Matthias Schlesinger, BSc.

An LSL-Based Sensor Platform for Mobile Brain Imaging, Brain-Computer Interfaces and Rehabilitation

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Software Development and Business Management

submitted to

Graz University of Technology

Supervisor

Ass.Prof. Dr. Reinhold Scherer

Institute of Neural Engineering

Graz, February 2016

Matthias Schlesinger, BSc.

Eine LSL-Basierte Sensor Plattform für Mobile Hirnforschung, Gehirn-Computer Schnittstellen und Rehabilitation

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Softwareentwicklung - Wirtschaft

eingereicht an der

Technischen Universität Graz

Betreuer

Ass.Prof. Dr. Reinhold Scherer

Institut für Neurotechnologie

Graz, Februar 2016

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, 02.02.2016 ______________________ Date

______________________ Signature

Eidesstattliche Erklärung[1]

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am 02.02.2016 ______________________ Datum

______________________ Unterschrift

[1]Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Abstract

Even though the brain proves constantly that it performs amazingly well in complex, ever-changing environments the majority of brain research is conducted while participants sit or lie motionless. Similarly, while the ultimate goal of brain-computer interface (BCI) systems is to aid patients in need, hardly any BCI system can be used by a patient at home without a supervisor present. Rehabilitation exercises, however, can be done by patients at home but due to their repetitive, monotone nature are often neglected.
Those seemingly unrelated statements express one and the same challenge: the transition from a stationary to a mobile environment or in other words the challenge of making technologies usable outside the laboratories and medical clinics to advance research and aid patients.
The aim of this thesis was to further explore the three fields mobile brain imaging, mobile BCI and mobile rehabilitation. Firstly, the requirements those areas impose on both hardware and software were assessed. For hardware devices those requirements include being lightweight, portable and ideally wireless. Software is required to be able to synchronize multiple devices, be easy to use and not to rely on proprietary software. Furthermore, both hardware and software should be affordable for patients that already have to bear high medical costs.
Based on those results the data acquisition and synchronisation software Lab Streaming Layer (LSL) was extended so that it supports hardware devices that facilitate this before-mentioned transition. Those hardware devices include medical-grade hardware for mobile brain research such as the ANT eegosports amplifier (ANT Neuro, Enschede, Netherlands) and two data gloves, namely, the 5DT DataGlove (5DT, Gauteng, South Africa) and the CyberClove (CyberGlove Systems, San Jose, CA, USA) but also affordable consumer-grade devices such as the Leap Motion controller (Leap Motion Inc., San Francisco, CA, USA) or the Thalmic Myo wristband (Thalmic Labs Inc., Ontario, Canada).
All this yields a software platform that can easily access and synchronize a wide range of hardware devices. To demonstrate its usefulness for mobile rehabilitation a proof-of-concept for a computer game for rehabilitation is presented. The game uses two of the newly implemented devices to control the

main character and LSL to record the data coming from those devices as well as in-game events.

Kurzfassung

Obwohl das menschliche Gehirn permanent beweist, wie gut es komplexe Situationen meistern kann, werden Teilnehmer in Hirnforschungsstudien dazu angehalten, sich so wenig wie möglich zu bewegen. Gleichzeitig ist es zwar ausgewiesenes Ziel von Gehirn-Computer Schnittstellen (auch: brain-computer interfaces, BCIs), Patienten zu helfen. Tatsächlich bei Patienten zu Hause zum Einsatz kommen aber ob ihrer Komplexität nur sehr wenige BCI Systeme. Physiotherapeutische Übungen zu Rehabilitationszwecken, andererseits, könnten zwar durchaus zu Hause durchgeführt werden, scheitern aber oft an der fehlenden Motivation der Patienten. Auf den ersten Blick mögen diese Feststellungen etwas beliebig gewählt sein. Bei genauerem Hinsehen offenbaren sie allerdings ein und dieselbe Herausforderung: den Übergang von einer stationären in eine mobile Umgebung oder, mit anderen Worten, die Herausforderungen, Technologien außerhalb von Forschungseinrichtungen und Kliniken verwendbar zu machen, mit dem Ziel, Forschung voranzutreiben und Patienten zu helfen.

Das Ziel dieser Arbeit war es, die drei Felder mobile Hirnforschung, mobile Gehirn-Computer Schnittstellen und mobile Rehabilitation und die Anforderungen, die sie an Hardware und Software stellen, zu erforschen. Zu den Anforderungen zählen die Verwendung portabler, kabelloser Hardwaregeräte sowie benutzerfreundliche Software, die in der Lage ist, mehrere solcher Hardwaregeräte zu synchronisieren. Außerdem sollten für sowohl Hard- als auch Software leistbare Lösungen angeboten werden, um Patienten, die ohnehin schon hohe Ausgaben zu tragen haben, nicht weiter zu belasten.

Ausgehend von diesen Ergebnissen wurde die Software Plattform Lab Streaming Layer (LSL) erweitert, so dass Hardwaregeräte, die den oben genannten Übergang in eine mobile Umgebung erleichtern, unterstützt werden. Zu den implementierten Geräten zählen medizinische Geräte wie der ANT eegosports EEG-Verstärker (ANT Neuro, Enschede, Netherlands) und zwei verschiedene Datenhandschuhe, und zwar der 5DT DataGlove (5DT, Gauteng, South Africa) und der CyberGlove (CyberGlove Systems, San Jose, CA, USA), aber auch günstige Consumer-Geräte wie der Leap Motion Controller (Leap Motion Inc., San Francisco, CA, USA) oder das Thalmic Myo Armband (Thalmic Labs Inc., Ontario, Canada).

Abschließend wird die Praktikabilität der entstandenen Software Plattform am Beispiel der spielbasierten Rehabilitation demonstriert. In diesem Machbarkeitsnachweis werden zwei der neu implementierten Geräte sowie LSL zur Erfassung des Spielfortschritts genutzt.

Acknowledgements

I would like to express my gratitude to my supervisor Prof. Reinhold Scherer for his support and guidance, to Martin Seeber for sharing his knowledge and for helping me conducting and analysing experiments as well as to the entire Lab team for constructive comments and suggestions.

Furthermore, I would like to thank Christian Kothe and David Medine from the University of California, San Diego, for answering countless LSL-related questions and for insightful discussions.

My sincere thanks also goes to Prof. Rajesh Rao, the University of Washington, Seattle, as well as the Center for Sensorimotor Neural Engineering for making my research trip to Seattle possible and to the GRID Lab team and Dimi Gklezakos for welcoming me in their lab and making my time in Seattle as great as it was.

Lastly, I would like to thank my family and my girlfriend for supporting me throughout my studies and beyond.

Contents

List of Figures

Glossary

ALS	Amythrophic Lateral Sclerosis.
API	Application Programming Interface.
BCI	brain-computer interface.
CP	Cerebral Palsy.
ECOG	electrocorticography.
EEG	electroencephalography.
EMG	Electromyography.
ERD	Event-Related Desynchronization.
ERP	Event-Related Potential.
FFT	fast Fourier transform.
fMRI	Functional Magnetic Resonance Imaging.
hBCI	Hybrid BCI.
HCI	human computer interaction.
IMU	Inertial Measurement Unit.
LSL	Lab Streaming Layer.
MEG	Magnetoencephalography.
MoBI	mobile brain imaging.
PET	Positron Emission Tomography.
SSVEP	Steady-State Visually Evoked Potential.
UI	User Interface.

XML	eXtensible Markup Language.

1. Introduction

1.1. Motivation

On a first glance brain imaging - the process of recording and imaging the structure and activity of the human brain -, brain-computer interfaces (BCIs) - systems that record and analyse brain activity to control a computer - and rehabilitation have very little in common. But all these areas are currently facing a similar challenge. Namely: the transition from a stationary to a mobile environment. Or in other words the challenge of making existing technologies usable outside controlled research laboratories and medical clinics.
For brain-imaging that transition to a mobile environment would mean being able to record brain activity while participants actively interact with their environment. For a brain-computer interface system to be truly mobile it must be both affordable and easy to use so that it can be used by patients at home. And, lastly, mobile, home-based rehabilitation would allow patients to perform the majority of their exercises at home while a computer- and/or game-based system tracks their progress.

With this imminent change in environment also the hardware requirements change. Brain imaging in a mobile environment calls for compact recording devices that are light enough to be worn on the participant's body and use very few cables to not limit the participant's range of motion. Furthermore, both mobile brain-computer interfaces and mobile rehabilitation require hardware to be robust, easy to use and most importantly to be affordable for patients. So having one sensor platform that combines all those hardware devices is of utmost importance and would be the first step towards a software system that facilitates the mobile scenarios mentioned above. The platform should acquire data from many different hardware devices and make this data available to whichever application is interested.

The aim of this thesis is to do just that. Therefore, an existing software platform called the Lab Streaming Layer (LSL) is extended to facilitate mobile brain imaging, mobile brain-computer interfaces and mobile rehabilitation. To do so support for a wide range of new hardware devices that meet the requirements briefly outlined above was added to LSL.

1.2. Structure

The structure of this work is as follows. Chapter 2 introduces the key concepts that are relevant for this thesis. The fields of mobile brain imaging (section 2.1), mobile BCIs (2.2) as well as mobile rehabilitation (2.3) are presented and their implications on both hardware and software are discussed.
Subsequently, the existing software platform LSL will be introduced in chapter 3. After a general introduction in section 3.1 its applicability for the three mobile scenarios will be assessed in section 3.2. Based on this information, the following chapter 4 gives an overview of promising hardware devices that can facilitate mobile brain imaging, mobile BCI and mobile rehabilitation but are not yet supported the Lab Streaming Layer.
The second half of this work focuses on the software developed as part of this thesis. In chapter 5 new applications written for LSL are described. In section 5.1 the LSL Configurator, a tool that facilitates configuration of multi-application projects, is presented. After that a description of the applications written for the hardware devices discussed in chapter 4 is given in sections 5.2 and 5.3. Test results using those applications are reviewed in the subsequent chapter 6. To conclude the discussion of the practical part of the thesis a proof-of-concept of an LSL-based rehabilitation game is presented in chapter 7.
Finally, the presented work is discussed in chapter 8 before this thesis is concluded in chapter 9.

2. Background

This chapter will introduce the three emerging areas mobile brain imaging, mobile brain-computer interfaces and mobile rehabilitation that should be facilitated by the sensor platform presented in this thesis. Furthermore, requirements and challenges associated with the transition from a stationary to a mobile environment for each of these areas will be discussed.

2.1. Mobile Brain Imaging

For the purpose and importance of mobile brain imaging (MoBI) to become evident firstly the traditional brain imaging process has to be described.

2.1.1. Traditional Brain Imaging

Brain imaging measures neuronal activity. This activity can be detected for instance by either measuring electrical activity (Electroencephalography (EEG)) or magnetic fields (e.g. Magnetoencephalography (MEG)) when groups of neurons become active or by measuring cortical blood flow (Functional Magnetic Resonance Imaging (fMRI), Positron Emission Tomography (PET)) working under the assumption that active brain regions require more blood than inactive ones (Crosson et al., 2010; Gramann et al., 2011).
Most of those methods (PET, fMRI, MEG) require the use of large scanners. EEG uses more lightweight electrode caps that are placed on the participants head. What all those methods have in common though is, that they are usually conducted in soundproof and electrically shielded laboratories and that they require the participant to lie or sit still and move as little as possible (Kranczioch et al., 2014). The reason for that is that movements contaminate the recorded brain signals with non-brain related noise or artefacts (Makeig et al., 2009). Such artefacts can be induced by head and neck movements as well as by processes the participants might be unaware of such as eye-blinking (Gramann et al.,

2011).

Even though those well-established recording paradigms restrict the participant's movements they have led to countless important findings about how the human brain works and have given us a better understanding of motor control, emotion, attention and many others (Gramann et al., 2014). Nevertheless, researchers argue that with only using this "limited behavior approach" as Makeig et al. (2009) calls it, the human brain in its natural state cannot be explored.

2.1.2. Mobile Brain Imaging

Traditional brain imaging requires participants to move as little as possible to not contaminate the recorded brain signal. Nevertheless, we know that the human brain is trained to adjust to an ever-changing environment and to react to events in our surroundings automatically and in a timely fashion (Gramann et al., 2011; Makeig et al., 2009). Studies on animals have shown that brain states differ depending on the behavioural state. Niell and Stryker (2010) described that the activity in a mouse's visual cortex doubles when it starts moving. This suggests that the brain adjusts to movement and depending on the amount of incoming sensory information (Gramann et al., 2011). Traditional brain imaging studies where participants are not allowed to move disregard this link between human behaviour and brain activity (Kranczioch et al., 2014).

So the idea of mobile brain imaging is to simultaneously record brain activity and body movements while participants are actively interacting with their environment (Gramann et al., 2011) and to thereby give a more detailed answer to how brain dynamics and human behaviour are linked. When MoBI was proposed it was defined as the simultaneous study of what the brain is doing, what the brain is sensing, and what the brain is controlling while performing naturally motivated actions. This requires the recording of brain activity, body and eye movements as well as everything the subject sees and hears (Delorme et al., 2011). However, ordinarily the term MoBI also refers to non-stationary brain imaging where no or hardly any scene information is captured. Most notably in this regard are studies in human upright walking that record EEG and data from foot-force sensors (e.g. Seeber et al., 2015, Lau et al., 2014, De Sanctis et al., 2014).
But regardless of the amount of environment data that needs to be captured

the new experimental modalities introduced with MoBI impose many new requirements on both hardware and software.

2.1.3. MoBI Requirements

There are two main requirements for mobile brain imaging technologies.

1. **Precision of Measurement:** The precision of measurement of brain imaging technologies is usually determined by its temporal and spatial resolution. Temporal resolution indicates how precise a method can measure changes in neural processing. This means how long it takes for a change in brain activity to become apparent in the recorded data. Spatial resolution indicates how precisely the origin of the change in brain activity can be pinpointed. As goal-directed movements are initiated and executed within milliseconds, technologies used for MoBI must have high, millisecond accurate, temporal resolution. To effectively remove artefacts from the signal a spatial resolution in the range centimetres is required (Gramann et al., 2011; Mullen et al., 2015).
2. **Portability:** The technology of choice must be portable enough to allow the subject to move freely. It should be small, lightweight and ideally head mounted and transmit data wirelessly or through very few cables (Kranczioch et al., 2014; Makeig et al., 2009).

In section 2.1.1 commonly used brain imaging technologies were mentioned and should now be assessed for MoBI. FMRI and PET scanners, that measure the blood flow, have great spatial resolution (millimetre accurate) but have very low temporal resolution (seconds for fMRI, tens of seconds for PET). Moreover, due to the scanner size they require the subject to remain stationary and are too heavy for a mobile solution. Same holds true for MEG. While it has better temporal resolution (in the range of milliseconds) than the methods above it shares their disadvantage of being inherently stationary because of the MEG scanner's size. Hence it does not fulfil the portability requirement for mobile brain imaging that was discussed above. Hence, the most promising recording technology for mobile brain imaging is EEG. It provides excellent temporal resolution in the range of milliseconds, acceptable spatial resolution in the range of centimetres and is lightweight enough to allow the subject to move around freely (Gramann et al., 2011; Crosson et al., 2010).

Therefore when Makeig et al. (2009) introduced the concept of MoBI for the first time they proposed an EEG based mobile brain imaging system. In their original proposal they used a 256 channel high density EEG that, due to the size and weight of the EEG recording hardware, was mounted on a crane-like structure above the participant (Gramann et al., 2011). This setup, while a big step forward from stationary recordings, still restricts the participants' movements (Kranczioch et al., 2014). With recent developments regarding mobile EEG amplifiers mobile brain imaging can be further improved. As opposed to traditional EEG amplifiers mobile amplifiers are extremely portable. They are either fully head-mounted or lightweight enough to be worn on the participant's body and transmit their data wirelessly.

With the adapted hardware requirements also the software requirements change significantly for MoBI recordings compared to traditional EEG recordings. Traditionally only EEG data and event markers have been recorded. With MoBI much more data, including motion capture and audio and video data, is recorded simultaneously (Ojeda et al., 2014).
The main challenge here is the synchronization of the multiple devices recorded. As processes studied with MoBI are initiated and executed within milliseconds even the slightest time shifts can lead to misinterpretations of the results (Gramann et al., 2011; Reis et al., 2014). Additionally, the recording devices have different internal clocks and potentially different sampling rates. Also they might not run on the same computer device. Hence network and operating system delays complicate the synchronization process (Delorme et al., 2011).

2.2. Mobile Brain-Computer Interfaces

2.2.1. Brain-Computer Interfaces

Brain-computer interfaces (BCIs) are systems that record brain activity and translate this activity into control commands for a (computer) device (Rao, 2013). Brain activity can be recorded using invasive (e.g. electrocorticography (ECOG)) and non-invasive (e.g. EEG, MEG) technologies. For this thesis the term BCI refers to EEG-based BCIs.
Originally, BCIs were developed to aid patients having only very limited neuromuscular control which can be caused by diseases like Amythrophic Lateral Sclerosis (ALS) and Cerebral Palsy (CP) but also spinal cord injuries or stroke (Klose, 2007). Those conditions have in common that the normal information flow from the human brain to the muscles is disrupted. Either due to damage

to the neural pathways that control muscles or to the muscles themselves. BCIs give the brain a new, non-muscular, communication and control channel and the patient a way to interact with its environment (Wolpawa et al., 2002). In recent years fostered by faster and cheaper computers and advances in our understanding of how the brain works researchers began to explore BCIs for able-bodied individuals as well (Rao, 2013).

BCI Applications

As mentioned above BCI systems play an important role in a clinical environment to help so called locked-in patients. The goal of BCI applications for this group of patients is to most importantly provide means for basic communication and movement control. Such applications include spellers (e.g. Farwell and Donchin, 1988) or cursor control (e.g. Wolpaw et al., 1991) as well as environment control (e.g. Gao et al., 2003). Furthermore, BCI systems might be used as a therapeutic tool to help patients with motor disabilities, most notably patients that suffered from stroke, to relearn useful motor function. Either by training them to produce more normal brain activity as shown by Buch et al. (2008) or by using BCI to control a movement assisting device as Daly et al. (2008) proposed.

With regard to able-bodied individuals BCI research aims at navigating virtual worlds for gaming (Scherer et al., 2011) or monitoring alertness to assess the degree of engagement and attention in a learning environment (Szafir and Mutlu, 2012).
In conclusion, applications for BCI systems are diverse as are potentials and requirements for mobile and affordable solutions. Those will be described in the following section.

2.2.2. Hybrid BCI

Hybrid BCI (hBCI) are systems that combine a BCI with either another BCI system (e.g. Pfurtscheller et al., 2010) or other input devices (e.g. Scherer et al., 2007). The former will be refereed to as hybrid BCI-BCI systems. The latter, as they utilize human computer interaction (HCI) hardware, will be refereed to as hybrid BCI-HCI systems. BCI-BCI systems use the same acquisition hardware and are only hybrid in the sense that they combine two different signal processing approaches (Pfurtscheller et al., 2010). Hence they are of lesser

interest for this thesis. BCI-HCI systems, however, utilize multiple hardware devices simultaneously and could benefit from a sensor platform that supports a wide range of such devices.
According to Müller-Putz et al. (2011) typical applications of hybrid BCI-HCI systems are applications for severely disabled but not completely locked-in patients. As the authors point out these patients are usually able to interact with their environment using a wide range of assistive devices. Such devices are favoured over BCIs as they provide a more natural and reliable way of communication. However, fatigue, tremors or other inferences might render the patient unable to physically control a device. Then an input method that does not rely on muscular activity such as a BCI becomes an essential alternative. Consequently, BCI-HCI systems monitor both traditional input devices, such as game controllers, and a BCI simultaneously and provide flexible and adaptive means of communication (Kreilinger et al., 2011).
Another application for hybrid BCI-HCI systems is game-based rehabilitation as proposed by Muñoz et al. (2013). Here BCIs are combined with a Kinect (Microsoft Inc., Seattle, WA, USA) motion-tracking sensor. The BCI is used as an additional input channel as well as to monitor the patient's attention. The field of mobile rehabilitation is discussed in section 2.3 in more detail.

2.2.3. Mobile BCI

While helping patients in need is the ultimate goal when developing a BCI the majority of BCI research is conducted in laboratories. And the number of patients that are able to use a BCI system at home without experts present is still very low (Nijboer and Broermann, 2010).
The idea of mobile BCIs is to create BCI systems that are suitable for daily life applications outside a laboratory without requiring constant medical and/or technical supervision (De Vos et al., 2014). Hence the meaning of *mobility* is quite different to mobile brain imaging where *mobility* only meant moving freely in a controlled, scientific environment. Similarly, this transition from a controlled to an uncontrolled, mobile environment causes the requirements for BCI hardware and software to change as the following section will discuss.

2.2.4. Requirements for Mobile (Hybrid) BCIs

While some demands from mobile BCI systems vary among the different user-groups there are some core requirements that apply to all mobile scenarios.

