

DLR-IB-RM-OP 2018-38

**Arm-Intent-Detection with Bio-Signals
in Virtual Reality**

Masterarbeit

Justin Iszatt



DLR

Deutsches Zentrum
für Luft- und Raumfahrt

MASTERARBEIT

**ARM-INTENT-DETECTION
WITH BIO-SIGNALS IN
VIRTUAL REALITY**

Freigabe:

Der Bearbeiter:

Unterschriften

Justin Iszatt



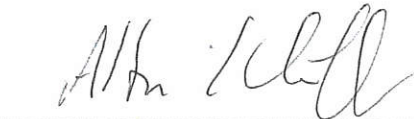
Betreuer:

Mathilde Connan



Der Institutsdirektor

Prof. Alin Albu-Schäffer



Dieser Bericht enthält 80 Seiten, 29 Abbildungen und 3 Tabellen



TECHNISCHE
UNIVERSITÄT
MÜNCHEN

Department of Informatics

Master's Thesis in Informatics

Arm-Intent-Detection with Bio-Signals in Virtual Reality

Justin Iszatt





TECHNISCHE
UNIVERSITÄT
MÜNCHEN

Department of Informatics

Master's Thesis in Informatics

Arm-Intent-Detection with Bio-Signals in Virtual Reality

Arm-Intentionserkennung durch Bio-Signale in der virtuellen Realität

Author: Justin Iszatt

Supervisor: Prof. Dr. Alin Albu-Schäffer

Advisor: Dipl.-Ing. Mathilde Connan

Submission date: 15.03.2018

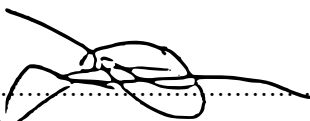


Statutory Declaration

Ich versichere, dass ich diese Masterarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich 15.03.2018:


.....
(signature Justin Iszatt)

Abstract

Looking at a huge number of amputees experiencing neuropathic phantom pain limb, this thesis presents the VITA 2.0 platform, targeting phantom pain relief by implementing the classical mirror therapy in virtual reality (VR).

Broadly speaking the work surveys myography data collection, interpretation towards arm intention detection and visualisation of sEMG data in a virtual world. It also presents a way of low-cost, occlusion invariant arm constellation tracking to control the pose of a virtual arm.

Apart from developing a new and versatile platform, a set of user focused experiments was conducted demonstrating the influence of the arm constellation for a successful intention detection. Whilst targeting healthy participants at this stage, future endeavours are set to include experiments with trans-radial amputees.

Keywords: surface Electromyography, machine learning, intention prediction, virtual reality, bio-signals

Kurzfassung

Mit dem Blick auf die große Zahl Amputierter die mit denen als Phantomschmerz bekannten Komplikation zu kämpfen haben, präsentiert diese Arbeit das VITA 2.0 System, das die meist genutzte Therapieform, die Spiegeltherapie, in das Medium der virtuellen Realität portiert.

Die Arbeit beschäftigt sich dabei im Detail mit der Datenaufzeichnung und Analyse hinsichtlich der Arm Intention sowie der Visualisierung selbiger in einer virtuellen Welt. Des Weiteren wird eine kostengünstige und verdeckungsunabhängige Möglichkeit aufgezeigt, mit der sich die Armkonstellationen messen und in der virtuellen Realität übertragen lässt.

Abgesehen von der Erstellung dieser neu- und vielseitigen Plattform, wurden nutzerbezogene Experimente durchgeführt, die den Einfluss der Arm-Konstellation auf die erfolgreiche Intentionserkennung aufzeigen.

Während im Fall der Arbeit ausschließlich gesunde Probanden untersucht wurden, so sehen zukünftige Pläne Nutzertests mit trans-radial Amputierten vor.

Schlüsselwörter: Oberflächen Elektromyografie, Maschinelles Lernen , Intentionserkennung, virtuelle Realität, Biosignale

Acknowledgements

Many people contributed to the success of this thesis at DLR Oberpfaffenhofen, so many in fact, that I am likely going to forget some.

However Mathilde Connan, my advisor; Dr. Claudio Castellini, head of the adaptive biointerfaces group; Prof. Dr. Alin Albu-Schäffer, my supervisor as well as Johannes Engelsberger, a good friend, deserve particularly to be mentioned. A big Thank You also to all colleagues in the biointerfaces group; all students I have been in touch with and everybody else, who I might have forgotten.

Last but not least, I would like to thank my friends and my family, who were there when I needed them.

THANK YOU!!

Justin Iszatt

Contents

List of Figures	V
List of Tables	VIII
Index of Abbreviations	VIII
1 Introduction	1
1.1 Goals and challenges of VITA 2.0	2
1.2 A brief look at the following chapters	3
2 Basics	5
2.1 Amputation	5
2.1.1 Mirror therapy	6
2.2 Surface Electromyography - sEMG	6
2.3 Virtual reality	7
3 Literature Survey	10
4 Material and Methods	13
4.1 Off-the-shelf hardware	14
4.1.1 The HTC VIVE Virtual Reality System ¹	14
4.1.2 The Alienware laptop	15
4.1.3 The MYO armband ²	15
4.2 Software and Toolkits	18
4.2.1 Unity3D	18
4.2.2 Other tools	19
4.2.3 Interactive Myocontrol	21
4.3 A new arm for VR	22
4.4 Arm positional tracking	24
4.4.1 Creating a custom tracker	25
4.5 Experiment design	29
4.6 Brief look at the resulting platform	34
5 Results	36
5.1 Tracking	37
5.2 Participant data	38
5.2.1 Results represented over action	39
5.2.2 Results represented over positions	42
6 Discussion	44
6.1 The tracking	44
6.2 Experiment results	45

7 Conclusion	47
7.1 Conclusion	47
7.2 Outlook	48
8 Bibliography	52
A Appendix	I

List of Figures

Fig. 1	The three main steps of this work: platform setup, experiment design, data acquisition and analysis	XII
Fig. 2	The three parts of this thesis: platform setup, experiment design and Data acquisition and analysis.	3
Fig. 3	Trans-radial amputation and its effect on rotation ability of the lower arm and wrist. ³	5
Fig. 4	Representing the first big part of this thesis, the platform setup, chapter 4 depicts used materials and tools for its implementation.	13
Fig. 5	The hardware used in this setup: 1: Vive Lighthouses ; 2: Vive controller ; 3: Vive Tracker; 4: HTC Vive Headset; 5: Dell Alienware 15 R3; 6: The created hardware combining one HTC Vive Tracker, a MYO armband as well as the created custom tracker.	14
Fig. 6	<i>On the left:</i> The MYO armband by Thalmic Labs and its bluetooth dongle. <i>On the right:</i> Teardown of the device with 1: sEMG-sensor with operational amplifier board attached to it; 2: detached operational amplifier board; 3: two LiPo-batteries; 4: main board with a MK22FN1M0 main processor, a NRF51822 BLE-module and a Invensense MPU-9150 9-DOF motion sensor; 5: plastic casing ⁴	16
Fig. 7	Unity3D in its standard layout. From the left to the right: The game view with a live preview of the game running; The scene editor view, where game objects can be placed, resized and rotated; The scene hierarchy, showing all game objects and how they are (inter-)connected; The object inspector showing the properties of a selected game object, including their attached scripts; The debug console for displaying debug and error messages; Lastly the file browser for managing assets.	19
Fig. 8	The in-house developed Interactive Myocontrol software acquiring data from a MYO armband.	21
Fig. 9	The full transition from MakeHuman over Blender to Unity3D. From left to right: 1: Creating a rigged human avatar in the MakeHuman software. 2: Deleting all but the arm of the avatar in Blender. 3: Resulting model in blender. 4: Ready to use textured arm in Unity3D.	23
Fig. 10	Animation of the virtual hand. <i>On the left:</i> The four actions of interest in VR and reality. <i>On the right:</i> Unity Blend Tree for the hand and wrist movement in the Animation Controller of the arm, allowing far more poses than the ones presented on the left.	24
Fig. 11	Drawing of supi- and pronation of the hand with muscle and skeletal constellation ⁵	25

Fig. 12	Low-cost gyroscopes and their zero rate level [in degrees per second (dps)] on the left ⁶ and the used <i>Adafruit Precision NXP 9-DOF Breakout Board</i> on the right.	26
Fig. 13	1: Circuit for the custom created tracker based on an Adafruit Feather nRF52 Bluefruit and the Adafruit Precision NXP 9-DOF Breakout Board ⁷ . 2: Prototype setup on an arm (top figure), followed by a rendering of the 3D-printed setup (bottom figure). 3: Final setup attached to the user's arm. Combining the MYO armband, the here created custom tracker and the HTC Vive tracker, a multi modal hardware platform was created, capable of recording orientation data of three independent sensors.	27
Fig. 14	As platform setup was presented in the preceding subsections, part 1 of the thesis is finished. This chapter marks the beginning of part two of this thesis: the experiment design.	29
Fig. 15	Experiment setup and computer renderings of a participant wearing the arm-tracking device during an experiment. The graph in the middle shows the used point cloud as a 2D plot, including the training position in the middle and the three groups: 1: "upper group", 2: "middle group" and 3:"lower group", consisting of 3, 2 and 3 points respectively.	30
Fig. 16	Creation of a participant specific experiment-set. Starting with a random subtask-sequence -abbreviated seq.-, the follow up sequences are generated by shifting the sequence by two indices. The right part of the figure not only depicts the composition of Tasks and TaskSets, but also shows how Task 1 to 4 are interchanged in between TaskSets.	31
Fig. 17	Experiment-set creation for all participants based on the latin squares of a random sequence and its follow ups on the left, with the final, comparable and participant specific experiment-sets on the right. . . .	32
Fig. 18	The immersive VR setup on the left compared to its real life counterpart, an office in DLR, Oberpfaffenhofen on the right.	34
Fig. 19	The overall software structure in one picture: On the left: Inter program communication and software structure of the setup. On the right: The Unity3D part of the setup and its (sub)modules. . .	35
Fig. 20	This section marks the beginning of the last part of the thesis: data acquisition and analysis.	36

⁶<https://learn.adafruit.com/comparing-gyroscope-datasheets/overview>, [10.03.2018]

Fig. 21 Data excerpt of the three tracking systems in quaternions. The Vive Tracker is displayed by a dashed red line, the MYO armband's orientation-data is presented in blue. Lastly, the green line represents data of our custom created tracker. While ground truth and MYO data are very similar, data of the custom created hardware shows a strange, "jumpy" behaviour, being even more present when displaying the data in VR. 37

Fig. 22 Plot of Vive Tracker (dashed red line) and MYO armband orientation data (blue) in quaternions over time. Given data being closely related, the MYO's data was finally used to control the virtual arm. 38

Fig. 23 Plot of the overall task success rate over the given actions. The x-axis resembling the actions 'rest', 'power', 'point', 'tridigital' accordingly, with the y-axis representing the success rate in a scale from 0 to 1. . . 40

Fig. 24 Plot of the Task Completion Time [TCT] for successful tasks over the given actions. The x-axis resembling the actions 'rest', 'power', 'point', 'tridigital' accordingly, with the y-axis representing the TCT in seconds. 41

Fig. 25 Plot of the accumulated Time In Task [TIT] for failed tasks over the given actions. The x-axis resembling the actions 'rest', 'power', 'point', 'tridigital' accordingly, with the y-axis representing the TIT in seconds. 41

Fig. 26 Plot of the overall task success rate over the given positions. The x-axis resembling the position groups 'upper group' [position 0, 3 ,6], 'middle group' [position 2, 5], 'lower group' [position 1, 4 ,7] accordingly, with the y-axis representing the success rate in a scale from 0 to 1. 42

Fig. 27 Plot of the Task Completion Time [TCT] for successful tasks over the given positions. The x-axis resembling the position groups 'upper group' [position 0, 3 ,6], 'middle group' [position 2 ,5], 'lower group' [position 1, 4 ,7] accordingly, with the y-axis representing the TCT in seconds 43

Fig. 28 Plot of the accumulated Time In Task [TIT] for failed tasks over the given positions. The x-axis resembling the position groups 'upper group' [position 0, 3 ,6], 'middle group' [position 2 ,5], 'lower group' [position 1, 4 ,7] accordingly, with the y-axis representing the TIT in seconds 43

Fig. 29 Graph of possible network-based-visualisation on different devices and platforms. With the UDP data protocol already in use, every device in the same network as the machine running Interactive Myocontrol can be reached, making data representation on PCs, tablets and smartphones possible. Without too many changes, VITA 2.0 could run on a Google Cardboard or any other inexpensive set of phone powered VR or AR googles. 50

List of Tables

Tab. 1	Excerpt of the task logging file for participant no. 3. Consisting of 128 sub-tasks in the according number of lines, the file is composed of four taskSets à 32 sub-tasks each. The first column holds the subtask-number [Task] followed by the taskSet-number [TaskSet], the desired target [Target] and action [Action] with a boolean value encoding success or failure of the sub-task [Success]. Task Completion Time [TCT] with the maximum time for each task [maxTT], the time of reach [TOR] as well as the subtraction of the TCT and TOR [TCT-TOR] and the Time In Task [TIT] form the second block.	39
Tab. 2	Average success rate of actions for the eight participants (Part1-8), with the mean action-success-rate over all participants represented in the last column.	40
Tab. 3	Average success rate at locations for all participant with the mean success rate over all participants in the last column.	42

Index of Abbreviations

AHRS	Attitude and heading reference system
API	Application programming interface
AR	Augmented Reality
BLE	Bluetooth Low Energy
DLL	Dynamic Link Library
DLR	The German Aerospace Center - German: Deutsches Zentrum für Luft- und Raumfahrt e.V.
DOF	Degree of Freedom
FDM	Fused Deposition Modeling (3D printing method)
GUI	Graphical user interface
HMD	head mounted display
IMU	Inertial measurement unit
LLSQ	Linear Least Square
MT	Mirror therapy
PLA	Poly lactid
PLP	Phantom limb pain
SD(-Card)	Secure Digital-format
SDK	Software development kit
SSD	Solid State Drive
STL	Stereolithography
TCT	Task Completion Time
TIT	Time In Task
TOR	Time Of Reach
VR	Virtual Reality
dae	Collada FileFormat
Gyro	Abbreviation for gyroscope

Outline of the Thesis

Part 0 : Introduction & Theory

In the first part of this thesis, I would like to focus on the main objectives of this project. An outline of most of the prerequisite knowledge required will also feature in this part. The structure is as follows:

Chapter 1: Introduction

This chapter will give a brief introduction to the subject of this thesis.

Chapter 2: Background

This chapter revises the basics of Virtual reality and Electromyography.

Chapter 3: Literature Survey

This chapter will turn the reader's gaze towards similar work.

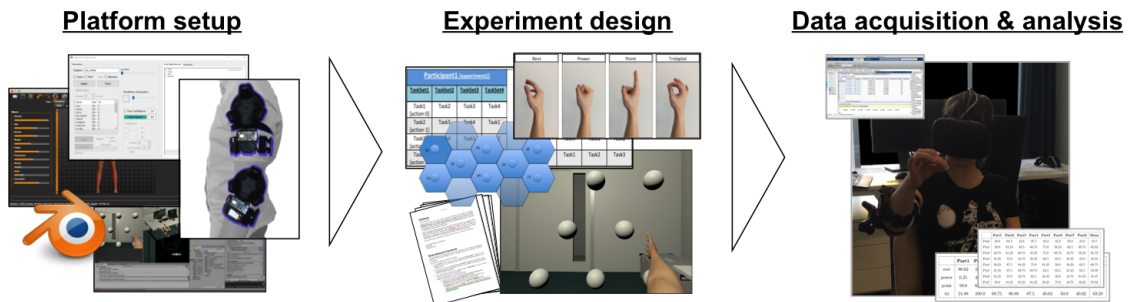


Figure 1: The three main steps of this work: platform setup, experiment design, data acquisition and analysis

From here onwards, the work can be split in three main parts, depicted in figure 1:

Part 1: Platform design

In this section, material and methods will be presented to create a new VR test-platform. *Chapter 4: Material and Methods* will cover most of this step.

Part 2: Experiment design

This part is all about creating an experiment in VR. Most, but not all will be covered in *Chapter 4*, with some more additional facts mentioned in *Chapter 5: Results*.

Part 3: Data acquisition and analysis

The last Part of the thesis will be about acquiring data, which will be presented in *Chapter 4*, *Chapter 5* and the final analysis in *Chapter 6: Discussion*, being finalised with *Chapter 7: Conclusion*.

“Medical science has made such tremendous progress that there is hardly a healthy human left.”

Aldous Huxley

1

Introduction

With an estimated 1.6 million people living with an amputation in the US alone [1] and many of them suffering of the neuropathic phantom limb pain (PLP) at least once in their lifetime [2], quick and successful therapy is one of the most valuable things needed.

So far the classic mirror therapy is the major approach that has already proved its ability to alleviate the phantom pain to regain the former quality of life[3]. Given its relatively simple setup it is very important to start early on with rehabilitation and pursue it continuously. A possible way to simplify this task can be seen in research towards implementing the classical mirror box in different media, making it available in an easy way.

This thesis *-Arm Intent Detection with bio-signals in Virtual Reality-* targets the specific task of creating a mirror therapy platform in VR. Based on the Virtual-Therapy-Arm-system (VITA) created by the adaptive biointerfaces group at the German Aerospace Center, DLR (german: Deutsches Zentrum für Luft- und Raumfahrt e.V.), this work surveys data collection, interpretation and visualisation of surface Electromyography data in VR with virtual mirror therapy in mind.

Usage of VR as a media not only gives advantages concerning space requirements in the real world, it also allows for great flexibility in experiment design and possible applications. Likewise present in the suggested setup, rapid prototyping of experiments becomes a valuable addition to the otherwise slow experiment design, making a VR setup significantly more adaptable.

Since broadly available commercial hardware was used, the presented system is affordable

enough to be used in rehabilitation centers, clinics or even at home.

