MANCHESTER
1824
The University of Manchester

Portable electroencephalography based closed-loop neuromodulation platform using lights and sounds

A thesis submitted to the University of Manchester for the degree of
Doctor of Philosophy
in the Faculty of Science and Engineering

2021

Nikhil Kurian Jacob
Department of Electrical and Electronic Engineering

Contents

Word Count: 47803

List of figures

List of tables

List of publications

N.K. Jacob, H.O. Kings, A.J. Casson, ”A smartphone based platform for portable non-invasive light and sound neuromodulation”, *Proc IEEE Eng Med Biol Soc*, pp. 5228–5231, Jul. 2020.

N.K. Jacob, E. Balaban, R. Saunders, J.C. Batchelor, S.G. Yeates and A.J. Casson, ”An exploration of behind-the-ear ECG signals from a single ear using inkjet printed conformal tattoo electrodes”, *Proc IEEE Eng Med Biol Soc*, pp. 1283–1286, Jul. 2018.

H.N. Locke, J. Brooks, L.J. Arendsen, N.K. Jacob, A.J. Casson, A.K. Jones, and M. Sivan, ”Acceptability and usability of smartphone-based brainwave entrainment technology used by individuals with chronic pain in a home setting”, *Br J Pain.*, vol. 14, pp. 161–-170, Aug. 2020.

Abstract

Neuromodulation is the intentional interference of brain-wave activity through the use of external stimuli such as lights and sounds. It has found various use-cases such as sleep engineering to enhance memory consolidation and chronic pain to enhance analgesic effects. Despite these, one of the major limitations here is the variability in outcomes that comes with the open-loop nature of sitmulation. To address this, we provide a smartphone based closed-loop stimulation platform where the stimuli is modified based on a user's current brain-state: a smartphone platform was used to increase ease of use and enable out-of-the-clinic experiments.

We present two different systems, each of which uses the Phase Locked Loop (PLL) to extract the feedback measure. The first is for phase-locked auditory stimulation for the sleep engineering application, where auditory tones are provided at the peak of a slow oscillation. The second platform is for continuous frequency-matched audio-visual stimulation, for the chronic pain application driver. Both these run on a smart-phone to enable 'out-of-the-clinic' experiments. The technology was validated using a test-phantom based setup to simulate a live human recording, allowing faster technology validation without any need for on-person tests.

For the sleep engineering platform, we obtained a phase error of 12±60 degrees and a loop latency of 70±24 ms. For the chronic pain platform, we report a frequency matching error of -0.17±1.29 Hz and 0.20±1.30 Hz for the visual and auditory modes, respectively. In addition to this, we showed that the PLL performs about 14% better in phase-tracking than the state-of-the art for all bands, confirming its wider applicability. Moreover, low inter-channel (<6% for 72 channels) and inter-subject (<4% for $n = 7$) variance was reported due to the adaptive nature of the PLL, negating any need for channel or subject specific tuning.

All these combined then provides a multi-purpose closed-loop platform running on a smartphone app, to enable *in-the-wild* research. Despite the target application drivers being chronic pain and sleep engineering, the system could be easily modified for other use-cases with different band bio-markers.

Declaration of originality

I hereby confirm that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright statement

i The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made *only* in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see `http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420`), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see `http://www.library.manchester.ac.uk/about/regulations/`) and in The University's policy on Presentation of Theses.

Acknowledgements

Firstly, I would like to express my sincere gratitude to Dr. Alex Casson for being the best supervisor one could have. Your guidance and supervision throughout my PhD have been instrumental and often sped things up; right from day 1, I appreciated the clarity, consistency and efficiency with which you worked. Hence, I must say it is not just the technical that you helped me with but also I was inspired by a lot of your non-technical qualities: the one that stuck with me most was treating projects as a 'marathon' and not a sprint.

I would also like to thank my co-supervisor Dr. Paul Wright for being an excellent mentor throughout my PhD. Oftentimes, when I was stuck or needed to bounce ideas off, you spent as long as needed without hesitancy and I am grateful for those long corridor talks, which often helped me gain clarity on problems I was facing. Moreover, thanks for the various technical contributions you bought to the PhD such as helping with the test-setup and giving critical feedback on the project, right from the beginning.

Next, I would like to thank everyone who made this PhD all the more special. It was during my time here that I found my love for entrepreneurship and I want to give a big thanks to Ellie Buckley, Laura Etchells, Tony Walker, Jess Grocutt, Lynn Sheppard and Nick Sanders for igniting this desire through your work at the Masood Enterprise Centre. I would also like to thank my team-mates at UoM Stories for the amazing time we had dreaming up and executing on this society idea: Jayesha Chudasama, Kathryn McGurk, Lorna Sinclair, Anuj Vaishnav and Donglin Liu, thank you all.

I would also like to thank my family. My father (Jacob Kurian), mother (Susan Jacob) and my sister (Nithara Jacob), have all been supportive throughout. Without them, I would not have received many of the opportunities I have had, so far.

Last but not least, I would also like to highlight that I was one of those lucky PhD students to have an amazing group of lab-friends around me, adding the fun element to my PhD. Thanks to Chris, Arturo, Andrius, Miguel, James, Eleanor, Sid, Meera, Mohamed, Tahmina and Ertan, for the amazing people you are.

Chapter 1

Introduction

1.1 Overview

The human brain is one of the most complex organs in the body. It consists of billions of neurons and their activity produces electrical fluctuations which are measurable on the scalp [1]. This is known as the electroencephalogram or EEG. The EEG is a complex multi-component signal and can be categorized into various frequency bands: Delta (<4 Hz), Theta (4–8 Hz), Alpha (8–13 Hz), Beta (13–30 Hz) and Gamma (>30 Hz), which in turn can be mapped broadly into different physiological states. These frequency bands can be used as biomarkers for different applications such as in sleep (delta band) and in chronic pain (alpha band)[2]–[8].

Brain activity can not only be monitored using modalities like the EEG, but it can also be influenced via neuromodulation, which is the intentional interference of brain activity via external stimulus such as light, sound and current [9]. Although most applications focus on using electrical modalities for neuromodulation, alternative modalities such as light and sound based neuromodulation exist [6], [8], [10]. Two application examples using light and sound neuromodulation are *sleep engineering* and *chronic pain* [2], [6], [7], [11]–[13]. Sleep engineering involves using sounds during sleep to enhance memory consolidation; this is done via delivering auditory tones at specific stages of sleep to enhance certain brainwaves known as slow oscillations, which are linked to memory consolidation (further details are given in Section 2.6.2). On the other hand, for chronic pain, recent work has used light and sounds to reduce the perception of pain [6], [7], [10], [11], [14]. It has been shown that increasing EEG activity in the alpha band (8–13 Hz) maps to reduced pain perception. Hence, groups are using lights and sounds tuned at the alpha frequency to enhance alpha power via brain wave entrainment, which in turn results in reduced pain perception [6]–[8], [10], [11]. Using lights and sounds for neuromodulation offers new possibilities for out-of-the-lab neuromodulation, as light and sound stimulation can be built into portable smartphone based platforms, which many people already have access to.

Despite such promising applications of neuromodulation, the technology is still at an early stage of development, and mainly restricted to laboratory based tests in controlled environments [2]–[8], [11], [15]. While suitable for fundamental discovery science studies, the potential for translation into real-world use is not yet built into the technological platforms avaiable for neuromodulation. In addition, most neuromodulation is performed *open-loop*, where the stimulation parameters are pre-set and fixed. However, each individual is different and would respond differently to the stimulation applied, and a ‘one-size-fits-all’ is not the ideal solutions. For example, for the chronic pain application driver, applying stimulation at the Individual Alpha frequency (IAF) as opposed to a pre-set value within the alpha band (e.g. 10 Hz), is hypothesized to result in a greater enhancement of alpha power, which in turn may lead to better therapeutic outcomes [8], [16], [17]. However, no such technology exists to provide IAF matched audio-visual stimulation which would help validate such hypotheses, as evidenced in Section 2.6.3. The inability to provide personalised, time-varying, brain state dependent, *closed-loop* neuromodulation is hypothesised as being one of the main causes of the large variability in results observed from neuromodulation experiments [8], [10], [18], [19].

Moreover, having a closed-loop neuromodulation technology running on a smart-phone would enable ‘out-of-the-clinic’ experiments, which not only would be easier for participants to part-take in, in the comfort of their own homes, but would also enable larger scale studies at reduced costs. For the sleep engineering application, various groups have explored technical platforms for closing the loop using EEG phase [2]–[5], [12], [13], [20]. Here, most platforms are lab-based and of the ones that are portable, run on custom embedded hardware [12], [13]. These limit use-cases and do not enable truly *in-the-wild* experiments; a better approach would be a platform that runs on an ubiquitous device such as a mobile phone.

With these in mind, this thesis focuses on creating a multi-purpose light and sound based closed-loop brain stimulation platform, that runs on a phone. It is capable of delivering phase-locked auditory stimulation, and frequency-matched audio-visual stimulation, for applications in sleep and in chronic pain. This thesis provides the first on-phone implementation of such approaches for ‘out-of-the-clinic’ experiments on a ubiquitous device such as the smart-phone. It provides the first of its kind platform to allow future clinical work to answer basic science research questions such as whether IAF matched stimulation does improve therapeutic outcomes compared to open-loop stimulation.

1.2 Thesis structure and contributions

This thesis begins with a literature review, which is then followed by three technical Chapters which detail in series the making of the mobile closed-loop neuromodulation platform; we first start with the open-loop stimulation platform in Chapter 3; we then conduct an offline analysis for the pre-requisites required for closed-loop operation in Chapter 4; and finally, we characterize the complete system on a phone in Chapter 5.

Chapter 2 - Closed-loop neuromodulation: What, Why and How?

This Chapter reviews the literature and gives an overall background and overview for the design of the mobile closed-loop system. The closed-loop system consists of a brain sensing component, and an actuating component, along with the use of appropriate feedback measures extracted from the EEG to close the loop. Hence, the Chapter starts with a discussion on various sensing modalities available covering invasive, semi-invasive and non-invasive and motivates our choice of non-invasive sensing, in particular the EEG. Then, it moves on to brain actuation modalities covering visual, auditory and electromagnetic stimuli, after which a broad discussion on closed-loop neuromodulation platforms available today is presented. Here, the major motivations of closing the loop is also presented, such as reducing variability in results observed using open-loop stimulation. Moreover, challenges and constraints to closing the loop are discussed such as: the limited amount of data to work with; the limited processing power on a phone and the need for low latency signal processing algorithms that adapt to non-stationary signals like the EEG. Following this, we discuss briefly the need for mobile platforms, which is then followed by the *how* of closing the loop, focusing on using EEG phase and frequency feedback measure, both of which were chosen from our target application drivers (chronic pain and sleep engineering). Then, for real-time extraction of these features, a review of the state-of-the-art algorithms is laid out, motivating our proposed algorithm, the Phase Locked Loop (PLL). Finally an outline of the clinical application drivers is laid out, alongside the state-of-the art closed-loop platforms available for these, which then eventually contextualizes our technology contributions, in light of the gaps present.

Chapter 3 - Open-loop light and sound. based neuromodulation on a phone

This Chapter details the implementation of an open-loop audio-visual stimulation app running on a phone, which as the first step in designing the closed-loop platform. We start the chapter exploring three visual stimulation methods on a phone, to overcome the limited 60 Hz phone refresh rate constraint found on the majority of current smart-phones; these are the square interpolation method, the square-sine method and the sine method. From the analysis, we show why the square-interpolation method is well suited for open-loop applications, but not for closed-loop use-cases, where frequency is expected to change. For the latter, we show why the sine method (a simple sinusoidal waveform) is best, as it ensures signal continuity at frequency changes via the use of a phase accumulator. We also discuss here why the square-sine method would be the best choice for future phones with higher refresh rates of 90 Hz and 120 Hz. Next, we detail the implementation of audio stimulation on a phone, using *binaural beats*, and how the addition of a phase accumulator ensures signal continuity at frequency changes, for it to be usable in the closed-loop platform later on in Chapter 5. We then end the Chapter briefly highlighting how this open-loop app is currently being used by our clinical collaborators as the first on-phone neuromodulation platform.

Chapter 4 - Pre-requisites to closing the loop: phase-frequency extraction

Following the characterization of the open-loop audio-visual stimulation in Chapter 3, we present a new investigation into the pre-requisites required before the complete on-phone closed-loop can be created. This mainly consists of understanding the dynamics of the EEG signal in the alpha and delta bands allowing tuning and characterizing the accuracy of the PLL in extracting the feedback measures (phase and frequency) required, to close the loop between sensing and actuation. For this, we first start with characterizing the EEG signal for its amplitude, frequency and frequency rate of change; the first two of which feeds into the PLL tuning methodology and the last to motivate closing the loop on a sample-by-sample basis. Here, we show for the first time that the frequency varies on a sample by sample basis at a rate of 9.75±2.67 Hz/s. This is followed by detailing the PLL tuning methodology. The phase-frequency extraction accuracy is characterized via three separate analyses based on the operation mode: Continuous phase locked stimulation (general use), Discrete phase-locked stimulation (sleep engineering) and continuous frequency matched stimulation (chronic pain). For the continuous phase-locked stimulation mode, the accuracy is characterized across all EEG bands for use in a wider range of applications and shows approximately 14% better performance than the previous state-of-the-art. Further-more, is shown that the inter-channel and inter-subject variance is low, negating any need for channel or subject specific tuning. We then characterize the discrete phase-locked stimulation mode for the sleep engineering application driver, obtaining a phase error of 11.2±11.2 degrees comparable to other lab-based systems, but now on a smart-phone platfomr. Finally, we characterize accuracy for the continuous frequency matched stimulation mode in Section 4.6; here, a frequency matching error of 0.004±0.15 Hz was obtained.

Chapter 5 - Closed-loop light and sound based neuromodulation on a phone

With the stimulation platform in Chapter 3 and the offline analyses in Chapter 4, we then finally combine all these together in this Chapter, for the complete on-phone implementation of the closed loop platform. Hence, this Chapter details the on-phone implementation of the complete system for the two application drivers: sleep engineering and chronic pain. To characterize performance, we first detail a test phantom based setup, which allows for more controlled and faster technology validation, without any on-person tests. With this, we first detail the performance of the platform for the sleep engineering application driver, which delivered phase-locked auditory stimulation. Here, an overall phase targeting error of 12±60 degrees was reported, alongside an over-all loop-latency of 70±24 ms, both of which were comparable to other lab-based and portable platforms available. Here, our contribution was providing the first on-phone implementation of the sleep engineering platform, enabling out-of-lab research on a more ubiquitous mobile platform such as the smartphone. We then moved on to demonstrate the performance for the chronic pain platform, which provided frequency matched audio-visual stimulation. For this, a frequency matching error of -0.17±1.29 Hz and 0.20±1.3 Hz error for the visual and auditory modes, respectively, was obtained. Here, the contribution is in providing the first of its kind platform for a sample-by-sample frequency-matched audio-visual stimulation. This will enable future clinical studies on personalised, time-varying, brain state dependent, *closed-loop* neuromodulation which have not previously been possible.

1.3 Writing style

In this thesis, the pronoun ‘we’ is used throughout to write in first person, in the style of a journal paper; hence, unless otherwise explicitly stated, ‘we’ refers to work I have done myself.

Chapter 2

Closed-loop neuromodulation: What, Why, and How?

2.1 Introduction

In this Chapter, the overall background and motivation for building a portable closed-loop neuromodulation platform is outlined. Any such platform will consists of a brain sensing and actuating component, along with the use of feedback measures to close the loop with. Hence, the Chapter begins with a discussion on various sensing modalities available today in Section 2.2. This is then followed by a discussion on brain stimulation technologies covering visual, auditory and electromagnetic stimuli, in Section 2.3. This is then followed by a broad discussion on closed-loop neuromodulation and the challenges of building such a platform in Section 2.4, with a particular emphasis on building a portable platform to enable 'out-of-the-clinic' applications. This is then followed by a discussion on the 'how' of closing the loop in Section 2.5, focusing on the phase and frequency as feedback measures, both chosen from our application drivers. Then, state-of-the-art algorithms for extracting these in real-time are reviewed and after summarizing its limitations, we propose the Phase Locked Loop as the chosen algorithm. This then leads on to the clinical drivers for this research: sleep engineering and chronic pain in Section 2.6, where we discuss the clinical need and the state-of-the-art platforms available for these. This then contextualizes our technology contribution, in light of the gaps present.