Firstly, the recording devices should be robust, portable, lightweight and easy to set up (Nijboer et al., 2014). Additionally, they should require as few wired connections as possible or ideally transmit data wirelessly (Debener et al., 2012). This is particularly important for hybrid systems as they by definition utilize more than one device. Hence, wireless solutions should be preferred as cables are not only inconvenient but a potential safety threat for motor impaired patients.

Another factor that is of lesser importance when developing BCI systems for a research environment is usability. Traditionally, EEG-based BCI systems use wet electrodes. While the use of conductive gel leads to better signal quality it also takes significantly longer to set up and requires hair washing after use (Reis et al., 2014). This is inconvenient for able-bodied users but an enormous challenge for severely disabled patients and their caregivers (Nijboer and Broermann, 2010). While most of the consumer-grade EEG systems use dry electrodes they usually provide too few and too inaccurate electrodes for high accuracy recordings particularly in mobile scenarios. Regarding the number of electrodes, Lau et al. (2012) suggest at least 35 electrodes for EEG recordings during locomotion whereas available consumer-grade systems provide fewer electrodes (e.g. 14 in the case of the device presented in section 4.2.1). Regarding achieved accuracies, Estepp et al. (2009) showed that the correlation between a signal recorded with dry and wet electrodes can drop to as low as 0.45 while never being higher as 0.82. However, very recently Mullen et al. (2015) demonstrated a wireless EEG system for mobile recordings that provided 64 electrodes and, with a signal correlation of 0.9, achieved comparable signal quality to simultaneously recorded wet electrodes.

The transition from a lab environment to the patient's home does not only require improved hardware but also a new take on BCI software. In this regard Kaufmann et al. (2012) note that existing BCI software systems are extremely complex. They allow researchers to modify a wide range of parameters to configure everything from stimulus presentation over signal processing and classification. While the achieved flexibility is advantageous in a research environment it is overwhelming for layman. Hence, existing BCI software needs to be radically simplified so that the end-user or his/her caregiver can use the system without requiring medical supervision. Faller et al. (2012), Kaufmann et al. (2012), and Kübler et al. (2013) recently proposed such systems that combine a simple user interface with auto-calibration and, for the systems presented by Kaufmann et al. and Kübler et al., also game-based elements. Similarly, Scherer et al. (2015) presented a user-centred, game-based BCI to increase the subjects' motivation and cooperation during the demanding BCI calibration phase.

But probably the most important requirement that holds true for both BCI

hardware and BCI software is to be affordable. Nijboer and Broermann (2010) pointed out that patients that require a BCI system usually face enormous medical costs for inpatient treatment, drugs, physical therapy or costly adaptations at home. They also stated that some of those expenses are not covered by health insurances as are the acquisition costs for a BCI system. Consequently, developing devices that provide good accuracies without being as expensive as currently used medical-grade equipment must be a priority. The same holds true for BCI software. The objective must be to create a system with low hardware requirements that does not rely on proprietary software like the commonly used and fairly expensive MATLAB (MathWorks Inc., Natick, MA, USA) environment.
For hybrid BCIs those software requirements are identical. Additionally, due to the simultaneous use of multiple devices in hybrid BCI-HCI systems, device synchronization is an particularly important requirement for hBCI software. The challenges associated with multi-device synchronization were described in detail in section 2.1.3.

2.3. Mobile Rehabilitation

Other than the before mentioned mobile brain imaging and mobile BCIs the field of mobile rehabilitation has been more widely used and studied. The reason for that is simple. With an ageing society and a higher number of often multiple and complex health conditions not only the target population but also the socio-economic (health-care) costs are significantly higher (Patel et al., 2012). For example stroke alone leads to 5 million long-term disabled patients each year (Burke et al., 2010).
The idea of mobile rehabilitation is to make rehabilitation exercises available for patients at home. So, similar to mobile BCIs, *mobility* in this context means providing hardware and software solutions that are usable outside medical clinics without requiring the presence of medical doctors or therapists. The main objectives of doing so is to reduce healthcare costs and lengths of inpatient rehabilitation (Johnson et al., 2004) and to simplify rehabilitation particularly for patients living in rural areas (Patel et al., 2012).
Of course, simple home-based exercises have always been prescribed by therapists. Nevertheless, studies have found that a large percentage of patients do not exercise as recommended. Shaughnessy et al. (2006) reported that only 31% of patients suffering from stroke exercised as recommended. Ellis et al. (2011) investigated exercise behaviour of Parkinson's disease patients and found

that about one third of patients don't exercise at all. Researchers attribute this lack of motivation to the monotone nature of many rehabilitation exercises (e.g. Alankus et al., 2010 or Flores et al., 2008). Hence, game-based approaches are often recommended to raise motivation and engagement (Flores et al., 2008; Scherer et al., 2013). Furthermore, studies showed that game-based or virtual-reality-based therapy can be more effective than standard rehabilitation (Corbetta et al., 2015).
Apart from the challenge of raising motivation, precise control (and adjustment) of movements is challenging in a non-clinical environment with no therapist present. There are robotic-based systems that enforce and measure correct limb positioning. However, those systems are complex and expensive and might endanger the patients' safety in case of system failures (Steinisch et al., 2013).

In recent years, affordable consumer devices emerged that have the potential to facilitate out-patient rehabilitation. Particularly the Nintendo Wii game console and the Microsoft Kinect motion tracking sensor attracted the attention of researchers (e.g. Galna et al., 2014, Pompeu et al., 2012, Mhatre et al., 2013, Lange et al., 2011). While both systems benefit game-based rehabilitation applications particularly the Kinect sensor proves capable to measure clinically relevant movements (Galna et al., 2014a). Scherer et al. (2013) proposed a framework that combines the Kinect sensor with an EEG recording device. The authors argue that such a system could provide important insights on the effect of motor rehabilitation on the human brain (neuroplasticity) as well as on the patients' attention and motivation during exercises.
The number of affordable hardware devices continued to grow in recent years. While most of them are being marketed as game-controllers for consumers, they have great potential to facilitate mobile rehabilitation. See chapter 4 for a further discussion of some potentially useful devices.

2.3.1. Requirements for Mobile Rehabilitation

For hardware to be of use for mobile, home-based rehabilitation it, firstly, must be accurate enough to measure timing of clinically relevant movements (Galna et al., 2014a). Ideally, depending on the desired application, it should do so in such a way so that small changes in the patient's performance are noticeable (Tung et al., 2015). Secondly, they must be easy to use for inexperienced and/or cognitively or physically impaired users (Barry et al., 2014) and should be small and portable enough to be used on the patients' bedside (Scherer et al., 2013). Lastly, as for mobile BCI the hardware devices should be easily affordable for

patients that already have to bear high medical bills.
Regarding software requirements a mobile rehabilitation platform should be able to handle multiple different hardware devices both simultaneously as well as interchangeably. Firstly, *simultaneity* means that multiple devices can be used at the same time. This is similar to MoBI even though millisecond-precise synchronization is of lesser importance in this scenario. The reason for this requirement is that many of the affordable consumer devices serve largely different purposes. While for instance Microsoft's Kinect accurately tracks posture and gross body movements but struggles to measure smaller movements such as finger tapping (Galna et al., 2014a) or limb rotation (Rahman, 2015) such movements can accurately be tracked using the Leap Motion controller (Leap Motion Inc., San Francisco, CA, USA) (Tung et al., 2015) or the Myo wristband (Thalmic Labs Inc., Ontario, Canada), respectively, which will be presented in chapter 4. Therefore, a combination of multiple consumer devices might prove to be most effective. Such multi-sensor applications have been proposed by Rahman (2015) and Caraiman et al. (2015). While Rahman combined the Kinect and the Leap Motion sensor proposed a rehabilitation and monitoring system using the Kinect and Leap Motion as well as an EEG system and an eye tracker.
Secondly, *interchangeability* means that controller input (e.g. a button press, hand gesture) and in-game output (e.g. movement of a character) should be loosely coupled. Thereby the platform can be utilized for a wide-range of therapeutic exercises and patient populations.
Apart from the underlying architecture the application should be designed in a way that motivates and engages but not overwhelms the patient (Scherer et al., 2013). Such design principles have been discussed by Burke et al. (2010), Shah et al. (2014), Borghese et al. (2013) and many others.

2.4. Summary

Even though the meaning of *mobility* is different for mobile brain imaging, mobile brain-computer interfacing and mobile rehabilitation there are many aspects they have in common. For all those areas the transition into new, less controlled environments comes with many challenges. In terms of hardware new devices will have to be introduced. For MoBI new high-accuracy mobile amplifiers are of interest. Mobile BCIs require robust and affordable hardware solutions. Lastly, mobile rehabilitation can make use of a wide range of consumer hardware that was not available a couple of years ago.
In terms of software this means that a wide range of different hardware devices

must be supported. For many applications in those three areas multiple hardware devices will be used simultaneously. Particularly MoBI systems require millisecond-precise synchronization. For consumer oriented applications software systems must be easy to use, platform independent and as inexpensive as possible.
To achieve this goal the software platform presented in this thesis will build on an existing platform called the Lab Streaming Layer. This platform is described in more detail in the following chapter.

3. A Sensor Platform for Mobile Brain Imaging, BCI and Rehabilitation

To achieve the long-term goal of having a software system that facilitates the areas presented in the previous chapter firstly a sensor platform to access a wide range of hardware devices must be created. Such a sensor platform should be able to acquire and synchronize data from multiple hardware devices and should make this data available for further processing. Additionally, it should be extremely modular to be applicable for the diverse range of applications that arises from the areas mentioned above. Moreover, to be usable by patients it should be affordable and hence should not require any proprietary software.

A software approach that can fulfil those requirements is the Lab Streaming Layer (LSL). The LSL is a software platform developed by the University of San Diego. It is described as *"a system for the unified collection of measurement time series in research experiments that handles both the networking, time-synchronization, (near-) real-time access as well as optionally the centralized collection, viewing and disk recording of the data"* (LSL, 2015a).

3.1. Lab Streaming Layer (LSL)

LSL is a system to collect and synchronize data from potentially many different sources. To do so LSL provides a core transport library, called liblsl, and a series of tools or applications. These tools are responsible either for making the data of different hardware devices (like EEG amplifiers) available or to retrieve and process the provided data (like a recording program). The former will from now on be referred to as *hardware applications* or *hardware apps* while the latter will be called *client applications* or *client apps*.

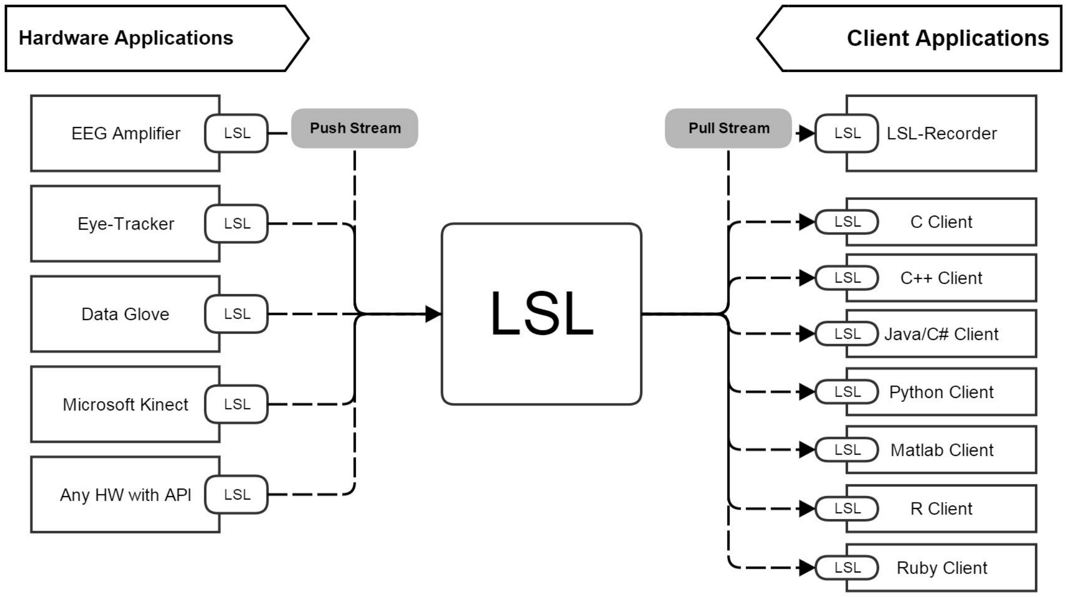


Figure 3.1.: LSL Scheme: Hardware applications make device data available to LSL using the push stream call. Client applications may access this data using the pull stream call.

The basic idea of LSL is that an arbitrary number of hardware devices stream (or push) their data to the LSL transport library using the dedicated LSL-applications while an again arbitrary number of client programs retrieve (or pull) this stream.

LSL currently supports a wide range of hardware including multiple EEG amplifiers, eye trackers, motion capture hardware as well as input hardware such as computer keyboards, computer mice and joysticks LSL (2015b). But the range of application can easily be extended as LSL provides language interfaces written in C, C++, C#, Java, MATLAB and Python. A schematic overview how hardware applications, client applications and the LSL transport layer interact can be seen in figure 3.1.

LSL's synchronization works using an arbitrary number of applications that run on different computers and with data streams with different and even varying sampling rates. This architecture allows for an incredibly flexible and modular design and experiment setup. Hardware applications can be combined in an arbitrary fashion without having to rewrite or recompile code. The following section 3.1.1 will describe the synchronization feature in some more detail. Section 3.1.2 discusses the basic structure of LSL applications.

3.1.1. Time Synchronization

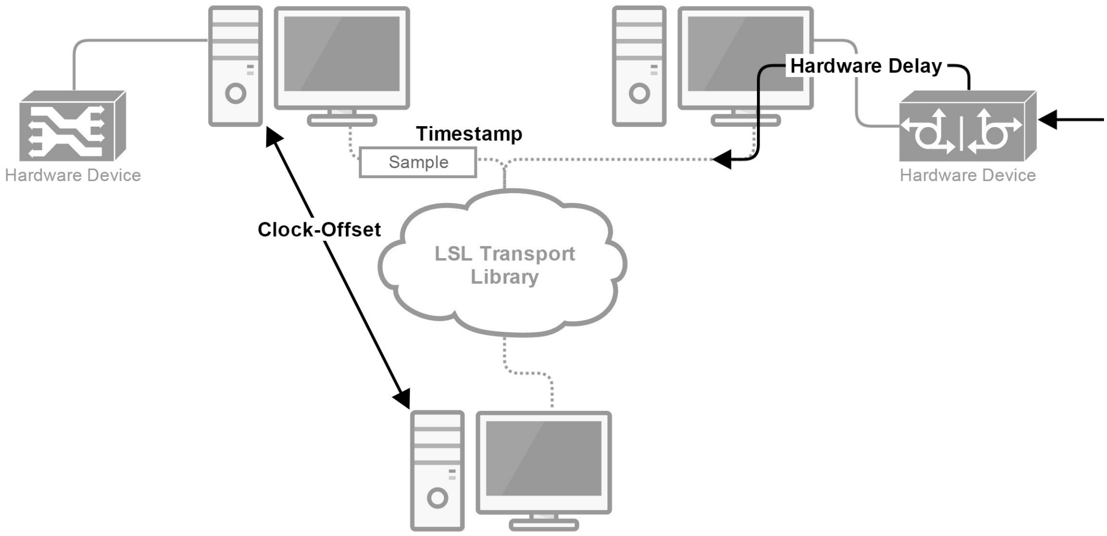


Figure 3.2.: LSL Synchronization: The three key values LSL uses to synchronize streams. The sample's timestamp, clock-offset between machines and hardware delay.

Synchronization is the key requirement to facilitate MoBI as well as mobile and hybrid BCI applications. LSL can achieve millisecond precision or better (Ojeda et al., 2014; Grivich, 2015). To do so it relies on three values: timestamps, clock-offset measures and a hardware-delay measure.

1. **Timestamps:** A timestamp is collected for every block of data (usually one or multiple samples) an application pushes to the transport library. It is either provided by the user or implicitly set by the LSL library.

2. **Clock-Offset Measures:** These values indicate the offset between the sender's and the receiver's internal clock. For each data stream clock-offset measures are computed periodically and transmitted to the receiving computer.

3. **Hardware-Delay Measure:** The hardware delay is the time between sample capture and transmission to network. This value is usually constant for each device and can be made available to LSL using a delay information file (Ojeda et al., 2014).

To synchronize streams one has to make use of those three values. The easiest approach is to add the most recent clock-offset measure to and subtract the hardware-delay measure from each timestamp. A more advanced approach would be to compute a linear fit through all available clock-offset measures and add those values to the timestamps (LSL, 2015c).

In preparation for this thesis, LSL's synchronization abilities were validated and confirmed. Please refer to section 6.1 for detailed results.

3.1.2. LSL Applications

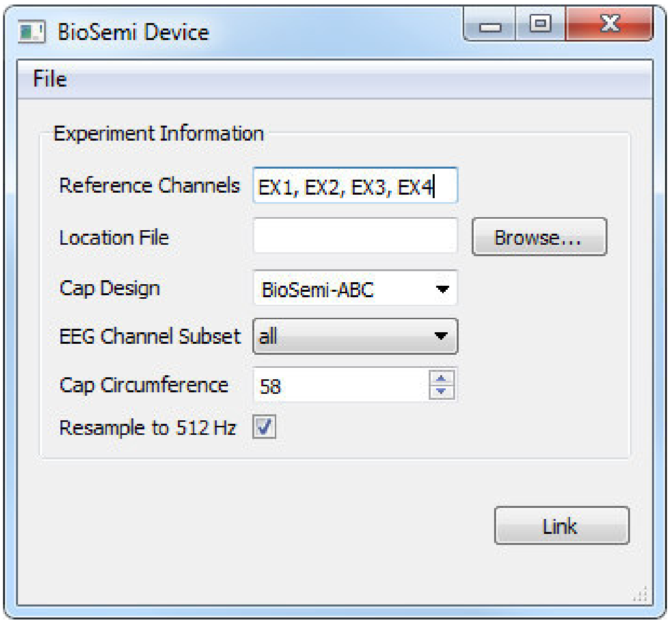


Figure 3.3.: A typical LSL hardware application. The application provides means to modify devices settings, a file menu to load and store configuration files as well as a button to start streaming the device's data to the LSL transport library (Link Button).

As explained in the introductory part and figure 3.1 LSL is application based. This means that for every instance of every hardware device there is one application running. While applications can be written in many different programming languages existing hardware applications are typically written in C++ using the QT framework.
Usually LSL applications consist of three main parts.

1. **Device Settings:** This collection of User Interface (UI) elements allows the user to modify basic device settings (e.g. the device's sampling rate) and

specify LSL configuration parameters (e.g. channel names).

2. **Link Button:** This UI element applies the settings specified by the user to the hardware device and starts streaming the device's data to the LSL transport library. Now other applications can retrieve this data stream.

3. **File Menu:** The file menu allows the user to save and load a configuration file for the application in use. A configuration file stores all parameters specified in the Device Settings section. There is also a default configuration file which will be applied at start-up.

3.2. LSL for Mobile Brain Imaging, BCI and Rehabilitation

The features of LSL were described above. This section will describe how LSL can facilitate MoBI, mobile and hybrid BCIs and mobile rehabilitation.

LSL for Mobile Brain Imaging

As mentioned in the previous chapter one of the biggest data acquisition related challenge of MoBI is the synchronization of EEG, motion-capture and maybe even audiovisual scene recording (Reis et al., 2014). LSL's ability to achieve millisecond-accurate synchronization of multiple data sources makes it the ideal solution for MoBI scenarios. Its predecessor was recommended by Makeig et al. when proposing MoBI. In subsequent publications the same research group recommends LSL for data acquisition in MoBI experiments (Makeig et al., 2009; Gramann et al., 2014; Ojeda et al., 2014).
It is important to note that LSL is not in itself a software solution for mobile brain imaging but only one building block that handles hardware access, data acquisition and synchronization (or synchronization preparation). To present stimuli to the user and to analyse the recorded data other software solutions have to be used. Gramann et al. (2014) for example proposed the software platforms SNAP (for stimulus presentation) and MoBILAB (for data analysis). But the data recorded using LSL can also easily be integrated into the existing code-base of research laboratories.

LSL for Mobile BCI

Translating brain activity into computer commands as done by a BCI system usually involves a set of processing stages. Pfurtscheller et al. (1993) identified signal processing and classification as the two main stages of a BCI. Based on this definition Wolpaw and Winter-Wolpaw (2012) and Rao (2013) describe the four stages of a BCI system as follows:

1. **Signal acquisition:** Brain activity is recorded
2. **Signal processing:** The recorded signal is prepared and analysed
3. **Feature translation:** The signal is translated into computer commands to operate applications
4. **Sensory feedback:** The changes in the environment caused by the BCI are fed back to the brain via stimulation

This sequence is commonly known as the BCI loop. While software systems exist that cover all aspects of this loop (see Brunner et al. (2012) for a review) LSL only manages signal acquisition and, if necessary, synchronization with other hardware devices. The advantage of an LSL based BCI system is that existing source code for signal processing and feature translation can easily be reused.
With the signal acquisition part of the BCI loop being loosely coupled with the rest of the system it becomes easily expendable with new applications. Also multiple hardware devices can be combined without requiring recompilation of source code as is currently necessary for some BCI systems. Rozado et al. (2015) used an LSL based BCI system. In their system they synchronized EEG with eye-tracking data which nicely demonstrated the abilities of LSL for BCI studies.
Similarly, the synchronization abilities of LSL are of great use for hybrid BCI applications where multiple input devices (one of which being a BCI) are used simultaneously.

