



Master Thesis

**Performance of
A Novel Manual Wheelchair Propulsion Device
Compared to Conventional Pushrim Propulsion
– Benchmarking Mechanical Efficiency and
Movement Characteristics**

carried out for the purpose of obtaining the degree of Master of Science (MSc or Dipl.-Ing. or DI),
submitted at TU Wien, Faculty of Mechanical and Industrial Engineering, by

Anja Arbter

Mat.Nr.: 01230372

Vöslauerstraße 26, 2500 Baden, Austria

under the supervision of

Univ.Prof. Dipl.-Ing. Dr.techn. Margit Gföhler

Institute of Engineering Design and Product Development

Reviewed by

Univ.Prof. Dipl.-Ing. Dr.techn. Margit Gföhler

Institute of Engineering Design and Product Development

Getreidemarkt 9, 1060 Vienna, Austria

This work was supported by TU Wien, Biomechanical
Optimization and Design of a Novel Device for Manual
Wheelchair Propulsion (P 25507-B24)

Affidavit

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume.

Vienna, 19 June 2019

Signature

Abstract

Biomechanical research has previously highlighted musculoskeletal overuse injuries of the upper extremity of long-term wheelchair use. Scientists suspect that one of the reasons is repetitive mechanical loading during wheelchair propulsion. It was proposed that optimizing wheelchair propulsion may result in lowering shoulder load, and therefore the risk of overuse injuries of the shoulder complex. This thesis describes the methodological approach used in the post-processing and analysis of a measurement series on pushrim and crank propulsion of wheelchair users. Using data from a measurement series conducted by *TU Wien* at 25 W and 35 W power output, the methodological approach was applied. The results of kinematic, kinetic, spirometric and electromyographic data analysis and the comparison of the two propulsion techniques confirm that the novel handle-based propulsion mechanism may be a valuable alternative for long-term wheelchair users who are suffering from joint injuries. The presented methodological approach and the developed Matlab codes generally suited the requirements in post-processing and analysis, and thus provides an applicable procedure for future measurement series.

Biomechanische Forschung zeigt muskuloskeletale Probleme der oberen Extremitäten bei der Langzeitbenutzung des Rollstuhlantriebs auf. Forscher vermuten, dass einer der Gründe die repetitive mechanische Belastung während des Rollstuhlantriebs darstellt. Es wird angenommen, dass die Optimierung des Rollstuhlantriebs zur Senkung der Schulterbelastung und damit des Risikos von Verletzungen durch Überbeanspruchung des Schulterkomplexes führen kann. Diese Arbeit beschreibt den methodischen Ansatz, der bei der Nachbearbeitung und Analyse der Messreihe an Rollstuhlfahren zum Einsatz kam. Die Messungen wurden im Zuge eines Projekts an der *TU Wien* an Rollstuhlfahrern zum Vergleich von Greifring- und Kurbel-Antrieb bei 25 Watt und 35 Watt durchgeführt. Die Ergebnisse der kinematischen, kinetischen, spirometrischen und elektromyographischen Daten und der Vergleich von Kurbel-Antrieb mit dem Greifring-Antrieb zeigen, dass der verwendete neuartige Kurbel-Antrieb eine wertvolle Alternative für Rollstuhlfahrer mit Gelenksbeschwerden sein kann. Der methodische Ansatz entsprach den Anforderungen der Datenaufbereitung und -analyse und die *Matlab* Codes bieten ein anwendbares Verfahren für zukünftige Messreihen.

List of Abbreviations

BB	backbone
BI	musculus biceps brachii
bpm	beats per minute
CL	articulatio sternoclavicularis
CoF	centre of force
DA	musculus deltoideus anterior
deg, °	degree
DM	musculus deltoideus medialis
DOF	degree of freedom
DP	musculus deltoideus posterior
EL1	elbow lateral epicondyle
EL2	elbow medial epicondyle
EMG	electromyography
En, Eeh	energy expenditure
FA	forearm
FI	index finger, 2 nd metacarpophalangeal joint
F _x , F _y , F _z	force in x-, y-, z-direction
GCOS	global coordinate system
HC	handle centre
HCOS	handle coordinate system
HF	heart frequency
HH	sliding guide of handle
HT	handle top
Hz	Hertz
Infra	Infraspinatus
J	Joules
ME	mechanical efficiency
Ms	millisecond
m/s	meter per second
MVC	maximum voluntary contraction
m	meter
mm	millimeter

M_x, M_y, M_z	torque in x-, y-, z-direction
P	musculus pectoralis
PO	power output
PU	portable unit
rad	radian
RER	respiratory exchange ratio
RMS	root mean square
ROM	range of motion
SC	SmartWheel centre
SH	shoulder, acromion process
SR	edge of SmartWheel
T La	musculus triceps caput laterale
T Lo	musculus triceps caput longum
Tmin	Teres Minor
V	Volt
V_{CO_2}	oxygen uptake
V_{O_2}	carbon dioxide production
W	Watt
WR	wrist, processus styloid radialis
WU	wrist, processus styloid ulnaris
%	percent

Table of Contents

Abstract	5
List of Abbreviations.....	7
1. Introduction	11
2. Aim of Thesis	20
3. Methods.....	21
3.1 Wheelchair setup	21
3.1.1 Test rig	21
3.1.2 Propulsion devices	23
3.2 Devices for Data Collection	25
3.2.1 Force Handle	25
3.2.2 SmartWheel.....	25
3.2.3 Electromyography.....	26
3.2.4 Motion capturing.....	26
3.2.5 Spirometry setup	26
3.3 Software for data capturing, post-processing and analysis.....	27
3.4 Measurement Series at <i>Rehabilitation Centre Weißer Hof</i>	29
3.4.1 Measurement Setup.....	29
3.4.2 Subjects	30
3.4.3 Testing protocol	31
3.5 Data processing and analysis	35
3.5.1 Spirometry analysis.....	38
3.5.1 Post-processing of motion data in <i>Cortex</i>	39
3.5.2 Post-processing of kinetic and kinematic data in <i>Matlab</i>	39
3.5.3 EMG post-processing and analysis	48
3.5.5 Analysis in <i>OpenSim</i>	51
4. Results	54
4.1 Mechanical efficiency and heart frequency.....	54
4.1.1 Results of wheelchair users.....	54
4.1.2 Wheelchair users vs. non-wheelchair users	57
4.2 EMG	61
4.3 Kinematic Analysis with OpenSim	66
5. Discussion	73

6. Bibliography.....	82
7. List of Figures	85
8. List of Tables.....	88
9. Annexes.....	89
9.1 Annex 1: Matlab Codes	89
9.1.1 Handle Code.....	89
9.1.2 SmartWheel Code	96
9.1.3 EMG code	102
9.1.4 On-/Off-Code.....	107
9.1.5 OpenSim Code	109
9.3 Annex 3: Testing Protocols	113
9.2 Annex 3: Frame File	114

1. Introduction

According to the World Health Organization (WHO), 10 % of disabled persons, representing 10 % of the global population, require a wheelchair for daily use [1]. Wheelchairs provide mobility to people who suffer from injuries, such as spinal cord injuries, and enhance quality of life. Nevertheless, wheelchair users often develop overuse injuries of the upper extremity [1]. The development of wheelchairs that not only fit the wheelchair user's needs, but also prevent the user from acquiring injuries due to their usage, is of huge importance.

Wheelchair development has changed focus over the years, firstly looking at mechanical aspects and then shifting to a biomechanical point of view. Over time, it has highlighted the prevalence of musculoskeletal problems due to long-term wheelchair use [2]. This leads to a higher incidence of shoulder, elbow and wrist joint issues [3]. The likelihood of wheelchair users to developing carpal tunnel syndrome, a vulnerable injury caused in the wrist region, is between 49 % and 73 % [4]. Pain in the upper extremity area is a serious problem. Because this area is of huge importance in daily life activities, these injuries may lead to loss of independence, increased cost of care and additional secondary injuries [5]. Researchers suggest that these injuries may be the result of the highly repetitive mechanical loading during wheelchair propulsion [3].

In the following paragraphs, the methods of motion analysis, the biomechanics of the arm, as well as the arm muscles are firstly discussed, before the movement pattern of wheelchair propulsion is further elaborated.

Motion analysis is a measurement technique that evaluates human movement. Data from joint movement, timing and coordination of muscle activity and forces are obtained. With this information, the efficiency of movement patterns can be analysed and, as a result, further optimized. Different devices and methods, such as video recording, EMG detection and force detection, as well as software modelling, may be applied.[6]

For motion recording, reflective markers are attached to the body whose location is detected by a three-dimensional camera system. Special computer software analyses this information and generates a 3D model. Additionally, EMG sensors may be taped to muscle bellies to measure their electrical activity. The obtained signals are transmitted to a corresponding software. Furthermore, special force measuring devices, such as force plates, can be used to record forces applied during a special movement. Ideally, computer software automatically synchronizes the

data from the cameras, EMG sensors and force measuring devices and provides graphs for further analysis. In addition, biomechanical models for the simulation of movements may be used in specific computer software. Graphs may show movement, joint angles, muscle activity and force production during movement. These results may be compared with previous data and provide valuable input for further treatment or development of devices.[6]

When taking a closer look at the biomechanics of the arm, the three major joints, which are the shoulder joint, the elbow joint and the wrist joint, must be considered. For better understanding of the following tables, Figure 1 shows the arm muscles.

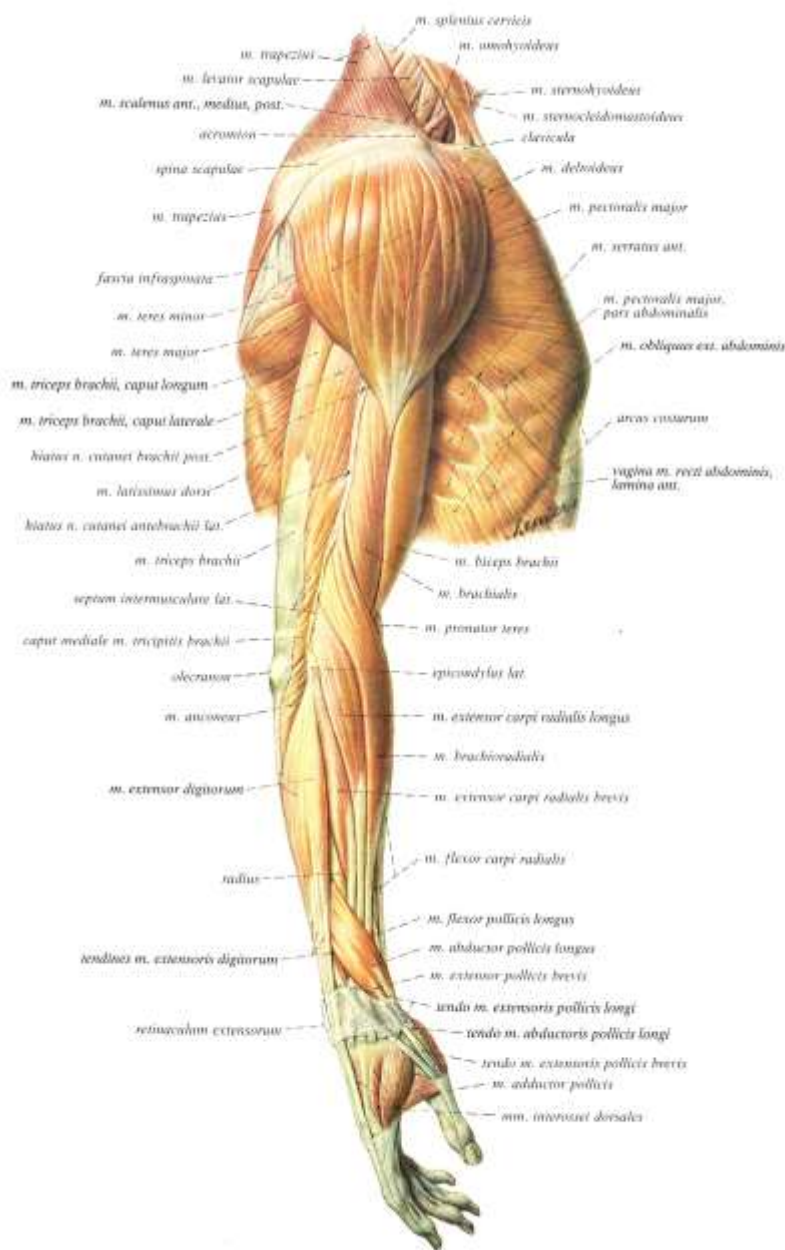


Figure 1: Muscles of the upper extremity. [7]

The shoulder joint is the proximal joint of the upper limb and it is considered to be the most mobile joint of the human body. As it is a ball-and-socket joint, movement of the upper limb within the three planes in space and within the three major axes is possible. They are described as follows (Figure 2): Flexion and extension in the sagittal plane, abduction (limb moves away from the body) and adduction (limb moves towards the body) performed in the frontal plane, internal and external rotation in a horizontal plane with the elbow flexed to 90° (rotation of the arm about its long axis), and horizontal flexion and extension in the horizontal plane.

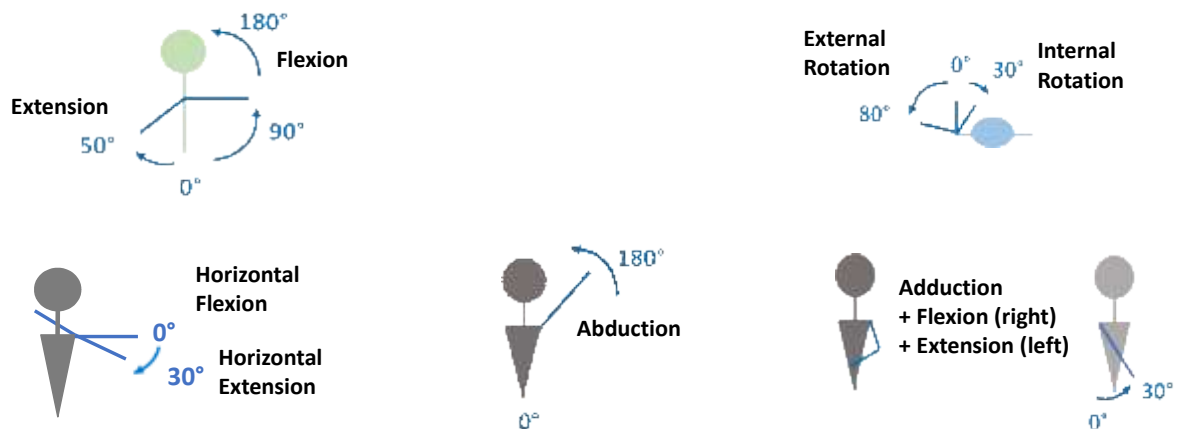


Figure 2: Sketch of extension, flexion, external rotation, internal rotation, abduction and adduction movement of the shoulder joint.

The reference position of flexion, extension, abduction and adduction of the shoulder joint is defined as when the arm is hanging vertically alongside to the body. Flexion is a movement of great range with up to 180°. In contrast, extension is a movement of small range up to 45-50°. The phase between 60° to 120° is reached due to the rotation of the scapula, and the final phase of flexion (120-180°) is achieved due to the movement of the spinal cord.

Adduction may only be performed with the combination of extension (small adduction movement possible) or flexion (30-45°). Abduction can be divided into three phases with the first movement taking place only in the shoulder joint (0-60°). Further abduction requires the recruitment of the scapula (60-120°) and flexion of the trunk to the opposite side (120-180°). The arm's position at 180° of abduction is equal to that at 180° of flexion. Pure abduction is rare, mostly it is combined with flexion.

0° of shoulder rotation marks the position where the arm is held loose with an elbow angle of 90° [8]. Positive angle values of shoulder rotation represent the internal rotation of the shoulder joint (up to 30°), whereas negative angle values indicate the external shoulder rotation (up to

80°). Fingers point across the body during an internal rotational movement, while they point away from the body during an external rotation [8].

For horizontal flexion and extension, the reference position is when the arm is adducted 90° in the frontal plane. During horizontal flexion, the limb moves towards the body in horizontal plane (up to 140°), during horizontal extension, it moves away from the body in horizontal plane (30-40°).

Table 1 shows the muscles involved in the abovementioned movements of the shoulder joint.

Shoulder movement	Muscles involved
Flexion	Musculus deltoideus (anterior, medialis), musculus biceps brachii, musculus pectoralis major, musculus coracobrachialis, musculus serratus anterior
Extension	musculus teres major, musculus latissimus dorsi, musculus triceps brachii caput longum, musculus deltoideus (posterior)
External rotation	musculus infraspinatus, musculus teres minor, musculus deltoideus (posterior)
Internal rotation	musculus subscapularis, musculus pectoralis major, caput longum of musculus biceps brachii, musculus deltoideus (anterior), musculus teres major, musculus latissimus dorsi
Adduction	musculus pectoralis major, musculus triceps brachii caput longum, musculus teres major, musculus latissimus dorsi, caput breve of musculus biceps brachii, musculus deltoideus (anterior, posterior)
Abduction	musculus deltoideus, musculus supraspinatus, caput longum of musculus biceps brachii, musculus serratus anterior, musculus trapezius

Table 1: Muscles involved in flexion, extension, external rotation, internal rotation, adduction and abduction of the shoulder joint.[9]

The elbow joint is the intermediate joint of the upper extremity, linking the upper arm to the forearm. It allows flexion and extension and is also involved in the pronation and supination movement. The reference position is reached when the axes of the arm and forearm are in a straight line. During extension, the forearm moves posterior, during flexion it moves anterior.

Actively, the joint may reach an angle of 145° and passively, even 160°. Table 2 shows the muscles involved in elbow joint movement.

Elbow movement	Muscles involved
Flexion	musculus biceps brachii (mainly), musculus brachialis, musculus brachioradialis, musculus extensor carpi radialis longus, musculus pronator teres
Extension	musculus triceps brachii (mainly caput laterale, caput mediale)
Pronation	musculus pronator quadratus, musculus pronator teres, musculus flexor carpi radialis, musculus extensor carpi radialis longus, musculus brachioradialis, musculus palmaris longus
Supination	musculus supinator, musculus biceps brachii, musculus abductor pollicis longus, musculus extensor pollicis longus, musculus brachioradialis

Table 2: Muscles involved in flexion, extension, pronation and supination movement of the elbow joint [9].

The wrist joint is the distal joint of the upper extremity and has two DOF. Flexion (palmar surface of the hand moves towards the anterior aspect of the forearm; positive angle values) and extension (dorsal surface of the hand moves towards the posterior aspect of the forearm; negative angle values) may be performed in the sagittal plane. Wrist deviation is separated into ulnar deviation (positive values) and radial deviation (negative values) of the wrist. Looking at the right hand from the palmar side, ulnar deviation defines the little finger movement in the frontal plane towards the ulna whereas radial deviation defines the thumb movement in frontal plane towards the radius, as displayed in Figure 3 [6].

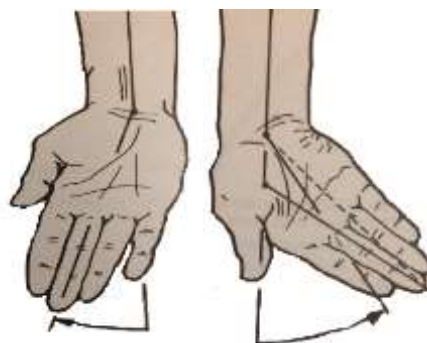


Figure 3: Ulnar deviation (right) and radial deviation (left) of the wrist joint. [8]

Table 3 shows the muscles involved in wrist joint movement.

Wrist movement	Muscles involved
Dorsalflexion	musculus extensor digitorum, musculus extensor carpi radialis longus, musculus carpi radialis brevis. Musculus extensor indicis, musculus pollicis longus, musculus extensor digiti minimi
Palmarflexion	musculus flexor digitorum superficialis, musculus digitorum profundus, musculus flexor carpi ulnaris, musculus flexor pollicis longus, musculus flexor carpi radialis, musculus abductor pollicis
Radial deviation	musculus extensor carpi radialis longus, musculus pollicis longus, musculus extensor pollicis longus, musculus flexor carpi radialis, musculus flexor pollicis longus.
Ulnar deviation	musculus extensor carpi ulnaris, musculus flexor carpi ulnaris, musculus extensor digitorum, musculus extensor digiti minimi.

Table 3: Muscles involved in flexion, extension, radial deviation and ulnar deviation movement of the wrist joint.[9]

Based on the described basics of the biomechanics of the arm, the ergonomics of conventional wheelchair propulsion can be described as follows. In the push phase, the shoulder is flexed, adducted, internally rotated, the elbow is extended, and the wrist joint is moving in ulnar abduction. Pushrim propulsion is the most prevalently used wheelchair propulsion technique. Nevertheless, the pattern of this propulsion technique is seen as inefficient due to the discontinuous and complex upper extremity movements. Due to the contact and release pattern associated with pushrim propulsion, abrupt direction and speed changes can cause excessive strain on the joints and lead to high jerk forces [10]. Also, in order to maintain the propulsion cycle, joints exhibit a large range of motion and muscles produce relatively high forces which can result in injuries [11]. For example, shoulder injuries are exacerbated by high mechanical demand on the shoulder muscles that stabilize the high reaction forces at high speed and short contact duration [10]. Wrist injuries, like carpal tunnel syndrome, occur due to repetitive contact and release periods as well as due to the deviation of the wrist during propulsion, where

the median nerve within the carpal tunnel is compressed [11]. Radial and ulnar deviation lead to inflammation of tendon sheaths [11].

Optimizing wheelchair propulsion is a promising strategy for lowering shoulder load and therefore the risk of overuse injuries of the shoulder complex [3]. Alternative modes of manual propulsion, such as arm crank propulsion, have been taken into account. Currently, different hand propelled wheelchairs, such as lever driven wheelchairs and handcycles, are available on the market. These are becoming increasingly popular and are used in rehabilitation, outdoor activities and sports [2]. Arm crank propulsion is considered energetically more efficient and less “straining” for the cardiorespiratory system [3]. The force is applied uniformly over several muscle groups during one cycle, and low peak forces decrease the risk of overuse injuries of arm muscles. The usage of a lever mechanism is more effective in transmitting power than pushrim propulsion, as the arms are placed in a more natural position [12]. The levers are connected to the wheels by a chain and sprocket, and when the lever is pushed forward, a torque is transmitted to the wheels [12]. In contrast, no propulsive torque is generated whilst pulling the lever backwards [12]. Nevertheless, these propulsion devices have geometrical restrictions, such as large frame sizes, which make them inconvenient for daily use [10].

At *TU Wien*, a novel handle-based wheelchair propulsion mechanism was developed by Kurup et al. in [10] with an optimized propulsion pattern (Figure 4). This propulsion shape is adapted to the musculoskeletal architecture of the upper extremity and allows movement within the ergonomic joint ranges [4]. As a result, over-exertion of joints during propulsion is avoided, and this differs from pushrim propulsion as indicated on the right of Figure 4.

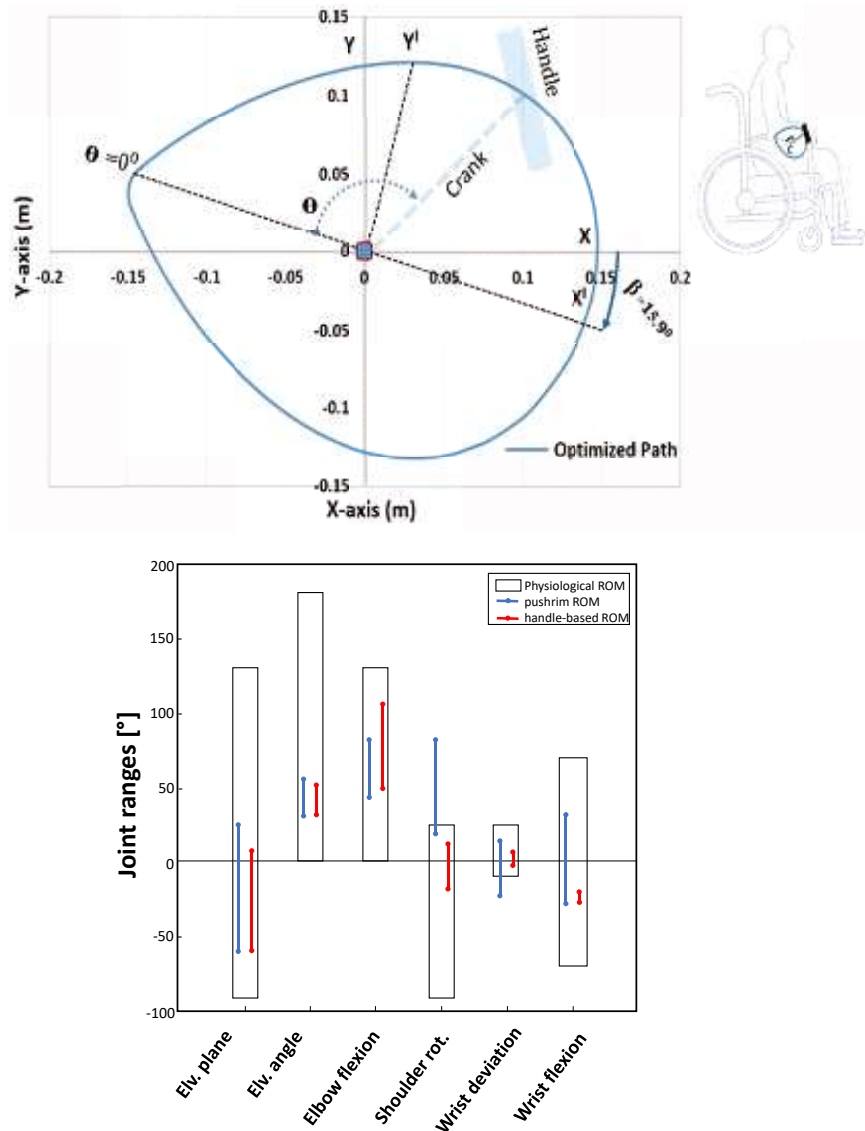


Figure 4: Dynamically optimized propulsion path (up) of novel wheelchair propulsion mechanism. Comparison of the joint range of motions (ROM) of the upper extremity between pushrim propulsion and handle-based propulsion (down).[10]

Figure 5 shows joint motion, such as shoulder flexion/extension, shoulder abduction/adduction (elevation angle), shoulder rotation, elbow flexion, wrist deviation and wrist flexion/extension, during handle-based propulsion. One turn of crank propulsion can be divided into a push-phase (hand moves away from the thorax) and a pull-phase (hand moves towards the thorax). During the pull-phase, the shoulder is extended, abducted and externally rotated. During the push phase, the shoulder is flexed, adducted and internally rotated. Furthermore, this propulsion movement is cyclic, continuous and improves hand contact whilst propelling the wheelchair. Also, agonist and antagonist muscles are alternatively activated, which may help to postpone local muscle fatigue of the upper extremity.[10]

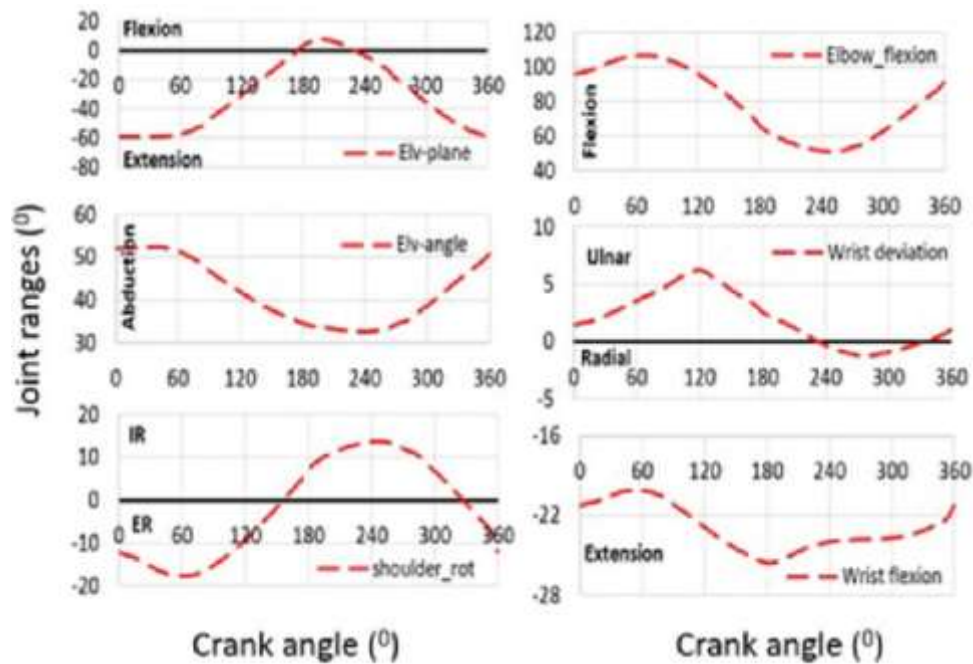


Figure 5: Joint motion during handle-based propulsion.[10]

The muscles activated during propulsion are similar in both propulsion techniques. *Musculus biceps brachii* and *musculus triceps brachii caput longum*, which are the major elbow muscles, exhibit a large range of both positive and negative work [13].

Kurup et al. showed in [10] that during the push phase, *musculus deltoideus anterior*, *musculus triceps brachii caput longum*, *musculus pectoralis*, *Infraspinatus* and *Teres Minor* were mainly excited. During the pull phase, mainly *musculus deltoideus posterior*, *musculus biceps brachii* and *Subscapularis* showed excitation. *Musculus deltoideus medialis* and *Supraspinatus* were active during the whole cycle.

The aforementioned study, which investigated whether this crank propulsion device could reduce the risk of injuries of the upper extremity by focusing on the joint kinematics and kinetics of the wrist joint, was carried out on non-wheelchair users [11]. This thesis continues this investigation on wheelchair users, and it furthermore post-processes and evaluates spirometric, electromyographic and kinematic data.

2. Aim of Thesis

In 2018, the research group *Functional Biomechanics and Rehabilitation Engineering* from *TU Wien* conducted a measurement series on wheelchair users at the *Rehabilitation Centre Weißer Hof*, comparing two different propulsion techniques: crank propulsion and pushrim propulsion. The aim of this thesis is to present a suitable methodological approach for the post-processing and analysis of the collected measurement data.

Pursuing this objective, the thesis will combine various software tools and apply them to different aspects of post-processing and analysis (*Methods*). The presented methodological approach will then be applied to the spirometric, electromyographic, kinematic and kinetic data of the *TU Wien* research study.

The obtained results on the differences between the two propulsion techniques are presented with respect to four parameters: mechanical efficiency, heart frequency, muscular activity and joint angles. Additionally, comparison to previous measurements of *TU Wien* on non-wheelchair users and other scientific sources are made (*Results*).

Both the methodological approach and the results will then be critically assessed (*Discussion*). Strengths and limitations of the presented approach are highlighted. The potential influence of the methodology on the analysis and the resulting findings is considered.

The thesis will end with a succinct summary and recommendations for future research.

3. Methods

This chapter introduces the hardware and software used in the performed measurements, the setup, subjects and test protocol of the measurement series at the *Rehabilitation Centre Weißer Hof* and the applied post-processing and analysing methodology of kinematic, kinetic, spirometric and electromyographic (EMG) data.

3.1 Wheelchair setup

The following subchapters describe the basis of this measurement series: the wheelchair-based test rig and the propulsion devices.

3.1.1 Test rig

For this measurement series, a wheelchair-based test rig is used, on which different hand driven propulsion devices can be easily mounted. This test rig was developed at *TU Wien*. Conventional pushrims are applied for pushrim propulsion whilst handles are installed for crank propulsion.

The wheelchair-based test rig for manual wheelchair propulsion allows easy and reproducible testing of different manual propulsion devices. As the setup uses timing belts for the transmission of resistance torques instead of friction rolls, it offers an advantage in its slip less measurement [15]. Slip less measurement is also the main difference to other experimental setups which alternatively use treadmills.

Going into more detail, the test rig is a lightweight, manual wheelchair of *Meyra Ortopedia* with a seat depth of 0.42 m, a seat width of 0.50 m and an altered seat height of 0.22 m [11]. The test rig consists of a wheelchair, a base frame with a mounting profile, a drive resistance unit, a motor-control unit and removable propulsion devices (Figure 6).

The wheelchair is raised and mounted on a square tube welding construct in order to avoid direct contact between the wheels and the floor. Four vibration dampening elements enable stable positioning and precise measurement during the whole test series (Figure 6). One of the main parts of the test rig wheelchair is the aluminium profile between the back tubes of the wheelchair. This ensures that the construct's position does not change during the propulsion movement and allows easy fixation of other modules, such as the drive resistance unit.[14]

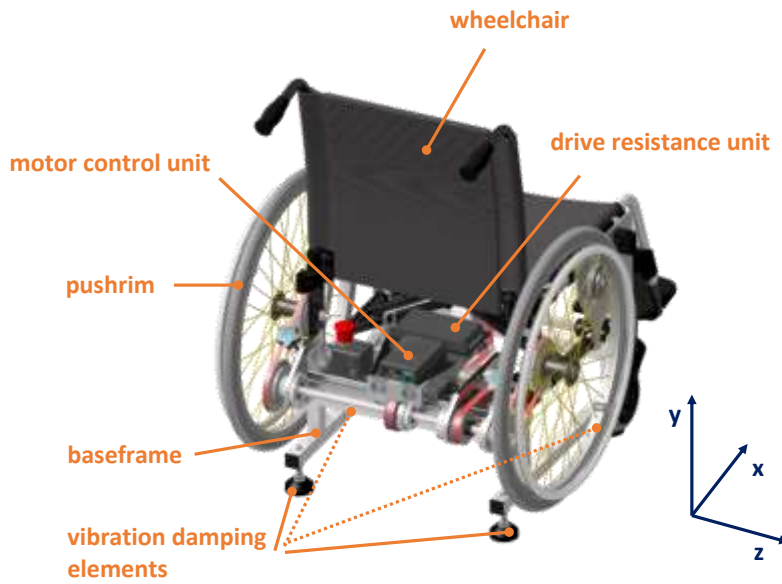


Figure 6: Test rig wheelchair with the capturing coordinate system.[14]

The drive resistance unit is composed of a brushless motor with a gearbox attached to one end of the motor shaft and a fly wheel on the other side of the motor (Figure 7). On both ends of the transmission shaft, pulleys enable mounting of conventional pushrim propulsion devices with timing belts linked to one of these pulleys. The timing belts allow slip free power transmission to the wheels and front connection points, and are set to optimal tension during the turning of the handwheel. The handles for crank propulsion are mounted using the timing belt in the middle of the shaft that is directed to the front of the wheelchair.[15]

The motor-control unit, fixed onto the mounting frame, consists of a power supply and a motor controller. A motor integrated hall sensor is used for control of the motor and delivers important speed information.[15]

During propulsion, the brushless motor runs in current mode and simulates real propulsion conditions. The angular velocity of the wheels is measured in real time and the torque required to maintain the targeted power is calculated. This torque is applied as a resistive torque by the motor to the wheels or crank propulsion device through the timing belts (Figure 7).[11]

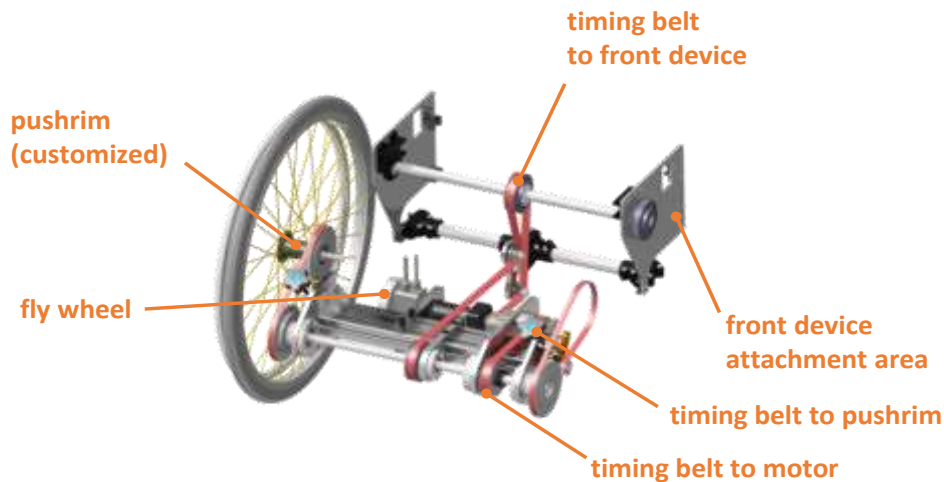


Figure 7: Drive-train of test rig wheelchair.[14]

3.1.2 Propulsion devices

For this measurement series, handles are installed for crank propulsion whilst conventional pushrims are applied for pushrim propulsion.