Software-wise, the created Unity3D based platform incorporates the in-house developed *Interactive Myocontrol*, enabling use of many input devices previously unavailable in VR. As the software not only allows for data acquisition, but also offers a variety of algorithms to interpret this data, it takes care of the intent detection task.

Theoretically compatible with input devices as the *tactile bracelet* presented in [4] and many other devices, the MYO armband by Thalmic Labs was the used device of choice. Given a way of low-cost, occlusion invariant arm constellation tracking, experiment design and conduction was the last step of the thesis.

Looking upon four actions, "rest", "power", "point" and "tridigital", 8 participants had to probe eight spatial separated positions in a virtual world.

Reimplementing the VITA system's features and adding to them largely, the here presented system not only includes the full feature-set of *Interactive Myocontrol* in a virtual world for the first time, but also manages to build a low-cost markerless arm tracking setup, creating a new iteration of the system. Therefore the here presented work can be referred to as VITA 2.0, a modular, future-proof, low-cost VR-platform.

1.1 Goals and challenges of VITA 2.0

With the given project in mind, a definition of goals for this body of work has been established. The summary thereof is based on the requirements of the job posting for this master's thesis as well as several additions by myself and reads as follows:

- Literature survey
- Getting familiar with the existing hardware and software (Unity, HTC Vive setup, bio-signals input devices).
- Importing the drivers for the different input devices available to DLR to the VITA Unity3D software.
- Expand the current VITA system, so that the software includes a full arm up to the shoulder.
- Control and test the arm in VR.
- Design of experiment including online analysis on intact subjects.

As with every project, some of the goals were harder to reach than others. The biggest challenges by far regarding those goals were of two kinds:

- *Importation of the drivers for the different input devices into the VITA system.*
The inclusion of the drivers in Unity3D turned out to be impossible due to software compatibility problems. Because of time constraints an actually far superior, more modular, network based system was implemented.
- *Completion and control of a complete arm up to the shoulder.*
Whilst the completion and control of a virtual arm as presented in chapter 4.3 was rather easy, it actually proved far more difficult achieving the arm's pose by controlling it via real world input. As illustrated in the result-section, chapter 5, and furthermore thoroughly reviewed in the discussion (chapter 6), the creation of a custom tracking system failed, whilst the presented fallback solution was a great success.

1.2 A brief look at the following chapters

With an overview of the reports structure already hinted in previously presented goals, I decided to create the recurring figure below illustrating "the three parts" to help structuring the work.

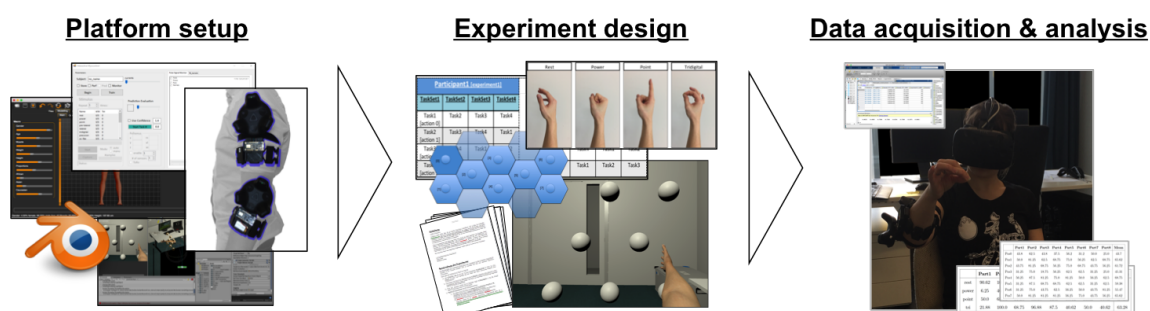


Figure 2: The three parts of this thesis: platform setup, experiment design and Data acquisition and analysis.

Without taking the follow-up chapters on basics (refer to chapter 2) or the none the less important literature survey (see chapter 3), the work is split in three major parts:

Part 1: Platform setup

This part describes the procedure of creating the VITA 2.0 test-platform. Consisting of a description of used hard- and software followed by a closer look at changes made to Interactive Myocontrol, a VR arm setup and finally the creation of a custom tracker included in the also created tracking testbed.

Part 2: Experiment design

Part two on the other hand depicts the creation of the presented experiments.

Part 3: Data acquisition and analysis

Part 3, tightly connected to part 2, handles data acquisition, final analysis and a review of the results. Apart from a brief look at the VITA 2.0 platform (see chapter 4.6), the thesis follows the standard structure of scientific texts, presenting results followed by a discussion, with a conclusion given at the end.

“True knowledge exists in knowing that you know nothing.”

Sokrates

2

Basics

This chapter presents general background knowledge useful to fully understand the later sections of this thesis. Amputation and the resulting phantom limb pain, a possible therapy for the latter, the mirror therapy, as well as Electromyography are discussed here. With the chapter ending with a brief look at VR technologies of past and present.

2.1 Amputation

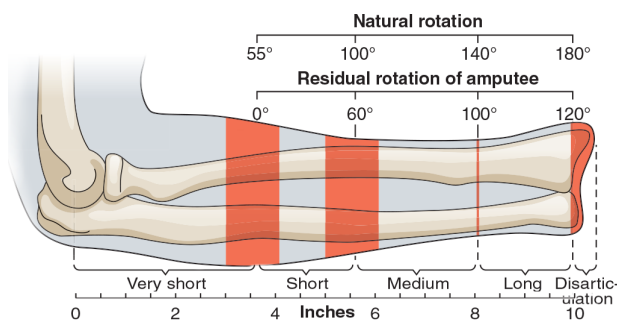


Figure 3: Trans-radial amputation and its effect on rotation ability of the lower arm and wrist.¹

Defined as “the action of surgically cutting off a limb.” in the *English Oxford Living Dictionaries* [6], amputation is a medical intervention performed on over 1.6 mio individuals alone in the USA [state 2005 [1]], with approximately 185,000 amputations occurring every year [7] with a rising trend. The procedure is the measure of last resort in cases of irreparable bone fracturing,

complications of the vascular system or traumata often resulting in necrosis, as it is mostly decreasing quality of life drastically. To minimise the impact of such a surgery, prosthetic limbs are used to restore some of the lost functionalities. Three types of prosthesis exist: passive prosthetics worn for cosmetic reasons, body-powered prosthetics actuated by chest or shoulder movement and lastly smart, robotic-prosthesis [8].

¹based on [5], http://fadavispt.mhmedical.com/data/books/1901/royrehab4_ch15_uf022.png

There exists another problem concerning the amputation that not only impacts the ability to wear prosthesis but also affects everyday life: phantom limb pain (PLP). First described by Ambroise Pare, in 1552, the term “phantom limb pain” was coined in 1872 by Silas Weir Mitchell [9] and is classified as neuropathic pain sensed in the amputated limb. 42.2 to 78.8% of all amputees [2] experience some sort of PLP (other sources even stating that 90 to 98% “experience a vivid phantom” [3]) which manifests itself as a sensation ranging from sense of amputated limb presence, warmth, cold and itching over to throbbing, burning, cramping and even to the feeling of being stabbed [3][2].

PLP almost always begins immediately after amputation, but in some cases can be delayed by days or weeks[10].

2.1.1 Mirror therapy

Today, PLP is treated with a multi-modal approach, combining pharmacotherapy, electrical nerve stimulation and biofeedback as mirror therapy[9], the focus of this work. The mirror therapy is based on creating a reflective illusion of an affected limb to trick the participant’s brain into thinking that the amputated limb is regained with full functionality. Likely based on the visual feedback component of the therapies, decreased PLP intensity and duration have been observed [11] [12].

2.2 Surface Electromyography - sEMG

Since this work is based on non-invasive surface Electromyography, there is need to present the underlying principle and some advantages and disadvantages of the technology. Electromyography (EMG) is a technique that studies the electrical activity of skeletal muscles. In case of the non-invasive surface Electromyography (sEMG), this is achieved by placing two or more electrodes on the skin above the targeted muscle, to measure the potential difference (voltage difference). Measuring electrical or neurological activation of muscle cells, this can tell a lot about medical abnormalities, or more interesting in case of this work, the muscles activation level.

Advantages of the technology are:

- *Easy data acquisition:* Recording data is rather simple as passive electrodes are placeable at any desired locations [13].
- *Large amount of encoded information* EMG data not only includes information on pure muscle actuation but one can also acquire information about the applied force, overall muscle fatigue or possible ailments [14] and even limb impedance [15].

- *Easy intention prediction:* As EMG activity and motion are correlated, prediction is relatively simple. On top of that, muscle activity actually precedes limb motion due to the so-called Electro Mechanical Delay explained by musculoskeletal dynamics [15], making intention recognition even easier.

Disadvantages on the other hand are:

- *Crosstalk:* One of the biggest problems with sEMG is cross talking of adjacent muscles. Having 27 DOFs in the hand alone[16], the region of interest is densely packed including overlapping muscles groups, what makes reading the data of a desired muscle accordingly difficult.
- *User- and action-specificity:* With surface electrodes being restricted to superficial muscles, they tend to be influenced by the depth of the subcutaneous tissue, what itself is user specific and can therefore vary. Apart from the users physical condition, many other parameters affect, as for example varying skin conductance, the muscle activation recordings.

A further point limiting their utilisation is the deformation of the lower arm on actuation, that distorts EMG readings due to minimal changes of the electrodes position. While this can also be used for *optical myography* as presented in [17] and [18], it remains a problem for the more classical Electromyography.

2.3 Virtual reality

Virtual reality or VR in short is a technology creating immersive, interactive digital scenarios with use of real time computer rendering and physics simulation. In theory not only limited to head mounted displays (HMD), most of the current systems make use of a display placed directly in front of the users face.

A brief look at the history of this technology

While the term was coined by Author Damien Broderick [19] in the early 1980s, an even longer history precedes VR, making it much older than commonly considered. Concepts date back to as early as the 1930s [20], followed by first implementations back in the 1960s. Ivann Sutherland created "A head-mounted three dimensional display" [21] based on military technology, even including a motion control implementation. For a long time systems like the aforementioned one were used in military or industrial applications only, as their development, control and maintenance was costly.

Everything changed in the 1990s with VR coming to the then popular arcades and later

even entering the consumer market with Nintendo's mediocre home VR system, the Virtual Boy [22]. Being rather expensive, neglecting head-tracking entirely and only using a red monochrome display, the device was no success and led to a rapid decline of the technology.

Starting with the Oculus Rift-Kickstarter campaign in 2012² a new age of VR began, supporting a much wider range of applications than ever before. The resulting surge of interest in the technology made companies all over the world enter the market, including Google with their low-cost Google Cardboard[23], Sony, Microsoft, HTC and Valve Corporation.

Principle of the state of the art technology

As already mentioned, most systems used today are based on head mounted displays, which seemingly consist of not much more than an OLED-screen hanging in front of the users eyes. In fact, the human eye is incapable of focusing on such a close point [24], hence lenses are used to create a projected virtual image that can be focused.

In more expensive googles, so-called Fresnel lenses -lenses made of several concentric grooves etched into a substrate such as plastic- are used [25]. While capable of focusing light similar to a conventional lens, a fresnel lens can be significantly lighter and more cost-effective, lowering production cost and overall weight of HMDs.

Another feature all modern systems have in common is stereoscopy mentioned as early as 1838 [26]. Side by side stereo visualisation is the process of creating one image per eye, varying the two camera locations slightly (usually about as far apart as human eyes actually are). The human brain combines both perceived images and hence creates a (fake) conception of depth.

Software-wise multi-camera rendering is a demanding task, requiring large amount of resources. One approach to reduce the necessary processing time is the creation of a single virtual camera centering the two eyes. This allows to do the full shading calculations only once, followed by a much quicker recompilation for the two separate camera positions of interest. This can speed up rendering-times significantly compared to rendering each eye in a separate pass. *Advanced vr rendering* by A. Vlachos [27] presents even more advanced rendering concepts that would for sure go beyond the scope of this thesis.

The equally important task of head-tracking is directly connected to stereo rendering. While the position of the virtual camera -and therefore the rendering path- is traditionally given by a human machine interface (such as a controller or a mouse), newer VR setups deploy IMUs or other means of tracking to determine the actual viewing direction of a

²<https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game>

user in the real world, to later be used to mirror orientation changes in the virtual world by moving the virtual camera accordingly.

Despite being great value, this immersive setups brings problems associated with classical motion sickness [28] -or in this case more precisely simulator sickness [29]- which is a condition were a discrepancy between visually perceived movement and the vestibular systems sense of movement happens, often leading to severe nausea.

Trackers and rendering with higher data rates, that are still subject of research, already minimise the problem of simulation sickness, it is still not yet fully solved. However a promising approach for non-stationary cases was introduced by Bozgeyikli et al. in "Point & Teleport" [30].

“Few things are harder to put up with
than the annoyance of a good example.”

Mark Twain

3

Literature Survey

Although many papers were already mentioned throughout the introduction and in chapter 2, I am going to present a handful of very important and useful papers found during literature research.

Firstly, ways to acquire needed data via myography are reviewed. Secondly, a look at VR setups of different sorts is presented, followed by several papers regarding the subject *mirror therapy*. Lastly, a report on desired motion tracking hardware ends the chapter.

EMG: Referring to 2.2 for a broader paper reference regarding amputation, phantom limb pain and EMG, I would like to accentuate Artemiadis’s [15] review. Even though released in 2012 and missing newer hardware, it still is a good starting point.

More recent papers are the *Evaluating and Exploring the MYO ARMBAND* by Rawat et al.[31] and [32], presenting a case study based on the same hardware.

Even though slightly older, Castellini et al. [4] presents a low-cost setup based on force-sensing resistors also used in the 2017 paper by Nissler et al. [33] and in this [34] interesting article by Jaquier et al.. The latter even combined sensor modalities to improve activity detection. As done in this work, ridge regression with random Fourier features is applied for intention prediction.

Another very important paper concerning EMG is [35] by Jiang et al.. Investigating the impact of different arm constellations on EMG-data the paper is one of few comparing amputees and healthy participants. The outcome of EMG-data being less affected by arm

constellation changes for amputees than for healthy participants is even more interesting, as this is valuable knowledge for a setup like the here presented.

Optical myography presented in [18] and [17] is another recent worth mentioning addition to the field.

Virtual reality-setups: With standard papers regarding hard- and basic software implementations presented in section 2.3 and chapter 4.1.1 referring to some more HTC Vive specific topics, numerous non-medical applications exist.

Hilfert et al. [36] for example present an immersive system for engineering and construction tasks running in VR. While not directly connected to this work, it successfully presents a non-game related virtual reality environment. Diverging not only in the reached goal, the system by Hilfert et al. is greatly relying on the LeapMotion as an input device. The infrared-based-hand-tracking-system not just incorporates the standard problems of optical tracking (self-/occlusion and noise [37]), but on top of that only works for healthy participants. "You cannot track something that is not there" applies, making it unsuitable for use with amputees. The same applies for the anyways interesting work [38] by Ling et al. or the more recent [39] by Zhang et al..

Less referred to than originally intended, was *Visual design methods for virtual reality*[40], giving principle concepts for interaction-design in virtual worlds.

Mirror-therapy: With their release, Ramachandran et al. [3] published a paper with a huge impact on the research community, defining a then new therapy, the mirror-therapy. With all sorts of releases as *Butcher's Tongue Illusion* [41], the 2016 release [42] by IJsselsteijn et al. is the most interesting for this thesis. Building upon another meaningful publication of 2015 [43], IJsselsteijn et al. compare the so-called *rubber hand illusion* in mixed- and virtual-reality-setups to the real world. Findings suggest that the intermodal perceptual illusion works in any of the presented cases, stating that mixed- or virtual-reality setups are of particular interest as they are easily adaptable. That being said, the same adaptability holds for the here presented system, with illusion effectiveness to be proved in future experiments.

Nielsen also presented a comparison of real world and VR mirror therapy with use of optical motion tracking to determine the affected limb position in [44]. Being a really interesting work, the main outcome seems to be that the presented experiments were badly designed. Cole et al. presented a less recent study in 2009 [12], where they investigate effectiveness of VR regarding PLP.

Rodriguez et al. on the other hand presented a more classical implementation of the mirror therapy in VR. In [45], the contra-lateral limb is tracked by a combination of LeapMotion and motion sensors, to control a virtual phantom limb. Described as a low-cost VR-setup based on the Oculus Rift DK2 it bears -apart from only mirroring the contra-lateral arm- a close resemblance to the work presented here.

For more PLP related VR, [46] by Dunn et al. is a fabulous reference.

Hardware: [47] and the consecutive work by Li et al. [48] were great sources of inspiration that already discussed similar setups, and how they were implemented. With use of two MYO armbands, a 7 DOF robot arm was telemanipulated (in real life and simulation). While not directly connected to this thesis on first sight, controlling a virtual arm in Unity3D is not much different from the classical telemanipulation approach presented in the above mentioned publications.

Apart from using the MYO armband's orientation output many system for arm tracking exist, among others, mechanical ([49]), optical ([50]), acoustic, magnetic ([51], [52]) or AHRS ([53]) based motion tracking approaches. The ones particularly standing out were AHRS based systems with their rather low-cost-profile, small footprint and broad availability.

In hindsight, [54] by Bertolotti et al. is a great source, as it gives an overview of wearable IMU based tracking systems. Xi Chen's doctoral dissertations [55] is another great source.

[56] and S.Madgwick's[57], are two defining papers, describing algorithm for sensor-fusion of low-cost AHRS systems.

In [58] Zhu et al. present a hybrid setup that combines AHRS and mechanical measurement of elbow displacement via a flex sensor for arm tracking in VR applications.

“Thought by itself however moves nothing, but only thought directed to an end, and dealing with action.”