But apart from the technical requirements for a BCI to really be usable for patients at home affordability was a key requirement. LSL being open-source, freely available online and not restricted to a particular, potentially expensive, programming environment fulfils this requirement. Also it runs on all major operating systems.

LSL for Mobile Rehabilitation

Section 2.3 introduced the concept of mobile computer-based rehabilitation. It was stated that using consumer devices mobile rehabilitation has the potential to be an affordable, motivating and effective alternative to in-patient therapy. The two key requirements for a software platform for mobile rehabilitation were to handle multiple hardware devices both simultaneously and interchangeably. Both of which can be achieved using LSL.
Utilize multiple hardware devices simultaneously can easily and very effectively be implemented using LSL as the area of mobile brain imaging, where LSL is already used, demonstrates. Even though millisecond-accurate synchronization is of lesser importance for mobile rehabilitation the underlying architecture of sending and receiving an arbitrary number of different data streams is equally usable.
Secondly, due to its modular design also the requirement of interchangeability can be achieved using LSL. Different hardware devices can be streamed independently of each other requiring no source recompilation or project reconfiguration. Naturally, the receiving application has to support this modularity as well but with the multi-language interfaces provided by LSL this can be easily achieved.
Lastly, LSL is easily expendable which makes it easy to integrate new devices that will emerge in the years ahead.

So it can be concluded that LSL has great potential to be used as a common sensor platform for the areas of mobile brain imaging, mobile BCIs and mobile rehabilitation. Other than many of the existing software solutions that were considered (e.g. platforms discussed by Brunner et al. (2012)) LSL is flexible enough to be applicable for all those areas. But while LSL already provides applications for a wide range of hardware devices there are some devices which are of great interest for those scenarios which are not implemented yet. Those devices will be presented in the following chapter.

4. New Hardware for Mobile Brain Imaging, BCI and Rehabilitation

This chapter presents a selection of promising hardware devices which were not supported by LSL, as introduced in the previous chapter, and therefore have been added to the platform as part of this thesis.
Both medical-grade and consumer devices were considered to fit the needs of researchers, who require hardware to record high quality data in a mobile environment, and patients, who benefit from affordable hardware devices. *Medical-grade hardware* refers to hardware which is intended to be used in a clinical or research environment. *Consumer hardware*, however, refers to electronic devices which are commercially available and intended for personal use (Oxford Dictionaries, 2015). The intended application, such as entertainment or computer interaction, will often differ from the application proposed for this thesis. Typical consumer hardware devices which are of interest are game controllers or game consoles.
The main requirement imposed on both medical-grade and consumer hardware was to facilitate mobile data recording. Additionally, a medical applicability of the consumer devices was assessed.
The structure of this chapter is as follows. Section 4.1 will introduce medical-grade hardware devices. Namely the ANT eegosports in section 4.1.1 and two data gloves in section 4.1.2. Consumer-grade hardware devices will be discussed in section 4.2. Devices presented in this chapter include the Emotiv EPOC (4.2.1), the Leap Motion controller (4.2.2) and the Thalmic Myo wristband (4.2.3). See table 4.1 for an overview of selected hardware devices.

Selected Hardware			
Name	**Type**	**Category**	**Price**
ANT eegosports	EEG Amplifier	Medical	~USD 50.000
CyberGlove	Data Glove	Medical	USD 10.000/15.000
5DT DataGlove	Data Glove	Medical	USD 1.000/5.500
Emotiv EPOC	EEG Headset	Consumer	USD 400/500
Leap Motion	Controller	Consumer	USD 70
Thalmic Myo	Controller	Consumer	USD 200

Table 4.1.: A list of selected hardware devices.

4.1. Medical-Grade Hardware

4.1.1. ANT eegosports

As mentioned above the central idea of mobile brain imaging is to investigate the state of the human brain while the subject actively interacts with its environment. To facilitate this idea there's a need for mobile EEG recording devices. While traditionally EEG recording required the use of multiple, bulky and hardly mobile EEG amplifiers new mobile solutions have emerged in recent years. Those devices try to be as lightweight as possible (ideally light enough to be attached to the subject's body without restricting his/her mobility) and to reduce the number of wired connections to a minimum. Apart from the ANT eegosports (ANT Neuro, Enschede, Netherlands) described below the MOVE system (Brain Vision LLC, Morrisville, NC, USA) which transmits data from the electrodes to the amplifier wirelessly, the Cognionics wireless EEG (Cognionics Inc., San Diego, CA, USA) and the g.Tec g.Nautilus wireless acquisition system (g.Tec Medical Engineering GmBH, Graz, Austria) are mobile EEG systems worth noting. (Reis et al., 2014)

The ANT eego sports is a relatively new and compact amplifier that targets motion and movement experiments. The eego sports system uses a compact, battery-powered amplifier and a Microsoft Surface tablet (Microsoft Inc., Redmond, WA, USA). Those two devices are stored in a small backpack worn by the user. The tablet temporarily stores the data coming from the amplifier and forwards it to a remote computer where the data is stored permanently.

Figure 4.1.: The ANT eegosports amplifier as distributed including a tablet, backpack and cap. (ANT Neuro, 2015)

This solution significantly reduces the risk of data loss (Reis et al., 2014). With a combined weight of below 2 kilograms (amplifier: 500 grams, tablet: 600 - 1200 grams depending on configuration) and wireless data transmission to the remote computer the ANT eego sports is a truly mobile EEG system.

The amplifier comes in two configurations with either 32 or 64 channels and has a maximum sampling rate of 2048 Hz (ANT Neuro, 2015). Amplifiers like the ANT eego sports can significantly simplify mobile brain imaging experiments. Using commonly used bulky EEG amplifiers proposed MoBI systems relied on crane-like structures (e.g. Gramann et al., 2011) or moveable carts (e.g. Ehinger et al., 2014) to hold recording equipment. As the ANT eego sports is wireless in the sense that no wires lead from the participant to a stationary computer (only to the device inside the participants backpack) usability is greatly improved. Hence, the amplifier is likely to be utilized in upcoming movement experiments and was included in the software framework discussed in this thesis. More details regarding the ANT eego sports implementation will be given in section 5.2.1.

4.1.2. Data Gloves

Data gloves are devices worn like a glove which are equipped with a set of sensors to capture hand and finger movements as well as hand gestures (Pam-

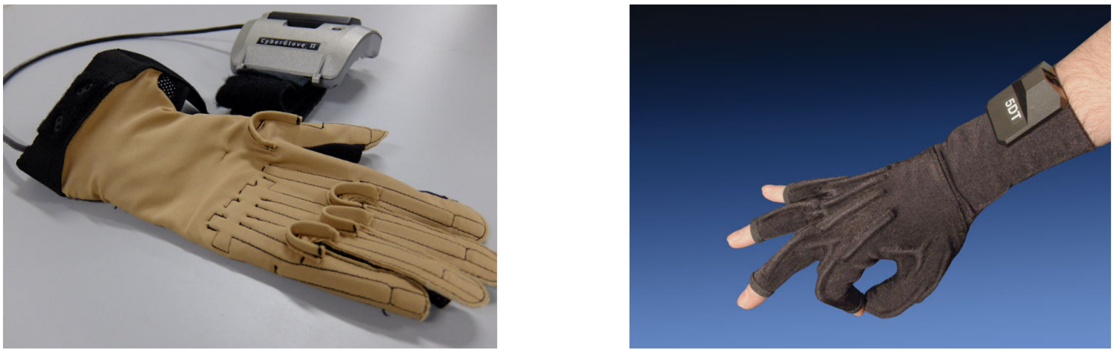


Figure 4.2.: The CyberGlove (left) and the 5DT Data Glove (CyberGlove Systems Inc, 2015; 5DT Inc, 2015)

plona et al., 2008). The devices play an important role in computer assisted rehabilitation (e.g. Jack et al., 2000 or Yamaura et al., 2009) as well as in motor cortical research (e.g. Scherer et al., 2009 or Blakely, 2013).
Hence, it was decided to add support for two different data glove systems. Firstly, an application for the medical-grade CyberGlove (CyberGlove Systems, San Jose, CA) was developed. Secondly, the cheaper DataGlove (5DT, Gauteng, South Africa) which is targeted for motion capture and animation was integrated.

The CyberGlove is a data glove with either 18 or 22 sensors. It measures flexure and abduction of each finger and wrist as well as palm arch. While the 18 sensor version has two flexure sensors per finger the 22 sensor version has three of such bend sensors. (CyberGlove Systems Inc, 2015)
The DataGlove is available with either 5 or 14 finger sensors. It measures flexure of each finger and in the case of the 14 sensor version also abduction between fingers. The glove also measures orientation of the user's hand.

Both devices are widely used in both research and rehabilitation hence no detailed assessment of their abilities is required for the aim of this thesis. For a detailed description of the data glove's integration into the system please refer to section 5.2.2.

Figure 4.3.: The Emotiv EPOC wireless EEG headset (Emotiv, 2015)

4.2. Consumer Hardware

One of the key requirements of a mobile rehabilitation or hybrid BCI system that was described in sections 2.2.4 and 2.3.1 is to be affordable for patients. One of the first commercially available devices that have been used for computer-based rehabilitation is Microsoft's Kinect motion tracking sensor. While it has originally been developed to control computer games through body movement numerous publications have shown the benefits of Kinect-based rehabilitation and physical therapy systems investigating both the Kinect's accuracy (e.g. Obdrzalek et al., 2012, Galna et al., 2014) as well as its impact on rehabilitation (e.g. Hondori and Khademi, 2014) and EEG research (e.g. Scherer et al., 2012).

With computers and sensors getting smaller, faster and cheaper a wide range of new compact and low-cost hardware devices emerged over the course of the last couple of years. Hence, it is desirable to create a software platform that can make use of low-cost consumer hardware as well.
Three of such affordable consumer devices will be introduced in this section.

4.2.1. EMOTIV EPOC+

The major cost factor of EEG based systems are EEG recording devices. Systems that cost USD 40.000, such as the ANT eegosports presented above, and more pose a significant financial challenge. Particularly for patients that would profit the most from an EEG system, such as patients with severe motor and/or neurological impairments who already have to bear high medical costs.

Fortunately, more and more commercial EEG devices have become available in recent years. Among others the Emotiv EPOC (Emotiv Inc., Hong Kong), NeuroSky MindWave (Neurosky Inc., San Jose, CA, USA) or the Enobio (Neurolectrics, Barcelona, Spain) have been released. (Badcock et al., 2013; Kranczioch et al., 2014). These devices are typically marketed as EEG systems for gaming or personal health and well-being. According to Badcock et al. (2013) these systems are usually characterized by a common set of properties. Some of those characteristics stand in clear contrast to medical-grade EEG devices. Firstly, the number of electrodes used is typically small. Ranging from one single electrode (e.g. NeuroSky MindWave) to 14 (e.g. Emotiv EPOC) or 20 (e.g. Enobio). Secondly, consumer-based EEG systems put emphasis on the ease of use. Hence, they are typically wireless. Also, they usually require hardly any adjustment of electrodes. Lastly, these systems often use cotton pads and saline solution instead of the sticky conductive gel used in medical-grade EEG systems. (Badcock et al., 2013)

While there is an increasingly wide range of commercially available EEG systems the Emotiv EPOC caused the most attention in the scientific community. The standard Emotiv EPOC provides 14 sensors plus 2 reference channels as well as a 2-axis gyroscope and has a sampling rate of 128 Hz. The EPOC+ device has a sampling rate of 256 Hz and a 9-axis Inertial Measurement Unit (IMU) that provides gyroscope, accelerometer and magnetometer data. Figure 4.4 shows the position of the 14 channels on the user's head.
At time of writing the standard version of the Emotiv EPOC costs USD 399 the EPOC+ USD 499. (Emotiv, 2015)

EPOC Assessment

For an EEG system to really be of benefit for disabled or locked-in patients it must be accurate enough to control a BCI. To recap, a BCI is a system that utilizes brain activity, recorded using for instance EEG, to control a computer. For a signal to be strong enough to be picked up by EEG a large group of brain cells must respond to a particular stimulus. This can be achieved using some well-known paradigms like Event-Related Potential (ERP) or Steady-State Visually Evoked Potential (SSVEP). In case of ERP the user is presented with a relevant and rare stimulus. This stimulus causes increased brain activity with its peak 300ms after the stimulus occurred (Rao, 2013). The idea of SSVEP is that when the participant focuses on a stimulus at a specific frequency increased

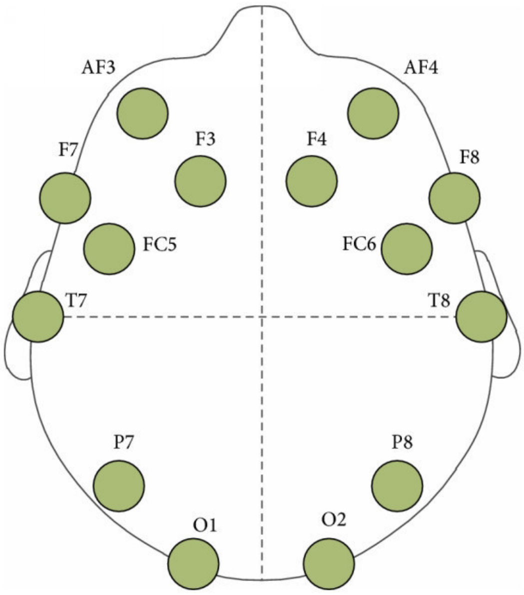


Figure 4.4.: The positions of the 14 channels provided by the Emotiv EPOC. (Choi et al., 2014)

brain activity at the same frequency can be noticed (Kübler and Müller, 2007).

In recent years the EPOC's abilities to pick up either ERPs or SSVEPs have been assessed. Duvinage et al. (2013) assessed the EPOC's abilities to pick up P300 visual ERPs while the subject was either sitting or walking on a treadmill. Badcock et al. (2013) validated the EPOC system for measuring auditory ERPs. Most importantly, both studies concluded that the EPOC does in fact record EEG and does not only pick up muscular or ocular artefacts as it has been criticized in the past. Both studies also showed that well-studied changes in the EEG such as an ERP can be measured using the Emotiv EPOC (Duvinage et al., 2013). Nevertheless, both studies found that the EPOC does not perform as well as a medical-grade EEG system. According to Duvinage et al. (2013), who compared the EPOC system to a medical-grade system (ANT DC amplifier and 128 channel WaveGuard cap, ANT Neuro, Enschede, Netherlands), the signal-to-noise ratio is lower for the EPOC system. Hence, the EPOC headset misclassifies more data than a medical-grade system does. They also observed that the misclassification rate was significantly higher for the walking scenario. Similarly, Badcock et al. (2013) state the EPOC compares well with a medical-grade system for standard auditory ERPs but point out that it struggles detecting more elaborate signal components.

The EPOC's potential to detect SSVEP was investigated by Pröll (2012) and Liu et al. (2012). Pröll successfully detected SSVEPs using a 17.14 and a 20 Hz stimulus while Liu et al. demonstrated similar results for a stimulus of 13 Hz. Furthermore, they compared the EPOC to a medical-grade system (g.Usb Amplifier, g.Tec Medical Engineering GmBH, Graz, Austria). Their results showed that while the EPOC's accuracy is significantly higher than chance (83%) it underperforms the medical-grade system by about 10%. Also, it is worth noting that the accuracies varied substantially between subjects. The lowest accuracy achieved was at 65% compared to 85% achieved for the same subject by the medical-grade system. The authors attributed that to the strong alpha wave of the subject in question which was stronger than the 13Hz SSVEP.
Similar results (i.e. highly varying accuracies with a mean of 61% and a high deviation of 25%) were reported by Nijboer et al. (2015). More interestingly the authors investigated the user-friendliness of the EPOC system. They found that users usually rated the speed of set-up and appearance of the system as being very high but found it significantly more uncomfortable to wear than comparable systems. Similar results were published by Pröll (2012). The author asked the subjects to rate the wearing comfort of the EPOC headset on a scale from 0 (very uncomfortable) to 10 (very comfortable) at various points throughout a two hour long recording session. While the subjects initially rated the wearing comfort relatively high (mean: 7.45, standard deviation: 2) those ratings dropped significantly resulting in a final rating of 2.73 (standard deviation: 2.31). The reason for this might be that due to the design EPOC sensors press hard against the user's head (Nijboer et al., 2015).
Another important point regarding the EPOC's sensors was made by Duvinage et al. (2013). They pointed out that the default arrangement of electrodes does not aid motor imagery paradigms as there are no electrodes placed over the motor cortex. In this regard, the work of Debener et al. (2012) is worth noting. In their publication the authors showed how to combine the EPOC headset with a state-of-the-art electrode cap, which yield a portable, wireless system (the advantages of the EPOC) with improved signal quality (the advantages of the medical-grade wet electrode cap) (Debener et al., 2012). Additionally, with regard to the findings of Nijboer et al. (2015) mentioned above, the system becomes more comfortable to wear when using a cap instead of the rigid sensor frame. As all parts are commercially available, this system can easily be adopted as shown for instance by Stopczynski et al. (2014).

To summarize, the discussed studies showed that the Emotiv EPOC is able to record EEG as well as to detect common patterns. However, for critical

Figure 4.5.: The Leap motion controller and the hand tracking model. (Leap Motion, 2015)

applications and brain research the standard EPOC might not be suitable. Nevertheless, for game-based hBCIs, for rehabilitation as well as for measuring motivation it might be used. Taking into account the work by Debener et al. (2012) the range of possible applications increases as the authors demonstrated the flexibility and expandability of the device. When used with a custom electrode-cap the acquired signals improve significantly and make the system usable for mobile BCI and even MoBI applications as described in the above-mentioned publication.
Hence it was decided that the Emotiv EPOC is the most promising consumer-grade EEG system and should be included as a low-cost alternative for non-critical, and with modifications, even critical applications.

4.2.2. Leap Motion

The Leap Motion controller is a motion tracking device specifically designed for tracking fingers, hands and finger-shaped tools. It provides data about finger position and movement and hand velocity and rotation as well as basic gesture recognition. As shown by studies discussed in this section the Leap Motion can be a viable finger-tracking alternative to the significantly more expensive data gloves presented in 4.1.2.

The sensor uses three infrared LEDs and two infrared cameras to track the users hands. This yields a hemispherical field of view with the device being in the centre. The sensors are able to track hands positioned between 25 and 600 millimetres above the device which is typically placed on a flat surface in front of a computer (Hondori and Khademi, 2014).

Leap Motion's Accuracy

The Leap Motion's accuracy has been investigated by Weichert et al. (2013) and Guna et al. (2014). Both tests consisted of two scenarios. A static scenario to test the controller's accuracy and repeatability and a dynamic scenario to determine its ability to track a moving object (Weichert et al., 2013). Repeatability describes the ability of the system to compute the same location for a stationary object over a series of measurements (Guna et al., 2014). As a reference system Weichert et al. used an industrial robotic arm while Guna et al. used a high-precision motion capture system.

For the static scenario Guna et al. determined a minimal standard deviation of 0.0081 millimetres for hands positioned directly above the controller and a maximal standard deviation of 0.49 millimetres for hands positioned at the leftmost and topmost position of the controller's field of view. Weichert et al. computed an average standard deviation of below 0.05 millimetres as well as a repeatability of 0.2 millimetres. While both studies provide similar results for the static scenario the results for the dynamic scenario differ greatly. Weichert et al. reported accuracies of 1.2 millimetres on average with a maximal deviation of 2.5 millimetres. Guna et al. recognized a significant drop in accuracy for samples taken more than 250 millimetres above the controller regardless of where on the (imaginary) hemisphere the hand is located. Tests closer to the controller yield accuracies at around 2 millimetres and are similar to those reported by Weichert et al. But outside that 250 millimetres frame the accuracies drop significantly resulting in a mean deviation of 6 millimetres and above.

Another study by Tung et al. investigated the reliability and accuracy of the Leap Motion controller with regard to clinically relevant neuromotor assessments such as reaching and pointing (Tung et al., 2015). Other than the studies by Weichert et al. and Guna et al. the authors evaluated the controller using human test subjects instead of robotic actuators. Their participants were instructed to move their finger from an initial starting position to one of 15 targets presented on a computer screen. This setup resembles the Trail Making Test. A test commonly used for neuropsychological evaluation (Arnett and Labovitz, 1995) for instance following acute stroke (Tamez et al., 2011).
The measured accuracies by Tung et al. with a root mean square error of 17

millimetres were significantly worse than those described by Weichert et al. and Guna et al. The authors attribute this large difference to the recording of actual human arm movement instead of the idealistic robotic movements recorded by the before mentioned authors.