For crank propulsion, a special force measuring handle is linked to a sliding guide on the right-hand side. On the left-hand side, a handgrip without a measuring function, is installed. The propulsion unit consists of the shape plate engraved with the unique optimized propulsion shape, with a circulatory ratio of 0.89. The sliding guide can move back and forth on the plate and can result in a maximum crank length of 175 mm. This results in the novel propulsion movement for the handle.[11]

The handgrip allows a comfortable grip and rotates freely around an axis perpendicular to the crank. The mounting frame allows adjustment of the crank centre position to the sitting height and arm length of the subject.[11]

For pushrim propulsion, conventional pushrims are mounted. One of the pushrims is an instrumented *SmartWheel* which collects kinetic data like generated forces. On the other side, a pushrim wheel without a measuring function is mounted. Both wheels are adapted with a pulley for the timing belts of the test rig setup. An axel nut is used for the mounting and demounting process. Also, the crank propulsion unit has to be fixed at the outermost position of the mounting frame in order to avoid interference with the pushrim propulsion movement.

The upper image in Figure 8 shows the described propulsion units fixed to the test rig setup. The circle marks the location of the force measuring sensor. During measurement, one of the

devices is mounted on the test rig. The lower image displays the left-hand side handle, which has no measuring function, being linked to the sliding guide.

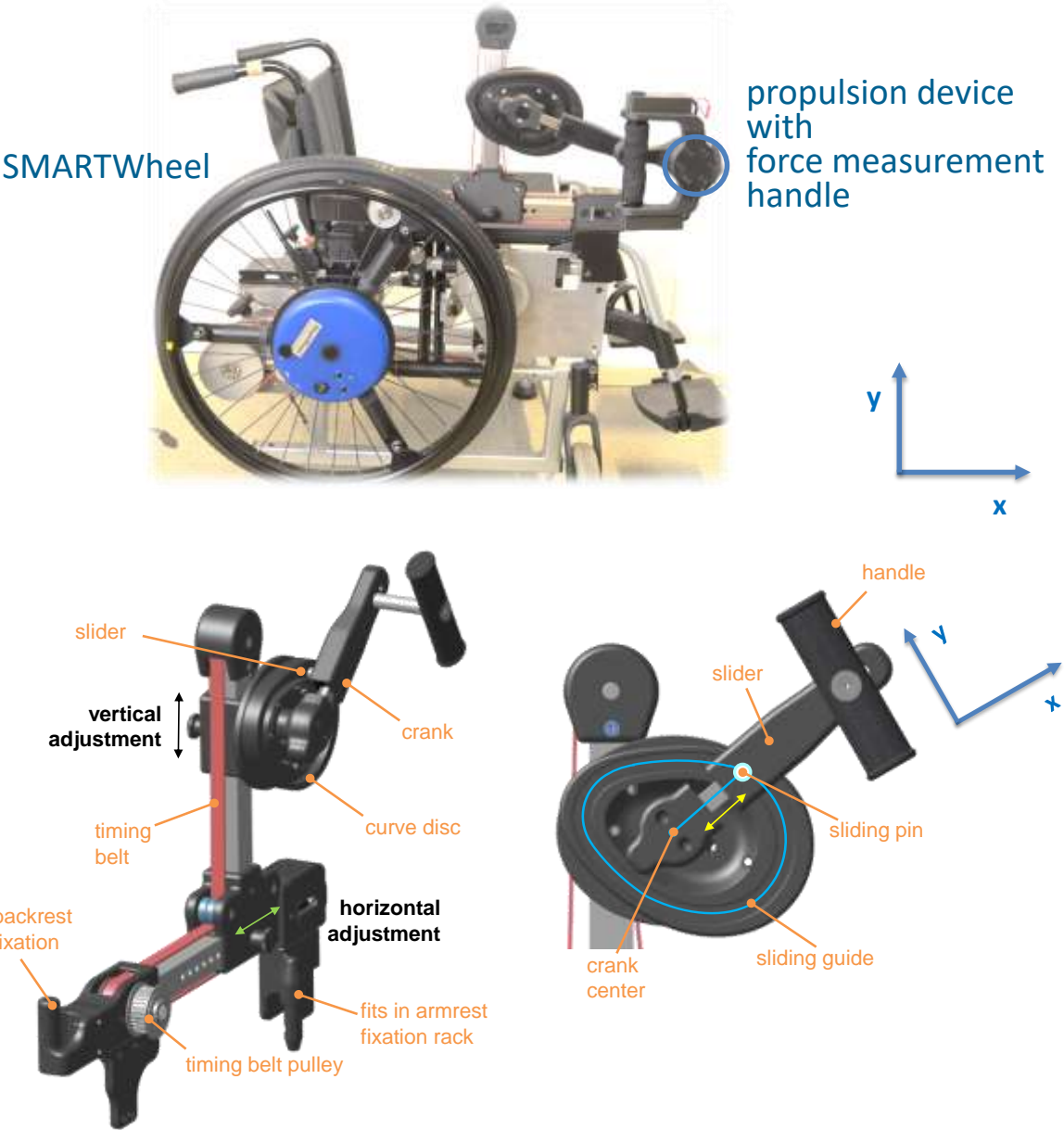


Figure 8: Test rig wheelchair with mounted SmartWheel and Handle, the circle marks the location of the forces measuring sensor (up). Crank propulsion device (down).[11]

3.2 Devices for Data Collection

The following subchapters describes the devices which are used for data collection.

3.2.1 Force Handle

For crank propulsion, a 3D printed special force measuring handle, developed at *TU Wien*, is used for real-time data recording on the right-hand side of the test rig. It is equipped with a force sensor (*6-axis force sensor K6D40, ME-Messsysteme GmbH*) which records three-dimensional forces (F_x , F_y , F_z) and torques (M_x , M_y , M_z) at 100 Hz [11]. The force-measuring handle is connected to the computer via Bluetooth.

3.2.2 SmartWheel

For examination of pushrim propulsion, the right pushrim wheel is replaced by a *SmartWheel* made by *Out-Front* (Figure 9, left). It is generally used in research and provides a better understanding of the physical demands of manual wheelchair use by analysing each push on the handrim. The measuring device is battery powered and it is available in different sizes. In this measurement series, a 24'' wheel is used.

During measurement, automated reports of the push forces, frequency, length, and speed (at a chosen frequency of 100 Hz) are created. The frequency can vary between 10 Hz to 240 Hz, depending on the other measurement devices that are used.[16]

The captured *SmartWheel* data is received via WLAN and permanently stored in the computer.



Figure 9: *SmartWheel* (left) and EMG device of Delsys with surface electrodes (right).

3.2.3 Electromyography

Electromyography (EMG) is used in order to examine the muscular activity during the propulsion movement. EMG is an experimental technique used to record and analyse voluntary myoelectric signals for physiological and biomechanical research purposes. EMG signals are formed by action potentials at the muscle fibres, which result from depolarization and repolarization processes. The depolarization-repolarization cycle forms a depolarization wave, or electrical dipole which travels along the surface of a muscle fibre and is detected by a bipolar electrode.[17]

For this measurement series, seven surface electrodes were applied (Figure 9, right), which are mainly used for kinesiological studies because of their non-invasive character and easy handling. Alternatively, needle electrodes may be used, as they are very accurate, but their usage was not possible for this experiment.[17]

3.2.4 Motion capturing

For motion capturing, kinematic data from highly reflective markers is collected at 100 Hz with an eight-camera motion analysis system (*Motion Analysis Corporation*). Hereby, reflections of materials such as glasses and lighting should be avoided, as they can affect the data capturing process.

3.2.5 Spirometry setup

The experimental spirometry setup is essential in order to obtain further data for evaluating and comparing the propulsion techniques. It consists of a heart rate belt, turbine flowmeter assembly, reference gas, and an exercise mask which is connected to a K5 portable unit (PU).

The turbine flowmeter assembly includes a bidirectional turbine, an optoelectronic reader and an optional wind cover. Inside the turbine, infrared light interruptions of the spinning blade are measured by the reader. The wind cover is mounted on the reader and it is used for protecting the turbine from external conditions, like wind, in order to avoid errors. Different sizes of the exercise mask are available. It is made of silicone and has to be disinfected after usage. The K5 PU consists of O₂ and CO₂ analysers, a sampling pump, a transmitter, barometric sensors, electronics and a battery and is connected to the software via Bluetooth. Batteries may be charged for about four hours. [18]

3.3 Software for data capturing, post-processing and analysis

The following software was used for data capturing, post-processing and analysis. The performed steps are described in more detail in chapter 3.5. Table 4 provides an overview for which purpose the software was used:

Software for		
data recording	data post-processing	data analysis
<ul style="list-style-type: none">• <i>Cortex</i>• <i>Cosmed Omnia K5</i>• <i>Desys Trigno Lab</i>• <i>LabView®</i>	<ul style="list-style-type: none">• <i>Cortex</i>• <i>Matlab</i>• <i>Microsoft Excel</i>	<ul style="list-style-type: none">• <i>Matlab</i>• <i>Microsoft Excel</i>• <i>Mokka</i>• <i>OpenSim</i>

Table 4: Overview of software used for data recording, post-processing and analysis.

Cortex

Cortex is a motion analysis software for handling all phases of motion capture. The initial setup, calibration of the cameras, tracking and post processing of the kinematic data is done within this program. [19]

Capture (*.cap) files, which show motion trails for markers and forces, can be shown live and in post processing mode [19]. Each data set has its own marker names, linkages, templates and optional skeleton definitions that can be post processed separately. After post processing, the motion files are saved in *.c3d format. With tab selection, the particular marker set is selected and highlighted, as shown in 10 [19].

Figure 10 shows a screenshot of the software in post-processing mode.

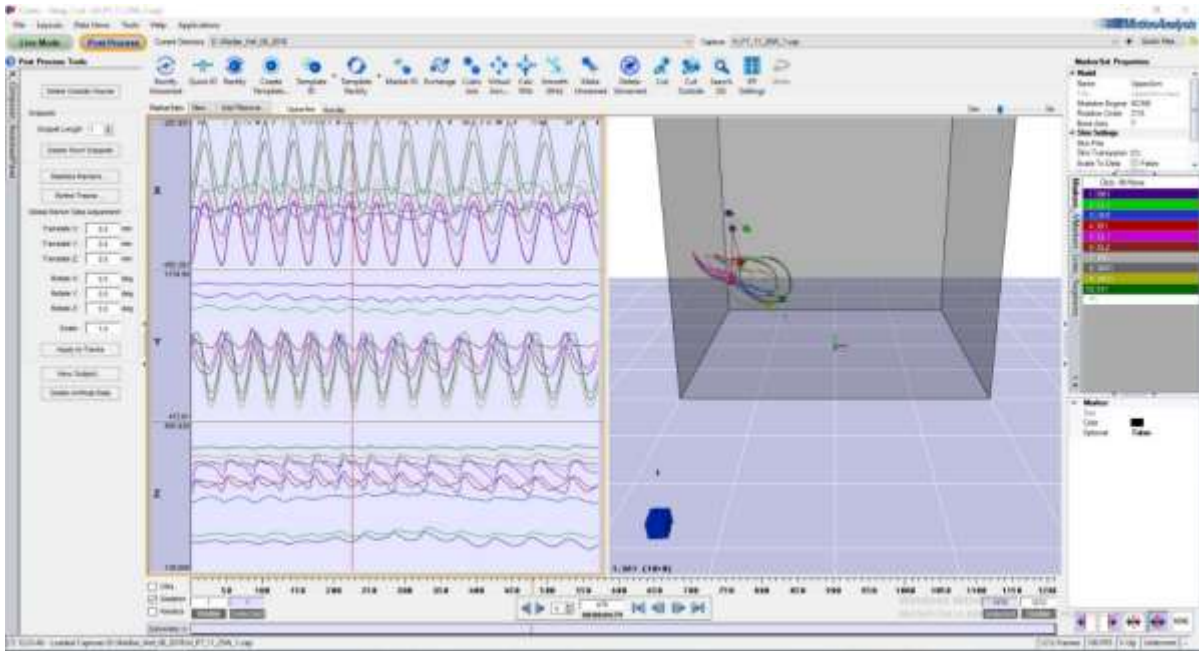


Figure 10: Screenshot of Cortex software showing marker selection in post-processing mode.

Cosmed Omnia K5

OMNIA by *COSMED* is used for data management and interpretation of the spirometry test [22]. Parameters, such as the oxygen uptake VO_2 , the respiratory exchange ratio RER, the heart frequency HF, the bike velocity v_{bike} and the bike wheel cadence n , are captured and analysed in *Excel*.

Delsys Trigno Lab and Trigno Control Utility

Electromyographic (EMG) signals are recorded with a wireless EMG signal detection system (*Delsys Trigno Lab*). Connection and numbering are controlled with *Trigno Control Utility*.

LabView®

The test rig is controlled with a *LabView®* program via USB connection. It enables control of the motor via parameters such as velocity, inclination and floor condition profiles. For this test series, the software was used to set the wanted power out value and for collecting the test rig data. The graphical user interface (GUI) also shows the actual speed and the target speed in a line graph during measurement. The test rig can be used in constant power or incremental power mode. The constant power mode was used for motion capture measurements, whereas the incremental power mode was used for a progressive aerobic test. [15]

Matlab

Matlab is used for synchronisation of kinematic (videometry) and kinetic (instrumented handle and *SmartWheel*) data, as well as for post processing and analysing the EMG data.

Microsoft Excel

Excel was used for the analysis of the spirometric data. Calculations of the mechanical efficiency were performed and figures were created easily within the program.

Mokka

Mokka stands for *Motion Kinematic and Kinetic Analyzer* and is an open-source and cross-platform software which can easily analyse biomechanical data [20]. It is used for visualization of *.c3d files which were post-processed in *Cortex*. All markers that are visible during the whole test duration are checked at this point.

OpenSim

OpenSim is a software for biomechanical modelling, simulation and analysis [21]. Movements, such the propulsion of a wheelchair, may be simulated and results, such as joint angles, may be obtained. Filtered kinematic and kinetic data for each subject is provided as an input into the *OpenSim* software. The tool of *Inverse Kinematics* is used to determine joint kinematics of the shoulder, elbow and wrist.

3.4 Measurement Series at *Rehabilitation Centre Weißer Hof*

In the following subchapters, the performed measurement series at the *Rehabilitation Centre Weißer Hof* is further described with respect to the measurement setup, the participating subjects and the testing protocol. The study was approved by the local ethical committee.

3.4.1 Measurement Setup

A draft and an image of the experimental setup at the *Rehabilitation Centre Weißer Hof* are shown in Figure 11. Eight cameras are placed around the test rig wheelchair (black), five (green) are positioned high up (H2, H3, H5, H6, H8) and three are positioned upside-down (D1, D4, D7). All cables are taped on the floor and lead to a desk where two laptops and all measuring devices (blue) are placed. Black tapes are used in order to minimize reflections that interfere with motion capturing. Calibration of the cameras with the software *Cortex* has to be done before starting the test series, as cameras must not be moved afterwards. Signals are captured at a frequency of 100 Hz.



Figure 11: Draft and image of experimental setup at the Rehabilitation Centre Weißer Hof.

3.4.2 Subjects

Each of the eight subjects who took part in the measurements taken at *Rehabilitation Centre Weißer Hof* in Klosterneuburg (Austria) gave informed consent to participate in this study. Subjects were eligible for the measurements if they were long-term wheelchair users and had no upper limb injuries. Both male (five) and female (three) subjects participated, five suffering from incomplete paraplegia, two from complete paraplegia and one from single leg amputation. A profile of each subject (sex, age, height, weight, nature of disability) is given in Table 5. Subjects were asymptomatic for upper extremity pain or injury as determined by questions addressing their medical history, and no subject reported upper extremity involvement related to their disability.

Subject	Sex	Age	Height	Weight	Nature of disability
	male/female	years	m	kg	
1	female	53	1.79	62	incomplete
2	male	27	1.88	80	complete
3	female	43	1.60	75	incomplete
4	female	56	1.60	63	complete
5	male	52	1.75	65	single leg amputation
6	male	45	1.92	85	incomplete
7	male	21	1.85	62	incomplete
8	male	51	1.73	93	incomplete

Table 5: Profile of the wheelchair dependent subjects who participated in the study.

Thirteen able-bodied volunteers with no wheelchair experience (non-wheelchair users) participated in spirometry measurements at *TU Wien* for comparison reasons. Table 6 shows the profile of all male (six) and female (seven) subjects.

Subject	Sex	Age	Height	Weight
	male/female	years	m	kg
1	female	23	1.65	70
2	male	57	1.86	68
3	female	21	1.70	63
4	male	21	1.85	92
5	female	21	1.62	54
6	female	19	1.65	65
7	male	23	1.81	58
8	male	33	1.73	74
9	male	24	1.69	70
10	female	26	1.75	58
11	male	19	1.87	61
12	female	43	1.68	60
13	female	35	1.76	64

Table 6: Profile of the non-wheelchair dependent subjects who participated in the study.

3.4.3 Testing protocol

For the performed measurements at the *Rehabilitation Centre Weißer Hof* in August 2018, the following general test protocol is used:

Whilst each subject familiarises themselves with the experimental setup by sitting in the wheelchair, personal data is recorded. The crank centre's position is adjusted horizontally to the middle of the thigh and vertically to approximately elbow height. Subjects were able to practise pushing on the test rig until they were comfortable with the set-up.

Then, ten highly reflective markers and seven EMG sensors are placed on the subject's torso and right upper limb. Tapes are used for better fixation. Distances between the markers (EL1-PoA (point of application of isometric measurements), SH1-PoA, CL1-SH1, SH1-EL1, EL1-WR1, WR1-FI1) are measured for later analysis (Annex 2). Abbreviations are described in 3.4.3.1.

Lastly, subjects are asked to perform tests as further described in the following sections. During the tests at the *Rehabilitation Centre Weißer Hof*, isometric measurements, motion capture measurements and spirometry measurements were taken, whereas only spirometry tests were performed at *TU Wien*.

A rough overview of the testing protocol is illustrated in Figure 12. The sequences of the attachment of the EMG sensors and markers as well as the measurements itself are discussed in more detail in the following paragraphs.

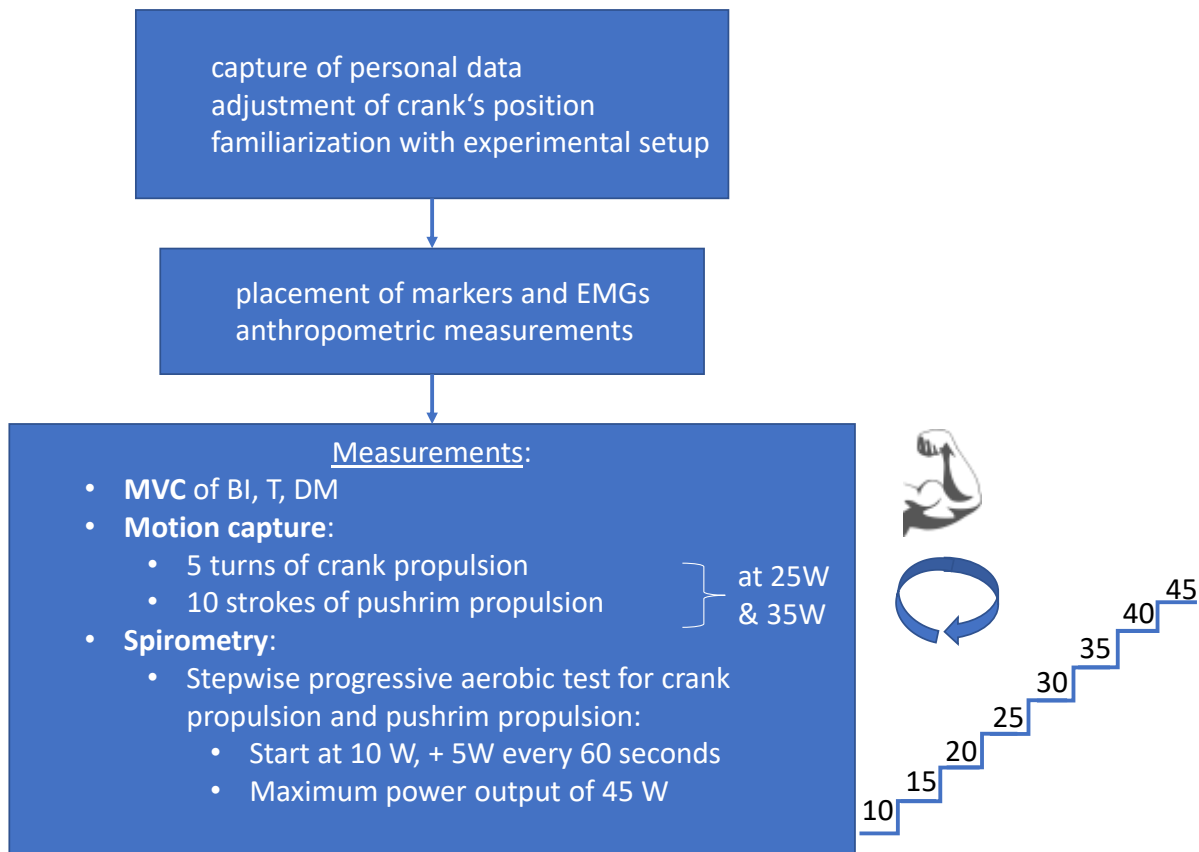


Figure 12: Overview of testing protocol at the Rehabilitation Centre Weißer Hof.

3.4.3.1 Attachment of EMG sensors and markers

The EMG sensors, which are placed on the skin with *Sensor Adhesive Interface* from *Delsys*, record the electrical activity of *musculus deltoideus anterior* (DA, 1), *musculus deltoideus medialis* (DM, 2), *musculus deltoideus posterior* (DP, 3), *musculus pectoralis* (P, 4), *musculus biceps brachii* (BI, 5), *musculus triceps brachii caput laterale* (T La, 6) and *musculus triceps brachii caput longum* (T Lo, 7).

Highly reflective markers are placed on the back bone BB1, shoulder SH1 (acromion process), articulo sternoclavicularis CL1, *musculus biceps brachii* BI1, elbow EL1 (lateral epicondyle) and EL2 (medial epicondyle), forearm FA1 (1/3 way between processus styloid radialis and lateral epicondyle), processus styloid radialis WR1, processus styloid ulnaris WU1 (dorsal surface half way between styloid processes) and index finger FI1 (2nd metacarpophalangeal joint).

Four markers are placed on the crank propulsion device (at the crank center HC, on the sliding guide HH, and on the top HT and bottom HB of the handle) and three markers are put on the

pushrim propulsion device (one in the centre SC, one close to the edge of the wheel SR, and one in-between). For attachment adhesive rings (*Medi Ware*) are used.

The precise positions of the EMG sensors and markers are illustrated in Figure 13.

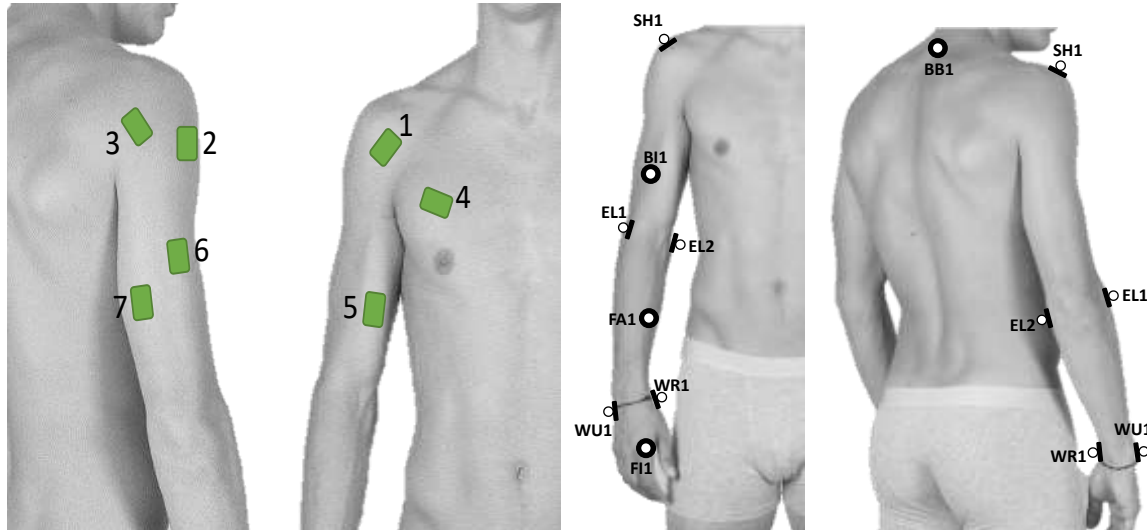


Figure 13: Positions of EMGs (left) and markers (right) attached to subject.

3.4.3.2 Isometric measurements

For measuring maximum isometric joint torques and to determine the maximum voluntary contraction (MVC) of *musculus biceps brachii*, *musculus triceps brachii* and *musculus deltoideus medialis*, subjects are asked to push against the force measuring handle.

MVC tests of *musculus biceps brachii* (starting position: shoulder in neutral position, 90° elbow flexion, forearm in supination, resistance in increased elbow flexion), *musculus triceps brachii* (starting position: shoulder in neutral position, 90° elbow flexion, forearm in pronation, resistance in elbow extension) and *musculus deltoideus medialis* (starting position: 90° shoulder joint in abduction, 90° elbow flexion, resistance in increased abduction) are performed on the static force-measuring handle whilst EMG data is recorded. These measurements are each repeated three times.

The left image in Figure 14 shows a subject during the MVC test of *musculus deltoideus medialis*.

3.4.3.3 Motion capture measurements

Motion and EMG data is collected by recording at least five turns of crank propulsion and ten strokes of pushrim propulsion.

Subjects are asked to propel at a velocity of 1.1 m/s (± 0.1 m/s). The required and actual wheel velocities were continuously presented on a computer screen. The timespan of data collection of approximately five turns of crank propulsion and ten strokes of pushrim propulsion is orally announced. The resistance setting of the test rig is set to 25 W and 35 W respectively.

This test is performed with crank propulsion first. After replacing the crank propulsion device with the pushrim propulsion device, the test is repeated. Overall, four data sets are collected (crank propulsion at 25 W, crank propulsion at 35 W, pushrim propulsion at 25 W, pushrim propulsion at 35 W).

3.4.3.4 Spirometry measurements

Aerobic capacity was determined during a standardized maximum wheelchair exercise test on the test rig wheelchair. Respiratory parameters (V_{O_2} , HF, RER, Eeh etc.) were continuously measured every ten seconds with *Cosmed Omnia K5*. After creating a patient chart and selecting a spirometry testing method, K5 is calibrated for about one minute. The measurement starts automatically after another two minutes. The measurement must not be started manually as errors may occur. [18]

The turbine of the spirometry setup, which is very shock sensitive, is disinfected and calibrated with reference gas before each test and new subject. For flowmeter calibration, the turbine is attached to the optoelectronic reader and is connected to a 3-liter calibration syringe. The piston is moved in and out to simulate six breath cycles. For assembling the whole setup, the turbine is plugged into the mask adapter by pushing and rotating it clockwise. The optoelectronic reader is inserted over the turbine and plugged into the mask. The sampling tube is plugged into the wind cover and the turbine cable is connected to the K5 PU.[18]

After that, the mask and heart rate belt are firmly attached without hurting the subject and should remain in position during the measurements. The heart rate belt is worn on bare skin and the electrodes are moistened with an ultrasound gel to enable optimal measurement. The portable unit is placed between the upper legs of the subject, so that the distance to the antenna is more than 25 mm [18].

Subjects performed a continuous stepwise (60 seconds) progressive aerobic test. The resistance setting of the test rig increased by 5 W each minute, after initially starting at 10 W. Subjects propelled at a fixed targeted velocity of 1.1 m/s. The required and actual wheel velocities were continuously presented on a computer screen. A fluctuating horizontal bar represented the

wheel velocity. The full duration of the maximum test was limited to eight minutes and a power output of 45 W (in order to maintain the mechanical stability of the setup). The test was terminated when the subject reached physical exhaustion. Maximum bike wheel cadence and maximum PO are also noted.

This test is performed with crank propulsion first. After some rest (until the subject feels recovered, or a maximum of five minutes) the same test is performed with pushrim propulsion.

After every subject, the mask and heart rate belt are disinfected with a disinfectant spray and air-dried. Steam sterilization can optionally be performed on the silicone parts of the mask. The remaining parts of the mask, including the turbine, are sterilized via dunking in a *CIDEX OPA ortho-Phthalaldehyde* high-level disinfectant solution for 30 seconds.[18]

Figure 14 (right image) shows a subject wearing the spirometry setup during the continuous stepwise progressive aerobic test at the *Rehabilitation Centre Weißer Hof*.



Figure 14: Subject during MVC test of *musculus deltoideus medialis* (left) and continuous stepwise progressive aerobic test at the *Rehabilitation Centre Weißer Hof*, wearing spirometry setup (right).

3.5 Data processing and analysis

Data processing is a main area of this thesis. The goal is to provide guidance about the post-processing process and to create a universal *Matlab* code that is usable for further measurements. The whole *Matlab* code which has been developed in this thesis can be found in Annex 1. For the measurement data obtained by the measurement series at the *Rehabilitation Centre Weißer Hof*, the following steps have been covered:

Before analysis is performed, the captured data must be processed with the *post-processing* tools of *Cortex*, as well as in *Matlab*. *Cortex* is used for post-processing of motion data whereas *Matlab* is used for aligning all data (data from the test rig, handle or *SmartWheel*, and motion capture) to the correct time and capture frame. It is also used for creating analysis files and input files for *OpenSim*, as well as post processing and analysing EMG data. *Excel* is applied in order to create figures. The whole process is sketched in Figure 15.

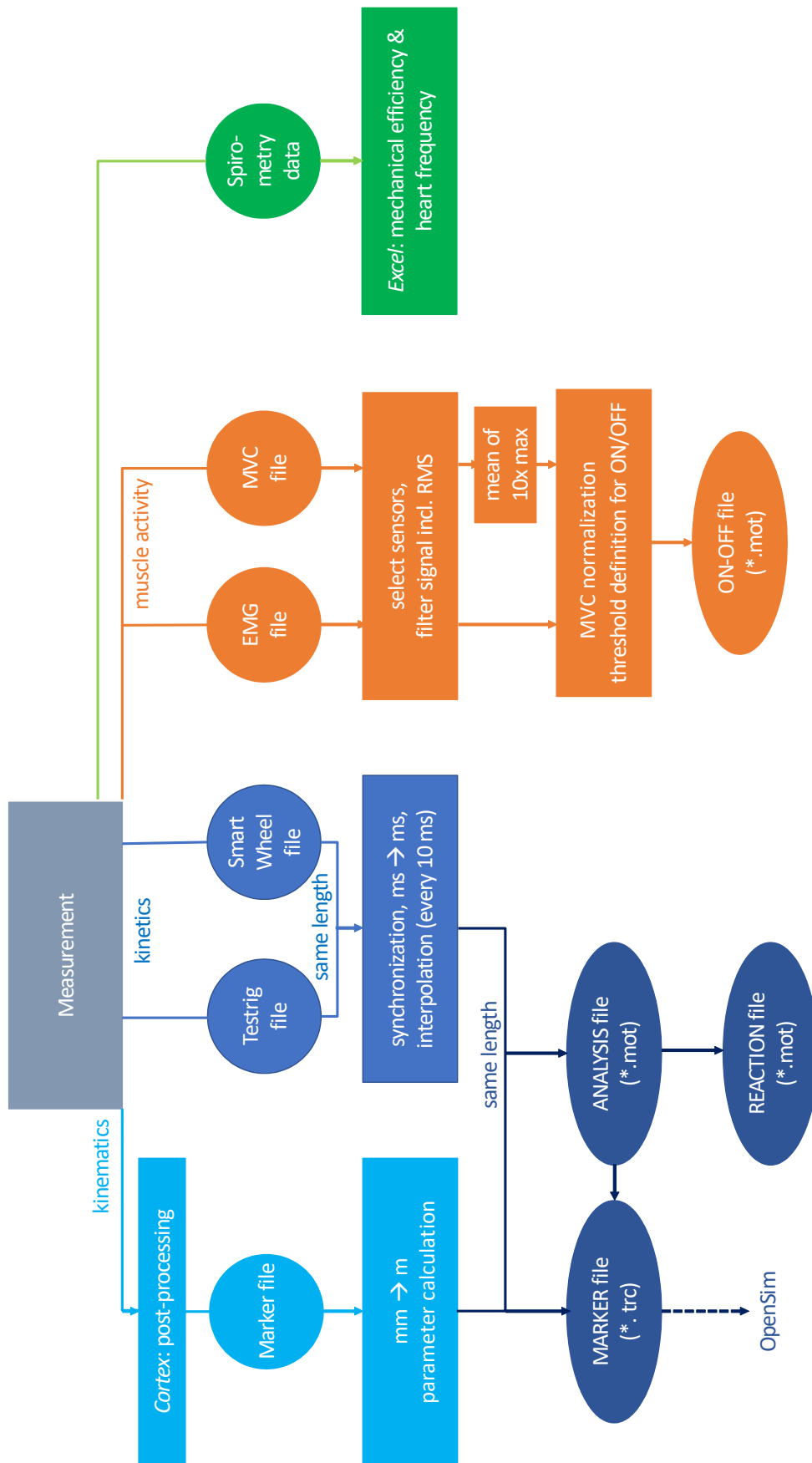


Figure 15: Overview of post-processing.

In the following subchapters, the methodology is described in detail.

3.5.1 Spirometry analysis

For analysis of spirometry measurements, the mechanical efficiency (ME) is calculated. Mechanical efficiency is defined as the percentage of total chemical energy expended that contributes to external work, with the remainder being lost as heat [27]. It is calculated as the external work accomplished (energy output, here: power output PO) divided by the energy expenditure (En) multiplied by 100 (Formula 1):

$$ME = \frac{PO}{En} \cdot 100 [\%] \quad (1)$$

The power output (PO), which varied depending on the test performed, was set in the test rig software. To calculate the mechanical efficiency of wheelchair propulsion, three different formulas were used in order to compare their outcome:

In Formula 2, the energy expenditure (En) was calculated from the oxygen uptake (V_{O_2}) and the respiratory exchange ratio (RER) referring to [28]. The RER is defined as the ratio of oxygen uptake and carbon dioxide production (V_{CO_2}). V_{O_2} and the RER were obtained through the *Cosmed Omnia* software.

$$En = \frac{V_{O_2}}{1000} \cdot \frac{(4940 \cdot RER + 16040)}{60} [W] \quad (2)$$

In addition, an alternative calculation formula was used (Formula 3). The indicator Eeh represents the energy expenditure obtained by *Cosmed Omnia K5*. EEh is multiplied by a factor of 1.163 (because unit conversion must be done), noting 1 kcal/h is equal to 1.163 W.

$$ME = \frac{PO}{EEh \cdot 1.163} \cdot 100 [\%] \quad (3)$$

The third formula for calculating the mechanical efficiency is referring to [29]. As working for one minute at 1 W is equivalent to 60 J, and as each litre of oxygen is equivalent to about 20934 J, mechanical efficiency can be calculated as follows (Formula 4):

$$ME = \frac{60 \cdot PO}{20936 \cdot \frac{V_{O_2}}{1000}} \cdot 100 [\%] \quad (4)$$

Calculations are completed and the resulting graphs are created in *Excel*. Only fully completed wattage (60 seconds) are included in further analysis. This means that the power output levels of patients who did not complete the current wattage level and quit the spirometry exhaustion test prior to maximum power output of 45W are not included in further analysis.

