

Masterarbeit

Efficient Object Detection through Grasp Intention

Marian Theiss

30.11.2015

Albert-Ludwigs-Universität Freiburg im Breisgau
Technische Fakultät
Embedded Systems Engineering

Eingereichte Masterarbeit gemäß den Bestimmungen der Prüfungsordnung der Albert-Ludwigs-Universität Freiburg für den Studiengang Master of Science (M. Sc.) Embedded Systems Engineering vom 3. 6. 2014.

Bearbeitungszeitraum

18. 06. 2015 – 30. 11. 2015

Gutachter

Prof. Dr. Kristof Van Laerhoven

Prof. Dr. Bernd Becker

Betreuer

Prof. Dr. Kristof Van Laerhoven

Dipl.-Inf. Philipp M. Scholl

Erklärung

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

Abstract

Several wearable sensing approaches have in the past years been suggested that automatically detect the object in a person's hand. This could be used in a variety of application scenarios, from assisting engineers in maintenance work as they switch between hand-held tools, to recording wet lab protocols based on object usage. Several of the proposed approaches use RFID-tags on the objects together with wrist- or hand-worn RFID readers, others use wearable cameras that visually detect the objects, or detect capacitive characteristics of the object. All of these methods are power-hungry, however, as they continuously and actively sense for any nearby objects. This work proposes to improve such systems by detecting when grasps occur, and to only then activate objects detection.

An off-the-shelf Inertial Measurement Unit (IMU) with an accelerometer and a gyroscope as well as a surface electromyography (EMG) array have been used for the grasp detection, with both sensors worn on the forearm near to the elbow. In the studies, the data from the forearm sensors are streamed to grasp detection software running from a standard computer, which activates a prototype wrist-worn RFID-reader via Bluetooth Low Energy (BLE), whenever grasps are detected.

As part of this work, research into the anatomy of grasps has been performed and a model for a grasp detection has been developed. Based on this, an investigation into possible resource-efficient pre-processing routines (so-called classification features) from the inertial and electromyography data has been carried out. The developed grasp detection algorithms are tested in several studies with multiple participants: This grasp detection approach achieved a recall of 95.5 %, which is considerably better than a baseline approach using a standard machine learning technique.

Finally, energy consumption of the developed method has been studied and compared with an object detector without a grasp detection: The overhead for the grasp detection is estimated at 12 mA at 3 V, depending on the use case, with a state-of-the-art object detection approach consuming at least 60 mA. In the studies a case dependent energy saving has been found.

Zusammenfassung

In den vergangenen Jahren wurden mehrere Technologien für tragbare Sensoren vorgeschlagen, welche automatisch gehaltene Objekte erkennen. Objektetektoren können in einer Vielzahl von Anwendungsfällen eingesetzt werden, angefangen bei der Unterstützung von technischen Aufgaben, welche einen häufigen Gerätewechsel einschließen, bis hin zur objektbasierten Aufnahme von Laborprotokollen. Viele der vorgestellten Ansätze verwenden RFID. Die RFID-Tags werden dafür auf den Objekten angebracht und von einem am Handgelenk oder an der Hand getragener RFID-Leser ausgelesen. Andere Ansätze verwenden tragbare Kameras, welche die Objekte visuell detektieren. Auch kapazitive Eigenschaften des Objektes können gemessen werden. Alle diese Methoden sind energetisch ineffizient, da sie kontinuierlich und aktiv nach nahen Objekten suchen. Die Idee hinter dieser Arbeit ist den richtigen Zeitpunkt für die Objektdetektion zu finden und nur dann den Sensor einzuschalten. Hierfür wurde eine Griffdetektion entwickelt.

Für die Sensorik der Griffdetektion wurde ein Inertialsensor für Beschleunigung und Winkelgeschwindigkeiten, sowie ein mehrkanäiger Oberflächen-Elektromyographie-Sensor eingesetzt. Die Sensoren werden in der Nähe des Ellenbogens am Unterarm befestigt. Die Unterarmdaten werden in der Studie zu einer Griffdetektionsapplikation gesendet, welche auf einem standardmäßigen Computer läuft. Diese aktiviert bei jeder Griffdetektion einen am Handgelenk getragenen Prototypen eines RFID-Lesers. Die Kommunikation wurde über Bluetooth Low Energy realisiert.

Die zur Greifbewegung gehörige Anatomie wurde untersucht und ein Modell zur Griffdetektion entwickelt. Auf dieser Grundlage wurde nach einer sparsamen Verarbeitung der Intertial- und Elektromyographie-Daten gesucht. Die entwickelten Algorithmen wurden in diversen Testreihen mit mehreren Teilnehmern erprobt: Ein Recall von 95.5 % wurde erreicht, was deutlich über den Referenztests lag, welche ein normales Maschinelles Lernverfahren verwenden.

Zum Abschluss wurde der Energieverbrauch der entwickelten Methoden zusammengefasst und mit einem Objektdetektor ohne Griffdetektion verglichen: Abhängig vom Anwendungsfall wurde der Grundverbrauch der Griffdetektion auf 12 mA bei 3 V geschätzt, während ein Objektdetektor nach Stand der Technik mindestens 60 mA benötigt. Die Menge der eingesparten Energie ist stark vom Anwendungsfall abhängig.

Contents

Abstract	v
Zusammenfassung	vii
List of Figures	xv
List of Tables	xvii
Acronyms	xix
1. Introduction	1
1.1. Objective of this Work	1
1.2. Applications	2
1.3. Prototype Overlook	2
2. Related Work	5
2.1. Gesture Recognition	5
2.2. Grasp Modelling and Recognition	7
2.3. Object Detection	10
2.4. Activity Detection	11
2.5. Deep Layer Muscle Surface Electromyography	13
3. Key Concepts	15
3.1. Electromyography	15
3.2. Inertial Measurement Unit	18
3.3. Radio Frequency Identification	18
3.4. Bluetooth Low Energy	19
3.5. Random Forest Classification	21
4. System Concepts	23
4.1. Steps for a Grasping Sequence	23
4.1.1. Grasp Steps Basics	23
4.1.2. Moving Hand to Object	23
4.1.3. Closing Hand around Object	24
4.1.4. Holding Object	25
4.1.5. Releasing Object	25
4.2. Grasping Behaviours	25

4.3.	Grasp Detection Model	26
4.3.1.	Basics of the Grasp Detection Model	26
4.3.2.	Model Layer Functions	26
4.3.2.1.	Layer 1: Base Activity Recognition	26
4.3.2.2.	Layer 2: Grasp Segments Recognition	27
4.3.2.3.	Layer 3: Final Grasp Evaluation	27
4.3.3.	Model Layer Realisations	28
4.3.3.1.	Layer 1: Base Activities Features	28
4.3.3.2.	Layer 2: Grasping Features	28
4.3.3.3.	Layer 3: Sequencing of the Features	29
4.4.	Prototype	30
4.4.1.	Prototype Overview and Connection	30
4.4.2.	Microcontroller	32
4.4.3.	RFID Components	32
4.4.4.	EMG and IMU Sensor System	33
4.4.5.	Bluetooth Configuration	34
4.4.6.	Object Detector Case	34
4.4.7.	Development Environment	34
4.4.8.	Baseline Test Environment	35
4.5.	Forearm Muscles and EMG Signals	36
4.5.1.	Definitions	36
4.5.2.	Detected Signals for Movements	36
4.5.3.	Posterior, superficial Muscles	38
4.5.4.	Anterior, superficial Muscles	40
4.5.5.	Posterior, deep Muscles	41
4.5.6.	Anterior, deep Muscles	41
4.6.	Feature Extraction	41
4.6.1.	Signal Preprocessing	41
4.6.2.	Statistical Values	42
4.6.3.	Methods	43
4.7.	Developed Software	44
4.7.1.	Logger	44
4.7.2.	Feature Extraction Script	45
4.7.3.	Grasp Detection Scripts	46
4.7.4.	Object Detector Software	46
4.7.5.	Device Communication	47
5.	Experiments	49
5.1.	Experimental Setup	49
5.2.	Logfile Detection	49
5.3.	Real-Time Detection	52
5.4.	Grasp Detection	52
5.5.	Grasp Recognition False Positive Grading	53
5.6.	Classification Baseline	53

5.7. Object Detection with Grasp Detection Underlay	54
5.8. Energy Approximation	55
6. Results	57
6.1. Features for Grasp Detection	57
6.1.1. EMG Features realized in the Grasp Detection	57
6.1.2. IMU Features realized in the Grasp Detection	59
6.2. Grasp Detection via Logfiles Results	60
6.3. Grasp Detection in Real-Time Results	62
6.4. Grasp Recognition False Positive Grading Results	62
6.5. Classifier Baseline Results	64
6.6. Object Detection with Grasp Detection Results	66
6.7. Energy Savings	66
7. Conclusions	69
7.1. Discussion	69
7.2. Future Work	69
Acknowledgement	71
A. Algorithm Energy Efficiency	73
B. Bluetooth Low Energy Communication Commands	77
B.1. Hard- and Software Configuration	77
B.2. Reading and Writing to Serial Port via Matlab	77
B.3. Packet Structure	78
B.4. General Commands	79
B.5. Communication with Myo	81
B.6. Communication with RFduino	83
C. Test Scripts	85
C.1. Logged Grasp Detection and Object Detection Test Script	85
C.2. False Positive Test Script	86
D. Test Protocols	89
E. Program Sources and Flowcharts	95
E.1. Object Detector Code	95
E.2. Forearm Sensor Logger Code	96
E.3. Forearm Sensor Data Plotter Code	97
E.4. Feature Space Visualization Code	98
E.5. Grasp Detection via Log Files Code	99
E.6. Grasp Detection in Real-Time and Object Detection Code	100
Bibliography	101

List of Figures

2.1. Early grasp taxonomy by Cutkosky et al. [35]. The figure has been included with the permission of Mark R. Cutkosky. ©1989 IEEE	8
2.2. Later grasp taxonomy by Feix et al. [38]. The figure has been included with the permission of Thomas Feix. ©2015 IEEE	9
3.1. Temporal evolvement of a motor unit action potential. It starts with a rise of the potential (depolarisation), which is followed by a repolarisation and a hyperpolarisation to regain the rest potential.	16
3.2. Detected voltage difference in a two electrode EMG sensor by concerning a single MUAP. Positive peak is reached when the MUAP covers electrode A. Second peak is reached when the MUAP covers electrode B. While the MUAP is outside of the electrodes the difference is zero.	17
3.3. Protocol stack of Bluetooth Low Energy. Host and Controller are typically implemented on several chips.	19
4.1. Illustration of a complete grasping sequence. Step representation: (1) posture before a grasp intention (null), (2) Moving Hand to Object, (3) Closing Hand around Object, (4) Holding Object, (5) Releasing Object, (6) posture after a grasp (null).	24
4.2. Approach to use the three layer grasp detection model. Layer 1 is based only on acceleration data, which is sparse in contrast to the other sensors. On basis of the results in Layer 1 further checks in Layer 2 and Layer 3 may be necessary, which are based on the more costly EMG data. If a grasp has been found in processing the whole grasp detection model, the most costly object detector is enabled.	27
4.3. Body-worn prototype hardware components. A commercial EMG and IMU sensor (Myo from Thalmic Labs) collects the data of the muscles in the arm as well as the movements of the arm. The RFID reader is responsible for the reading of RFID tags and is establishing an object detection by means of the information attached to the tag. Both components are in the prototype connected via BLE to a grasp detection system on a separate, assisting computer.	30

4.4. Prototype connection of the subsystems. The forearm sensor is collecting acceleration and gyroscope data. This sensor data is sent to the grasp detection, running on an extra computer which is assisting the prototype. Detected grasps are enabling the object detection. The object detector is constructed via a communication and controlling module and a RFID reader module. Object mounted tags are detected via RFID. The system is connected via BLE.	31
4.5. EMG positioning and IMU directions of the forearm sensor.	33
4.6. 3D printed case, used for the object detector. It is been mounted with an until 3 cm broad armband to the arm. The closed back is intended for placing the microcontroller and the RFID module. The front is intended for the battery. The opening in the top is for inserting the device, closed with the armband which is intended to be merged in the two holes in the side. The opening in the front can be used for antenna cable.	35
4.7. Posterior muscles of the forearm. The left image is showing the superficial and the right image the deep layer. Images are drawn by Dr. Henry Vandyke Carter and under the public domain license.	38
4.8. Anterior muscles of the forearm. The left image is showing the superficial and the right image the deep layer. Images are drawn by Dr. Henry Vandyke Carter and under the public domain license.	39
5.1. Items used in the experiment were tagged with RFID tags and were chosen to be similar with the objects reported in research project on wearable RFID object detection [1].	50
6.1. Activation signals of the eight EMG sensor units over time in a series of seven grasps of different objects, from the same participant. Grasping steps are numbered according to the steps in Figure 4.1. Coloured bars indicate a step change, where the third step is a relatively short period of time. The EMG sensor units have been worn equidistantly on the right forearm, near the elbow, with a clockwise numbering starting on the anterior side.	57
6.2. Recall for the participants with all used objects. Since the number of grasps is varying per object, two different results are plotted. The blue bar is illustrating the recall in percent per grasp, while the red bar is showing the recall when every test object has the same influence in the result.	61
6.3. Recall for the chosen objects for all participants. Since every grasp of the object have been counted and the grasping differed per participant, this value has varied. The blue bar illustrates the recall when every grasp counts even, while the red bar illustrates the recall when every participant has the same influence on the result.	61

6.4. Recall, false positive rate and precision of the false positive grading tests. Results are per participant.	63
6.5. Detections of the objects and tasks in the false positive grading tests.	63
6.6. Results of the false positive grading tests for Participant 2 with changed parameters.	64
6.7. Recall of the baseline tests with Random Forests classification. Datasets are equal to grasp detection with log file. Tested has been for the step "Holding Object". Null is represented everything else.	65
6.8. Precision of the baseline tests with Random Forests classification. Datasets are equal to grasp detection with log file. Tested has been for the step "Holding Object". Null is representing everything else.	65
E.1. Flowchart of the microcontroller program.	95
E.2. Flowchart of the simple forearm sensor logger, communicating with Myo Connect software.	96
E.3. Flowchart of the raw forearm sensor data visualizer.	97
E.4. Flowchart of the feature space visualizer of the grasp detection.	98
E.5. Flowchart of the grasp detection via log files.	99
E.6. Flowchart of grasp detection in real-time. Green marked boxes are extensions for the object detection.	100

List of Tables

1.1.	Three subsystems belonging to the prototype. The functions that are relevant for the user, are described in this table.	3
4.1.	Movements measurable by an eight channel surface EMG. Channel ordering according to figure Figure 4.5. Signal strength are in static positions in time domain classified as weak (w), moderate (m), strong (s) activation.	37
A.1.	Operations for incoming acceleration data packets with 50 Hz.	74
A.2.	Operations of incoming EMG data packets with 200 Hz.	74
A.3.	Operations of incoming RFID data packets with $\ll 1$ Hz.	74
A.4.	Operations of the calculation for the grasping features of Layer 2 in 4.5455 Hz.	74
A.5.	Operations of the calculation for the resting features of Layer 1 in 4.5455 Hz.	75
A.6.	Operations of the calculation for the walking features of Layer 1 in 4.5455 Hz.	75
A.7.	Operations of the temporal analysis with 4.5455 Hz.	75
B.1.	General packet structure of the BLED112 HCI Commands [70]	78
B.2.	Command class IDs for the second octet in the BLED112 HCI Commands [70]	79
B.3.	Packet Structure of a GAP Discovery Scan Response [70]	80
B.4.	Connection status of connected Bluetooth devices [70]	81
D.1.	False positive rate of the false positives test per participant.	90
D.2.	Recall of the false positives test per participant.	90
D.3.	Object detection test protocol of Participant 1.	91
D.4.	Object detection test protocol of Participant 2.	92
D.5.	Object detection test protocol of Participant 3.	93
D.6.	Object detection test protocol of Participant 4.	94

Acronyms

ADL activities of daily living

BLE Bluetooth Low Energy

EMG electromyography

FFT fast Fourier transform

GAP Generic Access Profile

GATT Generic Attribute Profile

GPIO general purpose input/output

HCI Host Controller Interface

HMM hidden Markov model

IFFT inverse fast Fourier transform

IMU inertial measurement unit

L2CAP Logical Link Control and Application Protocol

MFCC Mel frequency cepstral coefficients

MUAP motor unit action potential

PCA principal component analysis

PDF probability distribution function

RFID radio frequency identification

SMP Security Manager Protocol

USB Universal Serial Bus

1. Introduction

1.1. Objective of this Work

This work aims at developing a prototype for a low power detection system for radio frequency identification (RFID) tagged objects. Object detection has been already shown to be possible in various works [1, 2, 3, 4].

Since body-worn object detectors are limited to the battery lifetime and the object detector's sensor typically draw a high amount of energy, state of the art systems are limited to a few hours of using. In this work, reduction of energy shall be achieved via a grasp detection, which only activates the object detector at a grasp. For this reason, grasp detection muscle activity and arm movement shall be interpreted. Muscle activity is detected via an eight channel electromyography (EMG) sensor, placed around the forearm near to the elbow. Arm movement is detected via an inertial measurement unit (IMU) placed on the forearm near the elbow. The sensor used is a commercial device named Myo [5] from Thalmic Labs.

As part of the work suitable testing and development methods had to be found. A determination of suitable hardware components has to be made. The communication of the systems have to be established.

Based on the data, a classification method for grasps has to be developed and features have to be found and evaluated. A layer model for the detection of grasps was introduced. A second classification method, based on Random Forests (section 3.5), have been used as baseline.

The system had to be tested on a high number of study participants in several tests. The system has been tuned to keep the missed grasps low while achieving a reasonable amount of time the object detector is running.

The RFID object detection has been developed and coupled with the grasp detection. The full prototype, including forearm sensor, grasp detection and object detector, is presented.

The energy consumption was evaluated and compared to object detection systems without grasp detection.

Chapter 2 starts with introducing related work in this field. Chapter 3 is presenting actual technical standards, used in this work. Chapter 4 is showing the methods, which have been developed and used in this work. Grasp recognition related findings, test evaluations and energy consumptions are presented in chapter 6. Chapter 7 is summarizing the work and its results and is giving a short outlook.

1.2. Applications

Object detection is often used to infer the activity a person is doing. There is a wide spread of different possible applications for activity detection. Example usages of activity detection have been shown in previous works, described in section 2.4.

Activities of daily living (ADL) is a field in psychology and healthcare, which have been widely researched [6, 7, 8, 9]. ADL are defined as

"The things we normally do in daily living including any daily activity we perform for self-care such as feeding ourselves, bathing, dressing, grooming, work, homemaking, and leisure." [10]

The term instrumental ADL is describing further terms, which are not necessary for self keeping, but help to live in a community. Single actions in this field can vary. It is used to measure the ability in many disorders, basically in elderly care or psychiatric clinics.

Since following the daily routines of a person is time consuming, an automation of this process is helpful. Wearable sensors were recommended in previous works as a cheap way to track ADL [11, 12, 13].

Besides the field of reasoning of the activity of persons, it can also help in activities like stocktaking, where objects shall be noticed and documented. An example for a work which validates a wearable RFID reader in stocktaking is [14].

For assisting blind people, some applications of wearable RFID reader have been tested. An example is a shopping environment with tagged products for a text-to-speech applications [15].

Guiding workflows can also be a possible application. A concrete example is using several wearable sensors in a laboratory environment for capturing and guiding experiments [16].

1.3. Prototype Overlook

This work is mainly using off-the-shelf units. Three subsystems are coupled in the prototype over Bluetooth Low Energy (BLE). In a finalized system the subsystems are ought to be implemented in a single device, worn at the forearm.

Myo from Thalmic Labs [5] has been used as sensor. It is collecting EMG as well as IMU data (acceleration and angular velocity) from the forearm.

The grasp detection of the prototype is running on a separate computer. Matlab [17] has been used for the analysis. Connection is established over a serial connection with the BLE dongle BLED112 [18].

A third subsystem is the object detector. It is using the microcontroller RFduino [19] as control system and for establishing the connection between the subsystems.

1.3 Prototype Overlook

It is been connected with the RFID reader module Skyetek Skyemodule M1 Mini [20]. The RFID antenna is wrist worn and developed in a study about efficient wrist worn RFID readers [1].

On the software side, a logger for the sensor data has been developed, its main code is from Thalmic Labs. A connection tool Myo Connect from Thalmic Labs is also been used for communication between the logger and the sensor.

Grasp detection and a visualizer have been developed in Matlab code. Connection and logging is also possible via these scripts.

The microcontroller has been programmed in the typical Arduino [21, 22] environment. The driver for the RFID reader module is coming from the work WRIFD [23].

The machine learning baseline is tested via GRTool [24], based on GRT [25].

Table 1.1 is showing the locally separated subsystems with the including functions.

Table 1.1.: Three subsystems belonging to the prototype. The functions that are relevant for the user, are described in this table.

Forearm Sensor	Object Detector	Assisting Computer
Collect Acceleration Data	Scan RFID Tags	Grasp Detection
Collect Gyroscope Data		Log Data
Collect EMG Data		Control Forearm Sensor and Object Detector

2. Related Work

2.1. Gesture Recognition

Bulling et al. [26] have written a tutorial for activity recognition with body worn IMU sensors. The paper shows several complications that emits in general pattern recognition as well as in detecting of human activities.

General pattern recognition complications can be:

- Intra-class variance: Inconsistencies in a single class. Even a single test person can emit these complications (for example dependent on form on the day). It is recommended to classify a lot of variation for a single test person. The training data for general test sets should be based on several persons.
- Inter-class similarities: Several classes can emerge similar features. Often these classes can only get separated by collecting more sensor data or temporal reasoning.
- Null class separation: Many classification problems have a class, which is not related to an activity ("null class"). In these cases a huge amount of data should not relate to an activity. Modelling of the null class is still a popular research field, which is guessed to be hard or impossible. The reason is an unlimited amount of structures of this class. Actual implementations usually take a threshold from the certainty of the class or of the raw feature values.

Complications in human activity recognition can include following problems:

- Definition of diversity: A single activity can be performed in a lot of variants. It has to be recognized which sensor values are relevant for this activity.
- Class imbalance: Generally different activities can have an immense difference in needed time for its performance (e.g. sleeping and jumping). A single activity can take a different amount of time. Interruptions in a single activity are typical (like drinking a coffee). Also the frequency of occurrence has an influence on an imbalance.
- Annotation: Ground truth annotation can be challenging because data can be hard to interpret and the interpretation can be time demanding.
- Recording of data: A suitable experiment has to be found for the test case. A sufficient number of participants have to be available. Often the tests are time demanding because also the recording of null class values have to be planned as comparison data.

Applications can also be challenging:

- Sensor values can vary for inner and outer factors. Examples are sensor drift, different temperatures and slipping of the appliance.
- Data or ground truth annotation can be inaccurate.
- Possibility of latency while obtaining data.
- The processing power can be insufficient, especially in embedded systems.

Bulling is recommending an activity recognition chain. Inputs for the chain are streams of sensor data. Usually, the sensor data has to be preprocessed to filter out signal variability. The next step is a separation in sections, in which probably an activity occurred. A sliding window approach, an energy based segmentation or an additional sensor can be used for the determination of time windows. Features belonging to this activity are extracted in the next step. Ground truth labels are attached in training phase. Later, data is processed by the algorithm dependent inference. Chain performance should be calculated as last step in the design phase of the project.

Kim et al. [27] used a single channel EMG sensor for real-time hand gesture recognition. The sensor was positioned on the anterior side of the forearm, near to the wrist. Detect muscles have been Flexor Carpi Radialis, Palmaris Longus and Flexor Digitorum Superficialis, which are responsible for wrist and finger movements.

The classification was based on a combination of kNN and Bayes classification. The chosen features have been from the time and frequency domain. An adaptive thresholding method with individual parameters for every user was used. Four different hand gestures had to be detected by the classifier: Pressing a fist, fast movement to left, fast movement to right and circling of the wrist. The overall recognition accuracy of this method is 94 %. This result shows that even with very simple structures EMG can achieve a reasonable classification result for hand movements.

Xiong et al. [28] showed how to develop a Human Computer Interface based on EMG and IMU. In the example a computer cursor control has been developed. While the IMU was used for the movements of the cursor, a three channel EMG sensor is used for left and right clicks, as well as wheel scroll. The EMG sensors are dispersed over the forearm, placed over the Pronator Quadratus, Extensor Indicis and Flexor Digitorum Superficialis. A Linear Discriminant Analysis has been applied for reducing the feature set for the four gestures and the resting in the EMG data. For the movement a Kalman Filter is used.

Forbes [29] worked also on Mouse Human Computer Interface with EMG and IMU. Forbes recommended in his work to use the EMG sensor on a set of the muscles Extensor Carpi Radialis, Extensor Carpi Ulnaris, Flexor Carpi Ulnaris, Flexor Carpi Radialis and Flexor Digitorum Superficialis at positions with little crosstalk for his purpose. He used a Linear Discriminant Analysis for classification because the

classifier is intended to achieve a high accuracy and computational efficiency. As features he used zero-crossings, waveform length, slope sign changes as well as mean absolute values.

Haque et al. [30] did similar research. The main difference to [28, 29] is that Haque was focussing more on user interaction instead of gesture detection and recognition. They used the Thalmic Labs Myo, which is also used in this work. They found that filtering can help against problems which occur when the use of one sensor infects the second sensor (e.g. a detected movement when closing the hand). Moreover, they stated that accurate pointing interactions are achieving a good performance in combination with hand gesture recognition when using EMG and IMU together.

2.2. Grasp Modelling and Recognition

Taylor and Schwarz [31] showed in their work the anatomy of the human hand and its mechanics. Moreover, they described a typical hand resting posture.

A lot of research has been done in grasp modelling and recognition. The first step to it has been the detection of different grasp types and building a taxonomy, like shown in [32, 33, 34, 35, 36, 37, 38].

Schlesinger [32] was mentioning six grasp types in his work (cylindrical grasp, tip grasp, hook grasp, palmar grasp, spherical grasp and lateral grasp). The types had been chosen by the shape of the grasped object.

Napier [33] sorted the grasps into power grips and precision grips, which are not mutually exclusive, and stated that a lot of objects are shaped in a form that the user can handle it well in the tasks. The two types of grasps he mentioned would therefore be influenced by the size and shape of the objects, but also by some other factors as weight, texture, temperature, wetness of the object as well as emotions of the user. The main influence he saw was the activity.

Cutkosky and Wright [34] continued by determining sixteen grasps, ordered in their taxonomy shown in Figure 2.1. The taxonomy divides first into power grips (stability, security) and precision grips (dexterity, sensitivity). Starting with dividing by tasks, the taxonomy ends by the object geometry relation. However, the taxonomy is incomplete since not every possible grasp is included. Slight variations ("children") of the grasps are possible, due to personal preferences and size/strength of the hands.

Cutkosky [35] stated already in 1989, because of the development of the multifingered robots, a taxonomy of the grasps could be helpful to evaluate the robot models. These models should help to understand the underlying relationship between requirements and chosen grasps; a copy of the same human grasps for robots does not have to be necessary. An expert system "Grasp-Exp" has been developed in this work, which was asking participants questions and reasoned about the grasp they

would choose. About 100 questions shall be needed to predict the grasp a human would chose. The system proved that the human grasp decision can be predicted, and so it can also be modelled for robots. However, he stated that the taxonomy would be more useful as a designing aid for robot hands than for the results of the system.