Particularly the study conducted by Tung et al. and Guna et al. revealed the limitations of the Leap Motion controller. Nevertheless, it can be concluded that for well positioned hands the accuracy of the sensor is promising by achieving significantly better results tracking hand movements than the well established Microsoft Kinect sensor (Weichert et al., 2013). As the latter struggles to measure fine movements such as hand clasping correctly (Galna et al., 2014a) and has a significantly higher latency than the Leap Motion Controller (Brown et al., 2014).

Leap Motion for Research and Rehabilitation

As mentioned above Tung et al. assessed the usefulness of Leap Motion controller specifically with regard to clinical and therapeutic applications. Apart from the issues discussed in the section above the study identified two major limitations. Firstly, they noticed a drop in the achieved accuracies near the ranges of the controller's field of view as described by Guna et al. Secondly, the study also found that the sensor is less reliable when the tracked hand touches a monitor and thereby is not clearly separable from its surroundings.

With regard to the relatively high inaccuracy the authors concluded that the Leap Motion controller is unsuitable for clinically-relevant measures like rapid pointing tasks. The reason for that is that the measured inaccuracies are higher than the mean difference between impaired and healthy subjects in such assessments.
Nevertheless, the authors argued that the Leap Motion controller is sufficiently accurate for tasks where no high positional accuracy is required. Among such are some motor assessment and clinical rehabilitation tasks as well as portable solutions for patients at home (Tung et al., 2015).

The applicability of the Leap Motion controller for motor rehabilitation tasks was further investigated by Godlove et al. The authors compared the Leap Motion controller to the 5DT Data Glove with 14 sensors. The objective was to assess the systems regarding their capabilities as a computer- and game-based rehabilitation tool. The study concluded that the Leap Motion controller was more accurate when measuring joint angles of healthy subjects than the Data Glove. However, the sensor struggles to measure joint angles for (partially) closed hands correctly which makes the controller difficult to use for hand-impaired patients (Godlove et al., 2014). However, Iosa et al. successfully used the Leap Motion controller for game-based neurorehabilitation of patients with subacute stroke. The authors did not report the patients having any issues using the controller. Furthermore, they found high motivation and participation levels for all subjects.

Conclusion

The presented research suggests that the Leap Motion controller is useful for both research and rehabilitation. As stated by Tung et al. (2015) the controller is accurate enough for some motor assessment and for rehabilitation tasks. Additionally, the work of Godlove et al. (2014) that compared the Leap Motion to a medical-grade data glove showed that the former is more accurate than the latter. Hence, the Leap Motion appears to be an interesting device for many different applications and was included in the software platform.

4.2.3. Thalmic Myo

Surface Electromyography (EMG) - the EMG variant where muscle activity is recorded from the surface (i.e. skin) above the muscle of interest - is of great interest for medical diagnosis (e.g. Zwarts et al., 2000), research (e.g. Yao et al., 2007) and rehabilitation (e.g. Mulas et al., 2005).

The Thalmic Myo armband is a wearable gesture and motion recognition device. To do so it uses eight medical grade EMG sensors as well as a 9-axis IMU which provides accelerometer, gyroscope and magnetometer data (Thalmic Myo, 2016). As the name implies the Myo is worn like an armband around either the lower or upper arm. It is designed to record activity generated by the wearer's fingers, palm and forearm (Kutafina et al., 2015).
While the Myo is marketed as an affordable consumer gadget to control applications and devices it also provides access to the raw EMG and IMU data. Moreover, the controller's data is transmitted wirelessly using a blue-tooth connection. Those two features make the Myo armband an interesting device for mobile brain imaging and rehabilitation scenarios.
In this section firstly the available data will be described. Secondly, an overview over existing literature on the quality of the recorded data is provided. Lastly,

Figure 4.6.: The Myo wearable gesture and motion detection armband. (Thalmic Myo, 2016)

existing and potential applications for the Myo controller in a research and rehabilitation environment are discussed.

Available Data

The Myo controller provides three kinds of data.

1. *Spatial Data:* The data measured by the IMU consists of orientation and acceleration data as well as the angular velocity of the wearer's arm. The orientation data is provided as a Quaternion, acceleration data in units of g and the angular data as degrees per second. The spatial data is made available at a sample rate of 50Hz.
2. *Gestural Data:* As shown in figure 4.7 the Myo armband automatically detects five predefined gestures. By combining those gestures with data from the IMU additional gestures can be distinguished.
3. *EMG Data:* The Myo also provides the raw EMG data of each of the eight EMG sensors. The (unitless) EMG data is provided with a sample rate of 200Hz as integers in the range of -128 to 128.

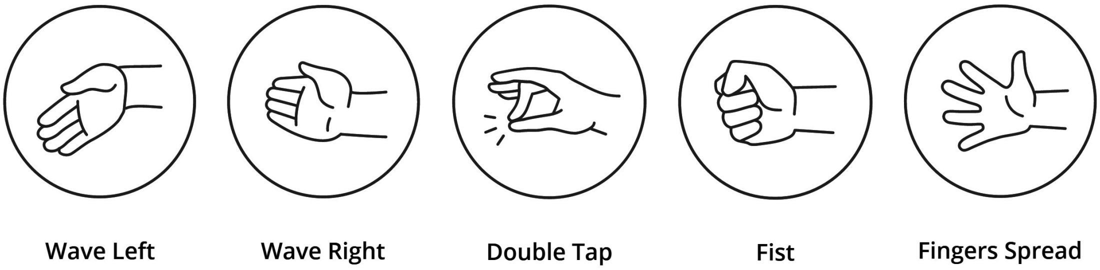


Figure 4.7.: Myo Gestures: The 5 preset gestures recognized by the Myo controller. (Thalmic Myo, 2016)

Myo Data Assessment

As the Myo armband is a fairly new device studies discussing the controller's accuracy are rare. Nevertheless, this section will collect and compare available data on the performance of the controller.
One feature that captured the interest of the scientific community is the Myo's ability to detect gestures. Gesture detection based on EMG data is a well-studied field in EMG research. Potential applications include graphical controllers for physically disabled patients, robotic device and arm control, control of a prosthetic limb as well as game control and exercise support (Ahsan et al., 2009). However, no publication scientifically investigated the gesture recognition abilities of the Myo controller. Galster and Abduo (2015) note that the preset gestures are usually recognized well even though some misclassifications between the fist and finger spread gesture occurred. Similar results were published by Hettig et al. (2015). They reported relatively high recognition rates (between 86 and 71%) for the fist, spread fingers, wave in and wave out gestures. However, recognition for the double tap gesture was fairly low with only 56%.
Regarding additional gestures computed from the raw data the literature is inconclusive. Galster and Abduo (2015) concluded that the Myo controller is not accurate enough to control a prosthetic hand with many degrees of freedom but did not provide data to back this claim. However, Tenore et al. (2009) found that the highest accuracies for gesture recognition can be achieved using 19 sensors. Hence, one might conclude that Myo's 8 sensors are not enough for high-accuracy detections. However, it has to be noted that - other than standard EMG sensors investigated by Tenore et al. - the Myo controller provides more data than just raw EEG data: Combining raw EEG data with Myo's IMU data yields 17 parameters on which classification can be based. Kutafina et al. (2015) demonstrated that using all of this data it is possible to train neural networks

that correctly classify fairly complex hand movements.

In conclusion, the limited data available prohibits a definite answer of how the Myo controller compares to medical grade EMG sensors. Nevertheless, to the best knowledge of the author there is no publication that questions the validity of the Myo's EMG data on a fundamental level. Hence, based on the data available it is safe to say that the controller is an affordable and user friendly alternative for applications that don't require the precision of high density EMG.

Myo for Research and Rehabilitation

In this section existing and potential applications that use the Myo controller for neurological research and/or rehabilitation are discussed. Lipovsky and Ferreira (2015) propose a low cost Myo based system for hand rehabilitation for patients suffering from stroke. The subjects wear the Myo controller on their healthy hand while wearing a robotic glove on their impaired hand. The movements they execute with their healthy hand while playing a virtual reality computer game are mimicked by the robotic glove (Lipovsky and Ferreira, 2015). Unfortunately, while the system has been proposed and tested on healthy subjects no results working with impaired patients have been published yet.
Qamar et al. (2015) present a multi-sensor therapy approach that uses the Myo controller as well as the Kinect sensor and the Leap Motion controller described in section 4.2.2. They use gestures recognized by those devices to control home appliances (e.g. lights, fan, etc.). Simultaneously they record the raw hand data to monitor the therapeutic progress (Qamar et al., 2015). Again, no data assessing the effectiveness of this kind of therapeutic intervention has been published yet.
Other studies that discussed applications in a non-medical environment are easily applicable for a medical-environment as well. Kerber et al. (2015) introduced a Myo-based smartwatch control. The authors argue that this one-handed control of a smartwatch is useful for *"situations that do not allow for opposite-side hand interactions"* (Kerber et al., 2015). Hence, applications for hemiplegic patients would spring to mind.
The same holds true for a study by Nymoen et al. (2015) that explored the Myo controller for musical interaction. The authors created a system that uses Myo's muscle and motion activity to select and modify sounds, control melody

and play impulsive sounds like drums. The study concludes that the data provided by the Myo controller is sufficient for sound production and modification (Nymoen et al., 2015). Using the ideas and results presented a therapeutic intervention that combines physio- and musicotherapy can be easily imagined.

Conclusion

To conclude this section, very little scientifically relevant data that assesses the abilities of the Myo controller is available. Nevertheless, from the data available and the applications that have been developed the Myo looks promising. Additionally, the Myo transmits all available data wirelessly which makes it a convenient device for mobile EMG recording (e.g. for hybrid BCIs), out-patient rehabilitation as well as for human computer interaction. Therefore, the Myo has been included into the proposed software platform. For a further discussion of the implementation see section 5.3.3

5. Development

As the main goal of this thesis was to lay the groundwork for an LSL based system for mobile brain imaging, BCIs and rehabilitation scenarios, this chapter will describe how LSL was extended to facilitate those scenarios. Before presenting applications for both medical-grade (see section 5.2) and consumer hardware (see section 5.3) the LSL Configurator will be discussed in section 5.1.

Note: All applications presented in this section were developed in accordance with the LSL-Team in San Diego and will be integrated into the official LSL release repository. Furthermore, the applications' source code is open-source and will be made publicly available on the LSL GitHub repository.

5.1. LSL Configurator

One of the drawbacks of LSL being application based is that, particularly with applications and experimental setups that use multiple applications or multiple instances of the same application (e.g. for accessing multiple amplifiers), the desktop environment can get cluttered and confusing. Hence, it becomes easy to overlook or misconfigure an application which potentially renders entire recording sessions useless. Also multi-app setups are difficult to store and share between multiple workspaces as there are no project files that allow for a clean project definition and start up.
Those issues should be resolved with the LSL Configurator shown in figure 5.1. The LSL Configurator is a tool to create, save, load, export and start multi-application projects. The software presents a list of all available LSL hardware applications. By default those are all the applications within the LSL git repository. Those applications are listed within a configuration file which can be easily extended or modified. By double-clicking one of the apps the selected app is added to the *Selected Hardware* section of the configurator and thereby added to the project.

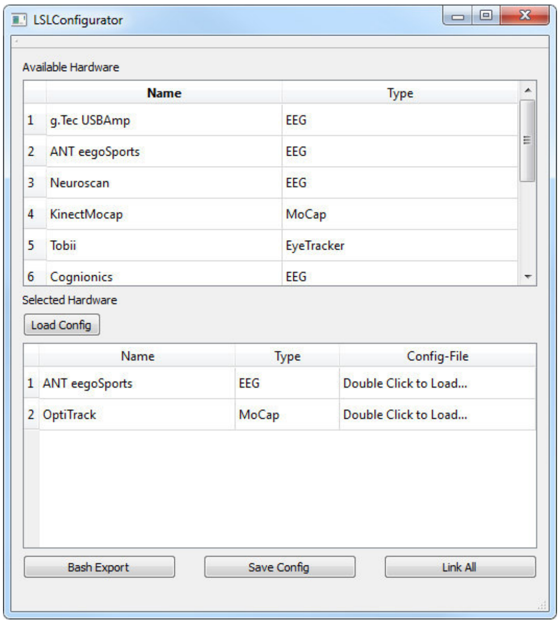


Figure 5.1.: The LSL Configurator. An application to manage multi-application projects.

As mentioned in section 3.1.2 LSL provides configuration files for many applications allowing the user to define configuration parameters such as sampling rate or chunk size in the case of amplifiers. Within the *Selected Hardware* section the user can specify such a configuration file for an application. The software then provides three different functionalities to use the configuration. The user can either save a project file, export a project file to an executable file or start all applications defined in a project.

Save a Project File

By saving a project file an eXtensible Markup Language (XML) file containing the entire project's information is created. The file contains all the selected applications and, if specified, their configuration files. Using those files project configurations can be shared between multiple users. Additionally, saved project files can be loaded from within the configurator to allow for project modification.

Export a Project File

While saving yields a non-executable XML file exporting creates an executable batch (.bat) file. A batch file is a plain text file that contains a series of commandline commands. In this case it contains commands to open each LSL hardware

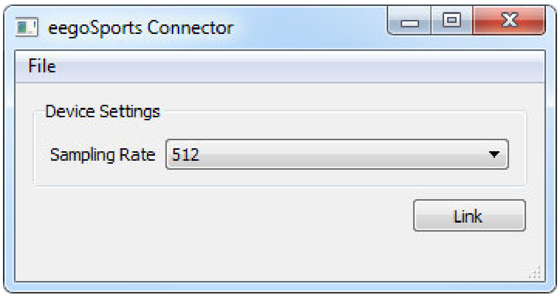


Figure 5.2.: The ANT eegosports LSL application

application specified in the project using the configuration files provided by the user. The applications and configuration files are referred to using absolute paths so exported batch files usually need to be adapted if executed on a computer different from the one that created the file.

Link a Project

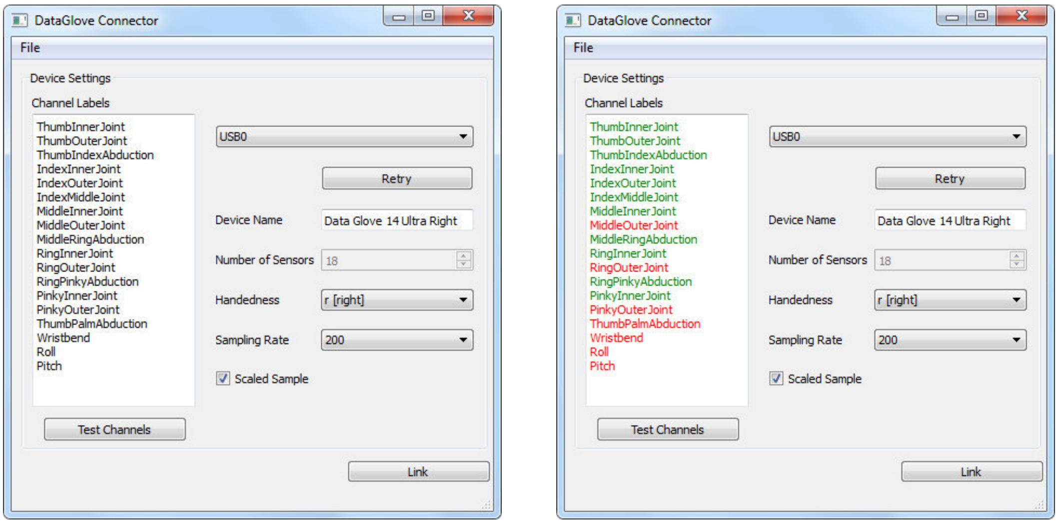
Besides saving and exporting a project file the user may also start all applications from within the configurator. The results are similar to when a project is exported and executed: each hardware application is opened and specified configuration files are applied.

5.2. Additional Hardware Applications - Medical-Grade Hardware

Some frequently used hardware devices were not yet supported by LSL. Apart from the configurator which improves the usability of LSL on a higher level, writing applications for those devices was the second major task of the thesis.

5.2.1. ANT Eego Sports

Whilst LSL supports a wide range of amplifiers the eego sports is not yet supported. As determined in section 4.1.1 the portable eego sports amplifier is useful for mobile brain imaging and mobile BCI applications. Hence, an LSL application for this amplifier has been developed.



(a)Default State (b)After Channel Test

Figure 5.3.: The DataGlove LSL application in its default state (5.3a) and after the channel test was run (5.3b). The application for the CyberGlove looks similar albeit providing more channels.

The Application Programming Interface (API) provided by the manufacturer does not allow for detailed configuration of the amplifier. Hence, the user interface of this application only lets the user specify the desired sampling rate and link the device as figure 5.2 shows.
Upon linking two LSL streams are created. The first stream provides the data coming from the amplifier's 32 or 64 channels. The second stream makes the data coming from the amplifier's trigger input available to LSL. Those to LSL streams are described in table 5.1 in more detail.

ANT eego sports Streams			
Stream Name	**Channels**	**Datatype**	**Sampling Rate**
eegoSports	32/64	Float	500 - 2048 Hz
eegoSportsMarkers	1	Integer	500 - 2048 Hz

Table 5.1.: The two streams provided by the ANT eego sports LSL application

5.2.2. Data Glove Applications

In section 4.1.2 two different data gloves were presented. Firstly the CyberGlove, secondly the 5DT Data Glove. While the implementations of the two applications differ due to the different interfaces provided by the manufacturers the applications themselves are structured similarly. See figure 5.3 for a screenshot of the DataGlove application. Both applications automatically detect all connected CyberGloves or DataGloves, respectively, as well as their handedness and their number of sensors. It is worth noting that even though all connected data gloves are displayed only the one selected is streamed to the LSL transport library. To stream multiple gloves multiple applications have to be opened.
Both data glove applications come with a feature to test the glove's sensors. If the *Test Channels* button is pressed the application will record a 5 second sample. During this time period the user is asked to extensively move his/her fingers. Depending on how many different values have been recorded the channel's colour in the channel-name list changes (red: <3; orange: 3 <= x <10; green: >10). The results of such a channel test can be seen in figure 5.3b.
When the link button is clicked a single LSL stream is created. Depending on the glove type the CyberGlove application streams either 18 or 22 channels as can be seen in table 5.2.

CyberGlove Streams		
Stream Name	**Channels**	**Datatype**
CyberGlove<SerialNumber >	18/22	Float

Table 5.2.: The stream provided by the CyberGlove LSL application

It is important to note that the values recorded by the CyberGlove and streamed by the application are not joint angles but 8 bit values that measure the resistivity of the bend sensors used (Blakely, 2013). Nevertheless, Blakely (2013) showed that the mapping between real joint angles and sensor resistivity is linear. Hence, the joint angles can easily be derived from the values provided by the glove.
The same holds true for the 5DT Data Glove. As shown in table 5.3 this application streams either 8 or 19 channels depending on which type of glove is used.

5DT Data Glove Streams		
Stream Name	**Channels**	**Datatype**
DataGlove<SerialNumber >	8/19	Float

Table 5.3.: The stream provided by the 5DT Data Glove LSL application

The 18-sensor CyberGlove provides the two bend sensors per finger, four abduction sensors to measure the abduction between fingers as well as sensors to measure thumb crossover, palm arch, wrist flexion and wrist abduction. The 22-sensor glove provides three bend sensors per finger. The values provided by those sensors are streamed by the LSL application as described in table 5.4.

CyberGlove: Streamed Data			
Index	**Value**	**Index**	**Value**
1 (1)	Thumb Near	12 (10)	Ring Near
2 (2)	Thumb Middle	*13*	*Ring Middle*
3 (3)	Thumb Far	14 (11)	Ring Far
4 (4)	Thumb/Index	15 (12)	Middle/Ring
5 (5)	Index Near	16 (13)	Pinky Near
6	*Index Middle*	*17*	*Pinky Middle*
7 (6)	Index Far	18 (14)	Pinky Far
8 (7)	Middle Near	19 (15)	Ring/Pinky
9	*Middle Middle*	20 (16)	Palm Roll
10 (8)	Middle Far	21 (17)	Wrist Flexion
11 (9)	Index/Middle	22 (18)	Wrist Abduction

Table 5.4.: The data streamed by the CyberGlove LSL application. Entries written in italics are only available for the 22-sensor CyberGlove. Indices in parenthesis are the indices as streamed by the 18-sensor Data Glove.

Regarding the 5DT DataGlove, the 5-sensor version provides one bend sensor per finger as well as wrist roll and wrist pitch. The 14-sensor version provides 2 bend sensors per finger as well as four finger abduction sensors and an additional wrist bend value. The structure of the stream created by the DataGlove LSL application can be seen in table 5.5.

5DT Data Glove: Streamed Data			
Index	**Value**	**Index**	**Value**
1 (1)	Thumb Near	*11*	*Ring Far*
2	*Thumb Far*	*12*	*Ring/Pinky*
3	*Thumb/Index*	13 (5)	Pinky Near
4 (2)	Index Near	*14*	*Pinky Far*
5	*Index Far*	*15*	*Thumb/Palm*
6	*Index/Middle*	*16*	*Wrist Bend*
7 (3)	Middle Near	17 (6)	Wrist Roll
8	*Middle Far*	18 (7)	Wrist Pitch
9	*Middle/Ring*	19 (8)	Gesture
10 (4)	Ring Near		

Table 5.5.: The data streamed by the Data Glove LSL application. Entries written in italics are only available for the 14-sensor Data Glove. Indices in parenthesis are the indices as streamed by the 5-sensor Data Glove.