For the analysis of the heart frequency, figures of the captured values are generated in *Excel*.

In the course of the spirometric analysis, additional parameters (velocity and radius) are calculated. For calculation of velocity v during the propelling of the wheelchair, the following formulas 5 and 6 are used for pushrim propulsion and crank propulsion:

$$v_{pushrim} = \frac{v_{bike} \left[\frac{km}{h} \right]}{3.6} \dots \left[\frac{m}{s} \right] \quad (5)$$

$$v_{crank} = \frac{n [rpm] \cdot \pi \cdot r [mm]}{30} \div 1.5 \dots \left[\frac{m}{s} \right] \quad (6)$$

Bike velocity and bike wheel cadence n are captured by *Omnia Cosmed K5* software and radius r is calculated as follows (Formula 7):

$$r = \frac{24'' \cdot 25.4 \frac{mm}{''}}{2} \quad (7)$$

3.5.1 Post-processing of motion data in *Cortex*

In *Cortex*, all motion data are checked and adapted for further use. The goal is to have signals from all markers for the whole measurement duration. A capture file is chosen (for example subject 1, crank propulsion at 25 W) and markersets of the handle or *SmartWheel* and the upper arm are assigned to the correct positions. If this is not done automatically, tools of ‘*exchange*’ (to change the position of two markers) or ‘*quick ID*’ (new assignment of markers) are used. In the event that a marker is covered by the moving upper arm, ‘*rectify*’ can restore a signal. Peaks due to interference may be deleted with ‘*Make unnamed*’.

For better adaption of measurement, files of other subjects’ templates of markersets (which do not change from subject to subject) from the handle and *SmartWheel* are saved. The final capture file is saved as a *.c3d file. For a final check and to ensure all markers are visible during the process, the *.c3d file is opened in *Mokka*.

3.5.2 Post-processing of kinetic and kinematic data in *Matlab*

Matlab is firstly used for synchronising the test rig and *SmartWheel* or *Handle* measurement files. Secondly, this data is further merged with the motion and EMG data. Lastly, three output files are created for final analysis. In the following paragraphs, the performed steps for post processing measurement data during crank propulsion are explained in more detail. The

differences between the post processing process of *Handle* data and the approach of *SmartWheel* data are highlighted at the end of this subchapter.

Firstly, the test rig file (*.xlsx) and *Handle* file (*.xlsx) are read in. The latter starts shortly before the test rig file, and therefore the starting time of the test rig file is set at 0 ms. After confirming that the files are the same length, spline interpolation of the data saved from the test rig and handle is performed in a certain timestep. The timestep chosen for this measurement series of 10 ms is based on the fact that the data is collected at a frequency of 100 Hz as explained in 3.4. In reality, the timesteps may also be slightly longer than 10 ms due to digital delay. By interpolating all of the data for a time gap of exactly 10 ms, a consistent timing is ensured. The table *interpolated_TR_H* shows the aligned data from the test rig and *Handle* at the same time (Table 8), and lists values of frame number, time, angular velocity w , velocity v , their deviations dw and dv , angular acceleration α , acceleration a , their deviations $d\alpha$ and da , Position [deg], revolutions per minute RPM, Handle RPM, Handle v , Actual power, forces and torques in x-, y- and z-direction F_x , F_y , F_z , M_x , M_y and M_z . Forces and torques are still in handle coordinates which need to be traversed into global coordinates as explained later.

Before the motion data is loaded and added to the processed test rig and *Handle* data, the *.c3d is converted into a *.trc file using the converting tool of *Mokka* and is saved as an *.xlsx file. After making sure that the file has the same length as the interpolated data from the test rig and *Handle*, the unit of the marker values must be converted from millimetres to meters by dividing by 1000.

After reading in all kinetic and kinematic data as described in the steps above, the calculation of the crank centre, the centre of forces (CoF), the crank radius, the crank angle and beginning of each cycle, the angular velocity and the tilt angle of the handle, as well as the transformation of the forces and torques into the global coordinate system, are performed.

For calculation of the coordinates of the crank centre, which is a fixed point, the first three non-zero values are averaged in x-, y- and z-directions respectively (Formula 8). The CoF, which represents the point from which a force acts, is the centre between the top and bottom marker of the handle (Formula 9). The coordinates are saved with respect to the previously calculated crank centre, which is then seen as the origin with the coordinates (0/0/0). The CoF is also saved in its global coordinates, as they are required for the input file of *OpenSim*. The distance

between the crank centre and CoF is the radius that changes its length according to the crank's position (Formula 10).

$$Centre_{x,y,z} = \frac{HC_{x,y,z1} + HC_{x,y,z2} + HC_{x,y,z3}}{3} [m] \quad (8)$$

$$CoF_{x,y,z} = \frac{HB_{x,y,z} + HT_{x,y,z}}{2} [m] \quad (9)$$

$$radius = \sqrt{CoF_x^2 + CoF_y^2} [m] \quad (10)$$

The crank angle is calculated by using the x- and y-coordinates of the CoF. Looking from the wheelchair user's perspective, 0° marks the x-axis in a positive direction or the horizontal right position of the crank moving clockwise. With the *atan2d* command, the four-quadrant inverse tangent of a point (X/Y) in regards to the positive x-axis of the global coordinate system is calculated in degrees [23]. This results in angle values from 0° to ±180°, with negative values in a clockwise direction as shown in Figure 16. As the aim is to obtain an angle between 0° to 360° for the propelling movement, further adaptations have to be made. The values in quadrant 3 and 4 (which are negative) are positivised and angle values in quadrant 1 and 2 (which range from 0° to +180°) are subtracted from 360° in order to obtain values from 180° to 360°. Also, special cases of 0° and 180° need to be defined. As a result, the starting frames of each cycle for crank propulsion are selected when a crank angle value is less than the previous one.

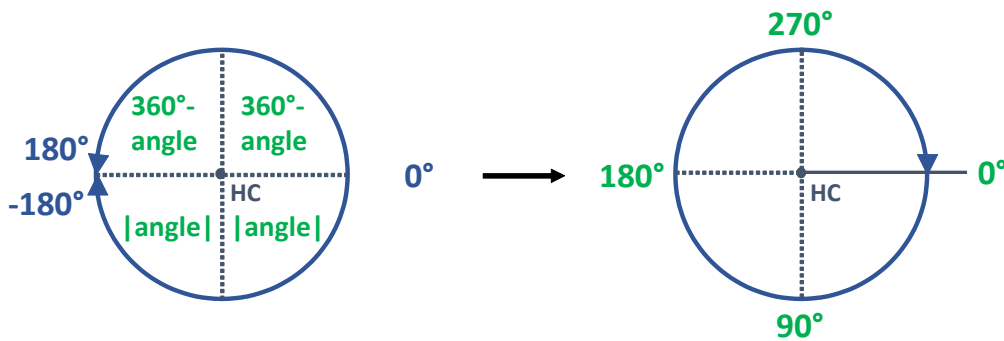


Figure 16: Crank angle calculation.

The angular velocity is calculated as the change in angle (in radians) divided by the time change which is again 10 ms (Formula 11). This means that the current angle is subtracted from the previous angle values and is divided by 0.01 seconds.

$$angular\ velocity = \frac{d(crank\ angle)}{dt} [rad/s] \quad (11)$$

The tilt angle represents the angle in which the handle is tilted during the propelling movement, with 0° pointing in the y-direction of the global coordinate system. It may be calculated similarly to the crank angle with the coordinates of the CoF (which then represents the origin) and the top marker of the handle. As 0° represents the positive y-axis and the tilt angle is positive in an anticlockwise direction and negative in a clockwise direction, the following adaptations need to be made: If the angle is positive (0° - 180°), 90° needs to be subtracted in order to obtain values between $\pm 90^\circ$. The same is done for angles which are between 0° and -90° , resulting in angle values between -90° and -180° . Angles between -90° and -180° need the addition of 270° in order to be traversed into angle values between 90° and 180° . Figure 17 summarizes the previously described process.

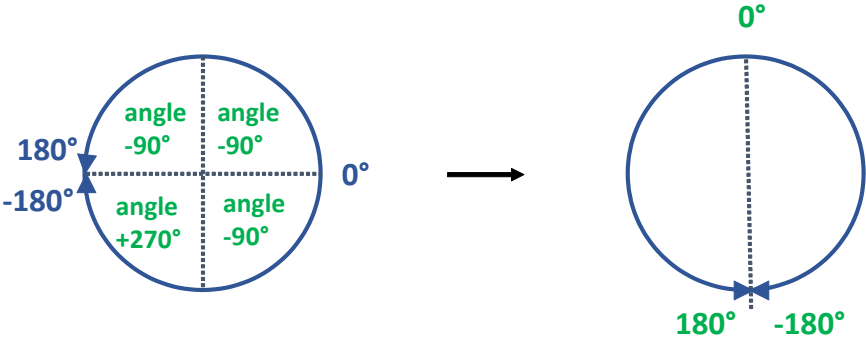


Figure 17: Tilt angle calculation (- clockwise, + anticlockwise).

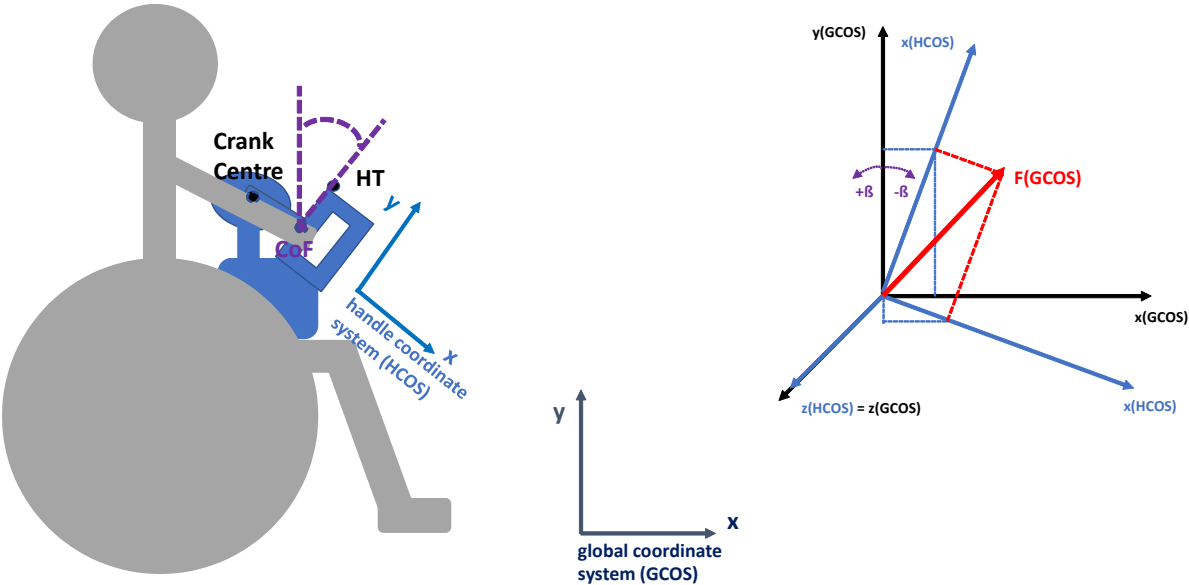


Figure 18: Sketch of calculation of tilt angle (abbreviated as β) as well as conversion of measured forces from handle coordinate system to global coordinate system.

It is important to state that the force measuring handle has its own coordinate system (HCOS) as illustrated in Figure 18. Only the z-direction is equal for both coordinate systems. Therefore, the forces and torques are converted into the global coordinate system (GCOS) as follows. Values in the x-direction in the GCOS are equal to the negated values in the y-direction in the HCOS. Forces and moments in the y-direction in the GCOS are equal to the corresponding values in the x-direction in the HCOS. Lastly, the force vector or torque vector, with its components in the x-, y- and z-direction, is multiplied with a conversion matrix as stated in the following Formulas 12, 13 and 14. Calculations are done as follows, and assisting sketches are shown in Figure 18.

$$F_{GCOS} = F_{HCOS} \cdot A_H \quad (12)$$

with

$$F_{HCOS} = \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} \quad (13) \quad \text{and} \quad A_H = \begin{pmatrix} \cos(\text{tilt angle}) & -\sin(\text{tilt angle}) & 0 \\ \sin(\text{tilt angle}) & \cos(\text{tilt angle}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (14)$$

Based on the previously described steps, three output files are created. The first output, called the ANALYSIS file, contains information on frequency, coordinates of crank centre, frame number (counting from 1 upwards), time, *interpolated_TR_H* with forces and torques in the GCOS, the CoF, the calculated crank angle, the crank radius, and the angular velocity. An example of this file is shown in Table 9.

The MARKER files consist of frequency of capturing, number of frames selected, starting and ending frame, number of markers, units, frame number, time and all marker data in the x-, y-, and z-direction. In addition to the fourteen markers placed on the subject, the test rig and handle, the coordinates of the CoF are listed. An example is listed in Table 10.

The REACTION file must have a special format. It contains the name, the number of columns and rows, and the time range of the file (starting and ending time) - each in a separate row and concluded with the *endheader* line. After the *endheader*, the data is listed as follows: time, ground reaction forces in the x-, y- and z-direction and the point on which the ground reaction force is applied and ground reaction torques in the x-, y- and z-direction of the chosen time frame [21]. It is required to negate the force and torque values in order to obtain reaction values. Table 7 shows an example of this file.

For further analysis in *OpenSim*, an extract of a chosen time frame range, which represents a certain number of cycles, forms the MARKER file as well as the REACTION file. In this thesis, the MARKER file is the input file for *Inverse Kinematics* analysis in *OpenSim*. The REACTION file could be used for *Inverse Dynamics* analysis in *OpenSim*.

name H_P7_11_25W_1.txt_REACTION.mot									
datacolumns 10									
datarows 269									
range 4.7 7.38									
endheader									
time	r_ground force_vx	r_ground force_vy	r_ground force_vz	r_ground force_px	r_ground force_py	r_ground force_pz	r_ground_ torque_x	r_ground_ torque_y	r_ground_ torque_z
4.7	17.306	24.732	1.3916	-0.29255	0.67774	0.69527	-0.070235	-0.14754	-1.7639
4.71	17.93	25.275	1.541	-0.29427	0.66332	0.69503	-0.071789	-0.1548	-1.8176
...

Table 7: Example of a REACTION file for two frames.

frame	time	w [rad/s]	v [m/s]	dw [rad/s]	dv [m/s]	alpha [rad/s ²]	a [m/s ²]	dalpha [rad/s ²]	da [m/s ²]	Position [deg]	RPM [1/min]	Handle v [m/s]	Actual power [W]	Fx [N]	Fy [N]	Fz [N]	Mx [Nm]	My [Nm]	Mz [Nm]	RPM [1/min]	Handle RPM [1/min]	Handle RPM [1/min]	
1	0	4.45382181	1.35752437	-0.01422119	-0.00433463	445.382169	135.752485	-1.42212015	-0.43346222	1039.51852	42.5308646	2.03628707	25	10.690445	6.191552	-4.055782	0.119685	-0.779206	-0.548414	42.5308646	63.7962963	63.7962963	
2	0.01	4.52492774	1.37919752	0.05688474	0.01733852	452.492769	137.919796	5.68848059	1.73384889	1041.92593	43.2098768	2.0687967	25	10.669295	4.95151623	-4.25627585	0.14005692	-0.76858985	-0.532238362	43.2098768	64.8148148	64.8148148	
3

Table 8: Example of interpolated table of test rig and Handle data for time frame 1 and 2.

H_P5_11_1_25W__ANALYSIS.txt																							
Marker freq : 100Hz																							
Crank_Center_XYZ (m) : -0.49587 0.68529 0.57902																							
Marker values in meters & Force in Newtons																							
Frame#	Time	w[rad/s]	v[m/s]	dw[rad/s]	dv[m/s]	alpha[rad/s ²]	a[m/s ²]	dalpha[rad/s ²]	da[m/s ²]	Position[deg]	RPM[1/min]	Handle RPM [1/min]	alpha[rad/s ²]	a[m/s ²]	CoFx [m]	CoFy [m]	CoFz [m]	Crank Angle [deg]	Crank Radius [m]	Handle RPM[1/min]	Handle v[m/s]	Handle v[m/s]	
1	0	3.5605	1.0852	0	0	356.05	108.52	0	0	1165	34	34	0	0	1165	0	0	1165	34	34	51	1.6278	1.6278
2	0.01	3.2773	0.99892	0.035332	0.01077	369.11	112.5	63.564	19.374	1154.4	31.296	31.296	369.11	112.5	0.067727	-0.073127	0.067727	25.144	0.17211	31.296	46.944	1.4984	1.4984
3

Table 9: Example of ANALYSIS file of two frames.

PathFileType	(X/Y/Z)	H_P7_11_25W_1.txt																
4																		
CameraRate	NumFrames	NumMarkers	Units	OrigDataRate	OrigDataStartFrame	OrigNumFrames												
100	269	14	m	100	471	739												
Frame #	BB1						SH1						BL1					
	X1	Y1	Z1	X2	Y2	Z2	X3	Y3	Z3	X4	Y4	Z4	X5	Y5	Z5			
471	-0.73274	1.083	0.26643	-0.61373	0.97703	0.29515	-0.68226	10.123	0.47065	-0.6117	0.80273	0.52023	-0.6503	0.71411	0.62669			
472	-0.73401	10.832	0.26649	-0.61458	0.97741	0.29479	-0.68273	10.117	0.47068	-0.61204	0.8012	0.51852	-0.65043	0.71239	0.62479			
473			
EL2																		
X6	Y6	Z6	X7	Y7	Z7	X8	Y8	Z8	X9	Y9	Z9	X10	Y10	Z10				
-0.66174	0.67447	0.55469	-0.54729	0.72104	0.63218	-0.38491	0.67395	0.62638	-0.37857	0.60537	0.65633	-0.29255	0.67774	0.69527				
-0.66327																		
0.67308	0.55312	-0.54716	0.71591	0.63047	-0.38672	0.66239	0.62586	-0.38267	0.5939	0.6559	-0.29427	0.66332	0.69503					
...		
HT1																		
X11	Y11	Z11	X12	Y12	Z12	X13	Y13	Z13	X14	Y14	Z14							
-0.25567	0.74056	0.65054	-0.25581	0.55296	0.66866	-0.35296	0.54034	0.63805	-0.47754	0.68098	0.58278							
-0.25557	0.72499	0.65053	-0.26152	0.53759	0.66793	-0.359	0.52799	0.63748	-0.47741	0.68114	0.58287							
...		

Table 10: Example for MARKER file for two frames.

The same procedure is done for pushrim propulsion measurements with the following adaptations. The test rig file (*.xlsx) and *SmartWheel* file (*.csv) have a different starting point and timestamp. The timestamp counting (in ms) of the test rig file starts on the 1.1.1904 (due to the Labview script [24]) whereas the *SmartWheel* file starts on the 1.1.1970 (due to the Python script which uses UNIX time epoch [25]). As a first step the timestamp of the test rig file is adapted by subtracting 2 082 844 800 000 ms (66 years including lap years), which is equal to the difference of the two mentioned dates. As a result, the counting starts at 1.1.1970 for both files.

Capturing of *SmartWheel* data starts before and ends after capture of test rig data, as this correlates with the recording order. Therefore, the first and last value of the test rig timestamp is taken as the starting and ending point of the measurement, and is synchronised with the *SmartWheel* timestamp. The data outside of this time span is deleted. Furthermore, the starting time of the test rig file is set at 0 ms, and the following values collected, including SmartWheel values, are adapted and a unit conversion into seconds is performed.

Furthermore, the selection of the frames of a stroke cycle for pushrim propulsion is based on the torque in the z-direction M_z . The moment when M_z exceeds 2 Nm is fixed as the starting point of a new stroke. Figure 19 shows an example of a M_z curve of raw data, where the marked areas indicate the excitation above 2 Nm and the start of a stroke cycle.

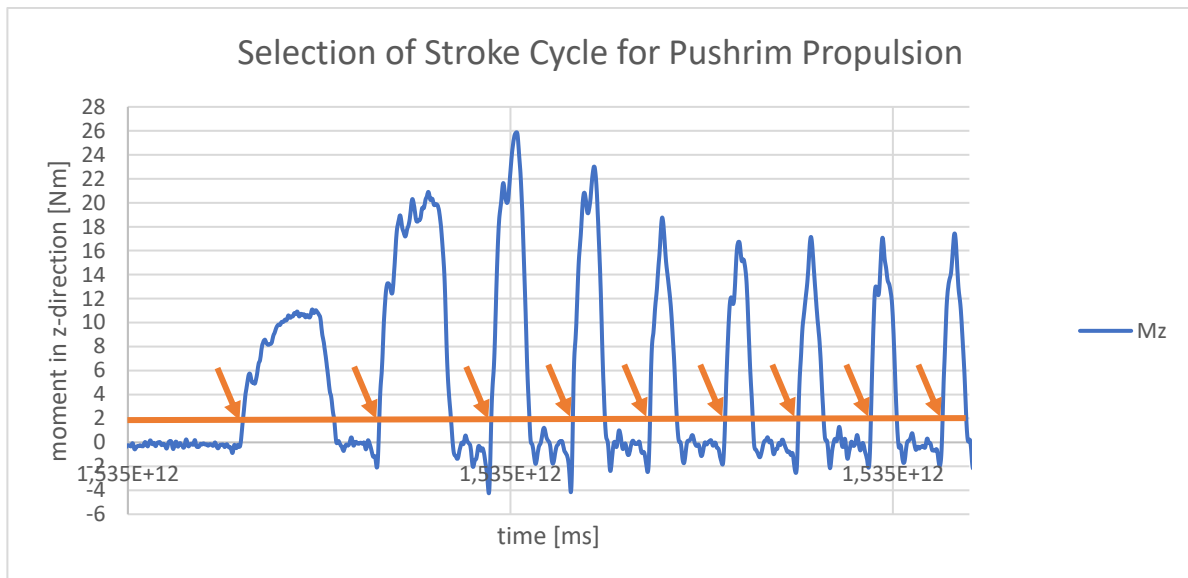


Figure 19: Selection of stroke cycle for pushrim propulsion.

3.5.3 EMG post-processing and analysis

EMG data obtained by this measurement series (crank propulsion at 25 W, crank propulsion at 35 W, pushrim propulsion at 25 W, pushrim propulsion at 35 W), as well as MVC tests (*musculus biceps brachii*, *musculus triceps brachii* and *musculus deltoideus medialis*) are also processed in *Matlab*. An overview of the procedure is sketched in Figure 20.

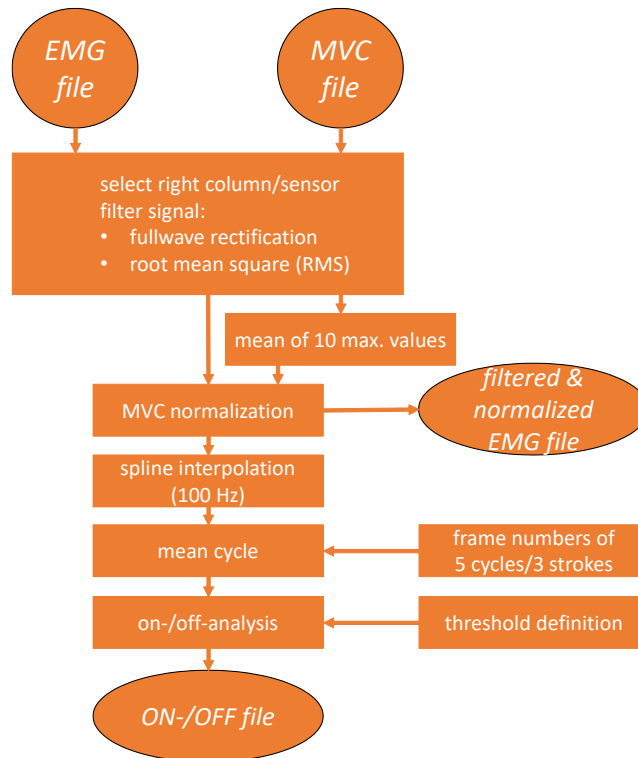


Figure 20: Overview of EMG post-processing and analysis.

The EMG data, recorded at a frequency of 2000 Hz, is filtered in a two-step procedure. Besides the muscular activity, the raw recording file contains additional columns of acceleration signals. Therefore, the proper columns that include the seven sensors are selected. Firstly, full wave rectification of raw EMG data is performed to convert all negative spikes into positive amplitudes. The resultant effect is that standard amplitude parameters, such as mean and maximum value, can be applied to the curve. The mean of raw EMG data would be zero. Secondly, the recommended digital smoothing algorithm is applied with the root mean square (RMS) calculation in order to minimize the influence of interference. The root mean square, which is the square root of the mean square, is calculated as follows (Formula 15):

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \quad \text{with } n = 50 \quad (15)$$

The RMS EMG “*reflects the mean power of the signal*” [17]. A time window of 50 ms where RMS is applied ($n = 50$) is chosen, and it falls within the range of the recommended interval of 50-100 ms by Konrad et al. in [17]. As a result, there is one value every 50 ms and thus the signal appears smoother. A larger time window would increase the risk of a phase shift during the contraction phases with high signal increase [17].

Next, amplitude normalization is performed by normalizing the signal to the corresponding averaged ten maximum values of the MVC measurement. Therefore, the MVC measurement data is read in automatically and is also processed with full wave rectification and RMS calculation. Before that, only the targeted column of the respective EMG sensor is extracted from the raw EMG file. The target reference value (the mean of the ten highest values of the MVC) can only be obtained for *musculus deltoideus medialis*, *musculus biceps brachii* and *musculus triceps brachii caput laterale*, as well as *musculus triceps brachii caput longum*. The EMGs for *musculus deltoideus anterior*, *musculus deltoideus posterior* and *musculus pectoralis* are not considered as MVC tests were not carried out for these muscle groups. As a result, the EMG signal is displayed as the percentage of the maximum innervation capacity. In other words, it shows “how much demand ergonomically a work task is asked from a worker” [17]. This MVC normalization, which is calculated as a percentage, has the benefit that it enables direct quantitative comparison of EMG findings between subjects

Finally, the signal is interpolated to 100 Hz (spline interpolation in steps of 10 ms) to enable a choice of propulsion cycles (for crank propulsion) or strokes (for pushrim propulsion) with the same frame numbers.

After preparing and saving the data as described previously, analysis can be performed. Five cycles of crank propulsion or three strokes of pushrim propulsion are chosen visually and a file, which contains the starting and ending frame number of those cycles or strokes, is read in. This file is provided in Annex 3. The cycles are averaged in order to describe typical movement characteristics. Humans are not robots and even repetitive motion varies as it is difficult to consistently reproduce a movement [17]. Therefore, those cycles most likely do not have the same duration and frame number. Firstly, the original frame scale is converted into a percentage scale ranging from 0 % (start of cycle), - in steps of 1 % - to 100 % (end of cycle) [17]. Secondly, the mean value of the corresponding frames of all cycles is calculated. As a result, one mean cycle is generated.

After performing the previously described steps for each subject, all final files containing the mean cycle are averaged and used as input files for the on-/off-analysis described below. Before the averaging process, plots of the mean cycles are checked visually and muscle groups that exceed 100 % of their activity level are excluded so that results are not falsified (as stated within the code). A reason for this is that the subject reached higher values during wheelchair propulsion than during the MVC test. This phenomenon could be mainly observed in subjects 1 and 4 who suffer from paralysis that occurs at a higher level of their spine. Particularly during the MVC tests it was noted that they had difficulties keeping their balance whilst seated.

For evaluating the activity of the muscles during one cycle or stroke of wheelchair propulsion, an on/off analysis is performed. A muscle is considered as being “*off*” when it is not actively contributing to the propulsion movement, whereas a muscle is labelled as “*on*” when it is active [17]. In order to distinguish between these two states, a threshold must be defined. This process is critical. [17] suggests different methods of threshold definition. A threshold definition by EMG-baseline noise is the most popular approach. This approach suggests multiplying the standard deviation of the EMG baseline before the activity by a factor of 3 in order to define the threshold [17]. However, as noise can vary between subjects and measurement series, and as it is very low for modern EMG amplifiers, factors of 8 or higher are required [17]. For this analysis, this approach is not chosen. This is due to the fact that no recordings of complete relaxation of all muscle groups are taken, as the measurement starts only when the wheelchair is propelled already. In addition, during MVC recording, muscles are not completely relaxed as the arm is held in the required position.

An alternative method for threshold definition is used for this analysis. This method is used to define a fixed microvolt value or percentage for MVC normalized recordings. In literature, there is no consensus on the optimal threshold value [26]. After visual inspection, the threshold is defined as 12 % of the MVC normalized EMG data. All values greater than the threshold value of 12 % are replaced by ‘1’, and all values less than 12 % are replaced by ‘0’. An on-/off figure is generated in *Excel*.

Figure 21 shows the mean cycle/turn of crank propulsion at 25 W, with the red line at 12 % of the muscular activity indicating the visually defined threshold. Above this threshold, the

muscles are considered as being “on”, whereas below this threshold, they are considered as being “off”.

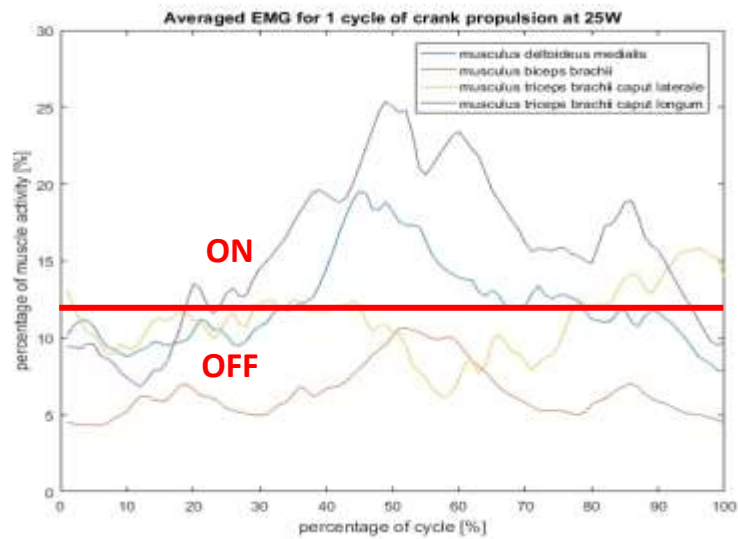


Figure 21: Mean EMG cycle of crank propulsion at 25 W with threshold of 12 % (red line).

3.5.5 Analysis in *OpenSim*

OpenSim is used to analyse joint kinematics of the shoulder joint, elbow joint and wrist joint with *Inverse Kinematics* (IK). IK goes through every time step of the recorded motion, which are the time frames of the capture, and calculates the joint angles using the marker data [30].

A musculoskeletal model which includes the major musculotendon actuators spanning shoulder, elbow and wrist joints is used for the following process. The dynamic musculoskeletal model, used for *OpenSim* simulation, was utilised based on the work by Kurup et al. in [10]. The model consists of a fixed thorax segment, the right upper arm, the right forearm defined by individual components of ulna and radius, and the hand segment. The shoulder joint was defined with three degrees of freedom (DOF) (elevation plane, shoulder elevation angle, shoulder rotation angle), the elbow joint had one DOF (extension/flexion) and the wrist joint had two DOF (flexion/extension, ulnar/radial deviation). 15 musculotendon actuators, which simulate an active contractile element, a passive elastic element and an elastic tendon of Hill’s muscle model, span the shoulder, elbow and wrist joints [26]. Figure 22 shows the musculoskeletal model in *OpenSim*, and markers which are visible from that viewpoint are marked with their abbreviations.

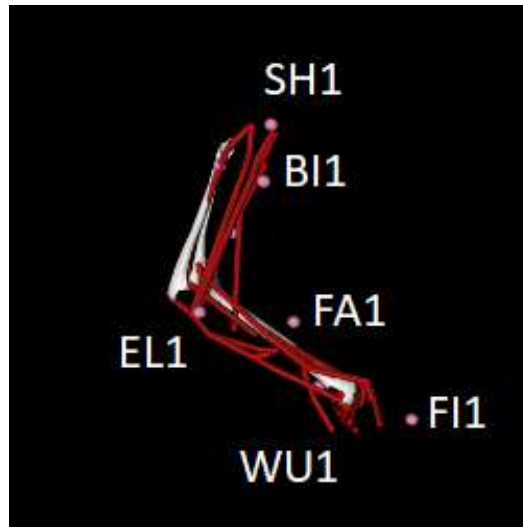


Figure 22: Musculoskeletal model used for OpenSim with marking of markers which are visible from this viewpoint.

This model must be adapted to the anthropometry of each subject individually by applying the *Scaling* tool. As the output of this step, a musculoskeletal model, scaled to the dimensions of the subject, is generated and is used for the following *Inverse Kinematics* analysis. Scaling the model (*.osim) with experimental marker data of the Marker file (*.trc) obtained by *Matlab* (2.7.2), and thereby matching the model to the subject, is crucial for getting good results. Additional to the Marker file, a setup file (*.xml) is used as an input file. The setup file specifies the time range of the marker file, which in turn contains the chosen cycles or strokes of crank or pushrim propulsion, and matches the experimental markers to the virtual markers. It also weights the markers (for all markers a weight of 1.0 is used). Furthermore, a single time frame defines the static posture of the subject, giving the approximate location of the markers. [30]

For IK, another setup file is needed which includes information such as marker weightings (CL1: 10, SH1: 15, BI1: 5, EL1: 20, FA1: 10, WR1: 70, EL2: 1, WU1: 70, FI1: 90). As an output, a motion file containing the joint angles is generated which is shown in Table 11. The header contains information about name, version, number of rows and columns, and information on the unit of the parameters. *Endheader* symbolises the end of the header. The columns are the time starting at 0, the time of the starting moment of the cycle, the elevation plane, shoulder elevation, shoulder rotation, elbow flexion, and deviation and flexion of the wrist joint. These results may also be plotted with the *Plotting* tool.

```
UNKNOWN
version=1
nRows=317
nColumns=8
inDegrees=no
endheader
```

time	time	elv_angle	shoulder_elv	shoulder_rot	elbow_flexion	deviation	flexion
0	4.25	17.4531543	31.9640065	11.9698385	48.1369364	-9.25114333	-0.29087066
1	4.26	17.1167385	31.4049925	12.2190058	47.1920185	-9.19003667	-0.29354775
2

Table 11: Example of output file of Inverse Kinematics in OpenSim, containing joint angle data

The described procedure is applied for all subjects. Afterwards, the original frame scale of each of the five cycles or three strokes is converted into a percentage scale ranging from 0 % (start of cycle) - in steps of 1 % - to 100 % (end of cycle) in *Matlab* [17]. The mean cycles or strokes are averaged over all subjects, and figures are created within *Excel*. The whole code can be found in Annex 1.

4. Results

In this chapter, the results of the measurement series performed at the *Rehabilitation Centre Weißer Hof* (wheelchair users) and at the *TU Wien* (non-wheelchair users) are presented. The first section presents the spirometric data, the second section displays the EMG results and the final section presents the results of the inverse kinematic analysis.

4.1 Mechanical efficiency and heart frequency

The first subchapter compares pushrim propulsion to crank propulsion in relation to mechanical efficiency and heart frequency. A comparison between wheelchair users and non-wheelchair users is done in the second subchapter.

4.1.1 Results of wheelchair users

Table 12 shows an overview of the mean mechanical efficiency values of all wheelchair users calculated with Formula 2, as stated in 3.5.1 (ME1), as well as standard deviation and mean velocity. For all subjects, the mean value of crank propulsion is higher in comparison with pushrim propulsion. The overall mean value of mechanical efficiency of pushrim propulsion is 7.7 ± 1.2 %, while crank propulsion is 10.1 ± 2.1 %. The mean velocity of pushrim propulsion $v_{\text{mean,pushrim}}$ is 1.2 ± 0.3 m/s, while crank propulsion $v_{\text{mean,crank}}$ is 1.3 ± 0.2 m/s.