2.2 Sensing the brain

2.2.1 Introduction

The human brain is one of the most complex organs in the human body and is made up of billions of neurons which dictates its various activity. A single neuron, when it receives sufficient input from other neurons, fires and produces action potentials or spikes; this is a result of an increase and decrease in membrane potentials, due to an influx and outflux of sodium and potasium ions, respectively [1]. These fluctuations in electrical activity can be measured on a populational level both invasively and non-invasively, and is the basis of sensing brain activity. In this Section, we discuss various electrical sensing modalities for brain activity measurements, categorizing them as: invasive (Section 2.2.2) and non-invasive (Section 2.2.3) sensing.

2.2.2 Invasive sensing

Invasive measurements often include surgical procedures to place the electrodes inside the brain. One can either take single neuronal measurements using intra-cellular or extra-cellular recordings. The former measures action potentials by measuring voltage across the cell-membrane, while the latter takes the measurements in close-proximity to the cell-body, without penetrating the cell[1]. Both these are limited to tracking activity of single neurons to provide high temporal resolution but at the expense of missing the bigger picture; single neuronal activity do not give much insight into brain functions which are a result of complex interactions neuronal networks. On the other hand, recordings from multiple neurons could be taken using technologies like tetrodes and multi-electrode arrays. The former usually involves four wires tightly bound together that can track up to 20 neurons [1]. The latter involves arrays of electrodes placed over a brain region and is capable of tracking larger number of neurons [1]. In contrast to implants inside the brain, there are also semi-invasive modalities like Electrocorticography (ECoG), which uses electrodes placed on the surface of the brain, for sensing electrical activity [1]. This is used in applications like intractable epilepsy, where it is used to localize diseased regions of the brain before removing them. The ECoG still requires an invasive procedure to place the electrodes but it is capable of measuring a broader region of the brain, in contrast to the more localized implants [1].

The main advantage of invasive recordings are that it provides high temporal resolution and higher Signal to Noise Ratio (SNR), when compared with its non-invasive counterparts [1]. This however comes at the cost of an expensive and risky surgical procedure, with additional bio-compatibility issues between the electrode and brain tissue interface [1]. For example, when multi-electrode arrays are left in the brain for long, tissue scars could form around it which would cause a loss in SNR over time [1].

2.2.3 Non-invasive sensing

One of the older and most widely used non-invasive technologies is the Electroencephalogram (EEG). These are electrical activity measured via electrodes placed at the scalp; a consumer grade set-up is depicted in Fig. 2.1. The electrodes are usually placed on the scalp as per the international 10-20 system (Figure 2.2), which is a standardised system recognized internationally with an equal inter-electrode spacing. EEGs are a result of the summation of post-synaptic potentials from a large group of neurons oriented radially to the scalp. They predominantly capture electrical activity from the cerebral cortex and not deep brain regions [1].

EEGs are usually detected using electrodes placed on the scalp. The weak signals detected

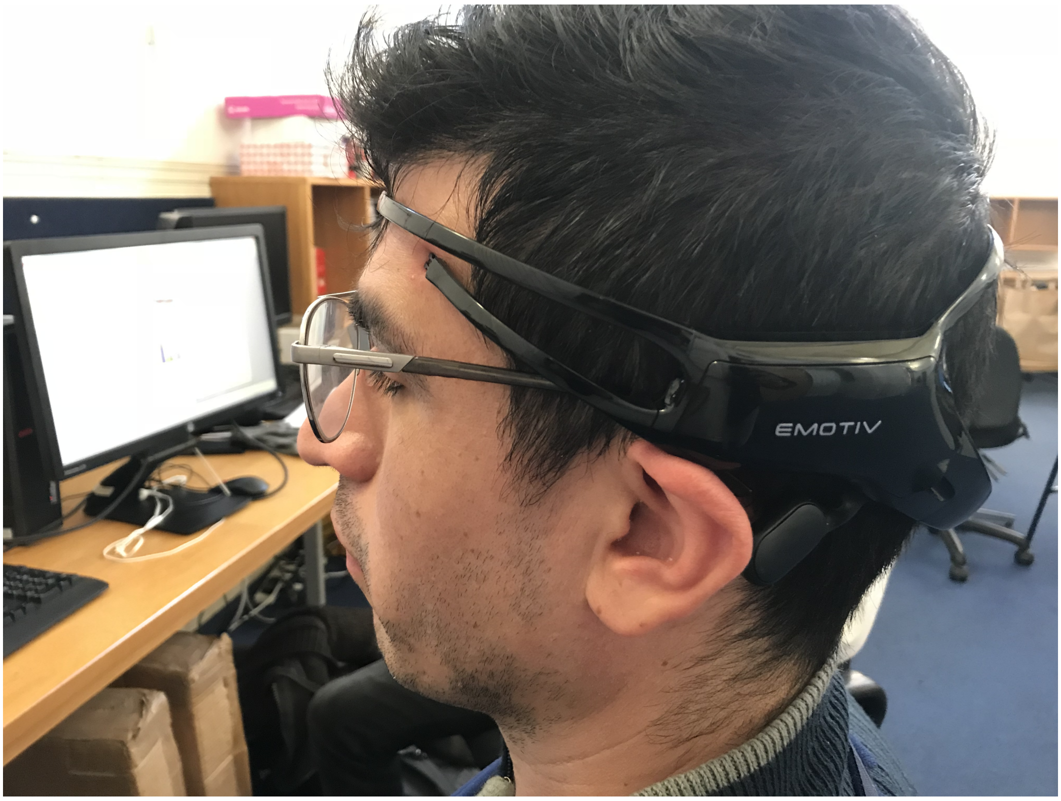

Figure 2.1. A consumer grade EEG unit: Emotiv Epoch.

(in the 10’s of microvolts range) are amplified 1000-100,000 times, after which various forms of filtering are done to remove corruptions due to noise; noise sources included the 50 Hz power-line interference, electrical activity due to eye-blinks and muscle movements, all of which mix with the signals originating from the brain [1].

Moreover, the synchronous firing of large populations of neurons produce oscillatory EEG which can be categorized based on their frequency content as: Delta (< 4Hz), Theta (4–8 Hz), Alpha(8–13 Hz), Beta(13–30 Hz) and Gamma (>30 Hz) bands, which in turn can be broadly mapped to different physiological states [21]. For example, high alpha power is often associated to a wakeful relaxed state and high beta power is often associated to a alert state [1].

The EEG has a very good temporal resolution in the millisecond range but a poor spatial resolution (6–8 cm) [22]. One reason for this is the volume conduction related distortion of the signals as it travels from the brain, through the skull and all the way to the scalp [1]. Moreover, the mixing in with artefacts such as those arising from eye-blinks and muscle movements, worsens this. Despite these, EEGs are relatively cheap (£5–£10k for research grade units), portable and hence, is an attractive choice for mobile applications.

Other non-invasive sensing technologies include the Magnetoencephalogram (MEG). Here, unlike the EEG, the MEG measures the magnetic fields induced due to the currents originating in the brain, dictated by Maxwell’s equations. These are measured using Superconducting Quantum Interference Devices (SQUIDS) which are bulky expensive machines [1]. However, MEGs do provide better spatial resolution (few mm) than the EEG (few cm) as the magnetic fields are not distorted by the skull or scalp, as is done for the electric fields

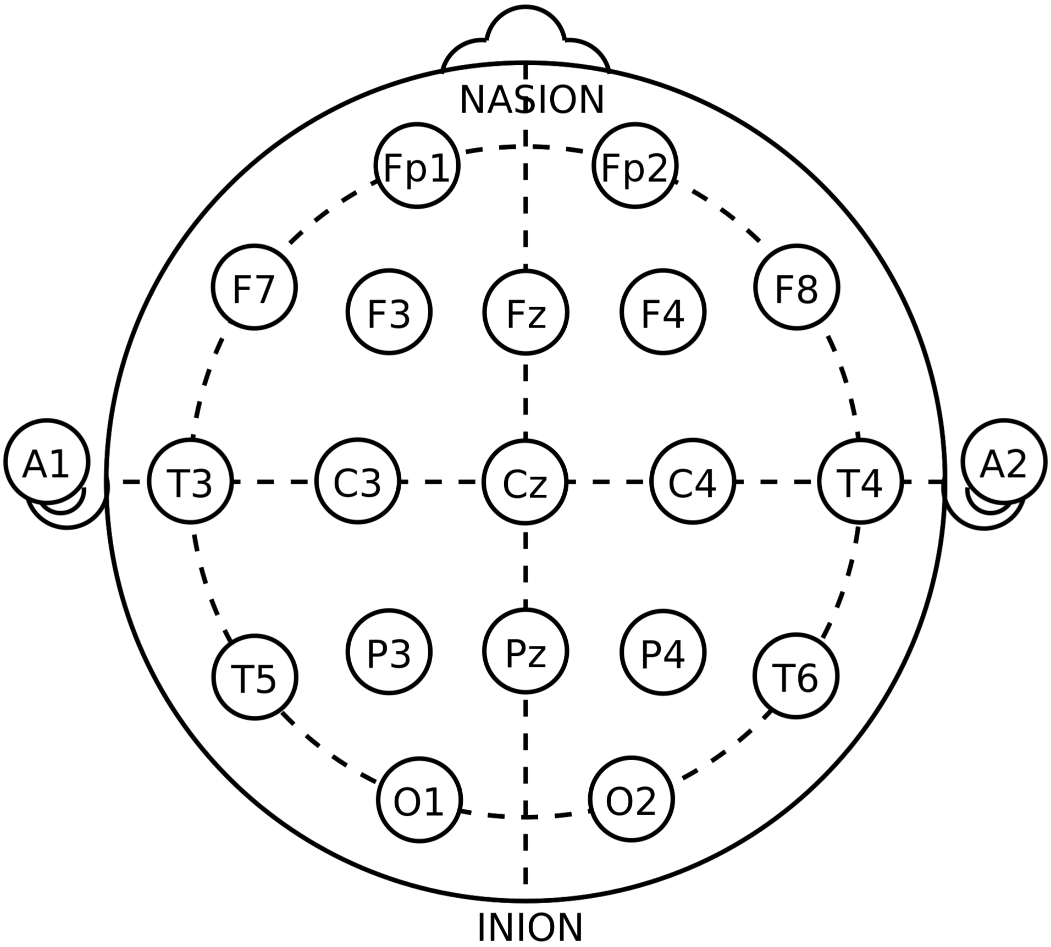

Figure 2.2. Electrode locations as per the international 10-20 system. Obtain from [23], Public Domain.

generated by the EEG [1]. Moreover, like the EEG, it has high temporal resolution too, in the millisecond range. Other than MEG, indirect measurements of brain activity could be taken using technologies like the functional Magnetic Resonance Imaging (fMRI) which detects brain activity by measuring changes in blood-flow. These have low temporal resolutions (1–4 seconds), but a high spatial resolution (few mm) [1]. However, like the MEG, they require bulky and expensive equipment for operation. A close cousin of fMRI would be functional near-infrared spectroscopy (fNIRS), which also tracks brain activity indirectly via monitoring changes in blood flow. However, unlike fMRI, it is portable and fit for mobile platforms with a higher temporal resolution (10 ms), despite a lower spatial resolution (3 cm) [1].

2.2.4 Modality of choice

Invasive sensing is risky, expensive and involves bio-compatibility issues between the tissue-electrode interface and hence is less suitable for out of the lab settings [1]. On the other hand, non-invasive techniques like the MEG and fMRI are bulky and expensive machines, and hence, is again ill-suited for mobile applications [1]. fNIRS is fit for portable use, and has a slightly lower temporal resolution and higher spatial resolution than the EEG; however, they are indirect measurements of the brain's electrical activity and are relatively newer than the EEG. Given these and the need for a modality which can be used for out-of-the clinic applications, the choice then boils down to the EEG: it is portable, offers high temporal resolution, all of which are important in the design of a high-fidelity mobile closed-loop system. More importantly, both our application drivers use EEG as the bio-marker as detailed in Section 2.6.3 and 2.6.2; hence, we used EEG for our work.

2.2.5 Summary

In this Section, we discussed technologies to sense brain activity, categorizing them as invasive and non-invasive. After discussing the various limitations of invasive modalities and non-invasive ones like fMRI and MEG, we then narrowed down our choice to the EEG. These are most suitable for *out-of-the lab use* due to its form factor and affordability.

2.3 Actuating the brain

2.3.1 Introduction

Having discussed sensing modalities available, the next step is to discuss technologies to actuate the brain, also known as neuromodulation. Neuromodulation is the intentional interference of brain activity via external stimulus such as lights, sound and current [9]. Evidence of its usage can be traced all the way back to the ancient Greeks, an example being the use of electrically charged torpedo fishes in clinical foot-baths to treat headache [24]. From using uncontrollable torpedo fishes as agents for electrical stimulation, the field has advanced significantly with a wider range of modalities to actuate the brain. For example, rhythmic lights and sound stimuli have been used to entrain brain activity with use-cases such as pain-relief [10]. It must be noted here that, for the purpose of this thesis, the term neuromodulation and entrainment is used inter-changeably. Entrainment is a phenomena wherein the EEG dominant frequency synchronizes to that of the stimulus, via the frequency following effect. For example, in the chronic pain application driver, the rhythmic lights and sounds themselves are neuromodulatory signals intended to affect neural activity for pain-relief, via alpha entrainment [6], [8]. With this in mind, brain actuation modalities are outlined in this section, and they are categorized as: electromagnetic (Section 2.3.2), visual (Section 2.3.3) and auditory (Section 2.3.4).

2.3.2 Electromagnetic stimulation

Electromagnetic stimulation of the brain is the most direct form of neuromodulation and can directly influence neural activity, which are in itself electromagnetic in nature. This can be done via invasive or non-invasive technologies.

Invasive stimulation covers the likes of Deep Brain Stimulation (DBS) where electrodes are implanted deep inside the brain through complex surgical procedures, which then sends

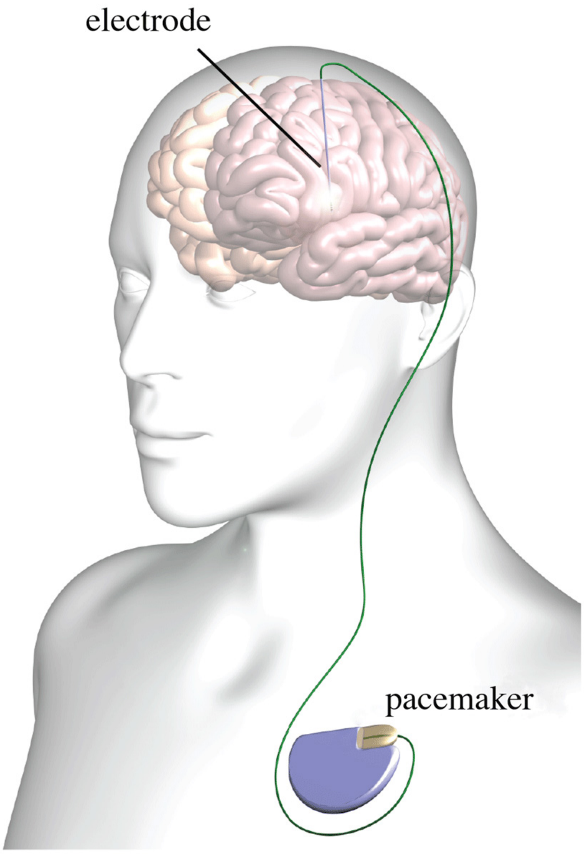

Figure 2.3. A typical DBS setup. Obtained from [37], CC BY 3.0 license.

electrical pulses to specific brain targets [19] [25] [26]. This can be thought of as a brain equivalent of a cardiac pacemaker, as shown in Fig. 2.3. DBS has been used widely for various therapeutic applications such as for Parkinson's and managing symptoms like tremor [27]–[30]. Other applications include Obsessive Compulsive Disorder (OCD), Chronic Pain and Treatment Resistant Depression [19], [25], [26], [31]–[36]. As obvious, the main advantages of this technology include focused and targeted stimulation of deep brain regions, which also translates to reduced side-effects due to less focal stimulation [9]. However, this does come at the cost of complex surgical procedures and higher risk, lending itself as an expensive technology [19]. A typical Deep Brain Stimulation (DBS) setup is shown in Fig. 2.3, where the electrodes are surgically implanted and the pacemaker provides the electrical pulses.

On the other side of the spectrum are non-invasive technologies, which involves stimulating from the scalp, as opposed to deep brain regions. Examples of this include magnetic based ones like Transcranial Magnetic Stimulation (TMS) and electrical based ones such as Transcranial Direct Current Stimulation (tDCS), Transcranial Alternating Current Stimulation (tACS), and Transcranial Random-Noise Stimulation (tRNS). The electrical based stimulation modalities all come under the umbrella of Transcranial Electrical Stimulation (TES).