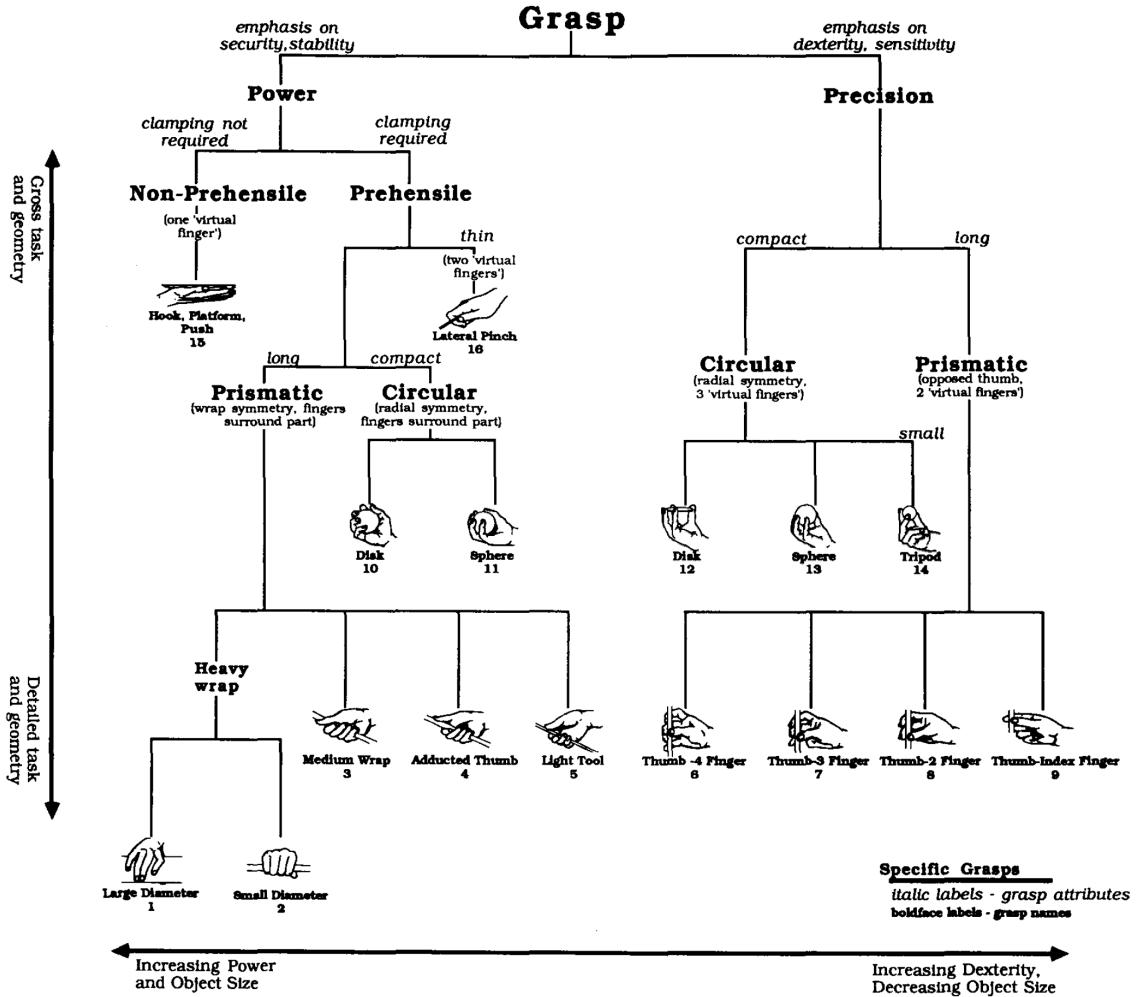


Figure 2.1.: Early grasp taxonomy by Cutkosky et al. [35]. The figure has been included with the permission of Mark R. Cutkosky. ©1989 IEEE

Feix et al. [37, 38] summarized the taxonomies of previous literature. They have defined the word grasp as following:

"A grasp is every static hand posture with which an object can be held securely with one hand, irrespective of the hand orientation."

That is a difference to the definition of grasp used in this thesis since the tests have not been static, objects have also been hold with both hands (moving the box) and the secure holding was not ensured (opening box). Their main orderings are

power/precision, palm/pad/side opposition and thumb position. Pad opposition can be described when grasping with the four medial fingers and the thumb near the pads, palm opposition as grasping between the palm and fingers and side opposition as holding between the sides of the two fingers. 147/211 grasp examples have been found in literature (differs in papers), 47 grasps have been classified. By their definition of grasps 33 different grasp types have been found, which could be arranged to result in 17 grasp when not concerning the object's geometry. The taxonomy is shown in Figure 2.2.

Opp: VF:	Power						Intermediate				Precision					
	Palm		Pad				Side		Pad			Side				
	3-5	2-5	2	2-3	2-4	2-5	2	3	3-4	2	2-3	2-4	2-5	3		
Thumb Abducted	 	 	 		 		 									
Thumb Adducted	 				 											

Figure 2.2.: Later grasp taxonomy by Feix et al. [38]. The figure has been included with the permission of Thomas Feix. ©2015 IEEE

Ekvall and Kragić [39] showed three methods for grasp recognition in a programming by demonstration framework. Two main differences of their study to this thesis exist. Their system knows that a grasp is being performed and it has to categorize the performed grasp. Also, they use a magnetic sensor at the fingertips and back of the hand as sensor. The first method used a hidden Markov model (HMM) to model the fingertip posture sequence in a grasping sequence. The second method relied in the hand trajectory and hand rotation. In the third method they combined the former methods. Their intention was also to detect the grasp before its sequence is completed.

The evaluation of the system is based on the knowledge that a grasp is performed and the correlation to one of ten grasps. The HMM method was able to detect eight of ten grasp types with 100 % for a single user. The hybrid model achieved 97 % for all grasps with a single user. In multi-user tests a recognition rate of 65 % has been reached with the hybrid model for all 10 grasps. For single users the hybrid method was already available to recognize 95 % of the grasps at 60 % completion.

2.3. Object Detection

Schmidt et al. [2] published early research on a wearable RFID reader for human computer interaction. They were detecting objects in the environment when handled by the user. Antennas were implemented into clothing, especially gloves, as small coils. An application triggering was also described in the work.

Fishkin et al. [3] was published a how-to for integrating of wearable RFID reader in a glove. He also showed how to integrate it into a wrist worn bracelet system as an alternative system, which is ought to be a less-obtrusive solution.

Berlin et al. [1] showed how to develop a RFID recognition system prototype for grasped object. It is the most important paper to this thesis since not only a lot of basics are described, but also the developed antenna was used and the described box test is comparable to the used testing method. The master thesis of Liu [40] is providing further information on the topics.

It was reasoned about the use of an additional IMU to detect further patterns. Concretely, known motions or postures could be detected by using the inertial sensors that help to give some more information about the performed task. As an example, it was mentioned that a hammering movement could be detected after grasping a hammer. Also a possible sleep mode of the system while no accelerations are registered is mentioned.

With a tuning of the 7.5 cm x 11 cm oval, wrist-worn antenna the RFID reading range could be increased, by increasing the Q-value. This increases the power output and therefore the range, but conflicts with the bandpass characteristic of the reader. The resulting reading range of their developed antenna was up to 14 cm.

Reading frequency is also been optimized in this work. This is described as the essential power consumption versus detection trade-off. The antenna and RFID reader unit was described with 60 mA while reading, 15 mA in idle mode, and 60 μ A in sleep state. One reading needed 20 ms to 68.4 ms, depending on tag type and whether the reading is successful. With one reading per second it was possible to decrease the average current consumption to approximately 18.23 mA. Battery lifetime of a 600 mAh was described with 8.5 h in the highest possible frequency of 16 Hz versus 28 h in 1 Hz.

Three different testing methods have been used for evaluation. The box test was similar to the testing methods in this thesis, used for receiving the optimal reading

range and frequency. The hit rates started with 100 % for 16 Hz and dropped to 65 % for 1 Hz with a reasonable Q-value. A scripted gardening task has been tested for the interaction with tools and objects in a real life task at 1 Hz. As last testing method, a test person performed eleven household tasks for three continuous days.

An approach to use wearable in-hand object detection to interact with the environment was shown by Wolf et al. [4], who introduced an interaction device with a wearable object pick-up detection system. The work sought to switch devices from standby to active mode when taking into hand instead of a manual activation technique. The system uses a combination of a RFID ring with an embedded gyroscope ("PickRing") and the gyroscope data of the devices. Gyroscope data is sent via Bluetooth to all coupled devices and is compared with their gyroscope data. The whole wearable part of the system, with microcontroller and Bluetooth active, has a reported run-time of 15 hours with a 9 V battery.

2.4. Activity Detection

Philipose et al. [12] demonstrated the possibility of detecting activities of daily living with wearable object detection. A glove based RFID reader with a sampling rate of 2 Hz was used, which lasts for 2 hours. They collected a series of tags, corresponding to an activity, and calculated the probability that one activity is executed. Besides modelling errors, they also identified sensor errors (missed tags) as a source of ambiguity. A recall of 73 % with a precision of 88 % was achieved. They also reported a recall of 33 % for their worst recorded activity. As a reason for missed detections, they mentioned the absorption of the RFID signals by the environment.

Patterson et al. [41] show in a similar setup for fine-grained daily activity recognition that incorporating the knowledge of aggregate objects leads not only to good accuracy but also requires less training data to learn the activities' models. They demonstrated a method to also detect interleaved and interrupted activities, with shared objects in the single activities. The used models have been learned automatically. Abstraction smoothing has been used with a distinguish of object instances from object classes, while learning dependencies on aggregate features.

An approach for recognizing ADL and care activities with RFID and inertial sensors was published by Hein et al. [11]. Because of the combination of both technologies, they achieved information about which object is being held and if it is being used.

They described three different types of application domains:

- Activity profiling: How often and how long is an activity performed?
- Behaviour assessment: Are there any temporal abnormal behaviours or long term variations in daily routine?
- Proactive assistance: What is the intention of the user?

They used 562 different features from a half overlapping 1.28 s sliding window. An embedded Weka C4.5 decision tree in a HMM with one hidden state was used for processing.

As test sets, a four state breakfast environment and an eight state home care environment has been simulated. One result was that in case of ADL temporal data worked much better than classifying the single states without the HMM model. Moreover, the IMU gave better results than the RFID alone. For combination of both sensors in the HMM they achieved 97.8 % for a coarse breakfast state classification, 76 % for a fine breakfast state classification and 85.1 % for the care environment.

An ADL recognition approach with a wrist worn camera, microphone and accelerometer was shown by Maekawa et al. [13]. They propose some methods for annotating data in long time testing. The classes used as ADL have been designed for the single users (like a specific vacuum cleaner sound). The first method is also used in this thesis: The labels have been made via camera recordings. A second method they proposed was via annotation on a PDA. As third method they proposed a voice timing over a headset.

Because of privacy concerns, the microphone and video data is not transmitted as raw, but as abstract data. Moreover, a battery is emptied soon when transmitting raw data. Colour histograms have been chosen as visual features. Typical computer vision problems occur when using the camera. Mel frequency cepstral coefficients (MFCC) [42] have been used for extracting sound features. As acceleration features the mean, energy, frequency-domain entropy and dominant frequency have been chosen. Moreover, illuminance and compass data have been additional features.

Fifteen ADL have been chosen. AdaBoost M1 with decision stumps as well as C4.5 decision tree have been used as binary classifiers. A HMM is used for sequencing. Depending on the environment and classificator, on average their results have been between 52.0 % and 87.3 % precision and between 51.8 % and 84.8 % recall, while recall and precision have risen simultaneous.

An approach for detecting objects via the electromagnetic noise was published by Laput et al. [43]. Objects in this work have to be physically touched, the body is functioning as antenna. An advantage to RFID is that the objects do not have to be prepared (tagged). For robustness a background noise model have been designed. A higher rate in object noise changes towards the background noise is getting abused. Objects held for a while are integrated in the noise profile, which can be seen on the release of the object. They reported an accuracy of 97.9 % with a single trained SVM model of 23 objects plus null class (no object touched). The majority of the objects received 100 %.

2.5. Deep Layer Muscle Surface Electromyography

McGill et al. [44] tested the sensing of deep muscle activity via surface EMG. They found that it is possible for some muscles to detect an activity. However, they also stated that the noise level can be high towards the signal.

Koshio et al. [45] showed how to identify surface and deep layer muscles activity by surface EMG. They stated that it is hard to identify if a signal is emerged by the deep layer or superficial muscle. In their test environment, they used an electrode array and concentrated on rectangular overlapped muscles. They found that they are able to identify the muscle by the direction of the propagation. However, the muscles have to lie in different orientations and this method is not able to completely identify each muscle activity.

3. Key Concepts

This chapter enumerates and introduces several key concepts that this work strongly depends on: Apart from the use of the electromyography and inertial measurement modalities that are used for detecting grasps and the use of RFID for identifying the grasped objects, the BLE protocol and Random Forest classifier are explained.

3.1. Electromyography

EMG is a technique to detect activity of skeletal muscles via the electrical potential they produce. Skeletal muscles are one of the three main groups of muscles and describe the group that is responsible for an active controlled movement of the body. EMG is mainly used in medical diagnostic of neuromuscular diseases, rehabilitation, disorders of motor controls and in studying of kinesiology (e.g. in sport science). Also some prostheses work with EMG as indicator for a desired movement.

The record of EMG is also called electromyogram, but further referred as EMG data, and the sensor for collecting EMG data is also known as electromyograph, but further referred as EMG sensor.

The origin of the EMG signals is found in the activation of muscle contraction. Motor neurons are nerve cells, which are responsible for the activation of muscle activity. A single motor neuron can innervate one to several thousand muscle fibres, however, a muscle fibre is only controlled by a single motor neuron. In human arms 1 motor neuron to 5000 muscle fibres is a good ratio. The combination of motor neuron and all innervated muscle fibres is called motor unit. If a motor neuron is firing, which means it sends an activation to the innervated muscle fibres, an action potential is going lengthwise through the muscle fibres. If the contraction shall be kept up, the motor neuron has to fire several times, which forces a motor unit action potential (MUAP) train inside the fibre. The strength of the contraction is depending on the number of motor neurons firing, the number of fibres they innervate and the frequency of firing.

An action potential results from a rapid change in the permeability of the membrane to $\text{Na}^{2+}/\text{K}^+$. In the first step chemical messengers from the motor neuron binds on the receptors of the muscle fibre, which forces them to open and start a depolarisation. Hereby, the chemical Na^{2+} channels open and allow to decrease the voltage difference between the muscle fibre and the outside to rise from ~ -70 mV. At some

threshold (~ -55 mV) also the voltage driven Na^{2+} channels open and allow to rise the potential faster. The Na^{2+} inside the fibres forces the muscle to contract. At $\sim +30$ mV the Na^{2+} channels close and the repolarisation begins. In this step K^+ is driven outside the fibre to get back to the normal potential again. Moreover, an ATP driven ion-pump extracts the Na^{2+} outside of the fibre and the K^+ inside the fibre again to allow the next action potential. The process is illustrated in Figure 3.1

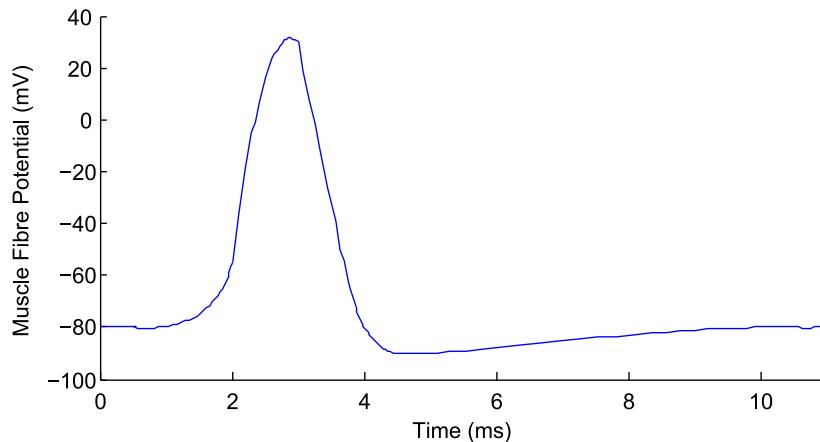


Figure 3.1.: Temporal evolvement of a motor unit action potential. It starts with a rise of the potential (depolarisation), which is followed by a repolarisation and a hyperpolarisation to regain the rest potential.

This action potential has a size of 1 mm^2 to 3 mm^2 . Its normal duration is 5 ms to 20 ms [46]. It moves alongside the muscle with $2 \frac{\text{m}}{\text{s}}$ to $6 \frac{\text{m}}{\text{s}}$ and stimulates the neighbour cells to behave like the reacting cells. Typical voltages recorded via EMG are between $50 \mu\text{V}$ and 20 mV .

When placing two different electrodes (A and B) on two positions alongside the muscle fibre, the following sequence of potential can be seen in the electromyogram [47]:

- While the muscle is not activated, the difference between the muscle is zero ($A - B = 0$).
- Muscle is activated and action potential passes A, but has not reached B yet, a spike can be seen in the difference ($A - B > 0$).
- Action potential reaches B while still in A, the action potential goes back to zero ($A - B \sim 0$).
- Action potential has passed A, but still in B, a negative spike can be seen ($A - B < 0$).
- Action potential has passed B, the difference goes back to zero ($A - B = 0$).

3.1 Electromyography

The sequence is also shown in figure Figure 3.2. It is to mention here that in case of surface electrodes a lot of fibres can be seen and because of the MUAP train several action potentials passes the fibres.

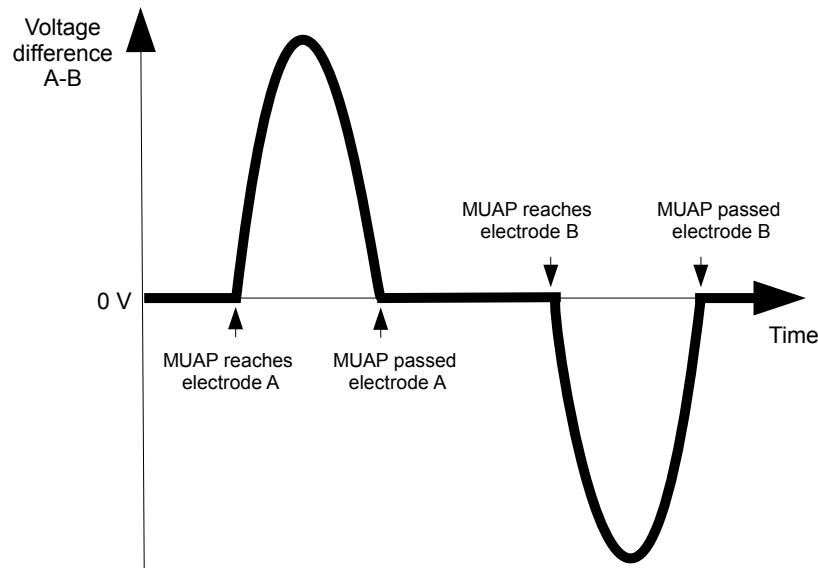


Figure 3.2.: Detected voltage difference in a two electrode EMG sensor by concerning a single MUAP. Positive peak is reached when the MUAP covers electrode A. Second peak is reached when the MUAP covers electrode B. While the MUAP is outside of the electrodes the difference is zero.

Two different EMG types are in use: Surface and intramuscular. Surface types place the surface electrode on the skin above the muscle belly. Surface electrodes detect mainly the superficial muscles and have problems in detecting the deeper muscles. Signal strength of surface EMG depends on different factors like body fat, hairs, skin type, muscle fatigue or dead cells [48]. Crosstalk is expected in places with a high amount of different muscles. A preparation of the skin can lead to better results. The skin also performs as a low pass filter.

Intramuscular types use fixed needle or flexible fine-wire electrodes to pierce into muscles for a more accurate signal. Because of the movement of the muscles flexible electrodes are preferred. Single motor units or even single muscle fibres can be detected via these electrodes. Deeper muscles can be accessed with these electrodes. This work concentrates on the surface EMG since in daily worn system a piercing of the muscles is not realistic.

3.2. Inertial Measurement Unit

IMU are devices to measure acceleration (accelerometers) and angular velocity (gyroscopes) in combined 6 degrees of freedom (DOF). Moreover, some IMU can measure the magnetic field (magnetometer) for more accurate results. Their main fields are robotics and flight navigation. In this work they are used to detect the movements of the forearm with speed and direction. Mainly the accelerometer is used. A gyroscope is also delivering suitable results, however, the resulting features are very similar to the accelerometer features.

In an IMU, three accelerometers are ordered orthogonal to detect the acceleration into the x-, y- and z-axis. Conceptual accelerometers are behaving like a damped mass on a spring system. Their measurement unit is typically $\frac{\text{m}}{\text{s}^2}$ or $g = 9.81 \frac{\text{m}}{\text{s}^2}$. Accelerometer data is also influenced by the gravity. The gravity can be filtered with a high-pass filter. This filtered signal is in the following called "linear acceleration". Accelerometers are very energy efficient with a typical value below 1 mA at 3 V.

Gyroscopes are used to measure the angular velocity in the x-, y- and z-axis. Their typical measurement unit is $^\circ \text{s}^{-1}$ or $\frac{\text{rad}}{\text{s}}$. The typical energy demand in systems like the implemented is around 4 mA at 3 V and worse than accelerometers. Also their reaction time is typically slower compared with accelerometers.

Magnetometers are devices to measure the magnetic field. The measurement unit of magnetometers is Tesla. Energy consumption can be below 1 mA at 3 V. A magnetometer has not been used in this work.

3.3. Radio Frequency Identification

Radio Frequency Identification (RFID) is a technology which allows an identification of tagged objects from a short ranged distance. Therefore, a transponder (also called reader) is requesting a receiving system (tag) to send out its information. It is an alternative to bar codes which is easier to read since it does not require a direct line of sight and no knowledge of the exact position of the tag. However, the tags are more expensive compared to printed bar codes.

Tags can contain an own energy source (active) or be powered by the reader (passive). This work concentrates on passive tags, which are also known as transponders. Passive tags are built with following three main parts. The antenna is responsible for the collection of the inducted energy as well as for sending the information, stored in the tag. A semiconductor chip regulates the processes. The encapsulation is protecting the tag against destruction from outside. Passive tags can be powered in near field, but also far field can deliver enough energy (10 μW to 1 mW) [49].

Transponders are called readers because they are responsible to access and process the information of the tags. They also provide energy for the passive tags via a high frequent electromagnetic field.

RFID serve a wide field of application. The main applications are identification of objects, theft protection in shops as well as access control. Implantable tags are in use for animal identification. Variations of RFID techniques are in frequency, reading range, reading latency, size of tags and cost per tags. The technique received a lot of critic because of concerns about informational self-determination.

3.4. Bluetooth Low Energy

Bluetooth Low Energy is a part of the Bluetooth 4.0 standard, developed by the Bluetooth Special Interest Group. It is a piconet, single hop solution. The trade-off in BLE is in the energy consumption, latency, piconet size and throughput. The throughput depends on the connection interval and connection slave latency. The main reference to this section is [50].

The BLE protocol stack is seen in Figure 3.3. It is usually divided into the main controller and host. The controller layers are usually implemented on a separated System-on-a-Chip with an integrated radio, while the host layers are implemented on the application processor. The Host Controller Interface (HCI) establishes the communication between host and controller.

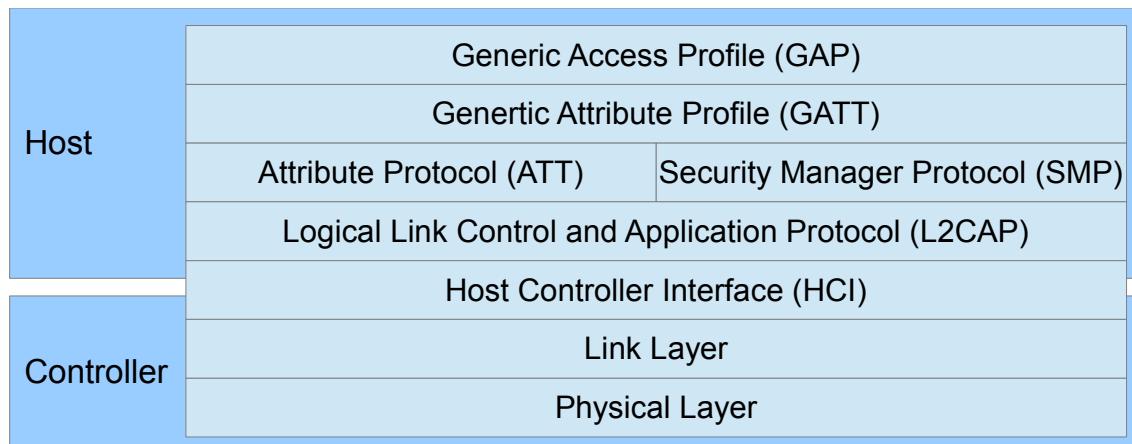


Figure 3.3.: Protocol stack of Bluetooth Low Energy. Host and Controller are typically implemented on several chips.

In the physical layer BLE operates in the Industrial Scientific Medical (2.4 GHz) band. It is divided into 40 radio frequencies. The two types of these channels are advertising channels and data channels. The three advertising channels are used for device discovery, connection establishment and broadcast transmission. The remaining 37 data channels are used for the bidirectional communication between the devices. A Gaussian Frequency Shift Keying modulation is used. The data rate is 1 Mbps. The possible range is determined by the signal strength and receiver sensitivity and typically a two-digit number of meters. An adaptive frequency hopping

algorithm is used to avoid band conflicts with other systems sending on the frequencies. The hopping algorithm selects 1 of the 37 data channels for communication for a given time.

In link layer, two different communication methods are used: Advertiser/scanner as well as master/slave. Advertiser are devices that broadcast data in the advertisement channels. Scanner only listen to those broadcasts. Scanner can also try to connect to these advertisers. In a connection the scanner becomes the master and the advertiser a slave. Only one device can be connected to a slave. In other words a BLE piconet is a star topology, where the master is the center and the slaves become the endpoints. Time Division Multiple Access is used for sending and listening to channels. While establishing the connection, masters inform the slaves about the channels and time periods the slaves have to listen. These parameters can also be upgraded afterwards. Slaves should stay in sleep mode while they are not sending or while they are not forced to listen to the channel. Parameters for the frequency hopping algorithm are sent by the master to the slaves. A connection event between a master and a slave is an exchange of packets between the devices. Between two packets, an Inter Frame Space of 150 μ s must pass. A connection event ends when both devices send the information that they do not have more data in a separated bit of the packet, the bit error rate is too high, the packet header is corrupted or the declared connection supervision timeout (100 ms to 32 s) has been reached. The time between two consecutive connection events is defined with the connection interval parameter (7.5 ms till 4 s, multiple of 1.25 ms). A defined connection slave latency allows the slave to stop listening to the master for 0 till 499 connection events. Flow control is achieved via a stop and wait mechanism with an one bit sequence number and an one bit next expected sequence number in the header.

Logical Link Control and Application Protocol (L2CAP) is used to multiplex the data of three higher layer protocols, ATT, SMP and Link Layer control. Upper layers data must be below the L2CAP maximum payload size of 23 bytes.

Independent of the master/slave definition, there is also a server/client assignment, maintained in the GATT layer. Attributes are data structured, holding information managed in the GATT layer. The ATT layer defines how the devices communicate with each other. Clients request the servers attributes or send commands to the server to change its attribute values. Server answer with unconfirmed notifications or indications, which has to be confirmed by the client.

Generic Attribute Profile (GATT) is structuring the memory hierarchical. One or more services are defined in the top layer, subsuming one or more related characteristics. Each characteristic can be addressed with an own handle and is storing a value and properties.

The Security Manager Protocol (SMP) controls the security services for protecting the information exchange between two connected devices. Various techniques are used.

Generic Access Profile (GAP) is defining the interoperation between devices. A role can be broadcaster (non-connectable advertiser), observer (non-connectable scanner), center (manage connections) or peripheral (single connections to center). Most device support multiple roles, but only one role can be active at a time. An application profile defines the interoperation of devices.

3.5. Random Forest Classification

The Random Forests classifier is an extension to the Classification and Regression Tree [51]. It had been developed by Ho [52] and further investigated by Breiman [53]. Besides classification, it can also be used for regression.

Classical trees typically have problems with generalization at high complexity. Typical method for increasing this complexity is pruning, however, on the cost of accuracy on training data. The Random Forest overcomes this problem and is intended for high dimensional data with more features than classes.

A random forest model is containing multiple trees. The trees are getting generated randomly by drawing learning samples instead of using the whole dataset.

Decision in classification is made for the class, which have been chosen by the most trees. In regression the result is the mean of all trees.

4. System Concepts

4.1. Steps for a Grasping Sequence

4.1.1. Grasp Steps Basics

A grasp can typically be divided into different steps, with each of the steps invoking different features. The collection of features for a step is typically detected within a short time interval, but steps can be spread in time. For this situation, it can be expected that features before a grasp are further away from the actual grasp than features for the holding step.

The used steps are found within this work. Reason for choosing especially these steps is the time separation of emerging features. An overlap is possible. Similar per step data is explainable by the stop/move actions of the arm and wrist before and while grasping as well as by different muscle zones of opening and closing the hand.

Four different steps are used in this work. Figure 4.1 shows the used steps, ordered by time. Picture 1 and 6 show a posture before and after the grasp. Data associated to this step should not be detected as part of a grasp, and is thus part of the null class. Null class consists of resting as well as moving without any grasp intention.