It can be stated that all subjects managed to complete a higher (six subjects) or equal (two subjects) power output level during crank propulsion in comparison to pushrim propulsion. More subjects (five) were able to reach the maximum power output level of 45 W, whereas only two subjects completed the last power output level during pushrim propulsion. All female wheelchair users (three) quit one power output level earlier during pushrim propulsion.

	Pushrim propulsion			Crank propulsion		
subject	ME1	s(ME1)	$v_{\text{mean,pushrim}}$	ME1	s(ME1)	$v_{\text{mean,crank}}$
	%	%	m/s	%	%	m/s
1	9.800148692	± 1.736414548	0.955128205	11.29750529	± 3.805787331	1.544061373
2	7.752085667	± 0.919663663	1.417270531	10.17460868	± 1.388208436	1.382213349
3	8.784076488	± 0.7429611	0.691666667	11.32214963	± 2.734121918	1.032496719
4	8.837336696	± 2.164981062	0.87254902	9.661224425	± 3.292055657	1.054413826
5	6.597641367	± 1.051395967	1.623809524	10.4018476	± 2.259874457	1.202492937
6	6.76034098	± 0.826915018	1.256349206	10.92752927	± 1.262958131	1.127337129
7	7.153562129	± 1.196314837	1.447990544	9.031920869	± 1.102511449	1.639392668
8	6.271995764	± 0.89807918	1.256535948	7.843787204	± 1.167318546	1.213358838
mean value	7.744648473	± 1.192090672	1.190162456	10.08257162	± 2.12660449	1.274470855
			± 0.29989722			± 0.21065251

Table 12: Mean mechanical efficiency of all subjects during crank and pushrim propulsion, calculated with Formula 2 with the standard deviation and mean velocity respectively.

As explained in the methods section, three different formulas for the mechanical efficiency were used. The different outcomes are displayed in Figure 23. Formula 2 and Formula 4 show similar values, whereas Formula 3 is lower in all power output levels.

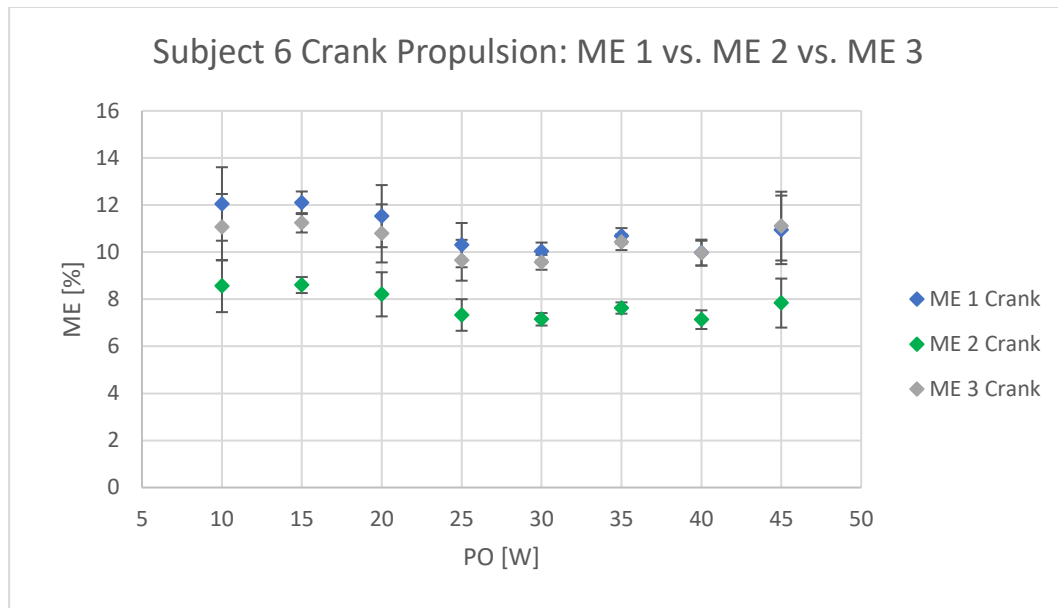


Figure 23: Comparison of different formulas for mechanical efficiency (ME1, ME2, ME3) of subject 6.

In the following Figures and analysis, Formula 2 is used and abbreviated as ME without the index 1 (as in ME1).

Figure 24 shows the mechanical efficiency of pushrim propulsion compared with crank propulsion. The average mechanical efficiency is calculated based on all subjects and is plotted over the power output. On the basis of this measurement series, mechanical efficiency of crank propulsion is always higher than pushrim propulsion. The difference at lower power output values (10 W, 15 W and 20 W) is even more distinct, as seen in Figure 25 (where subject 1 and subject 4 are excluded because of paralysis that occurs at a higher level of their spine). The trend of the pushrim curve shows a decline from the beginning until a power output of 25 W is reached. This is then followed by an incline. The crank curve has a similar course except for the mechanical efficiency, which increases from 30 W. The standard deviation is much higher at low power output values.

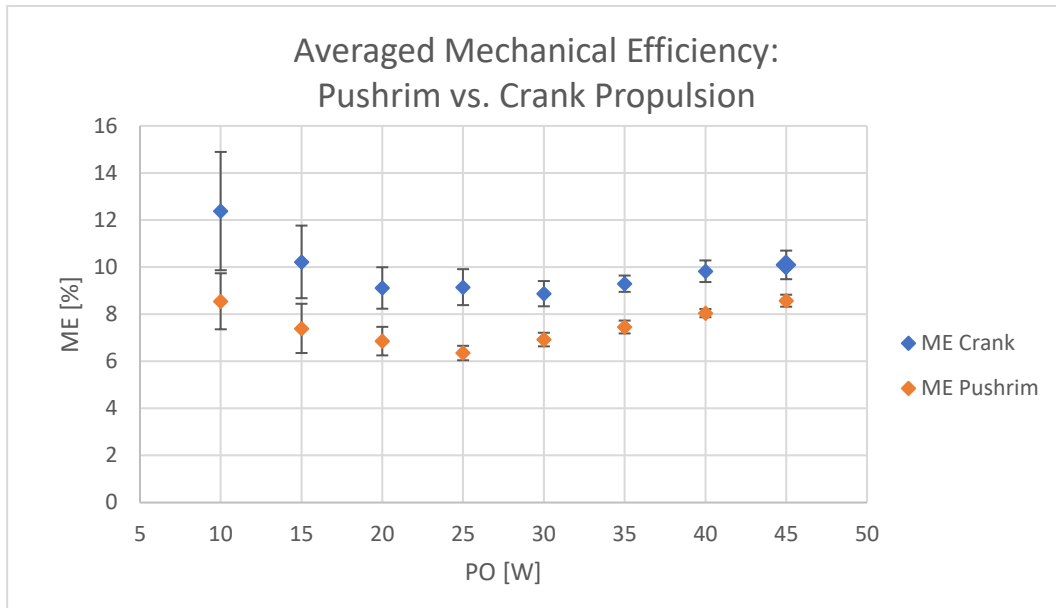


Figure 24: Mean mechanical efficiency of all wheelchair users of pushrim propulsion versus crank propulsion plotted over the power output.

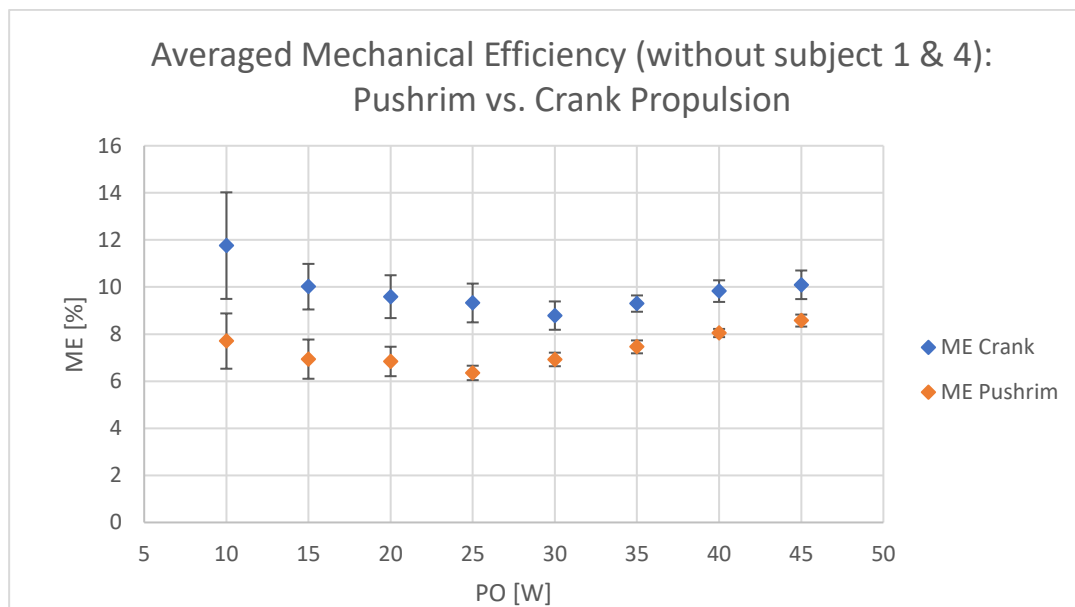


Figure 25: Mean mechanical efficiency of all wheelchair users (except subject 1 and subject 4) of pushrim propulsion versus crank propulsion plotted over decimal steps of power output.

Mean heart frequency (HF) of all subjects plotted over power output intervals of ten (1: 10W + 15W, 2: 20W + 25W, 3: 30W + 35W, 4: 40W + 45W) is displayed in Figure 26. Based on this measurement series, heart frequency during pushrim propulsion is higher in all power output intervals in comparison to crank propulsion. Both curves show a rising trend with increasing power output.

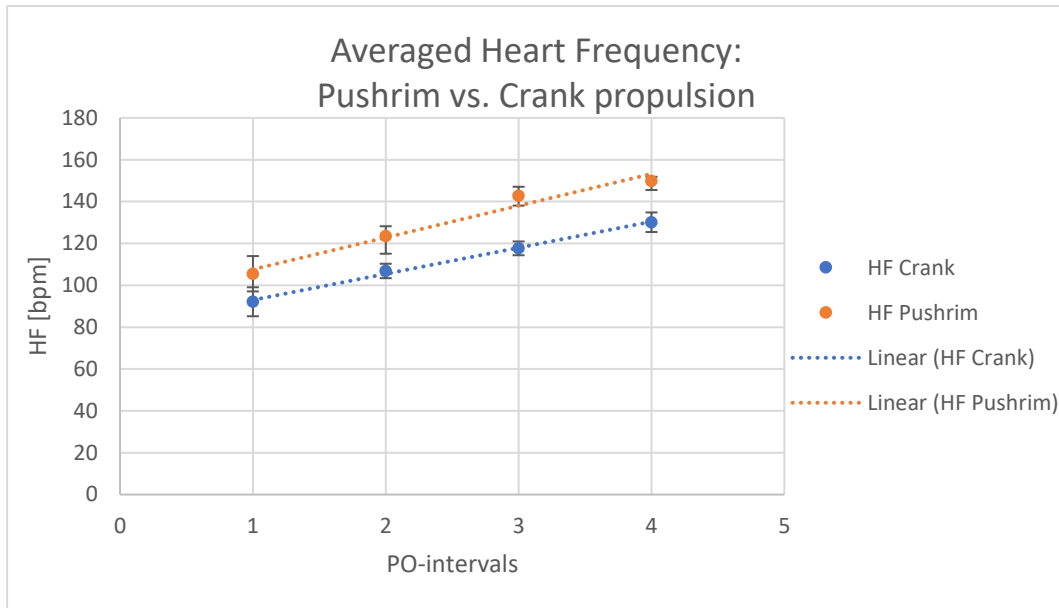


Figure 26: Mean heart frequency of all wheelchair users of pushrim propulsion versus crank propulsion plotted over decimal steps of power output.

4.1.2 Wheelchair users vs. non-wheelchair users

Table 13 shows an overview of mean mechanical efficiency values for the tests performed on all non-wheelchair users at the *TU Wien*. The mean values of crank propulsion tend to be higher compared to pushrim propulsion except for three cases where it is the opposite. The overall mean value for the mechanical efficiency of pushrim propulsion is 7.0 ± 1.1 %, while crank propulsion is 8.3 ± 1.4 %. The mean velocity of pushrim propulsion $v_{\text{mean,pushrim}}$ is 0.8 ± 0.3 m/s, whereas the mean velocity of crank propulsion $v_{\text{mean,crank}}$ is 1.2 ± 0.2 m/s.

For wheelchair users, the same observation concerning the ability to complete all power output levels (from 10 W until 45 W) could be done. All subjects managed to complete a higher (nine subjects) or equal (four subjects) power output level during crank propulsion in comparison to pushrim propulsion. More subjects (eleven subjects) were able to reach the maximum power output level of 45 W using crank propulsion, whereas only four subjects completed the highest power output level during pushrim propulsion. Female wheelchair users (five subjects) quit one power output level earlier during pushrim propulsion.

subject	Pushrim propulsion			Crank propulsion		
	ME1	s(ME1)	v _{mean,pushrim}	ME1	s(ME1)	v _{mean,crank}
	%	%	m/s	%	%	m/s
1	5.680488222	±1.811195672	1.216802168	8.509248795	±1.156665261	1.42071643
2	5.452914162	±0.843492405	0.849206349	6.57720437	±1.497367937	1.303002513
3	7.142756991	±1.084781041		11.32214963	±2.734121918	0.813131519
5	4.66423943	±0.568386681	0.907936508	6.782754185	±1.101428126	1.245956538
6	5.7090068	±1.059270924	0.739130435	6.757663271	±1.273036453	1.007411225
7	7.479805293	±1.280528312	0.942028986	8.605035154	±1.016476206	1.172158967
8	7.265493296	±1.070822762	0.820330969	8.71961628	±1.35415407	0.899153228
9	8.857408551	±1.431462655	0.880614657	8.300395208	±1.149709224	1.285345425
10	7.518951907	±0.850887986	0.787234043	7.1389149	±1.17485592	1.526658955
11	7.496890232	±1.294732137	0.786398467	7.753360544	±1.054995441	1.378611067
12	6.842360365	±0.795697689	0.883141762	10.33701825	±1.403045415	0.882401632
13	10.01862591	±1.131704288	0.529693487	9.52780234	±1.576214499	1.131411841
14	6.696892017	±1.326497026	0.561594203	7.312832735	±1.137598439	1.389366541
mean value	6.986602552	±1.119189198	0.761854772	8.280307358	±1.356128378	1.188871222
			±0.293959583			±0.20885457

Table 13: Mean values and standard abbreviations of mechanical efficiency and velocity of pushrim and crank propulsion of non-wheelchair users.

The following two Figures show a comparison of all non-wheelchair users (*TU Wien*) to wheelchair users (*Weißer Hof*) with regards to the two examined propulsion techniques. The averaged mechanical efficiency for crank propulsion (Figure 27) and pushrim propulsion (Figure 28) is plotted over the power output. Based on this measurement series, mechanical efficiency of crank propulsion in non-wheelchair users (purple) is lower and more stable at smaller power output values in comparison to wheelchair users. Also, the trend does not show such a steep decline, as observed in wheelchair users (orange). From a power output of 30 W upwards, both groups show a similar trend of incline, whereas values of non-wheelchair users tend to be lower than those of wheelchair users. For pushrim propulsion, mechanical efficiency of non-wheelchair users is again more stable throughout all power output levels.

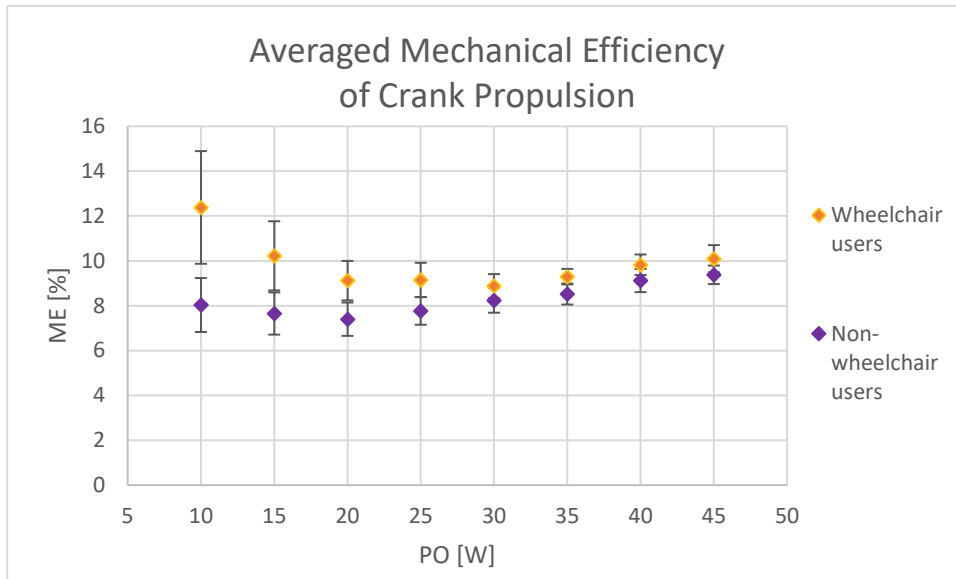


Figure 27: Mean mechanical efficiency of crank propulsion for all non-wheelchair users compared to wheelchair users plotted over the power output.

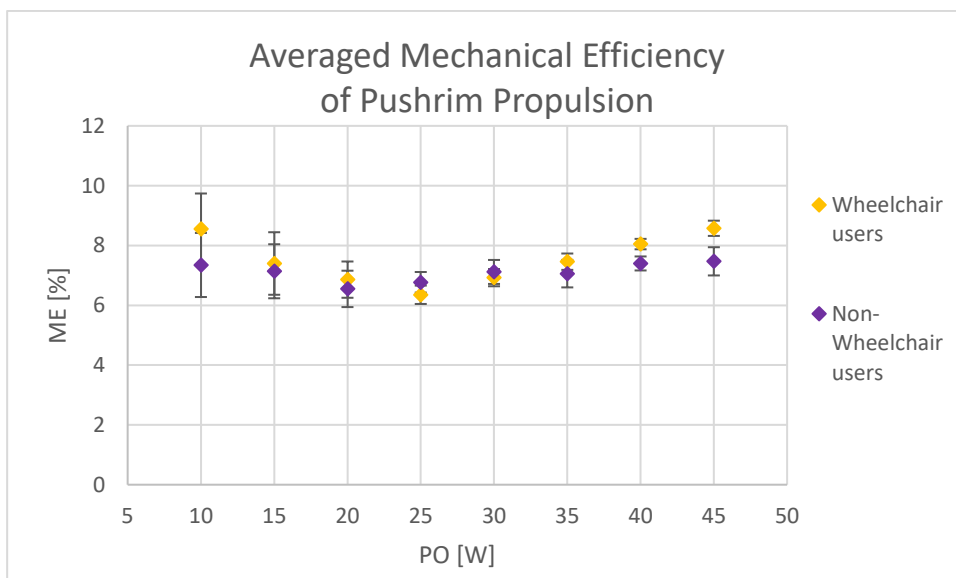


Figure 28: Mean mechanical efficiency of pushrim propulsion for all non-wheelchair users compared to wheelchair users plotted over the power output.

The mean heart frequency of all non-wheelchair users is compared with the mean values of the wheelchair users and this is plotted over decimal steps of power output (1: 10W + 15W, 2: 20W + 25W, 3: 30W + 35W, 4: 40W + 45W). Figure 29 shows the averaged heart frequency values during crank propulsion and Figure 30 shows the averaged values during pushrim propulsion. Based on this measurement series for crank propulsion, it can be observed that non-wheelchair users have a higher mean heart frequency at all power output intervals with a similar overall rising trend. This is not the case for pushrim propulsion as mean heart frequency for non-

wheelchair users is only slightly higher at interval 1 (10W+15W) and interval 4 (40W+45W). At power output interval 2, the mean value is slightly lower, whereas at interval 3, a gap of 21 bpm between non-wheelchair users and wheelchair users can be observed. In comparison to wheelchair users, non-wheelchair users have much higher heart frequency during the last interval when comparing pushrim propulsion to crank propulsion. Nevertheless, both curves show a rising trend with increasing power output.

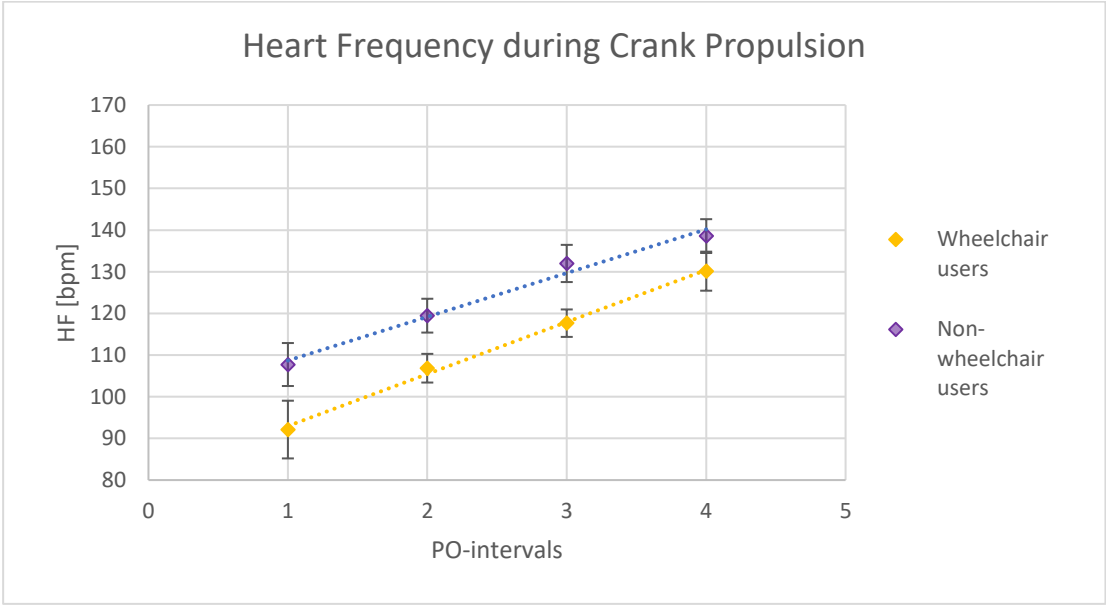


Figure 29: Mean heart frequency of all non-wheelchair users compared to all wheelchair users during crank propulsion plotted over decimal steps of power output.

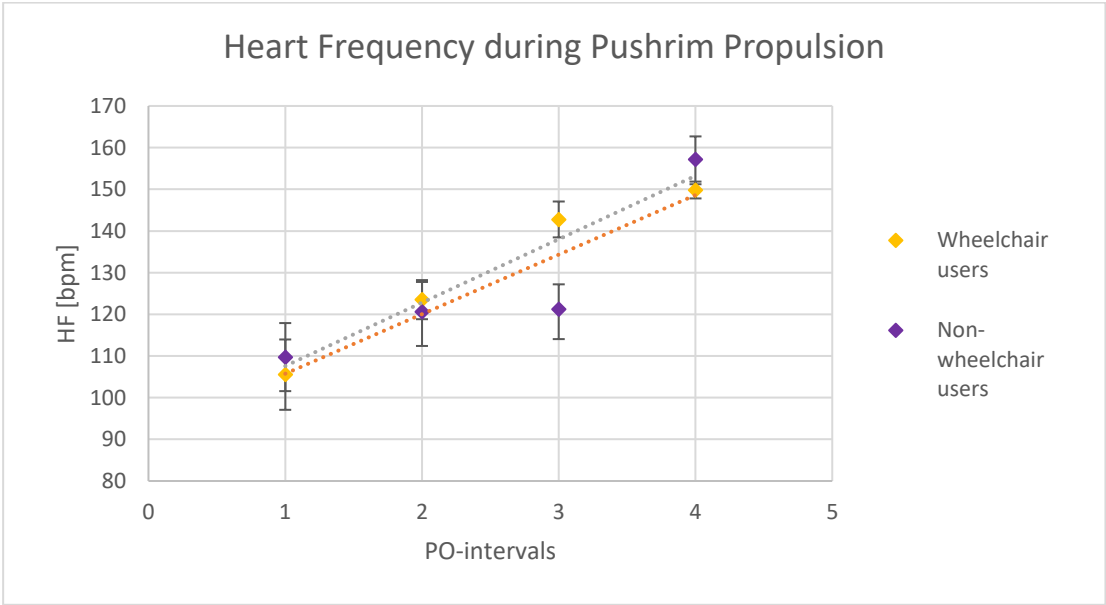


Figure 30: Mean heart frequency of all non-wheelchair users compared to all wheelchair users during pushrim propulsion plotted against decimal steps of power output.

4.2 EMG

Results of EMG data post processing and analysis for crank propulsion are presented in this subchapter.

Figure 31 shows the raw EMG signal of *musculus deltoideus medialis* as well as the EMG signal after full wave rectification, which is the first step of filtering. RMS-filtered EMG signals are displayed in Figures 32 and 33.

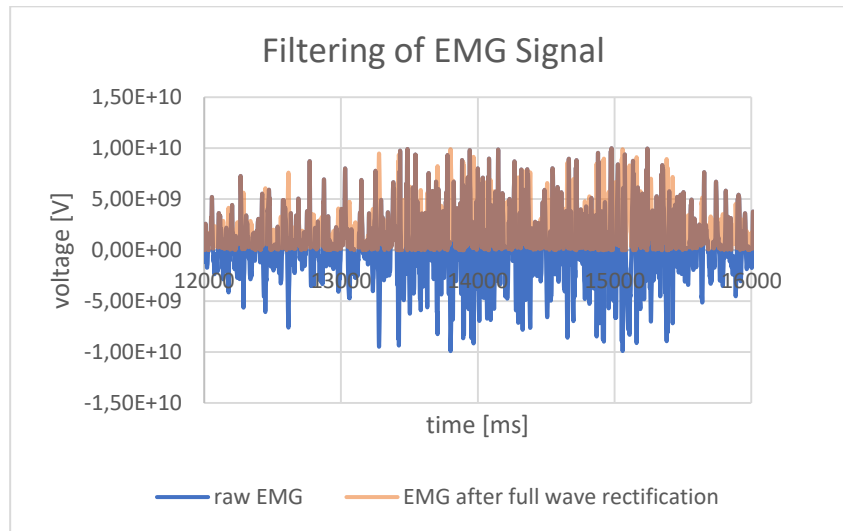


Figure 31: The raw EMG signal, the EMG signal after full wave rectification and the RMS-filtered EMG signal.

The original frame scale of five cycles (for crank propulsion) or three strokes (for pushrim propulsion) are normalized into a percentage scale and are averaged to one cycle as described in 3.7.3. The final mean cycles of all subjects for crank propulsion at a power output of 25 W and 35 W are presented in Figure 32, and the equivalent for pushrim propulsion are illustrated in Figure 33.

Based on this measurement series, none of the muscles falls below 5 % muscle activity for crank propulsion at 25 W power output. *Musculus deltoideus medialis* and *musculus triceps brachii caput longum* show similar curve progression with an increasing tendency from 10 % to 20 % (for *musculus deltoideus medialis*) or 25 % (for *musculus triceps brachii caput longum*) reaching the peak shortly before the halfway point of the cycle and decreasing afterwards. These two muscles show a high range of muscular activity. *Musculus biceps brachii* has a smaller peak of 10 % shortly after the two previously mentioned muscles and in the middle of the cycle. *Musculus triceps brachii caput laterale* is around 10 % in the first part of the cycle and rises from 5 % to 15 % in the second half of the cycle.

As seen on the right of Figure 32, in the final mean cycle for crank propulsion at a power output of 35 W, a similar graph to 25 W can be observed for all muscles. Nevertheless, the peak values are slightly lower with 22 % (for *musculus triceps brachii caput longum*) and 18 % (for *musculus deltoideus medialis*). *Musculus triceps brachii caput laterale* has a peak of 15 % at around 30 % of the cycle and again rises from 8 % to 18 % in the second half of the cycle, which is slightly higher than observed at 25 W power output. *Musculus biceps brachii* has again a low peak of 10 % in the middle of the cycle. All curves do not fall below 5 %.

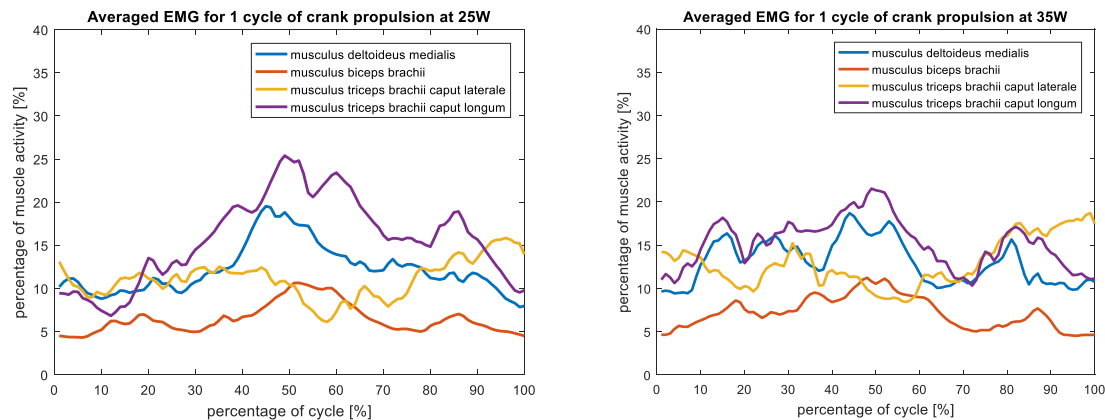


Figure 32: Mean EMG cycle of five normalized cycles of all subjects (except subject 1) at a power output of 25 W (left) and 35 W (right) during crank propulsion.

Based on this measurement series, all muscles except *musculus triceps brachii caput longum* (which has almost the same maximum value) reach higher maximum forces during pushrim propulsion at 25 W power output compared to crank propulsion. In the first 35 % of the cycle, *musculus biceps brachii* and *musculus triceps brachii caput longum* are most active, whereas in the second part of the cycle, *musculus deltoideus medialis* and *musculus triceps brachii caput laterale* show higher excitation levels. *Musculus deltoideus medialis* shows the highest peak value of 35 % (shortly before halfway through the cycle), which is 15 % more than during crank propulsion. A similar curve progression is observed for *musculus triceps brachii caput laterale*, which reaches its maximum value of 24 % at the same time as *musculus deltoideus medialis*. *Musculus biceps brachii* has a peak of 24 % at the beginning of the cycle, after which the level of activation decreases to 8-10 %. After a peak of 22 % at 28 % of the cycle, the muscular activity of *musculus triceps brachii caput longum* decreases and stays at an activation of around 8-10 %.

The final mean cycle for pushrim propulsion at a power output of 35 W, which is presented on the right side of Figure 33, shows a similar graph. Nevertheless, it is not completely equal as all peak values are lower than at 25 W. *Musculus deltoideus medialis* reaches a minimum value

of 27 % at 50 % of the cycle and *musculus triceps brachii caput longum* has the same peak of 23 % shortly before the halfway point of the cycle, but it has a higher level of activation (15-20 %) in the second half of the cycle than observed at 25 W power output. *Musculus biceps brachii* has its maximum of 19 % at the beginning of the cycle and a second lower peak of almost 15 % in the middle of the cycle. *Musculus triceps brachii caput laterale* is between 13 % and 22 % during 10-80 % of the cycle.

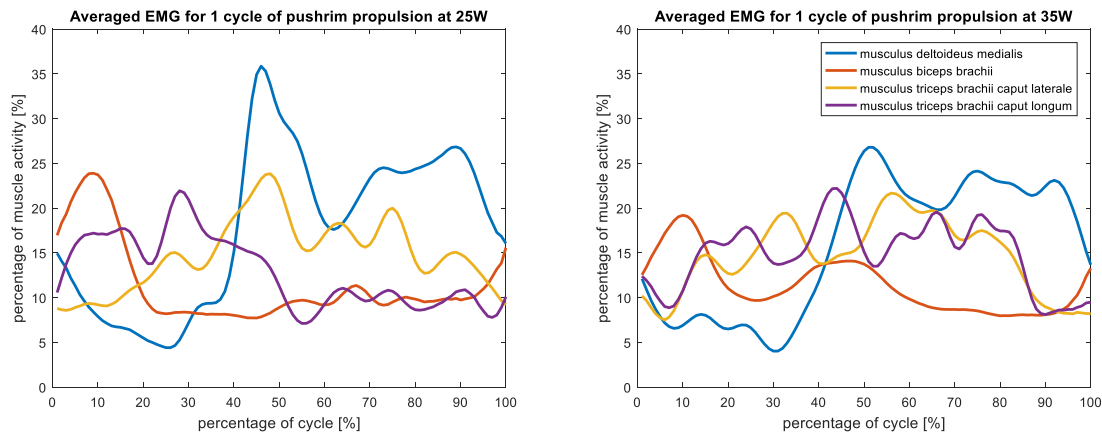


Figure 33: Averaged EMG of all subjects (except subject 1) for 1 cycle at a power output of 25 W (left) and 35 W (right) during pushrim propulsion. The legend applies for both figures.

On-/off-analysis is performed for 25 W and 35 W for *musculus deltoideus medialis*, *musculus biceps brachii*, *musculus triceps brachii caput laterale* and *musculus triceps brachii caput longum*.

The results for crank propulsion at a threshold of 12 % are presented in Figures 34 and 35. 0 % marks the start of the cycle for crank propulsion, as sketched to the right of the figure. 100 % shows the position when one cycle is completed. When the muscle is considered as being “on”, it is marked on the y-axis.

Both Figures (34 and 35) show that *musculus triceps brachii caput longum* is the muscle which is most active at a 25 W as well as at a 35 W power output. This muscle is “on” from 18-96 % at 25 W, and from 5-68 % as well as from 72-95 % at 35 W. *Musculus biceps brachii* and *musculus triceps brachii caput laterale* are active in the beginning and end of the cycle but are “off” between 45-70 % of the cycle. The latter also shows activation around 30-40 % of the cycle, but the dotted character indicates an alternating pattern of being “on” and “off” during this period. Additionally, *musculus biceps brachii* is active for a short period from 33-38 % at 35 W. At a 25 W power output, *musculus deltoideus medialis* is active from 35-79 % of the

cycle, while for a 35 W power output, it shows excitation from 9-60 % and from 74-83 % of the cycle.

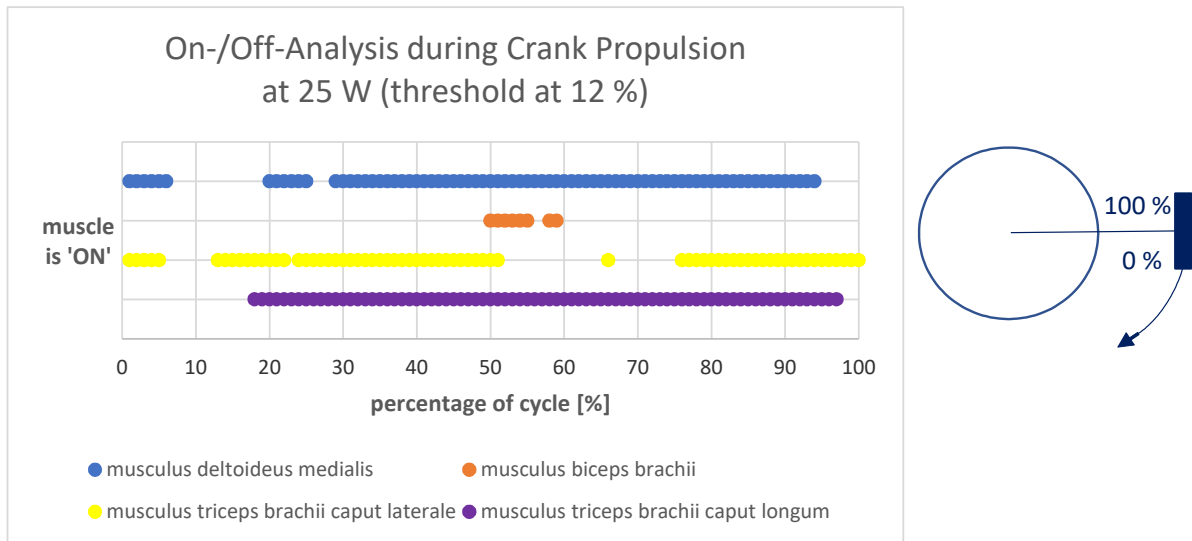


Figure 34: On-/off-analysis of crank propulsion for power output of 25 W at a threshold of 12 %.