Firstly TMS, the more established technology, involves the use of strong time-varying electromagnetic fields that are applied using coils connected to a magnetic stimulator, that is placed over the scalp [38]–[40]. Placed over the head-region where stimulation is desired, the magnetic field then passes through the skull and all the way to the brain, and induces

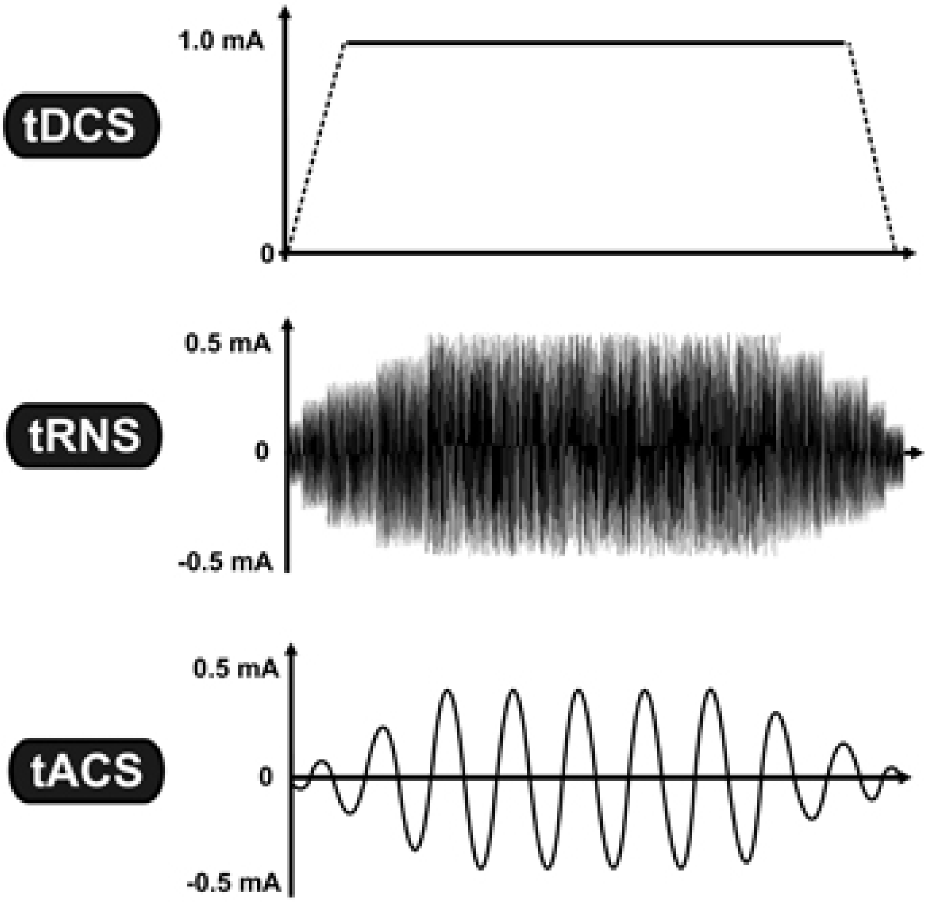

Figure 2.4. Typical tACS, tDCS and tRNS waveforms. Obtained from [41], CC BY 3.0 license.

electrical potentials [38], [39]. TMS can come in various forms such as repetitive TMS (rTMS) which as its name implies uses multiple pulses repeated at a specific frequency or it can also be delivered via single pulses. Advantages of TMS include the ability to quickly change stimulation sites since electrode placements are not necessary, while drawbacks include its bulkiness and higher cost, compared to counterparts like tDCS and tACS. [38].

Secondly, TES involve the use of current delivered via electrodes placed on the scalp. tDCS passes constant current from the skull to brain, with intensities typically in the range 0.5–2mA [42]. It is capable of affecting different brain functions such as motor, sensory and cognitive functions [39]. tACS on the other hand, uses time-varying alternating current to stimulate the brain, with typical intensities ranging from 0.25–1mA [39]. Compared to tDCS for which only the intensity of stimulation can be tuned, tACS enables the control of frequency and phase of stimulation, which is more suited for exploratory work, as brain waves are themselves oscillatory in nature [43], [44]. These stimulation parameters have influence on both the direction and duration of stimulation induced effects [43]. tACS is often applied at frequencies ranging from DC to 5kHZ, and enables controlled interference with ongoing brain rhythms when applied at frequencies in the EEG range (0.1-80 Hz) [43]. Example applications include: tACS at 0.75 Hz to enhance declarative memory consolidation and phase-locked tACS in the theta band to improve cognitive performance[43]. Moreover, side effects are also frequency dependent such as the induction of phosphenes which is the phenomena of seeing light without light actually entering the eye; for example, the varied geometrical shapes one sees after having rubbed their eyes while it is closed. These could also be induced by tACS with frequencies anywhere up to 80 Hz [43]. Other less common forms of TES include tRNS, which applies a random noise like current, with a broad frequency band ranging from 0.1–640 Hz. These are not ideal for closed-loop applications due to their broadband natures; there is no room to control parameters such as phase and frequency to close the loop with tRNS. Typical tACS, tDCS and tRNS waveforms are depicted in Fig. 2.4

2.3.3 Visual stimulation

Visual stimuli can also be used as a neuromodulatory signal. One is bombarded by visual stimulus all through their waking hours and these everyday stimuli can also be tuned and presented in a rhythmic fashion, to modulate neural activity via entrainment.

The purpose of entrainment is to synchronize brain activity to the frequency of an external stimulus, in this case visual. This is built upon a well observed phenomena known as the *frequency following effect*, which basically states that brain-wave frequencies synchronize to the frequency of a repetitive external stimulus, such as flickering light [45]. This occurs when the stimuli is strong enough to excite the thalamus and is then passed to the relevant processing areas in the brain, such as the visual cortex, where entrainment is observed [46]. At any one point in time, a user's brainwaves will be comprised of different frequency content and entrainment drives the dominant brain wave frequency to that of the stimulus frequency, in addition to phase-locking it to the stimulus frequency[21].

For the case of visual or photic entrainment, the resulting synchronized brainwaves are also known as Steady State Visual Evoked Potentials (SSVEP) [47]. SSVEPs can be obtained by even weak stimulation intensities such as those from monitor refresh-rates of up to 75 Hz [47]. Irritation due to the constant flickering of light is often reported when people use the lower frequency SSVEPs; this has fed interest in high frequency SSVEPs where the flicker is not perceivable anymore, with a trade-off of weaker SSVEP's compared to the lower frequency bands [47]. Moreover, recent research has shown SSVEPs to be obtainable from 'non-hair' regions such as the ear and forehead, which is useful as this provides superior electrode contact quality and requires less set-up time [48]. It has also been suggested that entrainment personalized to a user is more effective: for example, alpha entrainment at the Individual Alpha Frequency(IAF) has shown to result in stronger entrainment, than a one-size-fits all approach [16], [17], [49].

Apart from being widely used in SSVEP based Brain Computer Interfaces, photic entrainment have also found therapeutic use-cases, such as that of reducing pain-perception for chronic pain [6]–[8], [10]. Here, it has been shown that increased alpha power via photic alpha entrainment, to be capable of reducing pain perception [6]–[8], [10]. This is discussed in more detail in Section 2.6.2 [10].

2.3.4 Auditory stimulation

Apart from visual neuromodulatory signals, auditory ones could be used as well. For example, for brain-wave entrainment, one can do this via binaural beats, which are sinusoidal waves of 2 different frequencies, presented separately to each ear [46]. As a result, the dominant frequency of the EEG synchronizes to the frequency difference between the two tones, also known as the beat frequency [46], [50]–[53]. For example, if a 114 Hz tone is presented to the right ear and 124 Hz to the left, the beat frequency will be 10 Hz which is in the alpha range, as depicted in Fig. 2.5. The beat frequency must be less than 30 Hz for effect to take place and the carrier frequencies in the 200-900Hz range, with maximal probability of detecting binaural beats at 500 Hz [46].

Binaural beats has found a variety of use-cases; anxiety reduction, inducing sleep quicker, improving attention spans, memory, pain relief and creativity [10], [50]–[57]. Pain relief has been one of the late interesting use-cases, where they showed binaural beats at the alpha frequency caused significant reduction in pain perception, though not as strong an effect as visual entrainment [10]. Other applications include; [51] where on-the-spot 14 Hz beats were used to reduce mind-wandering amongst university students; [57] where beat frequencies that ramped down from 8-1 Hz over time was used to induce sleep quicker; [58] who showed 20 Hz beats to improve long term memory; [56] where 40 Hz beats was used to improve short term memory and studies using stimulation at the lower bands like alpha for anxiety and stress [53], [55]. It must also be noted here that although most applications use pure binaural beats, there have been cases where these tones were mixed with external stimuli such as music or Autonomous sensory meridian response (ASMR) sounds, to improve user-experience and reduce psychological irritation caused by these repetetive buzzing sound. ASMR is a perceptual experience, often in the form of a tingling sensation that starts from the scalp and moves down the back of the neck, in response to sensory stimuli [50]. For example, [50] masked binaural beats with ASMR sounds, to induce sleep quicker.

Despite all these promising applications, binaural beats are not without its critics; its efficacy in entraining the brain has been much debated and the variance in results, has not helped in reaching a common concensus [46], [60]. For example, Gao et al. found no significant entrainment for the delta, theta, alpha and beta frequency bands but instead found that entraining one band had an effect on the other [46]. For example, theta band binaural beats decreased relative power in the beta band and vice versa [46]. However, they also observed entrainment induced brain connectivity changes. More specifically, they showed that under delta, alpha and theta beats, theta band functional connectivity changes between posterior and anterior areas [46]. Similarly, [52] showed that binaural beats at both the alpha and gamma frequencies improve creativity and divergent thinking, but only for individuals with lower dopamine levels, measured indirectly by eye-blink rates (EBR). For individuals with high dopamine levels, divergent thinking was either unaffected or imparied in some cases as well [52].

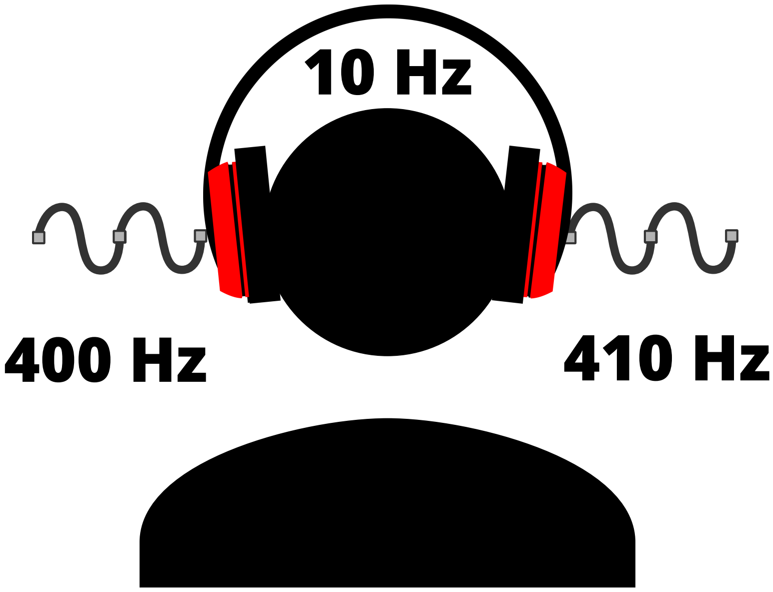

Figure 2.5. Auditory entrainment using binaural beats [59].

Such variability in results is not just common to binaural beats, but is observed in general with open-loop stimulation paradigms. This is one of the major motivators for brain-state dependent stimulation, which is hypothesized to reduce such variability in outcomes; these are discussed in detail in Section 2.4.3.

2.3.5 Modality of choice

For actuating the brain, the modality of choice is primarily driven by the application drivers: audio for sleep engineering and audio-visual for chronic pain. For both these applications, there are collaborators who have expressed interest in using the technology being developed for clinical work and these applications are discussed in more detail in Section 2.6. For this reason, we do not consider electromagnetic modalities.

2.3.6 Summary

In this Section, we discussed various modalities available to actuate the brain, categorizing them as: electromagnetic, visual and auditory. We outlined applications of each and noted that despite most research focusing on electromagnetic modalities, light and sound based are gaining popularity and have found various use-cases such as our application drivers: sleep engineering and chronic pain. Hence, the actuation modality of choice was audio and visual, both driven by our application drivers. Next, we move on to closing the loop between sensing and actuating the brain.

2.4 Closing the loop between sensing and actuating

2.4.1 Introduction

In the previous sections, both brain sensing and brain actuating (neuromodulation) modalities available have been presented, and the modality of choice was narrowed down to EEG for sensing and audio-visual for actuating, both driven primarily by our application drivers. The next step is to close the loop between brain sensing and actuating. To do this, we first lay out a definition of a closed-loop system in Section 2.4.2, after which we lay out the rationale for closed-loop stimulation in Section 2.4.3 and its general challenges in Section 2.4.4. We then end the Section discussing the need for mobile platforms, and the challenges of building one in Section 2.4.5. Note that this section is a broad and general overview of closed-loop platforms; a more specific discussion of the state-of-the art for non-invasive light and sound based platforms, in light of our application drivers, is discussed in Section 2.6; this is where our technology contribution lies.

2.4.2 What is closed-loop neuromodulation?

Closed-loop neuromodulation is a data driven approach to stimulating the brain, where stimulation is adjusted based on ongoing brain activity from one or more brain regions of the user [21], [61], [62]. An analogy could be made with the popular cardiac pacemakers where ongoing heart rhythms are continuously monitored and used to deliver pulses of current to restore abnormal heart rhythms. Replacing EEG sensors with the ECG sensors to measure brain rhythms instead of heart rhythms then gives us a cardiac pacemaker for the brain, where the goal is to modulate the brain state to a desired one [21]. However, one key difference is that unlike signals from the heart, ones from the brain are statistically more complex and non-stationary and how these brainwaves translates to meaningful physiological states are not that clear [21].

In its simplest form, a closed-loop system can be defined as a system that senses a brain-state of interest and then applies stimulation based on that. This is conceptualized in Fig. 2.6, where the acquired EEG data is processed using low latency signal processing techniques to extract biomarkers such as EEG phase and frequency, which represents an indicator of a brain-state. Once these features are extracted, stimulation is then applied accordingly; for example, phase-locked auditory stimulation has shown to improve memory consolidation [5], [12], [63], which is explained in more detail in Section 2.6.2. Next, the rationale for a closed-loop system is discussed in more detail.

2.4.3 Why close the loop?

Here, the rationale for building closed-loop neuromodulation platforms is discussed. To do so, the drawbacks of open-loop platforms are discussed first, followed by the benefits

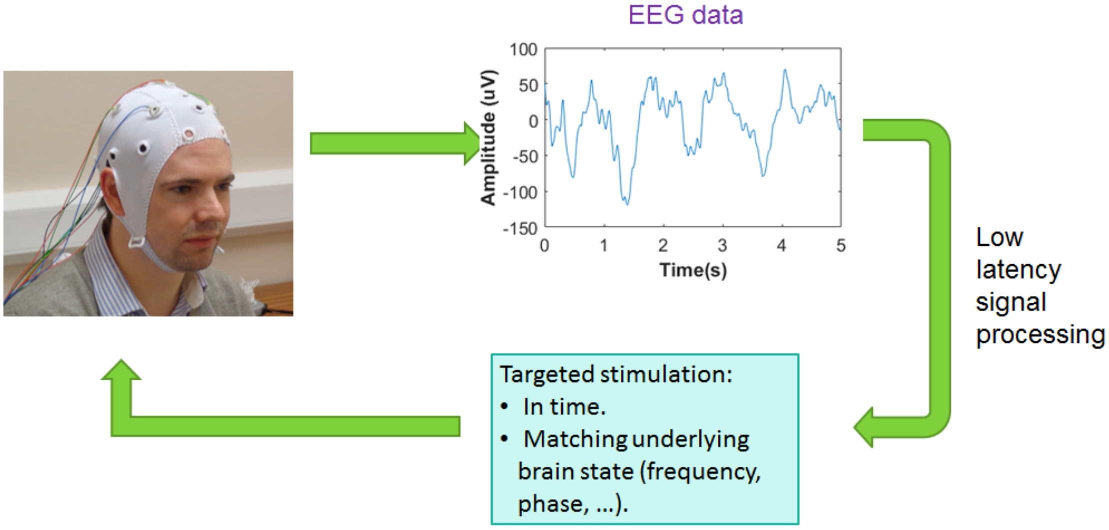
EEG data
Amplitude (uV)
100
50
0
-50
-100
-150
0
1
2
3
4
5
Time(s)
Low latency signal processing
Targeted stimulation:
• In time.
• Matching underlying brain state (frequency, phase, ...).

Figure 2.6. Components of a closed-loop neuromodulation platform.

of closed-loop systems, along with the challenges and constraints of doing so.