4.1.2. Moving Hand to Object

Before each grasp, a movement of the hand towards the object is common. This movement takes place before the user touches the object. In normal behaviour, this step starts around 600 ms before the grasp and has a smooth transition into the next step. It is a strong indicator for a grasp and it can cause a miss if a restrictive detector misses this step, for example at a long intermission before the next step. However, this intermission is in most cases no natural behaviour. Furthermore, if the feature detection is weak enough, there should still be a detection of this step.

This step can use motion features for changing the position of the lower arm, a directed movement of the arm and hands towards the object and, very typical, also a rotation of the hand. The EMG features normally representing the opening of the hand and a slight pulling back of the back of the hand.

It can happen that a lot of features of this step can be detected in other movements belonging to null.

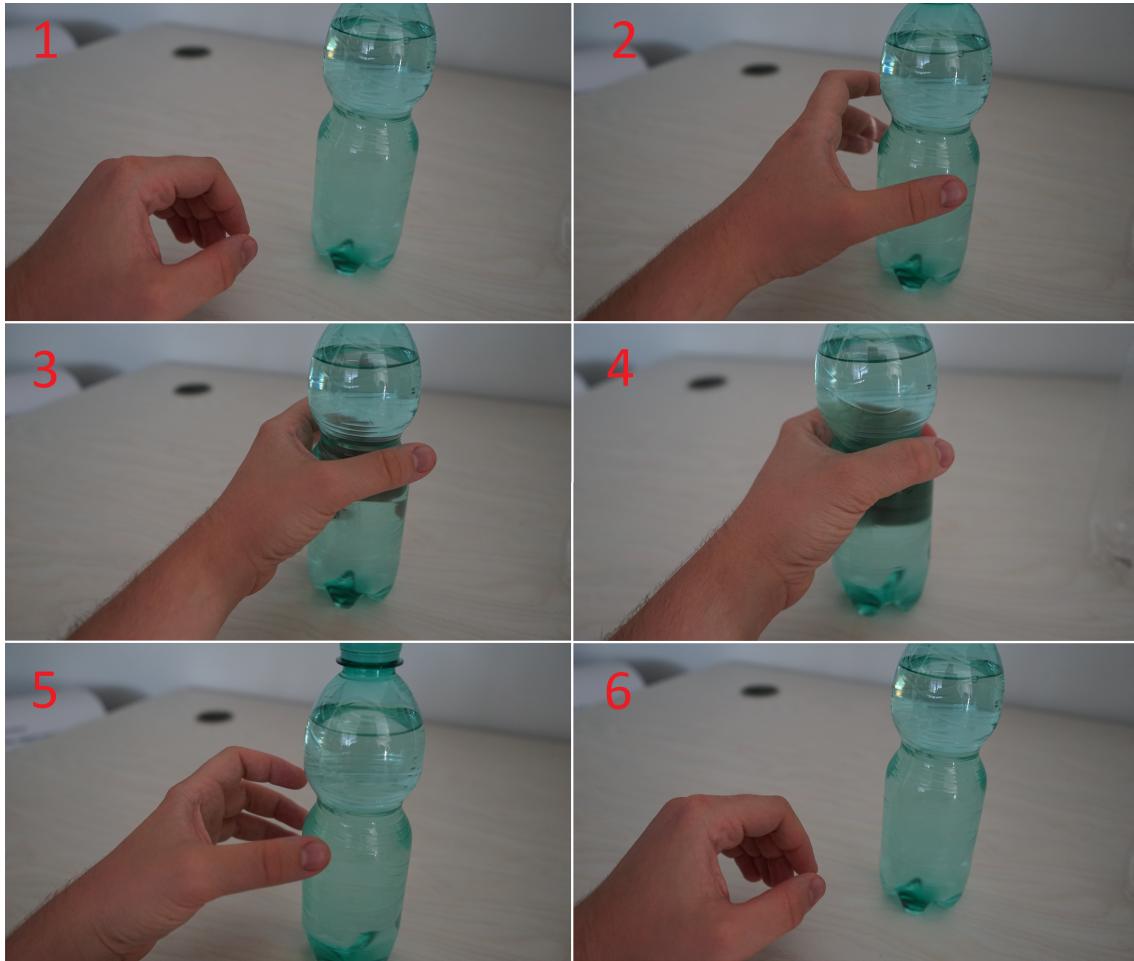


Figure 4.1.: Illustration of a complete grasping sequence. Step representation: (1) posture before a grasp intention (null), (2) Moving Hand to Object, (3) Closing Hand around Object, (4) Holding Object, (5) Releasing Object, (6) posture after a grasp (null).

4.1.3. Closing Hand around Object

A closing of the hand around the object is the second used step. In normal behaviour it is a very short time period, close to the neighbouring steps. Features to be detected here are short peaks in the data.

This step often tends to contain strong accelerometer signals, going in the opposite direction of the previous movement by slowing down of the hand or the colliding of hand and object. Specific muscle activation for a flexion of fingers and wrist can also be detected here.

4.1.4. Holding Object

The object detection should be activated in a step when the user is holding the object. Movements in this step are pretty arbitrary, hence the IMU signals are useless for a detection here. This is the main reason a detection with moderate false positives based on IMU alone is hard.

This step can be detected by continuous strong signals coming from muscular activity used for closing the hands. Lightweight and fragile objects can be problematic to detect in this step since no strong grasp is necessary or possible.

Even for short grasps, a minimum time around 650 ms for this step was found in the tests. This is used as time constraint for detecting a grasp and finishing the object scan.

4.1.5. Releasing Object

Releasing the object is the last step in the sequence. It includes the opening of the hand and moving it away from the object. It is fluently going into a new null class. The features invoked are similar to the step "Moving Hand to Object".

4.2. Grasping Behaviours

Some typical behaviours have been found in the tests. These behaviours can be used in the detection of grasps. Also, cases, in which these behaviours can cause errors, can be prevented by reasoning about them.

Most important is the fact that the grasps described in 2.2 are not equally often used. Some grasps are such seldom that they did not even occur in most test situations. Some of the seldom grasps are varying their features heavily from the more often grasps that they can be problematic. Handling these grasps will increase the false positive rate. Therefore, the system would be more unstable because of even weaker learners and a more complicated weighting in boosting. Ignoring seldom grasps would increase the false negative rate. How much the false negative rate increases is strongly dependent on the user's grasp preferences.

As acceleration thresholds some fixed values have been used, which worked well with the test participants. However, slower moving participants could be problematic in the actual configuration. Unfortunately, no elder or handicapped people have been found as test participants.

Body posture had some influence on the behaviours. Standing participants were typically faster than sitting participants. Also the trajectories have been different.

Feedback from the system has fatal influence on the using behaviour. Participants typically increasing their strength of the grasp until a feedback is achieved. Movements are typically slower to observe the point in time when they achieve a feedback. Also they increase the grasping strength slower. In tests it is advisable to disable every feedback to achieve a natural behaviour.

By performing the same type of grasp twice, by the same person with the same body position, there will be a variation in the resulting EMG data [48]. The random overlapping of the single MUAP in the muscle is only a small influence compared to the human variation of even very often performed and standardized movements. In EMG these influences are even higher because of continuous balancing of agonists, antagonists and synergists.

4.3. Grasp Detection Model

4.3.1. Basics of the Grasp Detection Model

For the detection of the grasps, a layer model has been implemented to decrease the level of abstraction from a detection in the whole system to a detection in smaller subsystems. Three layers are responsible for different tasks. Detections in the lower layers can abort further processing in later layers and shut down an unnecessary sensor sampling and calculating.

Figure 4.2 shows a possible implementation of the layer model to reduce the energy consumption. It illustrates the enabling of sensor checks, controlled by later layers, on the basis of earlier layer sensor data.

4.3.2. Model Layer Functions

4.3.2.1. Layer 1: Base Activity Recognition

In Layer 1, a detection of the persons long-time activity is performed. These activities are not directly linked to grasping, but can make a strong assumption of the probability of grasps. As an example some activities can rule out the possibility of a grasp without a change in the signals. In this context, a long-time activity should be larger than one second until several hours.

An instant entering in a state where no further grasp detections are performed is not that important. In the time where these states are not detected, only a little amount of energy for further processing and eventually not disabling of sensor sampling is used. However, the main power consumer are the object detectors, which should be disabled in the later layers. The re-entering in a state which allows a grasp detection should be fast because the missing of an end could lead to a missed grasp.

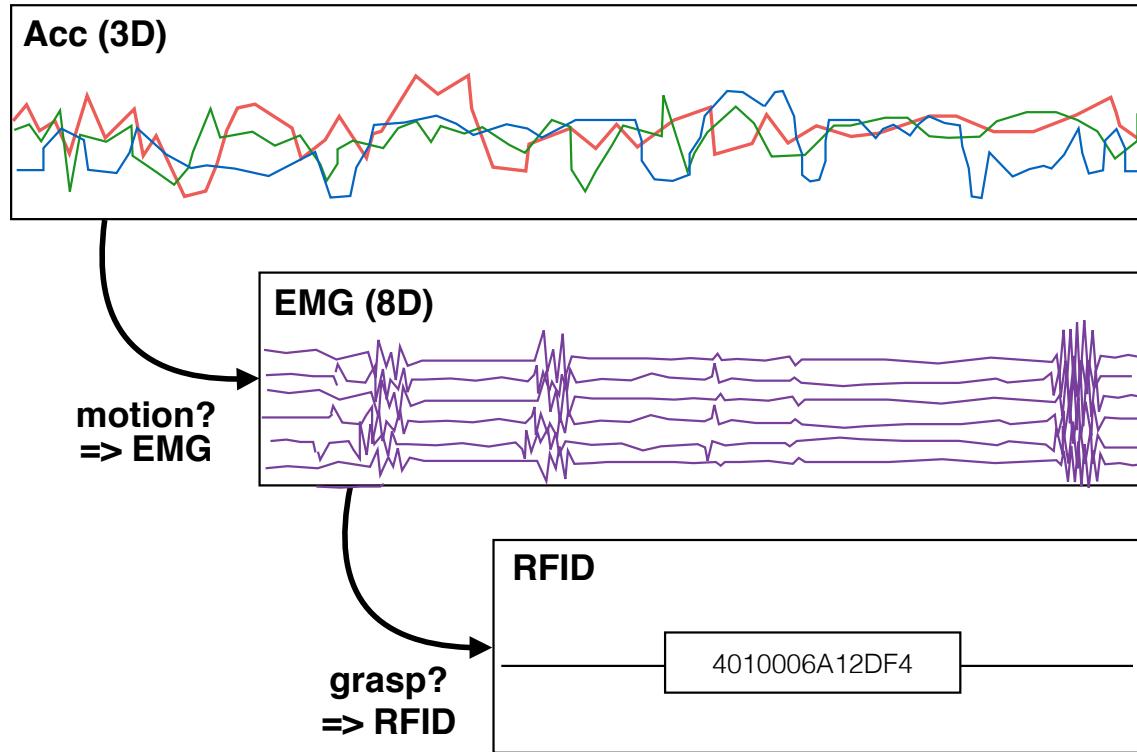


Figure 4.2.: Approach to use the three layer grasp detection model. Layer 1 is based only on acceleration data, which is sparse in contrast to the other sensors. On basis of the results in Layer 1 further checks in Layer 2 and Layer 3 may be necessary, which are based on the more costly EMG data. If a grasp has been found in processing the whole grasp detection model, the most costly object detector is enabled.

4.3.2.2. Layer 2: Grasp Segments Recognition

Features, which are directly connected to a grasp, are detected in Layer 2. To detect short grasps as well, a high reactivity of this layer is necessary. Tasks detected in that layer can be intra-second until a few seconds. This implies that the detection of the features should be within a few hundred milliseconds.

In Layer 1 and 2 all necessary sensor data is collected and the calculation of features is made. However, a wide variety of grasps and movements similar to grasps have the consequence that single features by themselves tend to be rather weak. For this reason a further processing is necessary.

4.3.2.3. Layer 3: Final Grasp Evaluation

Layer 3 is ought to keep the precision high by checking for the chronological sequence of the detected features. This higher precision allows to collect a lot of weak features

in Layer 2. Further, the high number of weak features allows to detect a wide variation of grasps to achieve a high recall under real operation conditions.

4.3.3. Model Layer Realisations

4.3.3.1. Layer 1: Base Activities Features

Three different activities have been implemented:

- Resting (no further processing)
- Walking (no further processing)
- Working (grasp detection active)

Resting was declared as periods of time with nearly zero movements. Since the arm is not moving, the probability is close to zero that a grasp is performed and the grasp detection can be disabled. This state was easily detected by checking the linear acceleration of all three sensors. If the mean of the absolute values are below 0.01 g, it is seen as resting. The resting class had a delay by enabling as it was not meant to be enabled while resting the arm before a grasp on the object. Because of minimal movements this time could have been still below one second.

Walking was found as another class because at least a short stop or change in the ordinary walking move of the user before a grasp is typical. The features of the walking activity were found in the frequency domain since a walk includes a strong swinging of the arm with a low frequency. For the acceleration in x-direction (seen by the detector) the mean of the value of the lowest cepstral coefficient had to be over a threshold and also the mean of the spectral energy was found to be over a threshold. In the y-direction a continuous higher spectral entropy was found which can be detected by its mean. For the z-direction the lowest cepstral coefficient was checked for an increased mean.

All activities which have not been detected as resting or walking are further called as working. Working means that the wearer is principally ready to grasp and further processing in later layers is needed.

Resting and walking could be also called as sleep modes since it disables further processing and EMG sampling.

4.3.3.2. Layer 2: Grasping Features

After working has been detected in Layer 1, the model is looking for grasps.

The EMG data was principally much more accurate than the IMU data because the IMU data always had the trade-off of a high recall to a very low precision and a recall that misses a few grasps to a moderate precision. Since the recall should be

near to 1, the detection via IMU data alone set the state to found grasps in two of three cases to true, while not resting or walking. A combination of both sensors can create a powerful detector.

The first feature found in the IMU data is a typical roll movement of the arm that the palm is pointing towards the body (right arm → clockwise movement). This feature was best found via gyroscope, where the maximal rotation should be detected. However, also the accelerometer can be used for this feature since it shows similar values. The accelerometer typically use less energy than the gyroscope and it was the only used gyroscope feature.

The second feature is the change in the position of the arm. Here it should be checked for the maximum of sum of acceleration x and y, as the gravity factor is a clear indicator.

The third and last used feature is a frontal pushing of the arm. This is found as the maximum of the linear acceleration x.

However, all the acceleration features found for Layer 2 can be emerged by a lot of other circumstances, for example by stopping the arm after a move in the opposite direction. Furthermore, because the maximal values are used, the speeds of the arm movement can differ under a lot of circumstances, even by the same person. Examples for the causes are whether the person is standing, sitting or lying and whether the person is tired or in an active phase.

The strongest detected EMG signals are from moving the wrist, but also some signals from fingers and elbow were received. The allocation of the sensors on the arm is shown in Figure 4.5. Some sensors can detect different muscle activities. Therefore, an activation in a sensor can have different reasons. Section 4.5.2 lists the possible movements which can be detected by the forearm EMG.

4.3.3.3. Layer 3: Sequencing of the Features

Typically, a grasp is following the steps described in section 4.1. An expected approach would be a state machine or a HMM. A clear detection of the states would have to be possible for an implementation of such an approach. However, there is no clear border between these steps. As an example, the features for the step "Closing Hand around Object" are usually just a very short peak which is very close to the neighbour steps. Moreover, a clear distinction of the states would increase the frequency of the feature calculation, which had the effect of an increased processor workload and eventually higher sensor sampling rates. Consequently, the energy consumption would increase. For this reason, it is easier to flag a detection of a feature for a period of time.

The step "Releasing Object" has no direct use to activate the object detector. However, it contains valuable information about the ending of the grasp. An example for the usage of this information is a re-enabling of the object detection after it was shut down by a successful detection.

4.4. Prototype

4.4.1. Prototype Overview and Connection

Figure 4.3 shows the body-worn devices of the prototype. IMU and EMG are embedded in the forearm sensor. The object detector is including a microcontroller with a BLE module for the device communication as well as the RFID components (reader and antenna).

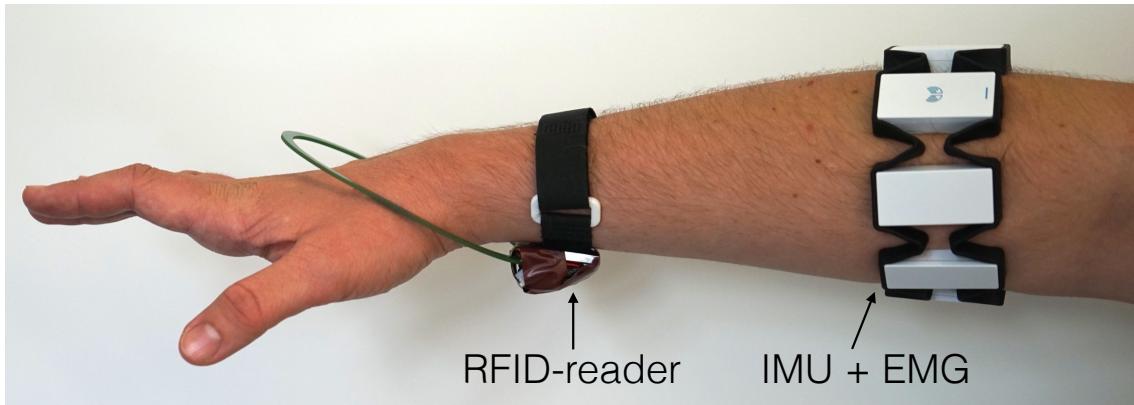


Figure 4.3.: Body-worn prototype hardware components. A commercial EMG and IMU sensor (Myo from Thalmic Labs) collects the data of the muscles in the arm as well as the movements of the arm. The RFID reader is responsible for the reading of RFID tags and is establishing an object detection by means of the information attached to the tag. Both components are in the prototype connected via BLE to a grasp detection system on a separate, assisting computer.

The device communication has been established via Bluetooth Low Energy. Figure 4.4 shows a systematic structure of this connection. Grasp detection is running on an assisting computer. It is communicating over a serial interface with the BLE dongle. The commands between the grasp detection and the BLE dongle are structured as HCI packets. Section 4.4.5 explains the BLE connection between the devices, 4.7.5 explains what is been sent and Appendix B gives an overview about the most important commands and their structure. The forearm sensor is accepting and sending predefined packets. The object detector include a sending and receiving in its microcontroller library.

The connection between the object detector microcontroller and the RFID reader module is established by a serial port configuration. On microcontroller side it uses two general purpose input/output (GPIO) pins and on the RFID reader module side the Tx/Rx pins.

The antenna is attached to the antenna pin and to the ground pin of the RFID reader module. RFID reader module's antenna matching circuit has been changed.

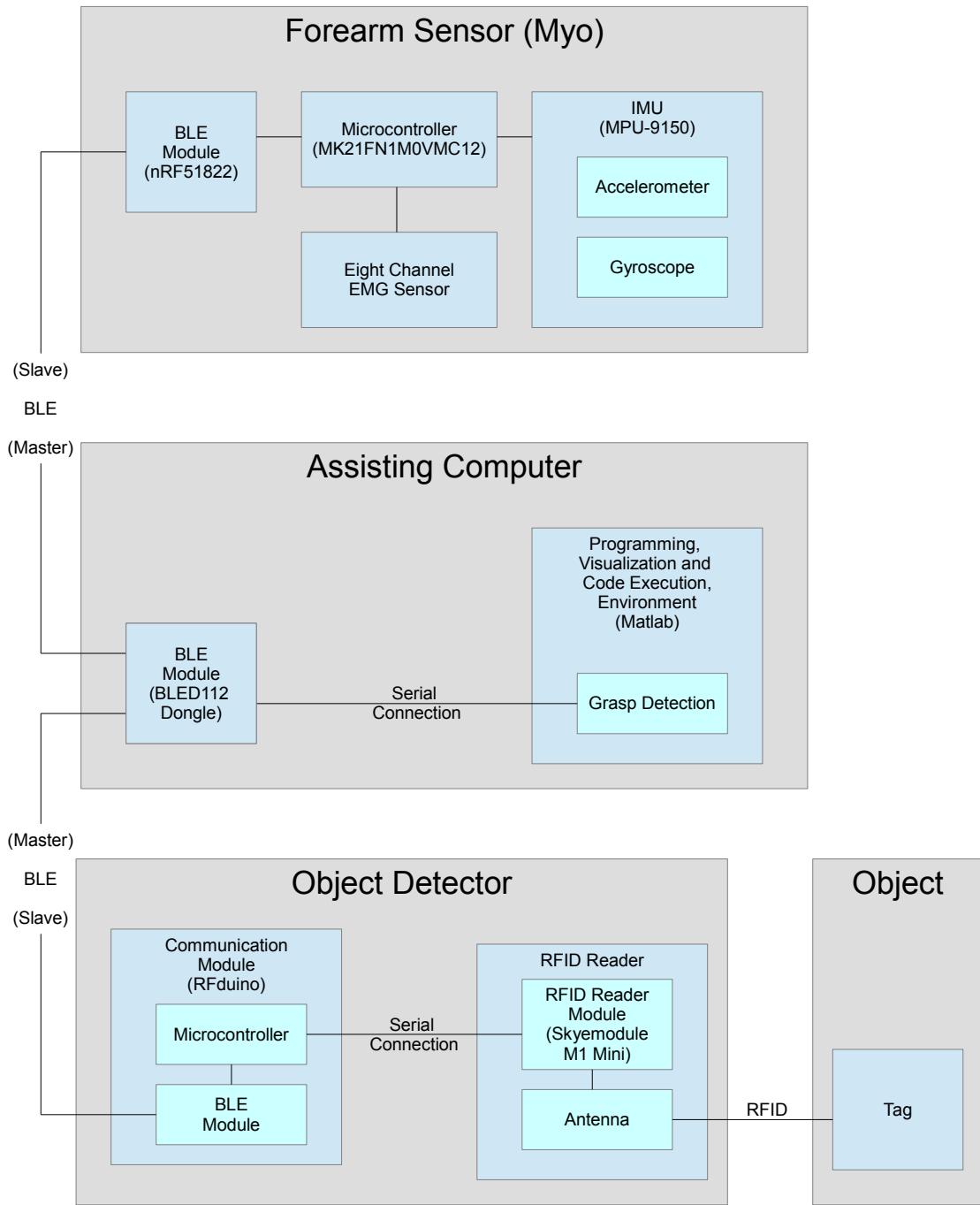


Figure 4.4.: Prototype connection of the subsystems. The forearm sensor is collecting acceleration and gyroscope data. This sensor data is sent to the grasp detection, running on an extra computer which is assisting the prototype. Detected grasps are enabling the object detection. The object detector is constructed via a communication and controlling module and a RFID reader module. Object mounted tags are detected via RFID. The system is connected via BLE.

4.4.2. Microcontroller

A microcontroller is needed in the prototype to communicate between the RFID reader and the grasp detection. In reasonable finalized devices, a single microcontroller should control the RFID reader, the IMU, the EMG as well as the grasp detection.

Arduino [21, 22] is the standard for the development board used in this work. Arduino is offering boards as well as a software IDE. It is an open source platform (hard- and software) with a huge amount of accessible knowledge. Programming is done in C/C++ with a setup block, main loop and possible interrupts.

RFD22301 RFduino BLE SMT [19] is a small sized microcontroller, compatible with the Arduino IDE. It provides a BLE connection. The size of the board is 15 mm x 15 mm x 3.5 mm. The CPU is a 16 MHz ARM Cortex-M0. A 128kb flash memory and 8 kb Ram is built in.

RFD22301 has been chosen because it provides a microcontroller with BLE in a small size format. Moreover, Arduino is well established in developing of prototypes and easy to use.

4.4.3. RFID Components

The prototype uses RFID for the object detection. Three different components are needed. On the prototype a module is used to create and process the RFID signals. The signals, which has to bridge a few centimetres, are sent over an antenna mounted to the RFID module. RFID tags have been placed on the objects. The tags have to be strong enough to bridge back the distance again.

As module SkyeModule M1-Mini [20] from SkyeTek has been used. Main reason is its small size (25.4 mm diameter and 2.8 mm to the highest part) and the ease in integrating. For the antenna matching circuit, the module has been mounted with a 818Ω resistor as R_{damp} , a 360 pF capacitor as $C_{parallel}$ and a 62 pF capacitor as C_{series} , according to an optimization of another research project [1]. Parts have been off the shelf surface mount devices. Connection to the microcontroller was established over a serial connection.

The used antenna was created in a study at the Technical University of Darmstadt [1, 40] and is worn on the wrist. Reason for choosing this antenna was its already proven effectivity and the possibility of a comparison of both works to show the results of a grasp detection. The antenna was oval shaped, had a size of 7.5 cm x 11 cm and showed in tests to be able to bridge a range of 14 cm.

Used RFID tags have been 5.5 cm x 5.5 cm sized 13.56 MHz passive sticker. Because of the large size, problems with the mounting occurred since they were, for example, larger sized than the Universal Serial Bus (USB) stick and should not be bent to much. However, the large size is also allowing a further reading range. Tags, used in this work, and some placement can be seen in Figure 5.1.

4.4.4. EMG and IMU Sensor System

As sensor system for the IMU and EMG data Myo [5] from Thalmic Labs is used. Myo was chosen because it includes an easy to use, state of the art and low cost EMG Sensor. It is a 93 grams heavy armband with eight uniform distributed EMG channels. The battery lifetime is described as one day while using. A sleep mode is available while it is not moving.

The EMG is described as "Medical Grade Stainless Steel EMG sensor". The sample frequency is 200 Hz. The received signals are described by the distributor as unitless muscle activation, which are in the range of 0 till 128. Per default a 50 Hz band rejection filter is included to filter out crosstalk of other electronic devices. The band rejection can be disabled. Channel numbering can be seen in Figure 4.5.

Myo contains an IMU with a three axis accelerometer, a three axis gyroscope and a three axis magnetometer. However, the magnetometer is not accessible per default. The IMU is sampled with 50 Hz. Directions of the accelerometer and gyroscope are given in Figure 4.5.

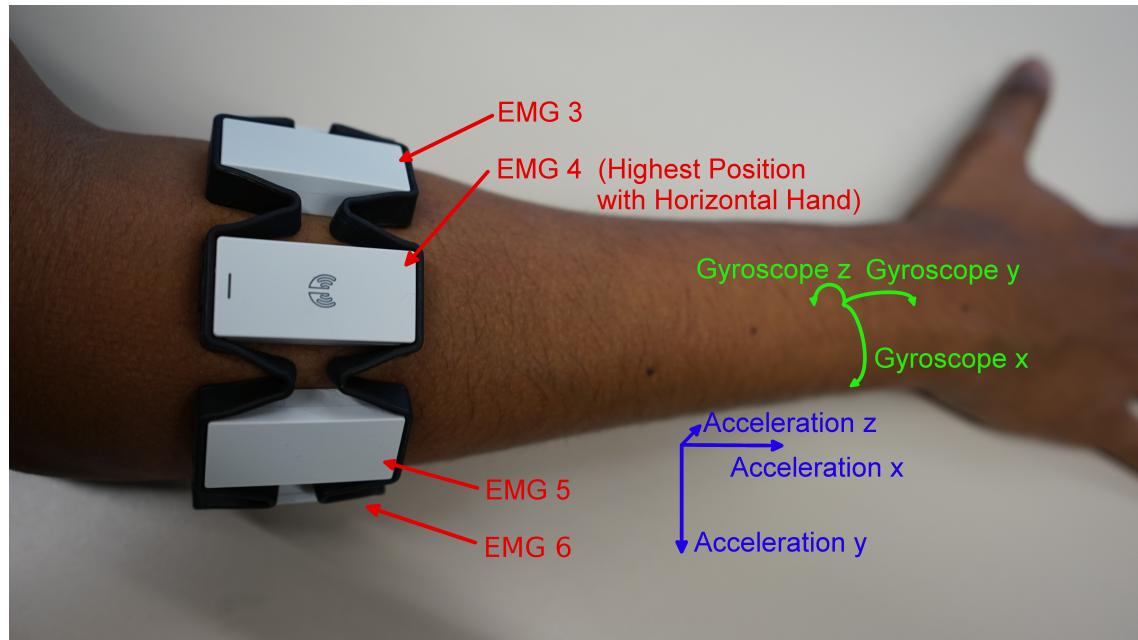


Figure 4.5.: EMG positioning and IMU directions of the forearm sensor.

An ARM Cortex M4 Processor is included, which is not accessible from outside. A haptic feedback is included with three different vibration modes. Signals are sent over the inbuilt BLE, which can be accessed as slaves.