Aristoteles

4

Material and Methods

In this chapter, the utilised hard- and software as well as the applied experiment design are presented. Whilst the first part of this chapter mostly concentrates on creating an immersive virtual world, the second part is all about experiment design.

With this structure in mind, I start out with a closer look at the off-the-shelf hardware, followed by a description of how a combination of tools was used to build the final VR arm. Lastly, I will look at how the design of the presented experiments came to be and what ensured statistical comparability.

Looking at the three main parts of this work (see fig. 4), I am starting out with *part 1*, the *platform setup*.

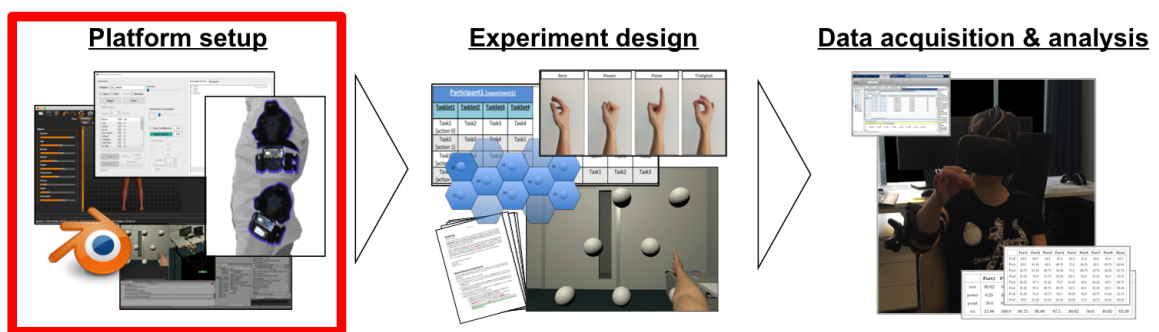


Figure 4: Representing the first big part of this thesis, the platform setup, chapter 4 depicts used materials and tools for its implementation.

4.1 Off-the-shelf hardware

A first step toward VITA 2.0 was the selection of the right hardware. While most of it was already available, this chapter reviews and reasons the choice of the used setup.

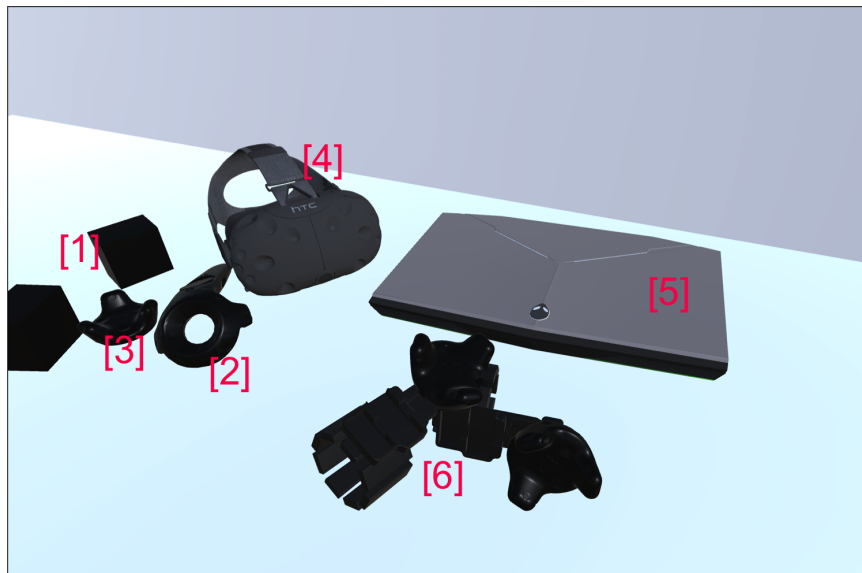


Figure 5: The hardware used in this setup:

1: Vive Lighthouses ; 2: Vive controller ; 3: Vive Tracker; 4: HTC Vive Headset; 5: Dell Alienware 15 R3; 6: The created hardware combining one HTC Vive Tracker, a MYO armband as well as the created custom tracker.

4.1.1 The HTC VIVE Virtual Reality System¹

The used *HTC VIVE Virtual Reality System*, consists of the *HTC Vive Headset* with the necessary *link box*, *two base stations* -often referred to as "lighthouses" and two *wireless controllers*. Developed by HTC and Valve Corporation and being one of the first low-cost VR consumer products entering the market on April 5th 2016, the system is not only able to create captivating experiences due to the 2160x1200 (1080x1200 per eye) resolution HMD, but also offers the critically acclaimed Room Scale VR.

Based on optical tracking, further explained in [59], the 6D-pose of headset, controllers and trackers can be estimated in reference to the lighthouses. With the system being precalibrated, the position and orientation of all tracked devices can be determined in a real world coordinate frame. While this enables natural interaction with the virtual space in regards to spatial movement, the classical input and interaction is realised with use of the input buttons of the HTC VIVE wireless controllers.

¹<https://www.vive.com/eu/>, [14.02.2018]

A relatively new addition to the HTC Vive ecosystem is the *HTC Vive tracker*. So far only available for developers, the trackers are often sold-out and cost about 100 \$ to 150 \$ [60]. Much smaller than the actual wireless controllers, they similarly offer a stable way for 6D tracking. Missing most input methods the original controllers offer -only including the menu-button-, the systems holds the possibility to transfer data to/from a PC by Pogo pin or USB interface [61]. With it being specifically designed for tracking real world objects in combination with its small footprint, it was used as a ground truth for our occlusion invariant arm-tracking-setup presented in chapter 4.4.

Even with shortcoming of image resolution compared to its new competitors and tracking quality [50], the system with a price of 600 \$ in its standard constellation is one of the best consumer products on the market right now and was therefore the system of choice for VITA 2.0.

With the announcement of HTC VIVE PRO in January 2018 [62] and the arrival of the new Windows Mixed Reality Headsets end of 2017, a price drop of the used hardware becomes more likely, making it even more affordable for home users.

4.1.2 The Alienware laptop

With the HTC Vive setup and VR-rendering being a performance-hungry task, the computer platform used was a powerful *Dell Alienware 15 R3²* laptop running Windows 10 Professional. It combines a 15,6" Full-HD IPS-Display, a quad core Intel Core i7-7700HQ main processors, 16GB RAM and a GeForce GTX1060 graphics card, based on the NVIDIA Pascal technology with 6GB GDDR5 memory.

By far the most expensive piece of hardware used in this thesis, with a price of around 2,500 \$³, the computer easily runs all of the used software and still holds enough reserve for future developments.

4.1.3 The MYO armband⁴

While the Interactive Myocontrol software (see section 4.2.3) can handle a variety of input devices, the *MYO Armband* by Thalmic labs was the one exemplarily used in this thesis.

The human machine interface device, with a price tag of about 200 \$⁵, was first available

²<http://www.dell.com/en-us/shop/cty/pdp/spd/alienware-15-laptop?~ck=mn>, [14.02.2018]

³<https://www.cnet.com/products/alienware-15-2016/review/>, [14.02.2018]

⁴<https://www.myo.com>, [22.02.2018]

⁵<https://store.myo.com>, [14.02.2018]

in a developer edition in July 2013⁶. Combining not only eight sEMG sensors, on-board data processing via a MK22FN1M0 Freescale Kinetis Cortex M4 120MHz pro main processor and a NRF51822 Bluetooth-Low-Energy-module for communication, but also an Invensense MPU-9150 9-DOF motion sensor (3D gyroscope, 3D accelerometer and 3D magnetometer). While preprocessed EMG data is being streamed at 200Hz as a unit-less uint_8 (NOT encoded in volts or millivolts), the AHRS data gets sent with a frequency of 50Hz [63].

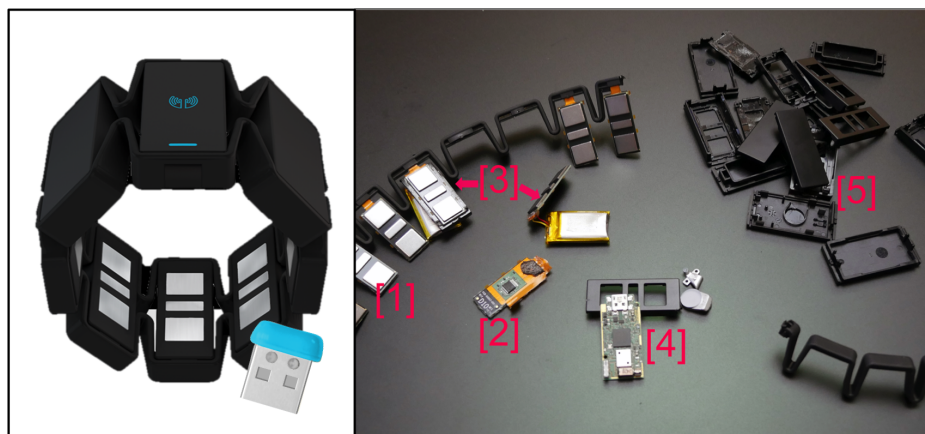


Figure 6: *On the left:* The MYO armband by Thalmic Labs and its bluetooth dongle. *On the right:* Teardown of the device with 1: sEMG-sensor with operational amplifier board attached to it; 2: detached operational amplifier board; 3: two LiPo-batteries; 4: main board with a MK22FN1M0 main processor, a NRF51822 BLE-module and a Invensense MPU-9150 9-DOF motion sensor; 5: plastic casing⁷

Due to its two LiPo-batteries (contained in the bigger compartments on either side of the one with the status led in it) the device becomes one of the few affordable systems for Electromyography in the consumer market. As a consequence, it is broadly used in research all over the world e.g. in [31], [64] and [32].

Other than the official SDK -including Unity3D-support- provided by Thalmic Labs, a variety of libraries and wrappers exist, with a full list to be found at [65].

Not only the small form-factor and the number of features, but the big community it built over time, make the device also the first choice for the here presented work.

While MyoSharp⁸-a C# implementation for interacting with the device- was the only community-developed library used, it handles communication between the MYO armband

⁶<https://beta.techcrunch.com/2013/07/03/myo-armband-developer-program/>, [25.02.2018]

⁷Based on:<https://static.thalmic.com/sapphire/techspecs/included3.png> <https://learn.adafruit.com/assets/30336>

⁸<https://github.com/tayfuzun/MyoSharp>, [5.03.2018]

and the later presented Interactive Myocontrol software.

4.2 Software and Toolkits

Proceeding the introduction of the systems hardware-side, this section presents the software components and plugins employed by the VITA 2.0 platform.

Starting with a description of Unity3D as the basic engine, followed by a look at Interactive Myocontrol, the data acquisition- and intention-prediction-software, with other software and plugins presented at the end, the chapter will give details on implementation and combination in between.

4.2.1 Unity3D

Unity3D, a game-engine developed by Unity Technologies⁹ was first released in June 2005 and is used in the stable version Unity3D 2017.2.

The cross-platform engine is utilised by more than 1.3 million developers [66] and supports over 25 target platforms, including not only all major computer operating systems, but also makes development for mobile platforms as iOS and Android possible. In all cases Unity3D supports 2D and 3D graphics with according standards as sprite-importation, bump-, occlusion- and normal-mapping as well as a series of post-processing filters.

Next to easy importation and animation of 3D objects using the Collada .dae, or the more broadly supported .blend -used by the renown Blender CAD software (see 4.2.2)-file format, shader programming is also possible.

More important than any of its so far mentioned features is Unity3D's huge library of free and paid plugins with most of them easily accessible via the official Asset Store¹⁰. With them either adding small features, 3D model-sets, whole new platforms for physic-simulations¹¹ or solutions for other tasks, plugins add specific features to Unity3D that can make a big impact on projects.

⁹<https://unity3d.com>, [10.03.2018]

¹⁰<https://assetstore.unity.com>, [10.02.2018]

¹¹<https://assetstore.unity.com/packages/tools/physics/bullet-physics-for-unity-62991>, [10.02.2018]

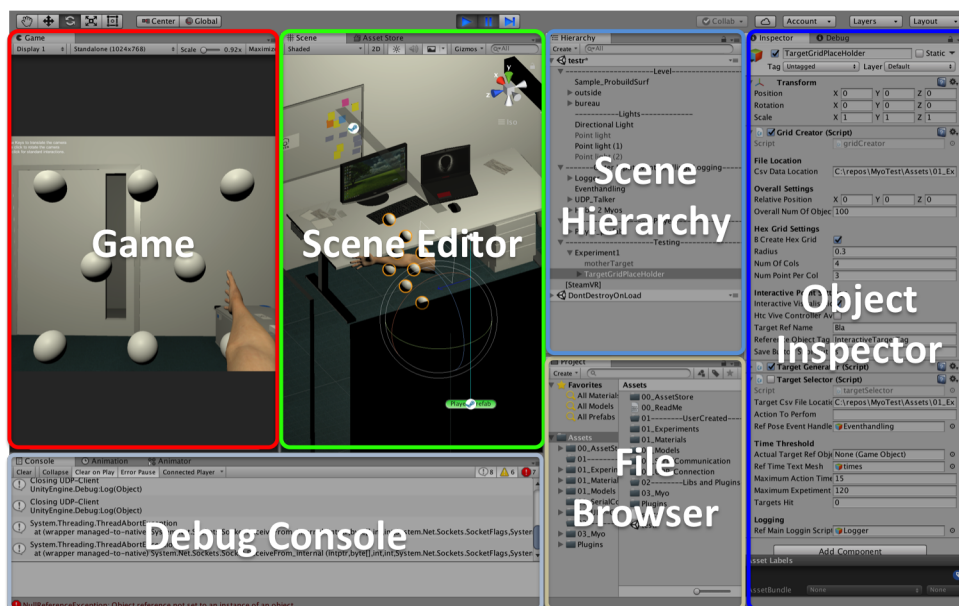


Figure 7: Unity3D in its standard layout. From the left to the right: The game view with a live preview of the game running; The scene editor view, where game objects can be placed, resized and rotated; The scene hierarchy, showing all game objects and how they are (inter-)connected; The object inspector showing the properties of a selected game object, including their attached scripts; The debug console for displaying debug and error messages; Lastly the file browser for managing assets.

Above all this, Unity3D's mechanics can easily be expanded by user-designed programs, so-called scripts, written either in the Unity Script language or in (and mostly) C#. For cross-platform compatibility MonoDevelop, included since Unity3D 3.4, takes care of code compilation and can also be used as code-editor. Working with Mono, the MonoDevelop 5.9.6 software included in the used Unity3D 2017.2 works with an equivalent of Microsoft .Net 3.5 or as desired in this project with an experimental setup comparable to Microsoft .Net 4.6.

Apart from this, Unity3D offers inclusion of C++ code via DLLs, making hardware specific or performance critical development much easier.

4.2.2 Other tools

Next to the two big platforms; Interactive Myocontrol and Unity3D (refer 4.2.3 & 4.2.1), several smaller tools and toolsets were used. The most important ones were:



Steam® VR

The **SteamVR Plugin**¹² for Unity3D is the most important plugin used. It was developed and released by Valve Corporation, one of the two companies behind the HTC Vive (see 4.1.1). The API provides tools for game creation on all major VR setups available. It includes specialised and user-friendly tools for integration of tracked devices, stereo rendering and room scale VR. SteamVR's compositor also allows live previewing software on computer and headset of choice at the same time, being a valuable tool on supervising later presented experiments during runtime. On top of the aforementioned features, a locomotion technique described as "Point & Teleport" [30] can easily be included as the most natural way of moving about in a level.

Further reading and the great "HTC Vive Tutorial for Unity" by Eric Van de Kerckhove can be found at [67].



The **Procore Bundle** originally developed by procore3d¹³ is a formerly paid set of plugins for the already presented Unity3D environment. It offers a variety of tools to create, texture-map and refine 3D objects and environments directly in the Unity3D editor. In case of this work, it was broadly used to create the immersive world in which the later experiments took place. As of mid February 2018¹⁴, proBuilder is part of the Unity3D software and can freely be downloaded, being included from Unity3D 2018.1 beta onwards.



Blender¹⁵ is a free and open-source 3D computer aided design software developed by the Blender Foundation. The software used in version 2.78 offers a gigantic spectrum of tools that include features as 3D modelling, simulations, rendering, texturing and many more tools as for example its own game engine.

In the case of this work, the prescribed features of Blender were used for composing as well as altering and finally rigging humanoid 3D avatars.

¹²<https://assetstore.unity.com/packages/templates/systems/steamvr-plugin-32647>, [10.03.2018]

¹³<http://www.procore3d.com>, [22.2.2018]

¹⁴<https://blogs.unity3d.com/2018/02/15/probuilder-joins-unity-offering-integrated-in-editor-advan> [27.02.2018]

¹⁵<https://www.blender.org>



With Blender, a powerful tool for creating and altering 3D models at hand, a second software, namely **MakeHuman**¹⁶ was used to speed up the content creation process tremendously

The used software in version 1.1.1 is an open-source tool for 3D human avatar creation.

4.2.3 Interactive Myocontrol

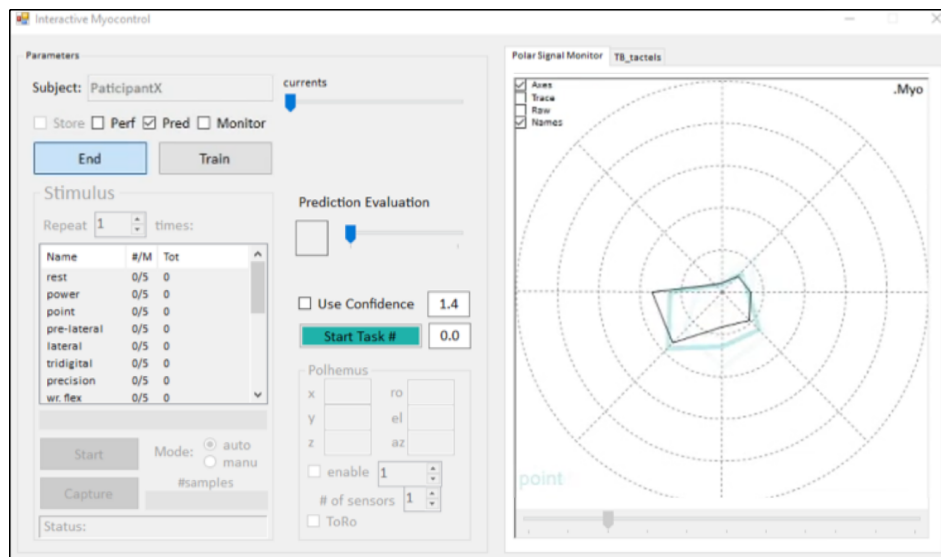


Figure 8: The in-house developed Interactive Myocontrol software acquiring data from a MYO armband.