Drawbacks of open-loop stimulation

Traditionally, neuromodulation has been applied in an open-loop manner, where the stimulation parameters are preset. This has been used for a variety of applications involving both invasive neuromodulation such as chronic pain , treatment resistant depression, Obsessive Compulsive Disorder (OCD) [19], [25], [26], [31]–[36], [64] and non-invasive applications like depression and stroke, amongst others [18], [65]–[67].

Despite the various applications, drawbacks of this approach are many. Firstly, the stimulation parameters are usually set in stone and adjusted manually by the operator based on observed brain signals; this human in the loop approach can introduces many related errors, especially when the parameters are set by less experienced operators [64], [68]. Moreover, these are re-adjusted very infrequently, often months apart [61], [68]. Moreover, various side-effects of open-loop stimulation include impaired cognition, gait, speech, verbal fluency, decision making ability and build up of stimulation induced effects in non-target brain areas [69], [70]. All these drawbacks then drives the need for a more intelligent and adaptive stimulation protocol that adjusts to the user's need and stimulates only when necessary; this we hypothesise may potentially reduce these side-effects, due to more focal stimulation at only the brain region of interest.

Benefits of closed-loop stimulation

There are many hypothesized benefits to a closed-loop system. It could improve the scientific understanding of brain network dynamics through enabling more flexible experimental paradigms, improve therapeutical efficacies of current open-loop platforms and most importantly, reduce variability of results [61], [64], [71], [72]. Current studies are mostly hypothesis driven due to lack of available platforms; hence, the aim here is to develop these to start answering such questions.

On the experimental side of things, closed-loop platforms could accelerate the finding of relevant biomarkers [61], [64]. One example of this would be improving declarative memory consolidation via enhanced slow oscillations through auditory clicks played during specific phases of the slow oscillations during sleep [2]. For such applications, the optimal phase for maximal Slow Oscillation (SO) enhancement can only be found using a closed-loop platform capable of delivering phase-locked stimulation to the EEG. This would then empower such hypothesis driven experiments. A more detailed discussion for this is given in Section 2.6.1.

Therapeutical application also benefit from closed-loop platforms. For example, spinal chord electrical stimulation based on a subjects body position has improved pain relief; in treating epilepsy, more than 40 percent reduction in seizures was observed when electrical stimulation was applied in a closed-loop manner; for Parkinsons closed-loop stimulation yielded better therapeutical results than open-loop cases in both primates and humans and finally, modalities such as audio-visual stimulation has been investigated for the treatment of Attention Deficit Hyperactivity Disorder (ADHD) [68], [71]–[74]. With regards to the invasive side, more work has been done here, with use-cases such as: reducing seizure occurrences in epilepsy, reduced tremor for Parkinsons patients and to manage pain better for chronic pain patients [27]–[30], [75]–[79].

Moreover, it must be re-iterated here that most of the work, like the ones outlined previously, are electrical based and are invasive. These are discussed to show that closed-loop stimulation has shown improved therapeutic effect for these modalities, in the wider research area. This thesis does not aim to advance the state-of-the-art in electrical or invasive modalities; instead the focus is on portable light and sound based closed-loop neuromodulation platforms, and more specifically, portable platforms designed for the chronic pain and sleep engineering application drivers. The specific technology gaps and state-of-the-art closed-loop platforms for these two are discussed in more detail in Section 2.6, in the context of the clinical drivers. Moreover, once developed, these platforms would be used by collaborators working in this field as they have expressed interest, further confirming the clinical need. For these reasons, both electrical and invasive modalities are considered out of scope and they are just included here to cover the wider literature.

The *Variability* theme

Finally, one could argue that the most important benefit that closed-loop platforms could bring is an explanation or even better, a solution to the inter-intra subject variability in therapeutical outcomes, which is so often found in open-loop trials; not taking into account the brain-state of the user when stimulating is hypothesized to be one of the causes[18], [19], [40], [62], [68], [74], [80], [81]. Inconsistencies may be explained by three different factors; neuroanatomical, neurochemical and neurophysiological changes [40].

Firstly, variations in neuroanatomy exist both within and across subjects. These include different skull and cortical thicknesses across different subjects, in addition to variations in gyral shape and white matter structure [40]. Within individuals, these may change in a time scale of days to years and there exists some work which uses software solutions to take into account variations in head geometry [40]. Secondly, neurochemical changes may explain variability of results and examples of this include neurotransmitter availability and present hormone levels, which may vary on a time scale of minutes to years[40], [82]. Finally, neurophysiological changes such as endogenous oscillatory patterns and spectral fluctuations in these may also be used and this may be the most relevant, as these change on a shorter time-scale of seconds and hence is most relevant for designing individualized stimulation routines [40]. Example applications of this include: measuring ongoing phase changes in EEG and phase-locking TMS to the up-state (positive peak) to evoke larger motor potentials, and locking TMS pulses to to beta band power during task related de-synchronization (decrease in EEG power, in this case the beta band) for stroke rehabilitation [40], [62], [64], [74], [80]. Moreover, instead of adjusting stimulation timing, other parameters such as the intensity could be altered too.

The focus of this thesis would be on neurophysiology changes and not neuro-anatomical or neuro-chemical ones as the latter occurs in too long a time-scale and the more pressing concern would be adapting stimulation routines to neuro-physiological changes to see the various effects this could induce. Moreover, as discussed in Section 2.6, both our application drivers are based on closing the loop on the neurophysiological scale.

Next, the various challenges faced in designing a closed-loop neuromodulation platform are discussed.

2.4.4 Challenges to closing the loop

Firstly, a non-engineering challenge which is crucial is the identification of relevant EEG bio-markers that are clearly representative of a desired brain state state or are predictive of the intended therapeutical after-effects [21], [40], [64], [68], [74]. What is less clear in the wider literature is how these brain signals translate to a meaningful physiological or clinical state, which a solid biomarker should be able to do [21], [79]. These are not easy as the same EEG biomarkers could be observed by different pathological brain states; for example, alpha band as a biomarker is used in various applications like chronic pain and schizophrenia, whilst also being observed with high power, in normal adults when in a relaxed or eyes-closed state [10], [68], [83]. Hence, robust biomarkers should be able to clearly identify a brain-state relevant to the application [81]. It is also important that these are easily extractable from neuronal recordings, which is more of an engineering challenge [74]. For biomarker extraction, the need would be for efficient algorithms that could extract features like the phase and frequency in real-time. Note that phase and frequency are used as example biomarkers in the following text, as these are the relevant ones for our application drivers, which is discussed in more detail in Section 2.6.

Seondly, low latency algorithms are a necessity for most EEG based platforms, due to the fast time varying nature of the EEG [2], [44], [62], [84], [85]. Fast access to and processing of EEG data is required to provide stimulation at the desired brain-state on time and accurately; for example, a loop latency and jitter in the order of milliseconds is required for phase locked stimulation [86]. EEG phase changes much more rapidly compared to its spectral fluctuations and hence, the time-resolution requirements are much higher, especially for higher EEG bands [80]. To give an example, consider the delta band and within it, a 1 Hz oscillation; here, to get a phase resolution of 5 degrees i.e. 13.8 ms in time, an over-all latency lower than 13.8 ms is required. Similarly, for a 12 Hz alpha and 25 Hz beta oscillation, a 5 degrees resolution would require latencies lower than 1.2 ms and 0.5 ms, respectively. In light of this, there will be delays within the system. Firstly, delays could be introduced by the data-acquisition system: these could come from the electrodes, front-end amplifier filters, Analogue to Digital Converters (ADC) and the operating system of the data-acquisition system. Then, there would be the communications delay and then another operating system delay from that of the phone. For the communications delay, ideally a bio-signal recording device that gives non-buffered data in a few ms with a fixed delay, would be preferred [80], [87]. Finally, there would also be the delay due to signal processing such as phase shifts due to filtering; in a real-time scenario, no future data is available, hence, forward-backward or zero-phase filtering would not be possible, to compensate for these. When using IIR filters, the phase-shift is frequency dependent and is non-linear [87], [88], but the advantage is the filter order could be lower. For FIR filters, the phase-shift can be linear and hence, is easier to account for but requires very large order filters for good filtering [88], [89]. After signal processing, there would also be delays for the stimulation; for example, the delay for activating the sound card for auditory stimulation. Moreover, it must also be noted that the phone's operating system are not designed for time-critical applications since it is consumer oriented; neither Android nor IoS are Real Time Operating Systems (RTOS), where the execution of tasks can be guaranteed to be completed within a certain time-frame. Instead, the delays are task-dependent, and may vary over time. With these in mind, the need for computationally simple and low latency algorithms are clear, which when combined with the more unpredictable delays like those of the task-based operating systems on which the platform is running, would give the required phase resolutions needed, for the application at hand.

Thirdly, for real-time application, only data from the current time point and the past is available. Extracting features from a small window of data before the current time-point is challenging and typical off-line routines such as averaging over trials to enhance Event Related Potentials (ERPs) could not be done in this case [80]. Moreover, any window based methods, such as the use of band-pass filtering and Hilbert transform, would incur edge artefacts, which would require the most recent samples to be discarded,which in result would effect accuracy [89]–[93]. Hence, edge effects and minimal data loss is another challenge to be solved.

Moreover, EEG data is non-stationary and its properties like amplitude and dominant frequency changes both within subjects over time, and across subjects [44], [89], [90], [93]. Hence, algorithms that are sensitive and adapts to such variations over time will be required to account for the non-stationary nature of EEG signals.

Finally, artefact removal is another issue in the design of closed-loop platform; these artefacts may either be physiological such as those incurred due to eye-movements, blinks and motion or they could be ones due to the stimulus signal itself [74]. For the latter, it is more so common when using electromagnetic modalities: here, the stimulus signal's spectral content overlaps with that of the EEG signal and hence, the neural data becomes buried within the much stronger stimulus [74], [81]. To extract these features, real-time artefact removal algorithms need to be applied. This may involve classical filtering techniques such as filtering using band-pass, low-pass or Moving Average Filters. It may also be done using more effective but complex blind-source separation based algorithms like Independent Component Analysis(ICA) and Canonical Correlation Analysis (CCA), which are often used in off-line analysis to separate artefactual sources from neural ones. Real-time modification to these techniques exist such as the Online Recursive ICA (ORICA), which are available to use with the REST toolbox [94].

2.4.5 Mobile platforms

Why mobile platforms?

Building a platform that is mobile and portable is key to this thesis: almost everyone has a mobile phone in their pockets these days, and having a solution in the form of a mobile app like the one we create (see Figure 2.7) would allow the technology to scale, due to the ubiquitous nature of smartphones today. This would also be more suitable for 'out-of-the-clinic' applications, which would not be possible with the current lab-based platforms available today [2], [4], [5], [15], [63]. Hence, the rationale for developing mobile platforms comes mainly from the need to conduct experiments *in-the-wild* and to translate lab-based closed-loop therapies to real-world home-based therapies [85]. Moreover, these would also help improve the often reported problem of discomfort experienced by participants for long EEG based experiments, often conducted in tight controlled labs with limited movement [85]. This would be more so a concern for our chronic pain application driver, as chronic pain patients can get tired quickly, find it tiresome travelling for experiments and being out for too long [95]. A mobile platform may be a solution to this, as this enables the experiment to be conducted in the comforts of a user's home[85]. Also, mobile platforms will enable a wider range of experiments to be done which would not be possible in a lab based situation; an example of this would social neuroscience experiments where the complex interactions between multiple users need to be observed, at the same time [85]. Finally, with wait times for sleep labs being over a year, applications such as the sleep engineering one would highly benefit from home-based portable solutions [96]. This would enable larger scale studies to be conducted much faster and potentially for lower costs.

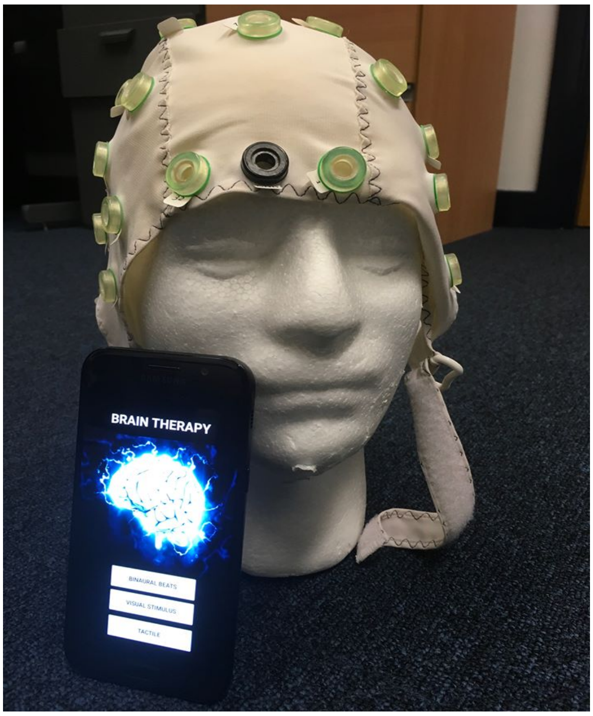

Figure 2.7. App based neurotherapy for out-of-the lab applications.

Challenges of building a mobile platform

The challenges of building a mobile closed-loop platform can be broadly categorized as: hardware, signal processing and stimulation related.

Firstly, challenges in the hardware domain need to be addressed. Typical EEG setups are wired and used for static experiments, where the subject is sat on a chair, in a tightly controlled lab environment [97]. For portability, a wireless device is preferred. For this, various consumer grade mobile EEG units exists, with different limitations. For example, a popular one is the Emotiv Epoch, but one of the issues here is that it uses wet electrodes, which do not give long lasting recordings, as it requires regular soaking in saline solution to maintain good quality recordings. Such limitations could be overcome using technologies such as dry-electrodes and in-ear, forehead or behind-ear EEG, where EEG signals could be acquired from non-hair regions [48] [98]. We do not choose to tackle problems in this domain, and instead use a commercially available EEG device: the smarting EEG amplifier provided via Mbraintrain to be specific, as detailed in Section 5.2 [99]. This is a wireless EEG amplifier which transmits EEG data directly to an Android phone via Bluetooth technology, making it a perfect fit for our smart-phone based platform.

Aside from hardware, the next challenge is in real-time extraction of the EEG features on a phone. One major constraint here is the latency and limited computational power available on a phone, compared to a PC. There will also be the wireless communications latency for signal transmission which would not be present in wired lab-based setups available today such as that of [12] and [13], both of which run on custom embedded hardware. With this in mind, it is then more so crucial to use algorithms that have minimal computational complexity like the Phase Locked Loop (see Section 2.5.4), which allows for operating on a sample-by-sample basis. Sample-by-sample signal processing would also help reduce latency down by avoiding any buffering induced delays, which otherwise would have been necessary with epoch-based methods like the Fast Fourier Transform based and Autoregressive modelling based techniques discussed in Section 2.5.4 [44].

Finally, regarding stimulation, providing audio and visual stimuli is also challenging on a phone. The low refresh rate of a smartphone (typically 60 Hz) limits the achievable frequency for visual stimulation and for auditory stimulation, one has to ensure audio is generated in real-time with no discontinuities at frequency changes. We detail our methods for tackling both these problems in Chapter 3.

2.4.6 Summary

In this Section, we first discussed the general definition of a closed-loop neuromodulation platform, after which we laid out the rationale for closing the loop. Here, we first explored the limitations of open-loop stimulation such as its 'hit-and-miss' approach, after which we highlighted the potential of closing the loop; the major one here would be addressing the variability in outcomes observed, with open-loop paradigms. We then discussed the general challenges of closing the loop such as the need for: low-latency algorithms, robust biomarkers and accurately extracting features from a limited amount of data. The section then ended with a discussion on the need for mobile platforms, and the challenges of developing on a mobile device. Having laid out a general overview of a closed-loop system, we next move onto the *how* of closing the loop.

2.5 How to close the loop?

2.5.1 Introduction

In this Section, we discuss the *how* of closing the loop and we focus on two different biomarkers: phase and frequency. We start first discussing the various applications and use-cases for these two biomarkers, in Sections 2.5.3 and 2.5.2. We then move onto discussing state-of-the-art algorithms in Section 2.5.4 for phase and frequency extraction, in real-time. We then conclude detailing our proposed algorithm the Phase Locked Loop (PLL), in Section 2.5.5.

2.5.2 Phase

One way to close the loop would be to use the phase of the brain oscillations as a potential bio-marker. Various studies have explored this: for example, Long Term Potentiation (strengthening of synaptic connections) and Long Term Depression (weakening of synaptic connections) could be induced by stimulating at the peaks and troughs of the hippocampal theta oscillation, respectively; and information about various memory processes in the brain could be extracted from EEG phase [21], [62], [91]. Phase also serves as a useful measure for other markers such as Phase-to-Amplitude Cross-Frequency Coupling (PA-CFC), where phase of slower EEG bands (theta, alpha) modulates the amplitude of the faster bands (beta, gamma) [100], [101]. This has found relevance as biomarkers for different cognitive and motor related impairments; for example, the PA-CFC has been shown to be higher in people with Parkinsons [102].