Five predefined gestures are available, which were not used in this work. A Software development Kit is available for C++ programs and LUA scripts. The C++ SDK has been used to create log files. It is also possible to access the BLE communication

directly, and parse and create own packets. This method was used in the real time detection. An explanation about this communication method is written in the appendix.

Several programs are available for the Myo. Myo Connect is worth mentioning because it is the standard interface for accessing the information sent by Myo.

4.4.5. Bluetooth Configuration

While in real applications the system should be realized in one device which does not need a device communication, the prototype is split into three subsystems. As communication method between the devices BLE has been used.

As BLE master the assisting grasp processing computer is chosen, which has access to the piconet via a BLE dongle. Communication to the dongle is established via a serial port interface. The BLE dongle (BLED112 [18]) is produced by Bluegiga and shipped with the Myo sensor.

Slaves are the Myo with an inbuilt BLE module as well as the RFduino. In real-time processing both systems have to connect with the master. In the development phase of the grasp detection only Myo was connected as slave.

4.4.6. Object Detector Case

A case has been 3D printed for the prototype as protection and mount to the arm. The complete size is 7.2 cm x 3.1 cm x 2.1 cm. The case has space for the approximately 2.8 cm x 2.8 cm x 1.9 cm sized object detector and the used 400 mAh battery with a size of 3.5 cm x 2.5 cm x 0.5 cm. It can get strapped with an up to 3 cm broad armband to the arm. The case can be seen in Figure 4.6. The model has been created with OpenSCAD [54].

4.4.7. Development Environment

MATLAB [17] is a software by MathWorks for solving mathematical problems. It has been used for finding appropriate features, testing the grasp detection with saved log files as well as the entire real-time analysis including the communication with the BLE Dongle.

Reason for the developing with Matlab was the huge amount on predefined functions for a rapid development. Matlab is able to easily visualize data in plots. Sensor signals have been plotted in the development phase for finding appropriate features. It is also already including used mathematical functions like a fast Fourier transform (FFT).

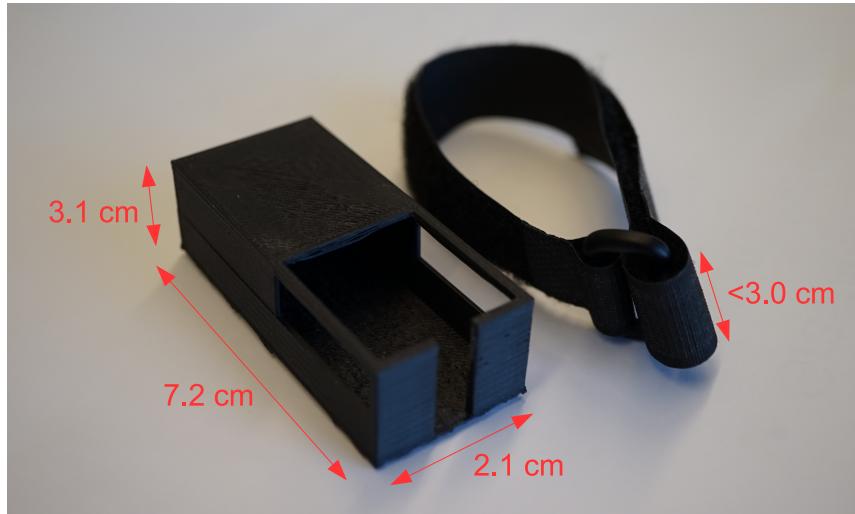


Figure 4.6.: 3D printed case, used for the object detector. It is been mounted with an until 3 cm broad armband to the arm. The closed back is intended for placing the microcontroller and the RFID module. The front is intended for the battery. The opening in the top is for inserting the device, closed with the armband which is intended to be merged in the two holes in the side. The opening in the front can be used for antenna cable.

However, for a practical solution, for example in an embedded system, Matlab should not be used because of its large amount of memory and CPU consumption as well as its often slower reaction times. For such applications, the code can easily be rewritten in another programming language or the Matlab automatic code generation can be used.

For logging the data, a C++ program was developed which communicates with Myo Connect. The main code of this software was published by Thalmic Labs developers [55].

4.4.8. Baseline Test Environment

Another approach was used to get a grasp detection baseline: Trying to search for grasps via well established machine learning methods. An application GRTTool [24] based on GRT [25] was used to search for grasps explained in section 5.6.

GRTTool was chosen because of the easy usage. A variety of preprocessing, postprocessing, and feature extraction algorithms as well as classification/regression methods can be selected. It reads in the logged data, searches for defined features and plots the results by a chosen classification method. Ground truth can be inserted in SubRip text file format or can be directly attached to the sensor samples.

4.5. Forearm Muscles and EMG Signals

4.5.1. Definitions

Some anatomic definitions are explained in this section for an easier description of the muscles and movements.

An extension of the wrist is a moving so that the back of the hand gets closer towards the forearm. An extension in the elbow is the moving away of the forearm from the upper arm. A flexion is the opposite. A twist of the elbow so that the palm is facing the body, is called a supination. As opposite, when the back of the hand is facing the body, is called pronation. A moving of the wrist so that the hand is going in the direction of the little finger, is called adduction, while moving towards the thumb is called abduction. Spreading the fingers is also called abduction, while moving it together is called adduction.

The back of the hand side of the arm is called posterior side. Here are the extensors of the forearm. Palm side is called the anterior side, where the flexors of the arm lie. The radius bone is sitting on the thumb side of the arm, while the ulna is on the side of the little finger. The side of the radius is also called lateral, while the side of the ulna is called medial.

Longus means long, brevis short. Carpi stands for wrist, pollicis for thumb and digitus for finger. The finger numbering starts with I (thumb) ascending to V (little finger).

Superficial means that muscles are close to the skin and are not covered in the main part like the deep muscles. Deep muscles are overlapped by other muscles

The Palmar Aponeurosis is a tendon plate in the palm of the hand. It covers the short muscles, nerves and blood vessels in the palm and protects them. By tensioning the skin it also creates a tight contact to grasped objects.

In the following the forearm muscles and the surface EMG signals are described for the superficial muscles. Figure 4.7 and Figure 4.8 illustrate the positions of the muscles. Sources for the position and functionality of the muscles have been [56, 57, 58, 59].

4.5.2. Detected Signals for Movements

Deep muscles are not as accurate to detect as the superficial muscles. Even if some works, like [28, 45, 44], have been working with the deep muscles, the signals in the tests of this work have been unsatisfactory. For this reason, the only non superficial muscle Flexor Digitorum Superficialis, which is lying on an intermediate layer, is depicted closer. A summary of the measurable movements, the signals and the corresponding muscles is provided in Table 4.1. The channels are placed according to Figure 4.5.

Table 4.1.: Movements measurable by an eight channel surface EMG. Channel ordering according to figure Figure 4.5. Signal strength are in static positions in time domain classified as weak (w), moderate (m), strong (s) activation.

Movement	Corresponding Muscles	Channel, Strength
Pronation Arm	Brachioradialis, Anconeus, Flexor Carpi Radialis, Pronator Teres	1w, 4w, 5w, 6m
Supination Arm	Brachioradialis, Anconeus	4w, 5w, 6w
Flexion Elbow	Brachioradialis, Extensor Carpi Radialis Longus, Extensor Carpi Radialis Brevis, Pronator Teres	1m, 2m, 3s, 4m
Abduction Wrist Posterior	Extensor Carpi Radialis Longus, Extensor Carpi Radialis Brevis	3w, 4w
Extension Wrist	Extensor Carpi Radialis Longus, Extensor Carpi Radialis Brevis, Extensor Digitorum, Extensor Digiti Minimi, Extensor Carpi Ulnaris	3m, 4s, 5s, 6m
Extension Finger II to V	Extensor Digitorum	4s, 5s
Extension Finger V	Extensor Digiti Minimi	5m
Adduction Wrist	Extensor Digiti Minimi, Extensor Carpi Ulnaris, Flexor Carpi Ulnaris	5s, 6s
Extension Elbow	Anconeus	5w, 6w
Flexion Wrist	Flexor Carpi Ulnaris, Palmaris Longus, Flexor Carpi Radialis, Flexor Digitorum Superficialis	7s, 8s, 1s
Tension Palmar Aponeurosis	Palmaris Longus	6w, 7w
Abduction Wrist Anterior	Flexor Carpi Radialis	1w
Flexion Finger II to V	Flexor Digitorum Superficialis	7w, 1w

4.5.3. Posterior, superficial Muscles

Brachioradialis is the muscle on the radialis side of the forearm. It is a flexor of the elbow and can also be used for a pronation and a supination, dependent on the position of the forearm. While supinating only very weak activation can be seen in EMG channels 4, 5, 6. While pronating a moderate activation can be seen in channel 6 and a weak activation in channel 4 and 5. Anconeus also affects supination and pronation. When supinated and strongly flexing this muscle against a resistance, a moderate activation can be seen in channel 1 till 4, with moderate force only weak activation is detected, mainly in channel 3.

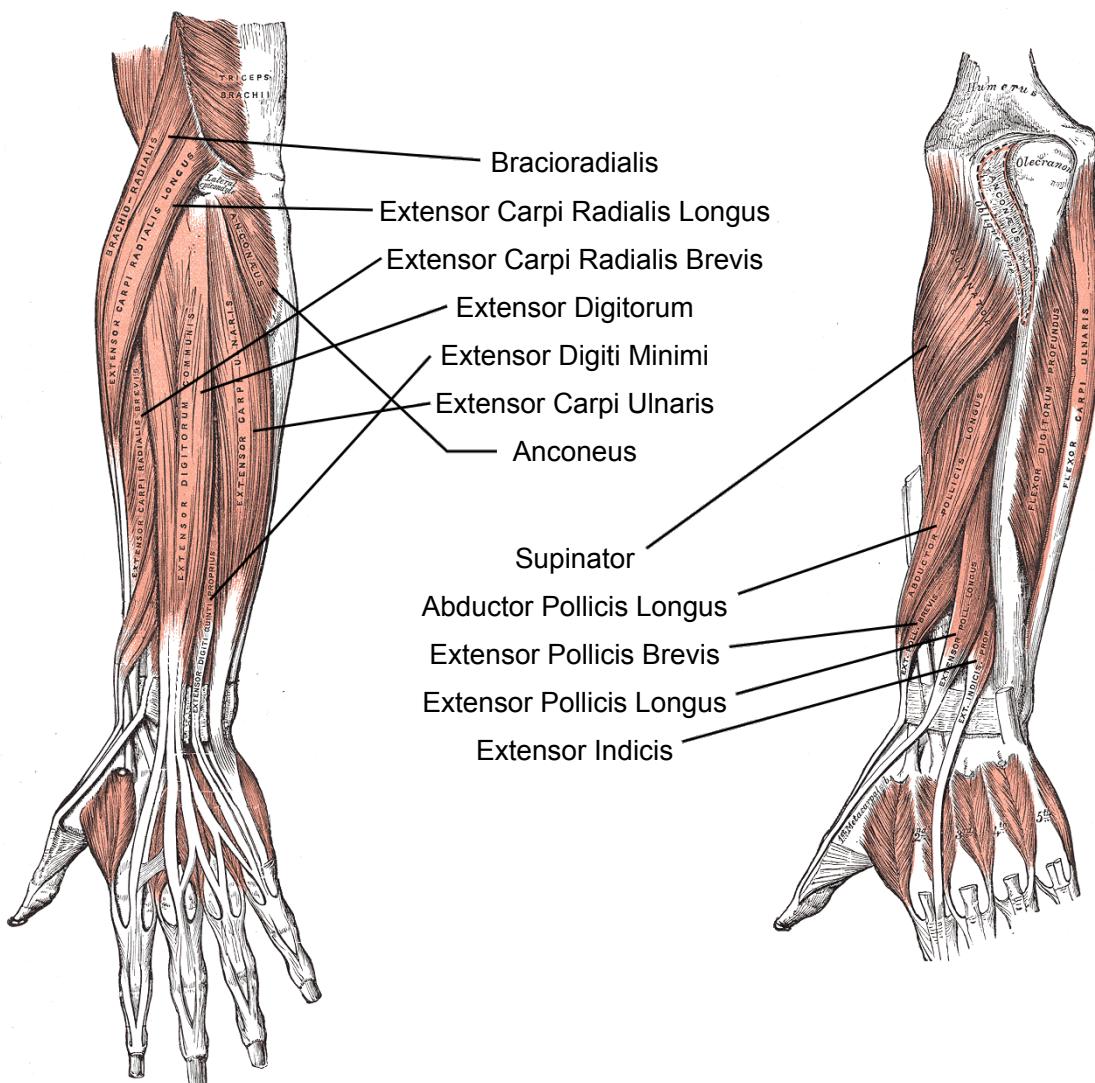


Figure 4.7.: Posterior muscles of the forearm. The left image is showing the superficial and the right image the deep layer. Images are drawn by Dr. Henry Vandyke Carter and under the public domain license.

Extensor Carpi Radialis Longus lies next to the Brachioradialis on the lateral side. It is covering the Extensor Carpi Radialis Brevis in the upper region, which is the next muscle in the row. Both muscles are responsible for an extension and abduction of the wrist. They also flex the elbow. An abduction is forcing weak activation values in the channel 3 and 4. A flexion of the elbow in basis state is creating moderate values in channel 3, with a small overlap to channel 2 and 4. The extension of the wrist without abduction and adduction is forcing, dependent on the strength, weak till strong values in channel 3 till 6, with its peak in 4 and a very weak affection of the remaining channels. For the wrist extension signals also the Extensor Digitorum, Extensor Digiti Minimi and Extensor Carpi Ulnaris are responsible.

Extensor Digitorum is lying straight in the middle of the posterior side. It is an extensor of the medial four fingers and can also extend the wrist. A strong extension of these four fingers is affecting all channels, but in channel 4 and 5 a moderate signal can already be seen with very small extensions.

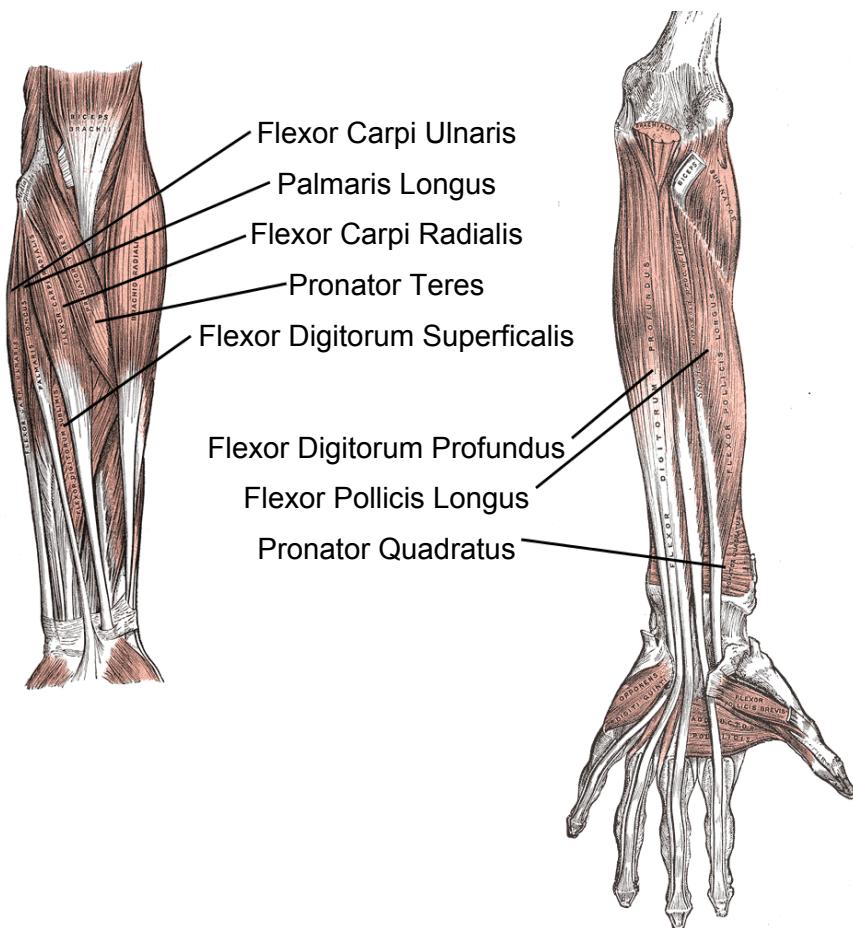


Figure 4.8.: Anterior muscles of the forearm. The left image is showing the superficial and the right image the deep layer. Images are drawn by Dr. Henry Vandyke Carter and under the public domain license.

Extensor Digiti Minimi is a muscle that starts in the little finger and lies next to the Digitorum. In upper region, however, it gets overlayed by other muscles. Some people have this muscle fused with Extensor Digitorum. It is an extensor of the little finger, which also contributes to the extension of the wrist. It also supports an adduction of the wrist. Extending the little finger is creating a weak till moderate signal in channel 5. An adduction is creating moderate till strong signals in channel 5 and 6, which is also contributed by the Extensor Carpi Ulnaris and Flexor Carpi Ulnaris.

Extensor Carpi Ulnaris is the muscle right next to the Extensor Digitorum and Extensor Digiti Minimi on the medial side. An extension and adduction of the wrist is possible.

Anconeus is a short muscle in the elbow region on the ulna side. Its function is to move the elbow at a pronation and a supination and also an extension of the elbow. The pronation is creating moderate signals in channel 6. The extension is creating weak signals in channel 5 and 6.

4.5.4. Anterior, superficial Muscles

Flexor Carpi Ulnaris is lying on the medial side of the arm. It is responsible for a flexion and an adduction of the wrist. The wrist flexion is creating weak till strong signals in channel 7, 8 and 1, depending on the strength of the flexion. The signal is also contributed by the Palmaris Longus, Flexor Carpi Radialis and Flexor Digitorum Superficialis.

Palmaris Longus is a muscle, lying in the middle of the anterior arm side. It is missing in an amount of people at one or both arms. In literature the number of misses is varying at around 15 % and 25 % [58, 57, 56]. A flexion of the wrist is possible. It also tensions the Palmar Aponeurosis. The signal strength of the tensions are rather weak in channel 6 and 7.

Flexor Carpi Radialis is next to the Palmaris Longus on the lateral side. Its main task is the flexion and the abduction of the wrist. With a stretched elbow it can also support the pronation. Weak signals can be seen by the abduction and the pronation in channel 1. The pronation is also contributed by the Pronator Teres.

Pronator Teres is a short muscle near to the elbow on the lateral side of the Flexor Carpi Radialis. It is responsible for the pronation of the forearm as well as for the flexion of the elbow. The flexion of the elbow can be seen by moderate signals in channel 1.

Flexor Digitorum Superficialis is lying actually in an intermediate layer and is mostly covered by other superficial muscles. It is in the middle of the anterior arm side with an ulna origin near the elbow. Its endings are going through the carpal tunnel to the fingers II, III, IV and V. It is responsible for a flexion of the four fingers and the

wrist. Even if the muscle is mainly overlapped, very weak signals in channel 7 and 1 have been found by flexing the fingers.

4.5.5. Posterior, deep Muscles

Supinator is a muscle on the radius side under the Brachioradialis. It supinates the forearm.

Abductor Pollicis Longus is a muscle that is going from the radius side near the wrist and ending in the middle of the forearm on the ulna side. Its main purpose is the abduction of the thumb. It can also contribute to an abduction of the hand as well as a supination.

Extensor Pollicis Brevis is right next to the Abductor Pollicis Longus on the medial side. It is also shorter than the Abductor Pollicis Longus. It extends and abducts the two joints of the thumb near the wrist, but not the last one.

Extensor Pollicis Longus is next to the Extensor Pollicis Brevis on the medial side. It extends and abducts all joints of the thumb.

Extensor Indicis is medial to the Pollicis Longus. It is responsible for the extension of the index finger.

4.5.6. Anterior, deep Muscles

Flexor Digitorum Profundus is lying beyond the Flexor Digitorum Superficialis in the middle of the arm. It starts also at the ulna and is also splitting, going through the carpal tunnel and to the end of fingers II to V. These four fingers are flexed by the muscle. Also a wrist flexion is possible.

Flexor Pollicis Longus is starting on the thumb, going through the carpal tunnel and is lying on the radialis. A flexion and the positioning of the thumb is possible. This positioning is essential in many grasps. Moreover, it supports the flexion of the wrist.

Pronator Quadratus is lying under Flexor Digitorum Profundus and Flexor Pollicis Longus. It is a short muscle near to the wrist. As the Pronator Teres this muscle is responsible for the pronation of the forearm.

4.6. Feature Extraction

4.6.1. Signal Preprocessing

Searching for features can be done in several domains. Time domain features have the advantage that they typically need no or little preprocessing, whereas frequency domain features need a costly transformation beforehand.

In the time domain, sensed values are shown over time as they are collected by a sensor. Often smoothing or filtering operations are done for these signals. Two filtering examples which have been used in this work are: A high pass filter for precluding the gravity factor in the acceleration data and the band rejection filter for precluding crosstalk from external voltage sources in the EMG signals. Some techniques also reduce the needed memory or processing power in time domain. An example is mSWAB [60], an approximation and matching technique reported to achieve good performances on human acceleration data.

The frequency domain is calculated from the signals in time domain. Instead of showing the raw data, it sums up the frequency fractions in a transformation. Several methods are used to derive different frequency domains. The most common methods are Fourier Series, Fourier transform, Laplace transform, Z transform and Wavelet transform. Ordering the bands linearly is not recommended in motion and EMG data since low frequency bands would achieve higher values than high frequency bands.

In this work a Discrete FFT has been used. Besides the Fourier transform, also the Wavelet transform is favoured in motion analysis because the Wavelet transform additionally includes information about the temporal resolution.

4.6.2. Statistical Values

It is easy to see movements, their strength and their directions in the time domain of IMU data. Grasps, however, are typically showing smaller maximal values than a lot of other movements. The maximal values are also varying per person and by the body posture of the person. Yet, fixed, person independent parameters have been chosen for the decision stumps.

Two useful characteristics of IMU time domain data exist. An intuition in which direction the arm has moved can be made and short movements often emit a strong peak. An example for the usage is the detection of a rotation of the wrist right before the grasp. Using these two properties can emerge person dependent problems, described in section 4.2. Another disadvantage is that stopping a movement is often emitting a peak which is higher than the movement itself. This behaviour can make a estimation of the direction in a series of movements complicated.

Frequency domain for IMU data is useful for detecting strong, continuous movements. In this work, peaks in frequency bands and the spectral entropy has been used in Layer 1. Layer 2 movements have been directed without any continuity, which makes a frequency analysis impractical.

The EMG features which have to be detected are varying by the application field. To find the right features five different questions are relevant [48]:

- Is the muscle active?
- Is the muscle activation increasing or decreasing?
- When is the muscle active in relation to other muscles?
- How strong is the muscle activation?
- Is the muscle getting tired over time?

This work is covering the first question and is searching for any muscle activation in its decision stumps. The third question is covered indirectly by checking for the muscle activation by a step order, but there is no direct calculation of the time between the activation of two different muscles. The value of the EMG signal is correlated to the used force, however, the used stumps are just testing against some noise level, emerged by random firing of single cells, measurement chain noise or ultra low usage of the muscles.

It is typical to use absolute values for extracting EMG features in time domain [48]. In literature mentioned EMG features are maximum, minimum, mean, median, integral, variance, region length, autoregression coefficients, root mean square, zero-crossing, slope and slope sign change [27, 48, 61, 62]. A distinction between positive and negative values or the variance between their peaks is usually not significant because of the method the signals are collected (see section 3.1). The root mean square is recommended as smoothing algorithm [63], however, since it was searched rather for short variation in the muscle than long time behaviour, no smoothing is used in this work.

Examples for EMG frequency domain features are frequency of the peak, peak power, values in the frequency bands, spectral entropy, middle and median frequency. Middle and median frequency are common features in static or long time tests [48], however, not useful in this work as it is been searched for short activations of the muscles. All implemented frequency domain features are using a broad low frequency band which is checked for an increased mean value, which may can be exchanged by a low- or bandpass filter or a zero-crossing approach in resource critical systems.

4.6.3. Methods

Two approaches for selecting the features have been made. Method one was the visualizing of the data streams and statistical patterns and searching by eye for outstanding values. Method two was to achieve knowledge of muscles and grasp trajectories.

Visualizing of the data has been implemented via Matlab since it provides easy methods to plot the series of data achieved by the sensors and also provides easy

methods to process the data. Some examples for processing the data are filters, transformations into frequency domain, make values absolute and calculating Euler angles. Visualizations then have been plotting data over time, plotting feature over feature (as point clouds and probability distribution function (PDF)) and spectral analyses (linear bands, cepstral coefficients, low frequency energy, spectral entropy).

A principal component analysis (PCA) has been tested to give a raw estimate how much features are needed to separate the steps of a grasp and eventually decrease the feature space. The steps are explained in section 4.1. Result of the PCA was that even with very easy grasps at least 4 to 6 components are needed to achieve a classification certainty for steps over 95 %. As consequence of this result, no classification method based on single strong features has been chosen, but rather a boosting method of a series of weak decision stumps.

Knowledge has been found in literature research and by simple reasoning about possible features. Muscle positions and their functions have been found in anatomy literature [56, 57, 58, 59]. They are documented in section 4.5. Emerged features from EMG have been found in previous papers about EMG [27, 48, 61, 62, 63], explained in the previous subsection. Grasping trajectory and possible IMU features have been found by reasoning and observing participants. Information about hand positions while grasping are found in papers about grasp taxonomies [31, 32, 33, 34, 35, 36, 37, 38] and summarized in section 2.2.

Found features of both methods were checked by the second method. Features found by knowledge were often impractical because the data was overlapped by a lot of other influences. The strength of knowledge found features is difficult to guess. Especially acceleration features are often getting lost behind stronger movements.

Statistically found features are often not reproducible in the real world because they are only found in easy testing areas. Testing in different environments and with several participants is recommended for statistical features. To give an example for a false detected feature, in the first tests a pitch movement of the arm before the grasp has been found which has been caused by an arm trajectory over the sidewall of the used box.

4.7. Developed Software

4.7.1. Logger

A logger for the IMU and EMG data, sensed by the Myo armband, has been developed. Most code has been published by Thalmic Labs [55], which has been slightly varied in this work. It is been written as a C++ console program. It requires the Myo SDK [5], which has been used in version 0.9.0. It communicates with the Myo Connect [5] software, which has been used in version 0.15.0.

For organization reasons the logfiles are saved with a given filename and the timestamp of program start. Capturing is event based by incoming packets. Since Myo does not send timestamps, the program is setting the timestamp of the processing, which differs by a few milliseconds from the collection time. As the sample frequency stays constant, the small variances between the timestamps do not have influence in processing.

Information about the connection losses and reconnections are displayed. In case of a reconnection, new logfiles are created with the timestamp of the reconnection in the filename.

EMG, gyroscope, accelerometer and orientation data is logged in four separated files. Also two action keys are implemented to mark timestamps in a separate file for postprocessing.

The object detection is logged only in the real-time grasp detection script, which can also be used for logging the data. Reason for this is the interplay of the grasp detection and the object detector. The individual object detector is expected to be working.

4.7.2. Feature Extraction Script

Extraction of the features has been done via Matlab. Methods of the extraction have been explained in section 4.6.

Two visualization scripts for the data have been developed. Both scripts use as input the log files of the Myo sensor, generated by the developed logger.

The first script does not need an additional data preparation and is intended for a fast observation of the logged data. It plots the EMG, accelerometer and gyroscope data over time. It also inserts bars when a timestamp has been marked in the logger.

The second script requires an additional prepared timestamp table of the grasping steps. This table has to be prepared by checking the ground truth data. The Euler angles and the linear acceleration is calculated in the first step. The output is marked per grasping step, and also per grasp performed.

Outputs are the accelerations, angular velocity, orientation, Euler angles and muscular activation over time. Outputs are also plotted over each other to search for hidden patterns. These plots are available as point clouds and PDF. The two dimensional PDF ellipses have been plotted with a chi square distribution with a 95 % level of acceptance. Besides to the time domain plotting, the same procedure have been done with Fourier transformed frequency domain data. Cepstral coefficients, frequency bands and spectral entropy has been chosen as frequency domain data. Frequency domain data has also been plotted against time domain data.

A principal component analysis is also included in this script. Output is the percentage of the ordered cumulative sum of the component variances.

4.7.3. Grasp Detection Scripts

Two Grasp detection scripts have been developed. The first is evaluating logged data, while the second is reading the sensor data in real-time and is also communicating with the object detector.

To create the log files for the first script, the developed logger has been used. The script is first loading these log files, then calculating linear acceleration, absolute EMG data and Fourier features.