Interactive Myocontrol, is an in-house software developed by the Bionics Group at DLR Oberpfaffenhofen. Over the years it has constantly been reworked, updated and extended. By now it can be seen as the platform for most experiments taking place in the research group.

Based on C# and the Microsoft .NET Framework in version 4.0, the software only runs on the later releases of the Microsoft operating system as Windows 7 or 10. Given the programs Windows Presentation Foundation based graphical interface, several devices for myography data acquisition as for example OttoBock MyoBock 13E200 sEMG electrodes (www.ottobock.com) or the here utilized MYO armband can be used.

Besides collecting and presenting data in various ways, the software also offers different machine learning algorithms for interpretation of data, including teaching and prediction of defined actions via Ridge Regression with Random Fourier Features, closer described in [68], [4] or [69].

¹⁶<http://www.makehuman.org>

The output of the machine learning part is a 9 value holding string in the following format:

[x.xxx, x.xxx, x.xxx, x.xxx, x.xxx, x.xxx, x.xxx, x.xxx, x.xxx]

Normalised data [-1, 1] is encoded as *thumb rotation* in the first index, *thumb flexion* in the second, *index finger flexion* in the third, *middle finger flexion* in index 4, *ring finger flexion* in index 5, *little finger flexion* encoded in 6, followed by *wrist flexion/extension* and *wrist adduction* (movement of bending the wrist to the thumb, or little finger) in the value index in 7 and 8. The last index encodes the *wrist supination and pronation* (rotation of wrist and lower arm).

While, other than intended initially, porting (parts of) Interactive Myocontrol to Unity3D, the task turned out to be impossible.

Reason for this can be seen in Unity3D's platform independent way to compile code, the open source Mono. As aforementioned, Mono is only a reimplementaion of the Microsoft .NET-platform, that cannot guarantee full code compatibility. As it turned out, serial communication, one of the most important interface for communicating with hardware, is one of the few classes that do not work as intended, rendering most of the hardware drivers incompatible to Unity3D.

Instead, the existing program was expanded by a UDP server, able to broadcast data to any network subscriber available, even a local data sink (as for example the Unity3D based VITA 2.0 running on the same PC). To make things easier, three booleans, encoding Interactive Myocontrol status (recording, training or predicting phase), were added to the 9x1 machine learning string, before broadcast. With this changes done to Interactive Myocontrol, the program was ultimately used to handle training and prediction of the four action of interest.

4.3 A new arm for VR

Since all important tools regarding software have been mentioned -including the basic toolset for arm-intention detection- the process of creating a new virtual arm up to the shoulder can be presented in the following text.

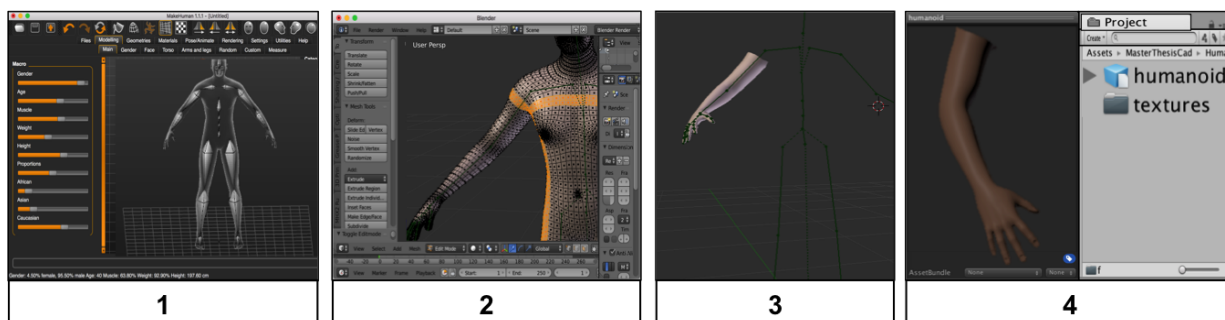


Figure 9: The full transition from MakeHuman over Blender to Unity3D.

From left to right: 1: Creating a rigged human avatar in the MakeHuman software. 2: Deleting all but the arm of the avatar in Blender. 3: Resulting model in blender. 4: Ready to use textured arm in Unity3D.

In the first step of the process, MakeHuman (see 4.2.2) was used to create a humanoid avatar of desired appearance and physiognomy. Additionally the software was used to rig the avatar automatically. Usually rigging -the process of creating a virtual skeleton for a 3D model- can be quite time-consuming, but is fundamental for animating and deforming the object later in-game.

As only the arm of the created avatar was needed, the character data was exported to Blender where unnecessary vertex groups could be pruned and likewise retouching of the rig and its deformable regions was possible.

With importing the now rigged and finalised model into Unity3D, the animation process of the arm started. Mimicking the real world, I generated a simplified arm with nine independent DOFs for hand (one for each finger, two for the thumb) and wrist (3 DOFs) as well as one DOF in the elbow, topped by three DOFs in the shoulder, summing up to a total of 13 DOFs (compared to the real human hand alone having 27 DOFs [16]).

Fortunately animation being a strength of Unity3D, I was first able to create animation clips for each of the 13 independent actuated joints. To be exact, the relaxed and flexed position was recorded. Next Unity's Blend Tree technology [70] was used to compose the desired animated actions by mixing two or more animation clips, resulting in smooth and natural move sets, with the outcome of the blend controllable via a script based external input to the animation controller.

With a 9 DOF of freedom hand, a magnitude of possible hand poses can easily be created. In case of this work the 9 value containing string generated by Interactive Myocontrol is directly fed into the Unity3D-part of the VITA 2.0 setup to pilot the blending of the arm, creating a lifelike hand movement.

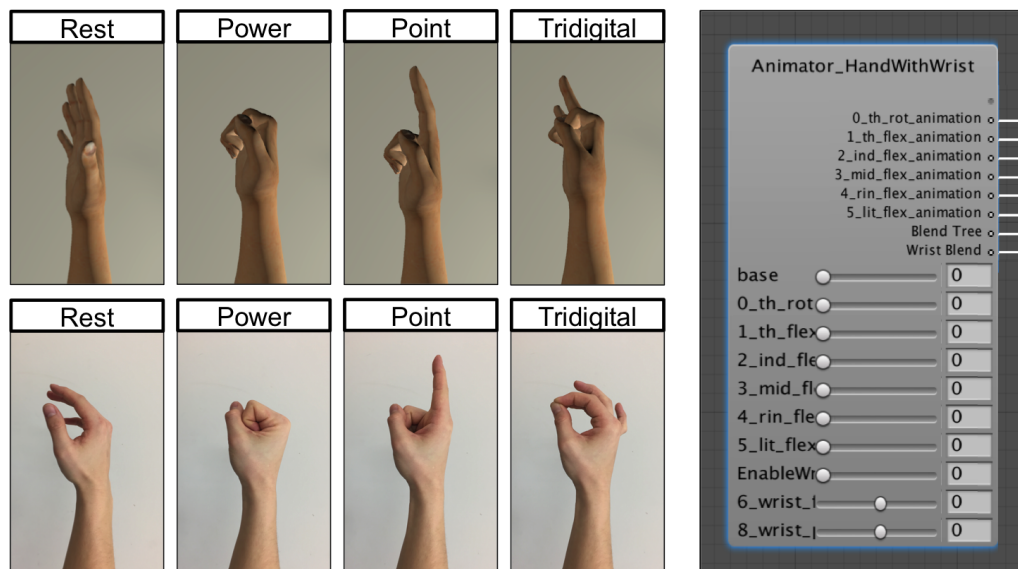


Figure 10: Animation of the virtual hand.

On the left: The four actions of interest in VR and reality.

On the right: Unity Blend Tree for the hand and wrist movement in the Animation Controller of the arm, allowing far more poses than the ones presented on the left.

4.4 Arm positional tracking

With the created VR setup, including a full arm up to the shoulder, the new focus of the thesis became controlling this arm with user signals. Already able to predict different hand- and wrist-poses using Interactive Myocontrol and setting them via animation blending, a way had to be found to animate the other joints of the presented virtual arm, namely the orientation of shoulder and elbow.

So far the VITA software was heavily relying on the 6D optical HTC Vive trackers presented in 4.1.1, what should change in VITA 2.0. While the trackers are already much smaller and more versatile than the original wireless controllers, they are -until now- only available for developers and are additionally hard to purchase.

On top of that, they are still heavy and expensive, taking into account the need for all other components of the HTC Vive Virtual Reality System to actually acquire tracking data. Further interference with natural or artificial light sources can have a significant impact on tracking quality [71], with occlusion being another one.

Stumbling over the video [47] supposedly created by members of *Robust Autonomy and Decisions-department* of the *University of Edinburgh* (Nardelli et al.), the article by Li et al. [48] was found, presenting a possible solution for arm tracking. In both cases, the

easy accessible orientation data of the MYO armbands was used to control a (simulated) robot.

A proof of concept was running quickly using Thalmic Labs' official Unity3D-plugin to acquire the MYO armband's pre-processed orientation estimation based on its Invensense MPU-9150 9-DOF motion sensor. While not as sophisticated or precise as the expensive Xsens MVN system [53], tracking was still impressive and hold up relatively well to the HTC Vive's tracking data (see sec. 5.1 of the results). Nonetheless three problems came to mind:

- Firstly, the MYO armband to acquire orientation data, at least **two armbands are needed** -one on the upper and one on the lower arm- to get an estimation of the wrist position.
- Secondly, using any other of the now available input devices for EMG-data acquisition, **future experiment setups become more bulky**, since on top of setups as presented [33] in or [34], the two MYO armbands would have to be attached to the participants arm.
- Lastly, and most importantly there occurs -as with all AHRS based tracking systems- the problem of **error accumulation**, better known as **drift**: an ever-increasing difference between estimation and actual location.

4.4.1 Creating a custom tracker

With the aforementioned problems in mind, design of a new custom tracker began.

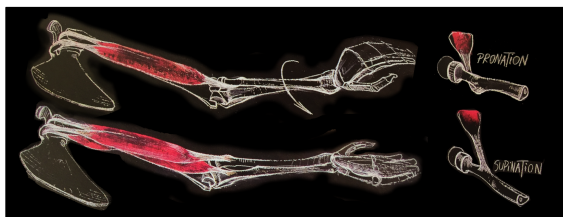


Figure 11: Drawing of supi- and pronation of the hand with muscle and skeletal constellation¹⁷.

While many setups -as [58]- oversimplify the human elbow as a one degree rotational joint, in truth the elbow (*articulatio cubiti*) is composed of three joints.

Articulatio humeroulnaris and *articulatio humeroradialis* -enabling flexion and relaxation of the arm- as well as the *articulatio humeroradialis proximalis* [72], responsible for supination and pronation of the hand. In other words, *articulatio humeroradialis proximalis* and therefore the underarm-elbow-complex takes part in wrist rotation, making measurement of the rotation close to the elbow valuable (refer to the outlook in the conclusion for more details).

¹⁷based on [72]

To measure rotation of the lower arm and to simplify the tracking setup by making it more modular, the same 3 DOF rotational tracker should be used for both joints, elbow and shoulder. With the MYO armband setup working to my satisfaction, an IMU based system was desired.

Figure 12, based on [73], presents a comparison of different low-cost microelectromechanical gyroscopes for use in AHRS. The measure of choice is the zero rate level [in degrees per second], being a measure of error/drift over time, generated when there is no movement of the device. Large zero rate levels will give a system the impression that the device is moving, when it is in fact stationary. With a perfect sensor, there would be no change or a constant zero rate level measured.

	+/- 250 dps	+/- 500 dps	+/- 2000 dps
L3GD20	10 dps	15 dps	75 dps
1 FXAS21002C	0.3906 dps	0.78125 dps	3.125 dps
LSM9DS0	10 dps	15 dps	25 dps
LSM9DS1	??? <= 30 dps	??? <= 30 dps	30 dps
2 MPU-9250 *	5 dps	???	???
BMI055	? 1 dps ?	???	???

* The MPU 9250 is seen as superior to the Invensense MPU-9150 9-dof motion sensor included in the MYO

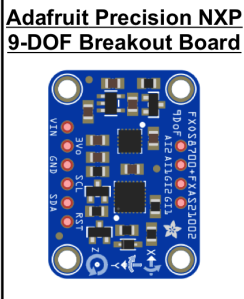


Figure 12: Low-cost gyroscopes and their zero rate level [in degrees per second (dps)] on the left¹⁸ and the used *Adafruit Precision NXP 9-DOF Breakout Board* on the right.

Including the MPU-9250, the predecessor of the MPU-9150-chip integrated in the MYO armband, the by far best system is the *FXAS21002* followed by the *BMI055*, a chip used in the renowned *Bosh Sensortec BNO055*.

Given the presented information, the *Adafruit Precision NXP 9-DOF Breakout Board* with a price of about 15 \$¹⁹ was the system of choice, combining two of the best low-cost motion sensors available: the *FXAS21002* 3-axis gyroscope and the *FXOS8700* 3-Axis accelerometer and magnetometer.

Capable of working with a supply voltage of 3.3V or 5V, the breakout board supports the SPI and I2C interfaces and can acquire magnetic- and acceleration data with a frequency of 1.563 Hz to 800 Hz, encoded via 14-(accelerometer) or 16-bit (magnetometer) analog-digital-conversion and gyroscope data with 12.5 Hz to 800Hz, encoded with 16bit.

Acceleration range is adjustable to $\pm 2g / \pm 4g / \pm 8g$ while the magnetic sensor has range of $\pm 1200\mu T$ with a $\pm 250/500/1000/2000/s$ configurable range for the gyroscope.

¹⁸<https://learn.adafruit.com/comparing-gyroscope-datasheets/overview>, [11.03.2018]

¹⁹<https://www.adafruit.com/product/3463>, [10.03.2018]

The chip was combined with another low-cost development board by Adafruit Industries, the *Adafruit Feather nRF52 Bluefruit LE*²⁰. With Nordic Semiconductor’s *nRF52832* [74] as a basis, the features of Adafruit’s board include a 64MHz M4F main processor, 512KB flash and 64KB SRAM, Bluetooth Low Energy with built in USB and battery charging on a small footprint.

Following Adafruit’s tutorial on usage of the NXP-breakout [75], elementary sensor fusion based on the often referred Mahony et al. [56] and Sebastian Madgwick’s [57] was implemented.

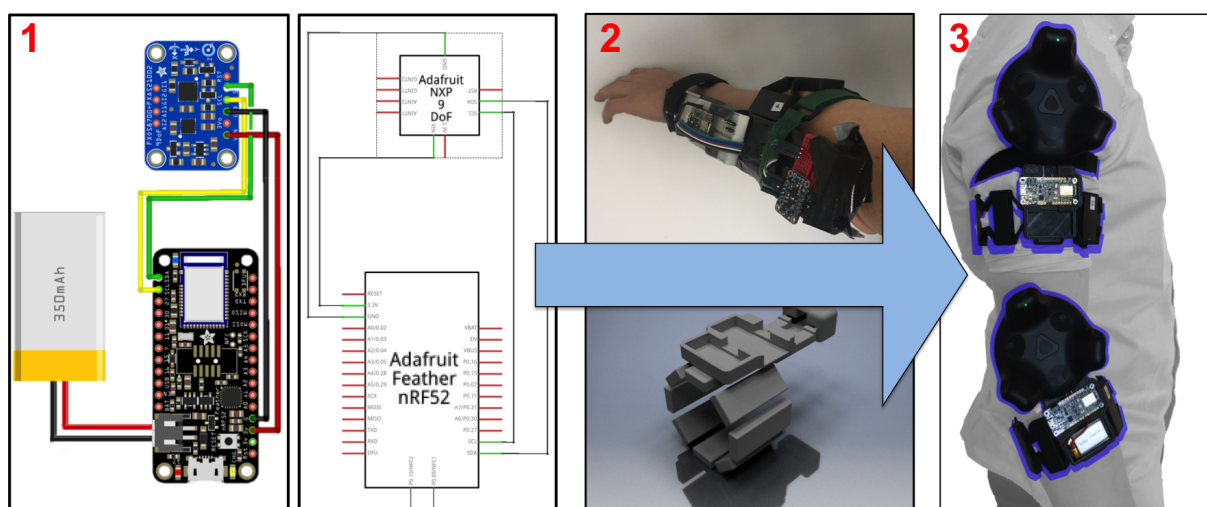


Figure 13: 1: Circuit for the custom created tracker based on an Adafruit Feather nRF52 Bluefruit and the Adafruit Precision NXP 9-DOF Breakout Board²¹.

2: Prototype setup on an arm (top figure), followed by a rendering of the 3D-printed setup (bottom figure).

3: Final setup attached to the user’s arm. Combining the MYO armband, the here created custom tracker and the HTC Vive tracker, a multi modal hardware platform was created, capable of recording orientation data of three independent sensors.

Rapid-Prototyping

After examining the circuit board on a breadboard and making sure that it works, several tests were done with a prototyping setup, directly strapped to the arm. Afterwards and to enable easier data comparison, custom created hardware to connect the custom tracker to the MYO armband and the HTC Vive tracker -the tracking testbed- was designed. Computer-aided manufacturing [76] tools were used to rapidly progress with the task.