5.3. Additional Hardware Applications - Consumer Hardware

While LSL already supports some consumer hardware like for instance the first generation of the before mentioned Microsoft Kinect sensor or the Nintendo Wiimote (Nintendo Inc., Kyoto, Japan) some promising hardware devices that were presented in chapter 4 are not supported yet.

In this section three new LSL applications for such devices will be introduced. Firstly, the LSL application for the consumer-grade EEG headset Emotiv EPOC will be presented. Secondly, the application for the Leap Motion controller that was introduced in section 4.2.2 will be described. Lastly, the application for the Myo wristband that was presented in section 4.2.3 will be discussed.

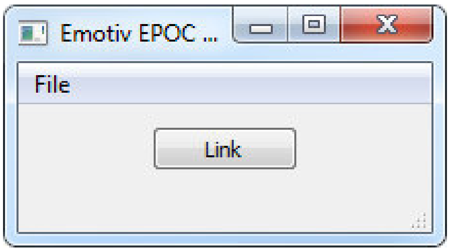


Figure 5.4.: The Emotiv EPOC LSL application

5.3.1. Emotiv EPOC

The Emotiv EPOC is a consumer grade EEG headset. As described in section 4.2.1 it provides 14 EEG channels as well as a two-axis gyroscope. Those values are streamed to LSL with a sampling rate of either 128 or 256 Hz, depending on the device type. Those sampling rates are fixed and cannot manually be set by the user. Therefore, the user interface of the EPOC's LSL application only provides the link button as figure 5.4 See the following table 5.6 for an overview of the streamed channels.

Emotiv Epoc: Streamed Data			
Index	**Value**	**Index**	**Value**
1	AF3	9	P8
2	F7	10	T8
3	F3	11	FC6
4	FC5	12	F4
5	T7	13	F8
6	P7	14	AF4
7	O1	15	Gyro X
8	O2	16	Gyro Y

Table 5.6.: The data streamed by the Emotiv Epoc LSL application. Sensor names refer to the international 10-20 electrode placement system.

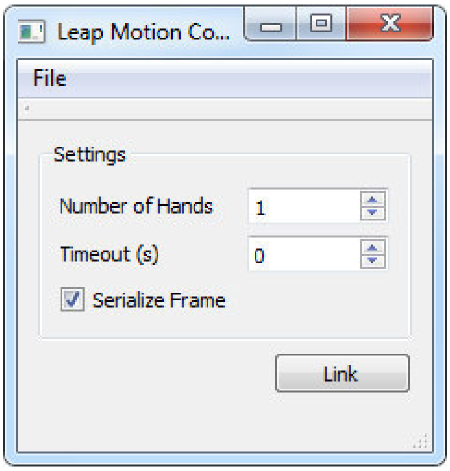


Figure 5.5.: The Leap Motion LSL application

5.3.2. Leap Motion

As described in section 4.2.2 the Leap Motion controller is a consumer hardware device that tracks the user's hand and fingers. The application which is shown in figure 5.5 can stream two different kinds of data. Firstly, hand and finger data of up to two hands. In this state the Leap Motion app basically mimics a data glove providing finger flexure and abduction data. Secondly, the application can make use of the serialization feature provided by the Leap Motion API. This mode of operation compresses all the data coming from the Leap Motion and allows the receiving application to use this data as if it was coming directly from the controller. Both streaming modes are listed in table 5.7 will be described in more detail below.

Leap Motion Streams			
Stream Name	**Channels**	**Datatype**	**Sampling Rate**
LeapMotion	20/40	Float	varying (~60Hz)
LeapMotionFrame	1	String	varying (~60Hz)

Table 5.7.: The two streams provided by the LeapMotion LSL Application

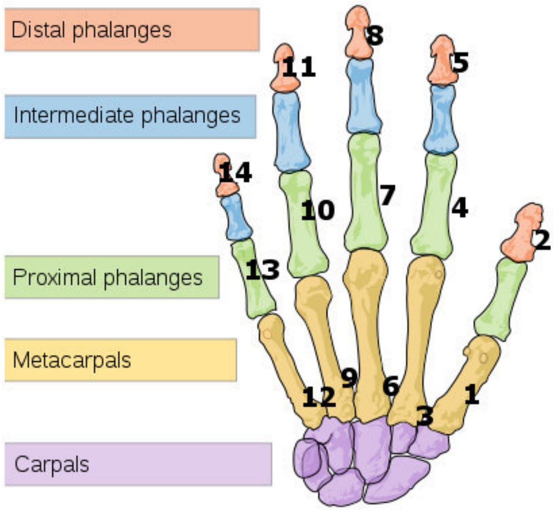


Figure 5.6.: The finger bones tracked by the Leap Motion controller and their respective indices within the samples streamed to LSL. Indices 3, 6, 9 and 12 refer to abduction angles between two adjoining fingers. All the other indices refer to the angle between the respective bone and the hand's palm.

Finger and Hand Data

This streaming mode is mostly targeted for research applications where raw finger and hand data is of interest. In this mode the application streams 20 values per hand for up to two hands. It is important to note that if two hands are tracked not both of them can be a right or left hand, respectively. If one left and one right hand are tracked the first 20 channels of the data stream characterize the left hand and the channels 21 to 40 the right hand. If only one hand is tracked only 20 channels are transmitted, regardless of a right or left hand is tracked.

For each finger the angle between the proximal phalanx and the distal phalanx respectively and the palm is computed. For the thumb the angle between the metacarpal and the distal phalanx respectively and the palm is computed. Additionally the abduction between two adjoining fingers is computed. The values are provided in degrees in order from thumb to pinky. Furthermore, the hand's direction as well as the hand's movement velocity is provided. The hand's direction is characterized by three rotation values: pitch (rotation around the x-axis), yaw (rotation around the y-axis) and roll (rotation around the z-axis). Similarly, the hand's velocity is described by providing the rate of change of the palm position along the x-, y- and z-axis. Table 5.8 provides an overview over the streamed data.

Leap Motion Tracking Data			
Index	**Value**	**Index**	**Value**
1	Thumb: metacarpal/palm	11	Ring: distal/palm
2	Thumb: distal/palm	12	Ring/Pinky
3	Thumb/Index	13	Pinky: proximal/palm
4	Index: proximal/palm	14	Pinky: distal/palm
5	Index: distal/palm	15	Hand: Pitch
6	Index/Middle	16	Hand: Yaw
7	Middle: proximal/palm	17	Hand: Roll
8	Middle: distal/palm	18	Hand: Velocity X
9	Middle/Ring	19	Hand: Velocity Y
10	Ring: proximal/palm	20	Hand: Velocity Z

Table 5.8.: The data streamed by the Leap Motion application for each hand. If two hands are tracked the indices refer to the indices of the first (=left) hand. The second (= right) hand's data has indices 21 to 40.

Note: The Leap Motion API does not match the standard anatomical naming system. As shown in figure 5.6 the thumb has no intermediate phalanx and thereby only three bones. For ease of programming the API introduces a zero-length metacarpal so that all fingers have four bones. The thumb's anatomical metacarpal bone is identified as a proximal phalanx and the anatomical proximal phalanx is identified as the intermediate phalanx in the Leap Motion finger bone model (Leap Motion, 2015). This does not affect the data provided by the application but is important for programmers working with the app's source code.

Serialization

The second streaming mode provided by the Leap Motion LSL application uses the concept of serialization. Serialization is a process where an object of a class is converted into a stream of bytes to store or in the case of this application to stream it to LSL.
Everything the Leap Motion controller sees such as hands, pointables, tools or gestures is stored in a *Frame* object which is outlined in figure 5.7. In other

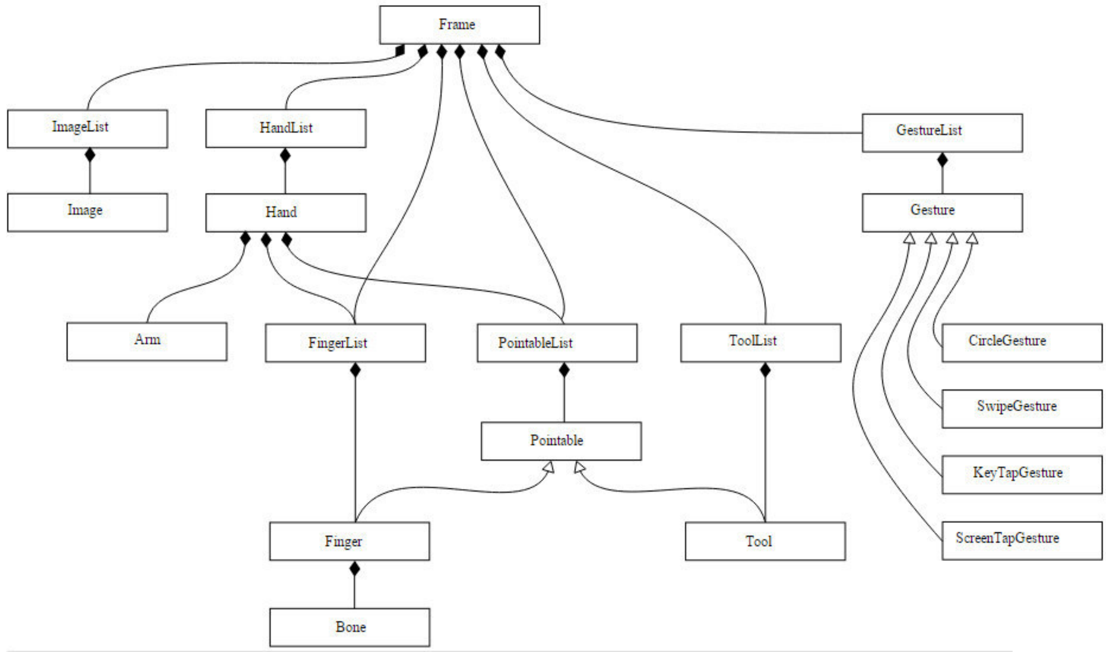


Figure 5.7.: The Leap Motion Frame object and all the information it contains. (Leap Motion, 2015)

words a Frame is basically a snapshot of the controller's field of view at a given time. The Frame's *HandList* for instance was used to compute the finger positions which were described in the section above. But the Frame provides much more information than just that. For example finger position, width and length, recognized gestures their speed and length, the hands sphere and much more can be retrieved. It would be infeasible and unnecessary for most applications to read out all this data and then stream it to LSL. Hence this second streaming mode was introduced. This mode serializes the Frame object to a Byte string and streams only this string to LSL. If an application is interested in this data it can then deserialize it (create a Frame object from the Byte string) using the Leap Motion API and use it as if the data was coming directly from the Leap Motion controller.

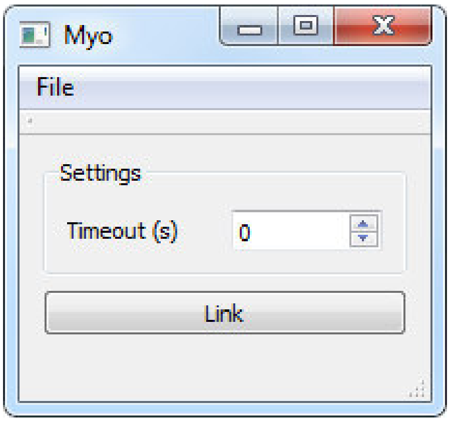


Figure 5.8.: The Thalmic Myo LSL application

5.3.3. Myo

The Myo controller was presented in section 4.2.3 and is a wearable consumer device that provides raw EMG, spatial as well as gestural data. The spatial data consists of orientation data, acceleration data as well as rotational data. It was also noted that while EMG data is streamed with a sampling rate of 200Hz, the spatial data is provided with only a quarter of that rate. Additionally the Myo API does not provide data in a continuous stream but uses an event-based mechanism for that purpose. It provides five event callbacks that fire when new EMG, acceleration, rotation, orientation, or gestural data is available. Due to this architecture it was decided that the Myo LSL application, which is shown in figure 5.8, should create five different LSL streams to not introduce any time lag as could be the case if for instance all spatial data would be combined into one stream. A list of streams is provided in table 5.9.

Thalmic Myo Streams			
Stream Name	**Channels**	**Datatype**	**Sampling Rate**
MyoEMG	8	Float	200 Hz
MyoAccelerometer	3	Float	50 Hz
MyoGyroscope	3	Float	50 Hz
MyoOrientation	7	Float	50 Hz
MyoGesture	1	Integer	variable

Table 5.9.: The five streams provided by the Myo LSL application

Figure 5.9.: The numbering of Myo's EMG sensors (Thalmic Myo, 2016). The numbers correspond to the indices in the data array provided by the LSL application.

EMG Data

The LSL stream provides the data coming from the eight EMG sensors integrated into the Myo wristband. Each of the band's links contains one sensor. The indices of the sensors never change. In accordance with the API implementation the link with the glowing Myo logo is always sensor four. To the left of it are sensors 1-3 and to the right sensors 5-8. For the sensor's numbering please refer to figure 5.9.

Spatial and Gestural Data

Apart from raw EMG the Myo wristband also provides spatial and gestural data. The spatial data consists of orientation data (x, y, z, w) as a Quaternion, acceleration data (x, y, z), provided in units of g, as well as angular data (x, y, z), provided in degrees per second. For each of those data streams there is one LSL stream. The LSL stream that provides orientation data also provides data describing the arm's roll, pitch and yaw. Those values are computed from the Quaternion provided by the Myo API. This yields a stream consisting of seven channels. The gestural data is transmitted as a single integer between 0 and 6. 0 indicates no detected gesture and 1-5 indicate one of the gestures described in figure 4.7. For the mapping between gestures and integer values please see table 5.10.

Myo Gestures	
Gesture	**Value**
Wave Left	1
Wave Right	2
Double Tab	3
Fist	4
Fingers Spread	5

Table 5.10.: The available gestures and their integer representation

5.4. Summary

In this chapter newly written LSL applications were presented. Using these applications LSL can now be used for data acquisition in mobile brain imaging, mobile BCI and mobile rehabilitation scenarios. An LSL-based application for one such scenario will be presented in chapter 7 where a LSL-based game for rehabilitation is introduced. Before that chapter 6 presents some of the experiments that were used to validate LSL and the newly written applications.

6. Validation

This chapter discusses experiments that were conducted to validate the newly implemented LSL applications as well as LSL itself. If not stated differently all used devices were connected to the same computer (Windows 7 32bit, 4GB RAM, 2 Core 2.4GHz).

6.1. LSL Network Synchronization Test

Even though there is very little doubt about the abilities of LSL to synchronize multiple data streams as it is currently used for synchronization sensitive tasks in other laboratories (e.g. Gramann et al., 2014) those abilities were tested. The experimental setup looked as follows: four medical-grade EEG amplifiers were used and plugged into two different computers in pairs of two. To have a comparable signal a signal generator (g.Tec g.SIGgen, g.Tec Medical Engineering GmBH, Graz, Austria) was used to generate a sine wave and was connected to all the amplifiers. The amplifiers' data was streamed to LSL using the LSL g.USB amp applications. See the following table 6.1 for the list of amplifiers and their configurations.

Amplifier Name	Location	Sampling Rate
Amp 1	Remote	256 Hz
Amp 2	Local	512 Hz
Amp 3	Remote	512 Hz
Amp 4	Local	512 Hz

Table 6.1.: The amplifiers used in this experiment. The location specifies whether the amplifier was connected directly to the recording computer (Local) or streamed its data over the network (Remove).

The signal acquisition was performed on one of the two computers using the LSL LabRecorder. Both computers ran Windows 7.

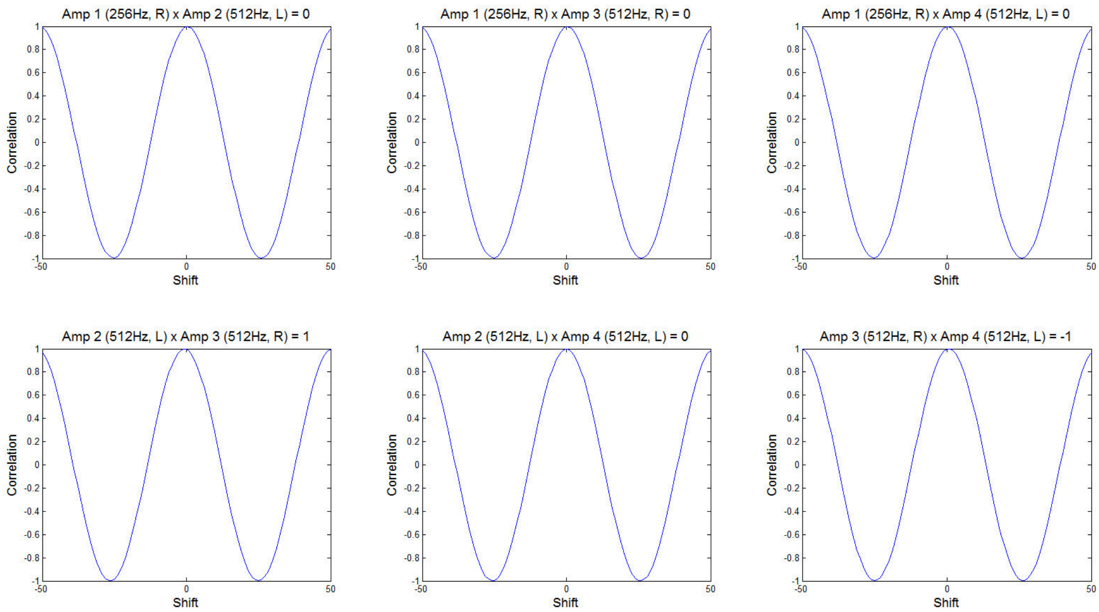


Figure 6.1.: The cross-correlations computed for the four amplifiers with each other.

Results

The data collected by the LabRecorder was imported into MATLAB (R2011a) and synchronized using the MATLAB Importer provided by LSL. The 256Hz signal was interpolated to match a 512Hz signal using the MATLAB function *interp1*. Subsequently, the cross-correlation of the four signals with each other was computed using the MATLAB function *xcorr* resulting in 6 cross-correlation values. As can be seen in figure 6.1 the cross-correlation was zero in four cases and one for the remaining two cases. A cross-correlation of one corresponds to an offset of not more than one sample. Hence, for 512Hz signals the offset is smaller than 2ms which is sufficiently accurate.

6.2. ANT eegosports and DataGlove

A standard finger-tapping experiment (e.g. Pfurtscheller and Neuper, 1992, Darvas et al., 2010, Paek et al., 2014) was conducted to validate the newly implemented ANT eegosports and DataGlove applications. According to Pfurtscheller and Neuper (1992) self-paced, voluntary finger movements cause a band power decrease around frequencies of 10 Hz. This movement-related band power decreased is called Event-Related Desynchronization (ERD) (Pfurtscheller and

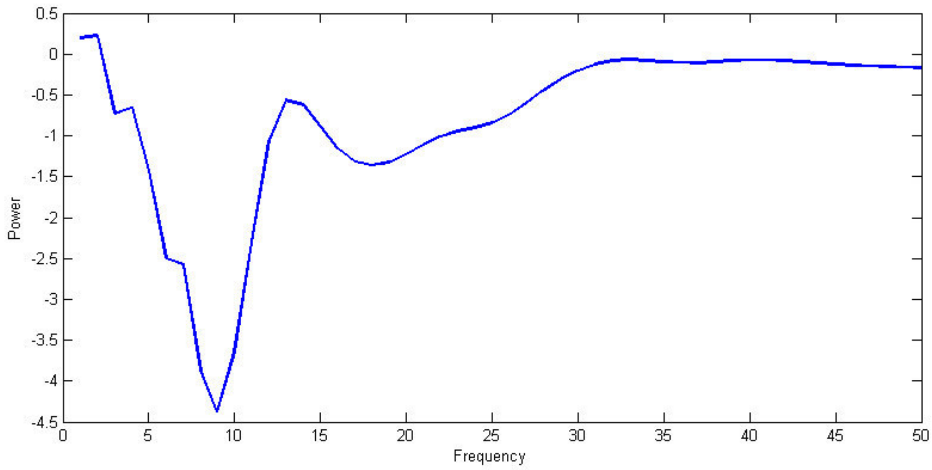


Figure 6.2.: Event-Related Desynchronization at 10 Hz computed from an EEG signal recorded with the ANT eegosports LSL application.

Lopes da Silva, 1999). The goal of the experiment was to replicate those findings. EEG data was recorded using 64 wet electrodes, the 14 channel 5DT DataGlove was used to record finger tapping. The two LSL streams were collected by the LabRecorder software provided by LSL. Four runs lasting for five minutes each were recorded. Each run consisted of multiple arbitrary long tap/no-tap sequences.