A sliding window approach is used to process the features. The window is moving for 220 ms, as it is the sampling time of the real-time detection.

Outputs for the script are found grasps, found rests and found walks. Grasps are plotted over the time scale, together with EMG data and the action timestamps of the logger. Walking and resting is marked in separate plots over time with acceleration and the action timestamps. However, a comparison with the ground truth is still to be made manually with the video data.

The real-time detection is starting with the establishing of the BLE connection to the forearm sensor and to the object detector, as described in Appendix B.

An endless running loop is continuously parsing the packets coming from the connected devices. The processing for linear acceleration, absolute EMG values and Fourier features is happening in the loop for short time intervals.

As output, the script is listing the found features in every sample step. The communication with the object detector is established when a grasp is found. Responses of the object detector are listed with ID of the communication process, tag ID (if found), the timestamp of the requesting as well as the timestamp of the acknowledge of the reader. Feedbacks, like a text message on the console, vibration in the sensor armband or controlling a LED in the object detector are possible. A plotting of incoming sensor data is also possible. All sensor data and outputs can be saved for a later evaluation.

For both scripts the same features have been used for base activity and grasp detection. Also the sequencing of both scripts is the same. The grasp detection model, explained in section 4.3, is the base for this processing.

4.7.4. Object Detector Software

The object detector is setting up the device name published in the broadcasts. Event triggered, it checks for the next readable tag and sends back a consecutive communication event ID as well as the detected tag ID. In case of no tag has been detected, it only returns the communication ID. While no read is requested, the microcontroller is in sleep mode.

The driver for the RFID reader module is from the WRIFD project [23]. Additional feedbacks could be activated, e.g. driving a LED for connection, disconnection or communication events.

4.7.5. Device Communication

Myo Connect has been used for parsing packets for the logger, while the real-time grasp detection was using its own parsing and sending method via HCI commands. In the start of the communication between Myo and the grasp detection system, configurations from the grasp detection system to Myo is sent. After this the grasp detection system is only listening to Myo. The first step is to connect to the Myo device. The sleep mode is getting disabled immediately for keeping the connection. Next step is the subscribing of the IMU and the four EMG services. Myo is receiving information about which data it has to send. Therefore, the IMU data is getting enabled. The EMG data with the 50 Hz bandstop is getting enabled. No gesture recognition in the Myo is needed. From this point Myo sends alternating 2 EMG packets and 1 IMU packet to the gesture recognition until an error occurs or Myo is getting told to stop. Errors examples can be a connection loss when displacing a subsystem too far or an empty battery.

Communication between the object detector and the grasp detection is event-based, triggered by the grasp detection script. When a grasp is getting recognized by the grasp detection script, it sends a consecutive communication event ID to the object detector. The object detector sends an acknowledge to the grasp detection with the same communication ID and, in case of a found Tag, the tag ID.

Object detector's microcontroller and the RFID reader module are communicating over a driver, developed by Philipp Scholl [23]. It is implemented as serial communication, which is sending the read command as well as the get firmware command from the microcontroller to the RFID reader module and the scanned tags from the RFID reader module to the microcontroller.

5. Experiments

5.1. Experimental Setup

Two different approaches have been tried for the evaluation of the system: In real-time and via logfiles. The evaluation was performed for the grasp detection system without object detector as well as for the complete system with a RFID reader.

Even if there are differences in the participants behaviour, which result in different performing of grasps, the features and the parameter calibration of the system was the same for all participants. Without any calibration phase for the participants the system has been used straight away. All participants were ought to use the right arm. Also the placement of the forearm sensor was near identical, without measuring the correct positions. The movements to the object, kind of grasp and duration they touched the object was their own choice.

5.2. Logfile Detection

In the detection via logfile analysis, the data of the experiment was postprocessed with the same configuration as one would use it in real time. Participants were told before the experiments how to wear the sensor and they have to perform tasks from a script (section C.1) while wearing it. They were also instructed to use the tools (objects which are later validated) only with the right hand. No further instructions on the purpose of these tests were made before the experiment. Participants were free to perform the tasks as they wanted. No instruction on the types of grasps were made. They did not receive any feedback of the system.

A ground truth annotation was done afterwards. A video camera has been used for this purpose. Timestamps and used objects were listed in tables. It was also tried to use keyboard inputs to set marks between the grasps, but it turned out that this method was difficult in execution, so this was just a rough indicator.

The purpose of the tasks was to take the participants mind off grasping. Therefore, the tasks were more about the acting with tools while sitting or standing with interleaved activities, such as walking or drinking from a bottle.

The task included RFID-tagged objects as similar as possible to the box test reported in a study about RFID object detection [1]. A filled 0.5 L¹ plastic water bottle and

¹Usage of the non-ISO unit L for litre, because of possible confusion with "1" and common practice.

cardboard box including a hammer, a screwdriver, a craft knife, an empty 1 L plastic bottle and an USB stick were used. These items are also depicted in Figure 5.1.



Figure 5.1.: Items used in the experiment were tagged with RFID tags and were chosen to be similar with the objects reported in research project on wearable RFID object detection [1].

The used objects had been chosen for two reasons: An expected difference in handling and a possible comparison of the results to a research project on wearable RFID object detection [1]. The expected difference in handling is caused by a difference in weight (stronger or weaker grasps necessary), difference in usage of the object (which cause different grasps from the taxonomies of section 2.2) and difference in form (other movements towards the object as well as different grasps).

Before the actual experiments, the participants were asked to make some movements with the arm and hand for a later synchronizing of the data with the video as well as for getting some kind of normalization for the data. The normalization has not been used afterwards.

The first two tasks involved an interleave by walking around in the room. The participants were asked to walk to a water bottle, grab the bottle, bring it back, drink a bit and put it on the table. Afterwards they were told to bring a cardboard box to the table, filled with tools needed for the next tasks.

The usage of the tools was performed in the first phase standing, in the second phase sitting. A nail has been hammered and a screw has been screwed into a wooden

5.2 Logfile Detection

board. Some cuts were made in a paper sheet with the craft knife. The empty bottle and the USB stick were just moved around between box and table. After the standing and before the sitting phase, the box has been flipped on its side that the opening of the box is pointing towards the participant (which was also supposed to be detected).

In the end of the experiment another sip of the bottle has been taken and another arm movement for a later synchronization of the data has been performed. At last, an explanation of the purpose of the experiment has been given. The whole test row after the build up needed between 7:30 and 9:30 minutes.

This is the main test for the grasp detection. Different reasons made this test suitable. Keeping the participants distracted from the grasps, giving them no feedback and instead giving them tasks is achieving results that are closer to reality, when trying to build a system in which the user should not be too much concerned about his grasping style and rather is concentrating on his work (see section 4.2). Standing, sitting and moving as well as a free decision on how to perform a task is allowing a change in the posture and position of the user, which is resulting in different angles and movements towards the object. The walking in the tests is a quick check for the performance of the walk detection, even if this test is more concentrating on Layer 2 of the model. Logging of the data is allowing a later evaluation on algorithm changes. Lastly, the ease to compare it with a well designed study for a system with similar components with a full and a divided duty cycle [1] was possible.

In the following, it is called a true positive when the system is detecting an occurring grasp and a false negative when the system is missing a grasp or detect it too late. It is called true negatives when the system is not detecting a grasp when no grasp occurred and false positives when the system is inferring a grasp when no grasp occurred.

A disadvantage in the experiment is that the definition of true negatives in this context is difficult because of the freedom of the performance of the tasks. The true negatives are measurable over the whole time, which would result in a negative prediction value close to 100 %. However, this is not realistic because the resting periods are very arbitrarily.

The false positives are strongly depending on the usage of the system (e.g. time spent resting). Also, it is challenging to decide if some movements should be rated as grasps (e.g. lying the hand on the leg or tightening a weaker grasp). A better way than counting, is a description of situations when false positives occur.

The degrees of freedom in this test have been on purpose. In contrast, a pedantically scripted test set with no degrees of freedom to the user by choosing a grasp would allow an evaluation of the false positives and true negatives and therefore could be visible in, for example, a comparable Receiver Operating Characteristic. However, this would restrict the variable and natural behaviour of the participants. Even for testing specific grasps, the participant would not grasp as he would do it usually. Moreover, this test would only be valuable in comparing different grasp detecting

algorithms, not real environment behaviour because the real life tasks of the users should be dissimilar and the system is mainly producing false positives in concrete situations.

5.3. Real-Time Detection

For testing the real-time behaviour of the system and getting some user feedback, a real time detection has been tested with the prototype. The tests have been performed with random visitors in the office as well as on some presentations of the work (presentation session of a practical course in the university; open day of the 20th anniversary of the technical faculty). The participants had been instructed on what the system does, how it works and they have got a visual and vibration feedback of the system. The objects were mostly the same as in the logfile detection. The tests rows were pretty short (around 1:30 minutes).

5.4. Grasp Detection

In these tests only the grasp detection has been tested, without the object detection. Therefore, only the forearm sensor has been worn, no object detector. The tests have been done like described in the logfile and the real-time detection.

Purpose of these tests was to get an idea about the recall of the grasp detection. True positives are only allowed in a short time interval. Therefore, a point in time has been estimated by checking the ground truth. This point is exactly the time when switching between the step "Closing Hand around Object" to "Holding Object". It is the start of the time when the object would not fall off the hand when moving it anymore. A threshold to detect a true positive around this point in time has been given. The threshold is 50 ms before to 350 ms after this point. For a counting true positive, the grasp detection has to find a detection within this timespan. A missing grasp detection in this threshold is counted as false negative. In combination with logfile detection, this was the most important evaluation of the grasp detection in this work since the recall should be high.

The recall could be manipulated by increasing the sensitivity and therefore decreasing the precision, but it was tried to get a reasonable precision. The precision is further tested in section 5.5.

Even if there was no direct usage of RFID tags at this point, they have been already placed on the object for comparison with the test with an object detector.

5.5. Grasp Recognition False Positive Grading

To get an idea about the amount of false positives of the grasp detection, an extra test have been performed. Following a "normal" working day of an amount of test participants, using the gesture recognition, and evaluating the results, has not been possible within this thesis. Instead, eleven gestures have been used and evaluated, which are similar to grasping.

The tests included a test row while standing and a test row while sitting. Three times each gesture has been performed. As cross check, every object used in section 5.2 has been grasped three times standing before and sitting after the non-grasp tests. Five participants have been evaluated. A test set have been around 11 min long.

Even if windmill and circulate wrist have been resulted in a number of false positives, they have not been used in the tests because it is not expected to be a typical, often performed activity. Gestures, which are expected to be similar to often performed non-grasping gestures, need some force and are different to each other gesture that have been used. Following gestures have been chosen:

- Touch leg (standing) / laying down hand on leg (sitting)
- Use light switch
- Scratch head
- Write a sentence on a keyboard
- Pull back the arm sideways
- Touch nose
- Sit down/stand up, while propping up on armrest
- Put hand in trouser pockets
- Twist forearm in a supination
- Twist forearm in a pronation
- Push a full, standing 0.5 L water bottle with the back of the hand

Results of this method are expected to be higher than in realistic long time usage. It has been tried to trick the system and to generate false positives on purpose. However, to get a number, this method seemed to be realizable in the period of the thesis.

5.6. Classification Baseline

The grasp detection has also been tested via some machine learning. The program GRTool [24], based on GRT [25], has been used as classification system. As classification method the Random Forest algorithm has been used. The log files have been

tested for the step "Closing Hand around Object" as well as for the step "Holding Object". The testing of step "Closing Hand around Object" has been tried with the acceleration as well as the EMG data, as both sensors can influence this result. Step "Holding Object" was just tested with the EMG data because movements are arbitrary in this step.

For grasping, a timespan which started 200 ms before and ended 200 ms after the start of the grasp has been used as training data. The start of the grasp has been the time when the object would not fall off the hand anymore when it would have been carried away. For holding, the timestamps of the start and end of the step "Holding Object" have been used. Movements, which are discussable for counting as a grasp, have no influence on the result.

Both cases had only two classes. For step "Closing Hand around Object" the classes "grasp" and "null" and for step "Holding Object" the classes "hold" and "null" have been used.

The tests have been hold very naive. As training and testing data the same log files as in the grasp detection via logging have been used. Inputs have been the log files with the labels. Data has been split in 250 ms sliding window segments for EMG and 200 ms for IMU. As features mean, range, variance, median, zero-crossings and root mean square have been used. Features have been z-normalized. Data has been divided into 50 % training and 50 % test set in every test. 100 random sampling steps for training and test data have been done.

No sequencing of the data has been performed, which means that the result is only dependent on the actual segment.

Since it is unclear, whether the models have just been learned on strong EMG or no strong EMG, the classification has also been tried on the false positive grading log with the highest precision/lowest false positive rate. Only the "Holding Object" timeslots of the grasped object have been chosen for the holding class.

5.7. Object Detection with Grasp Detection Underlay

For the object detection, the USB stick has been exchanged by another. The metallic case of the first USB stick was absorbing too much energy, which was drastically decreasing the range of the detection. Since the knife blade was metallic, too, it had similar characteristics as the USB stick. However, the absorption was not that strong and the objects should not differ too much from the objects used in the grasp detection. Moreover, a too strong folding or too short range characteristic of some tags made it necessary to exchange or to reposition them on the objects, which was influencing the handling of the USB stick.

5.8. Energy Approximation

The energy saving potential calculations are based on a comparison of the approximated energy of the developed system with an approximation of the time the user can shut the systems off towards an always on or duty cycle approach. A case in which the system is incorporated in another system, using the sensor systems anyway, is not provided. Therefore, only the additional calculations of the microprocessor would enter in the energy overhead, plus the time the object detector is on. Several methods can be used for the approximation of the energy consumption of the subsystems.

Since a commercial IMU with a detailed technical description is used, the easiest way to infer to the energy consumption is reading the data sheet. The approximation for the used accelerometer and gyroscope is based on the normal operating currents of the datasheet of the used InvenSense MPU9150 [64].

The energy consumption of the EMG sensor has not been provided by the Myo manufacturer. As calculation method, the other energy constraints inside the system have been approximated (IMU as described in the last paragraph, Bluetooth Low Energy from typical state of the art values, battery capacity) and the runtime of the system is taken from the developers web presence [5]. Consumption has also been compared with a low power EMG sensor described in [65].

The energy consumption of the RFID reader has been adopted from the datasheet of the used RFID module [66].

Several methods are possible to calculate the processor consumption. A very rough approximation is to take a value in the datasheet of a processor load typical for such systems. Another method is to simply measure the consumption of the microprocessor (ingoing minus outgoing) and take the mean over time. It can also be checked how much time the processor can spent in sleep mode and take the values for sleeping and running from a datasheet. An approach, which also informs about processors that are not touched yet, is to check the algorithm code, write down the operations and check the power for the used operations in the processor's datasheet. Müller et al. [67] defined the energy per logic operation as time period necessary for carrying out a logic operation times the power fed into the system in this period.

Since there is no device which is already combining the subsystems in one device, the actual processing power estimation has been done on counting operations in the Matlab grasp detection in Appendix A. Some approximations have been done on basis of the datasheet of the microcontroller as well as the documentation of the processor architecture. By comparing the estimated processor cycles with the possible speed, a generous duty cycle of the processor modes have been issued.

6. Results

6.1. Features for Grasp Detection

6.1.1. EMG Features realized in the Grasp Detection

EMG data can give a strong indication of tensing of the hand and bowing of the wrist. The first important finding in the EMG data is that some muscles only show an activation under a lot of stress, which means that each feature should be checked statistically across multiple persons. The second finding is that the signals are dependent on the person, the preparation of the skin, the persons form of the day and the exact positioning of the sensor. The last important finding is: Flexors of the hand are on the anterior, extensors are on the posterior side of the forearm and the muscle group of extensors can be used in different steps than the group of flexors.

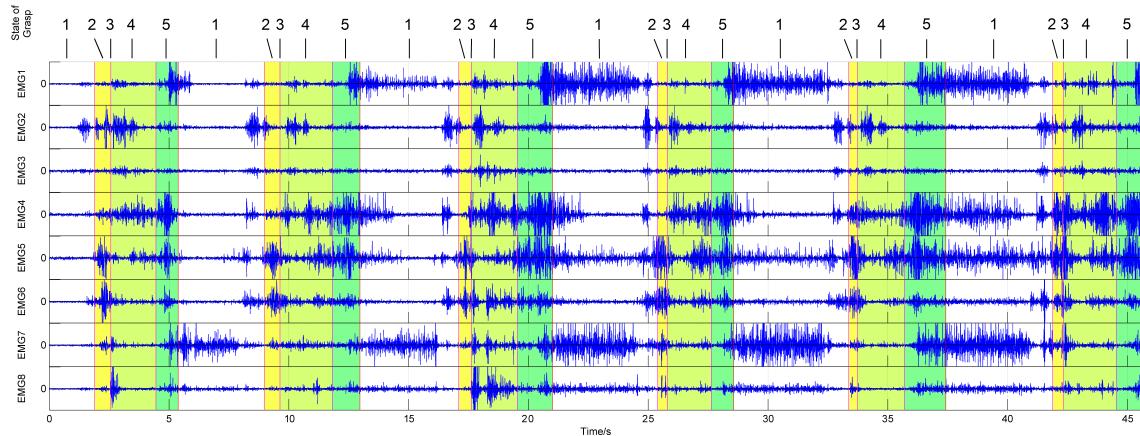


Figure 6.1.: Activation signals of the eight EMG sensor units over time in a series of seven grasps of different objects, from the same participant. Grasping steps are numbered according to the steps in Figure 4.1. Coloured bars indicate a step change, where the third step is a relatively short period of time. The EMG sensor units have been worn equidistantly on the right forearm, near the elbow, with a clockwise numbering starting on the anterior side.

Figure 6.1 shows a series of seven typical grasps. The numbering on top of the figure matches the numbers in Figure 4.1. The EMG sensors are worn equidistantly

around the right forearm, a few centimetre away from the elbow. The numbering of the sensors is made clockwise: Sensors 8, 1 and 2 are worn on the anterior side and sensors 4 till 6 are worn on the posterior side of the arm. Steps 1, 2, 4 and 5 are shown between the bars, step 3 is a very short time period marked with the second magenta bar of each grasp. Step 1 is part of the null class and is not used when building a detector, however, the features should be distinguishable from it. Step 2 is the arm movement to the object with the intention to grasp: Usually, sensors 5 and 6 show a voltage increase because of the extension of the wrist and fingers. Step 3 marks the closing of the hand around the object: Mostly, short spikes in the voltages of the flexor sensors 1 and 8 can be seen. Step 4 marks the holding of the object: Depending on the strength of the grasp, all sensors, but mainly sensor 4, show an increased voltage. Step 5 marks the letting go of the object which can, but must not, start with a flexion of the wrist, shown by a spike in the corresponding sensors and then an extension with an increased voltage in sensor 4, 5 and 6.

The exact used features are based on decision stumps with a sample period of 220 ms. The threshold of the decision stump was an increase by 1.5 to the normalization. The normalization was dynamically actualizing by the previous feature value and the former normalization value. Only one feature, which was based on the peak value, was using the former feature value instead of the normalization. Reasons for using a non-constant normalization are:

- Checks can be performed for an "activation" of the muscles instead of a maximal value
- A possible drift in signal strength
- Difficult to train a normalization factor because of several training methods which all not achieve a perfect result for this task [48]

Features are either extracted from the time domain or from the low frequency spectral band. Extracted from the low frequency spectral band are following features:

- EMG2: "Holding Object"
- EMG3: "Holding Object"
- EMG4: "Closing Hand around Object" and "Moving Hand towards Object"
- EMG5: "Moving Hand towards Object"
- EMG6: "Moving Hand towards Object"

Time domain features are:

- EMG1: "Holding Object"
- EMG4: "Closing Hand around Object"
- EMG5: "Moving Hand towards Object"
- EMG6: "Moving Hand towards Object"
- EMG7: "Holding Object"
- EMG8: "Holding Object"

6.1.2. IMU Features realized in the Grasp Detection

In the start of the work all available IMU data has been searched for strong features. One of the first assumptions was to check Euler angles, however, the Euler angles have not shown a strong distinction to null.

Another remarkable observation was that gyroscope and accelerometer data have emerged similar features. This finding is making one of both sensors unnecessary. Since the gyroscope is the more costly sensor in case of energy consumption and the accelerometer usually performed better, for features the accelerometer should be the first choice. Only one of the found features performed better with the gyroscope than with the accelerometer.

The step "Holding Object" is impossible to model with IMU features. The reason is the high variability in this class. Therefore, all described Layer 2 features are belonging to the step "Moving Hand to Object".

Inertial measurement units have the advantage that they are more common, cheaper to integrate into designs, and smaller and thus more comfortable to wear. However, in case of grasping, IMU signals alone were found to be not specific enough to accurately detect grasping gestures in the first studies. A recall of > 95 % was achieved with significantly more false positives than true negatives in Layer 2. Decreasing the false positive rate was decreasing the recall drastically. For this reason, an EMG sensor is recommended in Layer 2. As sampling time 220 ms is used to get a realistic time window for fast grasps.

The following Layer 2 IMU features are derived:

- Directed movement towards the object: The directed peak of the linear x-axis acceleration is larger than $0.04 \frac{m}{s^2}$.
- Change in positioning of the hand: Variance of x-axis acceleration summed with the variance of the y-axis acceleration is larger than $0.2 \frac{m}{s^2}$.
- Detection of a fast directed twist in the wrist before or during the grasp: This feature is based on angular acceleration. For this reason, the gyroscope is recommended. However, for sparing the gyroscope it can also be detected in the accelerations with some tricks. For a detection with the gyroscope the threshold $14 \frac{\circ}{s}$ was used.

Layer 1 features for resting and walking are derived. Resting is simply based on the undirected average of all of the three acceleration axes and has to be below $0.01 \frac{m}{s^2}$. Walking was based on strong low frequent swings. In the x-axis the lowest frequencies have to be above a threshold as well as the total of the lower frequencies have to be over a threshold. The y-axis was checked for its spectral entropy in the lower frequencies. The z-axis was tested for a threshold in the lowest spectral band.

6.2. Grasp Detection via Logfiles Results

The postprocessing of the log files was using the same features and inputs as the real time system. The timestamps of the recognition have been synchronized with ground truth annotations from video data taken during the experiments. For this ground truth, the time when the hand or fingers start touching the object with enough force so that the object would not fall off while moving was used. For an accepted true positive instance, the found timestamp had to be between 50 ms before and 350 ms after the ground truth mark.

The shortest period for step "Holding Object" have still been over 1 s, which means that the calculation in Appendix A, using the 650 ms as latency limit for the object detector after the moment the object is grasped, is sufficient and the 350 ms evaluation period after the ground truth mark could be extended.

Per participant, a total of at least 18 grasps should have been occurred. Because of a relatively loose usage of the objects and due to extra gestures between the grasps, this number often increased slightly. For instance, some participants were putting down the hammer first for grasping a nail. Grasping other objects, like the nails, have also been detected, but only moving of the mentioned objects was counted because it was sometimes challenging to distinguish whether a touch should be counted as a grasp. Other gestures for which it was difficult to decide whether they could be counted as grasps, were for example the laying down of the hand on the leg (which had only been detected in cases when the participant was really grasping the leg or made a closed fist).

Since the number of grasps differed per participant and per object, three different recall rates were received. A recall of 95.20 % was achieved when counting every grasp, 95.50 % was achieved when taking the mean recall over the participants for every grasp and 95.54 % when taking the mean of the objects' recall results for all participants. The participant's results can be seen in Figure 6.2. Five of twelve participants received a recall of 100 %. The two worst results were close above 85 %. Recall results for the objects can be seen in Figure 6.3. The cardboard box and the filled water bottle received the best results with 100 %. The more light-weight or easier to handle objects (USB stick, screwdriver and craft knife) received the worst results between 89.29 % and 92 %.

Results are expressed as recall, as detecting every grasp is most crucial to a object detection system. It is more difficult to summarize precision results since they are strongly depending on the usage of the system. Nearly all false positives occur around the grasp, when performing strong movements or when strengthening a grasp.

6.2 Grasp Detection via Logfiles Results

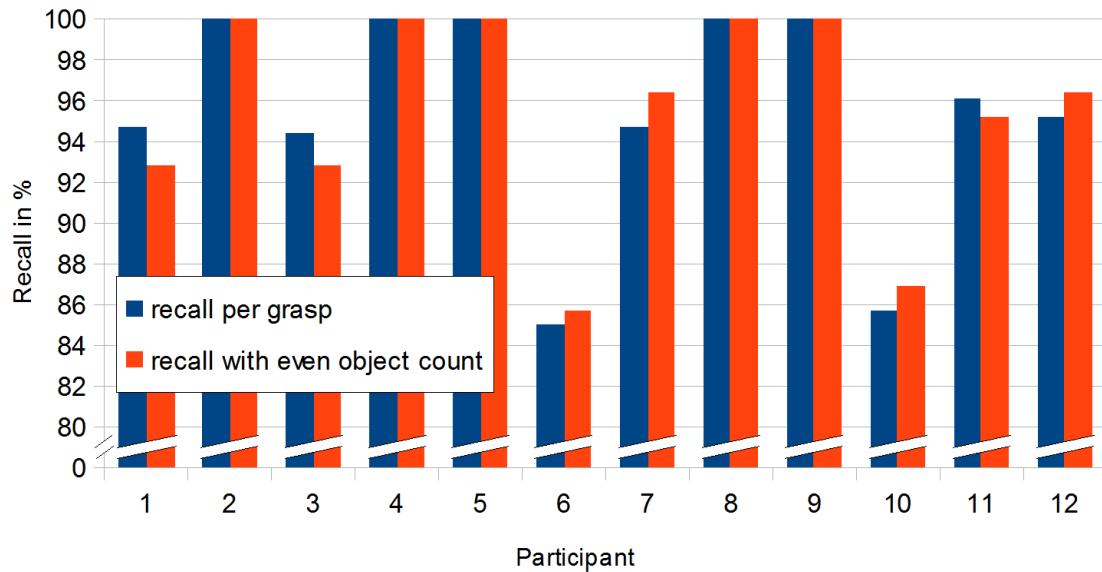


Figure 6.2.: Recall for the participants with all used objects. Since the number of grasps is varying per object, two different results are plotted. The blue bar is illustrating the recall in percent per grasp, while the red bar is showing the recall when every test object has the same influence in the result.

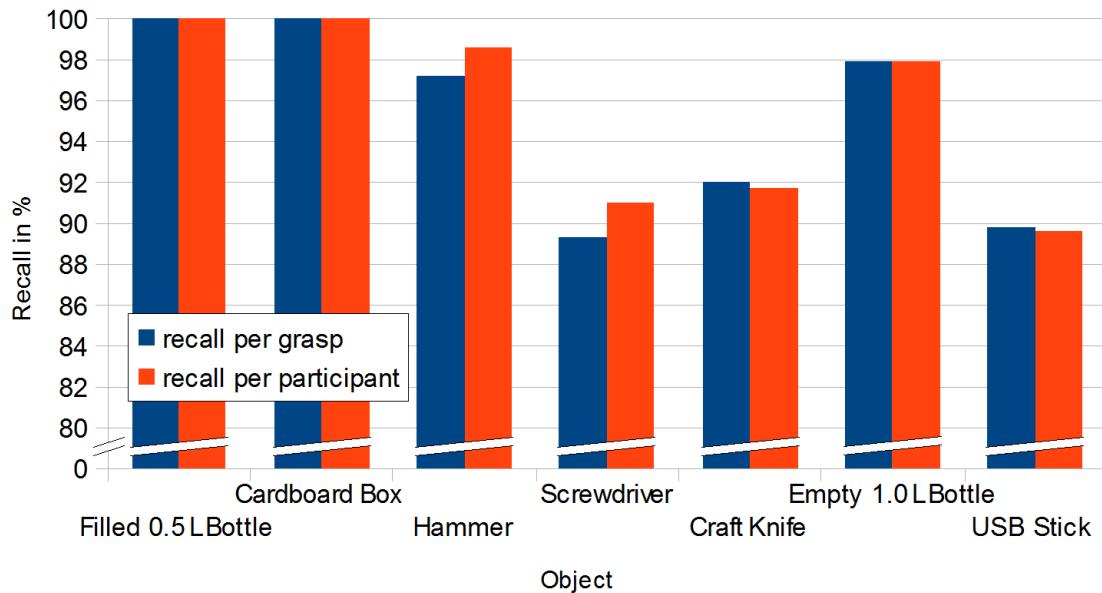


Figure 6.3.: Recall for the chosen objects for all participants. Since every grasp of the object have been counted and the grasping differed per participant, this value has varied. The blue bar illustrates the recall when every grasp counts even, while the red bar illustrates the recall when every participant has the same influence on the result.