²⁰<https://www.adafruit.com/product/3406>, [12.03.2018]

²¹Graphics were created using the open-source fritzing software - <http://fritzing.org/>, [1.03.2018]

First *Autodesk Inventor 2017*²² was used to create the design, followed by the 3D printing slicing software *Simplify3D*²³ and finalised by 3D printing the model with 2.85mm Colorfabb polylactic-acid-filament²⁴ on an *Ultimaker2+* 3D printer²⁵ (this process is depicted in 2 and 3 of fig. 13). Ending up with the smallest, finished looking and feature-rich setup I could come up with, it also became much more user friendly.

Due to creation of Unity3D-compatible code to communicate with each of the three sensors, data recording via quaternions finally became possible. Having comparison of the trackers in mind, the recorded data of each of them had to be processed as follows:

```

1  1.: Get the data in two comparable sets :
2    ground_truth      = QuaternionArray ;
3    dataSet           = QuaternionArray ;
4
5  2.: Select the timestamp based on which you want to compare
6    calibrationIndex = QuaternionArray.findeCalibrationIndex ;
7
8  3.: Get the relative offset from the ground truth at this index
9    Offset = Quaternion.Inverse(ground_truth[calibrationIndex]) * dataSet ;
10
11 4.: "Substract" offset to get the data in reference to ground truth :
12  dataSet_inRef    = dataSet1 * Quaternion.Inverse( Offset ) ;

```

Listing 4.1: Handling offset of quaternions in pseudo-code

Which resembles getting a specific relative position (the OFFSET) of the data set to the ground truth -the HTC Vive tracker- at a desired time-stamp -the calibration index- to apply it to every data point. Analysing the so created data sets, one can directly compare it with the ground truth.

Apart from quality of the measured data, three different control scenes for directing the virtual arm's shoulder and elbow are now available.

As closer dissected in the results, the MYO armband became the sensor of choice, making the arm overall constellation and a predicted hand position available to the system.

²²<https://www.autodesk.com/products/inventor/overview>, [8.03.2018]

²³<https://www.simplify3d.com>, [10.03.2018]

²⁴<https://ultimaker.com/en/products/ultimaker-2-plus>, [11.3.2018]

²⁵<https://colorfabb.com/standard-black>, [11.03.2018]

Given the last sections, platform design and setup is done. Following the three step agenda, this section will be about step two, experiment design.

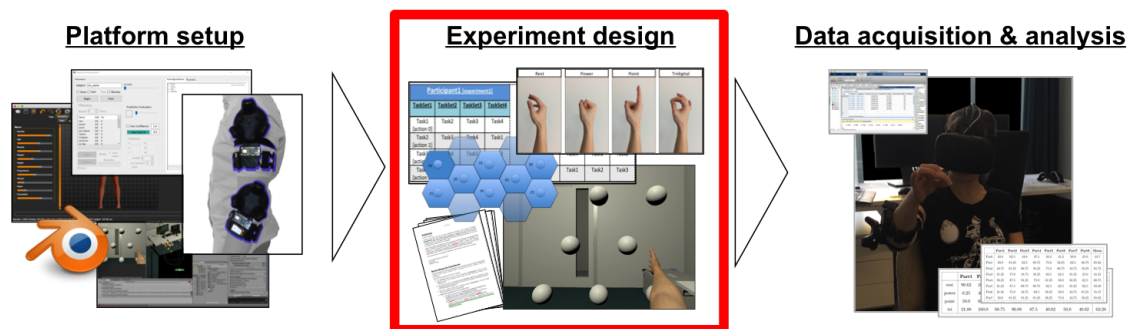


Figure 14: As platform setup was presented in the preceding subsections, part 1 of the thesis is finished.

This chapter marks the beginning of part two of this thesis: the experiment design.

4.5 Experiment design

With use of hard- and software allowing hand gesture recognition and estimation of the toolcenter point of a participant's hand, an experiment was created. Eight participants had to perform the four actions, "rest", "power", "point" and "tridigital" in eight spatial separated positions in a virtual world, using their right arm. Prediction quality of the used machine learning software was then used as a (direct and indirect) measure for varying EMG data quality.

Using the VITA 2.0's newly created "Experiment"-submodule, a target grid with 8 cubical targets was created. As a hex-grid was used as a basis for their location, the points are equidistant to their direct neighbours (this is presented in the leftmost part of figure 15). A virtual margin of 30cm and a target size of 7.5 cm was empirically determined and made easy reaching possible.

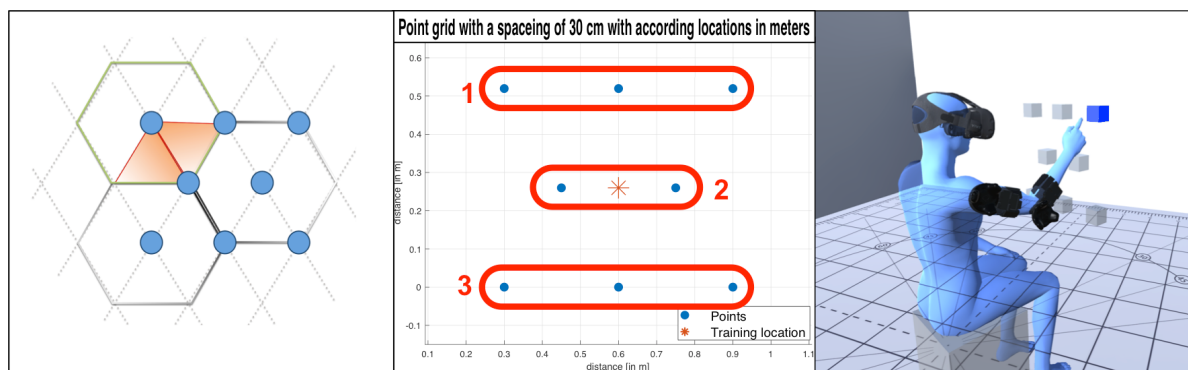


Figure 15: Experiment setup and computer renderings of a participant wearing the arm-tracking device during an experiment. The graph in the middle shows the used point cloud as a 2D plot, including the training position in the middle and the three groups: 1: "upper group", 2: "middle group" and 3: "lower group", consisting of 3, 2 and 3 points respectively.

For better comparison the locations were grouped as follows: The "upper group", consisting of three locations (namely targetIndex 0, 3 and 6), the "middle group" with two locations (index 2 and 5) and a third group, the "lower group" with three locations respectively (index 1, 4 and 7) (see the middle section of fig. 15).

On starting the VITA 2.0 software bundle -followed by a short calibration of the gyros- an accordingly seated participant is greeted by a virtual arm, mimicking the movement of his or her real arm (as only two trackers are used, the participant is encouraged to only move his/her arm). In the next phase, the experimenter chooses which actions to train via the Interactive Myocontrol and starts the training-phase. For the person in VR, this simply means repeating what the virtual hand does while staying in a predefined "training position" in the middle of the point grid (see fig. 15, middle). After three repetitions of the actions, "rest", "power", "point" and "tridigital", the machine-learning algorithm is trained and should now be able to predict the participant actions correctly (in this experiment, no retraining is allowed).

This training phase is followed by a 60-second familiarization phase, the participant becomes accustomed to system usage, before finally starting the actually experiment phase.

An exemplary depiction in the far right part of figure 15 shows the aforementioned experiment phase, the procedure of which includes several so-called *sub-tasks*, each composed of one of the pretrained actions ("rest", "power", "point", "tridigital") and one location, referred to as a target (indexed 0-7). A *Task* in turn is composed of 8 of those *sub-tasks*, all with the same action and varying locations. While changes in the target sequence are possible, each location is called exactly once. With four actions available, a *TaskSet*

consisting of four *Tasks* was put together, containing 32 *sub-tasks* in total.

To accommodate for learning effect as well for wear over time, a permutation of the *Tasks* in a *Taskset* was necessary, resulting in a total of 128 single *sub-tasks* per participant, split in four *TaskSets* à four *Tasks*, with each of them consisting of the same action in 8 different positions(each a single *sub-task*). Described sequence is depicted figure 16.

Naturally split by the actions, pauses of 30 seconds in between each of the *Tasks* were included on conducting the user-tests. Not only to minimise exertion of the participant, these pauses also enabled recalibration of the trackers before starting a new *Task*, minimising the influence of IMU drift. On top of that a non-mandatory pause was offered in between each of the four *TaskSets*, not to be used by any participant.

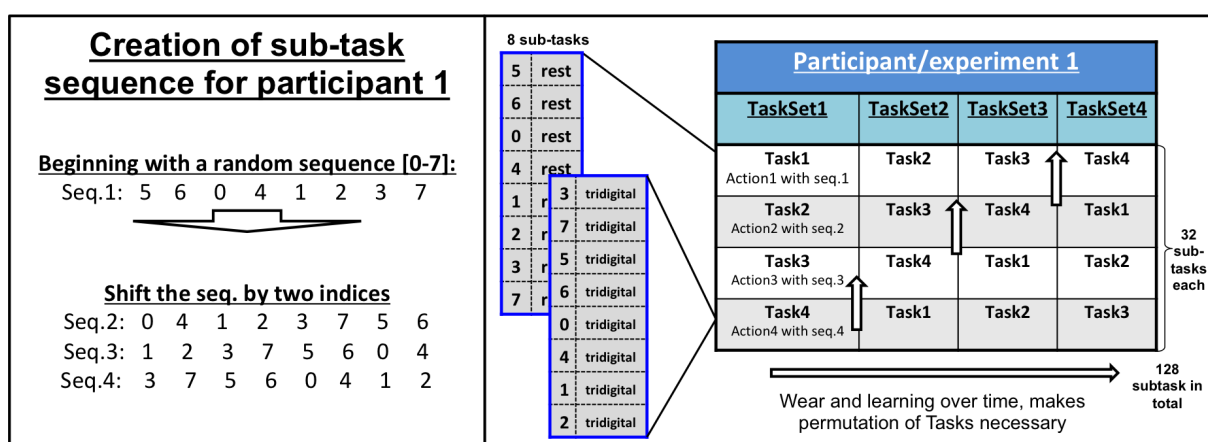


Figure 16: Creation of a participant specific experiment-set.

Starting with a random subtask-sequence -abbreviated seq.-, the follow up sequences are generated by shifting the sequence by two indices.

The right part of the figure not only depicts the composition of *Tasks* and *TaskSets*, but also shows how *Task 1* to *4* are interchanged in between *TaskSets*.

With 8 different positions the number of all possible combinations would equal the factorial of 8 and so 40.320 possible combinations. Multiplied by four -as for the four different actions- an unbearable large number of participants would be needed to create statistically meaningful results. To limit the number of repetitions, permutation was not only used in case of task-sequence-creation for single participants, but moreover to create participant specific experiment-sets for a smaller participant group consisting of 8 individuals. On top of the taken measures, the presented experiment is only looking at testing the right arm.

Based on the first random sequence for participant one, the latin square-method was used to create the first sequences for all other participants. Doing the same with the successive

sequences of the original sequence already seen in fig.16 (shifted by two indices), sequences 2 to 4 for all participants were created accordingly.

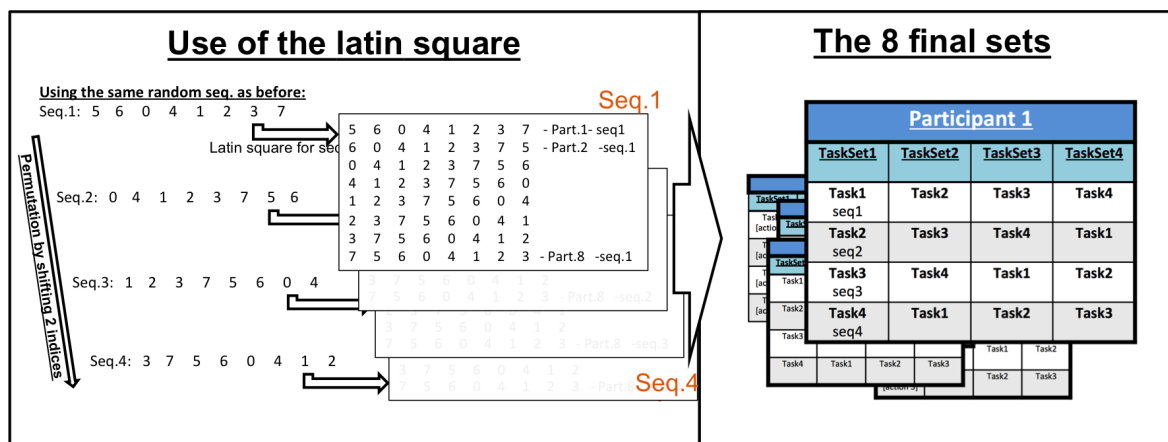


Figure 17: Experiment-set creation for all participants based on the latin squares of a random sequence and its follow ups on the left, with the final, comparable and participant specific experiment-sets on the right.

As shown in figure 17, the final batch of participant-specific, comparable experiment-sets consisted of 8 sets in total, with each of them composed of 128 sub-tasks.

Regarding the measure of success, a task was seen successful if a participant was able to hold a given action in a according location for more than 3 seconds.

Measured by the offset of Interactive Myocontrol's prediction-values and action-values used as stimuli during training, variation of up to 25% were seen as a successful action.

With the inclusion of a maximum time per task (maxTT) of 15 seconds, a time constraint was also included, enabling recording of the Task Completion Time [TCT] -defining how long it took a participant to succeed. Lastly and as the 3 second threshold was reset on losing the action, the Time In Task [TIT] -as a measure for the accumulated Time In Task- became a second important measure.

As a result of the design process, a formal description of the experiment was put together for handing out to the eight participants:

1. Briefing of the participant regarding the experiment, tracking and EMG data acquisition, followed by **attaching the input devices** to the participant's upper and lower arm.
2. **Start of the training phase.** In this phase, the participant mimics four hand

movements -the actions- as presented to him by the virtual hand in a predefined location. This step is repeated three times with four actions being "rest", "power", "point" and "tridigital".

3. After training the system, a 60 second lasting **familiarisation phase** is provided in which the participant becomes accustomed to system usage.

4. **Start of the experiments** and actual data acquisition:

- It is the participants task to reach a specific cube, highlighted with colour and to do one of the pretrained actions at this location.

After successfully doing so for 3 seconds, the colour of the cube will blend to green, followed by the next cube getting highlighted. If he fails to do the action, within 15 seconds, the next target will be selected automatically.

Cube colour and time will reset, if the participant fails to hold the virtual hand in the given pose or if he leaves the location.

- The entire experiment is composed of four so-called "TaskSets". Each of them containing 4 Tasks, a sequence of an action to be done in one of the 8 different positions.

5. On top of the aforementioned instruction, the participant is encouraged to only move his/her arm and head. If possible he should not move the upper body.

4.6 Brief look at the resulting platform



Figure 18: The immersive VR setup on the left compared to its real life counterpart, an office in DLR, Oberpfaffenhofen on the right.

The immersive VR environment developed with the aforementioned toolsets depicts an office closely resembling the place where most of the programming and experimenting took place.

Due to a modified version of Interactive Myocontrol running in the background and hosting a UDP server, (re)training of different actions is possible. Being provided with data, the Unity3D-part of the project takes care of presenting the desired actions.

While this connection is depicted in the left part of figure 19, the right panel visualises the composition of the Unity3D-part of the software. Following an intuitive naming-convention, the software is made up of four main modules:

- **In-/Output:** This module takes care of handling in- and output of different kinds, the "Serial handling" manages connecting to and reading device data that comes from the custom trackers and MYO armbands. The "UDP"-module on the other hand is taking care of UDP-communication between Unity3D and Interactive Myocontrol. Taking the received data as an input, the "Event handling" triggers different events as for example action recognition. Lastly "Logging" solely handles data logging and writing those logs to hard drive.
- **Player:** Only created of two submodules, the first is the "HTC Vive Player Prefab" a slightly changed prefab of the *Steam VR plugin* mentioned in 4.2.2. While it takes care of Room Scale VR, teleportation as well as head- and controller-tracking, the

second module named "The new Arm" is the part of the software controlling the participants virtual arm, including its constellation and actual hand-pose.

- **Experiment:** While the other modules take care of interaction, this module supervises the experiment setup with three powerful and adaptable sub-modules. The "Grid Creator" creates possible target positions -either randomly or with epipolar point distance- that can be altered online. Reading the so created or any other point list in the *comma-separated values*-format, the "Target Creator" takes care of setting up target-objects at those positions. The "Target Selection" module complements the "Experiment"-module by selecting one desired target at a time.
- **Level:** While this module is only a passive structure in the setup, it significantly helps to create a sense of full immersion. As interchangeable as the other modules, a quick scene change is possible. By replacing the module with any other preset scene, experiments in the virtual office are as possible as interactions close to the virtual ISS.

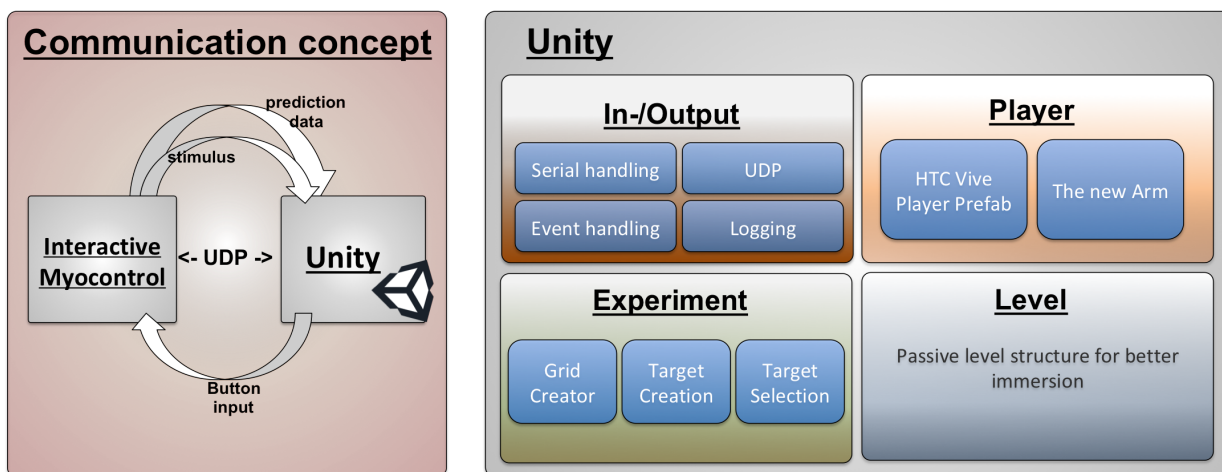


Figure 19: The overall software structure in one picture:

On the left: Inter program communication and software structure of the setup.