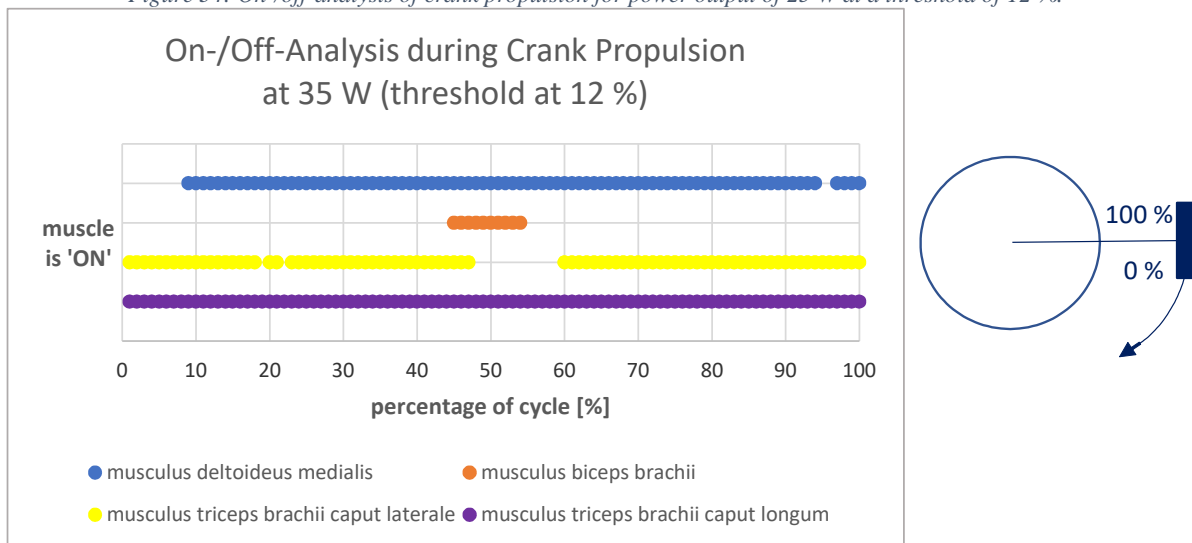


Figure 35: On-/off-analysis of crank propulsion for power output of 35 W at a threshold of 12 %.

Figures 36 and 37 present the results of on-/off-analysis for pushrim propulsion at a threshold of 12 %. The arm position during propulsion is sketched above the graphs.

Both Figures show long periods of activity for all muscles. *Musculus deltoideus medialis* is active during the starting moment and from 38-100 % of the cycle at a 25 W as well as at a 35 W power output. *Musculus biceps brachii* shows short activation at the beginning at 25 W and additionally at 50 % at 35 W. Based on this measurement series, *musculus triceps brachii caput longum* and *musculus triceps brachii caput laterale* are active between 10-86 % of a

cycle at 35 W. At a 25 W power output, *musculus triceps brachii caput longum* is labelled as “on” in the first part of the cycle, whereas *musculus triceps brachii caput laterale* is active between 20-96 %.

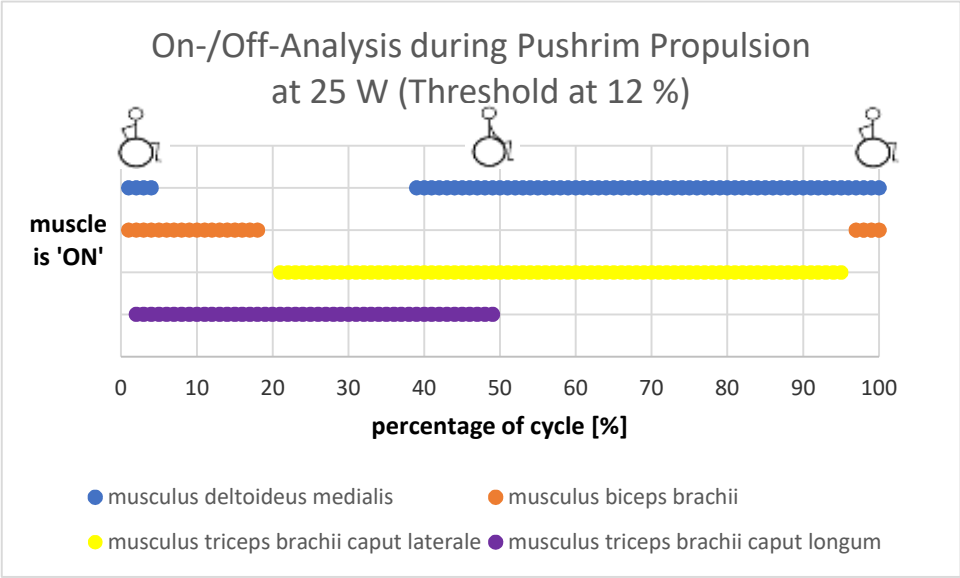


Figure 36: On-/off-analysis of crank propulsion for power output of 25 W at a threshold of 12 %.

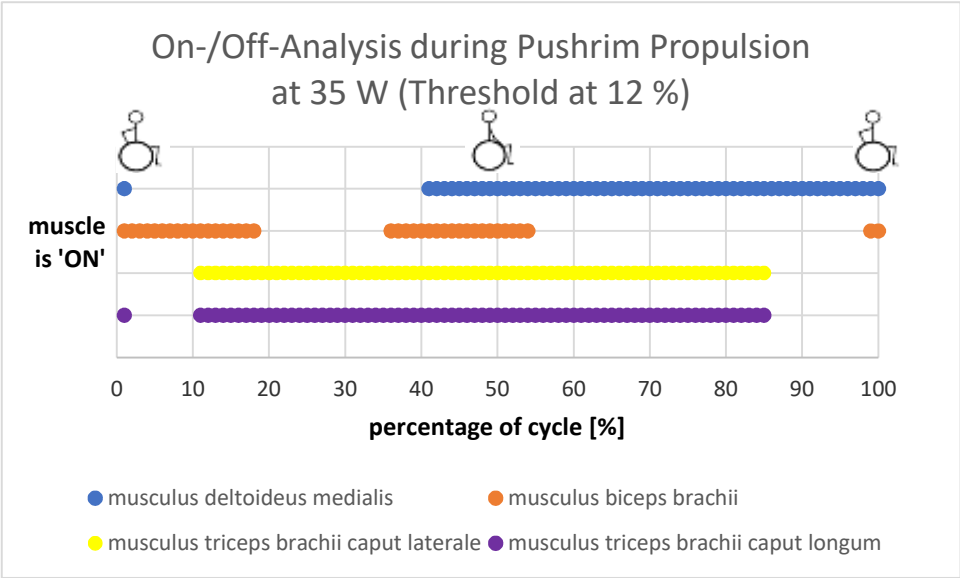


Figure 37: On-/off-analysis of crank propulsion for power output of 35 W at a threshold of 12 %.

4.3 Kinematic Analysis with OpenSim

In the following subchapter, figures of analysis data obtained by *OpenSim* and *Inverse Kinematics* are displayed.

The following figures show the mean angle values during crank propulsion, with the corresponding standard deviations of elevation plane (flexion and extension), shoulder elevation (adduction and abduction) and shoulder rotation (internal/external), elbow flexion, wrist deviation (ulnar/radial) and wrist flexion in degrees displayed on the propulsion cycle as a percentage. The ergonomic range of the joint is indicated by two bold lines. From the wheelchair user's perspective, 0° marks the x-axis in a positive direction or the horizontal right position of the wheel (moving clockwise) and 100 % defines the position after completing one propulsion cycle, as sketched next to the graphs below.

Figure 38 displays the elevation plane which varies between $+10^\circ$ and -64° for both 25 W and 35 W. Positive angle values represent the flexion of the shoulder, whereas negative angle values signify the extension of the shoulder. The maxima of 64° are reached after 53 % and 52 % of the cycle.

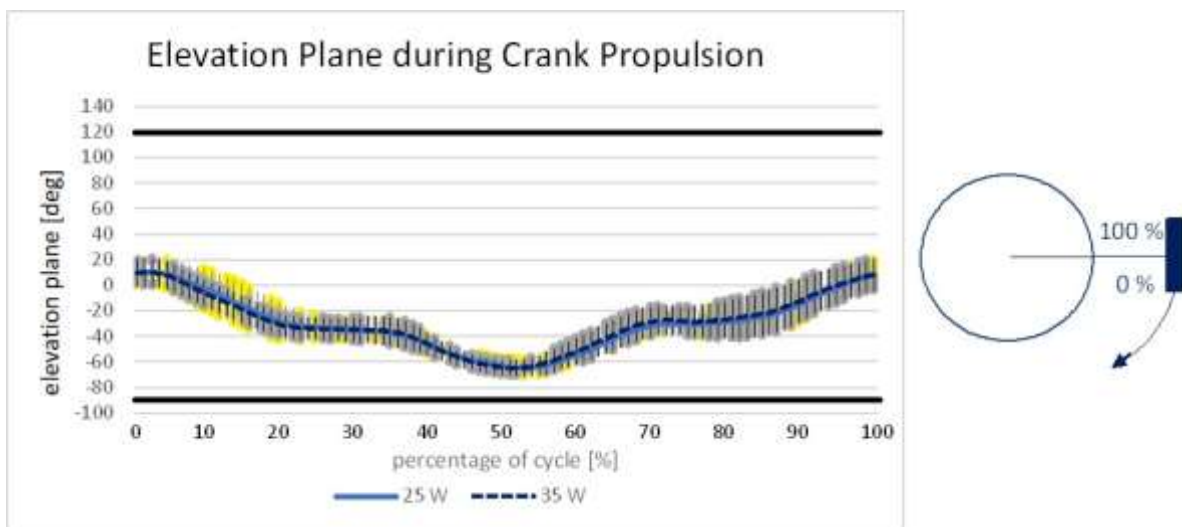


Figure 38: Elevation plane during crank propulsion (flexion +, extension -). The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device.

Figure 39 displays the shoulder elevation angle which varies between $+62^\circ$ and $+34^\circ$ for both 25 W and 35 W. Positive angle values represent the abduction of the shoulder, whereas negative angle values denote the adduction of the shoulder. The maxima of 62° are reached after 53 % of the cycle.

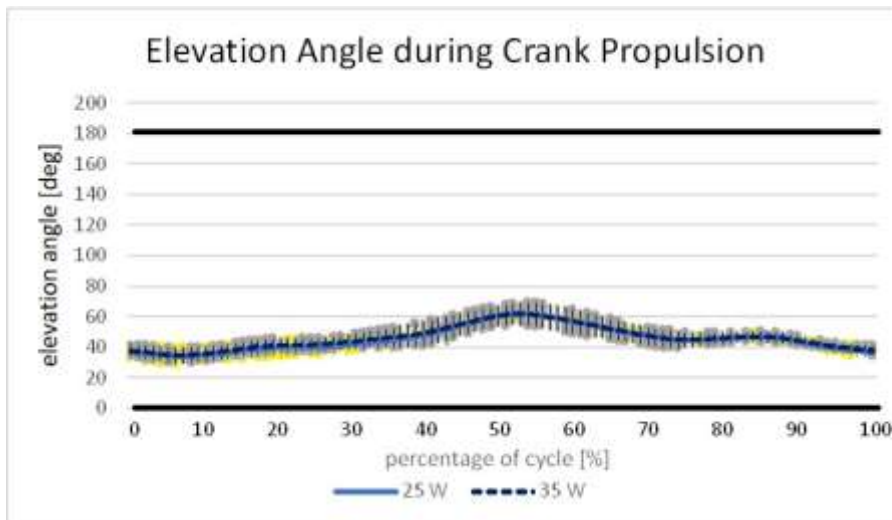


Figure 39: Shoulder elevation angle during crank propulsion (abduction +, adduction -). The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device.

Figure 40 displays the shoulder rotation angle which varies between $+14^{\circ}$ and -20° for 25 W, and between $+14^{\circ}$ and -21° for 35 W. Positive angle values represent the internal rotation of the shoulder, whereas negative angle values signify the external shoulder rotation. The maxima of 20° (for 25 W) and 21° (for 35 W) are reached after 58 % and 57 % of the cycle.

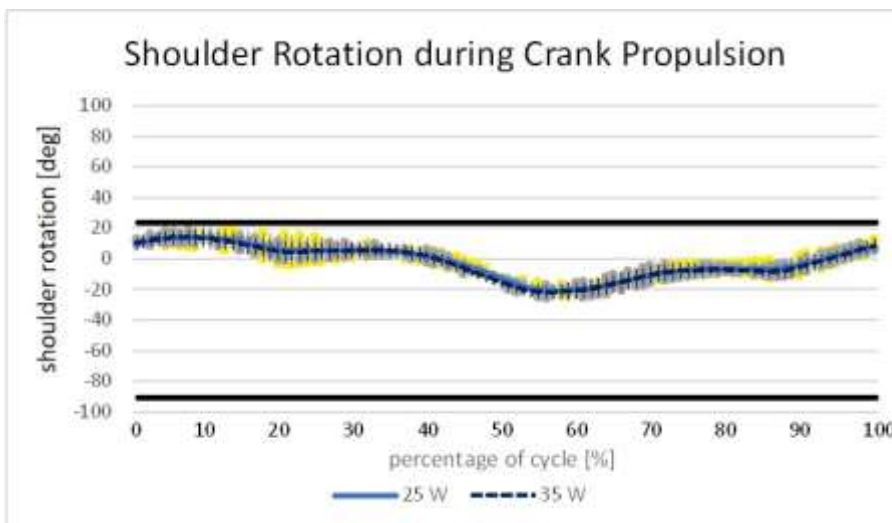


Figure 40: Shoulder rotation angle during crank propulsion (internal +, external -). The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device.

The elbow flexion angle (Figure 41) varies between +90° and +37° for 25 W, and between +86° and +38° for 35 W. The maxima of 90° (for 25 W) and 86° (for 35 W) are reached after 60 % of the cycle.

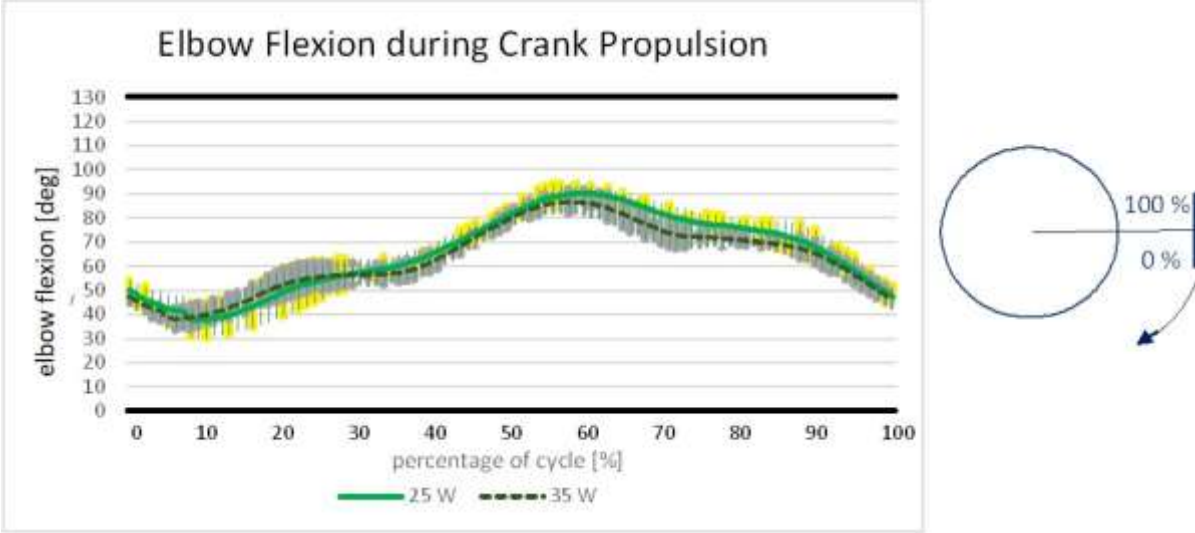


Figure 41: Elbow flexion angle during crank propulsion. The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device.

The wrist deviation angle varies between -10° and -16° for 25 W, and between -11° and -18° for 35 W, as shown in Figure 42. Positive angle values represent the ulnar deviation of the wrist and negative angle values represent the radial deviation. The graph shows that only radial deviation can be observed. The maxima of 16° and 18° are reached after 52 % of the cycle.

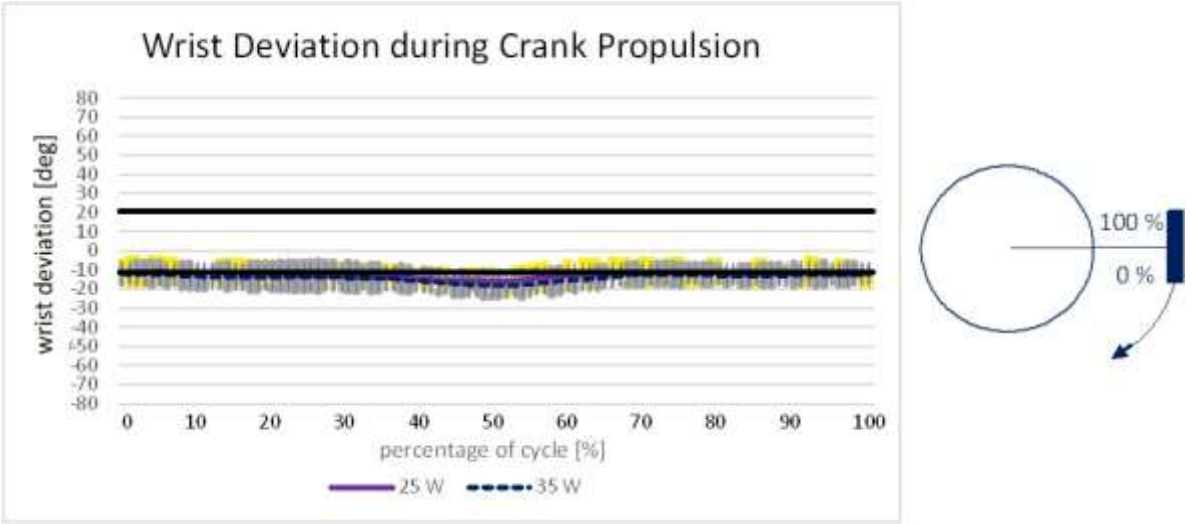


Figure 42: Wrist deviation angle during crank propulsion (ulnar +, radial -). The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device.

The wrist flexion angle (Figure 43) varies between -12° and -23° for 25 W, and between -17° and -30° for 35 W. Positive angle values represent flexion of the wrist while negative angle values stand for extension of the wrist. The maxima of 23° and 30° are reached after 92 % and 74 % of the cycle.

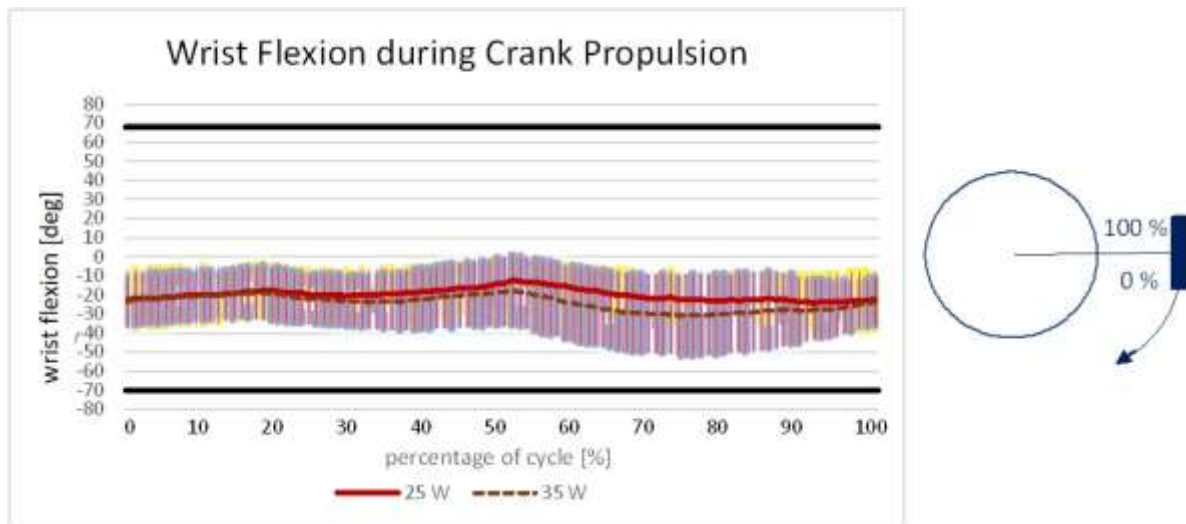


Figure 43: Wrist flexion angle during crank propulsion. The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device.

Table 14 represents the absolute maximum values of the angles of shoulder rotation, elbow flexion, radial and ulnar deviation, as well as wrist extension and wrist flexion at 25 W and 35 W during crank propulsion. The numbers in brackets represent the standard deviation. The lower, second part of the table shows values for comparison with other experiments and pushrim propulsion results of this measurement series. The first three rows are comparable results by Kurup et al. in [11] who have carried out examinations on non-wheelchair users. Wrist angles at different power output levels but at a comparable speed of 1.2 m/s are listed. Also, angle values of the wrist by Boninger et al. in [4] during pushrim propulsion are added to the table. The last two rows are mean maximum angle values for shoulder rotation and elbow flexion of wheelchair users obtained by this measurement series.

	elevation plane	elevation angle	shoulder rotation	elbow flexion	radial deviation	ulnar deviation	wrist extension	wrist flexion
	[deg]	[deg]	[deg]	[deg]	[deg]	[deg]	[deg]	[deg]
max. 25 W crank prop.	64.62 (11.3)	62.39 (4.0)	20.54 (5.7)	90.27 (39.8)	16.10 (6.1)	0.0	23.83 (13.5)	0.0
max. 35 W crank prop.	64.56 (11.4)	62.10 (5.6)	21.33 (4.8)	86.74 (7.9)	18.09 (6.3)	0.0	30.53 (16.7)	0.0
comparison								
max. 20 W crank prop.	-	-	-	-	2.5 (2.4)	16.0 (12.1)	28.5 (3.5)	0.0
max. 40 W crank prop.	-	-	-	-	1.9 (2.7)	12.2 (11.0)	33.1 (2.1)	0.0
max. 14 W pushrim at similar speed	-	-	-	-	25.1 (9.0)	20.0 (5.0)	38.6 (8.8)	7.1 (5.2)
pushrim propulsion 25 W	59.9 (8.8)	82.2 (4.2)	46.83 (9.1)	78.56 (5.2)	-	-	-	-
pushrim propulsion 35W	59.2 (8.7)	82.1 (3.7)	50.98 (7.0)	80.06 (5.7)	-	-	-	-

Table 14: Absolute mean maximum angle values during crank propulsion at power output of 25 W and 35 W and values for comparison to literature and pushrim propulsion.

Figure 44 shows the course of the mean shoulder flexion and extension of one stroke of pushrim propulsion. The sketches next to the curve show the right arm's position during one stroke. 0 % marks the beginning of the stroke and at 50 % the push is performed, after which the arm moves back to the initial position. The shoulder is extended during the whole stroke and reaches a maximum angle of 59° at the beginning of the stroke.

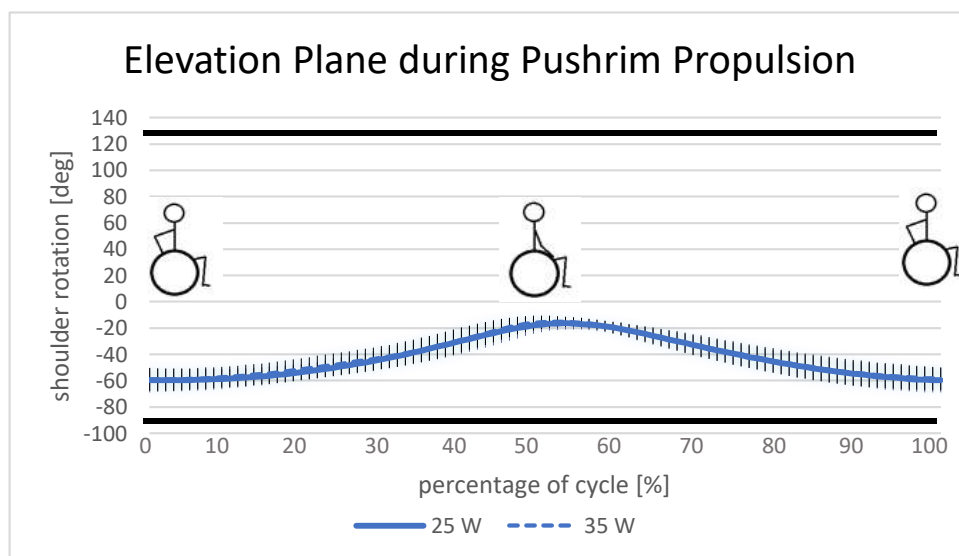


Figure 44: Elevation plane during pushrim propulsion (flexion +, extension -). The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device.

The shoulder abduction and adduction (Figure 45) during pushrim propulsion varies between +82° and +33° for 25 W, and between +82° and +35° for 35 W. The maxima of 82° are reached at the end of the stroke. Positive angle values represent the abduction of the shoulder whereas negative angle values represent the adduction of the shoulder.

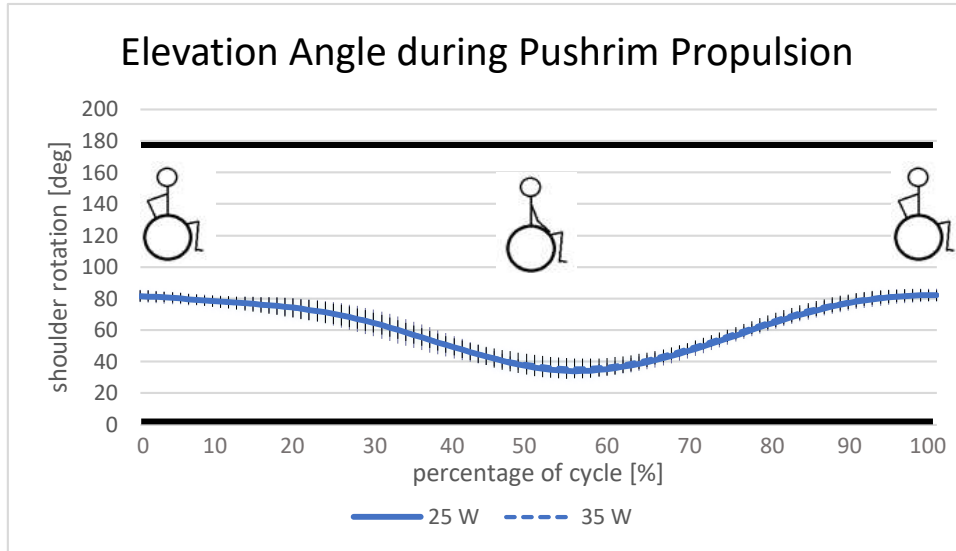


Figure 45: Elevation angle during pushrim propulsion (abduction +, adduction -). The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device.

The mean maximum angles of shoulder rotation, 46° and 50°, are reached at 49 % and 48 % of the stroke at 25 W and 35 W (Figure 46). Positive angle values represent the internal rotation of the shoulder whereas negative angle values denote the external shoulder rotation. As 27° and 27 ° represent the mean minimum angles, the movement is performed within the internal rotation of the shoulder only.

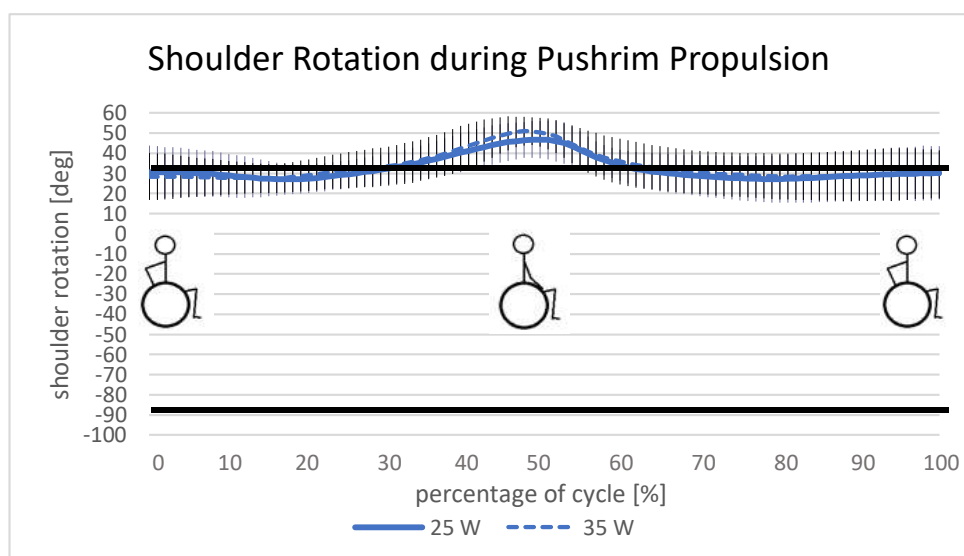


Figure 46: Shoulder rotation angle during pushrim propulsion (internal +, external -). The bold lines indicate the ergonomic joint region. Sketches show the position of the right arm during one stroke.

The elbow flexion angle (Figure 47) during pushrim propulsion varies between +78° and +30° for 25 W and between +80° and +30° for 35 W. The maxima of 78° (for 25 W) and 80° (for 35 W) are reached after 61 % of the cycle.

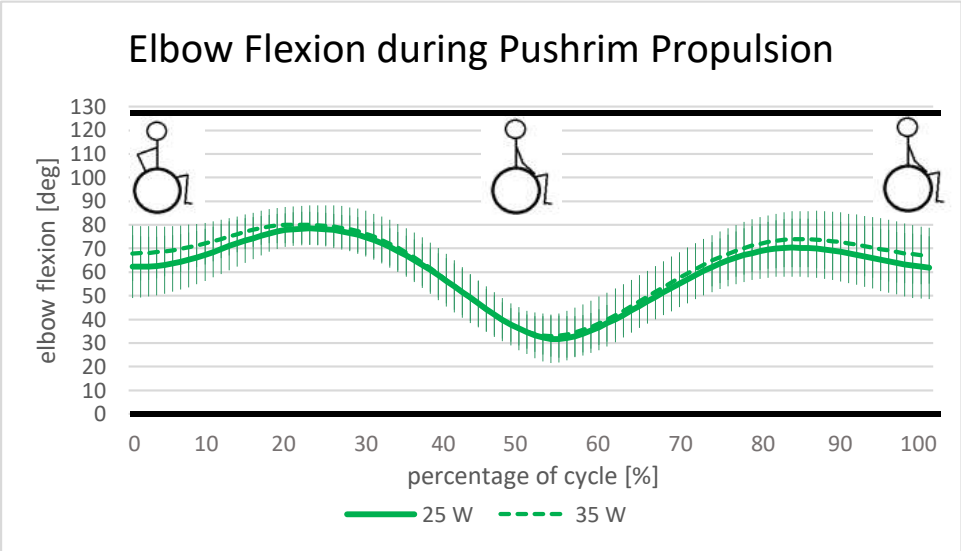


Figure 47: Elbow flexion angle during pushrim propulsion. The bold lines indicate the ergonomic joint region. Sketches show the position of the right arm during one stroke.

5. Discussion

The aim of this thesis was to present a suitable methodological approach for the post-processing and analysis of the measurement series performed at the *Rehabilitation Centre Weißer Hof*. This chapter discusses the presented methodological approach and the results themselves with regards to mechanical efficiency, heart frequency, muscular activity and on-off analysis and joint kinematics. Differences between the novel handle-based device and conventional pushrim propulsion mechanism are emphasised and are put into context of previous research. Additionally, limitations and suggestions for improvement as observed during the empirical application are introduced.

Methodological approach

In summary, the applied methodological approach for post-processing and analysing measurement data for comparing crank propulsion to pushrim propulsion produced usable results. The spirometric, electromyographic, kinetic and kinematic data was evaluated successful. The developed *Matlab* code may be used for future measurement series.

Reiterating the individual steps of the methodological approach, the following limitations and suggestions for improvement come to mind:

The first step was to post-process motion data in *Cortex*. As the signals from all markers were successfully obtained for the whole measurement duration, no further adjustments were necessary.

Next, kinetic and kinematic datasets during crank propulsion were aligned to the same starting time effectively, using *Matlab*. The created ANALYSIS file of the handle and test rig was provided to the research group for further use. No ANALYSIS file could be created for the *SmartWheel* data, as abnormalities in the recording of the device made a matching with the recording starting time of the test rig data unfeasible. According to the research group, this device has previously caused problems. Therefore, maintenance of the *SmartWheel* or the consultation of an expert is advisable for future measurement series.

Then, electromyographic data was filtered and MVC normalization was applied in *Matlab*. As a result, a mean cycle and a mean on-/off-figure of all wheelchair users for four muscles were created. Another benefit is the rescaling to a percent of a reference value that is unique and standardized for all subjects [17]. This eliminates the influence of signal detection conditions and enables direct quantitative comparison of EMG findings between subjects [17]. Before the

averaging process, plots of the mean cycles are checked visually and muscle groups which exceed 100 % of its activity were excluded in order to not falsify the results. An explanation for the MVC data exceeding 100 % is stated by Konrad et al. in [17]: Often supramaximal EMG data can be observed for submaximal dynamic activities what may be caused by electrical superposition within submaximal movements. As a result, MVC data can easily become invalid. It is also mentioned that detection conditions result in a 10 to 15 % variance of the outcome. A possible way to avoid this issue, is to perform MVC tests only on well-instructed subjects. Additional time for instructing and training would be necessary.

An additional issue regarding MVC tests is found in the test procedure of the maximum voluntary contraction test, as the force measuring handle was held by a researcher. A more stable measurement positioning is suggested by Konrad et al. in [17], which also includes *musculus deltoideus anterior/posterior* and *musculus pectoralis*. For this approach, a bench would be needed for MVC of *musculus biceps brachii* and *musculus triceps brachii*, which may be difficult to realize for wheelchair users. Yet, a special construction could lead to an improved outcome.

An important point to consider in the methodological approach is accounting the sensitivity of the EMG signal in the analysis. The range of EMG signals starts from a few microvolts and may have been influenced by artefacts [17]. Artefacts are signal disturbances that are not caused by the actual signal being measured [17]. They may be triggered by external noise sources such as electrostatic or electromagnetic fields. Also, motion artefacts of the electrodes because of highly dynamic movement or stiff clothing above the electrode may falsify the signal [17]. As a result, the EMG signal may show much greater amplitude than in actuality [17]. Proper skin preparation and electrode positioning avoids the impact of artefacts on the measured signal [17]. In this measurement series, EMG sensors were additionally fixed with tapes in order to decrease the risk of movement artefacts. Slight loosening during the propulsion movement, however, may have influenced the EMG signal. The presented methodological approach reduced the influence of artefacts by filtering the signal during the post-processing procedure in *Matlab*.