Many of the closed-loop studies using phase as a biomarker, have been done in the context of electromagnetic stimulation [103]–[105]. For example, TMS applied to the prefrontal cortex that is phase locked to the negative theta phase (negative peak) was shown to improve induced neuro-plasticity [104]. Alternatively, applying TMS at the motor cortex and measuring the resulting Motor Evoked Potentials is a very common clinical and diagnostic tool to measure cortico-spinal excitability [103], [105]. In this context, TMS phase-locked to the negative peak of the sensorimotor mu rhythm (7–11 Hz) has shown to evoke larger MEPs and hence, increased corticospinal excitability, when compared to open-loop stimulation [105]. Similarly, entraining beta oscillations via tACS and applying TMS pulses at various phases of the tACS signal has shown phase-dependent modulation of cortico-spinal excitability, as measured by the MEPs [103].

With regards to audio-visual stimulation, closed-loop studies using phase has been explored in fewer application areas. One example application is the use of EEG phase in optimizing temporal attention [106]–[109]. Visual stimulus presented to us in the real world are not always perceived; one explanation of this, as proposed by recent studies is that temporal attention and the ability to perceive a visual stimuli successfully, fluctuates with the EEG and is phase-dependent [108], [109]. For example many studies have shown perception to peak at the positive phase of alpha oscillations [107], [110], [111]. Consequently, visual targets presented in-phase with these rhythms are more likely to be perceived. One could then envision a closed-loop platform capable of measuring alpha-phase in real-time and presenting visual targets aligned with the positive phase of the alpha rhythm to maximize the likelihood of the target being perceived. Another approach would be to simply entrain the brain via an periodic stimuli like tACS or visual flicker to phase-lock EEG activity, and then provide visual stimuli at the desired phases to optimize perception [108]. Another emerging field for phase-locked stimulation is that of sleep engineering, where phase-locked auditory stimulation in the theta band has shown to improve memory consolidation [2], [63]. This is described in more detail in Section 2.6.1.

2.5.3 Frequency

Another potential biomarker is the EEG frequency. For example, due to the oscillatory nature of EEG, tACS has been hypothesized to be a better candidate than tDCS for closed-loop stimulation, since the former allows you to control the frequency, amplitude and phase of stimulation, while the latter is just a constant current applied [44]. Moreover, the dominant frequency in each band may vary across and within individuals over time and taking this into account in closed-loop system may help explain some of the variabilities observed [112]–[114]. For example, the Individual Alpha Frequency (IAF) varies both across and within subjects over time, with peaks anywhere from 7-13 Hz and the mean for young adults being 10 Hz, which is the frequency at the which rTMS and tACS is usually applied at [113], [114].

Hence, applying stimulus matched to the underlying brainwave frequency has been hypothesized to give stronger entrainment, and in some cases better behavioural outcomes [114]–[117]. For example, Widge et al. demonstrated a more pronounced entrainment when using frequency matched tACS in the alpha band, as opposed to when using open-loop stimulation [117]. Similarly, various behavioural studies have been done in this respect for applications such as schizophrenia, cognitive enhancement, anxiety and working memory, with significant improvements observed when frequency was matched to underlying brain rhythms [83], [113], [115], [116]. It must be noted firstly that most of these use electromagnetic modalities (e.g. tACS), and not lights and sounds: one exception would be work done by Pino et al. on anxiety reduction [115]. Here, they used audio-visual entrainment matched to the EEG frequency and found significant decrease in symptoms, when using frequency matched brain stimulation [115].

Moreover, it must also be noted that most frequency matched closed-loop systems are not really closed-loop but rather ‘semi closed-loop’, meaning they extract the individual alpha frequency (IAF) during a separate recording first and then apply stimulation at the IAF throughout the stimulation period [16], [17], [49]. Given that the IAF can change over time within a single subject, a ‘semi closed-loop’ approach is not ideal [118]–[120]. A better approach would be to change stimulation frequency over-time based on the dominant IAF at the time. For example, Pino et al. in their work on anxiety reduction, updated the frequency of the visual stimulus on a second by second basis [115]. However, they were using the frequency output from the Mindwave device’s SDK, which is a consumer grade EEG hardware, which means the software is proprietary and cannot be modified [115]. No work exists which matches frequency on a sample-by-sample basis, which is the theoretical limit and more importantly, that runs on a phone for audio-visual stimulation. Having make it work on a sample-by-sample basis which is the natural limit of what can be done, means it can also be mapped to any application regardless of the rate actually needed. Hence, for this thesis, the aim is to present a system for sample-by-sample frequency update, to ensure the stimulation adapts to the underlying brain rhythms, with a better time resolution [114], [115]. This would be useful for our chronic pain clinical driver, where stimulation matched to the IAF is hypothesized to improve therapeutic outcomes, as detailed in Section 2.6.3.

Having discussed the general use-cases of phase and frequency as feedback measures, we next explore real-time algorithms for phase-frequency extraction.

2.5.4 State-of-the-art: Real-time feature extraction algorithms

To date, only a handful of techniques are used for real-time phase and frequency estimation of EEG, in the context of closed-loop neuromodulation. These include time domain techniques (Autoregressive Modelling and the Phase Locked Loop), frequency domain based techniques (Fast Fourier Transform) and machine learning based ones [2], [44], [87], [89]–[91], [93].

Here, we will explore the state of the art, along with their limitations and advantages, all of which would motivate our algorithm of choice, which is the Phase Locked Loop.

Auto-regressive modelling

Auto-regressive (AR) modelling based techniques, which are a time-series prediction method, have often been used for real time phase extraction [90]–[93]. For example, in its implementation by Chen et al., they first take a small window of past data, and perform AR spectral estimation to optimize the band-pass cut-off frequencies within the delta band, to account for the time varying nature of the EEG dominant frequency. Then, they forecast the band-pass filtered signal into the future using AR time series prediction [91]. The Hilbert transform is then used to get the instantaneous phase and frequency of the forecast-ed signal [91].Note that since the window length (1 second) used is such that the current time point is halfway between end and start of the window, the edge artefacts could be discarded without data loss. Moreover, this method performs better with high signal power and amplitude [90]–[93]. It must also be highlighted here that the whole point of using AR modelling time-series prediction, is to improve the accuracy of the Hilbert based instantaneous phase measure, since the edge effects could be discarded from the edge of the predicted signal, which leaves the current time-point uncorrected. If edge effects were not there, then they could simply have used the Hilbert transform without any future prediction, to get the phase.

The advantages here is that it solves the edge effect problem through forward time series prediction, and is suited to cope with time varying changes in EEG frequency via the adaptive band-pass filtering [91]. Moreover, the forecasted signal is more representative of an EEG signal, as opposed to a simple sinusoid as used in FFT based methods, which are discussed below [44], [87] .However, all these gains in accuracy comes at the cost of added computational cost through expensive operations like the AR time-series prediction and its coefficient calculation, becoming more so acute with increasing model order [90], [91], [93]. For example, Chen et al. used a 50 order AR model, and the computational cost was 72 ms to calculate over a 400 ms window size (500 Hz sampling frequency), on a Mac Book Air 1.8 GHz Intel Core i5 laptop. Finally, another drawback is that there are many parameters that need to be tuned for optimal results, which is done offline via tools like the genetic algorithm [91].

Fast Fourier Transform

In contrast to the AR Modelling method which works in the time-domain, another approach is to work in the frequency domain, for example using the Fast Fourier Transform (FFT) based methods [44], [87], [93]. Here, EEG is assumed to be stationary over short time windows, and the FFT is taken over the window to get the dominant frequency of the band-pass filtered signal. A sine wave is then generated at this frequency to forecast the signal to the current time-point, and hence extract the phase in real-time [44], [87], [93].

In contrast to the AR based method, these are relatively non-parametric, and requires little tuning, except for the optimal past window sizes, which was found to vary with the frequency band under consideration [44], [93]. It is also computationally less demanding and hence optimal for low latency application [44]. For instance, in the study by Mansouri et al., whilst the AR modelling method makes use of a 50 order AR model along-side the Hilbert transform, the FFT based method just uses a order 10 IIR filter along with the Fast Fourier Transform [44], [91]. This is evident in the execution time difference when tested on a Mac Book Air 1.8 GHz Intel Core i5, where the FFT based method was found to be 2 orders of magnitude faster than the AR method: while the former took 0.68 ms of computation time to process 400 ms of past data, the latter took 72 ms to do the same [44]. Increasing the past window size to optimize for different EEG bands, only made slight differences, giving an execution time of less than 1ms for all cases [44].

Despite lower computational cost, one of the limitations with this method is that is assumes EEG to be stationary over short time windows and the forecast-ed signal is just a simple sinusoid which does not capture the true morphology and non-stationarity inherent in EEG signals [44]. Moreover the algorithm works effectively only when the dominant frequency in a band was stable over time, as it does not have an adaptive band-pass filter to account for dominant frequency variations over time, as done in the AR based method [44], [91]. Also, as with the AR method, performance improves with higher signal amplitudes [44], [93].

Regarding accuracy, the FFT based algorithms does not give any major improvements: in general, both algorithms showed a Phase Locking Value (PLV) of more than 0.5 for different bands in the comparison study by Mansouri et al., where a value of 1 indicates perfect phase-lock [44]. It was also found that performance was best with the alpha band, which is expected due to the dominant EEG oscillations here [44].

Machine learning based methods

Finally, there exists machine learning based approaches to phase extraction [89], [93]. The approach here is to train the model by learning the mapping between minimally pre-processed EEG data and its non-causally derived phase (using Hilbert transform) offline; this model is then deployed online for real-time EEG phase extraction [89]. This was shown to give better phase targeting accuracy, when compared against standard signal processing methods using the Hilbert transform [89]. However, it must be noted that the study was done on just the alpha band, where signal power and SNR is high and may not generalize well across different bands, where signal power is lower [89], [92].

We do not choose to go this route for our application because these would require a training period, and hence not ideal for *in-the-wild* use, where the ideal solution is a ‘fit-and-go’ one. Such requirements for a training period may also compromise user-comfort and makes it non-ideal for ambulatory use-cases as ours [89]. For these reasons, we choose to primarily focus on signal processing based methods.

2.5.5 Proposed algorithm: Phase Locked Loop

As seen from the previous Section, the current state-of-the art suffers from various limitations: high computational burden (AR modeling); assuming signal stationarity and inability to adapt to the time-varying frequency of the EEG (FFT based methods); and the need for training data every session (machine learning based methods) [44], [87], [89]–[93]. Moreover, since these methods operate on a window of data, drawbacks include: edge artefacts, delay, and the inability to operate on one sample at a time to drive latency down [90], [91].

To tackle these issues, we propose to use the Phase Locked Loop (PLL)instead. The PLL is a well established technique in the communication field and often used for high frequency applications such as clock synchronization and clock recovery [121]. Unlike other methods which operate on a window of data, a PLL can be used on a sample-by-sample basis which results in no edge artefacts and reduces signal processing latency [121]–[123]. Moreover, PLLs are usually low order (order 2 or 3) and also adapt to EEG dominant frequency variations, making any adaptive band-pass filtering unnecessary to account for frequency variations over time and other signal non-stationarity, as is done in the AR based method [91], [123]. All these makes this method an excellent candidate for real-time phase estimation as it could drive loop latency down with its sample-by-sample operations, adapt to signal variations with no edge effects, all of which are important for closed-loop applications.

PLLs have already been used in the context of closed-loop EEG applications for sleep engineering (see Section 2.6.2), with the focus being on the Slow Oscillation band (0.5–3 Hz); however, to the best of the author’s knowledge, no work exists investigating its applicability in other EEG bands [2]. Moreover, most PLL based platforms are lab based and none on phones [2], [4], [5], [13]; a more detailed review of these platforms are given in Section 2.6.2. Taking one such example, Santostasi et al. have designed and validated such a system on a PC, achieving phase errors of 12.51±28.85 degrees and 13.63 ± 9.88 degrees over the entire N3 sleep epochs (stage 3 of non-rapid eye-movement sleep, see Section 2.6.2 for more details) and parts with just slow waves, respectively [2]. Moreover, when just targeting the peak of a slow oscillation, the phase error was reported to be 0.37 ± 25.61 degrees. The total loop delay including data acquisition, processing and stimulation was 70 ± 5 ms [2]. We aim to improve upon this work by investigating phase targeting accuracy for all the EEG bands and not just the delta band, to open door for applications outside sleep engineering, and more importantly provide an phone based implementation for *in-the-wild* use-cases. These contributions are detailed in Section 4.4.3 and 5.3, respectively.

A note on Frequency

It must be noted that although most of the aforementioned algorithms have been used for phase extraction, little work exists exploring real time dominant EEG frequency extraction, in the context of a closed-loop system. One such example includes the work done by Pino et al.: however, here, no details were given on the algorithm used, since the frequency was extracted using the mindwave EEG device SDK [115]. Moreover, we also would like to highlight that we use the PLL to extract instantaneous frequency by taking the derivative of the instantaneous phase; in other words, we are closing the loop on a sample-by-sample to achieve the finest time-resolution. In contrast to work done by [115] where their time resolution is 1s, we aim to provide a system that adapts to the smallest change in frequency i.e. sample by sample. As to whether the frequency changes at such small time-scales to make closing the loop meaningful, no literature exists documenting the EEG frequency rate of change, as evidenced in Section 4.2.3 and 2.6.3. Hence, we did our own analysis in Section 4.2.3 and showed that it does, confirming the potential need for and relevance of our sample-by-sample approach.

2.5.6 Summary

In this Section, we discussed the *how* of closing the loop, using phase and frequency as feedback markers. We started with an exploration of various applications that use phase and frequency, as biomarkers. This was then followed by a discussion of the state-of-the-art real-time algorithms for extracting phase and frequency. Here, we noted that any suitable real-time algorithm should have: low computational complexity to drive down loop latency, be able to work with a limited amount of data to extract features accurately, and be able to adapt to the non-stationary nature of EEG signals. With these in mind, we then explored three state-of-the-art algorithms whilst highlighting their limitations: AR modelling based methods (high computional complexity), FFT based methods (non-adaptive), and machine learning based methods (requires training data-set per session). We then concluded the Section highlighting that the PLL is a well suited algorithm for real-time application that combat these constraints: they can operate on a sample-by-sample basis without any edge effects, is computationally simple and can adapt to the time-varying frequency changes of the EEG.

2.6 Clinical application drivers and technology contribution

2.6.1 Introduction

In this Section, we discuss our application drivers: sleep engineering and chronic-pain in Sections 2.6.2 and 2.6.3, respectively. The former uses phase as a bio-marker and the latter frequency. We also highlight in this section the state-of-the-art platforms for both these applications, which then contextualises our technology contribution.

2.6.2 Sleep engineering

Sleep plays a vital role in a variety of cognitive processes such as memory formation. Contrary to popular notion, the brain is highly active during sleep. The field of sleep engineering then tries to use technology to make use of time spent sleeping, to enhance our cognitive functioning such as memory formation. To understand how, the first step is to be aware of the different stages of sleep.

There are two kinds of sleep; the Non-Rapid Eye Movement (NREM) which makes up about 75% of the sleep duration while the remaining 25% is Rapid Eye Movement (REM) sleep [124]. One cycles through these stages starting from NREM and progressing through to REM sleep, and this repeats itself roughly 4-5 times of roughly 90 minute cycles per night [124].

NREM sleep is further divided into four stages, as per the American Academy of Sleep Medicine (AASM) standard. Stage 1, occurring at sleep onset, consists of relatively higher frequency EEG in the alpha range and this is the lighter stage of sleep where one is half awake [124], [125]. One then progresses to the deeper Stage 2 and 3 of sleep. The latter is often known as Slow Wave Sleep (SWS), as it is characterized by slow oscillations which has a relatively low average frequency of 0.8 Hz and large amplitudes of at least -75 μV, and is mostly generated in the frontal areas of the brain [124]. Sample SOs are shown in Fig. 2.8. Finally, NREM sleep usually decreases in intensity and duration throughout the night, unlike REM sleep which is characterized by rapid eye movements, and increases in duration and intensity through time.