On the right: The Unity3D part of the setup and its (sub)modules.

“God does not play dice”

Albert Einstein

5

Results

With the data acquisition and analysis, part 3 of 3 is finally reached. This last part presents the results in this chapter, with an analysis and a conclusion in the next two chapters.

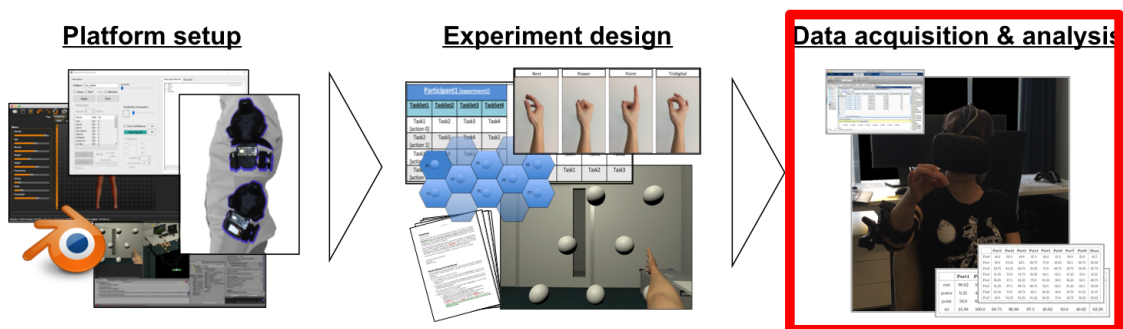


Figure 20: This section marks the beginning of the last part of the thesis: data acquisition and analysis.

The chapter results displays the three outcomes of the experiments: The first segment will concentrate on tracking-quality of the different tracking systems.

With the output of the user-tests presented in two subsections. The first one is about the varying prediction quality of the four different actions, with the second one looking at the overall action recognition quality regarding the eight locations.

5.1 Tracking

Creation of the presented custom-tracker and embedding it in the 3D printed tracking testbed enabled an easy comparison of the three available trackers. Figures 21 and 22 present a sample of the tracking data acquired during construction and experimenting. In both cases relatively small samples -meaning time frames in this case- were selected for clarity, forasmuch overall data look similarly for larger sets.

Although not very intuitive as against to Euler angles, the Quaternion representation was the desired presentation as it is continuous and easy to compare.

Graph 21 is a comparison of the three used tracking systems.

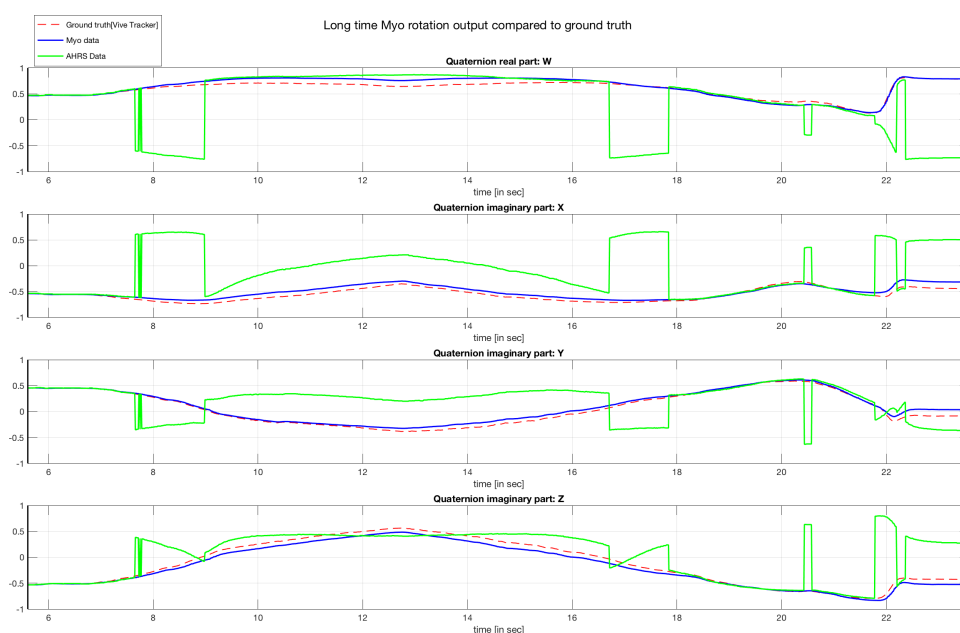


Figure 21: Data excerpt of the three tracking systems in quaternions.

The Vive Tracker is displayed by a dashed red line, the MYO armband's orientation-data is presented in blue. Lastly, the green line represents data of our custom created tracker.

While ground truth and MYO data are very similar, data of the custom created hardware shows a strange, "jumpy" behaviour, being even more present when displaying the data in VR.

As data originates from the custom tracker does not follow the readings of other trackers, the created tracker was dropped in favour of the better working MYO orientation data. All results presented from here on are created using the MYO armband as the main tracking system, with the HTC Vive trackers used to acquire a ground truth.

Figure 22 presents a comparison of MYO data over the optical ground truth coming from the HTC vive tracker. Originated from a user-test, the data spans a much longer time and shows -apart from outliers- close data resemblance between the two systems.

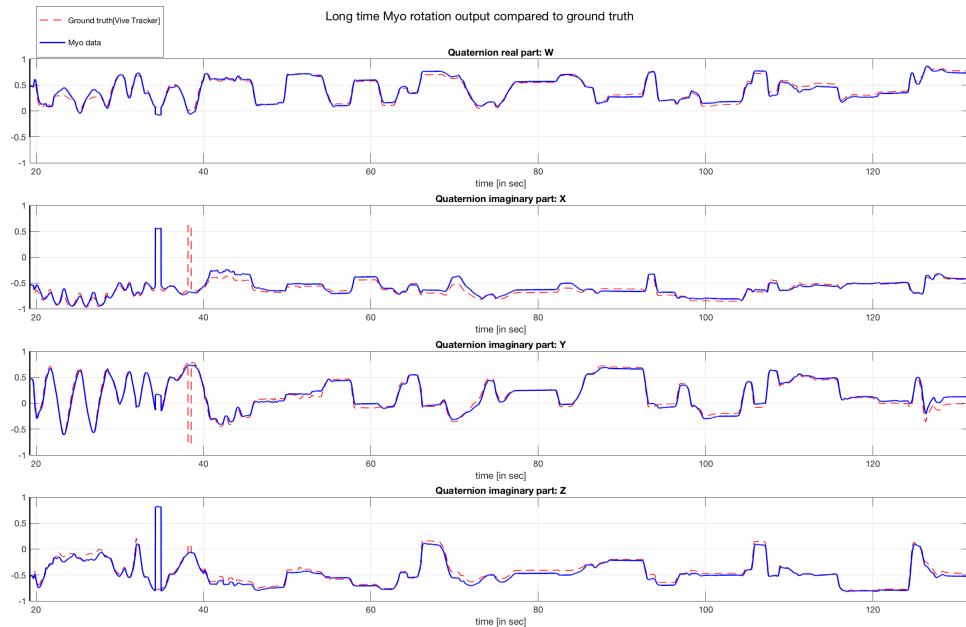


Figure 22: Plot of Vive Tracker (dashed red line) and MYO armband orientation data (blue) in quaternions over time. Given data being closely related, the MYO's data was finally used to control the virtual arm.

5.2 Participant data

As prefaced in section 4.5, the experiment-part of this thesis consisted of user-tests with 8 different participants. Each of them got a participant specific experiment-set and logging of the experiment was done in several ways.

Results presented in the following subsections are based on the 128 line holding logging-file seen in table 1. Encoding the 128 sub-tasks (four TaskSet à 32 sub-task each) each in a separate line, desired target and action as well as success or failure and different measured times can be mapped to a given sub-task.

Line	Task	TaskSet	Target	Action	Success	TCT	maxTT	TOR	TCT-TOR	TIT
1	1	1	0	0	0	15.00	15	2.248	12.76	0.7889
2	2	1	4	0	1	6.578	15	3.587	2.991	3.013
.
32	32	1	4	3	1	6.628	15	1.888	4.74	3.005
.
128	32	4	2	2	0	15.00	15	1.759	13.26	10.12

Table 1: Excerpt of the task logging file for participant no. 3. Consisting of 128 sub-tasks in the according number of lines, the file is composed of four taskSets à 32 sub-tasks each.

The first column holds the subtask-number [Task] followed by the taskSet-number [TaskSet], the desired target [Target] and action [Action] with a boolean value encoding success or failure of the sub-task [Success].

Task Completion Time [TCT] with the maximum time for each task [maxTT], the time of reach [TOR] as well as the subtraction of the TCT and TOR [TCT-TOR] and the Time In Task [TIT] form the second block.

While TCT -the Task Completion Time- and TIT -the (accumulated) Time In Task- were already referred to in 4.5, maxTT -the maximum task time- was only briefly mentioned and defined as 15 second. Also unfamiliar until now is the TOR -the time of reach- a measure in milliseconds, bearing information for the time it takes the participant to reach a desired target initially. The last remaining value, TCT-TOR, encodes a subtraction of the time of reach from the Task Completion Time.

5.2.1 Results represented over action

With results represented over the four trained actions, table 2 represents the average success rate for each action regarding to the eight participants. Whereas the last column illustrates the mean of the success rate for each action over all participants.

	Part1	Part2	Part3	Part4	Part5	Part6	Part7	Part8	Mean
rest	90.62	100.0	59.38	50.0	100.0	34.38	96.88	71.88	75.39
power	6.25	46.88	75.0	71.88	62.5	75.0	18.75	90.62	55.86
point	50.0	68.75	31.25	34.38	12.5	78.12	15.62	15.62	38.28
tri	21.88	100.0	68.75	96.88	87.5	40.62	50.0	40.62	63.28

Table 2: Average success rate of actions for the eight participants (Part1-8), with the mean action-success-rate over all participants represented in the last column.

Figure 23 illustrates the mean action-succes-rates graphically. Standard deviation and a mean over all actions is also presented in this chart.

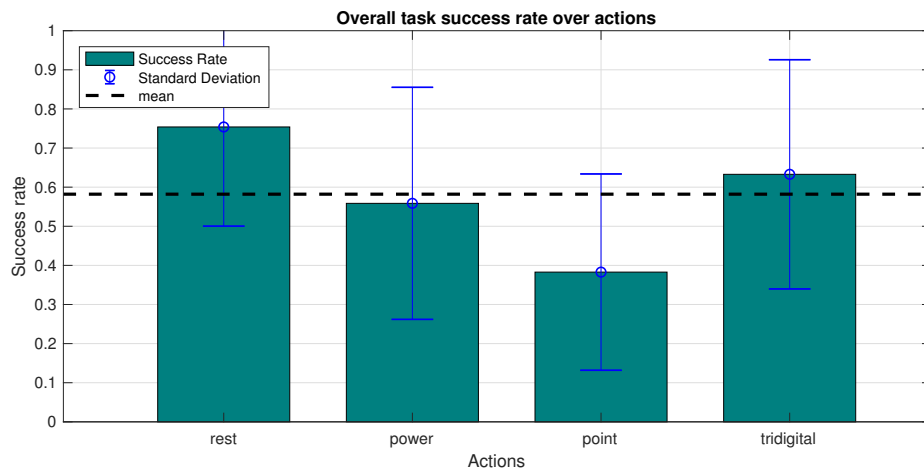


Figure 23: Plot of the overall task success rate over the given actions. The x-axis resembling the actions 'rest', 'power', 'point', 'tridigital' accordingly, with the y-axis representing the success rate in a scale from 0 to 1.

For further analysis, the obtained results were split in two groups: a group of successful and group of unsuccessful sub-tasks.

As measurements of a single participant tend to have a bias due to the aforementioned learning and wear over time, the data of all eight participants was averaged to present statistically meaningful results in the next three graphs.

A plot of Task Completion Time of the successful sub-tasks is presented in figure 24.

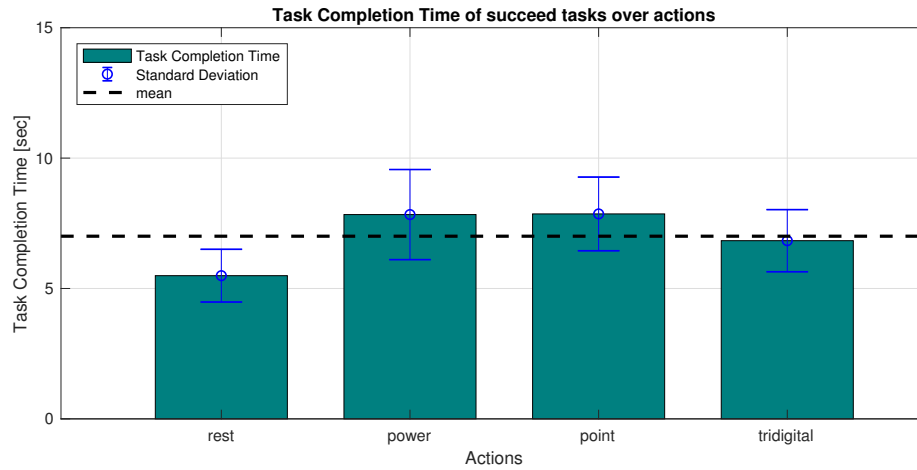


Figure 24: Plot of the Task Completion Time [TCT] for successful tasks over the given actions. The x-axis resembling the actions 'rest', 'power', 'point', 'tridigital' accordingly, with the y-axis representing the TCT in seconds.

With figure 25 presenting the Time In Task for unsuccessful sub-tasks.

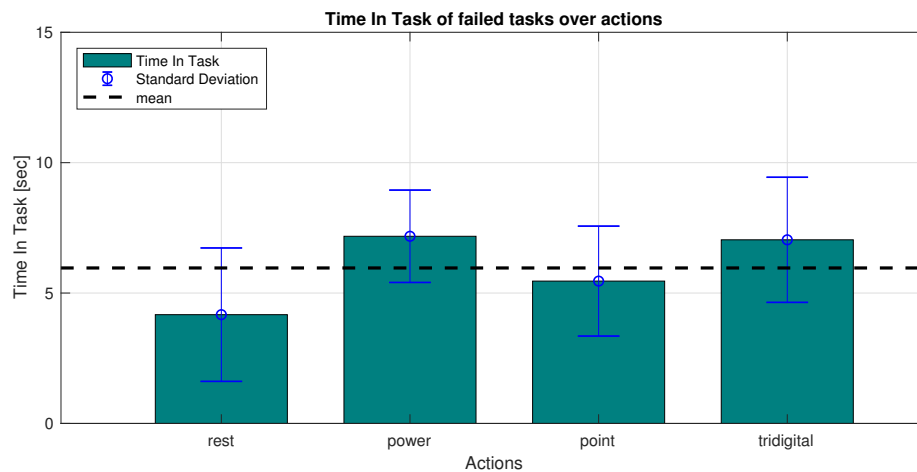


Figure 25: Plot of the accumulated Time In Task [TIT] for failed tasks over the given actions. The x-axis resembling the actions 'rest', 'power', 'point', 'tridigital' accordingly, with the y-axis representing the TIT in seconds.

5.2.2 Results represented over positions

As done similarly in the last section, this section presents participant-data related to the eight different target locations.

Table 3 represents the average success rate for each location regarding to the eight participants, and the overall mean success rate for each location over all participants.

	Part1	Part2	Part3	Part4	Part5	Part6	Part7	Part8	Mean
Pos0	43.8	62.5	43.8	37.5	56.2	31.2	50.0	25.0	43.7
Pos1	50.0	81.25	62.5	68.75	75.0	56.25	62.5	68.75	65.62
Pos2	43.75	81.25	68.75	56.25	75.0	68.75	43.75	56.25	61.72
Pos3	31.25	75.0	18.75	56.25	62.5	62.5	31.25	25.0	45.31
Pos4	56.25	87.5	81.25	75.0	81.25	50.0	56.25	62.5	68.75
Pos5	31.25	87.5	68.75	68.75	62.5	62.5	31.25	62.5	59.38
Pos6	31.25	75.0	43.75	62.5	56.25	50.0	43.75	81.25	55.47
Pos7	50.0	81.25	81.25	81.25	56.25	75.0	43.75	56.25	65.62

Table 3: Average success rate at locations for all participant with the mean success rate over all participants in the last column.

Analogously, graph 26 illustrates the mean success rate over the positions based on an average of all eight participants. As for all following depictions in this section, the locations were aggregated in the three groups presented in chapter 4.5, figure 15. Even though, success rates of the location groups differ, using the t-test as measure of choice, only the lower group was found to be statically different from the upper group (with a p-value of 0.0065).