Another point of criticism is the selection of the threshold for on-/off-analysis of the muscle activity during propulsion which was based on visual inspection. It was taken into account that this is not the most common choice for selection and is biased by personal skills. Besides being time consuming, this approach may be applied for small datasets which is applicable in this measurement series as stated by Drapala et al. in [34]. For future analysis, an alternative

approach may be advisable. Therefore, a multiplication factor (typically a factor of 2 or 3) of the standard deviation range of the EMG baseline is defined as a threshold. When the muscle activity exceeds this threshold, the muscle is labelled as “*on*” or activated. In order to obtain this baseline, an additional measurement of the baseline when all muscles are relaxed have to be recorded.

For the evaluation of the mechanical efficiency and the heart frequency during pushrim propulsion and crank propulsion, based on the spirometric data, *Excel* was used. Whilst the processing in *Excel* produced stable results, future research with large data sets may want to consider using *Matlab* for this step as well in order to be more time efficient.

For calculating the mechanical efficiency, Formula 2 was chosen instead of Formula 3 or 4. The decision for Formula 2 was based on the fact that the outcomes of the three formulas did not differ largely, with Formulas 2 and 4 showing very similar values as observed in Figure 23. Formula 2 was finally chosen due to its higher use in publications and research. As such, it can be observed by De Groot et al. in [31], by van Asbeck et al. in [32] and by van der Woude et al. in [33]. Furthermore, it includes more parameters (V_{O_2} , RER) from the spirometry measurements when compared to the other formulas.

Finally, kinematic analysis in *OpenSim* was successfully applied on all subjects for crank propulsion. Mean cycles for the angle joints were created in *Matlab*. For pushrim propulsion, only figures featuring shoulder flexion/extension, shoulder abduction/adduction, shoulder rotation and elbow flexion produced usable results. For wrist extension/flexion and wrist deviation, a more complex musculoskeletal model may be considered. REACTION files were provided to the research group and may be used for the *Inverse Dynamics* tool of the software.

Mechanical efficiency and heart frequency

Mechanical efficiency and heart frequency of wheelchair users as well as non-wheelchair users are higher during crank propulsion than during pushrim propulsion. No differences can be stated between wheelchair users and non-wheelchair users. This general finding is based on the following insights:

The overall mean value of the mechanical efficiency for wheelchair users of pushrim propulsion ($ME_{\text{pushrim}} = 7.7 \pm 1.2 \%$) and of crank propulsion ($ME_{\text{crank}} = 10.1 \pm 2.1 \%$) is within the range of previous studies. Similar results can be found in papers such as [31] by De Groot et al. with a ME between $7.45 \pm 0.87 \%$ and $8.11 \pm 0.56 \%$ for pushrim propulsion. Also, van Asbeck et

al. list a ME of 6.6 ± 1.9 % at the start of an active rehabilitation and 7.2 ± 1.8 % at the time of discharge from rehabilitation after three months in [32]. Van der Woude et al. states in [33] a ME of 7.8 ± 1.6 % during handbike propulsion at a velocity of 1.3 m/s at 0.7 % inclination and a power output of 24.3 ± 5.4 W. Those results are comparable to the crank propulsion technique in this study where at a power output of 25 W and 30 W, a ME of 9.15 ± 0.76 % and 8.9 ± 0.54 % can be observed, respectively.

In addition to Figure 24, which shows the mechanical efficiency averaged over all wheelchair users Figure 25 excludes subjects 1 and 4. Those two subjects suffer from paralysis that occurs at a higher level of their spine, and therefore had difficulties keeping their balance whilst seated. The figure shows even more explicit that crank propulsion has a higher mechanical efficiency than pushrim propulsion.

The mean values of ME at 10 W and 15 W show high standard deviation. This may occur due to starting difficulties during a constant rotation as well as during the start and speeding up of the propulsion movement in the beginning. It is assumed that this falsifies the results, with the ME being higher than in actuality. When looking at ME from a power output of 25 W and higher, the resultant trend may arise from a higher degree of homogeneous rotation and steady velocity.

Heart frequency during crank propulsion is lower than during pushrim propulsion. This observation is supported by Robertson et al. in [3], stating that crank propulsion is considered as energetically more efficient and less “straining” for the cardiorespiratory system. Heart frequency rises during both pushrim and crank propulsion which may be due to the increasing effort which correlates with the increasing power output. This seems reasonable as it is an exhaustion test and power output increases with time.

It must be considered that the procedure of the two exhaustion tests was not fully optimal. Ideally, the measurements should be done on two separate days with only one test a day or both tests in a different order. However, this procedure was not possible at this measurement series and therefore the resting time was only five minutes which had to be accepted due to the rehabilitation schedule of the wheelchair users at the *Rehabilitation Centre Weißer Hof*. Also, being a smoker seems to not have had a considerable influence. These points of consideration also apply to the next section.

Wheelchair users vs. non-wheelchair users

Comparison to additional measurements performed on non-wheelchair users at *TU Wien* yielded similar results:

The overall mean value of mechanical efficiency for non-wheelchair users is lower for both pushrim propulsion ($ME_{\text{pushrim}} = 7.0 \pm 1.1 \%$) and crank propulsion ($ME_{\text{crank}} = 8.3 \pm 1.4 \%$) in comparison to wheelchair users ($ME_{\text{pushrim}} = 7.7 \pm 1.2 \%$, $ME_{\text{crank}} = 10.1 \pm 2.1 \%$). Nevertheless, ME may be considered the same for wheelchair users and non-wheelchair users.

For crank propulsion, the bigger difference of standard deviation at lower power output levels may be due to the higher number of non-wheelchair users (13 to 8) as well as wheelchair users may have a better body balance as muscles of the lower extremities can be used for stability purposes.

For the ME of pushrim propulsion, no difference between wheelchair users and non-wheelchair users can be seen. With regards to pushrim propulsion, De Groot et al. states in [31] that a significant increase in mechanical efficiency can be observed after three weeks of practise, which evidently was not the case in this measurement series.

During crank propulsion, heart frequency in non-wheelchair users tends to be higher throughout all power output levels. On the one hand, an explanation for this observation can be that the cardio muscular system of wheelchair users adapts to the given circumstances of paralysis. On the other hand, this is not confirmed in pushrim propulsion. It was expected that wheelchair users would have a lower heart frequency as they are more familiar with the propulsion movement than non-wheelchair users, but this was not the case based on these measurements.

Muscular Activity

Most of the expectations concerning muscular activity during crank propulsion and crank propulsion could be confirmed by the mean cycle figures of the targeted four muscles. On-off analysis provided little complementary knowledge. The results are evaluated in more detail in the following paragraphs.

The statement by Rankin et al. in [13] that *musculus biceps brachii* and *musculus triceps brachii caput longum* exhibit large range of work in both propulsion techniques could be confirmed with respect to *musculus triceps brachii caput longum*. The mean cycles at 25 W and 35 W power output (Figures 32 and 33) showed that this muscle had the highest level of activity as well as a large range of muscular activity.

During crank propulsion (Figure 32), it was expected that *musculus biceps brachii* is excited from approximately 25-75 % of the cycle, whereas *musculus triceps brachii caput longum* and *musculus triceps brachii caput laterale* show muscular activity from approximately 75-100 % as well as 0-25 %. This is due to movement pattern, to be more precise the elbow flexion and extension as explained in the Introduction. Besides other muscles, *musculus biceps brachii* enables elbow flexion, elbow extension is performed by *musculus triceps brachii*, thus *musculus triceps brachii caput laterale* being more active than *musculus triceps brachii caput longum*. Furthermore, it is expected that *musculus deltoideus medialis* shows excitation in the middle of the cycle because of the shoulder abduction movement.

Based on this measurement series, these expectations are fulfilled concerning *musculus biceps brachii*, *musculus triceps brachii caput laterale* and *musculus deltoideus medialis*. In Figure 30 *musculus triceps brachii caput longum* shows excitation similarly to *musculus deltoideus medialis* which may be referred to the shoulder extension movement. The high level may also be related to the position of the arm depending on the individual physique. The second peak between 80-90 % of the cycle may originate in the arm movement during the push phase where the elbow passes the trunk.

Muscular activity tends to be lower at 35 W than at 25 W. This finding may be explained by a more homogenous propulsion movement.

During pushrim propulsion, considering Table 1, 2 and 3 (see Introduction) it was expected that during the push phase mainly *musculus deltoideus medialis* (shoulder flexion), *musculus triceps brachii caput longum* (shoulder adduction), *musculus biceps brachii* (shoulder flexion, internal rotation) show excitation. This is confirmed by the graphs in Figure 33. *Musculus deltoideus medialis* has its peak at the end of the push phase when the hand releases the pushrim. The second peak in the end of the stroke may indicate the shoulder abduction when the arm is moved behind the trunk. The fact that *musculus triceps brachii caput laterale* is more active than *musculus triceps brachii caput longum* after 35 % of the stroke may be due to elbow extension.

Overall, the mean cycle diagrams show that during crank propulsion lower peak and thus no excessive forces are produced except of *musculus triceps brachii caput longum*. Also, with *musculus biceps brachii* having a low activation level during pushrim propulsion, it can be concluded that this muscle which is capable of producing high forces is not optimally used during this propulsion technique. Furthermore, *musculus deltoideus medialis* shows a higher

level of activity during pushrim propulsion compared to crank propulsion which may be referred to stabilization demand by the shoulder muscles as mentioned in the Introduction [10].

When evaluating the on-/off figures and numbers little additional conclusion can be drawn as they do not reflect the levels of activity. For crank propulsion (Figures 34 and 35), the fact that *musculus triceps brachii caput longum* and *musculus deltoideus medialis* are “on“ during the expected periods of the cycle, confirms the statement of [10] saying that both muscles are active during both push and pull phase of crank propulsion. Also, *musculus triceps brachii caput laterale* is considered being active in the last and first quarter of the cycle. *Musculus biceps brachii* is stated being “off” due to the choice of the threshold. The on-off figures (Figures 36 and 37) during pushrim propulsion verify the above-mentioned muscle activity pattern.

Kinematic analysis with OpenSim (Joint angles)

Inverse Kinematics results obtained by *OpenSim* provide valuable information concerning angle values during crank and pushrim propulsion and their maximum values. IK was performed on both 25 W and 35 W of both propulsion techniques in order to obtain angle values of shoulder flexion/extension, shoulder adduction/abduction, shoulder rotation, elbow flexion/extension, wrist deviation, wrist flexion/extension. Overall, the ranges of joint motion of crank propulsion were lower when compared to pushrim wheelchair propulsion. There is no major difference between a 25 W and a 35 W power output. Hereafter, individual findings are presented:

The course of the joint angles during crank propulsion corresponds to Figure 5 by Kurup et al. in [10]. It has to be stated that in those figures there is a 180° shift concerning the start of the propulsion cycle. The elevation plane (Figure 38) and shoulder rotation (Figure 40) match to the courses in Figure 5. In addition, shoulder elevation (Figure 39), elbow flexion (Figure 41) and wrist flexion (Figure 42) show the same course, thus having lower or higher values. Wrist deviation (Figure 43) shows high discrepancy as it is not in ulnar deviation but in radial deviation only. Nevertheless, the expected course of ulnar deviation shortly after the middle of the cycle can also be observed based on this measurement series, although being in radial deviation.

The movement pattern during the push phase (shoulder is flexed, adducted and internally rotated) and the pull phase (shoulder is extended, abducted and externally rotated) are verified.

All movements lie within the joint ROM with the exception of constantly lower wrist deviation. Possible reasons for the discrepancies might be the use of a relatively simple musculoskeletal model, the deviation between the subjects and the low sample size.

With regards to pushrim propulsion, the movement characteristics of shoulder flexion combined with adduction and internal rotation as well as elbow extension are confirmed by the obtained figures (Figures 44, 45, 46 and 47). Shoulder rotation exceeds the joint ROM at the end of the push phase which aligns with the release pattern of pushrim propulsion movement and also corresponds to Figure 4.

The values obtained for wheelchair users (Table 14) are comparable to the non-wheelchair users concerning wrist flexion and extension. The joint kinematics of the wrist joint on five non-wheelchair users at a power output of 20 W and 40 W and a mean velocity of 1.2 m/s were analysed by Kurup et al. in [11], propelling the same test rig and handle-based wheelchair propulsion unit used in this measurement series. Those maximum angles are listed in Table 14. Insignificant wrist flexion is observed during crank propulsion whereas a wrist flexion angle of 7.1° may be observed during pushrim propulsion as stated by Boninger et al. in [4]. Wrist extension of 23.83° and 30.53° are comparable to the results by Kurup et al. in [11] (28.5° and 33.1°) who have carried out examinations on non-wheelchair users.

Also, it was not possible to adjust the mounting frame to the bead seat width. The propulsion movement was less homogenous for women than for broader subjects, as women tend to be lightly built. This may have had an influence on these results. Lightly built subjects remarked “uncomfortable” positioning of the arm. This may affect the curves of shoulder rotation, and wrist deviation as well as wrist flexion.

Further recommendations

The following subchapter presents recommendations on the wheelchair and the experiment in general:

With regards to the test rig setup, wheelchair users as well as doctors at the *Rehabilitation Centre Weißer Hof* objected that the handle device complicates the process of getting into the wheelchair. In order to overcome this constraint, a differing handle construction, where it is possible to fold away the propelling device, must be found. To enable propelling of the handle device, smaller wheels (when compared to pushrims) would be required, lessening the ability to navigate over ground sills due to a hampered leveraging effect.

A limitation of this test series is the small sample size. For more precise statements, a larger sample size is required.

Conclusion

Overall, this measurement series verifies that crank propulsion is more advantageous concerning mechanical efficiency, heart frequency and joint ranges of motion. Considering the results provided by Kurup et al. in [11] performed with the same experimental setup, also this measurement series on wheelchair users shows that the groups of muscles activated in the pull and push phase during crank propulsion were similar to conventional pushrim propulsion but at a different level of activity. Furthermore, joint ranges during crank propulsion are considerably reduced in comparison to pushrim propulsion.

Concluding, this novel propulsion mechanism with its special design may be a valuable alternative to pushrim propulsion for long-term wheelchair users who are suffering from joint injuries. Crank propulsion may help to reduce the risk of joint injuries.

6. Bibliography

- [1] World Health Organization, “Fact sheet on wheelchairs,” 2007.
- [2] U. Arnet, S. Van Drongelen, D. J. Veeger, and L. H. V. Van Der Woude, “Force Application during Handcycling and Handrim Wheelchair Propulsion: An Initial Comparison,” *J. Appl. Biomech.*, vol. 29, no. 6, pp. 687–695, 2013.
- [3] R. N. Robertson, M. L. Boninger, R. A. Cooper, and S. D. Shimada, “Pushrim Forces and Joint Kinetics DURING Wheelchair Propulsion,” *Arch Phys Med Rehabil*, vol. 77, no. September, pp. 856–864, 1996.
- [4] M. L. Boninger, R. A. Cooper, R. N. Robertson, and T. E. Rudy, “Wrist Biomechanics during Two Speeds of Wheelchair Propulsion: An Analysis Using A Local Coordinate System,” *Arch. Phys. Med. Rehabil.*, vol. 78, no. 4, pp. 364–372, 1997.
- [5] L. H. . Van der Woude, H. E. J. Veeger, A. J. Dallmeijer, T. W. J. Janssen, and L. A. Rozendaal, “Biomechanics and Physiology in Active Manual Wheelchair Propulsion,” *Med. Eng. Phys.*, pp. 713–733, 2001.
- [6] “Motion Analysis Laboratory.”, M.-M. Institute, Available: <https://www.unmc.edu/media/mmi/pdf/MotionAnalysis.pdf>. [Accessed: 16-Dec-2018]
- [7] H. Ferner and J. Staubesand, W. Platzer, W. Kahle, and H. Leonhardt, "Atlas der Anatomie des Menschen 1", 18th ed. München-Wien-Baltimore: *Urban & Schwarzenberg*, p.204, 1982.
- [8] A. I. Kapandji, "The Physiology of the Joints". *Churchill Livingstone*, pp. 69-135, 1982.
- [9] W. Platzer, W. Kahle, and H. Leonhardt, "Taschenatlas der Anatomie - Band 1: Bewegungsapparat", 4th ed. *Georg Thieme Verlag*, pp. 2-162, 1984.
- [10] N. Babu Rajendra Kurup, M. Puchinger, and M. Gföhler, “Forward dynamic optimization of handle path and muscle activity for handle based isokinetic wheelchair propulsion: A simulation study,” *Comput. Methods Biomech. Biomed. Engin.*, pp. 1–9, 2018.
- [11] N. B. R. Kurup, M. Puchinger, T. Keck, and M. Gfoehler, “Wrist Kinematics and Kinetics during Wheelchair Propulsion with a Novel Handle-based Propulsion

- Mechanism,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2018–July, no. 1, pp. 2146–2149, 2018.
- [12] A. Jenkins, S. D. Gooch, D. Theallier, and J. Dunn, *Analysis of a Lever-Driven Wheelchair Prototype and the Correlation Between Static Push Force and Wheelchair Performance*, vol. 19, no. 3. IFAC, 2014.
- [13] J. W. Rankin, A. M. Kwarciak, W. M. Richter, and R. R. Neptune, “The Influence of Wheelchair Propulsion Technique On Upper Extremity Muscle Demand: A Simulation Study,” *Clin. Biomech.*, vol. 27, no. 9, pp. 879–886, 2012.
- [14] M. Puchinger, 2015.
- [15] M. Puchinger, N. Kurup, and M. Gfoehler, “A Test Rig for Investigating Manual Wheelchair Propulsion Devices,” *TAR-Conference 2017*, vol. 20, no. 1, p. 12, 2017.
- [16] T. Rivers, “SmartWheel User ’ s Guide,”. September 2009, 2010.
- [17] P. Konrad, *The ABC of EMG - A Practical Introduction to Kinesiological Electromyography*, vol. 100. Arizona: Noraxon U.S.A, Inc., 2006.
- [18] Cosmed, “K5 user manual,” 2015.
- [19] “Cortex Software,” *Motion Analysis*. [Online]. Available: <https://motionanalysis.com/products/cortex-software/>. [Accessed: 16-Oct-2018].
- [20] A. Barré, “Mokka - Motion kinematic & kinetic analyzer.” [Online]. Available: <http://biomechanical-toolkit.github.io/mokka/>. [Accessed: 16-Oct-2018].
- [21] “OpenSim.” [Online]. Available: <https://simtk-confluence.stanford.edu/display/OpenSim/Motion+%28.mot%29+Files>. [Accessed: 16-Dec-2018].
- [22] omnia - Cardiopulmonary Diagnostic Site, "One integrated solution for effective Lung Function" , *Metabolic and Body Composition data management* . .
- [23] Math Works, “Math Works Documentation.” [Online]. Available: <https://ch.mathworks.com/help/matlab/ref/atan2.html>. [Accessed: 30-Feb-2019].
- [24] “LabView Timestamp,” 2018. [Online]. Available: <http://www.ni.com/tutorial/7900/en/>. [Accessed: 16-Dec-2018].

- [25] N. Matthew and R. Stones, "The Linux Environment." *Indianapolis, Indiana, US: Wiley*, 2008.
- [26] E. N. Kamavuako, E. J. Scheme, and K. B. Englehart, "Determination of Optimum Threshold Values for EMG Time Domain Features; A Multi-Dataset Investigation," *J. Neural Eng.*, vol. 13, no. 4, pp. 1–10, 2016.
- [27] W. McArdle, F. Katch, and V. Katch, "Exercise Physiology, Nutrition, Energy and Human Performance." *Lippincott Williams & Wilkins*, pp. 208, 2010.
- [28] L. Garby and A. Astrup, "The Relationship Between The Respiratory Quotient And The Energy Equivalent of Oxygen During Simultaneous Glucose And Lipid Oxidation And Lipogenesis," *Acta Physiol. Scand.*, vol. 129, no. 3, pp. 443–444, 1987.
- [29] H. Tropp, "Power Output for Wheelchair Driving on A Treadmill Compared With Arm Crank Ergometry," *Br. J. Sports Med.*, vol. 31, no. 1, pp. 41–44, 1997.
- [30] "OpenSim 3.3." [Online]. Available: <https://simtk-confluence.stanford.edu:8443/display/OpenSim33/>. [Accessed: 20-Dec-2018].
- [31] S. De Groot, D. H. E. J. Veeger, A. P. Hollander, and L. H. Lucas, "Wheelchair Propulsion Technique And Mechanical Efficiency After 3 Wk of Practice," *Med. Sci. Sports Exerc.*, vol. 34, no. 5, pp. 756–766, 2002.
- [32] F. W. van Asbeck *et al.*, "Course of Gross Mechanical Efficiency in Handrim Wheelchair Propulsion During Rehabilitation of People With Spinal Cord Injury: A Prospective Cohort Study," *Arch. Phys. Med. Rehabil.*, vol. 86, no. 7, pp. 1452–1460, 2005.
- [33] L. van der Woude, S. de Groot, F. J. Hettinga, F. Kerkhof, F. Woldring, and F. van Dijk, "Physical Strain of Handcycling: An Evaluation Using Training Guidelines for A Healthy Lifestyle As Defined by the American College of Sports Medicine," *J. Spinal Cord Med.*, vol. 36, no. 4, pp. 376–382, 2013.
- [34] J. Drapala, K. Brzostowski, A. Szpala, and A. Rutkowska-Kucharska, "Two Stage EMG Onset Detection Method," *Arch. Control Sci.*, vol. 22, no. 4, pp. 427–440, 2012.

7. List of Figures

Figure 1: Muscles of the upper extremity. [7].....	12
Figure 2: Sketch of extension, flexion, external rotation, internal rotation, abduction and adduction movement of the shoulder joint.....	13
Figure 3: Ulnar deviation (right) and radial deviation (left) of the wrist joint.[8].....	15
Figure 4: Dynamically optimized propulsion path (up) of novel wheelchair propulsion mechanism. Comparison of the joint range of motions (ROM) of the upper extremity between pushrim propulsion and handle-based propulsion (down).[10].....	18
Figure 5: Joint motion during handle-based propulsion.[10]	19
Figure 6: Test rig wheelchair with the capturing coordinate system.[14]	22
Figure 7: Drive-train of test rig wheelchair.[14]	23
Figure 8: Test rig wheelchair with mounted SmartWheel and Handle, the circle marks the location of the forces measuring sensor (up). Crank propulsion device (down).[11].....	24
Figure 9: SmartWheel (left) and EMG device of Delsys with surface electrodes (right).	25
Figure 10: Screenshot of Cortex software showing marker selection in post-processing mode.	28
Figure 11: Draft and image of experimental setup at the Rehabilitation Centre Weißer Hof.	30
Figure 12: Overview of testing protocol at the Rehabilitation Centre Weißer Hof.....	32
Figure 13: Positions of EMGs (left) and markers (right) attached to subject.....	33
Figure 14: Subject during MVC test of musculus deltoideus medialis (left) and continuous stepwise progressive aerobic test at the Rehabilitation Centre Weißer Hof, wearing spirometry setup (right)...	35
Figure 15: Overview of post-processing.	37
Figure 16: Crank angle calculation.....	41
Figure 17: Tilt angle calculation (- clockwise, + anticlockwise).	42
Figure 18: Sketch of calculation of tilt angle (abbreviated as β) as well as conversion of measured forces from handle coordinate system to global coordinate system.	42
Figure 19: Selection of stroke cycle for pushrim propulsion.	47
Figure 20: Overview of EMG post-processing and analysis.....	48
Figure 21: Mean EMG cycle of crank propulsion at 25 W with threshold of 12 % (red line).	51
Figure 22: Musculoskeletal model used for OpenSim with marking of markers which are visible from this viewpoint.	52
Figure 23: Comparison of different formulas for mechanical efficiency (ME1, ME2, ME3) of subject 6.....	55
Figure 24: Mean mechanical efficiency of all wheelchair users of pushrim propulsion versus crank propulsion plotted over the power output.....	56
Figure 25: Mean mechanical efficiency of all wheelchair users (except subject 1 and subject 4) of pushrim propulsion versus crank propulsion plotted over decimal steps of power output.	56

Figure 26: Mean heart frequency of all wheelchair users of pushrim propulsion versus crank propulsion plotted over decimal steps of power output.....	57
Figure 27: Mean mechanical efficiency of crank propulsion for all non-wheelchair users compared to wheelchair users plotted over the power output.	59
Figure 28: Mean mechanical efficiency of pushrim propulsion for all non-wheelchair users compared to wheelchair users plotted over the power output.	59
Figure 29: Mean heart frequency of all non-wheelchair users compared to all wheelchair users during crank propulsion plotted over decimal steps of power output.....	60
Figure 30: Mean heart frequency of all non-wheelchair users compared to all wheelchair users during pushrim propulsion plotted against decimal steps of power output.	60
Figure 31: The raw EMG signal, the EMG signal after full wave rectification and the RMS-filtered EMG signal.	61
Figure 32: Mean EMG cycle of five normalized cycles of all subjects (except subject 1) at a power output of 25 W (left) and 35 W (right) during crank propulsion.....	62
Figure 33: Averaged EMG of all subjects (except subject 1) for 1 cycle at a power output of 25 W (left) and 35 W (right) during pushrim propulsion.....	63
Figure 34: On-/off-analysis of crank propulsion for power output of 25 W at a threshold of 12 %.	64
Figure 35: On-/off-analysis of crank propulsion for power output of 35 W at a threshold of 12 %.	64
Figure 36: On-/off-analysis of crank propulsion for power output of 25 W at a threshold of 12 %.	65
Figure 37: On-/off-analysis of crank propulsion for power output of 35 W at a threshold of 12 %.	65
Figure 38: Elevation plane during crank propulsion (flexion +, extension -). The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device.	66
Figure 39: Shoulder elevation angle during crank propulsion (abduction +, adduction -). The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device.....	67
Figure 40: Shoulder rotation angle during crank propulsion (internal +, external -). The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device.....	67
Figure 41: Elbow flexion angle during crank propulsion. The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device.	68
Figure 42: Wrist deviation angle during crank propulsion (ulnar +, radial -). The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device.	68

Figure 43: Wrist flexion angle during crank propulsion. The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device. 69

Figure 44: Elevation plane during pushrim propulsion (flexion +, extension -). The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device. 70

Figure 45: Elevation angle during pushrim propulsion (abduction +, adduction -). The bold lines indicate the ergonomic joint region. Right sketch shows start (0%) and end (100%) of propulsion cycle of right-handed crank propulsion device. 71

Figure 46: Shoulder rotation angle during pushrim propulsion (internal +, external -). The bold lines indicate the ergonomic joint region. Sketches show the position of the right arm during one stroke... 71

Figure 47: Elbow flexion angle during pushrim propulsion. The bold lines indicate the ergonomic joint region. Sketches show the position of the right arm during one stroke..... 72

8. List of Tables

Table 1: Muscles involved in flexion, extension, external rotation, internal rotation, adduction and abduction of the shoulder joint.[9]	14
Table 2: Muscles involved in flexion, extension, pronation and supination movement of the elbow joint [9].	15
Table 3: Muscles involved in flexion, extension, radial deviation and ulnar deviation movement of the wrist joint.[9]	16
Table 4: Overview of software used for data recording, post-processing and analysis.....	27
Table 5: Profile of the wheelchair dependent subjects who participated in the study.....	30
Table 6: Profile of the non-wheelchair dependent subjects who participated in the study.	31
Table 7: Example of a REACTION file for two frames.....	44
Table 8: Example of interpolated table of test rig and Handle data for time frame 1 and 2.	45
Table 9: Example of ANALYSIS file of two frames.	45
Table 10: Example for MARKER file for two frames.	46
Table 11: Example of output file of Inverse Kinematics in OpenSim, containing joint angle data	53
Table 12: Mean mechanical efficiency of all subjects during crank and pushrim propulsion, calculated with formula 1 with the standard deviation and mean velocity respectively.	54
Table 13: Mean values and standard abbreviations of mechanical efficiency and velocity of pushrim and crank propulsion of non-wheelchair users.	58
Table 14: Absolute mean maximum angle values during crank propulsion at power output of 25 W and 35 W and values for comparison to literature and pushrim propulsion.....	70

9. Annexes

9.1 Annex 1: Matlab Codes

9.1.1 Handle Code

```
%% final code - crank propulsion
% read in TestRig (.xls) and Handle (.xls)
% interpolate + insert wanted columns in common file
% read motion file
% calculate crank angle, radius, angular velocity, tilt angle, global forces/torques
% get final files: ANALYSIS, MARKER, REACTION
% after selection of cycles, set 'cyclefiles' to YES to get files for OpenSim
% author: Anja Arbter

clear all
close all
clc

format long
format compact

% YES after overall start/end cycle file (Frames.xlsx) is created
cyclefiles = 'YES';
% frequency of motion capture
freq_marker = 100;

%% read in TestRig (.xlsx) and Handle (.xlsx) file
[fname1, pname1] = uigetfile('*.xlsx');
cd(pname1);
% Create fully-formed filename as a string
filename = fullfile(pname1, fname1);
% Check that file exists
assert(exist(filename,'file')==2, '%s does not exist.', filename);
% Read in the data
TestRig = xlsread(filename);

filename2 = [fname1(1:(end-12)) 'Handle'];
Handle = xlsread(filename2);

%% set correct time (in s)
size_Handle = size(Handle);
size_TestRig = size(TestRig);

% Handle file starts shortly before TestRig file
% set starting point 0ms (in TestRig) - in s
minus = abs(TestRig(1,1));
for r = 1:1:size_TestRig(1)
    TestRig(r,1) = (TestRig(r,1) - minus)/1000;
end

% convert in same ms format (in Handle) - in s
for p = 1:1:size_Handle(1)
    Handle(p,1) = (Handle(p,1) - minus)/1000;
end

% assure files have same length
if Handle(size_Handle(1),1)>TestRig(size_TestRig(1),1)
    Handle(size_Handle(1),:) = [];
    size_Handle = size(Handle);
elseif TestRig(size_TestRig(1),1)>Handle(size_Handle(1),1)
    TestRig(size_TestRig(1),:) = [];
    size_TestRig = size(TestRig);
end

disp ('common starting time set. ');
%% interpolation
steps = 1/freq_marker; % 1/100 = 0.01 [1/s]
% interpolation of Testrig file
x = TestRig(1:end-1,1);
y = TestRig(1:end-1,2:end);
xi = 0:steps:TestRig(size_TestRig(1),1);
yi = interp1(x,y,xi,'spline');

% interpolation of Handle file
```

```

v = Handle(1:end,1);
w = Handle(1:end,2:end);
vi = 0:steps:TestRig(size_TestRig(1),1);
wi = interp1(v,w,vi,'spline');

disp ('interpolation done. ');
%% insert wanted columns (!) in common file for TestRig and Handle
% TestRig: all columns
size_xi = size(xi);
frame_number = 1:size_xi(2);
interpolated_TR_H = zeros(size_xi(2),21);
interpolated_TR_H(1:end,1) = frame_number; % frame #
interpolated_TR_H(1:end,2) = xi; % time
interpolated_TR_H(1:end,3:15) = yi; % data

% Handle: fx fy fz Mx My Mz - still in Handle coordinates
fx = wi(1:end,1);
fy = wi(1:end,2);
fz = wi(1:end,3);
Mx = wi(1:end,5);
My = wi(1:end,6);
Mz = wi(1:end,7);
interpolated_TR_H(1:end,16) = fx;
interpolated_TR_H(1:end,17) = fy;
interpolated_TR_H(1:end,18) = fz;
interpolated_TR_H(1:end,19) = Mx;
interpolated_TR_H(1:end,20) = My;
interpolated_TR_H(1:end,21) = Mz;

disp ('interpolated_TR_H created. ');
%% open motion file (marker data)
% manually converted in Mokka from c3d to trc, saved as *.xlsx
filename3 = [fname1(1:(end-13)) '1'];
marker_file = xlsread(filename3);

% check if trc file and interpolated_TR_SW file have same length
% delete rest
size_marker = size(marker_file);
size_interpolated_TR_H = size(interpolated_TR_H);

if(size_marker(1) > size_interpolated_TR_H(1))
    marker_file((size_interpolated_TR_H(1)+1):end,:) = [];
elseif(size_interpolated_TR_H(1) > size_marker(1))
    interpolated_TR_H((size_marker(1)+1):end,:) = [];
else
end

size_marker_new = size(marker_file);

% convert all marker values from mm to m (/1000)
marker_file(:,3:end) = marker_file(:,3:end)/1000;

disp ('marker_file created. ');

%% calculate crank centre
% handle markers
HT_x = marker_file(:,33);
HT_y = marker_file(:,34);
HT_z = marker_file(:,35);
HH_x = marker_file(:,36);
HH_y = marker_file(:,37);
HB_x = marker_file(:,39);
HB_y = marker_file(:,40);
HB_z = marker_file(:,41);
HC_x = marker_file(:,42);
HC_y = marker_file(:,43);
HC_z = marker_file(:,44);

% finding first 3 non zero values in each column by index
% averaging them and saving as final coordinates as Crank_X,Y,Z
index_nonzero_HC_x = find(HC_x(:,1));
index_nonzero_HC_y = find(HC_y(:,1));
index_nonzero_HC_z = find(HC_z(:,1));

Centre_X = (sum(HC_x(index_nonzero_HC_x(1:3),1)))/3;

```

```

Centre_Y = (sum(HC_y(index_nonzero_HC_y(1:3),1)))/3;
Centre_Z = (sum(HC_z(index_nonzero_HC_z(1:3),1)))/3;

%% calculate centre of force (CoF)
% centre between handle top & handle bottom
% crank centre as origin, adapt HT, HB
HT_x_new = HT_x-Centre_X;
HT_y_new = HT_y-Centre_Y;
HT_z_new = HT_z-Centre_Z;
HB_x_new = HB_x-Centre_X;
HB_y_new = HB_y-Centre_Y;
HB_z_new = HB_z-Centre_Z;

CoF_x = (HB_x_new + HT_x_new)/2;
CoF_y = (HB_y_new + HT_y_new)/2;
CoF_z = (HB_z_new + HT_z_new)/2;

% centre of force for OpenSim (global COS)
CoF_x_OS = (HB_x + HT_x)/2;
CoF_y_OS = (HB_y + HT_y)/2;
CoF_z_OS = (HB_z + HT_z)/2;

%% calculate crank radius
% length of handle
radius = NaN(length(CoF_x),1);
for kl = 1:(length(CoF_x))
    x_dir = (CoF_x(kl,1)).^2;
    y_dir = (CoF_y(kl,1)).^2;
    radius(kl,1) = abs(sqrt(x_dir+y_dir));
end

%% calculate crank angle
vector_x = CoF_x;
vector_y = CoF_y;
angle_cr = atan2d(vector_y,vector_x);
% consider quadrants
crank_angle = NaN(length(angle_cr),1);
for mm = 1:length(angle_cr)
    t1 = CoF_x(mm,1);
    t2 = CoF_y(mm,1);
    % quadrants 3 and 4 (= 0°-180° of propulsion movement)
    if ((t1>0 || t1<0) && t2<0)
        crank_angle(mm,1) = abs(angle_cr(mm,1));
    % quadrants 1 and 2 (= 180-360° of propulsion movement)
    else if ((t1>0 || t1<0) && t2>0)
        crank_angle(mm,1) = 360-angle_cr(mm,1);
    % starting position of propulsion movement
    else if (t1>0 && t2==0)
        crank_angle(mm,1) = 0;
    % position at 180°
    else if (t1<0 && t2==0)
        crank_angle(mm,1) = 180;
    end
    end
end
end
end

% create file for markers and angle
table_marker_angle = [marker_file, crank_angle];
size_m_a = size(table_marker_angle);
disp ('table_marker_angle created. ');

% select beginning of cycles
frame_file = NaN(length(crank_angle),1);
fk = 1;
for ff = 2:length(crank_angle)
    if (crank_angle(ff-1,1) > crank_angle(ff,1))
        frame_file(fk,1) = ff;
        fk = fk+1;
    end
end
end
% delete rest
frame_file(fk:end,:)=[];
xlswrite(['Frames_' fname1(1:12)],frame_file);