From the different stages of sleep, SWS has in particular been a field of investigation to enhance memory consolidation. Recent research has shown that declarative memory consolidation can be enhanced through playing auditory tones during slow wave sleep to enhance slow oscillation (SO) amplitude, which has shown to be important for the stabilization of long-term memory [3]–[5], [12], [15], [20]. The same has also been reported using modalities such as Transcranical Electrical Stimulation but practical concerns remain such as difficulty of reading the stimulation distorted EEG waveforms and usability during sleep [126]. Hence, using auditory stimulation is an attractive option, due to both ease of use and fewer artefacts. Moreover, recent work has also shown that the timing of the auditory stimulus

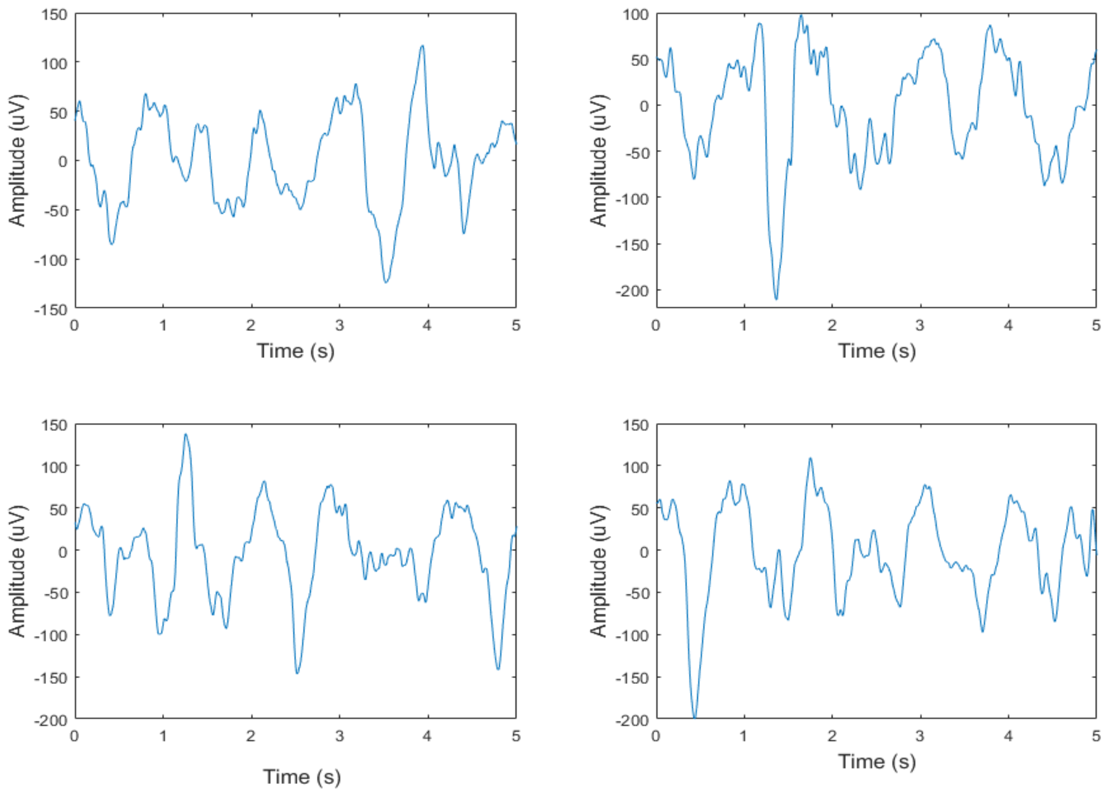

Figure 2.8. Sample slow oscillations.

relative to ongoing EEG phase matter for optimal SO enhancement [3]–[5], [12], [15], [20], [63], [127]. This then demands the need for algorithms that could synchronize the stimulation protocol to ongoing EEG accurately.

One of the earliest work done was by Ngo et al. [63], which involved delivering stimuli after a fixed time-delay following the first detected slow oscillation (SO) negative peak. Once it detected the negative peak, it then used a fixed time-delay to target the positive peak of the slow oscillation; this time-delay was determined on a subject-to-subject basis offline, by measuring the distance between the negative detected peak using the SO detection algorithm, and the positive peak that followed [63]. This pre-set nature of the algorithm is not very adaptive and moreover, there is the burden of tuning the algorithm on a subject-by-subject basis. Although they don't report phase targeting accuracies, Santostasi et al. replicated this algorithm and reports phase errors of 11.3±65.6 degrees (N=5) [2]. Leminen et al., used a similar threshold based algorithm, and reported high variance as well with phase error of -18±67 degrees (N=15) [20]. Other groups have tried using sine-fitting algorithms: for example, Cox et al. used the FFT over a 10s epoch of data to obtain the dominant peak frequency, and fits a sine-wave onto the EEG signal based upon this dominant frequency [15]. The phase was then determined on the sinus itself. Similarly, Debellementaire et al. uses the same approach, but this time using real-time linear regression (via recursive least square method) to determine the best fit on a 5 s window of data [12]. Based on these, Cox et al. reported 11.3±65.6 and 33.2±75.8 degrees (N=12) error, for the positive and negative peak, respectively and Debellememaniere et al. with 0±52 degrees (N=20) phase error reported on an embedded device [12], [15].

Study	Platform type	Method	Target phase (°)	Phase error (°)
Santostasi et al.	PC based	PLL	60	0±25.6
Ong et al.	PC based	PLL	60	10.28±26.69
Palambros et al.	PC based	PLL	340	11.5±73.4
Ngo et al.	PC based	Adaptive threshold	90	11.3±65.6
Leminen et al.	Portable	Adaptive threshold	0	-18±67
Cox et al.	PC based	Sinus fitting	90	11.3±65.6
			270	33.2±75.8
Ferster et al.	Portable	PLL	45	0.4±46.8
Debellemaniere et al.	Portable	Sinus fitting	45	0±52

Table 2.1. Summary of state-of-the-art platforms for sleep engineering [2], [4], [5], [12], [13], [15], [20], [63]. Target phase assumes a sine wave; so 90 degrees target phase would be the peak.

In contrast to these, other groups have used Phase Locked Loop (PLL) based techniques with slightly better performance compared to the algorithms, the results summarized in Table 2.1, along with the aforementioned algorithms. Firstly, Santostasi et al. used a classical PLL and achieved a phase error of 0±25.6 degrees (N=5) [2]. Similarly, Palambros et al. and Ong et al. conducted clinical studies using the same platform and reported phase errors of 10.28±26.69 (N=16) degrees and 11.5±73.4 (N=13) degrees respectively [4], [5]. It must be noted here that Palambros et al. reported a much higher variance (±73.4 degrees) using the same PLL; this is expected since Palambros et al. used older volunteers for their study, as their aim was to show that the same slow oscillation enhancement could be observed in an older population too. Given that older people have lower amplitude EEG's, the higher error is then expected, as the performance of the PLL has shown to be amplitude dependent, with poorer tracking for lower amplitudes [2], [4], [5]. This amplitude dependent performance is also something we demonstrate in Section 4.4.4.

It must also be noted here that all these platforms, both PLL based and others, used frontal channels for the phase-targeting algorithms, with the majority using the FpZ channel [2], [4], [5], [12], [13], [15], [20], [63]. This is expected since SO's are most prominent in frontal areas of the brain [124].

It must also be noted that most of these are lab based platforms and only three of them afford portability. Of these, Debellemeniere et al. used a commercial grade sleep EEG device which runs on an embedded system with on-board signal processing locked down, that others can't access or modify. On the contrary, Ferster et al. provides a more research oriented embedded device, with options to modify protocols and settings [12], [13]. In contrast to these embedded paltforms, Leminen et al. used a Surface Pro laptop-tablet hybrid in conjunction with the Enobio wireless EEG headset to provide an ambulatory system. However, they used external speakers to deliver the sounds and the surface pro laptop-tablet hybrid is not ideal for ambulatory purposes, as it is not as ubiqutous as say a smart-phone. More importantly, they use the same algorithm as Ngo et al., which stimulates based on a fixed time-delay once the negative peak of the SO is detected, to reach the peak (the time-delay being tuned on a subject-by-subject basis offline). This non-adaptive nature of the algorithm again explains the higher error they reported.

Technology contribution

To date, no platform exists on a ubiquitous platform such as a mobile phone, which is not just lighter and smaller, but also more familiar to operate for the wider population. More importantly, we hypothesize that using an adaptive algorithm like the PLL as opposed to the ‘fixed’ types like the one used by Leminen et al., would result in lower phase targeting error. Moreover, since the smartphone would have less dedicated computational resources than a laptop-tablet hybrid or an embedded device, using a computationally simple low latency algorithm algorithm like the PLL, would be advantageous. With these in mind, our contribution here would then be to provide the first smart-phone based platform for closed-loop auditory stimulation using the PLL (see Section 5.3). This would run on an Android app, making it easily accessible for the wider population and would allow larger-scale studies to be conducted at the volunteer’s home, avoiding the high costs and waiting times of sleep labs. More importantly, this would also help generalize findings from the lab to the real-world.

2.6.3 Chronic pain

Next, we discuss the second application driver; treating chronic pain using rhythmic lights and sounds. Pain when experienced, is partly physiological (what is sensed through the body) and partly psychological (what is perceived). The perception of pain can exaggerate and in effect, inhibit one’s capability to cope with it and this is more so the case for people suffering from chronic pain. Due to these different factors that contribute to the total ‘level’ of pain experienced, one can expect a variety of brain regions to be involved here. This is exactly what researchers have found to be the case; different brain regions interact and contribute to the experience of pain and this network is commonly called as the ‘pain matrix’ [128], [129].

Pharmaceuticals have been the go-to method for treating pain. Medicines such as paracetamol and anti-depressant drugs often help reduce the perception of pain and offer pain relief [14]. However, these come with various side effects and the effectiveness varies from individual to individual [14]. Other methods include psychological therapies and bio-feedback based techniques such as neuro-feedback, which involves reading the user’s EEG and feeding back information on the user’s current brain-state via modulated sounds or other means [130]. This helps the user train their brains to reach a particular state, through sheer effort and can be used to modulate the power of a certain EEG band [130]. In the case of chronic pain, an inverse relationship between the perception of pain and alpha power has been observed [10], [11]. Hence, by training the brain to increase its alpha power via neuro-feedback, it is possible to reduce the perception of pain, thereby providing an analgesic effect [131]. However, such techniques involve effort, time and depend on the user’s motivation and willingness to stick with the protocol.

This is where neuromodulation comes in, more particularly brain wave entrainment. As, detailed in Section 2.3.4 and 2.3.3, this is a quick and effective way to synchronize the dominant brain wave frequency to one of choice, in this case, alpha power. Whereas techniques like neuro-feedback involves the patient doing the work, entrainment involves the stimuli doing work, and hence, would be an easier experience for the user. Various groups have explored this to increase alpha power, and in turn, reduce pain perception [6]–[8], [10], [11]. For example, Ecsy et al. showed in healthy volunteers (N=32), that a brief 10 minutes of audio-visual stimulation in the alpha range reduced experimentally induced pain significantly, as measured by pain ratings; they used LED goggles for visual stimulation and binaural beats for the auditory stimulation [10]. Here, the visual stimulation resulted in a larger reduction in pain perception and in a follow-up study, they replicated the findings but this time correlating the reduction in pain-perception to an increase in EEG alpha power as well, and that too on a larger sample size (N=64) [6]. Taking this outside of experimental pain, Laura et al. used a brief 4 minutes of alpha visual stimuli (delivered via LED goggles), on participants with chronic musculoskeletal pain (N=22) [8]. They were not able to find any significant pain reduction, arguing that the 4 minutes of stimulation duration is not enough and longer durations may be needed, as done with other studies [6], [10]. Nevertheless, they still showed visual alpha stimulation was able to increase alpha power in chronic pain patients, and not just healthy volunteers [8]. Apart from visual stimuli, tACS has also been used: for example, Laura et al. in a different study, applied 20 minutes of alpha tACS over the somatosensory region on healthy volunteers (N=20), and showed a decrease in experimental pain, which was inversely correlated with alpha power [7]. Ahn et al. then replicated this finding on patients with chronic lower back pain, this time applying 40 minutes of alpha tACS, again over the somatosensory region [11]. All these studies then show the potential of alpha as a potential biomarker for chronic pain, with alpha power being inversely related to pain perception. Consequently, by increasing alpha power via entraining stimuli such as tACS, visual or auditory, one could reduce pain perception.

It must be noted here that all the aforementioned studies have been open-loop studies, where the alpha frequency stimulus was pre-set at 10 Hz [6]–[8], [10], [11]. As a result, variabilities in results were observed, as is common with open-loop neuromodulatory techniques [8], [10]. For example, for the visual alpha stimulation study by Laura et al. on chronic pain patients, some patients showed several points decrease in pain ratings, whilst others showed no improvements and some even reporting increased pain after stimulation [8]. Similar variabilities were observed by the study by Ahn et al., applying alpha tACS for chronic pain patients [11]. Both hypothesized that not taking inter-subject differences in Alpha frequency, i.e. the Individual Alpha Frequency (IAF), as a potential reason for this variability [8], [11]. This is a plausible hypothesis, as other studies have shown entrainment is more effective and pronounced, when the stimulation frequency is closer to the Individual Alpha Frequency (IAF) [16], [17], [49], [117], [132]. For example, Notbohlm et al., used photic entrainment in the alpha range delivered via LED goggles, and showed entrainment to be most effective when the flicker frequency was closer to the IAF [16]. Similarly, Widge et al. showed frequency matched alpha tACS stimulation to show pronounced entrainment and increase in alpha power, compared to the open-loop case [117]. Given these, it would be then reason-

Author	Sensing	Actuation	Invasive/Non-invasive	Application
Widge et al.	Implants in frontal cortex.	tACS	Invasive	General
Pino et al.	EEG (Mindwave device)	Visual	Non-invasive	Anxiety

Table 2.2. Closed-loop stimulation platforms using frequency as feedback measure.

able to hypothesize that an alpha power increase and associated pain relief, would be more pronounced and likelier, if stimulated at the IAF to take into account inter-subject variability; this may explain the variability in results observed with the open-loop paradigm.

That said, most studies in literature assume the IAF to be a static trait amongst individuals that does not vary over time within a subject but only across subjects; hence, the usual approach is to calculate the IAF during a base-line recording period on a per subject basis, and then use that as the stimulation frequency for the remained of the experiment [16], [17], [49]. However, recent evidence shows that the IAF does vary over time within a subject: for example, Haegens et al. showed that the IAF within a subject increases with increasing cognitive load; they calculated the IAF on a second by second basis, and showed that it varied by 10.3±0.9 Hz within subjects, over a 12 min recording. Similarly, Gutmann et al. showed a significant increase in IAF (n=10, p-value=0.012) after 30 minutes of intense exercise [120]. Finally, on a much longer time-scale, Weber et al. showed a significant increase in IAF (n=6, p-value=0.001) during 120 days of isolation [119].

Hence, given that the IAF could change over time, then the ideal platform for the chronic pain application driver should be able to measure these changes in IAF over-time and adapt the audio-visual stimulation frequency accordingly. No such platforms exist currently, most likely due to this being a very recent field of work, with the first results by Ecsy et al. published in 2017 [10]. Hence, naturally, open-loop stimulation was the first step, as no. closed-loop system existed to validate the IAF matching hypothesis. Outside of this, we checked to see if there are other closed-loop platforms, where the frequency feedback measure is used to control stimulation frequency. To the best of our knowledge, there exists two platforms, as summarized in Table 2.2.

The first one by Widge et al. provides tACS stimulation where the frequency of the tACS is matched to that of the on-going neural activity measured from an implant in the frontal cortex [117]. Here, they used analog circuitry to extract the feedback frequency measure, to minimize delay [117]. With this system, they conducted an in-vivo study on monkeys, and showed that 10s of alpha frequency matched tACS increased alpha power at the frontal cortex, whilst the control open-loop alpha stimulation decreased alpha power [117]. Even though the alpha power increase in closed-loop condition was promising, the open-loop case decreasing alpha power was counter-intuitive; they intuited this may have been due to the fact that they only applied 10s of tACS, whilst normal tACS experiments apply usually 20 minutes or more of tACS i.e. the dosage was too small [117]. Regardless of the results, this platform would not be suitable for our chronic pain application driver as it is based on invasive sensing and using tACS as the stimulation modality, both of which does not suit our requirement of a non-invasive light and sound based mobile platform.

Secondly, Pino et al., provided a non-invasive platform for frequency matched visual stimulation to reduce anxiety [115]. Here, the dominant frequency for each EEG band was obtained from the Mindwave EEG device, and its Software Development Kit (SDK), which outputs the frequency values every second [115]. This frequency was then used to tune the frequency of the visual flicker provided via a set of lamps, one per EEG band [115]. This platform, though its non-invasive and provides visual flicker, lacks various features we need. For example, this system is based on the consumer grade Neurosky EEG device by Mindwave, which is not suitable for research purposes, as these tend to be lower quality with only 1 channel to record from [115]. The system is also locked to tracking EEG dominant frequency changes every second, which is obtained from the proprietary SDK; we aim to track changes every sample, to get to the theoretical limit with more details in Section 4.6 and 5.4. Finally, this system also does not provide the option for audio binaural beats based stimulation, which we need.