6.3. Grasp Detection in Real-Time Results

The real-time tests yielded a harsh variety of different results. Participants, who were better instructed in the topic, usually achieved better results while the results of some participants were pretty bad in positives as well as in negatives.

The main reason for bad results was an unnatural behaviour. Most participants were trying to grasp very slow to see the results over the time. This behaviour was conflicting with the movement thresholds as well as with the implemented normalization of the muscle activation. Another unnatural behaviour was that participants were strengthening their grasp until they achieved a feedback.

A last problem could be that the calibration of the sensor on the arm was just too rough and the sensors were not detecting where they were ought to.

6.4. Grasp Recognition False Positive Grading Results

In this test, for a true positive the detections have to be in between 100 ms before to 200 ms after the grasps, which have to be detected on the objects. A false positive is called when a grasp has been detected 100 ms before starting to 200 ms after completing the task which should not be detected. It has been possible that during the task the hand was closed in a grasp.

A drop to 82.38 % in the recall have been found towards the box test. This is probably caused by the easy test setup, in which the participants are doing monotonously the same grasp three times for all objects in a row, without a strong variation of their position and movements and achieve an unnatural behaviour.

As false positive rate 78.48 % was achieved. This is indicating that the system does have a problem with too many false positives. But again, the chosen tasks were ought to be similar to grasping and a high sensitivity was chosen for parameters. Further extensions to decrease the false positives should be tested.

Precision have been at 51.29 %, with an equal distribution on positives and negatives.

Recall, false positive rate and precision per participant have been visualized in Figure 6.4. The detection rate per grasp and task is visualized in Figure 6.5. Detailed test results are listed in Appendix D.

A lower sensitivity has also been tried on Participant 2's test set. This test set have been chosen because it has the highest recall and number of overall detections. Two parameters have been changed in two runs. In the first run, the number of to be detected Layer 2 features was increased from nine of fourteen to eleven of fourteen. In the second run, the six Layer 2 time domain EMG features value's increase have been raised from 1.5x to 2x. Results are depicted in Figure 6.6.

6.4 Grasp Recognition False Positive Grading Results

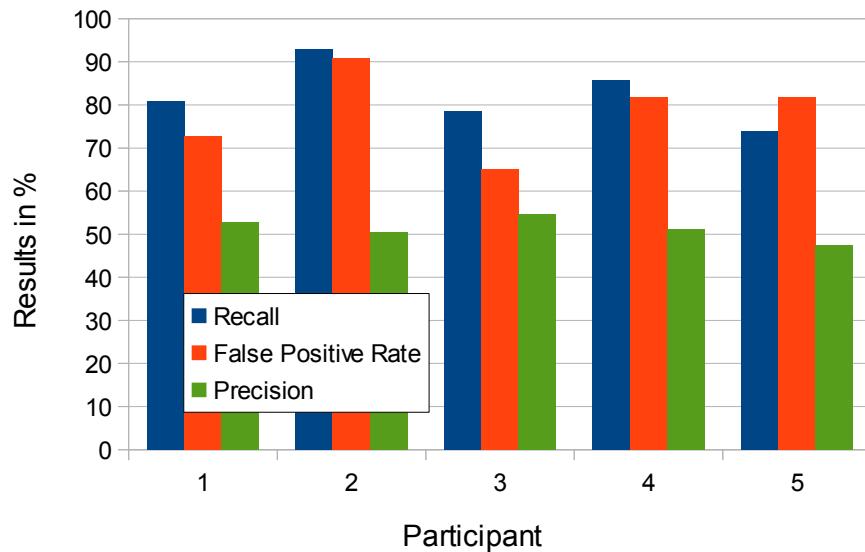


Figure 6.4.: Recall, false positive rate and precision of the false positive grading tests. Results are per participant.

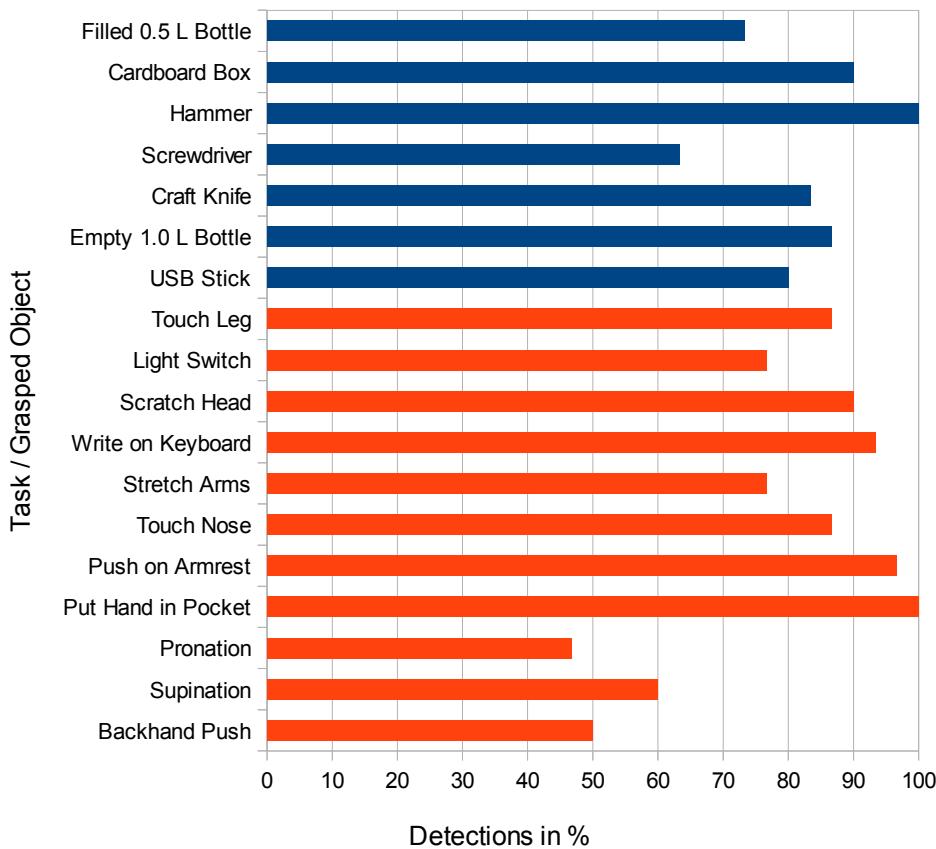


Figure 6.5.: Detections of the objects and tasks in the false positive grading tests.

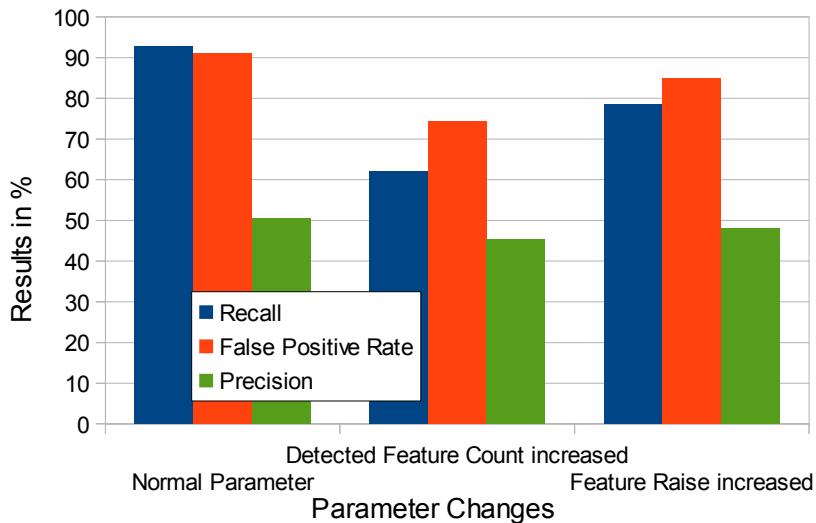


Figure 6.6.: Results of the false positive grading tests for Participant 2 with changed parameters.

6.5. Classifier Baseline Results

The classification of the step "Closing Hand around Object" has not worked out. Tests have been split on single EMG and single IMU (acceleration and angular velocity). The grasping class with EMG had a recall of less than 1 %, with a precision of 10 %. The null class had results close to 100 %, which was caused by the large class imbalance and is not representative. IMU had similar results.

However, the classification of "holding" actually had a quite well result for its naivety with a recall of 76.42 % and a precision of 74.05 % correct detections for the holding class. Moreover, the results are indicating that an ignoring of the borders between the class changes and a smoothing of the data would achieve in an even better result. Figure 6.7 is showing the recall and Figure 6.8 is showing the precision for nine of the twelve datasets which have been used in the grasp detection. The other three test sets have not been applicable because of a lack in video data.

In the test with the false positive grading logs, Participant 3's test set has been chosen as he achieved the highest precision/lowest false positives rate. The test, of course, was supposed to achieve worse results because the test set included a more difficult to distinguish null class. The recall dropped to 6.67 % with a precision of 31.10 % for the holding class. Null achieved a recall of 96.96 % and a precision of 83.45 %. Class imbalance has been 1:4.85.

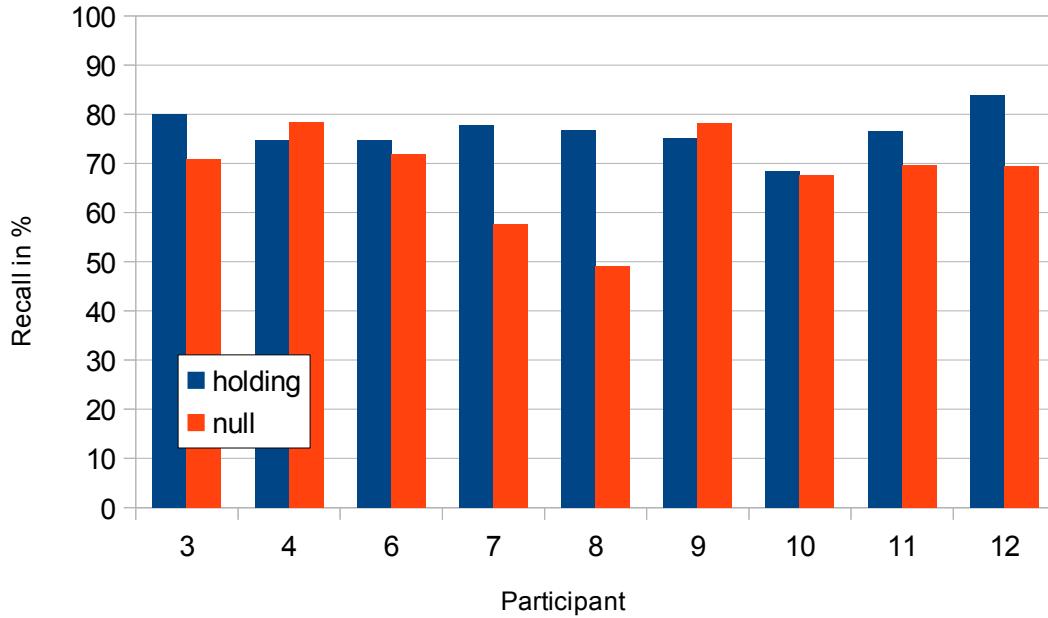


Figure 6.7.: Recall of the baseline tests with Random Forests classification. Datasets are equal to grasp detection with log file. Tested has been for the step "Holding Object". Null is represented everything else.

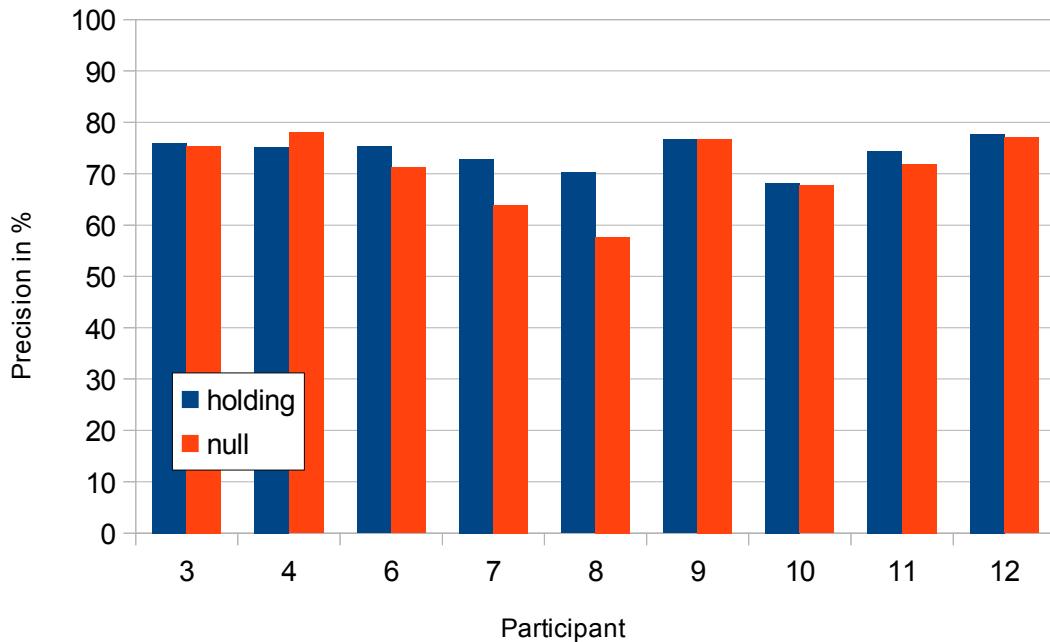


Figure 6.8.: Precision of the baseline tests with Random Forests classification. Datasets are equal to grasp detection with log file. Tested has been for the step "Holding Object". Null is representing everything else.

6.6. Object Detection with Grasp Detection Results

Grasp detections have been checked for the holding periods of the objects (as opposite to the former tests, where only a detection right at the touching was counted). Only once the grasp detection has failed within all four datasets. Failed object has been the USB stick. 94 grasps in total should have been detected.

Object detection has not worked out that well. Main reason for missed tags was the too short range of the RFID reader. A maximal distance of around 12 cm has been possible to bridge. The distance from the wrist to the tag has often been too far. Moreover, some absorptions have been possible by the hand in between, metal or water. For Participant 4 the tag of the crafting knife has been replaced from the middle to the end since no detection has been made yet. Besides the range, also a too early grasp detection (right before the grasp) or a too late (detection at letting go and latency) was possible. Sometimes the wrong tags have been detected because of the small space between the objects. 32 of the 94 objects have been missed completely.

Test sets have been around 314 s long. The sets have been pretty tight with less pauses between the tasks. Consequently, the number of random detections have been low. As random detections have been counted every detection which was not caused by one of the scripted tasks (also grasps). The normalization has not been counted as random detection. Participant 1 had one random detection, forced while repositioning the object detector. Participant 2 had seven random detections, in two different time slots. Participant 3 had six random detections in four different time slots. Two consecutive detections were caused by reposition the object detector. Participant 4 had twelve random detections: Ten consecutive detections by grasping armrest and pushing weight on it while sitting down and two consecutive detections by forming a fist.

In the mean, 304 times the grasp detection had a positive per set. This is nearly one per second. However, reason for the high number is also the tight set. Multiple detections of the same object can result from strengthening grasps, short regrasps, letting go of the object or, in seldom cases, fast movements. Mainly in the beginning of grasps some misdetections occurred because of too early grasp detections.

Complete test protocols are provided in Appendix D.

6.7. Energy Savings

The main purpose of this work was a reduction in the energy. Therefore, an investigation of the current consumption of state-of-the-art components (inbuilt in the prototype) is provided in this section. Energy consumption is intended to achieve by switching the devices in sleep mode while not used. The Bluetooth Low Energy

consumption has been neglected since in a finalized device the components probably are included in one device or are connected via cables.

For a reasonable energy consumption following estimations are made, based on available datasheets of the main components and with an operating voltage at 3 V:

- Accelerometer (InvenSense MPU9150): < 0.2 mA
- Gyroscope (InvenSense MPU9150): ~ 3 mA
- 8 channel EMG (based on ST 78589 VA1814): ~ 8 mA
- Processing (Freescale K21F MCU, ARM Cortex M4 based): ~ 7.7 mA
- RFID (SkyeTek Skyemodule M1-Mini): ~ 60 mA

Another example for a low energy EMG sensor is provided in [65]. An explanation for the estimated processing power is provided in Appendix A.

The energy consumption of the sensors is strongly increasing with the layers of the grasp detection model: Accelerometer (Layer 1) typically requires much less than 1 mA, the EMG sensor (Layer 2) requires around 8 mA, while the RFID reader (Layer 3) requires 60 mA.

In the early feature extraction of this work, one feature has been rather chosen from the gyroscope than from the accelerometer. Since the accelerometer data has in experiments led to similarly potent features as the gyroscope data, the 3 mA of the gyroscope can be neglected by using solely the accelerometer for motion features.

Layer 1 is based only on acceleration data. If Layer 1 does not see a chance for a grasping, further processing and the collection of EMG data is unnecessary and can be turned off. The time this neglection can be made is strongly depending on the activities of the user.

By thus neglecting the gyroscope's influence and assuming a half time turning off after Layer 1 (because of resting and walking) the system would come to an overhead of around 12 mA.

Energy consumption of RFID readers is mostly depending on the duty cycle and possible detection range. With a full duty cycle over short ranges, a nominal current consumption can reach over 100 mA. In [3], the antenna was shown to consume at least 18.23 mA with a low 1 Hz duty cycle and an average detection rate of 65 % for tags. With a full duty cycle of 16 Hz the reliability of tag reading was at 100 %, with a higher current consumption of 60 mA. Further improvements could include providing an increased detection range by using amplifiers between the RFID antenna and the reader, leading to larger units that require more energy. Moreover, the grasp detection can achieve a better timing than a low frequency duty cycle, which increases the reliability of the reads.

7. Conclusions

This chapter summarizes the most crucial results of this work and possibilities of further improvements.

7.1. Discussion

This work has shown in section 6.7 that it is possible to reduce the energy consumption of object detection by controlled switching on, even with an running grasp detection and additional sensors. The algorithm efficiency is estimated in Appendix A.

Results in section 6.2 show that especially for heavy objects, which require a "power grip", a high recall for the detection of the start of a grasp is possible. Results of section 6.6 show that also for "precision grips" mostly a grasp is detected while handling an object. The same section shows that the relatively short distances of the RFID readers can be a problem and more consuming readers could be necessary, which is a further argument for the grasp detection.

A drawback of this work is that a lot of false positives have been found in the grasp detection tests. Reason is the high sensitivity of the grasp detection, which was ought to achieve a high recall on small, fragile and light weight objects as well. Yet, an energy reduction is still possible.

In the tests it has been shown that the IMU alone is not reasonable to detect grasps because of relatively small values compared to random movements. Also the EMG alone has problems, especially on light weight objects. The three layer grasp detection model starts with the lowest consuming sensor and turns off the more costly sensors if possible. It provides an reasonable approach, which is able to reduce the energy overhead of EMG and calculations. Dividing of grasps into steps was establishing an information into the emerging of features and a possible sequencing, realized in Layer 3.

7.2. Future Work

As the distribution of the prototype on three different systems is not a reasonable approach in finalized devices, a single board device has to be implemented. This

would also reduce the BLE activity. Another IMU and EMG sensor has to be implemented as well as a more powerful microcontroller.

Other object detectors can be tested which would achieve a higher detection rate for objects. As example, a higher distance could be achieved by amplified (but higher-power) RFID readers.

Another approach that can be tested is a high frequent state model. For this, a clear distinction of the described steps has to be made. The actual grasp detection is mostly detecting the features of more than one step in a single sample period, which would require a higher sample rate, consequently with a higher processor and memory usage and a higher energy consumption. An idea would be to only relate on time domain features and preprocess the incoming signals. Eventually, a lot of the found Fourier domain features are still available by using a low pass filtered signal or a zero-crossing implementation instead of the discrete FFT. However, also in this approach it cannot be ensured that the features emerge in the guessed sequence.

An approach to tune a machine learning algorithm to detect the step "Holding Object" is also possible. Already good results have been reached in the naive baseline tests on the simple test set. However, both grasping steps, which are including arm movements, have not achieved promising results.

Further approaches can reduce the false positives. Examples for such an approach can be a disabling of a further grasp detection when a tag has been found. A procedure to find the step "Releasing Object" is also possible.

Acknowledgement

I want to express my gratitude to all members of the Embedded Systems department of the University of Freiburg, where this work has been written. Special thanks I want to express Prof. Dr. Kristof Van Laerhoven, as well as M.Sc. Philipp Scholl for advising my thesis, the helpful hints and support.

Also I want to express my gratitude for all the test participants for their willing to help and their patience.

The Albert Ludwigs University of Freiburg, the Hochschule Esslingen and the Eberhard Karls University Tübingen I have to thank for my academic training.

Lastly, I want to thank my family and friends for their support.

A. Algorithm Energy Efficiency

For roughly calculating the approximated minimum energy of the microprocessor for the grasp detection, the performance of the grasp processing algorithm on the microprocessor in an single device activity sensor is evaluated. The existing Matlab grasp detection is used for the calculation with some approximations. The algorithm is used without the Bluetooth Low Energy commands (reading, parsing, sending) and only using the running loop (no setup routines). Also some commands are not included in the calculation, since the only reason they are still in the code is for logging and demonstration purposes (like the calculation of Euler Angles). Some code can be neglected by exchanging calculations to constants or by sending data between the systems in another format.

Consumption has been evaluated for the MK21FN1M0VMC12 microcontroller, that has been inbuilt in the Myo. The processor got 120 MHz and is, according to its datasheet [68], achieving 1.25 Dhystone MIPS per MHz. This results in 150 MIPS in total. As approximation every Matlab operation have been given the same number of assembly instructions of 10. According to the ARM Cortex-M4 Technical Reference Manual [69], CPU cycles per logic operation can be approximated to 2. A single FFT and inverse fast Fourier transform (IFFT) has been approximated with 1000 operations each. As basis, operations in an optimized FFT and IFFT algorithm has been counted for 50 samples. The normalization has been approximated with 4 operations.

Incoming packets are calculated as shown in Table A.1, Table A.2 and Table A.3. With 50 Hz the acceleration data and with 100 Hz the EMG data are incoming. Incoming RFID reader data is approximated to be $\ll 1$ Hz and not further minded. Hence, a total of 62500 instructions per second are needed for the incoming sensor data.

The required calculation frequency of the features should not be too fast to save processing power but also should not be too slow in case of short grasps. Assuming a minimum grasping length of 650 ms, a maximal searching time for tags of 70 ms, a mode switching delay of the reader of 100 ms and a maximum of two calculations after the grasp, a delay between the calculations of 220 ms is appropriate. Therefore, the calculations of the features have been made with a frequency of 4.5455 Hz. The operation calculations of the grasp, walk and rest features are shown in Table A.4, Table A.5 and Table A.6. In Layer 1 275045 instructions per second are needed. Layer 2 also adds 231272 instructions per second. By far, here are the transformations the largest amount of operations.

Table A.1.: Operations for incoming acceleration data packets with 50 Hz.

Purpose	Operation	Amount	Instructions
Increase Data Counter	+	1	10
Scale data to Range	/	3	30
Calculate linear Acceleration	+	3	30
		3	30
	*	6	60
Save Data for later Processing	Save	9	90
			250

Table A.2.: Operations of incoming EMG data packets with 200 Hz.

Purpose	Operation	Amount	Instructions
Increase Data Counter	+	1	10
Calculate absolute Values	Abs	8	80
Save Data for later Processing	Save	16	160
			250

Table A.3.: Operations of incoming RFID data packets with $\ll 1$ Hz.

Purpose	Operation	Amount	Instructions
Save Data for later Processing	Save	9	90
			90

Table A.4.: Operations of the calculation for the grasping features of Layer 2 in 4.5455 Hz.

Purpose	Operation	Amount	Instructions
Calculating the Variance, Peaks and Mean Signal	Max, Min, Mean	12	120
Calculate the Normalization	Norm	11	440
Conversion into Frequency Domain	FFT	5	50000
Calculating the Mean	Mean	4	40
Compare with Threshold or Normalization	>	14	140
Save Data for later Processing	Save	14	140
			50880

Table A.5.: Operations of the calculation for the resting features of Layer 1 in 4.5455 Hz.

Purpose	Operation	Amount	Instructions
Calculate absolute Values	Abs	30	300
Calculating the Mean Signal	Mean	3	30
Compare with Threshold	>	3	30
Save Data for later Processing	Save	3	30
			390

Table A.6.: Operations of the calculation for the walking features of Layer 1 in 4.5455 Hz.

Purpose	Operation	Amount	Instructions
Conversion into Frequency Domain	FFT	4	40000
Conversion into Cepstral Coefficients	IFFT	2	20000
Calculating the Mean Signal	Mean	4	40
Compare with Threshold	>	4	40
Save data for later Processing	Save	4	40
			60120

The algorithm is ending with the temporal analysis (Layer 3) and informing the eventually starting of an object detection. Therefore, Table A.7 is summing the operations of the analysis, started after the feature calculation with 4.5455 Hz. The analysis requires 818 instructions per second.

Table A.7.: Operations of the temporal analysis with 4.5455 Hz.

Purpose	Operation	Amount	Instructions
Summing of the Activity Features	Sum	2	20
Checking for Detection	>	16	160
			180

Altogether, the algorithm is using 569635 instructions per second and only 337545 instructions when stopping after Layer 1. This is far below the possibility (150 MIPS) of the CPU. Taking the larger value and going for 2 CPU cycles per instruction, still only 1.139 MHz of 120 MHz would be used. Further, the 1.139 MHz are expected to be used for the energy calculation.

The MK21FN1M0VMC12 could work with 4 MHz in the "very-low-power run mode current at 3.0 V - all peripheral clocks enabled" mode for 1.88 mA. It could also operate for $\sim 1\%$ duty cycle and sleep for $\sim 99\%$ duty cycle. In "run mode - all peripheral clocks enabled, code executing from flash at 125 °C" 57.4 mA are used and "stop mode at 70 °C" is using 1.6 mA. Since "stop → run" (as well as "very low

power → run") is only requiring 4.4 µs, this configuration would be possible. For further processing, like the communication from the sensors to the microprocessor or communication with the object detector, the "wait mode, reduced frequency - all peripheral clocks disabled" with a typical 7.2 mA consumption and no switching delay is realistic. Calculating 99 % waiting current (7.2 mA), and 1 % running current (57.4 mA) would result in a total 7.7 mA consumption overhead for the grasp detection. By using the waiting periods for operations non-related to the grasp detection, the value of the overhead is further decreasing.

B. Bluetooth Low Energy Communication Commands

B.1. Hard- and Software Configuration

The current version of Matlab (2015a) has not implemented a full Bluetooth Low Energy communication yet, so it had to be necessary to implement it over a serial port. The Bluegiga BLED112 dongle had been used. A full 202 page long API [70] is available. In this configuration Matlab is used as master/scanner and the RFduino as well as the Myo are used as slaves/advertiser. The BLED112 dongle's standard configuration was used in this project. It allows a maximal connection of three slaves, which can be increased to eight.

B.2. Reading and Writing to Serial Port via Matlab

Mapping the serial port X to the variable s :

```
s=serial('COMX')
```

Open the serial port object saved in variable s :

```
fopen(s)
```

Close the opened serial port object:

```
fclose(s)
```

Reads all serial port objects from memory to Matlab workspace (useful to close all opened serial ports):

```
s=instrfind
```

Print information about available hardware connected to a serial port:

```
instrhwinfo('serial')
```

Print information about available Bluetooth hardware (only SPP protocol):

```
instrhwinfo('Bluetooth')
```

Write decimal values *byte1*, *byte2*, *byte3*, ... to the opened serial port object *s*:

```
fwrite(s, [ byte1 byte2 byte3 ... ] )
```

Read the data from serial port object *s* and places the pointer after the last read byte.

```
fread(s, size)
```

Size is optional and represents the number of bytes to be read. Standard size is 512 bytes and the timeout for collecting bytes while reading less bytes than size is a few seconds.

Flush the serial port object in case one does not want to read all the bytes sent to the serial port:

```
flushinput(s)
```

Pauses the script for *time* seconds:

```
pause(time)
```

A few milliseconds have to be paused between the sends of commands to Myo, else the second command will be ignored.

B.3. Packet Structure

The packet structure is shown in Table B.1.

Important Bluetooth 4.0 single mode messages with a typical payload length (byte 1) start with 0x00 command from host to stack, 0x00 response from stack to host, 0x80 event from stack to host.

