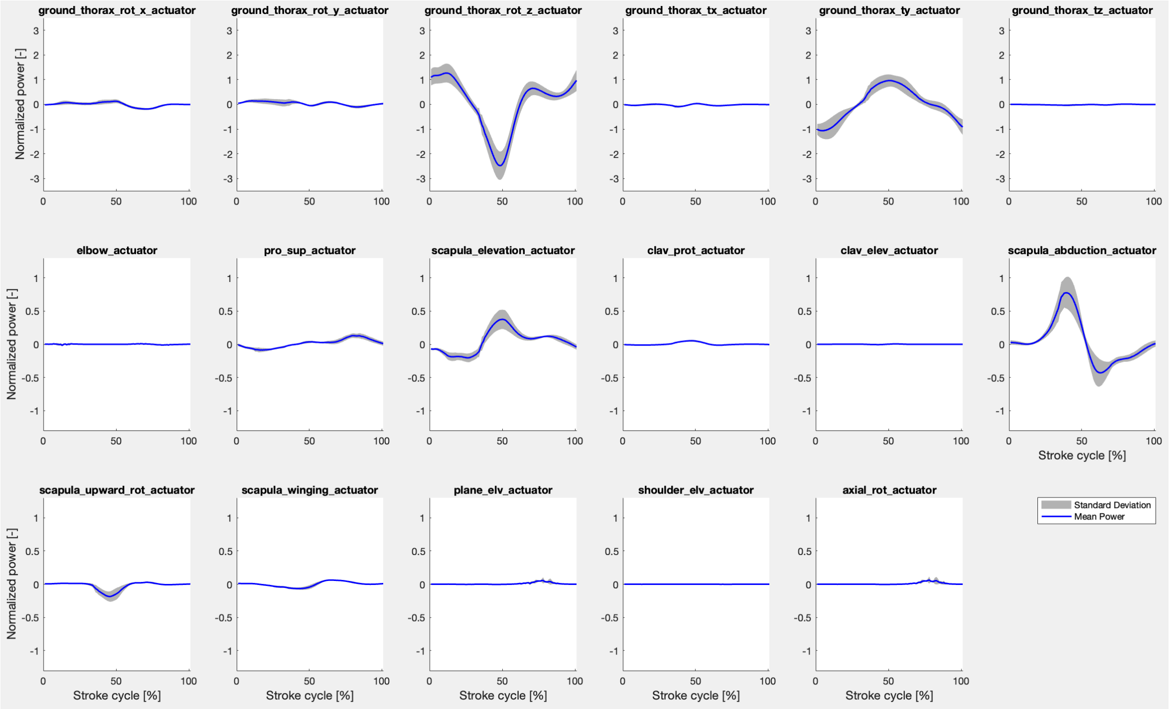


Figure D.3: Mean normalized reserve actuator's power of participant 10, with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