```

```

%% calculate angular velocity
% omega = d(angle)/dt with dt=10ms if frequency of motion capture is 100Hz
angular_velocity = NaN(length(crank_angle),1);
angular_velocity(1,1) = 0;
crank_angle_rad = deg2rad(crank_angle);
ac = 2;
dt = 1/freq_marker; % [1/s]
for av = 1:(length(crank_angle_rad)-1)
    angular_velocity(ac,1) = (crank_angle_rad(ac,1)-crank_angle_rad(av,1))/dt;
    ac = ac+1;
end

%% calculate tilt angle of crank
% alternative, works for this measurement setup (but not >90°?)
% for cr = 1:size_m_a(1)
%     tilt_angle(cr,1) = -atan2((HT_x(cr,1)-HB_x(cr,1))/(HT_y(cr,1)-HB_y(cr,1)));
% end

t_angle = NaN(length(HT_x_new),1);
for ppp = 1:length(HT_x_new)
    % set CoF as origin, adapt HT marker coordinates
    HT_x_tilt = HT_x_new(ppp,1)-CoF_x(ppp,1);
    HT_y_tilt = HT_y_new(ppp,1)-CoF_y(ppp,1);
    % angle between vector from CoF to HT and positive x-axis
    t_angle(ppp,1) = atan2d(HT_y_tilt,HT_x_tilt);
end
% change to angle from positive y axis (flip by 90°) & consider quadrants
% tilt angle: negative clockwise & positive anti-clockwise
tilt_angle = NaN(length(t_angle),1);
for ta=1:length(t_angle)
    % -90° < angle < 90°
    if t_angle(ta,1) >0
        tilt_angle(ta,1) = t_angle(ta,1)-90;
    % angle at -90°
    elseif t_angle(ta,1)==0
        tilt_angle(ta,1) = -90;
    % angle at 90°
    elseif t_angle(ta,1)==180
        tilt_angle(ta,1) = 90;
    % -180° < angle < -90°
    elseif (t_angle(ta,1)<0 && t_angle(ta,1)>-90)
        tilt_angle(ta,1) = t_angle(ta,1)-90;
    % angle at -180° (or 180°)
    elseif t_angle(ta,1)==-90
        tilt_angle(ta,1) = -180;
    % 90° < angle < 180°
    elseif t_angle(ta,1)<-90
        tilt_angle(ta,1) = t_angle(ta,1)+270;
    else
        disp('error. ');
    end
end

%% correct coordinates of handle forces & torques
% change x y z of handle for correct coordination system: x=-y, y=x, z=z
fx_h = -(wi(1:end,2));
fy_h = wi(1:end,1);
fz_h = wi(1:end,3);
Mx_h = wi(1:end,6);
My_h = wi(1:end,5);
Mz_h = wi(1:end,7);

F_GCOS = NaN(length(tilt_angle),3);
M_GCOS = NaN(length(tilt_angle),3);
for fi=1:size(tilt_angle)
    % matrix for conversion (Drehmatrix)
    A_H = [cosd(tilt_angle(fi,1)), -
sind(tilt_angle(fi,1)), 0; sind(tilt_angle(fi,1)), cosd(tilt_angle(fi,1)), 0; 0, 0, 1];
    % force vector in handle coordinate system (HCOS)
    F_HCOS = [fx_h(fi,1); fy_h(fi,1); fz_h(fi,1)];
    % force vector in global coordinate system (GCOS)
    F_GCOS(fi,:) = A_H*F_HCOS;
    % torque in handle coordinate system (HCOS)
    M_HCOS = [Mx_h(fi,1); My_h(fi,1); Mz_h(fi,1)];
    % torque in global coordinate system (GCOS)

```

```

M_GCOS(fi,:) = A_H*M_HCOS;
end

%% Writing overall ANALYSIS file
% adapted from Nithin Kurup
% file: frame #, time, handle values of interpolated_TR_SW, correct force-
% and moment values, CoF, crank angle, crank radius, angular velocity

Overhead.data{1,1} = [fname1(1:(end-12)), '_ANALYSIS.txt']; %print out file
Odata = [interpolated_TR_H(:,1:15), F_GCOS, M_GCOS, CoF_x, CoF_y, CoF_z, crank_angle, radius,
angular velocity];
[ro, co] = size(Odata);
Overhead.data{2,1} = ['Marker freq : ', num2str(freq_marker), 'Hz'];
Overhead.data{3,1} = ['Crank_Center_XYZ (m) : ', ' ', num2str(Centre_X), ' ',
', num2str(Centre_Y), ' ', num2str(Centre_Z)]; % name of centre marker
Overhead.data{4,1} = 'Marker values in meters & Force in Newtons';

% insert headers
lvm_titles =
{'Frame#', 'Time', 'w[rad/s]', 'v[m/s]', 'dw[rad/s]', 'dv[m/s]', 'alpha[rad/s^2]', 'a[m/s^2]', 'dalpa
[rad/s^2]', 'da[m/s^2]', 'Position[deg]', 'RPM[1/min]', 'Handle RPM[1/min]', 'Handle
v[m/s]', 'Actual Power[W]', 'Fx [N]', 'Fy [N]', 'Fz [N]', 'Mx [Nm]', 'My [Nm]', 'Mz [Nm]', 'CoFx
[m]', 'CoFy [m]', 'CoFz [m]', 'Crank Angle [deg]', 'Crank Radius [m]', 'Angular Velocity [rad/s]'};

for bb = 1:co
    Overhead.colm {bb,1} = lvm_titles{1,bb};
end
F_Overhead = fopen([fname1(1:(end-12)), '_ANALYSIS.txt'], 'wt' );

[MO,NO]=size(Overhead.data);
for i=1:MO
    for j=1:NO
        fprintf(F_Overhead, '%s\n', Overhead.data{i,j});
    end
end
[PO,QO]=size(Overhead.colm);
for k=1:PO
    for m=1:QO
        fprintf(F_Overhead, '%s\t', Overhead.colm{k,m});
    end
end
fprintf(F_Overhead, '\n');
fclose (F_Overhead);

% writing the .txt handle force data file

dlmwrite([fname1(1:(end-12)), '_ANALYSIS.txt'], ...
Odata, '-append', 'delimiter', '\t', 'newline', 'pc');
disp ('Complete analysis file written successfully!');

%% writing MARKER file (.trc) for selected cycles/strokes
% adapted from Nithin Kurup
% frame#, time, all markers
% headers & save as .trc file
% Printing c3d to trc marker file for OpenSim

% initialise the header with a column for frame # and time
% initialise the format for the columns of data to be written to file
dataheader1 = 'Frame#\tTime\t';
dataheader2 = '\t\t';
format_text = '%i\t%2.4f\t';

Start_Frame = 1;
End_Frame = size_marker_new(1);
nframe = Start_Frame:End_Frame;
% headers + right format
dynamic_name_markers =
{'BB1', 'CL1', 'SH1', 'BI1', 'EL1', 'EL2', 'FA1', 'WR1', 'WU1', 'FI1', 'HT1', 'HH1', 'HB1', 'HC1', 'CoF'};
dynamic_numof_markers = size(dynamic_name_markers);

for i = 1:dynamic_numof_markers(2)
    dataheader1 = [dataheader1 dynamic_name_markers{i} '\t\t\t'];
    dataheader2 = [dataheader2 'X' num2str(i) '\t' 'Y' num2str(i) '\t'...
'Z' num2str(i) '\t'];
    format_text = [format_text '%f\t%f\t%f\t'];
end

```

```

end

dataheader1 = [dataheader1 '\n'];
dataheader2 = [dataheader2 '\n'];
format_text = [format_text '\n'];

dit2 = [marker_file CoF_x CoF_y CoF_z];

% write output marker data during all time
f_TRC = fopen([fname1(1:(end-12)),'_MARKER.trc'],'wt ');
% write header data
fprintf(f_TRC,'PathFileType\t4\t(X/Y/Z)\t %s\n',fname1(1:(end-12)));
fprintf(f_TRC,'DataRate\tCameraRate\tNumFrames\tNumMarkers\tUnits\tOrigDataRate\tOrigDataStart
Frame\tOrigNumFrames\n');
freq_marker = 100;
fprintf(f_TRC,'%d\t%d\t%d\t%d\t%s\t%d\t%d\t%d\n', (freq_marker), (freq_marker),
length(nframe'),dynamic_numof_markers(2), 'm',(freq_marker),Start_Frame,End_Frame);
fprintf(f_TRC, dataheader1);
fprintf(f_TRC, dataheader2);
dlmwrite([fname1(1:(end-12)),'_MARKER.trc'],...
    dit2,'-append','delimiter','\t','newline','pc');
disp('Overall Marker TRC file written successfully!');
% close file
fclose(f_TRC);
%% REACTION file
% adapted from Nithin Kurup
% time, F_reaction, CoF in GCOS, Moment_reaction
% negate forces+torques

time = interpolated_TR_H(1:end,2);
Odata1 = [time, F_GCOS, CoF_x_OS, CoF_y_OS, CoF_z_OS, M_GCOS];

% inverse handle forces and torques
Odata_H = [Odata1(Start_Frame:End_Frame,1), -
(Odata1(Start_Frame:End_Frame,2:4)),Odata1(Start_Frame:End_Frame,5:7),-
(Odata1(Start_Frame:End_Frame,8:10))];

%headers
force.data{1,1} = ['name ',fname1(1:(end-12)),'_REACTION.mot'];
[rRF, cRF] = size(Odata_H);
force.data{2,1} = ['datacolumns ', num2str(cRF)];
force.data{3,1} = ['datarows ', num2str(rRF)];
force.data{4,1} = ['range ', num2str(Odata1(Start_Frame,1)), ' ', num2str(Odata1(End_Frame,1))];
force.data{5,1} = 'endheader ';

force.colm{1,1} = 'time'; force.colm{2,1} = 'r_ground_force_vx'; force.colm{3,1} =
'r_ground_force_vy';...
force.colm{4,1} = 'r_ground_force_vz'; force.colm{5,1} = 'r_ground_force_px';
force.colm{6,1} = 'r_ground_force_py';
force.colm{7,1} = 'r_ground_force_pz';
force.colm{8,1} = 'r_ground_torque_x'; force.colm{9,1} = 'r_ground_torque_y';force.colm{10,1}
= 'r_ground_torque_z';

f_RF = fopen([fname1(1:(end-12)),'_REACTION.mot'],'wt');
[M,N]=size(force.data);
for i=1:M
    for j=1:N
        fprintf(f_RF,'%s\n',force.data{i,j});
    end
end
[P,Q]=size(force.colm);
for k=1:P
    for m=1:Q
        fprintf(f_RF,'%s\t',force.colm{k,m});
    end
end
fprintf(f_RF,'\n');
fclose (f_RF);

% writing the .mot handle force data file
dlmwrite([fname1(1:(end-12)),'_REACTION.mot'],...
    Odata_H,'-append','delimiter','\t','newline','pc');
disp('Overall Reaction file written successfully!');

%% OpenSim files

```

```

if strcmp(cyclefiles,'YES')

    frame_file = xlsread('Frames.xlsx');
    if filename2((end-10):end) == '25W_Handle'
        column = 3+4*(str2num(filename2(4))-1);
    elseif filename2((end-10):end) == '35W_Handle'
        column = 4+4*(str2num(filename2(4))-1);
    else
        disp('column not found');
    end

    starting_frame = frame_file(column,1);
    ending_frame = frame_file(column,6);

    % A: Marker file for OpenSim (TRC file)
    % initialise the header with a column for frame # and time
    % initialise the format for the columns of data to be written to file
    dataheader1 = 'Frame#\tTime\t';
    dataheader2 = '\t\t';
    format_text = '%i\t%2.4f\t';

    % selected frame number
    nframe = starting_frame:ending_frame;
    dit1 = marker_file(starting_frame:ending_frame,:);

    % headers + right format
    dynamic_name_markers =
    {'BB1','CL1','SH1','BI1','EL1','EL2','FA1','WR1','WU1','FI1','HT1','HH1','HB1','HC1'};
    dynamic_numof_markers = size(dynamic_name_markers);

    for i = 1:dynamic_numof_markers(2)
        dataheader1 = [dataheader1 dynamic_name_markers{i} '\t\t\t'];
        dataheader2 = [dataheader2 'X' num2str(i) '\t' 'Y' num2str(i) '\t'...
            'Z' num2str(i) '\t'];
        format_text = [format_text '%f\t%f\t%f\t'];
    end

    dataheader1 = [dataheader1 '\n'];
    dataheader2 = [dataheader2 '\n'];
    format_text = [format_text '\n'];

    f_TRC = fopen([fname1(1:(end-12)),'_MARKER_OS.trc'],'wt ');
    % write header data
    fprintf(f_TRC,'PathFileType\t4\t(X/Y/Z)\t %s\n',fname1(1:(end-12)));

    fprintf(f_TRC,'DataRate\tCameraRate\tNumFrames\tNumMarkers\tUnits\tOrigDataRate\tOrigDataStart
    Frame\tOrigNumFrames\n');
    freq_marker = 100;
    fprintf(f_TRC,'%d\t%d\t%d\t%d\t%d\t%s\t%d\t%d\t%d\n', (freq_marker), (freq_marker),
    length(nframe'),dynamic_numof_markers(2), 'm',(freq_marker),starting_frame,ending_frame);
    fprintf(f_TRC, dataheader1);
    fprintf(f_TRC, dataheader2);

    % write output marker data for 3 cycles (for OpenSim)
    dlmwrite([fname1(1:(end-12)),'_MARKER_OS.trc'],...
        dit1,'-append','delimiter','\t','newline','pc');
    disp('Marker TRC file for OpenSim written successfully !!!');
    % close file
    fclose(f_TRC);

    % B: Reaction File for OpenSim
    time = interpolated_TR_H(1:end,2);
    Odata2 = [time, F_GCOS, CoF_x_OS, CoF_y_OS, CoF_z_OS, M_GCOS];

    % inverse handle forces and torques
    Odata_H2 = [Odata2(starting_frame:ending_frame,1), -
    (Odata2(starting_frame:ending_frame,2:4)),Odata2(starting_frame:ending_frame,5:7),-
    (Odata2(starting_frame:ending_frame,8:10))];

    %headers
    force.data{1,1} = ['name ',fname1(1:(end-12)),'_REACTION_OS.mot'];
    [rRF, cRF] = size(Odata_H2);
    force.data{2,1} = ['datacolumns ', num2str(cRF)];
    force.data{3,1} = ['datarows ', num2str(rRF)];

```

```

    force.data{4,1} = ['range ', num2str(Odata2(starting_frame,1)), '
', num2str(Odata2(ending_frame,1))];
    force.data{5,1} = 'endheader ';

    force.colm{1,1} = 'time'; force.colm{2,1} = 'r_ground_force_vx'; force.colm{3,1} =
'r_ground_force_vy';...
    force.colm{4,1} = 'r_ground_force_vz'; force.colm{5,1} = 'r_ground_force_px';
force.colm{6,1} = 'r_ground_force_py';
    force.colm{7,1} = 'r_ground_force_pz';
    force.colm{8,1} = 'r_ground_torque_x'; force.colm{9,1} =
'r_ground_torque_y'; force.colm{10,1} = 'r_ground_torque_z';

    f_RF2 = fopen([fname1(1:(end-12)), '_REACTION_OS.mot'], 'wt');
    [M,N]=size(force.data);
    for i=1:M
        for j=1:N
            fprintf(f_RF2, '%s\n', force.data{i,j});
        end
    end
    [P,Q]=size(force.colm);
    for k=1:P
        for m=1:Q
            fprintf(f_RF2, '%s\t', force.colm{k,m});
        end
    end
    fprintf(f_RF2, '\n');
    fclose (f_RF2);

    % writing the .mot handle force data file
    dlmwrite([fname1(1:(end-12)), '_REACTION_OS.mot'],...
    Odata_H2, '-append', 'delimiter', '\t', 'newline', 'pc');
    disp('Reaction file for OpenSim written successfully!');
end

% code END

```

9.1.2 SmartWheel Code

```

%% Smartwheel
% read in TestRig (.xls) and SmartWheel (.csv)
% synchronise times of TestRig and SW files - find common starting point
% interpolate + insert wanted columns in new common file
% read motion file
% calculate angle
% get final files: ANALYSIS, MARKER, REACTION
% after selection of cycles, set 'cyclefiles' to YES to get files for OpenSim
% author: Anja Arbter

clear all
close all
clc

format long
format compact

% YES after overall start/end cycle file (Frames.xlsx) is created
cyclefiles = 'YES';
% frequency of motion capture
freq_marker = 100;

%% read in TestRig (.xls) and Smartwheel (.csv) file
[fname2, pname2] = uigetfile('*.csv');
cd(pname2);
% Create fully-formed filename as a string
filename = fullfile(pname2, fname2);
% Check that file exists
assert(exist(filename, 'file')==2, '%s does not exist.', filename);
% Read in the data, skipping the first row
SmartWheel = csvread(filename);

filename2 = [fname2(1:(end-4)) '_TestRig'];
TestRig = xlsread(filename2);

```



```

%% synchronise times, find common starting point
% 1.1.1904 to 1.1.1970 = 66 years = -2082844800000 ms (incl. leap years)
TestRig_timesynchronised = TestRig(:,1) - 2082844800000;

TestRig(:,1) = TestRig_timesynchronised; % overwrite 1st column in TestRig

size_SmartWheel = size(SmartWheel);
size_TestRig = size(TestRig);

% find common starting time
% SW starts before TR
start_diff_zuvor = 100000000; %definition difference
end_diff_zuvor = 100000000;
for t = 1:1:size_SmartWheel(1)
    start_diff = TestRig_timesynchronised(1,1) - SmartWheel(t,1);
    end_diff = TestRig_timesynchronised(end,1) - SmartWheel(t,1);
    % starting index
    if abs(start_diff) < abs(start_diff_zuvor)
        index_start_diff = t;
        start_min_diff = start_diff;
    end
    % ending index
    if abs(end_diff) < abs(end_diff_zuvor)
        index_end_diff = t;
        end_min_diff = end_diff;
    end

    start_diff_zuvor = start_diff;
    end_diff_zuvor = end_diff;
end

% set starting point 0ms (in TestRig) - in s
starting_ms_TestRig = TestRig(1,1);
for p = 1:1:size_TestRig(1)
    TestRig(p,1) = (TestRig(p,1) - starting_ms_TestRig)/1000;
end

% convert in same ms format (in SmartWheel) - in s
for r = index_start_diff:1:index_end_diff
    SmartWheel(r,1) = (SmartWheel(r,1) - starting_ms_TestRig)/1000;
end
% %for offset analysis only
% for r = index_start_diff:1:size_SmartWheel(1)
%     SmartWheel(r,1) = (SmartWheel(r,1) - starting_ms_TestRig)/1000;
% end

% delete unused columnes in SmartWheel file
SmartWheel(index_end_diff:end,:) = []; %hide for offset analysis only
SmartWheel(1:index_start_diff,:) = [];

size_SmartWheel_new = size(SmartWheel);

% % for offset analysis only
% xlswrite('SW_P5_35W_offset',SmartWheel);

% assure files have same length
if SmartWheel(size_SmartWheel_new(1),1)>TestRig(size_TestRig(1),1)
    SmartWheel(size_SmartWheel_new,:) = [];
    size_SmartWheel_new = size(SmartWheel);
elseif TestRig(size_TestRig(1),1)>SmartWheel(size_SmartWheel_new(1),1)
    TestRig(size_TestRig(1),:) = [];
    size_TestRig = size(TestRig);
end

disp ('common starting time set. ');
%% interpolation
% interpolation of Testrig file
x = TestRig(1:end-1,1);
y = TestRig(1:end-1,2:end);
xi = 0:.01:TestRig(size_TestRig(1),1);
yi = interp1(x,y,xi,'spline');

% interpolation of Smartwheel file
v = SmartWheel(1:end,1);
w = SmartWheel(1:end,2:end);

```

```

vi = 0:.01:TestRig(size_TestRig(1),1);
wi = interp1(v,w,vi,'spline');

disp ('interpolation done. ');
%% insert wanted columns (!) in common file for TestRig and Smartwheel
% TestRig: all columns
size_xi = size(xi);
frame_number = 1:size_xi(2);
interpolated_TR_SW=zeros(size_xi(2),21);
interpolated_TR_SW(1:end,1) = frame_number; % frame #
interpolated_TR_SW(1:end,2) = xi; % time
interpolated_TR_SW(1:end,3:15) = yi; % data

% Smartwheel: fx fy fz Mx My Mz
fx = wi(1:end,18);
fy = wi(1:end,19);
fz = wi(1:end,20);
Mx = wi(1:end,21);
My = wi(1:end,22);
Mz = wi(1:end,23);
interpolated_TR_SW(1:end,16) = fx;
interpolated_TR_SW(1:end,17) = fy;
interpolated_TR_SW(1:end,18) = fz;
interpolated_TR_SW(1:end,19) = Mx;
interpolated_TR_SW(1:end,20) = My;
interpolated_TR_SW(1:end,21) = Mz;

size_interpolated_TR_SW = size(interpolated_TR_SW);

disp ('interpolated_SW created. ');

%% selecting stroke frames
% stroke starts when Mz exceeds 2 Nm
frame_file = NaN(length(Mz),1);
kk = 1;
for dd=1:size_interpolated_TR_SW(1)
    if(interpolated_TR_SW(dd,21)>2) && (interpolated_TR_SW(dd-1,21)<2)
        frame_file(kk,1) = interpolated_TR_SW(dd,1);
        kk = kk+1;
    end
end
% delete rest
frame_file(kk:end,:)=[];
% % hidden as this file would be empty for our input data
% xlswrite(['Frames_' fname2(1:(end-4))],frame_file);

%% open motion file (marker data)
% manually converted in Mokka from c3d to trc, saved as *.xlsx
filename3 = [fname2(1:(end-4)) '1'];
marker_file = xlsread(filename3);

% check if trc file and interpolated_TR_SW file have same length
% delete rest
size_marker = size(marker_file);
size_interpolated_TR_SW = size(interpolated_TR_SW);

if(size_marker(1) > size_interpolated_TR_SW(1))
    marker_file((size_interpolated_TR_SW(1)+1):end,:) = [];
elseif(size_interpolated_TR_SW(1) > size_marker(1))
    interpolated_TR_SW((size_marker(1)+1):end,:) = [];
else
end

size_marker_new = size(marker_file);

% convert all marker values from mm to m (/1000)
marker_file(:,3:end) = marker_file(:,3:end)/1000;

disp ('marker_file created. ');
%% angle calculation & table_marker_angle
% frame#, time, all markers, angle
% needed markers: centre marker SW (SC) + radial marker of SW (SR)
SC_x = marker_file(:,33);
SC_y = marker_file(:,34);
SC_z = marker_file(:,35);

```

```

SR_x = marker_file(:,39);
SR_y = marker_file(:,40);
SR_z = marker_file(:,41);
FI_x = marker_file(:,30);
FI_y = marker_file(:,31);
FI_z = marker_file(:,32);

% finding first 3 non zero values in each column by index
% averaging them and saving as final coordinates as Crank_X,Y,Z
index_nonzero_SC_x = find(SC_x(:,1));
index_nonzero_SC_y = find(SC_y(:,1));
index_nonzero_SC_z = find(SC_z(:,1));

Centre_X = (sum(SC_x(index_nonzero_SC_x(1:3),1)))/3;
Centre_Y = (sum(SC_y(index_nonzero_SC_y(1:3),1)))/3;
Centre_Z = (sum(SC_z(index_nonzero_SC_z(1:3),1)))/3;

%% angle values using centre of SW (SC) and radial marker of SW (SR)
% centre as origin, adapt SR
SR_x_new = SR_x-Centre_X;
SR_y_new = SR_y-Centre_Y;
angle_sw = atan2d(SR_y_new,SR_x_new);
%consider quadrants
angle_values = NaN(length(angle_sw),1);
for ss = 1:length(angle_sw)
    t1 = SR_x_new(ss,1);
    t2 = SR_y_new(ss,1);
    if ((t1>0 || t1<0) && t2<0)
        angle_values(ss,1) = abs(angle_sw(ss,1));
    elseif ((t1>0 || t1<0) && t2>0)
        angle_values(ss,1) = 360-angle_sw(ss,1);
    elseif (t1>0 && t2==0)
        angle_values(ss,1) = 0;
    elseif (t1<0 && t2==0)
        angle_values(ss,1) = 180;
    end
end

% create file for markers and angle
table_marker_angle = [marker_file, angle_values];

disp ('table_marker_angle created.');
```

```

%% Writing overall ANALYSIS file
% adapted from Nithin Kurup
% file: frame #, marker time, handle values, hand marker, centre, distance.
% interpolated_TR_SW, SC_x, SC_y, SC_z, SR_x, SR_y, SR_z, table_marker_angle(:,42)

Overhead.data{1,1} = [fname2(1:(end-4)), '_ANALYSIS.mot']; %print out file
Odata = [interpolated_TR_SW, SR_x, SR_y, SR_z, table_marker_angle(:,42)];
[ro, co] = size(Odata);
Overhead.data{2,1} = ['Marker freq : ', num2str(freq_marker), 'Hz'];
Overhead.data{3,1} = ['Smartwheel_Center_XYZ (m) : ', ' ', num2str(Centre_X), ' ',
    ' ', num2str(Centre_Y), ' ', ' ', num2str(Centre_Z)];
Overhead.data{4,1} = 'Marker values in meters & Force in Newtons';

% insert headers
lvm_titles =
{'Frame#', 'Time', 'w[rad/s]', 'v[m/s]', 'dw[rad/s]', 'dv[m/s]', 'alpha[rad/s^2]', 'a[m/s^2]', 'dalpha
[rad/s^2]', 'da[m/s^2]', 'Position[deg]', 'RPM[1/min]', 'Handle RPM[1/min]', 'Handle
v[m/s]', 'Actual Power[W]', 'Fx [N]', 'Fy [N]', 'Fz [N]', 'Mx [Nm]', 'My [Nm]', 'Mz [Nm]', 'SRx
[m]', 'SRy [m]', 'SRz [m]', 'angle [deg]'};

for bb = 1:co
    Overhead.colm {bb,1} =lvm_titles{1,bb};
end
F_Overhead = fopen([fname2(1:(end-4)), '_ANALYSIS.mot'], 'wt' );

[MO,NO]=size(Overhead.data);
for i=1:MO
    for j=1:NO
        fprintf(F_Overhead, '%s\n', Overhead.data{i,j});
    end
end
[PO,QO]=size(Overhead.colm);
```

```

for k=1:PO
    for m=1:QO
        fprintf(F_Overhead,'%s\t',Overhead.colm(k,m));
    end
end
fprintf(F_Overhead,'\n');
fclose (F_Overhead);

% writing the .mot handle force data file
dlmwrite([fname2(1:(end-4)),'_ANALYSIS.mot'],...
    Odata,'-append','delimiter','\t','newline','pc');
disp ('Complete analysis file written successfully !!');

%% writing MARKER file (.trc) for selected cycles/strokes
% adapted from Nithin Kurup
% frame#, time, all markers
% headers & save as .trc file
% Printing c3d to trc marker file for OpenSim

% initialise the header with a column for the Frame # and the Time
% initialise the format for the columns of data to be written to file
dataheader1 = 'Frame#\tTime\t';
dataheader2 = '\t\t';
format_text = '%i\t%2.4f\t';

Start_Frame = 1;
End_Frame = size_marker_new(1);
nframe = Start_Frame:End_Frame;
% headers + right format
dynamic_name_markers =
{'BB1','CL1','SH1','BI1','EL1','EL2','FA1','WR1','WU1','FI1','SC1','SM1','SR1'}; %adapt
dynamic_numof_markers = size(dynamic_name_markers);

for i = 1:dynamic_numof_markers(2)
    dataheader1 = [dataheader1 dynamic_name_markers{i} '\t\t\t'];
    dataheader2 = [dataheader2 'X' num2str(i) '\t' 'Y' num2str(i) '\t'...
        'Z' num2str(i) '\t'];
    format_text = [format_text '%f\t%f\t%f\t'];
end

dataheader1 = [dataheader1 '\n'];
dataheader2 = [dataheader2 '\n'];
format_text = [format_text '\n'];

dit = [marker_file(Start_Frame:End_Frame,:)];

% write output marker data during all time
f_TRC = fopen([fname2(1:(end-4)),'_MARKER.trc'],'wt' );
% write header data
fprintf(f_TRC,'PathFileType\t4\t(X/Y/Z)\t %s\n',fname2(1:(end-4)));
fprintf(f_TRC,'DataRate\tCameraRate\tNumFrames\tNumMarkers\tUnits\tOrigDataRate\tOrigDataStart
Frame\tOrigNumFrames\n');
freq_marker = 100;
fprintf(f_TRC,'%d\t%d\t%d\t%d\t%d\t%s\t%d\t%d\t%d\n', (freq_marker), (freq_marker),
length(nframe'),dynamic_numof_markers(2), 'm', (freq_marker),Start_Frame,End_Frame);
fprintf(f_TRC, dataheader1);
fprintf(f_TRC, dataheader2);

% write output marker data
dlmwrite([fname2(1:(end-4)),'_MARKER.trc'],...
    dit,'-append','delimiter','\t','newline','pc'); % change here also mot to txt
disp ('Overall Marker TRC file written successfully !!');
% close file
fclose(f_TRC);

%% REACTION file
% adapted from Nithin Kurup
% time, F_reaction, Handle_marker (FI/SR), Moment_reaction
% Smartwheel: negate frame (reaction forces)

Odata = [interpolated_TR_SW, FI_x, FI_y, FI_z];

% negate forces and torques

```

```

Odata_H = [Odata(Start_Frame:End_Frame,2), -
(Odata(Start_Frame:End_Frame,16:18)),Odata(Start_Frame:End_Frame,22:24),-
(Odata(Start_Frame:End_Frame,19:21))];

% headers
force.data{1,1} = ['name ',fname2(1:(end-4)), '_REACTION.mot'];
[rRF, cRF] = size(Odata_H);
force.data{2,1} = ['datacolumns ', num2str(cRF)];
force.data{3,1} = ['datarows ', num2str(rRF)];
force.data{4,1} = ['range ', num2str(Odata(Start_Frame,2)), ' ', num2str(Odata(End_Frame,2))];
force.data{5,1} = 'endheader ';

force.colm{1,1} = 'time'; force.colm{2,1} = 'r_ground_force_vx'; force.colm{3,1} =
'r_ground_force_vy';...
force.colm{4,1} = 'r_ground_force_vz'; force.colm{5,1} = 'r_ground_force_px';
force.colm{6,1} = 'r_ground_force_py';
force.colm{7,1} = 'r_ground_force_pz';
force.colm{8,1} = 'r_ground_torque_x'; force.colm{9,1} = 'r_ground_torque_y'; force.colm{10,1}
= 'r_ground_torque_z';

f_RF = fopen([fname2(1:(end-4)), '_REACTION.mot'],'wt');
[M,N]=size(force.data);
for i=1:M
    for j=1:N
        fprintf(f_RF, '%s\n', force.data{i,j});
    end
end
[P,Q]=size(force.colm);
for k=1:P
    for m=1:Q
        fprintf(f_RF, '%s\t', force.colm{k,m});
    end
end
fprintf(f_RF, '\n');
fclose (f_RF);

% writing the .mot handle force data file
dlmwrite([fname2(1:(end-4)), '_REACTION.mot'],...
Odata_H, '-append', 'delimiter', '\t', 'newline', 'pc');
disp ('Overall Reaction force file written successfully!');

%% OpenSim files
if strcmp(cyclefiles, 'YES')

    frame_file = xlsread('Frames.xlsx');
    if filename2(10:12) == '25W'
        column = 1+4*(str2num(filename2(5))-1);
    elseif filename2(10:12) == '35W'
        column = 2+4*(str2num(filename2(5))-1);
    else
        disp('column not found');
    end

    starting_frame = frame_file(column,1);
    ending_frame = frame_file(column,6);

    % A: Marker file for OpenSim (TRC file)
    % selected frame number
    nframe = starting_frame:ending_frame;
    dit1 = marker_file(starting_frame:ending_frame,:);

    f_TRC = fopen([fname2(1:(end-4)), '_MARKER_OS.trc'],'wt' );
    % write header data
    fprintf(f_TRC, 'PathFileType\t4\t(X/Y/Z)\t %s\n', fname2(1:(end-4)));

    fprintf(f_TRC, 'DataRate\tCameraRate\tNumFrames\tNumMarkers\tUnits\tOrigDataRate\tOrigDataStart
Frame\tOrigNumFrames\n');
    freq_marker = 100;
    fprintf(f_TRC, '%d\t%d\t%d\t%d\t%s\t%d\t%d\t%d\n', (freq_marker), (freq_marker),
length(nframe'), dynamic_numof_markers(2), 'm', (freq_marker), starting_frame, ending_frame);
    fprintf(f_TRC, dataheader1);
    fprintf(f_TRC, dataheader2);

    % write output marker data
    dlmwrite([fname2(1:(end-4)), '_MARKER_OS.trc'],...