Table B.1.: General packet structure of the BLED112 HCI Commands [70]

Octet	Octet bits	Length	Description	Notes
Octet 0	7	1 Bit	Message Type	0: Command/Response 1: Event
Octet 0	6:3	4 Bits	Technology Type	0000: Bluetooth 4.0 single Mode 0001: Wi-Fi
Octet 0	2:0	3 Bits	Length High	Payload Length (high Bits)
Octet 1	7:0	8 Bits	Length Low	Payload Length (low Bits)
Octet 2	7:0	8 Bits	Class ID	Command Class ID
Octet 3	7:0	8 Bits	Command ID	Command ID
Octet 4-n	-	0-2048 Bytes	Payload	Up to 2048 Bytes of Payload

Table B.2.: Command class IDs for the second octet in the BLED112 HCI Commands [70]

Class ID	Description	Explanation
0x00	System	Provides Access to System Functions
0x01	Persistent Store	Provides Access to Persistent Store (Parameters)
0x02	Attribute database	Provides Access to local GATT Database
0x03	Connection	Provides Access to Connection Management Functions
0x04	Attribute Client	Functions to access Remote Device GATT Database
0x05	Security Manager	Bluetooth Low Energy Security Functions
0x06	Generic Access Profile	GAP functions
0x07	Hardware	Provides Access to Hardware such as Timers and ADC

The command class IDs can be seen in Table B.2.

Addresses and GATTs have to be send in network byte order (big-endian), which means that the byte order have to be flipped when sending with Matlab.

B.4. General Commands

Start GAP discovery procedure:

00 01 06 02 01

Byte 5 is the GAP discover mode. 01 returns all slaves which are in discoverable mode.

Packet structure of scan responses can be seen in Table B.3

End GAP discovery procedure call:

00 00 06 04

The command should also be sent after connecting to a device, otherwise the dongle sends as response 0x0118: device not in state for command.

Connect to device

00 15 06 03 Addr6 Addr5 Addr4 Addr3 Addr2 Addr1 ...
00 06 00 06 00 64 00 00 00

Table B.3.: Packet Structure of a GAP Discovery Scan Response [70]

Byte	Type	Description
0	0x80	Message Type: Event
1	0x0B	Minimum Payload Length
2	0x06	Message Class: Generic Access Profile
3	0x00	Message ID
4	Int8	RSSI Value (dbm) Range: -103 to -38
5	Uint8	Scan Response Header 0: Connectable Advertisement Packet 2: Non Connectable Advertisement Packet 4: Scan Response Packet 6: Discoverable Advertisement Packet
6 - 11	Little Endian BT Address	Advertisers Bluetooth Address
12	Uint8	Advertiser Address Type 0: Public Address 1: Random Address
13	Uint8	Bond Handle if there is known Bond for this Device, 0xff otherwise
14	Uint8array	Scan Response Data

Addr6 till Addr1 are the little endian format of the Bluetooth address. Byte 10 can be 0x00 for public or 0x01 for random addresses. Byte 11 and 12 is the minimum connection interval and Byte 13 and 14 are the maximal connecting interval. Byte 15 and 16 are responsible for the timeout. Byte 17 and 18 represent how many connection intervals the slaves can skip.

Example for a successful connection response:

00 03 06 03 00 00 00

Byte 5 and 6 have to be zero for indicating an successful response. Byte 7 is the reserved connection handle.

Returning of all connections:

00 00 00 06

Response to returning of all connections is split into the response of maximal possible connections and connection status. An example for maximal possible connections response is seen here:

00 01 00 06 03

Byte 5 is the number of possible Connections. Connection status responses can be seen in Table B.4.

Table B.4.: Connection status of connected Bluetooth devices [70]

Byte	Type	Description
0	0x80	Message Type: Event
1	0x0B	Minimum Payload Length
2	0x06	Message Class: Connection
3	0x00	Message ID
4	Uint8	Connection Handle
5	Uint8	Connection Status Flag Bit 0: Connection exists Bit 1: Connection encrypted Bit 2: New Connection established Bit 3: Connection Parameters changed
6 - 11	Little Endian BT Address	Advertisers Bluetooth address
12	Uint8	Address Type 0: Public Address 1: Random Address
13 - 14	Uint16	Connection Interval (Units of 1.25 ms)
15 - 16	Uint16	Timeout (Units of 10 ms)
17 - 18	Uint16	Slave Latency
19	Uint8	Nonding Handle; 0xff if no stored Bonding Handle

B.5. Communication with Myo

Thalmic Labs published a header file with useful commands for the Myo Bluetooth communication [71].

Sending commands to Myo are structured in the following way:

```
00 payloadPacket 04 05 con GATTHandle2 GATTHandle1
    payloadCommand commandByte1 commandByte2 commandByte3 ...
```

Byte 1 till 5 are the HCI commands of the dongle. In the above case a write with acknowledge command. Byte 2 is the size of the payload of the packet (packet - 4). Byte 5 is the connection of the device on the dongle. Byte 6 and 7 are the GATT handle of the Myo service you want to address. Byte 8 is the payload size of the actual command to the service of the Myo. In the following only the sixth till the last Byte are described.

First thing to do is to tell Myo to send the filtered EMG data as well as the IMU data. Classification is unnecessary for our task:

```
25 00 05 01 03 02 01 00
    25 00 ... handle of the command characteristic
    05 ... payload
```

```

01 ... set mode
03 ... payload
02 ... send filtered EMG data (00 no, 03 raw)
01 ... send IMU data (00 no, 02 motion events, 03 both,
                     04 raw)
00 ... disable classifier (01 send classifier events)

```

Prevent Myo from sleeping:

```

25 00 03 09 00 01
25 00 ... command characteristic handle
03 ... payload
09 ... set sleep mode
00 ... payload (always 0)
01 ... never sleep (00 normal sleep)

```

Subscribing the IMU data service:

```
29 00 02 01 00
```

The EMG data service is split into four different handles. They send two different samples with every send. Subscribing EMG is done by following commands:

```

44 00 02 01 00
47 00 02 01 00
50 00 02 01 00
53 00 02 01 00

```

Force Myo to vibrate:

```

25 00 03 03 01 01
25 00 ... handle of the command characteristic
03 ... payload
03 ... vibrate
01 ... payload
01 ... short (02 medium, 03 long)

```

Receiving is established over following packet structure:

```

128 payloadPacket 04 05 con GATTHandle2 GATTHandle1
payloadData Data1 Data2 Data3 ...

```

IMU packets are sent over the handle 28 00. The data is structured with two bytes per entry as following:

```

PayloadData
Orientation w, x, y, z
Accelerometer x, y, z
Gyroscope x, y, z

```

The scales are 16384 for orientation, 2048 for accelerometer and 16 for gyroscope data.

EMG packets are sent over the handles 43 00, 46 00, 49 00 and 52 00. Data is send with one byte per entry. First eight bytes are belonging to the older reading, last eight bytes to the newer reading.

B.6. Communication with RFduino

RFduino uses the GATT handle 0x000E as sending DATA buffer and the handle 0x00 0x11 as receiving DATA buffer. To communicate with RFduino the "attribute write" (0x04 0x05), which is acknowledged by the remote end, is used. Also the "write command" (0x04 0x06), without an acknowledge, could have been used.

Subscribing the RFduino packets is possible via:

```
15 00 02 01 00 for notifications  
15 00 02 02 00 for indications  
15 00 02 03 00 for both
```

A send to RFduino looks in this work like this:

```
17 00 02 ID1 ID2  
17 00 ... handle receiving buffer  
02 ... payload  
ID1 ID2 ... ID of send
```

Received from RFduino uses the handle 14 00. Data in this work are structured like following, with one byte each:

```
Payload Data  
ID1 ID2  
Tag Type  
Tag1 Tag2 Tag3 Tag4 Tag5 Tag6
```

On RFduino side the following command is sending the tag:

```
RFduinoBLE.send( data , length )
```

Length is the byte length of the data array. Data include following, with 1 byte each:

```
ID1 ID2  
Tag Type  
Tag1 Tag2 Tag3 Tag4 Tag5 Tag6
```

When no tag is detected, only the ID will be sent.

Receiving is realized via the event handler

```
void RFduinoBLE_onReceive( char *data , int len )
```

Data is including the two bytes of the ID.

C. Test Scripts

C.1. Logged Grasp Detection and Object Detection Test Script

Explanation: Participant has to fulfil some tasks. Tools have to be handled only with the right arm.

Part 1: Synchronisation (video and data) and normalization (not used)

- Windmill 5x
- Circulate wrist 5x
- Form fists 5x
- Push down palm on table
- Push down back of the hand on table

Part 2: Standing tasks (grasps, which have to be detected)

- Walk to the full 0.5 L plastic water bottle and bring it to the table
- Drink something from the same bottle
- Walk to the cardboard box and bring it to the table
- Open the cardboard box
- Get the hammer out of the box, hammer a nail into a board, put the hammer back in the box
- Get the screwdriver out of the box, screw a nail into a board, put the screwdriver back into the box
- Get the crafting knife out of the box, cut a few times into a sheet of paper, put the crafting knife back into the box
- Get the empty 1 L plastic water bottle out of the box and put it on the table
- Get the USB stick of the box and put it on the table
- Put the USB stick back into the box
- Put the empty 1 L plastic water bottle back into the box
- Rotate the cardboard box, so that the opening is pointing towards the position the participant is sitting later (no barrier from the sidewall when grasping in)

Part 3: Sitting tasks (grasps, which have to be detected)

- Get the empty 1 L plastic water bottle out of the box and put it on the table
- Get the USB stick of the box and put it on the table
- Get the hammer out of the box, hammer a nail into a board, put the hammer back in the box
- Get the screwdriver out of the box, screw a nail into a board, put the screwdriver back into the box
- Get the crafting knife out of the box, cut a few times into a sheet of paper, put the crafting knife back into the box
- Put the USB stick back into the box
- Put the empty 1 L plastic water bottle back into the box
- Drink something from the filled 0.5 L plastic water bottle

Part 4: Synchronisation (video and data)

- Windmill 5x

C.2. False Positive Test Script

Explanation: Grasp every object or do every task three times and put back the hand in between.

Part 1: Standing recall tests (grasp objects)

- Cardboard box
- Screwdriver
- Hammer
- Filled 0.5 L plastic water bottle
- Empty 1 L plastic water bottle
- Crafting knife
- USB stick

Part 2: Standing false positive rate tests

- Touch leg
- Use light switch
- Scratch head
- Write a short sentence on a keyboard
- Pull back the arm sideways

C.2 False Positive Test Script

- Touch nose
- Sit down while propping up on armrest
- Put hand in trouser pockets and out again
- Twist forearm in a supination
- Twist forearm in a pronation
- Push a full, standing 0.5 L water bottle with the back of the hand

Part 3: Sitting false positive rate tests

- Lay down hand on leg
- Use light switch
- Scratch head
- Write a short sentence on a keyboard
- Pull back the arm sideways
- Touch nose
- Stand up while propping up on armrest
- Put hand in trouser pockets and out again
- Twist forearm in a supination
- Twist forearm in a pronation
- Push a full, standing 0.5 L water bottle with the back of the hand

Part 4: Sitting recall tests (grasp objects)

- Cardboard box
- Screwdriver
- Hammer
- Filled 0.5 L plastic water bottle
- Empty 1 L plastic water bottle
- Crafting knife
- USB stick

D. Test Protocols

Table D.2 and Table D.2 show the detailed results of the false positive tests.

In the Tables D.3, D.4, D.5 and D.6 detailed protocols of the object detection for every participant is provided. Used abbreviations for the tasks are random detection (RD), windmill (Wm), circulate wrist (CW), form fist (FF), push palm on table (PP), push back of the hand on table (PB), full 0.5 L water bottle (FB), box moving (BM), box opening (BO), hammer (H), screwdriver (S), craft knife (CK), empty 1.0 L water bottle (EB) and USB stick (US). Time, needed for the task, is provided in the second column. Column three is providing information about the number of positives in the grasp detection (true and false). Every 0.2 s an evaluation is performed. Column four is providing the number of the correct object detections. False object detections (reading the wrong tag) and comments to false positives are provided in the last column. Entries in the tables are listed by occurrence.

Table D.1.: False positive rate of the false positives test per participant.

	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Total		
	Stand	Sit	Stand	Sit	Stand	Sit		
Filled 0.5 L Bottle	3/3	2/3	3/3	2/3	1/3	3/3	10/15	12/15
Cardboard Box	3/3	2/3	3/3	3/3	3/3	2/3	3/3	14/15
Hammer	3/3	3/3	3/3	3/3	3/3	3/3	3/3	15/15
Screwdriver	3/3	3/3	2/3	2/3	1/3	1/3	2/3	9/15
Craft Knife	3/3	2/3	3/3	3/3	1/3	3/3	3/3	12/15
Empty 1 L Bottle	1/3	2/3	3/3	3/3	3/3	2/3	2/3	13/15
USB Stick	2/3	2/3	3/3	3/3	3/3	3/3	3/3	13/15
Total	18/21	16/21	20/21	19/21	15/21	18/21	16/21	87/105
								86/105

Table D.2.: Recall of the false positives test per participant.

	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Total		
	Stand	Sit	Stand	Sit	Stand	Sit		
Touch Leg	1/3	2/3	3/3	3/3	3/3	2/3	13/15	13/15
Light Switch	2/3	1/3	3/3	3/3	2/3	1/3	2/3	10/15
Scratch Head	3/3	3/3	3/3	3/3	1/3	3/3	3/3	13/15
Write Keyboard	3/3	2/3	3/3	3/3	2/3	3/3	3/3	13/15
Stretch Arms	3/3	1/3	3/3	3/3	2/3	1/3	3/3	9/15
Touch Nose	2/3	3/3	3/3	3/3	2/3	2/3	3/3	12/15
Push Armrest	3/3	3/3	3/3	3/3	2/3	3/3	3/3	14/15
Hand in Pocket	3/3	3/3	3/3	3/3	3/3	3/3	3/3	15/15
Pronation	2/3	3/3	2/3	1/3	1/3	1/3	0/3	7/15
Supination	1/3	2/3	1/3	2/3	3/3	2/3	1/3	10/15
Backhand Push	2/3	0/3	3/3	1/3	1/3	0/3	3/3	5/15
Total	25/33	23/33	31/33	29/33	23/33	20/33	28/33	24/33
								137/165
								122/165

Table D.3.: Object detection test protocol of Participant 1.

Task	Time/s	Pos	Tags	Remarks
Wm 5x	5.28	18		
CW 5x	2.60	10		
FF 5x	3.32	13		
PP	2.6	0		
PB	2.44	1		
FB	6.87	10	6	4 Tag Misses in beginning (too early, Range)
FB	5.92	12	11	1 Tag Miss in beginning (too early)
BM	8.65	8	5	3 Tag Misses in beginning (Range)
BO	5.92	10	4	6 Tag Misses (Range), multiple grasps
H	26.28	33	7	2 Tag Misses (too early), 24 Tag Misses (Range), 17 detections in 5.04 s hammering,
S	20.68	19	12	5 Tag Misses (Range) 12 detections in 8.56 s screwing
RD	9.5	1	0	repositioning sensor
CK	12.72	15	0	All Tag Misses (Range)
EB	1.84	4	3	1 Tag Miss (Range)
US	3.30	4	0	4 Tag Misses (Range)
EB	1.32	6	3	3 Tag Misses (Range)
US	2.24	2	1	1 Tag Miss (Range)
BM	3.28	5	1	4 Tag Misses (Range)
EB	1.6	6	0	All Tag Misses (Range)
US	2.28	8	0	All Tag Misses (Range)
H	23.96	22	20	2 Tag Misses (too early) 13 detections in 5.72 s hammering
S	18.76	19	8	11 Tag Misses (Range) 2x wrong detected Tag (closer)
CK	6.04	4	0	4 Tag Misses (Range) 1x wrong detected Tag (closer)
EB	1.36	4	2	2 Tag Misses (Range)
US	1.64	6	3	3 Tag Misses (Range)
FB	7.36	3	3	
Wm 5x	5.40	1		
Set	320.00	259	90	1 random detection

Table D.4.: Object detection test protocol of Participant 2.

Task	Time/s	Pos	Tags	Remarks
Wm 5x	6.64	1		
CW 5x	4.48	15		
FF 5x	4.32	10		
PP	1.92	4		
PB	1.44	0		
FB	7.28	10	4	6 Tag Misses (Range)
FB	6.72	4	3	1 Tag Miss in beginning (too early)
BM	8.46	7	0	All Tag Misses (Range)
BO	5.68	12	4	8 Tag Misses (Range)
H	13.64	6	6	detections in 2.92 s hammering
S	24.32	21	4	17 Tag Misses (Range) 13 detections in 7.96 s screwing
CK	13.56	17	0	All Tag Misses (Range) 3x wrong detected Tag
RD		4		
EB	4.20	5	2	3 Tag Misses (Range)
US	4.08	8	0	All Tag Misses (Range) 3x wrong detected Tag
EB	1.92	6	6	
US	1.96	0	0	no grasp detected
BM	3.28	13	0	All Tag Misses (Range)
EB	2.44	6	3	3 Tag Misses (Range)
US	3.12	6	0	All Tag Misses (Range) 1x wrong detected Tag
H	12.04	12	4	8 Tag Misses (Range) 1 detection in 2.48 s hammering
S	22.84	25	0	All Tag Misses (Range) 2x wrong detected Tag 17 detections in 4.08 s screwing
CK	10.2	8	0	All Tag Misses (Range) 1x wrong detected Tag
RD		3		
EB	2.4	9	3	6 Tag Misses (too early, Range)
US	1.32	7	0	All Tag Misses (Range)
FB	7.08	12	7	5 Tag Misses (too early, Range)
Wm 5x	5.30	0		
Set	303.00	276	60	7 random detections, 2x

Table D.5.: Object detection test protocol of Participant 3.

Task	Time/s	Pos	Tags	Remarks
Wm 5x	6.02	11		
CW 5x	4.6	14		
FF 5x	4.48	14		
PP	1.68	10		
PB	2.04	1		
FB	7.34	7	5	2 Tag Misses (too early, Range)
FB	3.56	5	1	4 Tag Misses (Range)
BM	9.70	7	0	All Tag Misses (Range)
BO	2.28	11	0	All Tag Misses (Range)
RD		2		
H	19.28	20	10	10 Tag Misses (too early/late, Range) 7 detections in 3.28 s hammering
S	24.04	25	0	All Tag Misses (Range) 21 detections in 7.48 s screwing
CK	14.32	22	0	All Tag Misses (Range)
EB	2.68	6	5	1 Tag Miss (too late)
US	3.40	7	0	All Tag Misses (Range)
US	1.08	3	0	All Tag Misses (Range)
RD		1		gesture, similar to pick up USB
EB	1.24	5	0	All Tag Misses (Range)
RD		1		
BM	8.12	11	0	All Tag Misses (Range)
RD		3		
BM	6.56	7	0	All Tag Misses (Range)
H	1.84	5	0	All Tag Misses (Range)
BO	0.60	3	0	All Tag Misses (Range)
EB	2.12	9	6	3 Tag Misses (too early/late, Range)
US	1.44	5	0	All Tag Misses (Range)
H	11.84	18	3	15 Tag Misses (Range) 8 detections in 3.44 s hammering
S	22.08	21	2	19 Tag Misses (Range) 5x wrong detected Tag 12 detections in 6.88 s screwing
RD		2		reposition object detector
CK	13.56	15	0	All Tag Misses (Range)
EB	2.22	2	0	All Tag Misses (Range)
US	1.58	4	0	All Tag Misses (Range) 1x wrong detected Tag
FB	3.68	5	0	All Tag Misses (Range)
Wm 5x	6.24	7		
Set	326.00	349	39	6 random detections, 4x

Table D.6.: Object detection test protocol of Participant 4.

Task	Time/s	Pos	Tags	Remarks
Wm 5x	5.36	15		
CW 5x	6.28	13		
FF 5x	5.44	13		
PP	1.52	6		
PB	1.56	0		
FB	9.30	9	2	7 Tag Misses (Range)
FB	3.68	5	0	All Tag Misses (Range)
RD		2		light fist
BM	8.00	12	0	All Tag Misses (Range)
BO	5.72	6	1	5 Tag Misses (Range)
H	10.36	18	6	12 Tag Misses (Range) 1x wrong detected Tag 7 detections in 1.96 s hammering
S	22.48	20	3	17 Tag Misses (Range) 1x wrong detected Tag 15 detections in 10.20 s screwing
CK	9.24	17	14	3 Tag Misses (Range, regrasp) 2x wrong detected Tag 12 detections in 3.44 s cutting
EB	3.04	9	7	2 Tag Misses (too late)
US	3.04	6	4	2 Tag Misses (Range)
US	1.88	5	1	4 Tag Misses (Range)
EB	2.00	9	3	6 Tag Misses (Range, too late)
BM	4.92	11	0	All Tag Misses (Range)
RD		10		sitting down, support grasp on armrest
EB	2.56	11	10	1 Tag Miss (too late)
US	2.40	6	0	All Tag Misses (Range)
H	10.48	18	6	12 Tag Misses (Range) 8 detections in 2.72 s hammering
S	14.72	14	1	13 Tag Misses (Range) 9 detections in 6.28 s screwing
CK	7.32	6	1	5 Tags Misses (Range)
US	2.04	10	2	8 Tag Misses (Range)
EB	2.12	7	1	6 Tag Misses (Range) 2x wrong detected Tag
FB	3.64	5	5	
Wm 5x	6.1	15		
Set	307.00	332	91	12 random detections (grasp)

E. Program Sources and Flowcharts

E.1. Object Detector Code

Program source, realized on the microcontroller, can be found in the "Object Detector" folder in the repository [72]. The driver for the RFID reader module can be found at [23]. Figure E.1 shows the flowchart of the program.

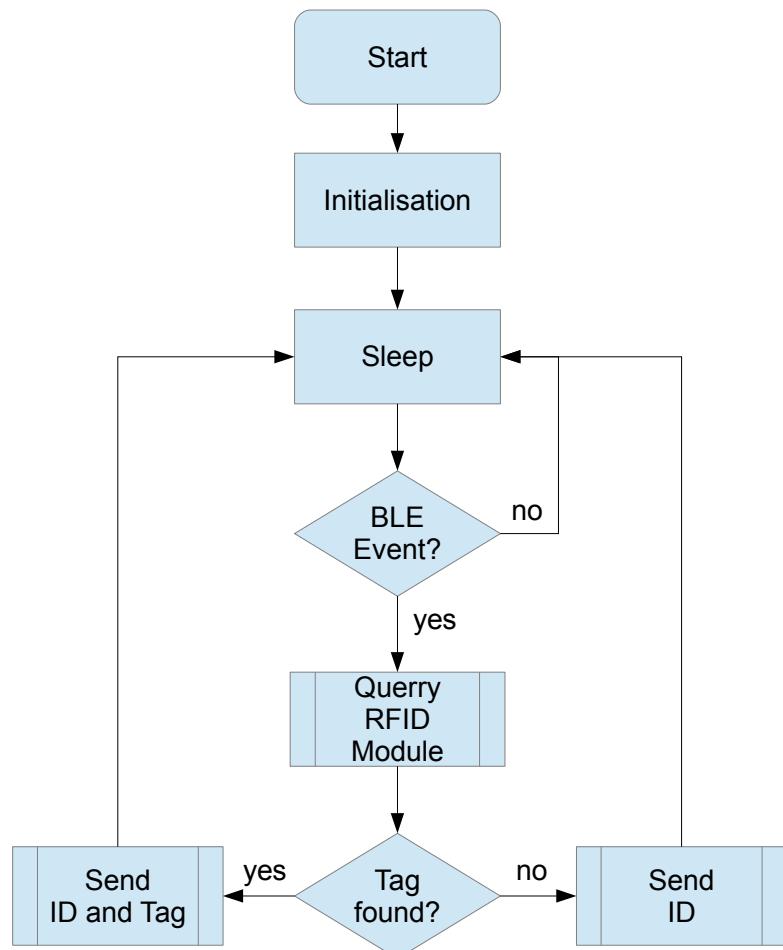


Figure E.1.: Flowchart of the microcontroller program.

E.2. Forearm Sensor Logger Code

Program source of the simple logger for the forearm sensor is available in the "Forearm Sensor Logger" folder in the repository [72]. It is communicating with the Myo Connect software and requires the Myo SDK [5]. Main part of the code is used from a project of the forearm sensor manufacturer [55]. The flowchart is seen in Figure E.2.

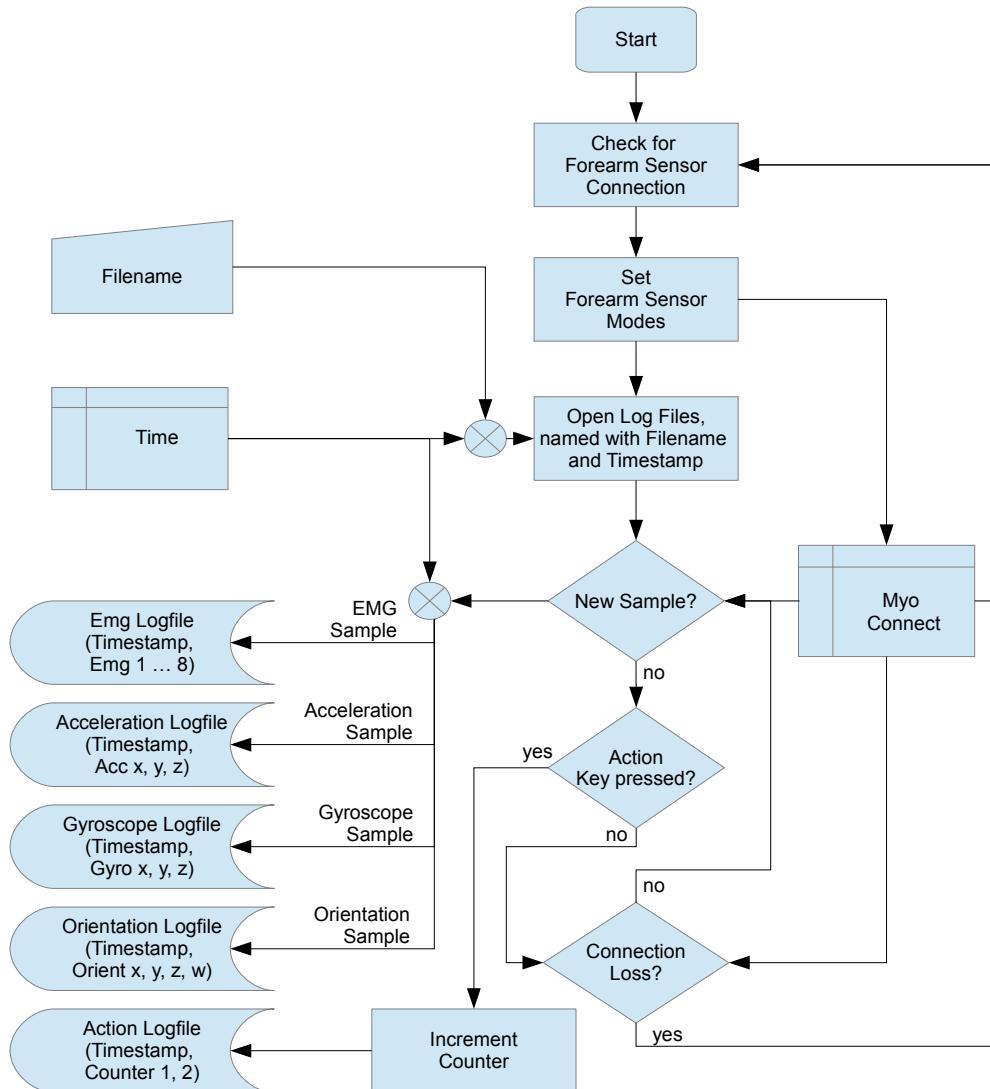


Figure E.2.: Flowchart of the simple forearm sensor logger, communicating with Myo Connect software.

E.3. Forearm Sensor Data Plotter Code

Source code for a simple visualizer for the raw forearm sensor data, without further manual processing, is available in the "Log Plotter" folder in the repository [72]. Figure E.3 shows the flow chart.