Results

The data recorded by the LabRecorder was imported into MATLAB and synchronized for offline analysis using the MATLAB importer. Tap and no-tap sequences were localized using the data provided by the DataGlove application and separated into one tap and one no-tap EEG signal. For both EEG signals the power spectral density was computed. Thereafter the no-tap frequency band was subtracted from the tap frequency band to single out frequency changes unique to the tap frequency band. As can be seen in figure 6.2 the obtained power spectrum showed a distinct decrease in band power at around 10 Hz. Hence the experiment to replicate previously obtained results was successful. This supports the conclusion that the newly developed ANT eegosports and 5DT DataGlove applications correctly acquire data from the amplifier and data glove, respectively.

6.3. Emotiv EPOC

As shown by Liu et al. (2012) and Pröll (2012) and discussed in more detail in section 4.2.1 the Emotiv EPOC is able to detect SSVEPs, that is increased brain activity at the same frequency as a visual stimulus the participant looks at. Hence, to validate the newly written LSL application a standard SSVEP experiment (e.g. Gao et al., 2003) was conducted. To detect a SSVEP the subject was asked to look for 30 seconds at a visual stimulus presented at either 7, 12 or 20 Hz while wearing the Emotiv EPOC headset. The EPOC data was provided to LSL using the newly developed application and recorded using the LSL LabRecorder.

Results

The recorded data was imported into MATLAB using the LSL MATLAB Importer. A 10 seconds sample was taken out of the 30 second recording. As visual stimulation yields evoked potentials in the visual cortex only the data from electrodes O1 and O2, which cover the occipital lobe where the visual cortex is located, were analysed (Kübler and Müller, 2007). A fast Fourier transform (FFT) was computed to get the frequency spectrum of those two electrodes. Using this setup detection of a SSVEP was unsuccessful as no significant increase in frequency at 7, 12 or 20 Hz, respectively, could be observed.
However, strong occipital alpha waves that occur while the subjects' eyes are closed as well as strong EEG artefacts, caused for instance by grinding of teeth, are visible in the EEG data recorded using the EPOC LSL application. Therefore, it was concluded that the inability to detect SSVEPs is a signal processing not a signal acquisition related problem. While finding ways to properly process the data coming from the Emotiv EPOC is important it was not within the aim of this thesis to do so.

6.4. Leap Motion

Leap Motion's ability to detect finger-tapping was verified. To do so the hand was recorded while wearing a 5DT DataGlove which has successfully been used for finger-tapping experiments before. Tapping of the right index finger was performed. For the Leap Motion controller the angle between proximal

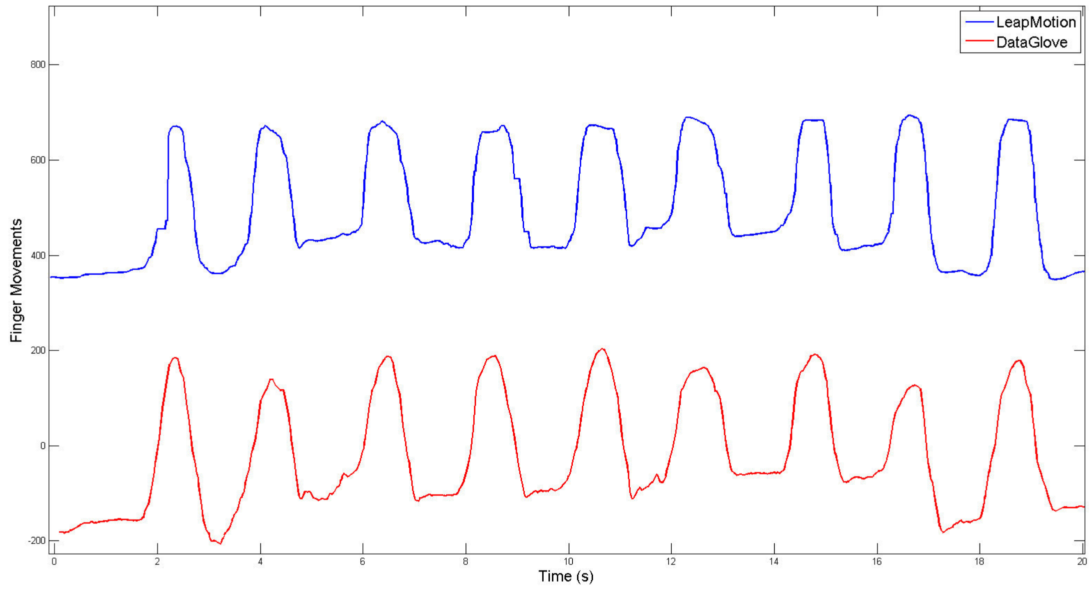


Figure 6.3.: Finger Tapping recorded by both the Leap Motion and the 5DT Data Glove. Streams recorded using the LSL LabRecorder and synchronized using the MATLAB Importer. The Leap Motion stream was shifted upwards for representational purposes.

phalange and palm was used. This corresponds to the Data Glove's 'Index Near' sensor. The data was recorded using the LSL LabRecorder.

Results

Again, the data collected by the LabRecorder was imported into MATLAB and synchronized using the MATLAB Importer provided by LSL. As can be seen in figure 6.3 both Leap Motion and DataGlove accurately recorded the tapping motion. There is no noticeable difference between the two streams regarding general finger movement. Moreover, the cross correlation between the two signals was computes using the MATLAB function *xcorr* which yield a correlation of 0.97. Some dissimilarity in the recorded data become apparent where the direction of movement changes and the tracked finger shakes ever so slightly. To quantify this divergence more elaborate tests using either a robotic hand or a highly accurate motion tracking system would be necessary. However, for this thesis the aim of the experiment was to validate the implementation of the LSL application, not the Leap Motion controller itself. The results presented in figure 6.3 and subsequently in chapter 7 do just that.

7. Application

7.1. Mobile Rehabilitation using LSL - A Proof of Concept

In section 3.2 multi-sensor applications for rehabilitation have been presented as one possible scenario where LSL might be of use. To reiterate, the idea of those systems is to use multiple (consumer) devices to facilitate out-patient, possibly game-based, rehabilitation. Apart from being affordable it was important to accurately measure the patient's progress.
To demonstrate the abilities of this LSL based, mobile system for a rehabilitation scenario a simple game was developed. Multiple affordable consumer devices are used to control the game's main character. LSL is used to retrieve data from the consumer hardware and to collect markers from within the game. The data streamed from the consumer devices and made available to the LSL transport layer is then used in two applications: Firstly, within the game where the raw data is mapped to controller commands. Secondly, the data as well as the in-game markers are recorded and stored for further analysis by an expert (e.g. the patient's therapist). This approach of combining raw movement data with game events, as proposed by Broeren et al. (2006), allows for a better interpretation of the patient's performance, limitations and improvements. For instance Cameirão et al. (2010) successfully used similar in-game events to analyse the performance of stroke patients using the rehabilitation gaming system proposed by the authors .
The purpose of this game is twofold. On the one hand, LSL's functionality to stream data remotely and to access streams simultaneously on multiple machines should be demonstrated. On the other hand, the newly developed LSL applications should be validated and the system's versatility should be demonstrated.

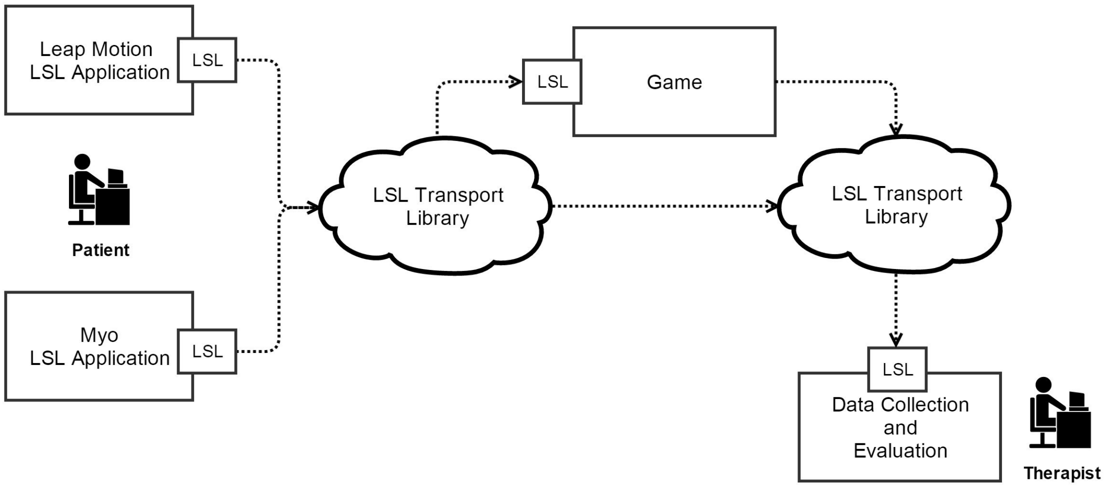


Figure 7.1.: Structure of the proposed game-based rehabilitation environment.

7.1.1. Used Hardware

For the first exemplary release of this game two hardware devices were included. Again, with regard to demonstrating an affordable solution using newly implemented devices the Leap Motion hand and finger tracker as well as the Myo EMG wristband were used.

7.1.2. Development Environment

The game was developed using the Unity game engine (Unity Technologies, San Francisco, CA, USA) version 5.2. In-game code is written in C#. The main reasons for selecting this game-engine were the multi platform support (including all major operating systems, game consoles, as well as iOS and Android), the large community and the fact that the platform is available for free.

7.1.3. Building Blocks

1. **The LSL Applications:** For each hardware device the respective LSL application has to be used to make the data available to the applications described below.

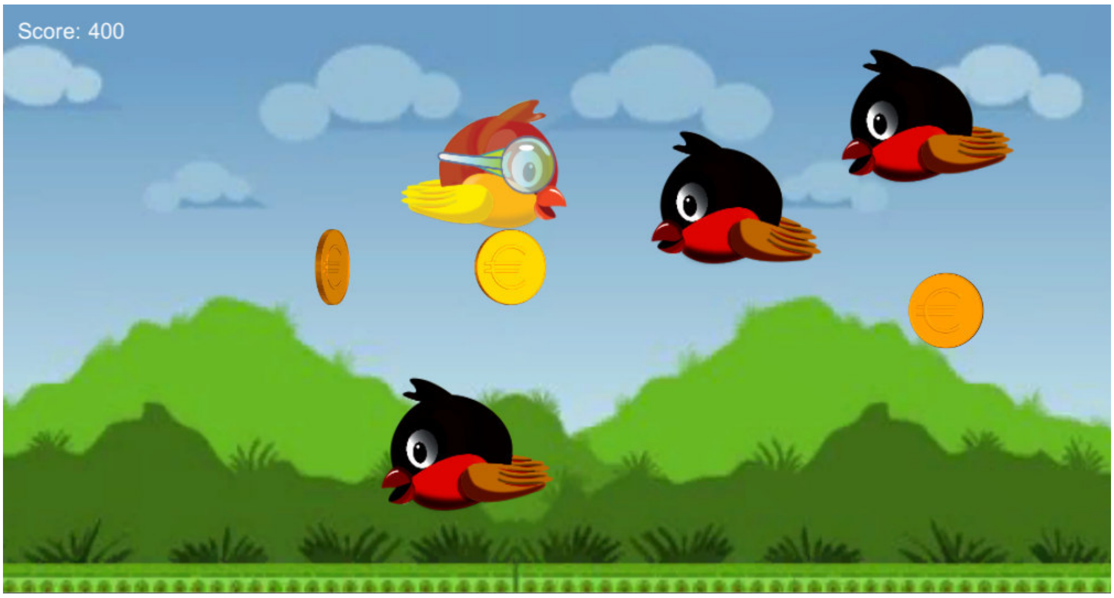


Figure 7.2.: The proposed game.

2. **Command Parser:** This module reads and interprets the data streams that were made available by the LSL applications and converts them into simulated keystrokes. For instance, and upwards movement of a specific finger will be converted into an *Arrow Up* key event. The generated key events will be recognized by all applications running on that machine.

3. **The Game:** The actual game where the generated key strokes will be used to control a character. Additionally, markers are streamed to LSL containing information of the patient's progress.

4. **The Therapist's Interface:** All LSL data is recorded using the recording software provided by LSL. This data comprises all the streams created by the hardware devices' LSL applications as well as in-game markers.

7.1.4. The Game

The premise of this game is simple. The game character, a bird, has to collect coins and dodge or shoot oncoming opponents (birds as well). For each coin the user collects 5 points are added to the overall score. For each coin the user misses 5 points are subtracted. Upon collision with oncoming birds the game ends. The main character's position on the x-axis (left-right) is fixed. Only its position on the y-axis (up-down) can be controlled. For each upward or downward movement the bird's position changes by a fixed amount.

LSL Integration

As mentioned above, LSL is not only used to retrieve data from the hardware devices but also to collect in-game markers. Both the hardware data and the in-game markers are then stored for further analysis.
The markers are streamed to LSL as strings with irregular timing as data is only pushed to LSL upon certain events occur. Such events are character controls (up, down, shoot), collection of a coin, not collected coins as well as collisions with opponents. As shown in table 7.1 each of these events is identified by a single character which is streamed to LSL. Those markers provide useful information for the therapist (cf. Cameirão et al., 2010). For instance the percentage of successful finger movements (i.e. how often the movement of the index finger actually triggers a character movement) becomes apparent. Figure 7.3 describes such a case by plotting index finger movements (provided by the LSL LeapMotion application) as well as *Player Up* events (streamed to LSL from within the game). As can be seen in the plot, the participant tried multiple times to trigger an upwards movement of the game character by raising his/her index finger but only the final most distinct motion led to a successful movement.

Markers	
Event	**Marker**
Player Up	u
Player Down	d
Player Shoot	s
Collected Coin	c
Missed Coin	m
Collision (Game Over)	g

Table 7.1.: Game events and their marker representation

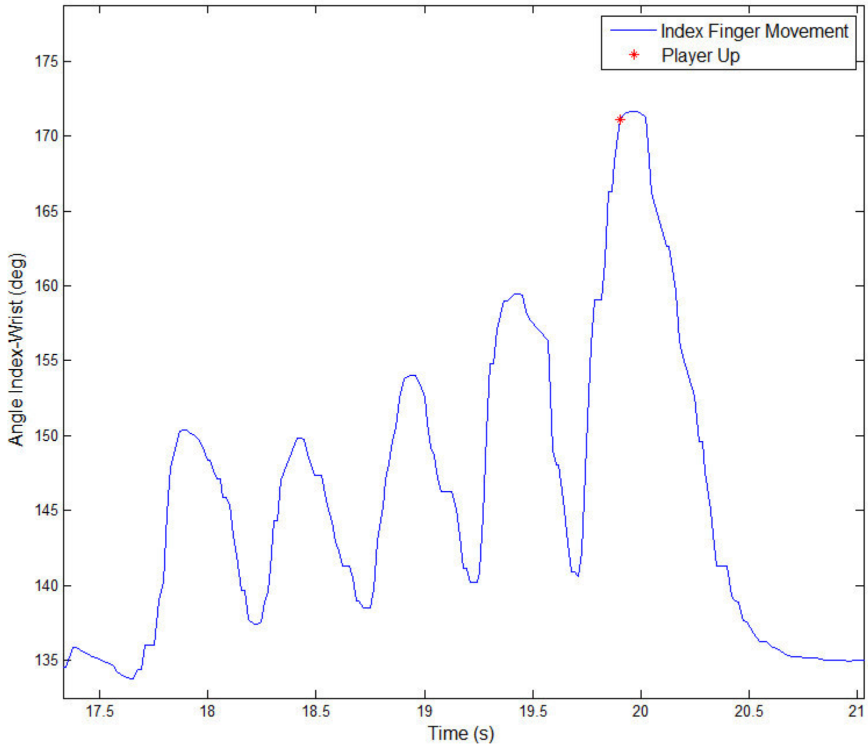


Figure 7.3.: The data recorded by LSL from the Leap Motion controller and the rehabilitation game. The participant tried multiple times to trigger an upwards movement of the game character but only the final most distinct motion led to a successful movement.

7.1.5. Game Control

At this point the game controls are rather simple consisting only of an upward and downward movement of the main character as well as the ability to shoot. To control the character either a standard computer keyboard or the consumer devices mentioned in 7.1.1 can be used.
Right now the game adapts two exercises that are widely used in physiotherapy for patients with movement disorders to control the main character. Firstly, extensive finger movement is used to control the game's main character. Finger movements are recorded using the Leap Motion controller. To do so the controller has to be positioned on a flat surface with the infrared sensors facing upwards. The user is then asked to position his/her hand at around 20cm above the controller. The character can now be controlled by stretching (up) or bending (down) the index finger.
Secondly, flexing of the user's forearm is used to let the game's main character

shoot. More specifically the user is asked to make a fist which is one of the default gestures detected by the Myo wristband. An overview over those controls is given in table 7.2.

Game Controls		
Control	**Keyboard**	**Sensor**
Player Up	Arrow Up	Leap Motion: Index finger up
Player Down	Arrow Down	Leap Motion: Index finger down
Shoot	Space	Myo: Fist

Table 7.2.: The controls used in the game and their mapping to both computer keyboard and sensor input

Flexibility of Controls Using this architecture of parsing control commands in a separate module allows for very flexible and interchangeable game control. The command parser takes stream input from LSL and converts it into universal control commands (e.g. arrow up, space). In this case a finger up movement detected by the Leap Motion controller is interpreted as an arrow up command. By using a different command parser another movement can trigger a specific control command, e.g. moving the entire hand upwards is interpreted as an arrow up command.

This approach is very useful for applications where a direct mapping of recorded movements to conventional input (e.g. keystrokes) is possible. Applications that use more precise controls (e.g. a mouse pointer) or virtual reality applications where the user's body and body movements are projected into the game would require a deeper integration of the recorded data into the application.

8. Discussion

In the previous chapters LSL was introduced as a powerful tool to acquire and synchronize data. The potential of LSL for mobile brain research, BCIs and rehabilitation was assessed and the range of applications was extended in this regard with a focus on mobile and affordable devices. It was shown that the data coming from the newly implemented applications is plausible and that, to the best knowledge of the author, LSL as such, as well as the additions that were made, work. Furthermore, by working in close collaboration with the creators of LSL the newly created applications will be added to the official LSL distribution and therefore can potentially be used in a wide range of research laboratories.

While the aim of this thesis was to get a better understanding of LSL and its abilities there are still steps to be taken to successfully integrate LSL into the existing research infrastructure as well as there is great potential to further extend the LSL software platform.

Firstly, as pointed out when assessing LSL for MoBI and mobile BCI (section 3.2) data acquisition and synchronization is only one building block of a complex data processing system. Hence, the next step will be the integration of LSL into the existing data processing environment. This task requires an excellent understanding of the field and the current work-flow. While this thesis can hopefully facilitate the integration it was neither the aim of the thesis nor within the expertise of the author to complete such an integration.

Secondly, LSL is a very promising platform that can be further explored in many directions. It has been argued throughout this thesis that affordability and mobility are key requirements for systems to be used by patients. In this regard, it would be interesting to explore the feasibility of running LSL on low cost single-board computers such as the Raspberry Pi (Raspberry Pi Foundation, Caldecote, UK). While it costs only USD 35 it provides hardware that should be powerful enough to acquire signals and stream it to LSL. The Raspberry Pi being credit-card sized it could be carried around easily which is important for

mobile, home-based BCIs as discussed by Nijboer and Broermann (2010). With a recent release of the operating systems Windows 10 (Microsoft Inc., Redmond, WA, USA) for Raspberry Pi this vision became even more viable. Therefore, the possibilities of running LSL on a Raspberry Pi are currently explored by researchers at the Institute of Neural Engineering.
Additionally, there are numerous interesting consumer devices that could be applicable for the scenarios defined in this thesis which were recently released or are still in development. Among those are the Sensoria (Sensoria Inc., Redmond, WA, USA) smart socks. The socks have three pressure sensors built in and could be useful for human gait experiments as well as for mobile rehabilitation. Another interesting device is the the Unlimited Hand (H2L Inc., Tokyo, Japan) wristband. The controller is similar to the Thalmic Myo presented in this thesis but provides haptic feedback which is a useful addition for hand rehabilitation. Another device worth noting is the Gest (Gest, Austin, TX, USA) for hand gesture recognition. Other than the Leap Motion discussed in this work, the Gest is a wearable device and transmits data wirelessly.
In short, a wide range of new affordable devices that track movements or fitness markers can be expected over the course of the next few years. And it will be interesting to see how those can be adapted for research and rehabilitation.

9. Conclusion

In this thesis the emerging areas of mobile brain imaging, mobile BCI as well as mobile rehabilitation were presented. Firstly, the requirements imposed on hard- and software by those three areas were discussed. EEG that uses lightweight, wireless amplifiers was presented as the most promising technology for MoBI. In terms of software the ability of synchronizing multiple devices that run on different computers was found to be a key requirement for MoBI. For BCIs it was said that mobile solutions for patients at home should be robust, portable and lightweight and, again, transmit data wirelessly. Also hybrid BCIs were introduced where, again, synchronization of multiple hardware devices is important. Mobile rehabilitation requires hardware devices that allow the user's progress to be tracked, are easy to setup and use and, most importantly, affordable.