Participant 12

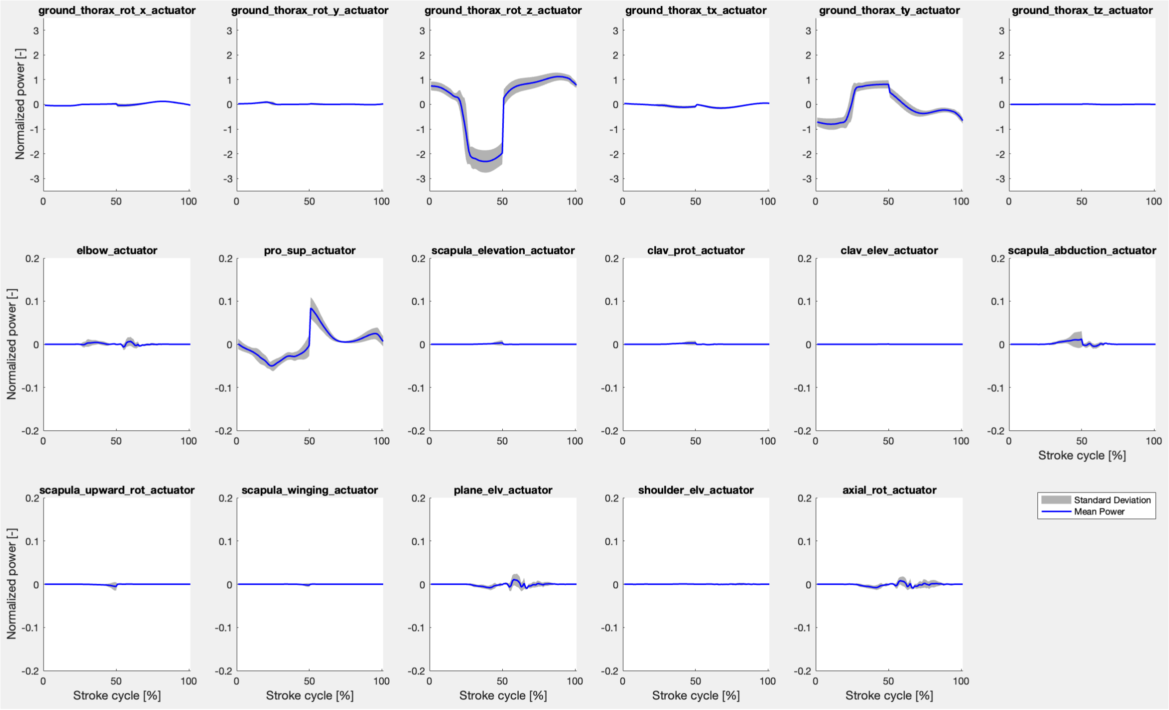


Figure D.4: Mean normalized reserve actuator's power of participant 12, with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.

Participant 25

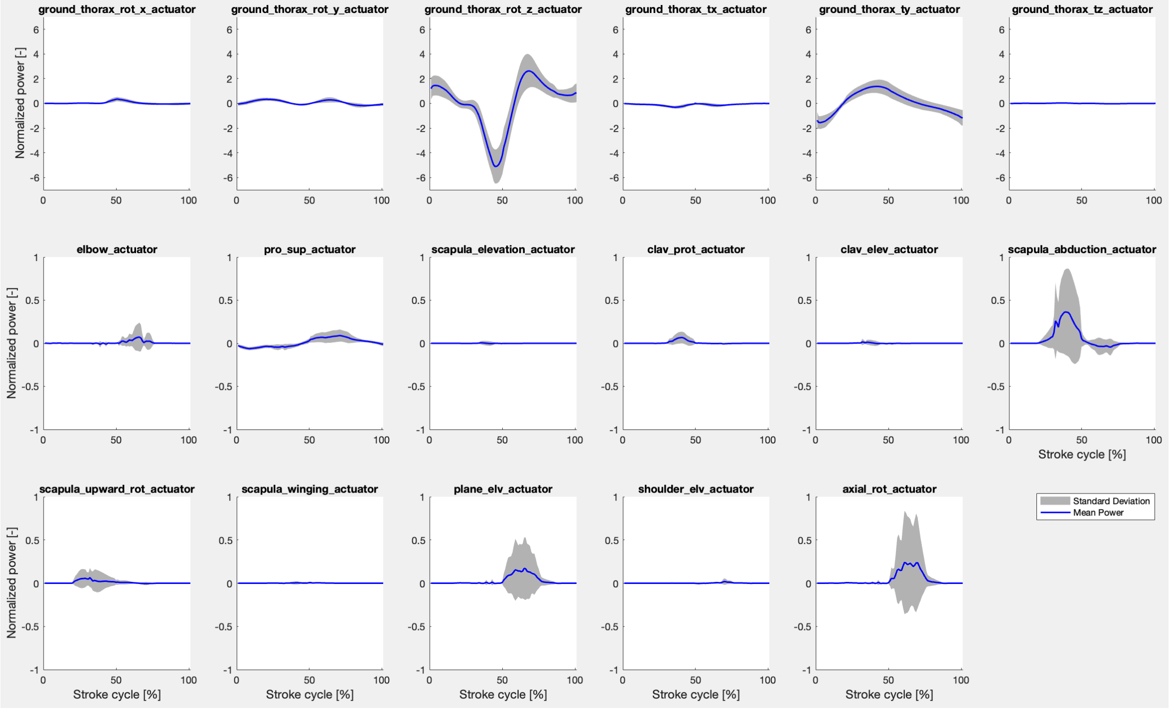


Figure D.5: Mean normalized reserve actuator's power of participant 25, with standard deviation depicted by the shaded grey area. The first 50% of the cycle corresponds to the push phase, while the latter 50% denotes the recovery phase.