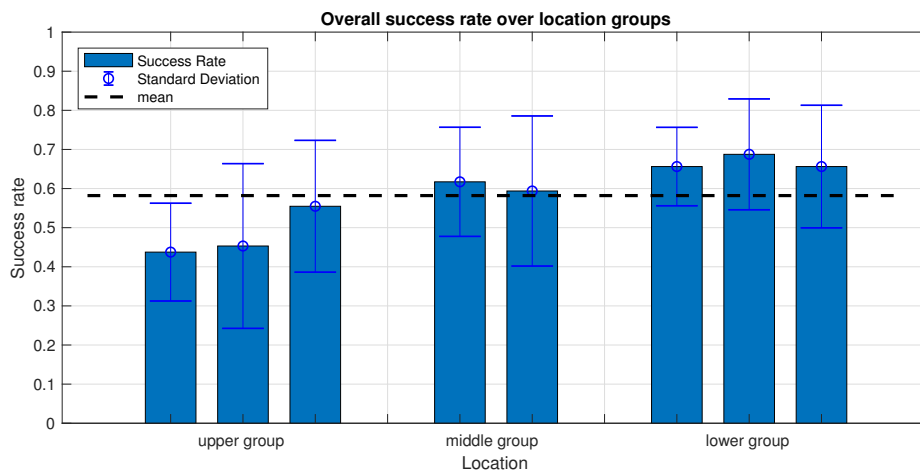


Figure 26: Plot of the overall task success rate over the given positions. The x-axis resembling the position groups 'upper group' [position 0, 3 ,6], 'middle group' [position 2, 5], 'lower group' [position 1, 4 ,7] accordingly, with the y-axis representing the success rate in a scale from 0 to 1.

Splitting the data in the same way as done in section 5.2.1, plot 27 depicts the location based Task Completion Time of successful sub-tasks.

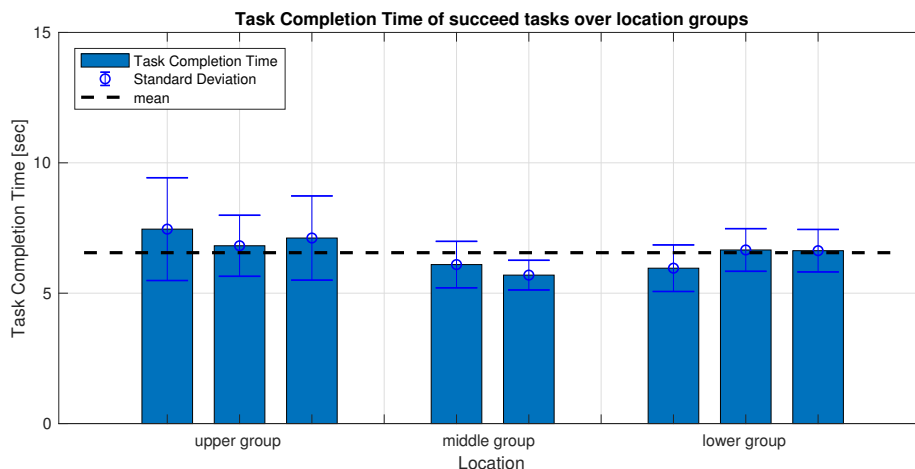


Figure 27: Plot of the Task Completion Time [TCT] for successful tasks over the given positions. The x-axis resembling the position groups 'upper group' [position 0, 3, 6], 'middle group' [position 2, 5], 'lower group' [position 1, 4, 7] accordingly, with the y-axis representing the TCT in seconds

As the last result-plot, figure 28 visualises the Time In Task for unsuccessful sub-tasks with regards to the given locations.

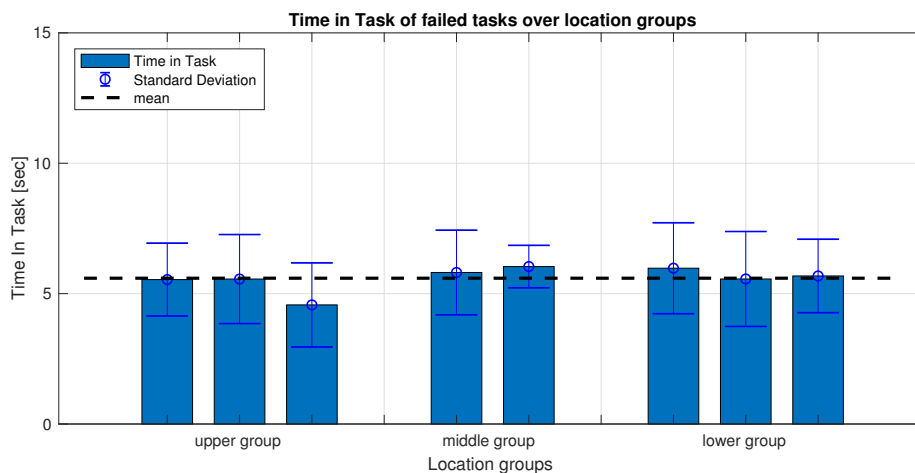


Figure 28: Plot of the accumulated Time In Task [TIT] for failed tasks over the given positions. The x-axis resembling the position groups 'upper group' [position 0, 3, 6], 'middle group' [position 2, 5], 'lower group' [position 1, 4, 7] accordingly, with the y-axis representing the TIT in seconds

“You can build something beautiful from stones that are put in your way.”

Erich Kästner

6

Discussion

With a presentation of the results in the preceding chapter, interpretation of them as well as thoughts about it will be presented briefly in this chapter.

First of, the quality of the designed custom tracker over the MYO’s orientation data and the ground truth of the optical Vive tracker will be analysed. Secondly, a synopsis of the acquired experiment data will be presented in regards of actions with a final review of the location based results.

6.1 The tracking

While the MYO’s orientation resembles data coming from a Vive Tracker -our desired ground truth to compare with- well enough to actually be used as a replacement for the latter, the custom tracker shows a strange, unforeseeable and ”jumpy” behaviour. This is most likely due to missparameterization of the used filter. Apart from the bad results, the presented set already portrays the best measurement produced using the custom tracker. Irrespective of how often a magnetic calibration was done or how the filter values were tuned, results resembled the data in figure 21.

With the custom tracker producing less optimal data (referring to fig. 21), and due to it being wired rather than connected wirelessly via BLE, the custom tracker was dropped in favour for the MYO armband. Why the acquired data is nowhere close to the quality of other trackers and what made the output ”jump” could not be fully determined.

In the aftermath of problems with porting device drivers for the different input devices, not much time was left for the implementation of a custom tracking solution. The reason

for the seen behaviour was possible hardware failure or, but most likely, poor sensor calibration or badly tuned sensor filtering/fusion.

While the presumably solution for the aforementioned problems could be found in setting different filter parameters, use of the MYO armband was a much quicker solution, as it already proved itself in practice. With close resemblance found in the depicted quaternion plots (figure 22), as well as a natural and almost drift free experience in VR, the MYO armband was used as the favoured IMU-based tracking system for the entire experiments that followed.

6.2 Experiment results

While section 5.2 is included to better understand the structure of the acquired data, chapter 5.2.1 presents more valuable information. Table 2 illustrates the action-success-rates for every single participant, where the huge deviation of the achieved success rates stands out. Likely based on different approaches during the training phase, many other factors as for example length of the experiment are influencing those success-rates, which makes easy interpretation in this scenario impossible. However they should be investigated in future experiments.

Over actions

With individual participants likely biased due to the aforementioned learning and wear over time, the values in the last column of table 2 as well as figure 23 present data averaged over all participants. While results are non-significant, the different actions still show a tendency. Interpreting a higher success rate as a better working intent-detection, the "rest"- and "tridigital"-actions work better while "power" worse, but still being more easy recognisable than the "point"-action.

Looking at figure 24, the Task Completion Time of the successful sub-tasks can be interpreted similarly to the mean action-success rates.

With a shorter Task Completion Time for better intention detection, the time scales according to the interpretation of figure 23: the better the task success rate, the shorter it takes a participant to successfully finish the task.

Interpretation of figure 25 is very counter-intuitive as the Time In Task for unsuccessful sub-tasks does not scale accordingly to the mean success rate or TCT for the actions. Expecting that "easier" actions would likely produce longer Time In Task values, the graph conversely shows a different outcome.

A review of the dataset revealed, that the reason for this unforeseen plot is likely the lack of data. To be more exact, some actions as "rest" show very high success-rates,

with some participants reaching even reaching the full 100 %. As their outcomes are not included in the subset of unsuccessful tasks, this leads to a bias towards results of the few unsuccessful participants. Namely, participant 6 who likely faced problems during the training-phase, resulting in a small Time In Task value, influenced the mean TIT for some actions drastically, as a much larger amount of his unsuccessful task were counted.

Over positions

While the same applies to location based data of participants as to any other single participant related data with it likely being biased, table 3 is not only included for the sake of completeness, but also shows that no participant produced a success rate of 100%, possibly reducing the impact on the subset of unsuccessful subtasks drastically.

The more interesting outcome presented in figure 26 shows a correlation of location and success-rate. While not always significantly different, the overall success rate around the training position (directly in between the two points of the middle group), has a tendency of getting worse on the upper and better on the lower point groups. With a t-test regarding upper and lower group even proving statistical significance.

Conversely to the presented action-based results, the plots of the location based TCT, plot 27 and plot 28 -visualising the TIT over location- substantiate the outcome. Plot 27 on the one hand can be similarly interpreted as the action based plot 23, with shorter Task Completion Time. Plot 28 on the other hand seems to be slightly biased as plot 25, with none of the locations to deviate far from the mean TIT of all unsuccessful tasks over the position.

“When the ambassadors of the Samians spoke at great length, the Spartans said to them, ”We have forgotten the first part, and the later part we did not understand because we have forgotten the first.”

Plutarch

7

Conclusion

In this, last chapter, a brief conclusion as well as an outlook for future work is presented. While closely tied to the obtained results, most of the ideas presented in the outlook section are based on my assumptions and have to be proven by other researchers.

7.1 Conclusion

With all defined goals met, including changes for the better, the here presented new iteration, VITA 2.0, is one of the few virtual reality setups incorporating such a large number of features.

With the decision to skip the direct integration of drivers, a much more modular VITA 2.0 setup came to be. Originally born due to time constraints and problems during portation, the solution to split the setup in a visualisation- and a data-handling-part by incorporating Unity3D and Interactive Myocontrol made it much more versatile. VITA 2.0 also became future proof, due to this splitting-solution circumventing possible platform incompatibilities of closed-source libraries or drivers, by making platform dependent compilation unnecessary.

Additionally each of the sub-modules adds countless features, may it be either Interactive Myocontrol’s support for many input devices, or Unity3D’s tools for creating the naturalistic, fully immersive virtual environment one can experience in VITA 2.0.

Whilst the creation of a custom tracking system failed, the presented fallback solution incorporating the MYO armbands’ orientation data was a great success. With completion

and control of a full arm up to the shoulder, and achieving pose control via real world, occlusion invariant input, the next step towards a virtual mirror box for home therapy was made.

Lastly, with data resembling real world counterparts, the user-centred VR experiments suggest that the used machine learning algorithms for arm-intent-detection not necessarily works equally well in different arm constellation. Further the outcome can be interpreted as: muscle constellation is not directly connected to the Cartesian coordinates of the arm end-effector. Similarly close end-effector positions can possibly be completely different regarding muscle constellation, with rotation of the lower arm likely having a high impact on the prediction quality.

With all this in mind, VITA 2.0 was an interesting project to work with, with the potential for many future iterations to come.

7.2 Outlook

At the end of the project, little time was left for trying new things. This write up presents a couple of ideas that could be further explored

Regarding the presented setup, repetition of the experiments with a bigger group of participants would be desired. It would particularly interesting to look at a setup taking into account actuation of both arms, with results likely to be mirrored. Besides that, experiments allowing retraining in phase 2, referring to the "formal description of the experiment" in 4.5, might have a different outcome, with overall better prediction quality. Even without recording more data, Fitts's law [77] could be applied to check if time requirements in VR mimic the real world.

Adapting the target grid to check for different arm constellations on training and reaching targets is another thing to be consider. Gaining valuable knowledge on overall constellation dependent prediction quality as well as probing for a better training location are possible.

Additionally changes regarding target behaviour, as for example targets following trajectories are conceivably. In this case, it would be equally interesting to look at the prediction quality and the recorded raw data as well as implementing online learning. While the first is straight forward to implement, the latter would involve redesign of the Interactive Myocontrol's learning methods alongside necessary improvements to the used

UDP protocol.

Possible but unknown to the author at this point is whether arm constellation is or is not a suitable input for the used machine learning methods. Taking into account the curse of dimensionality [78] the need for more training data - and therefore longer training - might likely outweigh the actual benefit. Nevertheless development of an algorithm dealing with the influence of arm posture in myocroncontrol should be investigated.

As the number of possible constellations might be a problem, a rather minimalistic approach could be investigated first, concerning only one single DOF of the arm pose. Namely the rotation of the lower arm. As this seems to effect the prediction quality a lot (see Simons et al. [79]), taking actual angle of the arm rotation as an extra input for learning might lead to better overall results, without bloating demands on the amount of training data too much.

Regarding the hardware-part of the setup, there is more to be desired. The first thing to be investigated are the problems with the presented custom trackers. While it is likely to be a software (filtering) error which has not been found due to the tight schedule, alternative hardware as the renown but also more expensive Bosch Sensortec BNO055 - Intelligent 9-axis absolute orientation sensor [80] should be investigated. Taking care of bluetooth communication and integration of it in Interactive Myocontrol -separating data acquisition and presentation for good- should than be the next step.

Doing so, one can step away from orientation tracking via MYO armbands and replace them with custom trackers, cutting weight and price of the system as well as making it more versatile. Separating tracking from EMG data acquisition also enables easier use of setups like the setup presented in [34]. On top of that, a third tracker could be placed on the spine of the participant to fully model the real world in VR and to accommodate for shoulder movement relative to the head.

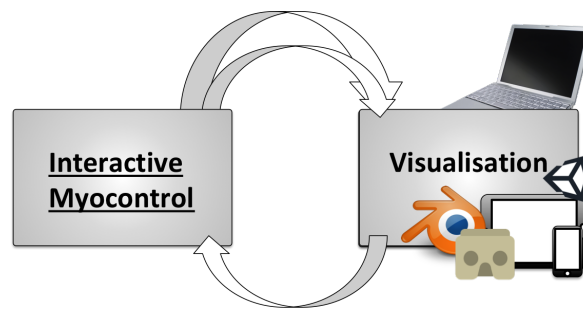


Figure 29: Graph of possible network-based-visualisation on different devices and platforms. With the UDP data protocol already in use, every device in the same network as the machine running Interactive Myocontrol can be reached, making data representation on PCs, tablets and smartphones possible. Without too many changes, VITA 2.0 could run on a Google Cardboard or any other inexpensive set of phone powered VR or AR goggles.

Based on the aforementioned changes, it would be possible to port the visualisation-part of the software over to any sort of device, even a mobile one. With a host computer running the Interactive Myocontrol software, any sort of client could capture the data. It would be splendid to have a port of the VR-world created in this thesis running on low-cost hardware as a smartphone, which can then create an immersive 3D world via Google Cardboard, making the system an optimal tool for a future home therapy setup.

8

Bibliography

- [1] K. Ziegler-Graham, E. J. MacKenzie, P. L. Ephraim, T. G. Travison, and R. Brookmeyer, “Estimating the prevalence of limb loss in the united states: 2005 to 2050,” *Archives of physical medicine and rehabilitation*, vol. 89, no. 3, pp. 422–429, 2008.
- [2] A. Houghton, P. Taylor, S. Thurlow, E. Rootes, and I. McColl, “Success rates for rehabilitation of vascular amputees: implications for preoperative assessment and amputation level,” *British Journal of Surgery*, vol. 79, no. 8, pp. 753–755, 1992.
- [3] V. S. Ramachandran and W. Hirstein, “The perception of phantom limbs. the do hebb lecture.” *Brain: a journal of neurology*, vol. 121, no. 9, pp. 1603–1630, 1998.
- [4] C. Castellini and V. Ravindra, “A wearable low-cost device based upon force-sensing resistors to detect single-finger forces,” in *Biomedical Robotics and Biomechanics (2014 5th IEEE RAS & EMBS International Conference on. IEEE*, 2014, pp. 199–203.
- [5] D. R. Falvo, “The rehabilitation specialist’s handbook,” *Journal of Applied Rehabilitation Counseling*, vol. 30, no. 1, p. 46, 1999.
- [6] Anon. Oxford university press - definition of amputation in english. [Online]. Available: <https://en.oxforddictionaries.com/definition/amputation>
- [7] M. F. Owings and L. J. Kozak, “Ambulatory and inpatient procedures in the united states, 1996.” *Vital and health statistics. Series 13, Data from the National Health Survey*, no. 139, pp. 1–119, 1998.

- [8] P. Dario, M. Carrozza, A. Menciassi, S. Micera, M. Zecca, G. Cappiello, F. Sebastiani, C. Freschi, S. S. Sant’Anna, and P. S. Valdera, “On the development of a cybernetic hand prosthesis,” in *Third IARP International Workshop on Humanoid and Human Friendly Robotics, IARP2002, Tsukuba, Japan, Dec.* Citeseer, 2002, pp. 11–12.
- [9] H. Flor, “Phantom-limb pain: characteristics, causes, and treatment,” *The Lancet Neurology*, vol. 1, no. 3, pp. 182–189, 2002.
- [10] C. M. Kooijman, P. U. Dijkstra, J. H. Geertzen, A. Elzinga, and C. P. van der Schans, “Phantom pain and phantom sensations in upper limb amputees: an epidemiological study,” *Pain*, vol. 87, no. 1, pp. 33–41, 2000.
- [11] E. E. Brodie, A. Whyte, and C. A. Niven, “Analgesia through the looking-glass? a randomized controlled trial investigating the effect of viewing a ‘virtual’ limb upon phantom limb pain, sensation and movement,” *European Journal of Pain*, vol. 11, no. 4, pp. 428–436, 2007.
- [12] J. Cole, S. Crowle, G. Austwick, and D. Henderson Slater, “Exploratory findings with virtual reality for phantom limb pain; from stump motion to agency and analgesia,” *Disability and rehabilitation*, vol. 31, no. 10, pp. 846–854, 2009.
- [13] J. R. Cram, *Cram’s introduction to surface electromyography*. Jones & Bartlett Learning, 2011.
- [14] N. Kunju, N. Kumar, D. Pankaj, A. Dhawan, and A. Kumar, “Emg signal analysis for identifying walking patterns of normal healthy individuals,” *Indian Journal of Biomechanics*, p. 118, 2009.
- [15] P. Artemiadis, “Emg-based robot control interfaces: past, present and future,” *Adv Robot Autom*, vol. 1, no. 2, p. e107, 2012.
- [16] G. ElKoura and K. Singh, “Handrix: animating the human hand,” in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, pp. 110–119.
- [17] I. Mashood Badshah, “Optical myography using convolutional neural networks for estimating finger poses,” Ph.D. dissertation, Technische Universität München, 2016.
- [18] C. Nissler, N. Mouriki, and C. Castellini, “Optical myography: detecting finger movements by looking at the forearm,” *Frontiers in neurorobotics*, vol. 10, p. 3, 2016.
- [19] D. Broderick, *THE JUDAS MANDALA*. Fantastic Books, 8 2009.