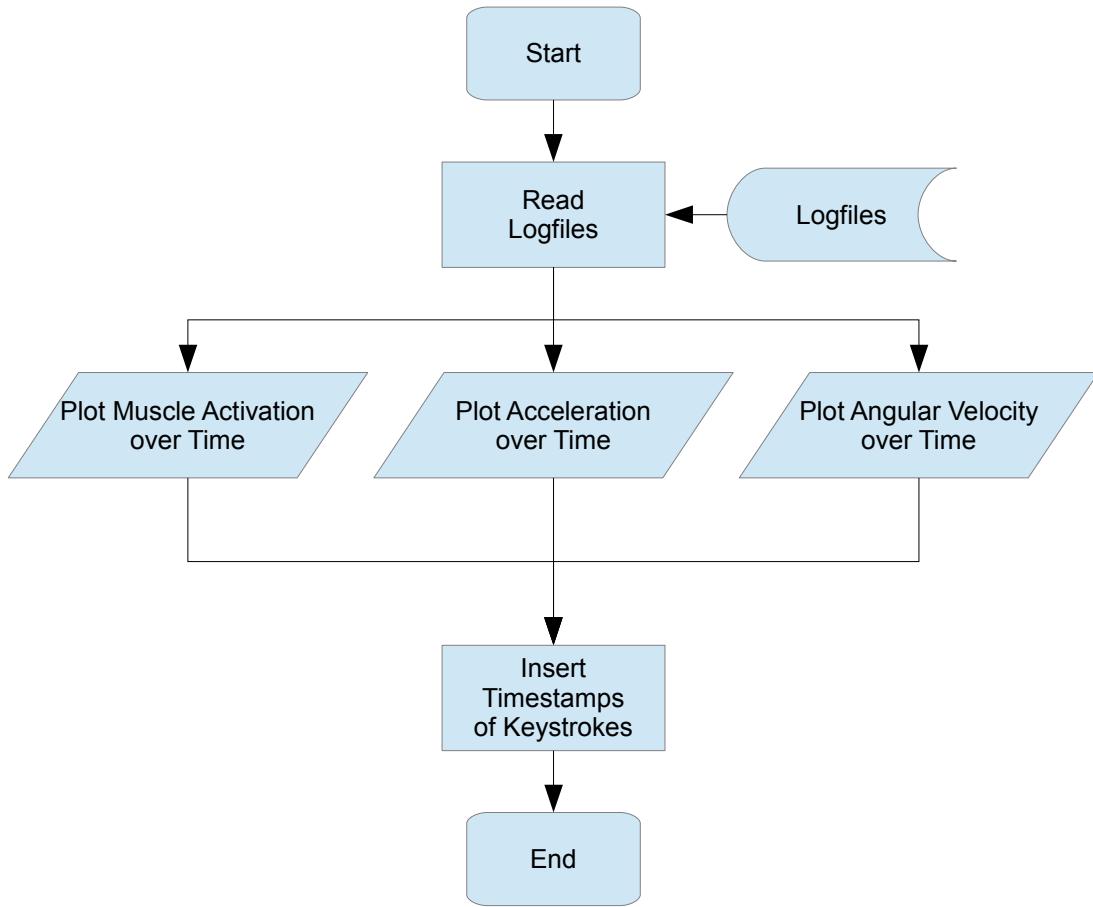


Figure E.3.: Flowchart of the raw forearm sensor data visualizer.

E.4. Feature Space Visualization Code

Program source to visualize the feature space of the grasp detection is available in the "Feature Space Visualizer" folder in the repository [72]. Inputs are logfiles, created with the simple logger, and an annotation of the timestamps between the steps per grasp. The flowchart is seen in Figure E.4.

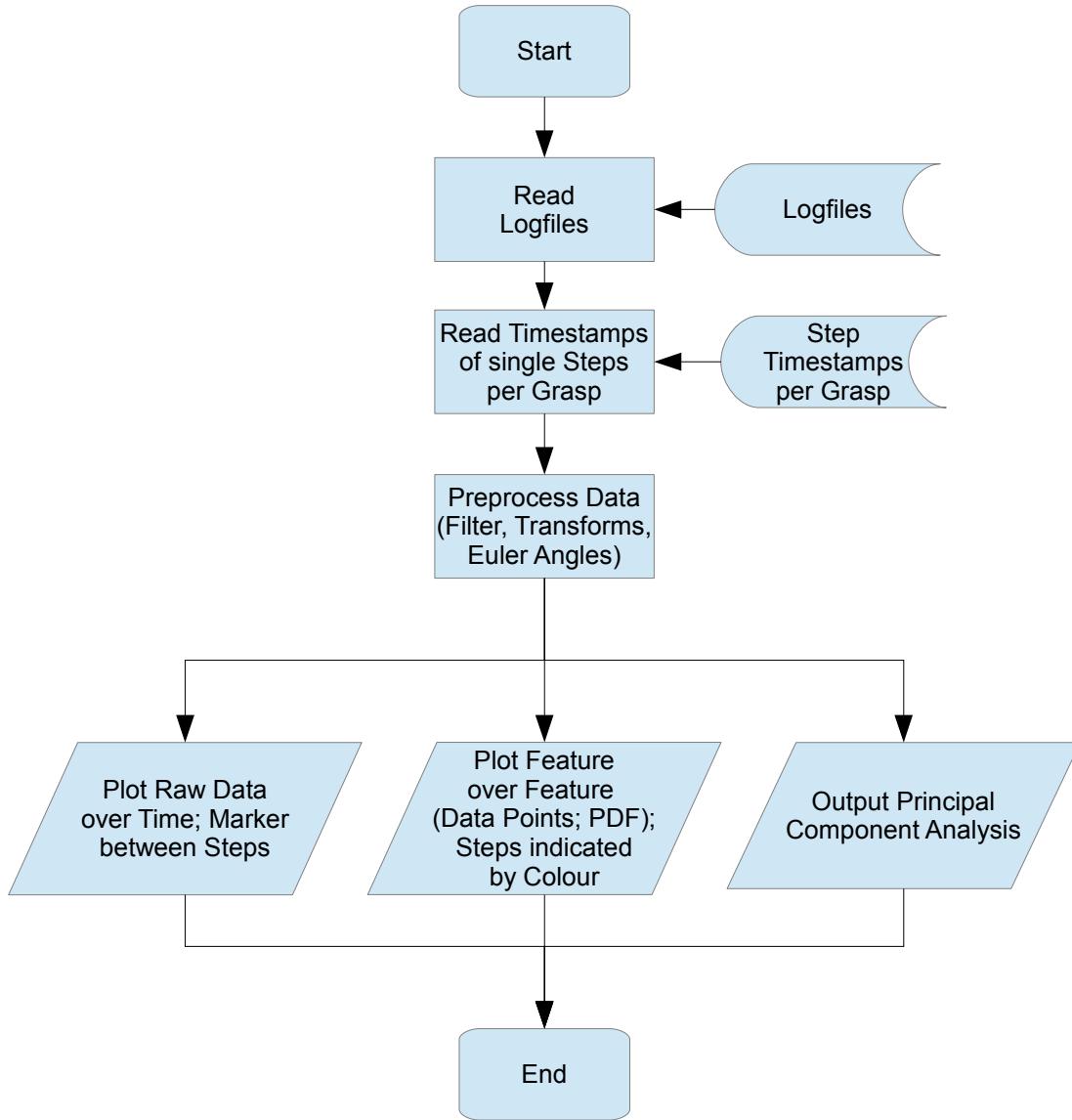


Figure E.4.: Flowchart of the feature space visualizer of the grasp detection.

E.5. Grasp Detection via Log Files Code

Program source for the grasp detection via logfiles can be found in the "Grasp Detection via Log Files" folder in the repository [72]. Figure E.5 is showing the flow chart of the grasp detection via log files.

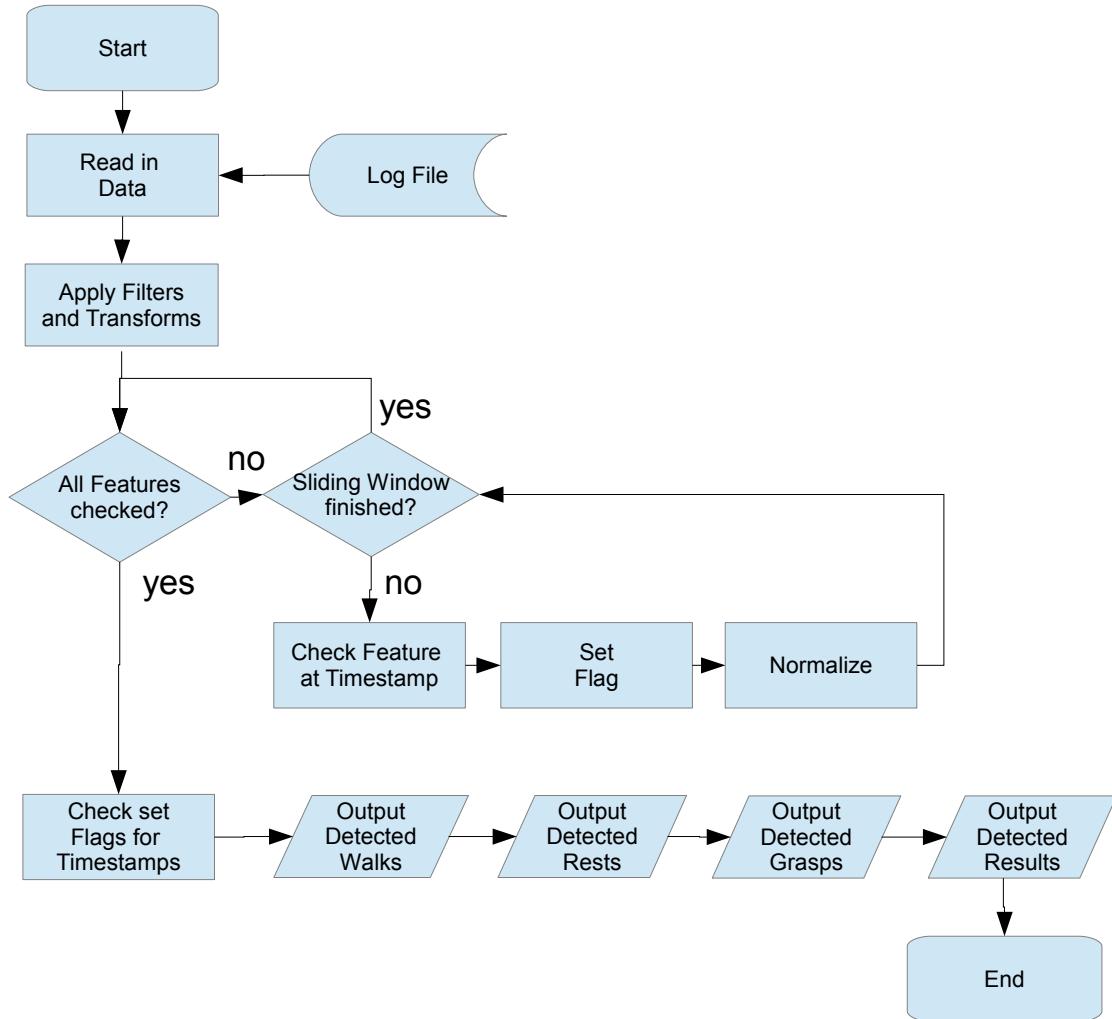


Figure E.5.: Flowchart of the grasp detection via log files.

E.6. Grasp Detection in Real-Time and Object Detection Code

The grasp detection in real time source can be found in the "Real-Time Grasp Detection" folder and the extension for the object detection can be found in the "Object Detection" folder in the repository [72]. Figure E.6 shows the flowchart for both programs. Green marked boxes are the extensions for the object detection.

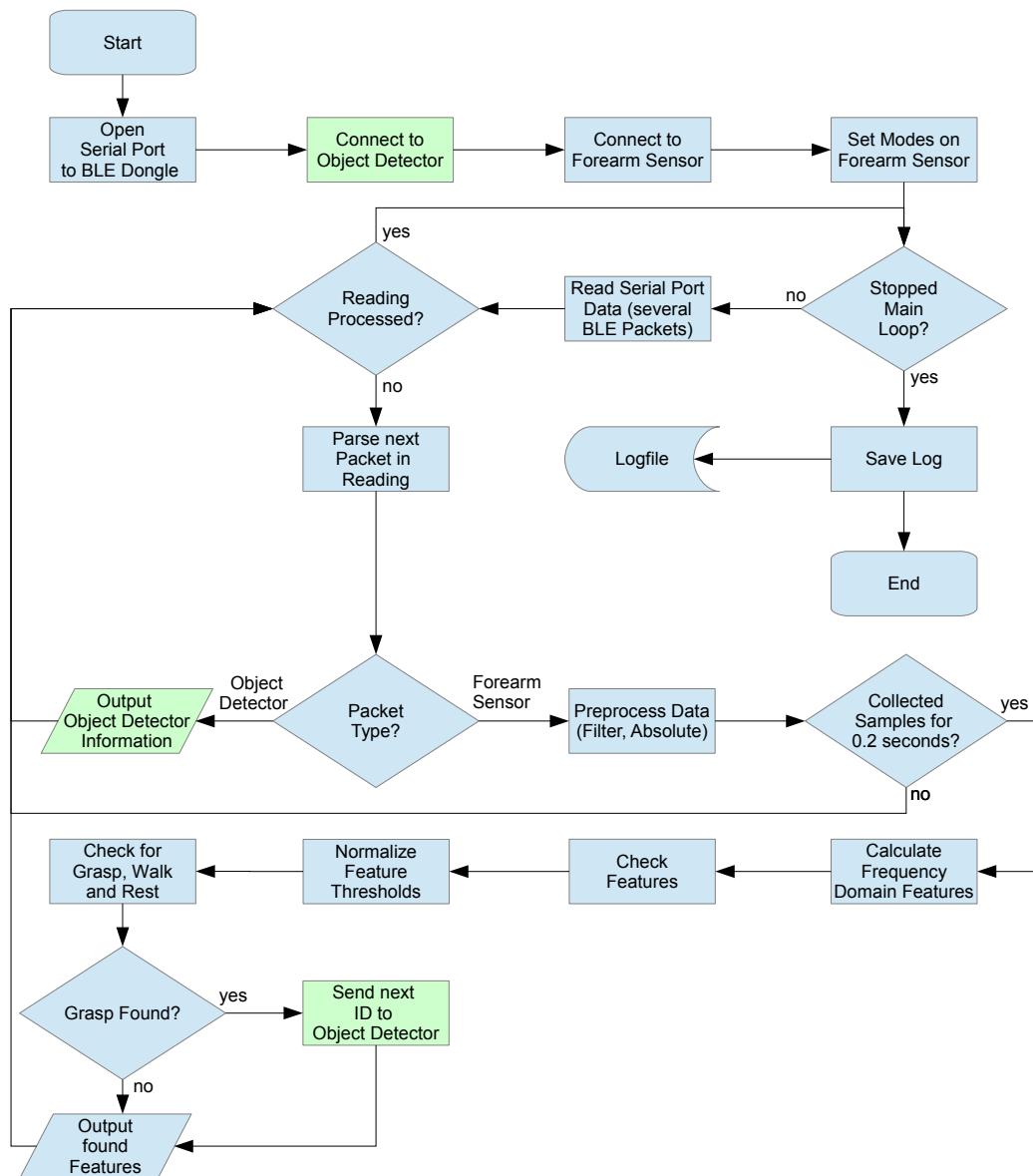


Figure E.6.: Flowchart of grasp detection in real-time. Green marked boxes are extensions for the object detection.

Bibliography

- [1] Eugen Berlin, Jun Liu, Kristof Van Laerhoven, and Bernt Schiele. Coming to Grips with the Objects We Grasp: Detecting Interactions with Efficient Wrist-Worn Sensors. In *International Conference on Tangible and Embedded Interaction*, pages 57–64, 2010.
- [2] Albrecht Schmidt, Hans-Werner Gellersen, and Christian Merz. Enabling Implicit Human Computer Interaction: A Wearable RFID-Tag Reader. In *Proceedings of the Fourth International Symposium on Wearable Computers*, pages 193–194, 2000.
- [3] Kenneth P. Fishkin, Matthai Philipose, and Adam D. Rea. Hands-On RFID: Wireless Wearables for Detecting Use of Objects. In *Ninth IEEE International Symposium on Wearable Computers*, pages 38–43, 2005.
- [4] Katrin Wolf and Jonas Willaredt. PickRing: Seamless Interaction through Pick-Up Detection. In *Proceedings of the 6th Augmented Human International Conference*, pages 13–20, 2015.
- [5] Thalmic Labs Myo. <http://www.myo.com>. 2015-11-12.
- [6] Sidney Katz, Austin Chinn, Lee Cordrey, Robert Grotz, William Newberry, Alexander Orfirer, Joan Wischmeyer, Alice Kelly, Ruth Mason, Margaret Ryder, Marion Bittman, Cleo Conley, Margery Hayward, Alma Hofferberth, Jean Holman, Lavina Robins, Mary Sherback, Susan Ritchie, and Stephen Takac. Multidisciplinary Studies of Illness in Aged Persons: II. A new Classification of Functional Status in Activities of Daily Living. *Journal of Chronic Diseases*, 9(1):55 – 62, 1959.
- [7] Sidney Katz, Amasa Ford, Roland Moskowitz, Beverly Jackson, and Marjorie Jaffe. Studies of Illness in the Aged. *JAMA: The Journal of the American Medical Association*, 185(12):914–919, 1963.
- [8] Powell Lawton and Elaine Brody. Assessment of Older People: Self-Maintaining and Instrumental Activities of Daily Living. *The Gerontologist*, 9(3):179–186, 1969.
- [9] Heidi McHugh Pendleton and Winifred Schultz-Krohn. *Pedretti's Occupational Therapy: Practice Skills for Physical Dysfunction*. Mosby Elsevier, Philadelphia, 7th edition, 2011.
- [10] Definition of ADLs at MedicineNet. <http://www.medicinenet.com/script/main/art.asp?articlekey=2152>. 2015-11-10.

- [11] Albert Hein and Thomas Kirste. A Hybrid Approach for Recognizing ADLs and Care Activities Using Inertial Sensors and RFID. In *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, volume 5615, pages 178–188. Springer Berlin Heidelberg, 2009.
- [12] Matthai Philipose, Kenneth P. Fishkin, Mike Perkowitz, Donald J. Patterson, Dieter Fox, Henry Kautz, and Dirk Hähnel. Inferring Activities from Interactions with Objects. *IEEE Pervasive Computing*, 3(4):50–57, 2004.
- [13] Takuya Maekawa, Yutaka Yanagisawa, Yasue Kishino, Katsuhiko Ishiguro, Koji Kamei, Yasushi Sakurai, and Takeshi Okadome. Object-Based Activity Recognition with Heterogeneous Sensors on Wrist. In *Pervasive Computing*, volume 6030, pages 246–264, 2010.
- [14] Leire Muguiria, Juan Ignacio Vázquez, Asier Arruti, and Jonathan Ruiz de Garibay. RFIDGlove: a Wearable RFID Reader. In *Proceedings of the IEEE International Workshop on Advances in RFID, at IEEE International Conference on e-Business Engineering*, pages 475–480, 2009.
- [15] Sreekar Krishna, Vineeth Balasubramanian, Narayanan Chatapuram Krishnan, Colin Juillard, Terri Hedgpeth, and Sethuraman Panchanathan. A Wearable Wireless RFID System for Accessible Shopping Environments. In *Proceedings of the ICST 3rd International Conference on Body Area Networks*, 2008.
- [16] Philipp M Scholl, Matthias Wille, and Kristof Van Laerhoven. Wearables in the Wet Lab: A Laboratory System for Capturing and Guiding Experiments. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 589–599, 2015.
- [17] MathWorks MATLAB. <http://mathworks.com/products/matlab>. 2015-11-12.
- [18] Bluegiga BLED112. <https://www.bluegiga.com/en-US/products/bled112-bluetooth-smart-dongle>. 2015-11-12.
- [19] RFduino. <http://www.rfduino.com>. 2015-11-12.
- [20] Skyetek Skyemodule M1 Mini. <http://www.skyetek.com/products/rfid/skyemodule-m1-mini>. 2015-11-12.
- [21] Arduino LLC. <http://www.arduino.cc>. 2015-11-12.
- [22] Arduino SRL. <http://www.arduinossrl.it>. 2015-11-12.
- [23] WRIFD. <http://github.com/pscholl/wrifd>. 2015-11-12.
- [24] GRTool. <http://github.com/pscholl/grtool>. 2015-11-12.
- [25] Nicholas Gillian and Joseph A. Paradiso. The Gesture Recognition Toolkit. *The Journal of Machine Learning Research*, 15(1):3483–3487, 2014.
- [26] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors. *ACM Computing Surveys*, 46(3), 2014.

- [27] Jonghwa Kim, Stephan Mastnik, and Elisabeth André. EMG-based Hand Gesture Recognition for Realtime Biosignal Interfacing. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, pages 30–39, 2008.
- [28] Anbin Xiong, Yang Chen, Xingang Zhao, Jianda Han, and Guangjun Liu. A novel HCI based on EMG and IMU. In *IEEE International Conference on Robotics and Biomimetics*, pages 2653–2657, 2011.
- [29] Timothy Forbes. Mouse HCI Through Combined EMG and IMU. Master thesis, University of Rhode Island, 2013.
- [30] Faizan Haque, Mathieu Nancel, and Daniel Vogel. Myopoint: Pointing and Clicking Using Forearm Mounted Electromyography and Inertial Motion Sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3653–3656, 2015.
- [31] Craig Taylor and Robert Schwarz. The Anatomy and Mechanics of the Human Hand. *Artificial limbs*, 2(2):22–35, 1955.
- [32] Georg Schlesinger. *Der Mechanische Aufbau der Künstlichen Glieder*. Springer, Berlin Heidelberg, 1919.
- [33] John Napier. The Prehensile Movements of the Human Hand. *The Journal of Bone and Joint Surgery (British Volume)*, 38-B(4):902–913, 1956.
- [34] Mark Cutkosky and Paul Wright. Modeling Manufacturing Grips and Correlations With the Design of Robotic Hands. In *1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 1533–1539, 1986.
- [35] Mark Cutkosky. On Grasp Choice, Grasp Models, and the Design of Hands for Manufacturing Tasks. *IEEE Transactions on Robotics and Automation*, 5(3):269–279, 1989.
- [36] Mark Cutkosky and Robert Howe. Human Grasp Choice and Robotic Grasp Analysis. *Dextrous Robot Hands*, pages 5–31, 1990.
- [37] Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron Dollar, and Danica Kragić. A Comprehensive Grasp Taxonomy. In *Robotics, Science and Systems: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation*, 2009.
- [38] Thomas Feix, Roland Pawlik, Heinz-Bodo Schmiedmayer, Javier Romero, and Danica Kragić. The GRASP Taxonomy of Human Grasp Types. In *IEEE Transactions on Human-Machine Systems*, pages 1 – 12, 2015.
- [39] Staffan Ekvall and Kragić Danica. Grasp Recognition for Programming by Demonstration. In *IEEE International Conference on Robotics and Automation*, pages 748 – 753, 2005.
- [40] Jun Liu. Evaluations for Long-term Activity Detection by Fusing RFID and Acceleration Data. Master thesis, TU Darmstadt, 2009.

- [41] Donald J. Patterson, Dieter Fox, Henry A. Kautz, and Matthai Philipose. Fine-Grained Activity Recognition by Aggregating Abstract Object Usage. In *Ninth IEEE International Symposium on Wearable Computers*, pages 44–51, 2005.
- [42] Beth Logan. Mel Frequency Cepstral Coefficients for Music Modeling. In *International Symposium on Music Information Retrieval*, 2000.
- [43] Gierad Laput, Chouchang Yang, Robert Xiao, Alanson Sample, and Chris Harrison. EM-Sense: Touch Recognition of Uninstrumented, Electrical and Electromechanical Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, pages 157–166, 2015.
- [44] Stuart McGill, Daniel Juker, and Peter Kropf. Appropriately placed Surface EMG Electrodes reflect Deep Muscle Activity (PSOAS, Quadratus Lumborum, Abdominal Wall) in the Lumbar Spine. *Journal of Biomechanics*, 29(11):1503 – 1507, 1996.
- [45] Toshiki Koshio, Shigeru Sakurazawa, Masashi Toda, Junichi Akita, Kazuaki Kondo, and Yuichi Nakamura. Identification of Surface and Deep Layer Muscles Activity by Surface EMG. *Proceedings of SICE Annual Conference (SICE)*, pages 1816 – 1821, 2012.
- [46] Heather Ma and Yuan-ting Zhang. Effects of the Physiological Parameters on the Signal-to-Noise Ratio of Single Myoelectric Channel. *Journal of Neuro-Engineering and Rehabilitation*, 4(1):1–10, 2007.
- [47] Tanja Schultz and Michael Wand. Biosignale und Benutzerschnittstellen, Biosignal: Muskelaktivität Entstehung, Messung (EMG), Anwendungen. Lecture Notes, 2012/2013.
- [48] Peter Konrad. EMG-FIBEL: Eine praxisorientierte Einführung in die kinesiologische Elektromyographie. Köln, 2005.
- [49] Roy Want. An Introduction to RFID Technology. *IEEE Pervasive Computing*, 5(1):25–33, 2006.
- [50] Charles Gomez, Joaquim Oller, and Josep Paradells. Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. *Sensors*, 12(9):11734–11753, 2012.
- [51] Leo Breiman. *Classification and regression trees*. Wadsworth International Group, Belmont, 1984.
- [52] Tin Kam Ho. Random Decision Forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1)*, pages 278–282, 1995.
- [53] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.
- [54] OpenSCAD. <http://www.openscad.org>. 2015-10-21.
- [55] Myo Logger by Thalmic Labs. <http://developerblog.myo.com/myocraft-logging-imu-and-raw-emg-data>. 2015-08-17.

Bibliography

- [56] DocCheck. <http://www.doccheck.com>. 2015-12-11.
- [57] TeachMeAnatomy. <http://teachmeanatomy.info>. 2015-12-11.
- [58] Karl Zilles and Bernhard Tillmann. *Anatomie*. Springer, Berlin Heidelberg, 1st edition, 2010.
- [59] Gerard Tortora and Bryan Derrickson. *Anatomie und Physiologie*. Wiley-VCH, Weinheim, 1st edition, 2006.
- [60] Kristof Van Laerhoven and Eugen Berlin. When Else Did This Happen? Efficient Subsequence Representation and Matching for Wearable Activity Data. In *International Symposium on Wearable Computers*, pages 101 – 104, 2009.
- [61] Rezwanul Ahsan, Muhammad Ibrahimy, and Othman Khalifa. Electromyography (EMG) Signal based Hand Gesture Recognition using Artificial Neural Network (ANN). In *4th International Conference on Mechatronics*, 2011.
- [62] Francesco Riillo, Lucia Quitadamo, Francesco Cavrini, Emanuele Gruppioni, Carlo Pinto, Nicola Cosimo Pastó, Laura Sbermini, Lorenzo Albero, and Giovanni Saggio. Optimization of EMG-based hand gesture recognition: Supervised vs. unsupervised data preprocessing on healthy subjects and transradial amputees. *Biomedical Signal Processing and Control*, 14:117 – 125, 2014.
- [63] John Basmajian and Carlo De Luca. *Muscles Alive: Their Functions Revealed by Electromyography*. Lippincott Williams and Wilkins, Philadelphia, 5th edition, 1985.
- [64] InvenSense Inc., <http://www.invensense.com>. *MPU-9150 Product Specification*, 4.3 edition.
- [65] Alireza Yousefian, Sebastian Roy, and Benoit Gosselin. A Low-Power Wireless Multi-Channel Surface EMG Sensor with Simplified ADPCM Data Compression. In *IEEE International Symposium on Circuits and Systems*, pages 2287–2290, 2013.
- [66] Skyetek, <http://www.skyetek.com/docs/mn/m1minidatasheet.pdf>. *Skye-module M1-Mini Datasheet*, 082515 edition.
- [67] Rüdiger Müller, Hans-Jörg Pfleiderer, and Karl-Ulrich Stein. Energy per Logic Operation in Integrated Circuits: Definition and Determination. *IEEE Journal of Solid-State Circuits*, 11(5):657–661, 1976.
- [68] Freescale Semiconductor, Inc., <http://freescale.com>. *Kinetis K21F Sub-Family Data Sheet*, rev 4 edition.
- [69] ARM Limited, <http://www.arm.com>. *Cortex-M4 Technical Reference Manual*, r0p0 edition.
- [70] Bluegiga, <http://www.bluegiga.com>. *Bluegiga Bluetooth Smart Software: V.1.2 API Documentation*, 3.1 edition.
- [71] Myo Bluetooth Low Energy Reference. <https://github.com/thalmiclabs/myo-bluetooth/blob/master/myohw.h>. 2015-08-17.

- [72] Program Sources of this Thesis on Github. <https://github.com/MarianTheiss/Efficient-Object-Detection-through-Grasp-Intention>.
2015-11-29.