Based on this information it was argued that a sensor platform that acquires and synchronizes data from a wide range of hardware devices that facilitate those mobile scenarios is a first important step toward the ultimate goal of a software system that provides solutions for all those areas. In this respect the Lab Streaming Layer (LSL) was introduced as a sensor platform that is able to fulfil those requirements. Furthermore, hardware devices were presented that can facilitate Mobile Brain Imaging, BCI and rehabilitation but were not supported by LSL yet. Apart from the medical-grade ANT eegosports amplifier and two different data gloves, promising consumer devices like the Emotiv EPOC EEG headset, the Leap Motion hand tracking controller and the Thalmic Myo EMG wristband were presented.

Subsequently, it was explained how LSL was extended to support those devices. Also, the LSL Configurator, a tool that simplifies multi-application projects, was introduced. Furthermore, the results of different experiments for validation purposes were presented. It was shown that the latency of multiple streams synchronized using LSL is minimal (not more than one sample). Moreover, the newly developed applications for the ANT eegosports EEG amplifier and the 5DT DataGlove were used to successfully replicate a standard finger-tapping experiment. Also, it was validated that the affordable Leap Motion controller (USD 70) provides similar data like the more expensive DataGlove (USD 5.500).

Lastly, a proof of concept for an LSL-based computer game for rehabilitation was demonstrated. It was proposed to use LSL to record both the hardware devices used to control the game and in-game event markers to provide therapists with a better insight into their patients' performance.

As all the applications that emerged from this thesis will be added to the official LSL repository a significant contribution could be made to the research lab in Graz and to the entire LSL community.

Appendix

Appendix A.

Working with LSL

A.1. Developing for LSL

A.1.1. Basics

LSL consists of a transport library and numerous applications that either send data to or read data from this transport library. The core library is written in C++ but can be accessed through language interfaces in a wide range of programming languages (C, C++, Python, Java, C#, MATLAB). Independent of the programming language accessing the transport library is pretty straightforward. To write data to the transport library three steps have to be taken:

1. Create a StreamInfo
2. Create an Outlet using the StreamInfo
3. Send data through this Outlet

Similarly, to receive data the following steps have to be taken:

1. Specify the stream of interest
2. Create an Inlet using found streams
3. Read data from this Inlet

Send Data

To send data firstly a StreamInfo has to be created. A StreamInfo characterizes the stream that is made available to LSL through this applications. It consists of the following information: the streams name and type (e.g. EEG), the number of channels that are transmitted through this streams, the sampling rate the data is provided with, the channel format (e.g. integer, float, string), as well a unique identification number to tell streams apart. Here's what such a StreamInfo specification would look like for an EEG stream with 8 channels and a SR of 512 in C++:

```
lsl::stream_info info("MyEEG",EEG,8,512,lsl::cf_float32,"ID");
```

Using this information an outlet can be created. An outlet is used to send data to the transport library. Here's how that works:

```
lsl::stream_info info("MyEEG",EEG,8,512,lsl::cf_float32,"ID");
lsl::stream_outlet outlet(info);
```

All that's left to do now is sending data to the transport library. This usually happens in some kind of loop where data is read from a hardware devices. But that's not required. Data can be send sample by sample or as a chuck of samples. Here's a while loop sending a single sample:

```
lsl::stream_info info("MyEEG",EEG,8,512,lsl::cf_float32,"ID");
lsl::stream_outlet outlet(info);
while(true) {
  float sample[8] = { 16, 2, 77, 40, 62, 54, 12, 11 };
  outlet.push_sample(sample);
}
```

If chunks of data should be streamed to LSL usually two dimensional arrays are used and the function *push_chunk* has to be used. That's basically it. Where ever data needs to be sent to LSL all that's needed is a StreamInfo, an outlet, and then this outlet can be used to push data to LSL.

Receiving Data

Receiving data works similarly. Firstly streams of interest have to be resolved. Streams can be identified using field-value pairs, where field can be for instance 'name' or 'type'. To resolve the above stream following call can be used:

```
std::vector<lsl::stream_info> results;
results = lsl::resolve_stream("name","MyEEG");
```

This returns a list of all streams (identified by their StreamInfo) that match the search query. Secondly, an Inlet has to be created through witch data can be read. To do so, pass the StreamInfo of the desired stream to a new inlet object. Like so:

```
std::vector<lsl::stream_info> results;
results = lsl::resolve_stream("name","MyEEG");
lsl::stream_inlet inlet(results[0]);
```

Lastly, data can be received using this inlet. Similar to the push call for sending data there are pull calls for receiving data.

```
std::vector<lsl::stream_info> results;
results = lsl::resolve_stream("name","MyEEG");
lsl::stream_inlet inlet(results[0]);
while (true) {
  float sample[8];
  double ts = inlet.pull_sample(sample,8);
}
```

In the example above *sample* now contains the received sample and *ts* the timestamp associated with this sample. For more examples see the LSL repository or `https://code.google.com/p/labstreaminglayer/wiki/ExampleCode`.

A.1.2. Application Development (C++)

Most of the LSL applications that acquire data from a hardware device are written in C++. This section quickly discusses the recommended development environment and code structure.

Development Environment and Setup

LSL applications make use of the Qt framework as well as the Boost library. At time of writing LSL applications use Qt 4.8.1. Even though there are newer version of Qt available it is recommended to use 4.8.X as well to keep things consistent. Same holds true for the Boost library where version 1.4.7 is used. To develop applications the Visual Studio environment is recommended which is freely available online. Qt4 natively supports VS2008 so this would be the recommended version of Visual Studio. Unfortunately, it is fairly old and not officially distributed anymore. If Qt is to be used with a newer version of Visual Studio it has to be compiled manually. To do so, the following links might be of help:

- Get QT Source zip from here: https://download.qt.io/archive/qt/4.8/
- Tutorial: http://menatronics.blogspot.fr/2012/12/compiling-qt-for -visual-studio-2012.html

- Fix for tutorial above: `http://stackoverflow.com/questions/18080625/`
- Tutorial x64: `http://stackoverflow.com/questions/12113400/`

Before starting the compilation make sure to replace the HashSet.h as well as to modify MathExtras.h as described in the links above. After compilation look at existing LSL applications on how to include Qt in your VS project. Adapt the paths for Additional Include Directories and for Additional Library Directories (right click on project and open Properties).

Application Structure
The file structure of a typical LSL application looks as follows:

- Form Files
  - mainwindow.ui
- Header Files
  - mainwindow.h
- Source Files
  - main.cpp
  - mainwindow.cpp

mainwindow.ui
A QT .ui file contains the user-interface declaration. Usually created and modified using the Qt Designer.

mainwindow.h
Header file for mainwindow.cpp. Contains definitions for both the MainWindow class and the reader thread.

main.cpp
Contains the main function. Reads command-line parameters such as default configuration files and creates the MainWindow.

mainwindow.cpp
Implementation of mainwindow.h. Contains methods to interact with the UI (e.g. load and save configuration files) and to read data from the desired hardware device using a separate thread.

Usually an LSL application to acquire data from a hardware device consists of two threads. The main thread handles UI interaction (device settings, load and save configuration files, etc.) and creates the second thread, the reader thread, when the user presses the link button. The reader thread (either a Boost or Qt thread) connects to the desired hardware device using, if available, the device

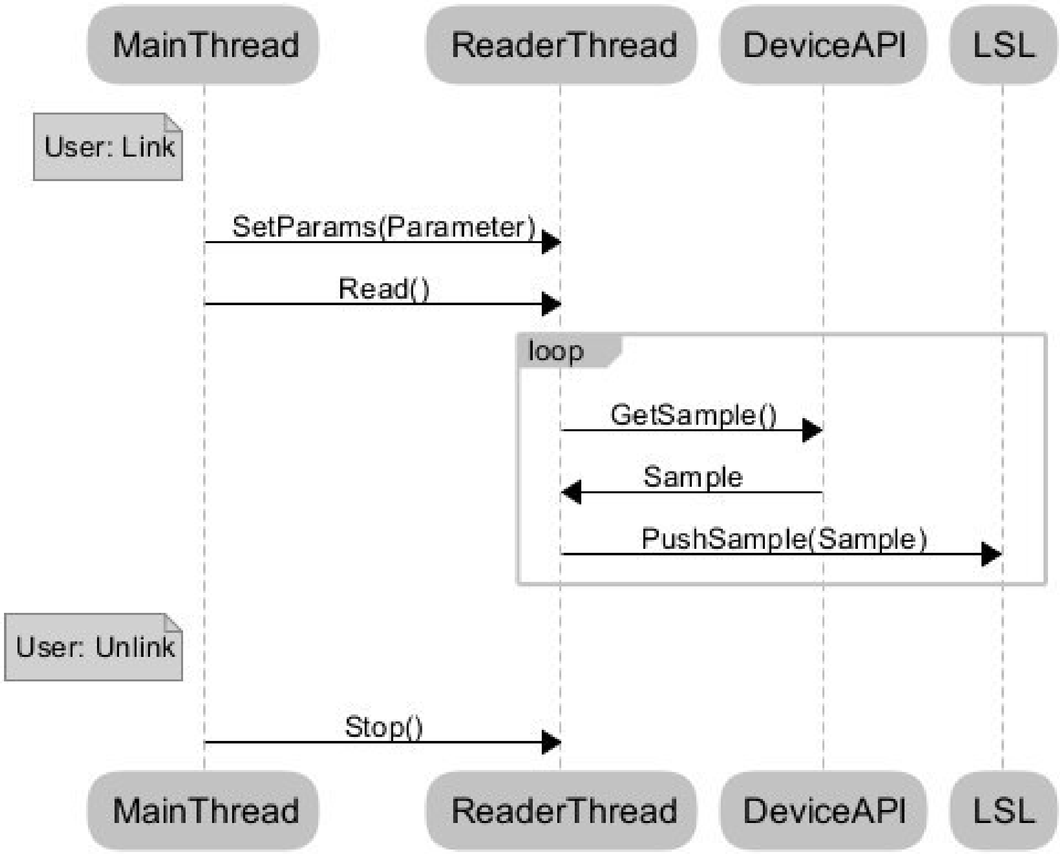


Figure A.1.: The interactions of the key components of an LSL application during a standard recording sequence initialized and terminated by the user's click on the Link button.

settings provided by the user and streams the data coming from the device to LSL. This is done until the user terminates the acquisition by clicking the link button again. Figure A.1 shows a simplified sequence diagram of a usual recording procedure.

A.2. Using LSL

There are three important parts of a successful LSL recording session:

1. LSL Hardware Applications
2. LSL LabRecorder
3. LSL MatlabImporter

Hardware Applications

Firstly, *hardware applications* as described above. Those applications are used to acquire data from a specific hardware device and to stream it to LSL. As soon as the application's link button is clicked its stream should be visible for other applications to read.

Lab Recorder

The LabRecorded is an application provided by LSL. It provides an overview over all streams available on the network (all applications that stream data to LSL) and means to record all of those or a selection of streams. The LabRecorder uses Python and PySide. To get the LabRecorder to work the following steps have to be taken (on a Windows machine):

1. Download Python 2.7 from here: `https://python.org/downloads/release/python-2711/`
2. Install PySide for Python 2.7 following this tutorial: `http://stackoverflow.com/questions/23576028`

See the following tutorial for how to use the LabRecorder: `https://code.google.com/archive/p/labstreaminglayer/wikis/LabRecorder.wiki`

The LabRecorder stores the entire stream data in an XDF file. XDF stands for Extensible Data Format, has been developed concurrently with LSL and is documented here: `https://code.google.com/archive/p/xdf/`.

MATLAB Importer

To import the data recorded by the LabRecorder into MATLAB the LSL MATLAB Importer can be used. The importer can be found within the LSL repository. To use it simply add the load_xdf.m file to your MATLAB path. Import an XDF-File using the load_xdf("Filename.xdf") call. Like so:

```
streams =load_xdf("recording.xdf");
```

Now *streams* is a cell array where each cell contains one stream. Each of those streams is a struct consisting of three fields:

1. **info:** contains all the information provided by the user when creating the StreamInfo as well as additional parameters such as clock offset measures or the effective sampling rate
2. **time_series:** the data that was recorded from this stream

3. **time_stamps:** the timestamp at which data was provided. Tor each entry in time_series there is one entry in time_stamps.

It is important to align the timestamps for all the streams. To do so find the minimal first timestamp of all available streams and substract this timestamp from every timestamp of every stream. Like so:

```
str1_ts = streams{1}.time_stamps;
str2_ts = streams{2}.time_stamps;
min_time_stamp = min(str1_ts(1), str2_ts(1));
str1_ts = str1_ts - min_time_stamp;
str2_ts = str2_ts - min_time_stamp;
```

Bibliography

5DT Inc (2015). *5DT DataGlove*. URL: www.5dt.com/?page_id=34 (visited on 11/23/2015) (cit. on p. 26).

Ahsan, R., M. Ibrahimy, and O. Khalifa (2009). *EMG signal classification for human computer interaction: A review* (cit. on p. 36).

Alankus, G., A. Lazar, M. May, and C. Kelleher (2010). "Towards customizable games for stroke rehabilitation." In: *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*. ACM Press, pp. 2113–2122 (cit. on p. 11).

ANT Neuro (2015). *ANT eego sports*. URL: https://www.ant-neuro.com/products/eego_sports (visited on 11/25/2015) (cit. on p. 25).

Arnett, J. and S. Labovitz (1995). "Effect of physical layout in performance of the Trail Making Test." In: *Psychological Assessment* 7, pp. 220–221 (cit. on p. 32).

Badcock, N., P. Mousikou, Y. Mahajan, P. de Lissa, J. Thie, and G. McArthur (2013). "Validation of the Emotiv EPOC(®) EEG gaming system for measuring research quality auditory ERPs." In: *PeerJ* 1 (cit. on pp. 28, 29).

Barry, G., B. Galna, and L. Rochester (2014). "The role of exergaming in Parkinson's disease rehabilitation: a systematic review of the evidence." In: *Journal of Neuroengineering and Rehabilitation* 11.1, pp. 33–43 (cit. on p. 11).

Blakely, T. (2013). "Decoding Coordinated Hand Movement in Human Primary Motor Cortex Using High Resolution Electrocorticography" (cit. on pp. 26, 43).

Borghese, N. A., M. Pirovano, P. L. Lanzi, S. Wüest, and E. D. de Bruin (2013). "Computational Intelligence and Game Design for Effective At-Home Stroke Rehabilitation." In: *Games for health journal* 2.2, pp. 81–88 (cit. on p. 12).

Broeren, J., M. Dixon, K. S. Sunnerhagen, and M. Rydmark (2006). "Rehabilitation after stroke using virtual reality, haptics (force feedback) and telemedicine." In: *Studies in health technology and informatics* 124, pp. 51–56 (cit. on p. 61).

Brown, M., W. Stuerzlinger, and E. J. M. Filho (2014). "The performance of un-instrumented in-air pointing." In: *Proceedings of Graphics Interface 2014*, pp. 59–66 (cit. on p. 33).

Brunner, C. et al. (2012). "BCI Software Platforms." In: *Towards Practical Brain-Computer Interfaces*. Ed. by B. Z. Allison, S. Dunne, R. Leeb, J. Del R. Millán, and A. Nijholt. Biological and Medical Physics, Biomedical Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 303–331 (cit. on pp. 20, 21).

Buch, E. et al. (2008). "Think to move: a neuromagnetic brain-computer interface (BCI) system for chronic stroke." In: *Stroke* 2, pp. 910–917 (cit. on p. 7).

Burke, J., M. McNeill, D. Charles, P. Morrow, J. Crosbie, and S. M. McDonough (2010). "Designing engaging, playable games for rehabilitation." In: *Proceedings of the 8th international conference on Disability, Virtual Reality and Associated Technologies*. ICDVRAT (cit. on pp. 10, 12).

Cameirão, M., S. B. Badia, E. Oller, and P. Verschure (2010). "Neurorehabilitation using the virtual reality based Rehabilitation Gaming System: methodology, design, psychometrics, usability and validation." In: *Journal of neuroengineering and rehabilitation* 7, p. 48 (cit. on pp. 61, 64).

Caraiman, S., A. Stan, N. Botezatu, P. Herghelegiu, R. G. Lupu, and A. Moldoveanu (2015). "Architectural Design of a Real-Time Augmented Feedback System for Neuromotor Rehabilitation." In: *2015 20th International Conference on Control Systems and Computer Science*. IEEE, pp. 850–855 (cit. on p. 12).

Choi, Y., N. Kwon, S. Lee, Y. Shin, C. Y. Ryo, J. Park, and D. Shin (2014). "Hypovigilance Detection for UCAV Operators Based on a Hidden Markov Model." In: *Computational and Mathematical Methods in Medicine* 2014 (cit. on p. 29).

Corbetta, D., F. Imeri, and R. Gatti (2015). "Rehabilitation that incorporates virtual reality is more effective than standard rehabilitation for improving walking speed, balance and mobility after stroke: a systematic review." In: *Journal of physiotherapy* 61.3, pp. 117–24 (cit. on p. 11).

Crosson, B., A. Ford, K. McGregor, M. Meinzer, S. Cheshkov, X. Li, D. Walker-Batson, and R. Briggs (2010). "Functional imaging and related techniques: an introduction for rehabilitation researchers." In: *Journal of rehabilitation research and development* 47.2, pp. 7–36 (cit. on pp. 3, 5).

CyberGlove Systems Inc (2015). *CyberGlove II*. URL: http://www.cyberglovesystems.com/cyberglove-ii/ (visited on 11/23/2015) (cit. on p. 26).

Daly, J. J., R. Cheng, K. Hrovat, J. M. Rogers, K. Litinas, and M. E. Dohring (2008). "Development and Testing of Non-Invasive BCI + FES/Robot System For Use in Motor Re-Learning After Stroke." In: *Proceedings of the 13th Annual Conference of the International Functional Electrical Stimulation Society*, pp. 200–202 (cit. on p. 7).

Darvas, F., R. Scherer, J. Ojemann, R. Rao, K. Miller, and L. Sorensen (2010). "High gamma mapping using EEG." In: *NeuroImage* 49.1, pp. 930–938 (cit. on p. 56).

De Sanctis, P., J. S. Butler, B. R. Malcolm, and J. J. Foxe (2014). "Recalibration of inhibitory control systems during walking-related dual-task interference: a mobile brain-body imaging (MOBI) study." In: *NeuroImage* 94, pp. 55–64 (cit. on p. 4).

De Vos, M., M. Kroesen, R. Emkes, and S. Debener (2014). "P300 speller BCI with a mobile EEG system: comparison to a traditional amplifier." In: *Journal of neural engineering* 11.3 (cit. on p. 8).

Debener, S., F. Minow, R. Emkes, K. Gandras, and M. de Vos (2012). "How about taking a low-cost, small, and wireless EEG for a walk?" In: *Psychophysiology* 49.11, pp. 1617–21 (cit. on pp. 9, 30, 31).

Delorme, A., T. Mullen, C. Kothe, Z. A. Acar, N. Bigdely-Shamlo, A. Vankov, and S. Makeig (2011). "EEGLAB, SIFT, NFT, BCILAB, and ERICA: New Tools for Advanced EEG Processing." In: *Computational Intelligence and Neuroscience* 2011 (cit. on pp. 4, 6).

Duvinage, M., T. Castermans, M. Petieau, T. Hoellinger, G. Cheron, and T. Dutoit (2013). "Performance of the Emotiv Epoc headset for P300-based applications." In: *Biomedical engineering online* 12.1, p. 56 (cit. on pp. 29, 30).

Ehinger, B. V., P. Fischer, A. L. Gert, L. Kaufhold, F. Weber, G. Pipa, and P. König (2014). "Kinesthetic and vestibular information modulate alpha activity during spatial navigation: a mobile EEG study." English. In: *Frontiers in human neuroscience* 8, p. 71 (cit. on p. 25).

Ellis, T., J. T. Cavanaugh, G. M. Earhart, M. P. Ford, K. B. Foreman, L. Fredman, J. K. Boudreau, and L. E. Dibble (2011). "Factors associated with exercise behavior in people with Parkinson disease." In: *Physical therapy* 91.12, pp. 1838–48 (cit. on p. 10).

Emotiv (2015). *Emotiv Epoc*. URL: `www.emotiv.com` (visited on 11/27/2015) (cit. on pp. 27, 28).

Estepp, J., J. Christensen, J. Monnin, I. Davis, and G. Wilson (2009). "Validation of a Dry Electrode System for EEG." In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, pp. 1171–1175 (cit. on p. 9).

Faller, J., C. Vidaurre, and T. Solis-Escalante (2012). "Autocalibration and Recurrent Adaptation: Towards a Plug and Play Online ERD-BCI." In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, pp. 313–319 (cit. on p. 9).

Farwell, L. and E. Donchin (1988). "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials." In: *Electroencephalogr. Clin. Neurophysio* 70, pp. 510–523 (cit. on p. 7).