- [20] S. G. Weinbaum, *Pygmalion's Spectacles*. Bookclassic, 2015.
- [21] I. E. Sutherland, "A head-mounted three dimensional display," in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*. ACM, 1968, pp. 757–764.
- [22] S. Boyer, "A virtual failure: Evaluating the success of nintendo's virtual boy," *The Velvet Light Trap*, no. 64, pp. 23–33, 2009.
- [23] N. Owano. Google offers cardboard path to virtual reality. [Online]. Available: <https://techxplore.com/pdf323060960.pdf>
- [24] T. E. Lockhart and W. Shi, "Effects of age on dynamic accommodation," *Ergonomics*, vol. 53, no. 7, pp. 892–903, 2010.
- [25] A. Davis and F. Kühnlenz, "Optical design using fresnel lenses," *Optik & Photonik*, vol. 2, no. 4, pp. 52–55, 2007.
- [26] D. Brewster, *The Stereoscope; Its History, Theory and Construction, with Its Application to the Fine and Useful Arts and to Education, Etc.* John Murray, 1856.
- [27] A. Vlachos, "Advanced vr rendering," in *Game Developers Conference*, 2015.
- [28] J. T. Reason and J. J. Brand, *Motion sickness*. Academic press, 1975.
- [29] D. M. Johnson, "Introduction to and review of simulator sickness research," ARMY RESEARCH INST FIELD UNIT FORT RUCKER AL, Tech. Rep., 2005.
- [30] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey, "Point & teleport locomotion technique for virtual reality," in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. ACM, 2016, pp. 205–216.
- [31] S. Rawat, S. Vats, and P. Kumar, "Evaluating and exploring the myo armband," in *System Modeling & Advancement in Research Trends (SMART), International Conference*. IEEE, 2016, pp. 115–120.
- [32] M. Sathiyarayanan and S. Rajan, "Myo armband for physiotherapy healthcare: A case study using gesture recognition application," in *Communication Systems and Networks (COMSNETS), 2016 8th International Conference on*. IEEE, 2016, pp. 1–6.
- [33] C. Nissler, M. Connan, M. Nowak, and C. Castellini, "Online tactile myography for simultaneous and proportional hand and wrist myocontrol," 2017.

- [34] N. Jaquier, M. Connan, C. Castellini, and S. Calinon, “Combining electromyography and tactile myography to improve hand and wrist activity detection in prostheses,” *Technologies*, vol. 5, no. 4, p. 64, 2017.
- [35] N. Jiang, S. Muceli, B. Graimann, and D. Farina, “Effect of arm position on the prediction of kinematics from emg in amputees,” *Medical & biological engineering & computing*, vol. 51, no. 1-2, pp. 143–151, 2013.
- [36] T. Hilfert and M. König, “Low-cost virtual reality environment for engineering and construction,” *Visualization in Engineering*, vol. 4, no. 1, p. 2, 2016.
- [37] R. A. Pambudi, N. Ramadijanti, and A. Basuki, “Psychomotor game learning using skeletal tracking method with leap motion technology,” in *Electronics Symposium (IES), 2016 International*. IEEE, 2016, pp. 142–147.
- [38] H. Ling and L. Rui, “Vr glasses and leap motion trends in education,” in *Computer Science & Education (ICCSE), 2016 11th International Conference on*. IEEE, 2016, pp. 917–920.
- [39] F. Zhang, S. Chu, R. Pan, N. Ji, and L. Xi, “Double hand-gesture interaction for walk-through in vr environment,” in *Computer and Information Science (ICIS), 2017 IEEE/ACIS 16th International Conference on*. IEEE, 2017, pp. 539–544.
- [40] M. Alger, “Visual design methods for virtual reality,” *Ravensbourne*. http://aperturesciencellc.com/vr/VisualDesignMethodsforVR_MikeAlger.pdf, 2015.
- [41] C. Michel, C. Velasco, A. Salgado-Montejo, and C. Spence, “The butcher’s tongue illusion,” *Perception*, vol. 43, no. 8, pp. 818–824, 2014.
- [42] W. A. IJsselsteijn, Y. A. W. de Kort, and A. Haans, “Is this my hand i see before me? the rubber hand illusion in reality, virtual reality, and mixed reality,” *Presence: Teleoperators and Virtual Environments*, vol. 15, no. 4, pp. 455–464, 2006.
- [43] H. H. Ehrsson, N. P. Holmes, and R. E. Passingham, “Touching a rubber hand: feeling of body ownership is associated with activity in multisensory brain areas,” *Journal of Neuroscience*, vol. 25, no. 45, pp. 10 564–10 573, 2005.
- [44] R. N. Nielsen and B. Henriksen, “Comparison of movements in virtual reality mirror box therapy for treatment of lower limb phantom pain,” 2017.
- [45] M. C. Rodriguez, B. Aruanno, M. Bordegoni, M. Rossini, and F. Molteni, “Immer-

- sive virtual reality system for treatment of phantom limb pain (plp),” in *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2017, pp. V001T02A068–V001T02A068.
- [46] J. Dunn, E. Yeo, P. Moghaddampour, B. Chau, and S. Humbert, “Virtual and augmented reality in the treatment of phantom limb pain: a literature review,” *NeuroRehabilitation*, vol. 40, no. 4, pp. 595–601, 2017.
- [47] N. et al. Direct baxter teleoperation with multiple gesture control armbands. [Online]. Available: <https://www.youtube.com/watch?v=fSskyIaWkMk>
- [48] C. Li, C. Yang, J. Wan, A. S. Annamalai, and A. Cangelosi, “Teleoperation control of baxter robot using kalman filter-based sensor fusion,” *Systems Science & Control Engineering*, vol. 5, no. 1, pp. 156–167, 2017.
- [49] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill, “A hand gesture interface device,” in *ACM SIGCHI Bulletin*, vol. 18, no. 4. ACM, 1987, pp. 189–192.
- [50] D. C. Niehorster, L. Li, and M. Lappe, “The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research,” *i-Perception*, vol. 8, no. 3, p. 2041669517708205, 2017.
- [51] R. Montoliu, J. Torres-Sospedra, and O. Belmonte, “Magnetic field based indoor positioning using the bag of words paradigm,” in *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*. IEEE, 2016, pp. 1–7.
- [52] T. M. Takala and M. Matveinen, “Full body interaction in virtual reality with affordable hardware,” in *Virtual Reality (VR), 2014 IEEE*. IEEE, 2014, pp. 157–157.
- [53] D. Roetenberg, H. Luinge, and P. Slycke, “Xsens mvn: full 6dof human motion tracking using miniature inertial sensors,” *Xsens Motion Technologies BV, Tech. Rep*, vol. 1, 2009.
- [54] G. M. Bertolotti, A. M. Cristiani, P. Colagiorgio, F. Romano, E. Bassani, N. Caramia, and S. Ramat, “A wearable and modular inertial unit for measuring limb movements and balance control abilities,” *IEEE Sensors Journal*, vol. 16, no. 3, pp. 790–797, 2016.
- [55] X. Chen, “Human motion analysis with wearable inertial sensors,” 2013.

- [56] R. Mahony, T. Hamel, and J.-M. Pflimlin, “Nonlinear complementary filters on the special orthogonal group,” *IEEE Transactions on automatic control*, vol. 53, no. 5, pp. 1203–1218, 2008.
- [57] S. Madgwick, “An efficient orientation filter for inertial and inertial/magnetic sensor arrays,” *Report x-io and University of Bristol (UK)*, vol. 25, 2010.
- [58] K. Zhu and L. Shi, “Motion control in vr—real-time upper limb tracking via imu and flex sensor,” -.
- [59] S. Buckley, “This is how valve’s amazing lighthouse tracking technology works,” 05 2015. [Online]. Available: <https://gizmodo.com/this-is-how-valve-s-amazing-lighthouse-tracking-technol-1705356768>
- [60] S. Stein. (2017, 12) Vive tracker review. [Online]. Available: <https://www.cnet.com/products/htc-vive-tracker/review/>
- [61] V. Corporation. (2017, 01) Htc vive tracker developer guidelines ver. 1.3. Valve Corporation. [Online]. Available: https://dl.vive.com/Tracker/Guideline/HTC_Vive_Tracker_Developer_Guidelines_v1.3.pdf
- [62] (2017, 01). [Online]. Available: <https://www.vive.com/eu/product/vive-pro/>
- [63] T. Labs, “Myo support -how do i access the raw emg data from the myo armband?” [Online]. Available: <https://support.getmyo.com/hc/en-us/articles/202536726-How-do-I-access-the-raw-EMG-data-from-the-Myo-armband->
- [64] K. Nymoen, M. R. Haugen, and A. R. Jensenius, “Mumyo—evaluating and exploring the myo armband for musical interaction,” 2015.
- [65] (2016 - ongoing, 01). [Online]. Available: <https://developer.thalmic.com/forums/topic/541/>
- [66] (2012, 11). [Online]. Available: <https://venturebeat.com/2012/11/02/game-developers-start-your-unity-3d-engines-interview/>
- [67] E. V. de Kerckhove. Htc vive tutorial for unity. [Online]. Available: <https://www.raywenderlich.com/149239/htc-vive-tutorial-unity>
- [68] A. Gijssberts, R. Bohra, D. Sierra González, A. Werner, M. Nowak, B. Caputo, M. A. Roa, and C. Castellini, “Stable myoelectric control of a hand prosthesis using non-linear incremental learning,” *Frontiers in neurorobotics*, vol. 8, p. 8, 2014.

- [69] I. Strazzulla, M. Nowak, M. Controzzi, C. Cipriani, and C. Castellini, “Online bimanual manipulation using surface electromyography and incremental learning,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 3, pp. 227–234, 2017.
- [70] kristiel. Education & technology blog: Game design tutorial: How to use blend trees in unity. [Online]. Available: <http://www.studica.com/blog/game-design-tutorial-blend-trees-unity>
- [71] (2016, 12). [Online]. Available: <https://skarredghost.com/2016/12/09/how-to-use-kinect-with-htc-vive/>
- [72] S. Dóró, *Künstleranatomie: Menschliche Körper zeichnen*, 1st ed. Haupt Verlag, 03 2015.
- [73] (2017, 12). [Online]. Available: <https://learn.adafruit.com/comparing-gyroscope-datasheets?view=all>
- [74] N. Semiconductor, *nRF52832 Product Specification v1.4*, v1.4 ed., Nordic Semiconductor ASA, 10 2017.
- [75] (2018, 02). [Online]. Available: <https://learn.adafruit.com/nxp-precision-9dof-breakout?view=all>
- [76] X. Yan and P. Gu, “A review of rapid prototyping technologies and systems,” *Computer-Aided Design*, vol. 28, no. 4, pp. 307–318, 1996.
- [77] P. M. Fitts, “The information capacity of the human motor system in controlling the amplitude of movement.” *Journal of experimental psychology*, vol. 47, no. 6, p. 381, 1954.
- [78] R. E. Bellman, *Dynamic Programming of Continuous Processes: July 1954:(With Ext. Bibl.)*. Rand Corporation, 1954.
- [79] D. C. Simons and E. Zuniga, “Effect of wrist rotation on the xy plot of averaged biceps emg and isometric tension.” *American Journal of Physical Medicine & Rehabilitation*, vol. 49, no. 4, pp. 253–256, 1970.
- [80] B. Sensortec, “Data sheet bno055-intelligent 9-axis absolute orientation sensor,” 2014.



Appendix

Experiment description

Figure 30: Original Experiment description in german

<p style="text-align: center;"><i>Experiment:</i></p> <h2>Einleitung</h2> <p>Lieber Teilnehmer,</p> <p>zunächst vielen Dank, dass Sie an diesem Experiment teilnehmen. Wir werden im Folgenden beschreiben, (a) was Sie im Rahmen des Experiments tun sollen und (b) die Risiken, die damit verbunden sind. Bitte lesen Sie das folgende Dokument sorgfältig. Nachdem Sie das Dokument vollständig gelesen und verstanden haben, bitten wir Sie die Einverständniserklärung zu lesen und zu unterschreiben. Danach können wir beginnen.</p> <p>Diese Untersuchung wird durch das DLR sowie durch die DFG finanziert. Die Forscher haben keinen drohenden Interessenskonflikt. Das Experiment ist konform mit der Deklaration von Helsinki des Weltärzgebunds (World Medical Association, WMA), bezüglich den ethischen Grundsätzen für medizinische Forschung mit menschlichen Testpersonen, in der letzten Fassung. Diese wurde in der 59. WMA-Vollversammlung im Oktober 2008 in Seoul genehmigt.</p> <p>Wir wünschen Ihnen eine angenehme Erfahrung. Vielen Dank im Voraus für Ihre Teilnahme,</p> <p>die Durchführenden des Experiments Justin Iszatt Mathilde Comnan</p> <h2>Beschreibung des Experiments</h2> <p>Die Hauptaufgabe des Teilnehmers besteht darin eine vorgegebene Anzahl an farblich vorgehobenen Würfeln in der Virtualen Realität [VR] zu erreichen und dort eine definierte Hand-Pose einzunehmen.</p> <p>Dabei läuft das Experiment im Detail wie folgt ab:</p> <ol style="list-style-type: none"> 1. Der Teilnehmer legt die für den Versuch konzipierten Eingabegeräte in Form von zwei Bändern an Ober- und Unterarm an und erhält eine kurze Einweisung bezüglich ihrer Funktionsweise 2. Das System wird durch ein Anlernen an den Teilnehmer angepasst [dies geschieht in einer vorgegebenen Position]. Dabei werden vier Posen mit dreimaliger Wiederholung dargestellt und vom Probanden nachgemacht. 3. Nach dem Anlernen erfolgt eine 60 Sekunden lange Trainingsphase in der das System erprobt werden kann, bevor es richtig losgeht. 4. Beginn des Experiments und der Datenaufzeichnung: <ul style="list-style-type: none"> • Der Teilnehmer muss eine Reihe von farblich hervorgehobenen Würfeln in VR erreichen und dort eine vorgegebene Hand-Pose einnehmen. Gelingt ihm das und hält er die Pose dabei für .3sec, erscheint das nächste Ziel. Andernfalls wird nach 1.5sec automatisch das nächste Ziel gewählt • Die Gesamtheit des Experiments ist in vier sogenannte „TaskSet“ unterteilt. Insgesamt besteht ein als „TaskSet“ bezeichneter Aufgabenblock dabei aus 4 Sequenzen mit jeweils gleicher durchzuführender Hand-Pose in acht variierenden Positionen [Locations] 	<p style="text-align: center;"><i>Experiment:</i></p> <h2>Begleitrisiken</h2> <p>Für das Experiment benutzen wir:</p> <ol style="list-style-type: none"> a. Ein Myo-armband mit 8 Elektromyographiesensoren b. Eine HTC-Vive in Kombination mit zwei zugehörigen Trackern c. Und unsere in Unity 3D erstellte Software <p>Bei der Anwendung sollte es zu keinerlei Beeinträchtigungen des Teilnehmers kommen. In seltenen Fällen kann es zu sogenannter Simulator-Krankheit/Übelkeit kommen.</p>
--	---

Figure 31: Original Experiment description in german

<p style="text-align: center;"><i>Experiment:</i></p> <p>Teilnehmer ID:</p> <p>Gender: Female/Male</p> <p>Age:</p> <p>Hand dominance: Left/Right</p>	
<p style="text-align: center;"><i>Experiment:</i></p> <p>Einverständniserklärung</p> <p>Ich, der Teilnehmer, erkläre hiermit:</p> <ol style="list-style-type: none"> 1. Ich wurde ausführlich über die Art und Durchführung des Experiments aufgeklärt, sowohl mündlich als auch schriftlich. Ich werde die Anweisungen der Versuchsleiter genau beachten und den Bestimmungen hinsichtlich der Geräte, mit denen ich in Kontakt bin, Folge leisten. 2. Meine Teilnahme an diesem Experiment ist freiwillig. 3. Ich kann zu jeder Zeit das Experiment abbrechen. 4. Während des Experiments werden Daten aufgezeichnet. Mir ist bekannt, dass diese Daten elektronisch am DLR gespeichert werden, und dass diese Daten anonym sind. Dies bedeutet, dass keine externe Person diese Daten in eine Beziehung zu mir bringen kann. Mir ist bekannt, dass diese Daten analysiert und mit Ergebnissen dieser Analyse veröffentlicht werden können. 5. Mir ist auch bekannt, dass ich zu jeder Zeit die permanente Löschung dieser Daten fordern darf. <p>Frau Mathilde Coman und Herr Justin Iszatt sind für diese Untersuchung verantwortlich. Ich habe diese Anweisungen sorgfältig gelesen und verstanden.</p>	<p style="text-align: center;">Ort und Datum _____</p> <p style="text-align: center;">Name & Unterschrift des Teilnehmers _____</p> <p style="text-align: center;">Name & Unterschrift des Versuchsleiters _____</p>