```

```

        ditl, '-append', 'delimiter', '\t', 'newline', 'pc');
disp ('Marker TRC file for OpenSim written successfully !!');
% close file
fclose(f_TRC);

% B: Reaction File for OpenSim
Odata2 = [interpolated_TR_SW, FI_x, FI_y, FI_z];

% inverse handle forces and torques
Odata_H2 = [Odata2(starting_frame:ending_frame,2), -
(Odata2(starting_frame:ending_frame,16:18)),Odata2(starting_frame:ending_frame,22:24),-
(Odata2(starting_frame:ending_frame,19:21))];

%headers
force.data{1,1} = ['name ',fname2(1:(end-4)), '_REACTION_OS.mot'];
[rRF, cRF] = size(Odata_H2);
force.data{2,1} = ['datacolumns ', num2str(cRF)];
force.data{3,1} = ['datarows ', num2str(rRF)];
force.data{4,1} = ['range ', num2str(Odata2(starting_frame,1)), ',
', num2str(Odata2(ending_frame,1))];
force.data{5,1} = 'endheader ';

force.colm{1,1} = 'time'; force.colm{2,1} = 'r_ground_force_vx'; force.colm{3,1} =
'r_ground_force_vy';...
force.colm{4,1} = 'r_ground_force_vz'; force.colm{5,1} = 'r_ground_force_px';
force.colm{6,1} = 'r_ground_force_py';
force.colm{7,1} = 'r_ground_force_pz';
force.colm{8,1} = 'r_ground_torque_x'; force.colm{9,1} =
'r_ground_torque_y'; force.colm{10,1} = 'r_ground_torque_z';

f_RF2 = fopen([fname2(1:(end-4)), '_REACTION_OS.mot'], 'wt');
[M,N]=size(force.data);
for i=1:M
    for j=1:N
        fprintf(f_RF2, '%s\n', force.data{i,j});
    end
end
[P,Q]=size(force.colm);
for k=1:P
    for m=1:Q
        fprintf(f_RF2, '%s\t', force.colm{k,m});
    end
end
fprintf(f_RF2, '\n');
fclose (f_RF2);

% writing the .mot handle force data file
dlmwrite([fname2(1:(end-4)), '_REACTION_OS.mot'],...
Odata_H2, '-append', 'delimiter', '\t', 'newline', 'pc');
disp('Reaction file for OpenSim written successfully!');
end

% code END

```

9.1.3 EMG code

```

%% EMG post-processing and analysis
% filtering of signal
% MVC normalization
% author: Anja Arbter

clear all
close all
clc

format long
format compact

% adapt for each subject and measurement
five_cycles = 'NO'; % 'NO' for SW, 'YES' for HBP

% may be adapted

```

```

framesize = 100; % RMS framesize (here: 50ms = 100 frames)
number_MVC = 4; % number of muscles that performed MVC test
MVC_1 = '_isom_D.xlsx';
MVC_2 = '_isom_BI.xlsx';
MVC_3 = '_isom_T.xlsx';
MVC_4 = '_isom_T.xlsx';
x_max = 100; % maximum value of x-axis
num_maxval = 10; % number of maximum values for mean calculation
%% read in main file
[fname1, pname1] = uigetfile('*.xlsx');
cd(pname1);
%Create fully-formed filename as a string
filename = fullfile(pname1, fname1);
%Check that file exists
assert(exist(filename,'file')==2, '%s does not exist.', filename);
%Read in the data, skipping the first row
EMG_file = xlsread(filename);

%% get names and frames for analysis files
subject = fname1(1:2);
analysis_file = ['analysis_' subject '_' fname1((end-10):(end-5))]; % filtered, normalized &
interpolated EMG file
cycle_file = ['mean_cycle_' subject '_' fname1((end-10):(end-5))]; % mean cycle file

frame_file = xlsread('Frames.xlsx');
if fname1((end-10):(end-5)) == 'SW_25W'
    column = 1+4*(str2num(subject(2))-1);
elseif fname1((end-10):(end-5)) == 'SW_35W'
    column = 2+4*(str2num(subject(2))-1);
elseif fname1((end-10):(end-5)) == 'H_25W'
    column = 3+4*(str2num(subject(2))-1);
elseif fname1((end-10):(end-5)) == 'H_35W'
    column = 4+4*(str2num(subject(2))-1);
else
    disp('column not found');
end
starting_frame1 = frame_file(column,1); % starting frames of 3 or 5 cycles
starting_frame2 = frame_file(column,2);
starting_frame3 = frame_file(column,3);
ending_frame = frame_file(column,6);
if strcmp(five_cycles,'YES')
    starting_frame4 = frame_file(column,4); % for HBP
    starting_frame5 = frame_file(column,5); % for HBP
end

% fixed parameters
file_length = ending_frame-starting_frame1+1;
cycle1_length = starting_frame2-starting_frame1;
cycle2_length = starting_frame3-starting_frame2;
if strcmp(five_cycles,'NO') % for SW
    cycle3_length = ending_frame-starting_frame3+1;
end
if strcmp(five_cycles,'YES') % for HBP
    cycle3_length = starting_frame4-starting_frame3;
    cycle4_length = starting_frame5-starting_frame4;
    cycle5_length = ending_frame-starting_frame5+1;
end
%% get columns of time & EMG sensors (DA, DM, DP, P, BI, T La, T Lo)
EMG_all = [EMG_file(:,1) EMG_file(:,2) EMG_file(:,10) EMG_file(:,18) EMG_file(:,26)
EMG_file(:,34) EMG_file(:,42) EMG_file(:,50)];
size_EMG_new = size(EMG_all);

%% filter raw EMG signal
% square & square root
for count = 2:1:8
    EMG_new(:,count) = sqrt(EMG_all(:,count).^2);
end
EMG_new(:,1) = EMG_all(:,1);
% root-mean-square(RMS)
div_frame = floor(size_EMG_new(1)/framesize); % how many times framesize fits into file
length
RMS_length = framesize*div_frame; % new file length

EMG_filtered = NaN(div_frame+1,size_EMG_new(2));
EMG_filtered(1,:) = 0; % start: 0 ms = 0 V

```

```

ee = 2; % count for new file length of RMS
for c3 = 2:framesize:(RMS_length)
    c4=c3+(framesize-1);
    for sensors = 2:8
        vector2 = EMG_new(c3:c4,sensors);
        rms_x2 = sqrt(mean(vector2.^2));
        EMG_filtered(ee,sensors) = rms_x2;
    end
    EMG_filtered(ee,1) = EMG_new((ee-1)*100,1); % time
    ee = ee+1;
end
size_EMG_filtered = size(EMG_filtered);

%% MVC file
EMG_MVC_DM = xlsread([subject MVC_1]);
EMG_MVC_BI = xlsread([subject MVC_2]);
EMG_MVC_T = xlsread([subject MVC_3]);

% get time and data for DM, BI, TLa, TLo muscles
MVC_DM = [EMG_MVC_DM(:,1) EMG_MVC_DM(:,10)];
MVC_BI = [EMG_MVC_BI(:,1) EMG_MVC_BI(:,34)];
MVC_TLa = [EMG_MVC_T(:,1) EMG_MVC_T(:,42)];
MVC_TLo = [EMG_MVC_T(:,1) EMG_MVC_T(:,50)];

% filter signal
%% square & root square
MVC_DM_new = sqrt(MVC_DM(:,2).^2);
MVC_BI_new = sqrt(MVC_BI(:,2).^2);
MVC_TLa_new = sqrt(MVC_TLa(:,2).^2);
MVC_TLo_new = sqrt(MVC_TLo(:,2).^2);

%% root-mean-square (RMS)
div_frame_DM = floor((length(MVC_DM_new))/framesize);
div_frame_BI = floor((length(MVC_BI_new))/framesize);
div_frame_T = floor((length(MVC_TLa_new))/framesize);
RMS_length_DM = framesize*div_frame_DM;
RMS_length_BI = framesize*div_frame_BI;
RMS_length_T = framesize*div_frame_T;

DM_filtered = NaN(div_frame_DM+1,2); % +1 because of starting time
BI_filtered = NaN(div_frame_BI+1,2);
T_filtered = NaN(div_frame_T+1,3);
DM_filtered(1,:) = 0;
BI_filtered(1,:) = 0;
T_filtered(1,:) = 0;
% RMS for DM
cd = 2; % count for new file length of RMS
for c5 = 2:framesize:(RMS_length_DM)
    c6=c5+(framesize-1);
    vector3 = MVC_DM_new(c5:c6,1);
    rms_x3 = sqrt(mean(vector3.^2));
    DM_filtered(cd,2) = rms_x3;
    DM_filtered(cd,1) = MVC_DM((cd-1)*100,1);
    cd = cd+1;
end
% RMS for BI
cb = 2;
for c7 = 2:framesize:(RMS_length_BI)
    c8=c7+(framesize-1);
    vector4 = MVC_BI_new(c7:c8,1);
    rms_x4 = sqrt(mean(vector4.^2));
    BI_filtered(cb,2) = rms_x4;
    BI_filtered(cb,1) = MVC_BI((cb-1)*100,1);
    cb = cb+1;
end
% RMS for TLa & TLo
ct = 2;
for c9 = 2:framesize:(RMS_length_T)
    c10=c9+(framesize-1);
    vector5 = MVC_TLa_new(c9:c10,1);
    vector6 = MVC_TLo_new(c9:c10,1);
    rms_x5 = sqrt(mean(vector5.^2));
    rms_x6 = sqrt(mean(vector6.^2));
    T_filtered(ct,2) = rms_x5;

```



```

T_filtered(ct,3) = rms_x6;
T_filtered(ct,1) = MVC_TLa((ct-1)*100,1);
ct = ct+1;
end

% get mean of certain number (num_maxval) of maximum values
MVC_ranking_DM = sort(DM_filtered(:,2), 'descend');
MVC_ranking_BI = sort(BI_filtered(:,2), 'descend');
MVC_ranking_TLa = sort(T_filtered(:,2), 'descend');
MVC_ranking_TLo = sort(T_filtered(:,3), 'descend');

largest10_DM = MVC_ranking_DM(1:num_maxval);
largest10_BI = MVC_ranking_BI(1:num_maxval);
largest10_TLa = MVC_ranking_TLa(1:num_maxval);
largest10_TLo = MVC_ranking_TLo(1:num_maxval);
mean_MVC_DM = mean(largest10_DM);
mean_MVC_BI = mean(largest10_BI);
mean_MVC_TLa = mean(largest10_TLa);
mean_MVC_TLo = mean(largest10_TLo);

table_meanmax = [mean_MVC_DM mean_MVC_BI mean_MVC_TLa mean_MVC_TLo];

%% MVC normalization (EMG/mean(max MVC))
% get data of DM, BI, T La, T Lo muscles
EMG_norm_input = [EMG_filtered(:,3),EMG_filtered(:,6:8)];
EMG_normalized = NaN(length(EMG_norm_input),number_MVC);

% normalization
for bb = 1:number_MVC
    for co = 1:length(EMG_norm_input)
        EMG_normalized(co,bb) = EMG_norm_input(co,bb)/table_meanmax(1,bb)*100;
    end
end
size_EMG_normalized = size(EMG_normalized);

%% interpolation (spline)
% interpolation (100 Hz)
x = EMG_filtered(1:end,1); % time
y = EMG_normalized(1:end,:);
xi = 0:.01:EMG_filtered(size_EMG_filtered(1),1);
yi = interp1(x,y,xi,'spline','extrap');

size_xi = size(xi);
frame_number = 1:size_xi(2);
EMG_interpolated = NaN(size_xi(2),(number_MVC+2));
EMG_interpolated(1:end,1) = frame_number; % frame #
EMG_interpolated(1:end,2) = xi; % time
EMG_interpolated(1:end,3:end) = yi; % data

size_EMG_interpolated = size(EMG_interpolated);
%% save post-processed EMG file
xlswrite(analysis_file,EMG_interpolated);

%% mean cycle (HBP: 5 cycles, SW: 3 cycles)
cycle1 = NaN(cycle1_length,size_EMG_interpolated(2));
cycle2 = NaN(cycle2_length,size_EMG_interpolated(2));
cycle3 = NaN(cycle3_length,size_EMG_interpolated(2));
cycle1(:,1:2) = EMG_interpolated(starting_frame1:starting_frame2-1,1:2);
cycle2(:,1:2) = EMG_interpolated(starting_frame2:starting_frame3-1,1:2);

if strcmp(five_cycles,'NO') % for SW
    cycle3(:,1:2) = EMG_interpolated(starting_frame3:ending_frame,1:2);
end

if strcmp(five_cycles,'YES') % for HBP
    cycle4 = NaN(cycle4_length,size_EMG_interpolated(2));
    cycle5 = NaN(cycle5_length,size_EMG_interpolated(2));
    cycle3(:,1:2) = EMG_interpolated(starting_frame3:starting_frame4-1,1:2);
    cycle4(:,1:2) = EMG_interpolated(starting_frame4:starting_frame5-1,1:2);
    cycle5(:,1:2) = EMG_interpolated(starting_frame5:ending_frame,1:2);
end

for tt = 3:size_EMG_interpolated(2)
    cycle1(:,tt) = EMG_interpolated(starting_frame1:starting_frame2-1,tt);
    cycle2(:,tt) = EMG_interpolated(starting_frame2:starting_frame3-1,tt);

```

```

    if strcmp(five_cycles,'NO') % for SW
        cycle3(:,tt) = EMG_interpolated(starting_frame3:ending_frame,tt);
    end
    if strcmp(five_cycles,'YES') % for HBP
        cycle3(:,tt) = EMG_interpolated(starting_frame3:starting_frame4-1,tt);
        cycle4(:,tt) = EMG_interpolated(starting_frame4:starting_frame5-1,tt);
        cycle5(:,tt) = EMG_interpolated(starting_frame5:ending_frame,tt);
    end
end

% normalize cycles (here x_max = 100 values)
DM1 = cycle1(:,3);
BI1 = cycle1(:,4);
TLa1 = cycle1(:,5);
TLo1 = cycle1(:,6);
xf_cycle = 1:1:x_max;
% interpft -> interpolation using FFT method
DM_f1 = interpft(DM1,x_max);
BI_f1 = interpft(BI1,x_max);
TLa_f1 = interpft(TLa1,x_max);
TLo_f1 = interpft(TLo1,x_max);
cycle1_final = [xf_cycle' DM_f1 BI_f1 TLa_f1 TLo_f1];

DM2 = cycle2(:,3);
BI2 = cycle2(:,4);
TLa2 = cycle2(:,5);
TLo2 = cycle2(:,6);
xf_cycle = 1:1:x_max;
DM_f2 = interpft(DM2,x_max);
BI_f2 = interpft(BI2,x_max);
TLa_f2 = interpft(TLa2,x_max);
TLo_f2 = interpft(TLo2,x_max);
cycle2_final = [xf_cycle' DM_f2 BI_f2 TLa_f2 TLo_f2];

DM3 = cycle3(:,3);
BI3 = cycle3(:,4);
TLa3 = cycle3(:,5);
TLo3 = cycle3(:,6);
xf_cycle = 1:1:x_max;
DM_f3 = interpft(DM3,x_max);
BI_f3 = interpft(BI3,x_max);
TLa_f3 = interpft(TLa3,x_max);
TLo_f3 = interpft(TLo3,x_max);
cycle3_final = [xf_cycle' DM_f3 BI_f3 TLa_f3 TLo_f3];

if strcmp(five_cycles,'YES') % for HBP
    DM4 = cycle4(:,3);
    BI4 = cycle4(:,4);
    TLa4 = cycle4(:,5);
    TLo4 = cycle4(:,6);
    xf_cycle = 1:1:x_max;
    DM_f4 = interpft(DM4,x_max);
    BI_f4 = interpft(BI4,x_max);
    TLa_f4 = interpft(TLa4,x_max);
    TLo_f4 = interpft(TLo4,x_max);
    cycle4_final = [xf_cycle' DM_f4 BI_f4 TLa_f4 TLo_f4];

    DM5 = cycle5(:,3);
    BI5 = cycle5(:,4);
    TLa5 = cycle5(:,5);
    TLo5 = cycle5(:,6);
    xf_cycle = 1:1:x_max;
    DM_f5 = interpft(DM5,x_max);
    BI_f5 = interpft(BI5,x_max);
    TLa_f5 = interpft(TLa5,x_max);
    TLo_f5 = interpft(TLo5,x_max);
    cycle5_final = [xf_cycle' DM_f5 BI_f5 TLa_f5 TLo_f5];
end

% averaging all cycles
cycle_final = NaN(length(xf_cycle),size_EMG_interpolated(2)-1);
cycle_final(:,1)=xf_cycle;

if strcmp(five_cycles,'NO') % for SW
    DM = [DM_f1 DM_f2 DM_f3];

```

```

    BI = [BI_f1 BI_f2 BI_f3];
    TLa = [TLa_f1 TLa_f2 TLa_f3];
    TLo = [TLo_f1 TLo_f2 TLo_f3];
end

if strcmp(five_cycles,'YES') % for HBP
    DM = [DM_f1 DM_f2 DM_f3 DM_f4 DM_f5];
    BI = [BI_f1 BI_f2 BI_f3 BI_f4 BI_f5];
    TLa = [TLa_f1 TLa_f2 TLa_f3 TLa_f4 TLa_f5];
    TLo = [TLo_f1 TLo_f2 TLo_f3 TLo_f4 TLo_f5];
end

for cc = 1:1:length(xf_cycle)
    cycle_final(cc,2) = mean(DM(cc,:));
    cycle_final(cc,3) = mean(BI(cc,:));
    cycle_final(cc,4) = mean(TLa(cc,:));
    cycle_final(cc,5) = mean(TLo(cc,:));
end
plot(1:x_max,cycle_final(:,2:5));
%% save mean cycle as Excel
xlswrite(cycle_file,cycle_final);
% END

```

9.1.4 On-/Off-Code

```

%% on/off of EMG
% creating overall mean cycle of crank/pushrim propulsion of all subjects
% setting threshold to define 'on' (1) / 'off' (0) state of muscle
% author: Anja Arbter

clear all
close all
clc

format long
format compact

num_files = 7; % number of input files (8SW,7H)
num_EMG = 4; % number of EMG signals
x_max = 100; % maximum value of x-axis
threshold = 10; % threshold for on/off-analysis (adapt)

%% read in all mean cycle files (created in EMG_end(_5cycles).m)
[fname1, pname1] = uigetfile('*.xlsx');
cd(pname1);
%Create fully-formed filename as a string
filename = fullfile(pname1, fname1);
%Check that file exists
assert(exist(filename,'file')==2, '%s does not exist.', filename);
%Read in the data, skipping the first row
onoff_file1 = xlsread(filename);
if num_files==8
    filename(end-12) = '2';
    filename2 = filename;
    onoff_file2 = xlsread(filename2);
end
filename(end-12) = '3';
filename3 = filename;
onoff_file3 = xlsread(filename3);
filename(end-12) = '4';
filename4 = filename;
onoff_file4 = xlsread(filename4);
filename(end-12) = '5';
filename5 = filename;
onoff_file5 = xlsread(filename5);
filename(end-12) = '6';
filename6 = filename;
onoff_file6 = xlsread(filename6);
filename(end-12) = '7';
filename7 = filename;
onoff_file7 = xlsread(filename7);
filename(end-12) = '8';

```

```

filename8 = filename;
onoff_file8 = xlsread(filename8);

%% average all mean cycle files
% in general
DM_overall =
[onoff_file1(:,2),onoff_file2(:,2),onoff_file3(:,2),onoff_file4(:,2),onoff_file5(:,2),onoff_file6(:,2),onoff_file7(:,2)];
BI_overall =
[onoff_file1(:,3),onoff_file2(:,3),onoff_file3(:,3),onoff_file4(:,3),onoff_file5(:,3),onoff_file6(:,3),onoff_file7(:,3)];
TLa_overall =
[onoff_file1(:,4),onoff_file2(:,4),onoff_file3(:,4),onoff_file4(:,4),onoff_file5(:,4),onoff_file6(:,4),onoff_file7(:,4)];
TLo_overall =
[onoff_file1(:,5),onoff_file2(:,5),onoff_file3(:,5),onoff_file4(:,5),onoff_file5(:,5),onoff_file6(:,5),onoff_file7(:,5)];

if num_files==8
    DM_overall =
[onoff_file1(:,2),onoff_file2(:,2),onoff_file3(:,2),onoff_file4(:,2),onoff_file5(:,2),onoff_file6(:,2),onoff_file7(:,2),onoff_file8(:,2)];
    BI_overall =
[onoff_file1(:,3),onoff_file2(:,3),onoff_file3(:,3),onoff_file4(:,3),onoff_file5(:,3),onoff_file6(:,3),onoff_file7(:,3),onoff_file8(:,3)];
    TLa_overall =
[onoff_file1(:,4),onoff_file2(:,4),onoff_file3(:,4),onoff_file4(:,4),onoff_file5(:,4),onoff_file6(:,4),onoff_file7(:,4),onoff_file8(:,4)];
    TLo_overall =
[onoff_file1(:,5),onoff_file2(:,5),onoff_file3(:,5),onoff_file4(:,5),onoff_file5(:,5),onoff_file6(:,5),onoff_file7(:,5),onoff_file8(:,5)];
end

% special case for this measurement series based on visual inspection
% hide other cases

% SW 25: DM: no2,4,8, BI: no4,8, T La: no1,4,8, TLo: no3,4,5,8
% DM_overall =
[onoff_file1(:,2),onoff_file3(:,2),onoff_file5(:,2),onoff_file6(:,2),onoff_file7(:,2)];
% BI_overall =
[onoff_file1(:,3),onoff_file2(:,3),onoff_file3(:,3),onoff_file5(:,3),onoff_file6(:,3),onoff_file7(:,3)];
% TLa_overall =
[onoff_file2(:,4),onoff_file3(:,4),onoff_file5(:,4),onoff_file6(:,4),onoff_file7(:,4)];
% TLo_overall = [onoff_file1(:,5),onoff_file2(:,5),onoff_file6(:,5),onoff_file7(:,5)];

% SW 35: DM: no2,4,8, BI:no4,8, T La: no1,4,8, TLo: no3,4,5,7,8
% DM_overall =
[onoff_file1(:,2),onoff_file3(:,2),onoff_file5(:,2),onoff_file6(:,2),onoff_file7(:,2)];
% BI_overall =
[onoff_file1(:,3),onoff_file2(:,3),onoff_file3(:,3),onoff_file5(:,3),onoff_file6(:,3),onoff_file7(:,3)];
% TLa_overall =
[onoff_file2(:,4),onoff_file3(:,4),onoff_file5(:,4),onoff_file6(:,4),onoff_file7(:,4)];
% TLo_overall = [onoff_file1(:,5),onoff_file2(:,5),onoff_file6(:,5)];

% H 25/35: DM: no1,2,4, BI: no1,2,3,4, T La: no1,4, TLo: no1,3,4
% onoff_file1 is subject 2
% DM_overall =
[onoff_file3(:,2),onoff_file5(:,2),onoff_file6(:,2),onoff_file7(:,2),onoff_file8(:,2)];
% BI_overall = [onoff_file5(:,3),onoff_file6(:,3),onoff_file7(:,3),onoff_file8(:,3)];
% TLa_overall =
[onoff_file1(:,4),onoff_file3(:,4),onoff_file5(:,4),onoff_file6(:,4),onoff_file7(:,4),onoff_file8(:,4)];
% TLo_overall =
[onoff_file1(:,5),onoff_file5(:,5),onoff_file6(:,5),onoff_file7(:,5),onoff_file8(:,5)];

mean_DM = nan(x_max,1);
mean_BI = nan(x_max,1);
mean_TLa = nan(x_max,1);
mean_TLo = nan(x_max,1);
for cc = 1:1:x_max
    mean_DM(cc,1) = mean(DM_overall(cc,:));
    mean_BI(cc,1) = mean(BI_overall(cc,:));
    mean_TLa(cc,1) = mean(TLa_overall(cc,:));

```

```

    mean_TLo(cc,1) = mean(TLo_overall(cc,:));
end

onoff_overall = [mean_DM, mean_BI, mean_TLa, mean_TLo];

prop = filename((end-10):(end-9));
plot(1:x_max,onoff_overall);
if prop=='SW'
    propulsion = 'pushrim propulsion';
elseif prop=='_H'
    propulsion = 'crank propulsion';
end
watts = filename((end-7):(end-5));
title_mean = ['Averaged EMG for 1 cycle of ' propulsion ' at ' watts];
title(title_mean)
xlabel('percentage of cycle [%]')
ylabel('percentage of muscle activity [%]')
legend({'musculus deltoideus medialis','musculus biceps brachii','musculus triceps brachii
caput laterale','musculus triceps brachii caput longum'},'Location','northeast')

%% calculate on (1) and off (0) state of muscles
x_axis = 1:x_max;
onoff_final = [x_axis',nan(x_max,num_EMG)];

for d = 1:x_max
    for b = 2:(num_EMG+1)
        if onoff_overall(d,b-1)>threshold
            onoff_final(d,b) = 1;
        else
            onoff_final(d,b) = 0;
        end
    end
end

% plot(onoff_final(:,1),onoff_final(:,2:(num_EMG+1)));
%% save mean cycle as Excel
onoff_file = ['onoff_' prop watts];
xlswrite(onoff_file,onoff_final);

% END

```

9.1.5 OpenSim Code

```

%% OpenSim mean file
% author: Anja Arbter

clear all
close all
clc

format long
format compact

% adapt for each subject and measurement
five_cycles = 'NO'; % 'NO' for SW, 'YES' for HBP
x_max = 100; % maximum value of x-axis

% read in OpenSim results file
[fname1, pname1] = uigetfile('*.xlsx');
cd(pname1);
%Create fully-formed filename as a string
filename = fullfile(pname1, fname1);
%Check that file exists
assert(exist(filename,'file')==2, '%s does not exist.', filename);
%Read in the data, skipping the first row
OS_file = xlsread(filename);

size_OS = size(OS_file);

% overwrite first column with frame numbering
OS_file(:,1) = OS_file(:,1)+1;

```

```

% get starting/ending frame numbers of frame file
subject = fname1(1:2);
frame_file = xlsread('Frames.xlsx');
if fname1(3:6) == 'SW25'
    column = 1+4*(str2num(subject(2))-1);
elseif fname1(3:6) == 'SW35'
    column = 2+4*(str2num(subject(2))-1);
elseif fname1(3:6) == '_H25'
    column = 3+4*(str2num(subject(2))-1);
elseif fname1(3:6) == '_H35'
    column = 4+4*(str2num(subject(2))-1);
else
    disp('column not found');
end

start_frame1 = frame_file(column,1);
start_frame2 = frame_file(column,2);
start_frame3 = frame_file(column,3);
end_frame = frame_file(column,6);
if strcmp(five_cycles,'YES') % for HBP
    start_frame4 = frame_file(column,4);
    start_frame5 = frame_file(column,5);
end

% get cycle length
file_length = end_frame-start_frame1+1;
cycle1_length = start_frame2-start_frame1;
cycle2_length = start_frame3-start_frame2;
if strcmp(five_cycles,'NO') % for SW
    cycle3_length = end_frame-start_frame3;
end
if strcmp(five_cycles,'YES') % for HBP
    cycle3_length = start_frame4-start_frame3;
    cycle4_length = start_frame5-start_frame4;
    cycle5_length = end_frame-start_frame5;
end

% change frame number to OS frame number format
minus = start_frame1;
starting_frame1 = start_frame1-minus+1;
starting_frame2 = start_frame2-minus+1;
starting_frame3 = start_frame3-minus+1; % OS file 1 frame shorter
if strcmp(five_cycles,'YES') % for HBP
    starting_frame4 = start_frame4-minus+1;
    starting_frame5 = start_frame5-minus+1;
end
ending_frame = end_frame-minus; % OS file 1 frame shorter

%% cycle file (5 cycles)
cycle1 = nan(cycle1_length,size_OS(2));
cycle2 = nan(cycle2_length,size_OS(2));
cycle3 = nan(cycle3_length,size_OS(2));
cycle1(:,1:2) = OS_file(starting_frame1:starting_frame2-1,1:2);
cycle2(:,1:2) = OS_file(starting_frame2:starting_frame3-1,1:2);
for tt = 3:size_OS(2)
    cycle1(:,tt) = OS_file(starting_frame1:starting_frame2-1,tt);
    cycle2(:,tt) = OS_file(starting_frame2:starting_frame3-1,tt);
end
if strcmp(five_cycles,'NO') % for HBP
    cycle3(:,1:2) = OS_file(starting_frame3:ending_frame,1:2);
    for ttt = 3:size_OS(2)
        cycle3(:,ttt) = OS_file(starting_frame3:ending_frame,ttt);
    end
end
if strcmp(five_cycles,'YES')
    cycle4 = nan(cycle4_length,size_OS(2));
    cycle5 = nan(cycle5_length,size_OS(2));
    cycle3(:,1:2) = OS_file(starting_frame3:starting_frame4-1,1:2);
    cycle4(:,1:2) = OS_file(starting_frame4:starting_frame5-1,1:2);
    cycle5(:,1:2) = OS_file(starting_frame5:ending_frame,1:2);
    for tttt = 3:size_OS(2)
        cycle3(:,tttt) = OS_file(starting_frame3:starting_frame4-1,tttt);
        cycle4(:,tttt) = OS_file(starting_frame4:starting_frame5-1,tttt);
        cycle5(:,tttt) = OS_file(starting_frame5:ending_frame,tttt);
    end
end

```

```

end
end

% normalize cycles (here x_max = 100 values)
elv_angle1 = cycle1(:,3);
shoulder_elv1 = cycle1(:,4);
shoulder_rot1 = cycle1(:,5);
elbow_flexion1 = cycle1(:,6);
pro_sup1 = cycle1(:,7);
deviation1 = cycle1(:,8);
flexion1 = cycle1(:,9);
% interpft -> interpolation using FFT method
elv_angle1_f = interpft(elv_angle1,x_max);
shoulder_elv1_f = interpft(shoulder_elv1,x_max);
shoulder_rot1_f = interpft(shoulder_rot1,x_max);
elbow_flexion1_f = interpft(elbow_flexion1,x_max);
pro_sup1_f = interpft(pro_sup1,x_max);
deviation1_f = interpft(deviation1,x_max);
flexion1_f = interpft(flexion1,x_max);

elv_angle2 = cycle2(:,3);
shoulder_elv2 = cycle2(:,4);
shoulder_rot2 = cycle2(:,5);
elbow_flexion2 = cycle2(:,6);
pro_sup2 = cycle2(:,7);
deviation2 = cycle2(:,8);
flexion2 = cycle2(:,9);
% interpft -> interpolation using FFT method
elv_angle2_f = interpft(elv_angle2,x_max);
shoulder_elv2_f = interpft(shoulder_elv2,x_max);
shoulder_rot2_f = interpft(shoulder_rot2,x_max);
elbow_flexion2_f = interpft(elbow_flexion2,x_max);
pro_sup2_f = interpft(pro_sup2,x_max);
deviation2_f = interpft(deviation2,x_max);
flexion2_f = interpft(flexion2,x_max);

elv_angle3 = cycle3(:,3);
shoulder_elv3 = cycle3(:,4);
shoulder_rot3 = cycle3(:,5);
elbow_flexion3 = cycle3(:,6);
pro_sup3 = cycle3(:,7);
deviation3 = cycle3(:,8);
flexion3 = cycle3(:,9);
% interpft -> interpolation using FFT method
elv_angle3_f = interpft(elv_angle3,x_max);
shoulder_elv3_f = interpft(shoulder_elv3,x_max);
shoulder_rot3_f = interpft(shoulder_rot3,x_max);
elbow_flexion3_f = interpft(elbow_flexion3,x_max);
pro_sup3_f = interpft(pro_sup3,x_max);
deviation3_f = interpft(deviation3,x_max);
flexion3_f = interpft(flexion3,x_max);

if strcmp(five_cycles,'YES') % for HBP
    elv_angle4 = cycle4(:,3);
    shoulder_elv4 = cycle4(:,4);
    shoulder_rot4 = cycle4(:,5);
    elbow_flexion4 = cycle4(:,6);
    pro_sup4 = cycle4(:,7);
    deviation4 = cycle4(:,8);
    flexion4 = cycle4(:,9);
    % interpft -> interpolation using FFT method
    elv_angle4_f = interpft(elv_angle4,x_max);
    shoulder_elv4_f = interpft(shoulder_elv4,x_max);
    shoulder_rot4_f = interpft(shoulder_rot4,x_max);
    elbow_flexion4_f = interpft(elbow_flexion4,x_max);
    pro_sup4_f = interpft(pro_sup4,x_max);
    deviation4_f = interpft(deviation4,x_max);
    flexion4_f = interpft(flexion4,x_max);

    elv_angle5 = cycle5(:,3);
    shoulder_elv5 = cycle5(:,4);
    shoulder_rot5 = cycle5(:,5);
    elbow_flexion5 = cycle5(:,6);
    pro_sup5 = cycle5(:,7);

```

```

deviation5 = cycle5(:,8);
flexion5 = cycle5(:,9);
% interpft -> interpolation using FFT method
elv_angle5_f = interpft(elv_angle5,x_max);
shoulder_elv5_f = interpft(shoulder_elv5,x_max);
shoulder_rot5_f = interpft(shoulder_rot5,x_max);
elbow_flexion5_f = interpft(elbow_flexion5,x_max);
pro_sup5_f = interpft(pro_sup5,x_max);
deviation5_f = interpft(deviation5,x_max);
flexion5_f = interpft(flexion5,x_max);
end

% averaging all cycles
cycle_file = nan(x_max,size_OS(2)-1);
x_cycle = 1:1:x_max;
cycle_file(:,1)= x_cycle';

if strcmp(five_cycles,'NO') % for SW
    elv_angle = [elv_angle1_f elv_angle2_f elv_angle3_f];
    shoulder_elv = [shoulder_elv1_f shoulder_elv2_f shoulder_elv3_f];
    shoulder_rot = [shoulder_rot1_f shoulder_rot2_f shoulder_rot3_f];
    elbow_flexion = [elbow_flexion1_f elbow_flexion2_f elbow_flexion3_f];
    pro_sup = [pro_sup1_f pro_sup2_f pro_sup3_f];
    deviation = [deviation1_f deviation2_f deviation3_f];
    flexion = [flexion1_f flexion2_f flexion3_f];
end

if strcmp(five_cycles,'YES') % for HBP
    elv_angle = [elv_angle1_f elv_angle2_f elv_angle3_f elv_angle4_f elv_angle5_f];
    shoulder_elv = [shoulder_elv1_f shoulder_elv2_f shoulder_elv3_f shoulder_elv4_f
    shoulder_elv5_f];
    shoulder_rot = [shoulder_rot1_f shoulder_rot2_f shoulder_rot3_f shoulder_rot4_f
    shoulder_rot5_f];
    elbow_flexion = [elbow_flexion1_f elbow_flexion2_f elbow_flexion3_f elbow_flexion4_f
    elbow_flexion5_f];
    pro_sup = [pro_sup1_f pro_sup2_f pro_sup3_f pro_sup4_f pro_sup5_f];
    deviation = [deviation1_f deviation2_f deviation3_f deviation4_f deviation5_f];
    flexion = [flexion1_f flexion2_f flexion3_f flexion4_f flexion5_f];
end

% creat final file, average of cycles
for cc = 1:1:x_max
    cycle_file(cc,2) = mean(elv_angle(cc,:));
    cycle_file(cc,3) = mean(shoulder_elv(cc,:));
    cycle_file(cc,4) = mean(shoulder_rot(cc,:));
    cycle_file(cc,5) = mean(elbow_flexion(cc,:));
    cycle_file(cc,6) = mean(pro_sup(cc,:));
    cycle_file(cc,7) = mean(deviation(cc,:));
    cycle_file(cc,8) = mean(flexion(cc,:));
end
disp ('final cycle file created.');
```

```

%% plot
plot(1:x_max,cycle_file);
title('Averaged data for 1 cycle')
xlabel('percentage of cycle')
ylabel('angle in deg')
legend({'Elv. Angle','Shoulder Elv.','Shoulder rot','Elbow
flex','Pro_sup','deviation','flexion'},'Location','northeast')

%% save as Excel
final_name = ['final_mean_OS' fname1(1:6)];
xlswrite(final_name,cycle_file);

%END

```


9.3 Annex 3: Testing Protocols

Subject		1	2	3	4	5	6	7	8
Dominant	r/l	r	l	r	r	r	r	r	r
Injury	y/n	n	n	n	n	n	n	n	n
Isometric Measurements									
Point of Handle Application									
E1-PoA	cm	26	25	20	22	23	27	26	25
S1-PoA	cm	29	27	27	24	23	32	29	27
Marker Placement									
Length measurements between markers									
CL1-SH1	cm	15	20	16	16	17	16	19	17
SH1-EL1	cm	34	32	30	29	28	35	35	32
EL1-WR1	cm	27	28	21	24	26	29	28	26
WR1-FI1	cm	7	9	8	9	8	10	11	9
Experiments									
Pushrim									
Max. PO	W	20	45	25	20	40	40	45	25
Max. RPM		44	53	39	34	61	53	57	53
Crank									
Max. PO	W	20	45	30	30	45	45	45	45
Max. RPM		45	49	41	43	47	48	60	45
Additional information									
Smoker	y/n	y	y	n	n	y	y	n	N
		SW off, only EMG							Handle measurement twice because of marker loosening

9.2 Annex 3: Frame File

subject	starting frame1	starting frame2	starting frame3	starting frame4	starting frame5	ending frame
1_SW_25	344	432	521	0	0	610
1_SW_35	360	460	556	0	0	664
1_H_25	0	0	0	0	0	0
1_H_35	316	508	710	0	0	1002
2_SW_25	367	464	562	0	0	650
2_SW_35	265	372	472	0	0	574
2_H_25	385	482	576	0	0	668
2_H_35	429	527	627	725	825	926
3_SW_25	570	641	720	0	0	795
3_SW_35	606	673	738	0	0	807
3_H_25	226	335	447	562	673	781
3_H_35	158	260	356	452	554	653
4_SW_25	150	234	290	0	0	348
4_SW_35	274	332	389	0	0	448
4_H_25	220	339	467	605	758	908
4_H_35	395	628	835	0	0	1046
5_SW_25	1070	1184	1292	0	0	1402
5_SW_35	684	786	890	0	0	996
5_H_25	453	561	673	0	0	784
5_H_35	679	784	889	994	1098	1203
6_SW_25	546	632	722	0	0	805
6_SW_35	850	939	1029	0	0	1116
6_H_25	106	211	317	423	531	637
6_H_35	666	787	908	1025	1140	1260
7_SW_25	712	806	896	0	0	990
7_SW_35	727	816	904	0	0	992
7_H_25	642	737	840	946	1053	1162
7_H_35	419	513	612	707	800	888
8_SW_25	425	496	570	0	0	640
8_SW_35	262	335	414	0	0	486
8_H_25	550	649	749	851	953	1054
8_H_35	435	531	626	718	808	898