Technology contribution

Given these, it is clear that both these systems are not fit for purpose, as these were not designed for the chronic pain application driver specifically. The one by Widge et al. is for frequency matched tACS with invasive sensing and the one by Pino et al. is a based on a consumer grade EEG device with just one channel, locked proprietary software, and connected to external lamps for visual flicker [115], [117]. Both these do not fit the requirement for frequency matched audio-visual stimulation on a portable device; no such platform exists today that is fit for purpose for the chronic pain application drivers, as this is a recent field and all work has been done using open-loop platforms.

Hence, our contribution here would be to provide the first in kind of a smart-phone based closed-loop platform designed solely for the chronic pain application driver; this would track the IAF on a sample by sample basis (theoretical limit for resolution) and tune the frequency of audio-visual stimulation accordingly. The audio stimulation would be provided in the form of binaural beats delivered from the phone and the visual stimulation as screen flicker from a phone. Most importantly, this would be a smart-phone based platform running on an Android app, which would enable large-scale at-home experiments on patients, with multiple sessions. This would be harder to do on lab-based platforms, as chronic pain patients are often fatigued quickly and find it tiresome travelling to and fro, making multiple session experiments harder. This would be easier with an ambulatory platform, especially one as ubiquitous as a smart-phone, as the patients could take part in the sessions from the comforts of their own homes. This platform contribution is detailed in Section 5.4 of Chapter 5.

Finally, it must also be noted here that we have active collaborators working in this field at the Royal Salford Hospital, who have authored lots of the publications in this field [6]–[8], [10]. They will be using the platforms developed here to move past open-loop stimulation;

in other words, figuring out if alpha frequency matched stimulation could result in stronger entrainment and drive down variability in results observed [8], [10].

2.6.4 Summary

In this Section, we detailed the two application drivers that our system is designed for: sleep engineering and chronic pain. For sleep engineering, the aim is to provide phase-locked auditory stimulation targeting the Slow Wave oscillation, to enhance memory consolidation. For this, we detailed various technology advancement work done to improve both speed and accuracy; of these, we highlighted that only two portable platforms exists, with both running on embedded hardware. None exists that run on a ubiquitous platform such as the smartphone to enable truly *in-the-wild* research, and this is where our contribution lies for this field. For the chronic pain driver, the aim is to provide frequency-matched audio-visual stimulation in the alpha band, to enhance entrainment and in-turn the associated analgesic effects. For this, ours would be the first-in kind of a platform and no state-of-the-art exists to compare to, making the novelty the contribution.

2.7 Conclusions

In this Chapter, we started our literature review describing the individual components of a closed-loop platform in isolation, starting with brain sensing and actuation. The EEG was chosen to sense brain activity due to its low cost, portability and good temporal resolution. For actuating the brain, the modality was application dependent: for example, auditory stimulation for sleep engineering and audio-visual stimulation for chronic pain. This was followed by a discussion on the various challenges of building a closed-loop platform and we highlighted its potential advantages over open-loop platforms. Advantages include addressing the variability in therapeutic outcomes and reducing negative side-effects due to over-stimulation. Moroever, we also highlighted here the need for mobile paltforms, to enable ‘out-of-the-clinic’ research. Next, a detailed discussion was given on the various feedback measures available to close the loop between sensing and actuating. Here, the feedback measures we focused on was EEG phase and frequency, both of which come from our application drivers. With this, we then discussed the challenges of real-time phase-frequency extraction and highlighted the need for low latency algorithms, after which state-of-the art algorithms were discussed. After showing the limitations of window-based methods like the FFT and AR modelling, we then showed that the use of a PLL was ideal. Unlike the other methods, the PLL could operate on a sample-by-sample basis to drive latency down, is computationally simple and can adapt to dominant frequency changes of the EEG over-time, all of which are important factors for a real-time system. The Chapter then concluded with a detailed discussion on our application drivers chronic pain and sleep engineering, alongside the state-of-the-art platforms available. This then contextualized our technology contribution, in light of the gaps present.

Chapter 3

Open-loop light and sound based neuromodulation on a phone

3.1 Introduction

The first step to a closed-loop platform is to design an open-loop stimulation one; i.e. one which can present visual and audio stimulation at the wanted frequencies, but with the settings fixed in advanced via a user interface rather than adjusted automatically in real-time based upon sensed user data. This would not only help characterize and validate the stimulation accuracies, but any closed-loop system should also have an open-loop mode within it, for use as a base-line. Moreover, the open-loop app in itself is being used by our collaborators in chronic pain, the details of which are presented in Section 3.4.2. With these in mind, the technical work in this chapter focuses on accurate stimulation for both auditory and visual modes. The GUI for this open-loop app can be seen in Figure 3.17 in Section 3.4.2.

The chapter then is then laid out as follows: the design, implementation and accuracy characterization for the visual stimulus is detailed in Section 3.2; then the one for auditory stimulus is presented in Section 3.3 and finally, the Chapter concludes with a brief discussion of the user experience related feedback, obtained from chronic pain patients and areas where the app is currently being used for.

3.2 Visual entrainment

3.2.1 Introduction

Here, we discuss the delivery of visual stimuli, via a smartphone. It must be noted here that although we are designing the app as a standalone open-loop app, we still need to ensure that the stimuli generation methods are fit for closed-loop stimulation for use in Chapter 5. This means that we not only want accurate stimuli delivered for the frequencies of interest in our study, but also we need to ensure that at frequency changes (expected in closed-loop), there would be no signal discontinuity. Visual stimulation is common place in SSVEP based brain computer interfaces, where the stimulus is a flickering light at a specific frequency, which also results in entrainment, as detailed in Section 2.3.3. Hence, we contextualize our contribution in light of the state-of-the-art in SSVEP literature.

Most SSVEP studies are conducted on lab-based systems, where the flicker is delivered either via a computer monitor or from a dedicated hardware (e.g. LED devices) [133]–[137]. While dedicated LED hardware cannot be reconfigured and adapted for different purposes, it can present a much wider range of frequencies. In contrast to this, computer monitor based SSVEP's has the advantage of being re-configurable and flexible, whilst being limited by the refresh rate of the screen; the frequencies presentable are limited to those that are integer divisors of the screen refresh rate [133]. For example, assuming a 60 Hz refresh rate, to deliver a 30 Hz flicker, the frame rendering scheme would be one frame-on (white color) and one frame-off (black color). On the other hand, a 11 Hz flicker is not directly possible since it would require 2.725 frames on and 2.725 frames off.

To address this issue, groups have used either approximations of a square wave or a sine wave, for the stimulus waveform [133]–[136]. For example, Andersen et al. used a method where certain frames are provided intermediate intensity values to get over the refresh rate limits (more details in Section 3.2.4) [133]. They tested this algorithm on a computer screen with refresh rate of 85 Hz and 120 Hz, and for just 4 frequencies in the 10–15 Hz range [133]. Apart from square wave approximation, other groups have used a sine wave as the stimulus waveform, the reasoning being that it could generate any frequencies up-to half the refresh rate of the screen [135], [136]. For example, Chen et al. used this method to generate signal in the 6–40 Hz range elicited on a 120 Hz refresh rate computer monitor screen [135].

It must also be noted here that all these methods have been so far implemented on lab-based system's (computer screen) and not on phones, which are expected to have lower refresh rates; for example 60 Hz for the Sony Xperia Z3 phone we used [91], [134], [136], [137]. To the best of our knowledge, only one study by Wang et al. explored the use of a mobile phone to deliver SSVEP on a 55 Hz refresh rate phone; however, they only investigated performance for a 11 Hz signal, which is easy as it is an integer divisor of the refresh rate [138].

Aside from portability, one other limitation of all these methods is that they were designed for SSVEP based Brain Computer Interfaces (BCI) and hence, are by default for open-loop applications where the frequency is preset and fixed i.e. not changing over time [48], [134]–[136], [138]. However, for closed-loop applications, the frequency is expected to change over time and hence, signal continuity needs to be ensured at these frequency changes; without this, there would be jumps and discontinuities in the generated signal. Although this chapter is about the open-loop stimulus implementation, the same stimulus method will be used for the closed-loop system later in Chapter 5, which makes signal continuity at frequency changes an important features. None of the aforementioned methods takes this into account, as they were designed for open-loop applications.

With these in mind, we aimed to improve upon these methods via two contributions: first by providing an on-phone implementation for the stimulus waveform, and second by ensuring the stimulus method ensures signal continuity, for it to be usable in closed-loop applications. In light of this, we first start the Chapter with the requirements and specifications in Section 3.2.2. Then, we detail the test setup and methods used to characterize the stimulation implementations, in Section 3.2.3. This is then followed by a discussion of the three different stimulus methods we investigated for giving different levels of frequency accuracy and performance: the square interpolation, square sine and sine, in Sections 3.2.4, 3.2.5 and 3.2.6, respectively. We then contextualize our method of choice, after comparing and contrasting these in Section 3.12.

3.2.2 Requirements and specifications

Firstly, we set the requirements and specification for the visual stimulus. These would be derived from our chronic pain application driver, where the target bio-marker is the alpha band (8–13 Hz), as detailed in Section 2.6.3. With this in mind, the specifications are:

- **Frequency range: 8–13 Hz** For the chronic pain application driver, the analgesic effects were seen via visual stimuli provided in the alpha band (8–13 Hz), as detailed in Section 2.6.3 [6], [8], [10]. Hence, we aim to deliver stimuli in these frequencies accurately.
- **Signal continuity at frequency changes** We highlight again although this chapter is about the open-loop app, we eventually aim to embed these stimuli generation methods into the closed-loop app later on in Chapter 5. Here, the Individual Alpha Frequency (IAF) is expected to change over time, which we also showed in Section 4.2.3 [118]–[120]. Hence, we need to ensure that there would be no discontinuity in stimulus waveform when frequency changes in the alpha band, to be fit for closed-loop application later on in Chapter 5.

3.2.3 Test methods

To test the accuracy of the visual stimuli provided, a Sony Xperia Z3 (quad-core 2.5 GHz Snapdragon 801 processor with 3GB RAM), with a 60 Hz screen refresh rate was used; this phone was used as this was the one that came with the wireless EEG amplifier we used for our closed-loop system, as detailed in Chapter 5. We tried also using a different phone model (Samsung Galaxy S3) for the closed-loop system but it resulted in a much higher bluetooth transmission latency; hence, we just stuck with the phone that came with the system i.e. the Sony Xperia Z3 phone. Measurements of screen brightness were taken using a PDA10CS-InGaAs photo-detector. This integrated chip contains a photo-diode, operational amplifier, and associated circuitry to provide accurate light intensity measurements. Tests were conducted by placing the phone and the sensor inside a sealed box to avoid noise from other light sources. This setup is depicted in Fig. 3.1.

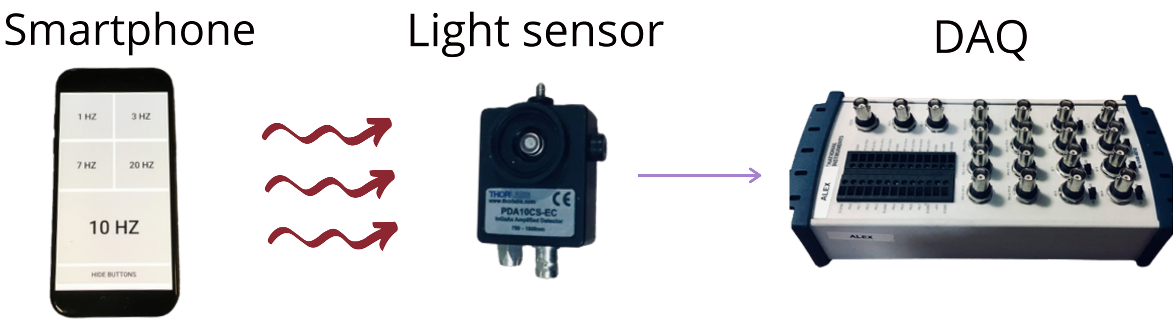

Figure 3.1. Test setup for visual stimulus.

Out of the three stimulation methods detailed in the following sections, two of them involve generating a square wave, as detailed in Section 3.2.4 and 3.2.5. For these, the brightness of the flickering screen mainly takes the shape of a square wave; we say mainly, because with square interpolation method in Section 3.2.4, interpolation is used to set the brightness at intermediate intensities at particular points in time. For these, measuring the instantenous frequency present is not straight-forwards as say measuring that of a pure sinusoid, where the Hilbert transform could be used to obtain the instantaneous frequency. Since the square-wave is not a mono-component signal like the sinusoid, we then needed to use a different approach to calculate instantaneous frequency. To address this, the signal was first band-pass filtered with a zero-phase order 10 Infinite Impulse Response(IIR) butterworth filter with a 8–13 Hz passband. Then, the Hilbert transform was used to calculate the instantaneous frequency of this filtered waveform, as an approximate to the square-wave frequency. The rationale here is that since the filtered waveform would not contain any harmonics of the original square waveform for the 8–13 Hz band, we could use this narrow-band sinusoid then as a good approximate for the square wave frequency. Moreover, to get instantaneous frequency, we take the derivative of the instantaneous phase. Consequently, since the derivative is a noise-sensitive operation, a smoothing filter is needed, especially for the flicker waveform obtained from on-phone testing in Section 3.2.4 and 3.2.6, which may contain residual noise from external light sources. Hence, even with prior band-pass filtering, we needed to smooth the output instantaneous frequency. For this, we used an Order 2 IIR Butterworth low-pass filter with a 0.3 Hz cut-off frequency. We chose this, since we are interested in DC trend as the frequency is expected to be a staircase) and any high frequency noise can safely be assumed to be derivative induced.

With these, for each of the stimulation methods, we first conduct an offline analysis to explore which of the three methods work best, under our constraints of a 60 Hz refresh rate. Moreover, with the simulations, we were also able to analyse the effect of varying the refresh rate on the waveform quality for each of the methods, which helps explore how the results may vary for phones with higher refresh rate: this is not possible otherwise in hardware, as we have only access to the Sony Xperia Z3 phone with a 60 Hz refresh rate. More importantly, this is the only phone supported for real-time EEG data streaming used in Chapter 5.

3.2.4 Stimulation method 1: Square interpolation

Square-waves are one of the common waveforms used for SSVEP studies in EEG literature [133], [134], [137]–[139]. As discussed in Section 3.2.1, it is also more likely to elicit entrainment in the brain, when compared to sine-waves [139].

Implementation

The main challenge for on-phone square-wave generation here is that the stimulus will be synchronized to the screen's refresh rate and hence, a limit exists to the number of frequencies presentable; the visual stimulation frequencies directly realisable on the smartphone screen are limited to integer divisors of the screen refresh rate of 60 Hz. To address this issue, we used an interpolation algorithm proposed by [133], where intermediate intensity values for the screen brightness are used at specific frames for frequencies that are not integer divisors of the screen refresh rate. An example waveform using this method is shown in Figure 3.2a. The intensity value (w) at each frame (i) is given as,

$$w[i] = \begin{cases} 1 & \text{if} \quad 1 \leq i \bmod \lambda \leq r_{\text{on}}\lambda \\ r_{\text{on}}\lambda + 1 - i \bmod \lambda & \text{if} \quad r_{\text{on}}\lambda < i \bmod \lambda < r_{\text{on}}\lambda + 1 \\ 0 & \text{if} \quad r_{\text{on}}\lambda \leq i \bmod \lambda \\ i \bmod \lambda & \text{if} \quad i \bmod \lambda < 1 \end{cases} \tag{3.1}$$

where $\lambda = R/f$. Here R is the screen refresh rate, f is the desired stimulus frequency, and r_{on} is the fraction of the stimulus cycle in which the stimulus is on[133]. Also note that the first line corresponds to fully-on state i.e. full intensity; the second line the transition from on-to-off; the third line the fully-off state and the fourth line, the off-on transition. In code, the control flow is executed from top to bottom in order, with a break statement for each condition: for example, if the full-on state condition is met, then the remaining lines are skipped.

This algorithm was used to generate flicker frequencies in the range 8–13 Hz, i.e. the alpha band, according to our specs. This was realised via the OpenGL graphics library which uses separate threads for frame rendering and screen updates, leaving processor resources available for other activities on the phone (data collection and real-time data analysis).

Results: Offline simulation

Before porting the algorithm onto a phone, an offline simulation was conducted first in MatLab, to investigate the effects of varying refresh rates, as mentioned previously in Section 3.2.3. This was done to investigate the effects of varying refresh rates and to check if the algorithm satisfies the requirement of signal continuity at frequency changes. Moreover, this also allowed us to investigate the effect of different refresh rates on waveform quality.