Flores, E., G. Tobon, E. Cavallaro, F. I. Cavallaro, J. C. Perry, and T. Keller (2008). "Improving patient motivation in game development for motor deficit rehabilitation." In: *Proceedings of the 2008 International Conference in Advances on Computer Entertainment Technology - ACE '08*. ACM Press, pp. 381–384 (cit. on p. 11).

Galna, B., G. Barry, D. Jackson, D. Mhiripiri, P. Olivier, and L. Rochester (2014a). "Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson's disease." In: *Gait and Posture* 39.4, pp. 1062–1068 (cit. on pp. 11, 12, 27, 33).

Galna, B. et al. (2014b). "Retraining function in people with Parkinson's disease using the Microsoft kinect: game design and pilot testing." In: *Journal of Neuroengineering and Rehabilitation* 11, p. 60 (cit. on p. 11).

Galster, M. and M. Abduo (2015). *Myo Gesture Control Armband for Medical Applications*. URL: `http://www.cosc.canterbury.ac.nz/research/reports/HonsReps/2015/hons%7B%5C_%7D1502.pdf` (visited on 11/24/2015) (cit. on p. 36).

Gao, X., D. Xu, M. Cheng, and S. Gao (2003). "A BCI-Based Environmental Controller for the Motion-Disabled." In: *IEEE Transactions On Neural Systems And Rehabilitation Engineering* 11 (cit. on pp. 7, 58).

Godlove, J., A. Tsu, and K. Ganguly (2014). *Comparison of low-cost hand-monitoring devices for motor rehabilitation*. Poster presented at Society for Neuroscience Conference, 2014 (cit. on pp. 33, 34).

Gramann, K., D. Ferris, J. Gwin, and S. Makeig (2014). "Imaging natural cognition in action." In: *International Journal of Psychophysiology* 91.1, pp. 22–9 (cit. on pp. 4, 19, 55).

Gramann, K., D. Ferris, J. Gwin, K. Oie, T.-P. Jung, C.-T. Lin, L.-D. Liao, and S. Makeig (2011). "Cognition in action: imaging brain/body dynamics in mobile humans." In: *Reviews in the Neurosciences* 22.6, pp. 593–608 (cit. on pp. 3–6, 25).

Grivich, M. (2015). *LSL Validation*. URL: `http://sccn.ucsd.edu/~mgrivich/LSL_Validation.html` (visited on 10/15/2015) (cit. on p. 17).

Guna, J., J. Grega, M. Pogačnik, S. Tomažič, and J. Sodnik (2014). "An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking." In: *Sensors* 14, pp. 3702–3720 (cit. on pp. 32, 33).

Hettig, J., A. Mewes, O. Riabikin, M. Skalej, B. Preim, and C. Hansen (2015). "Exploration of 3D Medical Image Data for Interventional Radiology using Myoelectric Gesture Control." In: (cit. on p. 36).

Hondori, H. and M. Khademi (2014). "A Review on Technical and Clinical Impact of Microsoft Kinect on Physical Therapy and Rehabilitation." In: *Journal of Medical Engineering* 2014 (cit. on pp. 27, 32).

Iosa, M., G. Morone, A. Fusco, M. Castagnoli, F. R. Fusco, L. Pratesi, and S. Paolucci (2015). "Leap motion controlled videogame-based therapy for rehabilitation of elderly patients with subacute stroke: a feasibility pilot study." In: *Topics in Stroke Rehabilitation*, pp. 306–316 (cit. on p. 34).

Jack, D., R. Boian, A. Merians, S. Adamovich, M. Tremaine, M. Recce, G. Burdea, and H. Poizner (2000). "A virtual reality-based exercise program for stroke rehabilitation." In: *Proceedings of the fourth international ACM conference on Assistive technologies*. New York, New York, USA: ACM Press, pp. 56–63 (cit. on p. 26).

Johnson, M. J., M. Trickey, E. Brauer, and X. Feng (2004). "TheraDrive: a new stroke therapy concept for home-based, computer-assisted motivating rehabilitation." In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Vol. 7, pp. 4844–7 (cit. on p. 10).

Kaufmann, T., S. Völker, L. Gunesch, and A. Kübler (2012). "Spelling is just a click away - A user-centered brain-computer interface including auto-calibration and predictive text entry." In: *Frontiers in Neuroscience* 6.5, pp. 1–10 (cit. on p. 9).

Kerber, F., P. Lessel, and A. Krüger (2015). "Same-side Hand Interactions with Arm-placed Devices Using EMG." In: *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '15*. ACM Press, pp. 1367–1372 (cit. on p. 37).

Klose, C. (2007). "Connections that Count: Brain-Computer Interface Enables the Profoundly Paralyzed to Communicate." In: *NIH Medline Plus* 2, pp. 20–21 (cit. on p. 6).

Kranczioch, C., C. Zich, I. Schierholz, and A. Sterr (2014). "Mobile EEG and its potential to promote the theory and application of imagery-based motor rehabilitation." In: *International Journal of Psychophysiology* 91.1, pp. 10–5 (cit. on pp. 3–6, 28).

Kreilinger, A., V. Kaiser, C. Breitwieser, J. Williamson, C. Neuper, and G. R. Müller-Putz (2011). "Switching between Manual Control and Brain-Computer Interface Using Long Term and Short Term Quality Measures." In: *Frontiers in Neuroscience* 5, p. 147 (cit. on p. 8).

Kübler, A., E. Holz, T. Kaufmann, and C. Zickler (2013). "A User Centred Approach for Bringing BCI Controlled Applications to End-Users." In: *Brain-Computer Interface Systems - Recent Progress and Future Prospects*. Ed. by R. Fazel-Rezai. InTech (cit. on p. 9).

Kübler, A. and K.-R. Müller (2007). "An introduction to brain-computer interfacing." In: *Toward Brain-Computer Interfacing*. Ed. by G. Dornhege, J. del Millán, T. Hinterberger, D. McFarland, and K.-R. Müller. MIT Press (cit. on pp. 29, 58).

Kutafina, E., D. Laukamp, and S. M. Jonas (2015). "Wearable Sensors in Medical Education: Supporting Hand Hygiene Training with a Forearm EMG." In: *Studies in health technology and informatics* 211, pp. 286–91 (cit. on pp. 34, 36).

Lange, B., C.-Y. Chang, E. Suma, B. Newman, A. S. Rizzo, and M. Bolas (2011). "Development and evaluation of low cost game-based balance rehabilitation tool using the Microsoft Kinect sensor." In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, pp. 1831–4 (cit. on p. 11).

Lau, T., J. Gwin, and D. Ferris (2012). "How Many Electrodes Are Really Needed for EEG-Based Mobile Brain Imaging?" en. In: *Journal of Behavioral and Brain Science* 02.03, pp. 387–393 (cit. on p. 9).

Lau, T., J. Gwin, and D. Ferris (2014). "Walking reduces sensorimotor network connectivity compared to standing." En. In: *Journal of neuroengineering and rehabilitation* 11.1, p. 14 (cit. on p. 4).

Leap Motion (2015). *Leap Motion Controller*. URL: https://www.leapmotion.com (visited on 11/27/2015) (cit. on pp. 31, 49, 50).

Lipovsky, R. and H. A. Ferreira (2015). "Hand therapist: A rehabilitation approach based on wearable technology and video gaming." In: *2015 IEEE 4th Portuguese Meeting on Bioengineering* 2, pp. 1–2 (cit. on p. 37).

Liu, Y., X. Jiang, T. Cao, F. Wan, P. U. Mak, P.-I. Mak, and M. I. Vai (2012). "Implementation of SSVEP based BCI with Emotiv EPOC." In: *2012 IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS) Proceedings*. IEEE, pp. 34–37 (cit. on pp. 30, 58).

LSL (2015a). *LSL Introduction*. URL: https://code.google.com/p/labstreaminglayer/ (visited on 10/15/2015) (cit. on p. 15).

LSL (2015b). *LSL Supported Devices*. URL: https://code.google.com/p/labstreaminglayer/wiki/SupportedDevices (visited on 10/15/2015) (cit. on p. 16).

LSL (2015c). *LSL Time Synchronization*. URL: https://code.google.com/p/labstreaminglayer/wiki/TimeSynchronization (visited on 10/15/2015) (cit. on p. 18).

Makeig, S., K. Gramann, T.-P. Jung, T. Sejnowski, and H. Poizner (2009). "Linking brain, mind and behavior." In: *International journal of psychophysiology : official journal of the International Organization of Psychophysiology* 73.2, pp. 95–100 (cit. on pp. 3–5, 19).

Mhatre, P. V., I. Vilares, S. M. Stibb, M. V. Albert, L. Pickering, C. M. Marciniak, K. Kording, and S. Toledo (2013). "Wii Fit balance board playing improves balance and gait in Parkinson disease." In: *PM & R : the journal of injury, function, and rehabilitation* 5.9, pp. 769–77 (cit. on p. 11).

Mulas, M., M. Folgheraiter, and G. Gini (2005). "An EMG-Controlled Exoskeleton for Hand Rehabilitation." In: *9th International Conference on Rehabilitation Robotics, 2005*. IEEE, pp. 371–374 (cit. on p. 34).

Mullen, T., C. Kothe, M. Chi, A. Ojeda, T. Kerth, S. Makeig, T.-P. Jung, and G. Cauwenberghs (2015). "Real-time Neuroimaging and Cognitive Monitoring Using Wearable Dry EEG." In: *IEEE Transactions on Biomedical Engineering* 62.11, pp. 2553–2567 (cit. on pp. 5, 9).

Müller-Putz, G. R. et al. (2011). "Tools for Brain-Computer Interaction: A General Concept for a Hybrid BCI." In: *Frontiers in Neuroinformatics* 5, p. 30 (cit. on p. 8).

Muñoz, J., O. Henao, J. F. López, and J. F. Villada (2013). "BKI: Brain Kinect Interface, a new hybrid BCI for rehabilitation." In: *Games for Health*. Ed. by B. Schouten, S. Fedtke, T. Bekker, M. Schijven, and A. Gekker. Springer Fachmedien Wiesbaden, pp. 233–245 (cit. on p. 8).

Niell, C. and M. Stryker (2010). "Modulation of visual responses by behavioral state in mouse visual cortex." In: *Neuron* 65.4, pp. 472–9 (cit. on p. 4).

Nijboer, F. and U. Broermann (2010). "Brain–computer interfaces for communication and control in locked-in patients." In: *Brain-Computer Interfaces*, pp. 1–20 (cit. on pp. 8–10, 68).

Nijboer, F., B. van de Laar, S. Gerritsen, A. Nijholt, and M. Poel (2015). "Usability of Three Electroencephalogram Headsets for Brain–Computer Interfaces: A Within Subject Comparison." In: *Interacting with Computers* 27.5, pp. 500–511 (cit. on p. 30).

Nijboer, F., D. Plass-Oude Bos, Y. Blokland, R. van Wijk, and J. Farquhar (2014). "Design requirements and potential target users for brain-computer interfaces – recommendations from rehabilitation professionals." In: *Brain-Computer Interfaces* 1.1, pp. 50–61 (cit. on p. 9).

Nymoen, K., M. Haugen, and A. Jensenius (2015). *MuMYO - Evaluating and Exploring the MYO Armband for Musical Interaction*. Louisiana State University (cit. on pp. 37, 38).

Obdrzalek, S., G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel (2012). "Accuracy and Robustness of Kinect Pose Estimation in the Context of Coaching of Elderly Population." In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1188–1193 (cit. on p. 27).

Ojeda, A., N. Bigdely-Shamlo, and S. Makeig (2014). "MoBILAB: an open source toolbox for analysis and visualization of mobile brain/body imaging data." In: *Frontiers in human neuroscience* 8, p. 121 (cit. on pp. 6, 17, 19).

Oxford Dictionaries (2015). *Consumer Electronics*. URL: `www.oxforddictionaries.com/us/definition/american_english/consumer-electronics` (visited on 11/25/2015) (cit. on p. 23).

Paek, A., H. Agashe, and J. Contreras-Vidal (2014). "Decoding repetitive finger movements with brain activity acquired via non-invasive electroencephalography." In: *Frontiers in Neuroengineering* 7, p. 3 (cit. on p. 56).

Pamplona, V., L. Fernandes, J. Prauchner, L. Nedel, and M. Oliveira (2008). "The Image-Based Data Glove." In: (cit. on p. 25).

Patel, S., H. Park, P. Bonato, L. Chan, and M. Rodgers (2012). "A review of wearable sensors and systems with application in rehabilitation." In: *Journal of neuroengineering and rehabilitation* 9, p. 21 (cit. on p. 10).

Pfurtscheller, G., D. Flotzinger, and J. Kalcher (1993). "Brain-Computer Interface—a new communication device for handicapped persons." In: *Journal of Microcomputer Applications* 16.3, pp. 293–99 (cit. on p. 20).

Pfurtscheller, G. and F. Lopes da Silva (1999). "Event-related EEG/MEG synchronization and desynchronization: basic principles." In: *Clinical Neurophysiology* 110.11, pp. 1842–1857 (cit. on p. 56).

Pfurtscheller, G. and C. Neuper (1992). "Simultaneous EEG 10 Hz desynchronization and 40 Hz synchronization during finger movements." In: *Neuroreport* 3.12, pp. 1057–1060 (cit. on p. 56).

Pfurtscheller, G. et al. (2010). "The hybrid BCI." In: *Frontiers in neuroscience*, p. 30 (cit. on p. 7).

Pompeu, J. E., F. A. D. S. Mendes, K. G. da Silva, A. M. Lobo, T. d. P. Oliveira, A. P. Zomignani, and M. E. P. Piemonte (2012). "Effect of Nintendo Wii™-based motor and cognitive training on activities of daily living in patients with Parkinson's disease: a randomised clinical trial." In: *Physiotherapy* 98.3, pp. 196–204 (cit. on p. 11).

Pröll, M. (2012). "Using a Low-Cost Gyro and EEG-based Input Device In Interactive Game Design" (cit. on pp. 30, 58).

Qamar, A., A. Khan, S. Husain, A. Rahman, and S. Baslamah (2015). "A Multi-Sensory Gesture-Based Occupational Therapy Environment for Controlling Home Appliances." In: *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval - ICMR '15*. ACM Press, pp. 671–674 (cit. on p. 37).

Rahman, A. (2015). "Multisensor Serious Game-Based Therapy Environment for Hemiplegic Patients." In: *International Journal of Distributed Sensor Networks* 2015 (cit. on p. 12).

Rao, R. P. N. (2013). *Brain-Computer Interfacing - An Introduction*. first edition. Cambridge University Press (cit. on pp. 6, 7, 20, 28).

Reis, P. M. R., F. Hebenstreit, F. Gabsteiger, V. von Tscharner, and M. Lochmann (2014). "Methodological aspects of EEG and body dynamics measurements during motion." In: *Frontiers in human neuroscience* 8 (cit. on pp. 6, 9, 19, 24, 25).

Rozado, D., A. Duenser, and B. Howell (2015). "Improving the performance of an EEG-based motor imagery brain computer interface using task evoked changes in pupil diameter." In: *PloS one* 10.3 (cit. on p. 20).

Scherer, R., E. Friedrich, B. Allison, M. Pröll, M. Chung, W. Cheung, R. Rao, and C. Neuper (2011). "Non-invasive Brain-Computer Interfaces: Enhanced Gaming and Robotic Control." In: *Advances in Computational Intelligence*. Ed. by J. Cabestany, I. Rojas, and G. Joya. Springer Berlin Heidelberg (cit. on p. 7).

Scherer, R., G. Moitzi, I. Daly, and G. R. Muller-Putz (2013). "On the Use of Games for Noninvasive EEG-Based Functional Brain Mapping." In: *IEEE Transactions on Computational Intelligence and AI in Games* 5.2, pp. 155–163 (cit. on pp. 11, 12).

Scherer, R., G. Müller-Putz, E. Friedrich, V. Pammer-Schindler, K. Wilding, S. Keller, and J. Pirker (2015). "Games for BCI Skill Learning." In: *Handbook of Digital Games and Entertainment Technologies*. Ed. by R. Nakatsu, M. Rauterberg, and P. Ciancarini. Singapore: Springer Singapore, pp. 1–19 (cit. on p. 9).

Scherer, R., G. Müller-Putz, and G. Pfurtscheller (2007). "Self-initiation of EEG-based brain-computer communication using the heart rate response." In: *Journal of neural engineering* 4.4, pp. 23–9 (cit. on p. 7).

Scherer, R., J. Wagner, G. Moitzi, and G. Müller-Putz (2012). "Kinect-based detection of self-paced hand movements: Enhancing functional brain mapping paradigms." In: *Engineering in Medicine and Biology Society*, pp. 4748–4751 (cit. on p. 27).

Scherer, R., S. P. Zanos, K. J. Miller, R. P. N. Rao, and J. G. Ojemann (2009). "Classification of contralateral and ipsilateral finger movements for electrocorticographic brain-computer interfaces." In: *Neurosurgical Focus* 27.1, E12 (cit. on p. 26).

Seeber, M., R. Scherer, J. Wagner, T. Solis-Escalante, and G. R. Müller-Putz (2015). "High and low gamma EEG oscillations in central sensorimotor areas are conversely modulated during the human gait cycle." In: *NeuroImage* 112, pp. 318–26 (cit. on p. 4).

Shah, N., A. Basteris, and F. Amirabdollahian (2014). "Design Parameters in Multimodal Games for Rehabilitation." In: *Games for health journal* 3.1, pp. 13–20 (cit. on p. 12).

Shaughnessy, M., B. M. Resnick, and R. F. Macko (2006). "Testing a model of post-stroke exercise behavior." In: *Rehabilitation Nursing* 31.1, pp. 15–21 (cit. on p. 10).

Steinisch, M., M. G. Tana, and S. Comani (2013). "A Post-Stroke Rehabilitation System Integrating Robotics, VR and High-Resolution EEG Imaging." In: *IEEE transactions on neural systems and rehabilitation engineering* 21.5, pp. 849–859 (cit. on p. 11).

Stopczynski, A., C. Stahlhut, J. E. Larsen, M. K. Petersen, and L. K. Hansen (2014). "The smartphone brain scanner: a portable real-time neuroimaging system." In: *PloS one* 9.2 (cit. on p. 30).

Szafir, D. and B. Mutlu (2012). "Pay Attention! Designing Adaptive Agents that Monitor and Improve User Engagement." In: *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 11–20 (cit. on p. 7).

Tamez, E., J. Myerson, L. Morris, D. White, and C. Baum (2011). "Assessing executive abilities following acute stroke with the trail making test and digit span." In: *Behavioural Neurology* 24, pp. 177–85 (cit. on p. 32).

Tenore, F. V. G., A. Ramos, A. Fahmy, S. Acharya, R. Etienne-Cummings, and N. V. Thakor (2009). "Decoding of individuated finger movements using surface electromyography." In: *IEEE transactions on bio-medical engineering* 56.5, pp. 1427–34 (cit. on p. 36).

Thalmic Myo (2016). *Thalmic Myo Wristband*. URL: `https://www.thalmic.com/en/myo` (visited on 11/27/2015) (cit. on pp. 34–36, 52).

Tung, J., T. Lulic, D. Gonzales, J. Tran, and C. DIckerson (2015). "Evaluation of a portable markerless finger position capture device: accuracy of the Leap Motion controller in healthy adults." In: *Physiological Measurement* 36, pp. 1025–1037 (cit. on pp. 11, 12, 32–34).

Weichert, F., D. Bachmann, B. Rudak, and D. Fisseler (2013). "Analysis of the Accuracy and Robustness of the Leap Motion Controller." In: *Sensors* 13, pp. 6380–6393 (cit. on pp. 32, 33).

Wolpaw, J., D. McFarland, G. Neat, and C. Forneris (1991). "An EEG-based brain-computer interface for cursor control." In: *Electroencephalogr. Clin. Neurophysio* 78, pp. 252–259 (cit. on p. 7).

Wolpaw, J. and E. Winter-Wolpaw (2012). *Brain Computer Interfaces - Principles and Practice*. first edition. Oxford University Press (cit. on p. 20).

Wolpawa, J., N. Birbaumer, D. McFarland, G. Pfurtscheller, and T. Vaughan (2002). "Brain–computer interfaces for communication and control." In: *Clinical Neurophysiology* 113, pp. 767–791 (cit. on p. 7).

Yamaura, H., K. Matsushita, R. Kato, and H. Yokoi (2009). “Development of hand rehabilitation system for paralysis patient - universal design using wire-driven mechanism.” In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, pp. 7122–5 (cit. on p. 26).
Yao, B., S. Salenius, G. Yue, R. Brown, and J. Liu (2007). “Effects of surface EMG rectification on power and coherence analyses: an EEG and MEG study.” In: *Journal of neuroscience methods* 2, pp. 215–223 (cit. on p. 34).
Zwarts, M., G. Drost, and D. Stegeman (2000). “Recent progress in the diagnostic use of surface EMG for neurological diseases.” In: *Journal of Electromyography and Kinesiology* 10.5, pp. 287–291 (cit. on p. 34).