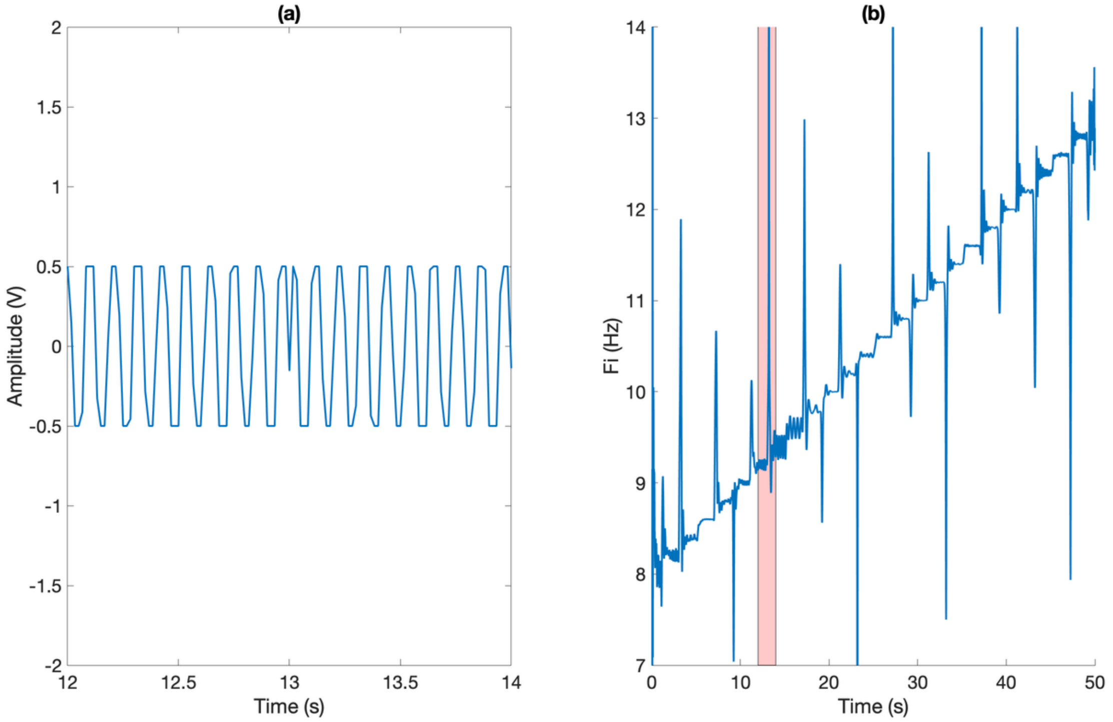

Figure 3.2. Raw waveform (a) and its instantaneous frequency F_i (b) using the square interpolation method. Shaded area corresponds to the time period displayed in (a).

To test this, we ran a simulation in Matlab where this method was used to generate a waveform that changed its frequency from 8–13 Hz, in 0.2 Hz steps (changing every 2s). The change in frequency of this waveform would allow verification of any signal continuity, at these step changes. Then, we repeated this for a 90 Hz and 120 Hz refresh rates, to see the effect different refresh rates would have on the waveform quality.

Firstly, a portion of waveform produced by the interpolation algorithm are shown in Fig. 3.2a. As observed, when there is a step change in frequency (t=13s), there is signal discontinuity present. This is also evident from the instantaneous frequency measurement and the sharp spikes observed, in Fig. 3.2b. Here, the spikes represent step changes of 0.2 Hz in the signal and they are amplified more so due to the noise sensitivity of the derivative operation. Apart from that, the raw waveform look square like, and the intermediate interpolated intensities are clearly visible i.e. the parts where the amplitude is neither 0 or 1. Overall, the trend is for the flicker frequency present to match the one asked for to a high level of accuracy, as long as no frequency transitions are present.

Fig. 3.3 shows the results of varying the sampling frequency (refresh rate on phone) of the waveform, and as observed, with increasing refresh rates, the waveform starts to look more square like. This suggests that phones with higher refresh rates (90 Hz and 120 Hz), would give higher quality square like flickers, due to increased sampling frequency. However, the issue of signal continuity still persists, as expected.

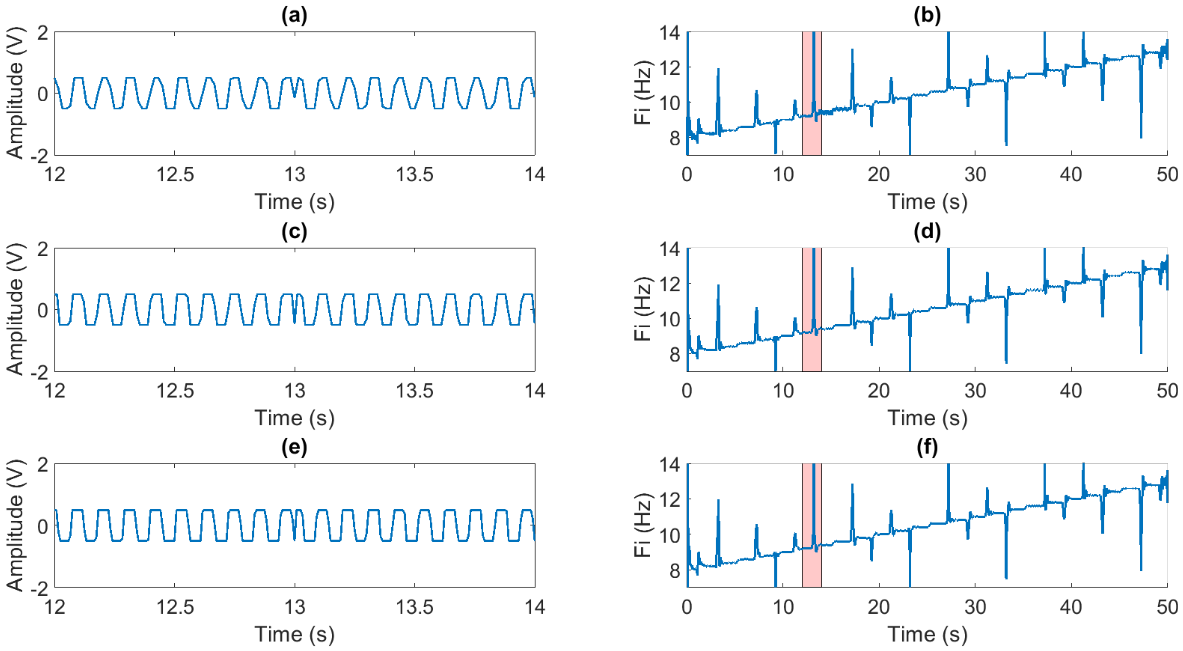

Figure 3.3. Varying sampling frequency (simulated refresh rate) for the square interpolation method. (a) Raw waveforms for F_s=60 Hz. (b) Instantaneous frequency for F_s=60 Hz. (c) Raw waveforms for F_s=90 Hz. (d) Instantaneous frequency for F_s=90 Hz. e) Raw waveforms for F_s=120 Hz. (f) Instantaneous frequency for F_s=120 Hz. Shaded area in red for all right column plots correspond to the time period displayed in the left hand plots.

Results: On-phone testing

After investigating the waveform dynamics from the offline simulations, we learned that this method, due to its signal discontinuity at frequency changes, is not suitable to be embedded into the closed-loop platform later on in Chapter 5, where frequency is expected to change. However, it may still be advantageous to use this technique for open-loop applications; for example, as detailed in Section 3.2.6, square waves were more likely to elicit an SSVEP response [139]. Hence, we did not conduct on-phone testing for the closed-loop mode where we program frequency changes into the app, to test for continuity; instead we simply tested it with fixed frequency in the alpha range.

To do this, we measured 3 minutes of stimulation for each frequency in the range 8–13 Hz (in 1 Hz steps). Three trials were taken for each of the 3 minute measurements. The frequency for each trial (epoch) was measured via obtaining the power spectrum of the signal, and measuring the dominant frequency of the signal (frequency with the highest power). To do this, we used the welch transform (hamming window of 1.3s length and a 50% overlap); with a sampling frequency of 250 Hz, a 1.3s window length gives us a 0.8 Hz frequency resolution, which would be enough to distinguish between the frequencies in the alpha range (8–13 Hz).

As observed from Figure 3.5, the app accurately delivers the square wave stimulation for the entire alpha band here, as evidenced by the R squared error value of 1 between the fitted line and the measured one. Moreover, there was no difference between trials, i.e. an inter-trial standard deviation of 0 Hz, which is why the error bars are not visible on Figure 3.5. Also, observing Fig. 3.4, we see the raw flicker waveform is square-like but noisy;

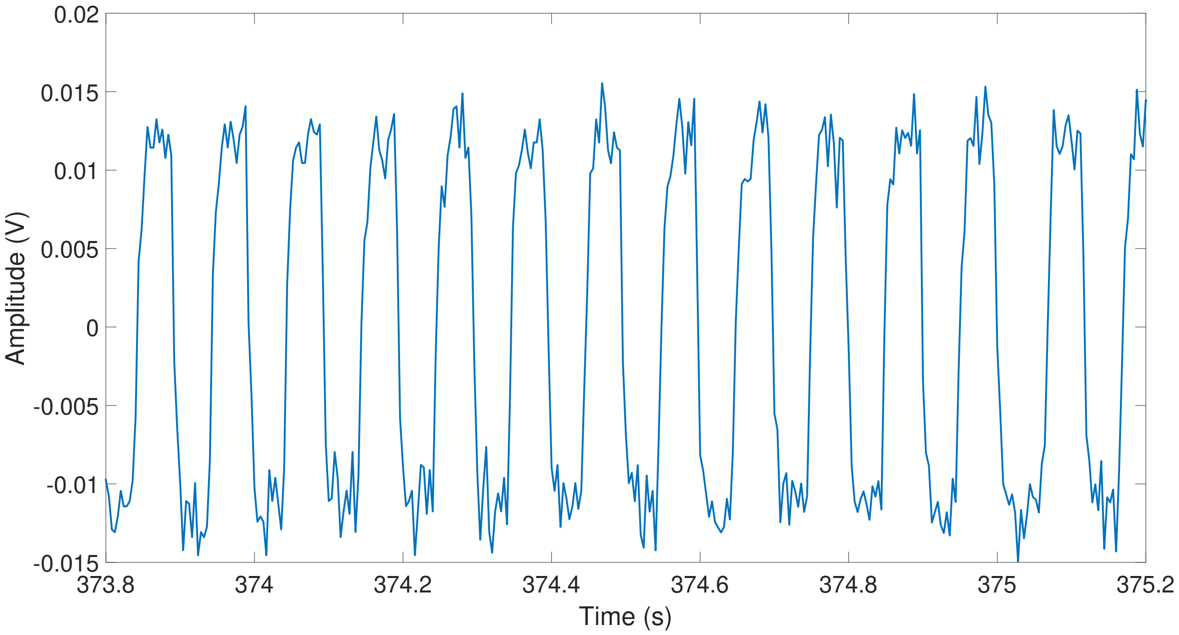

Figure 3.4. On phone testing: Square interpolation method raw waveform.

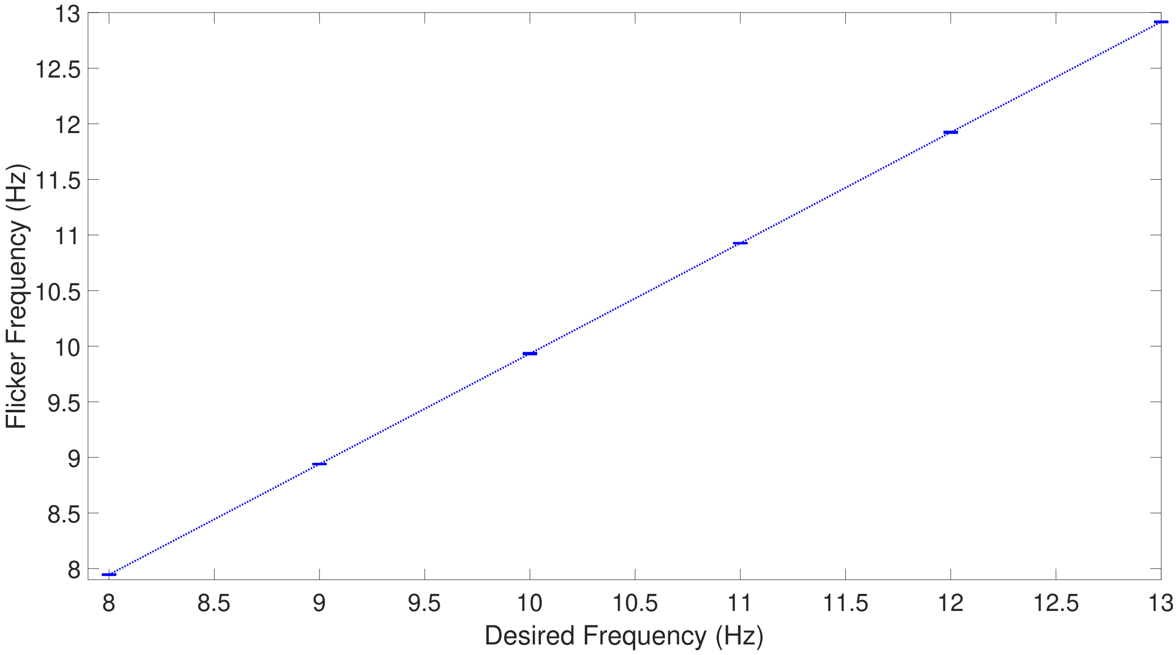

Figure 3.5. On phone testing: Square interpolation method frequency plot. Each marker represents the pre-set frequencies tested, with an error bar indicating the inter-trial standard deviation, which in this case is 0 Hz.

these are noise picked up from external sources (from the test setup) and they may have been picked up due to the high sensitivity of the photo diode sensor used. We confirmed this by measuring the waveform without any stimulus i.e. measuring the noise floor of the test-setup; the peak-to-peak amplitude of this waveform was around 5 mV, which is roughly the same as that of the ripples observed in the flicker waveform in Fig. 3.4. However, these are not an issue, since they are high frequency noise and is removed by the band-pass filtering done in the offline analysis of the waveform, as detailed in Section 3.2.3. Moreover, it must be noted here that the original authors of this method implemented this algorithm on a computer screen with refresh rates of 85 Hz and 120 Hz, and did not test it on a phone [133]. We filled this gap by implementing the algorithm on a phone and showing that for a 60 Hz refresh rate phone, the method could accurately deliver stimulus in the alpha range

(8–13 Hz). Most importantly, they did not test for signal continuity when the frequency changes, as their aim was for it be used in SSVEP based BCI's, which are open-loop by nature [133]. Hence, the method won't be fit for use in closed-loop applications like ours. Groups that have used sine-based waveforms, as discussed in Section 3.2.6, also do not take signal continuity at frequency changes into account, again due to their target applications being open-loop [135], [136]. Hence, these would not be suitable for closed-loop applications but would be relevant for open-loop applications, where signal continuity is not required. To make this work for closed-loop system, we next move onto methods which would ensure signal continuity at frequency changes.

3.2.5 Stimulation method 2: Square sine

A problem with the square interpolation method discussed previously was signal discontinuity when the frequency changes. Both the square-based method and sine-based methods in literature do not attempt to solve this issue, as their target applications are open-loop (SSVEP-based BCI's) [133]–[136], [138]. This then motivates the 'Square sine' method we propose and detail in this Section, which essentially generates a square waveform, while satisfying the signal continuity requirement.

Implementation

This technique stems from the knowledge that a square wave can be approximated by a sum of sinusoids, using fourier expansion. We then add the desired frequency sinusoid, along with its first harmonic, to get an approximation of the square wave. Note that only the first harmonic was used and not the rest, to keep computation cost minimal and to avoid exceeding the nyquist limit, given the screen refresh rates we deal with are pretty low. Since this technique uses a sum of sinusoids to approximate the square wave, we can then satisfy the signal continuity requirement by embedding a phase accumulator into each of the sinusoids, which ensures continuity when the frequency changes. The equation for the generated waveform then is given as,

$$y[i] = \sin(\phi[i]) + \frac{1}{3}\sin(3\phi[i]) \tag{3.2}$$

Where, $y[i]$ is the waveform amplitude at each sample i and $\phi[i]$ is defined as,

$$\phi[i] = \phi[i-1] + \Delta\phi, \quad and \quad \Delta\phi = 2\pi\frac{f}{f_s} \tag{3.3}$$

Here, f and f_s are the stimulation frequency and sampling frequency (refresh rate) respectively and by using a phase accumulator (3.3), phase memory is embedded in the waveform, to ensure signal continuity at frequency changes.

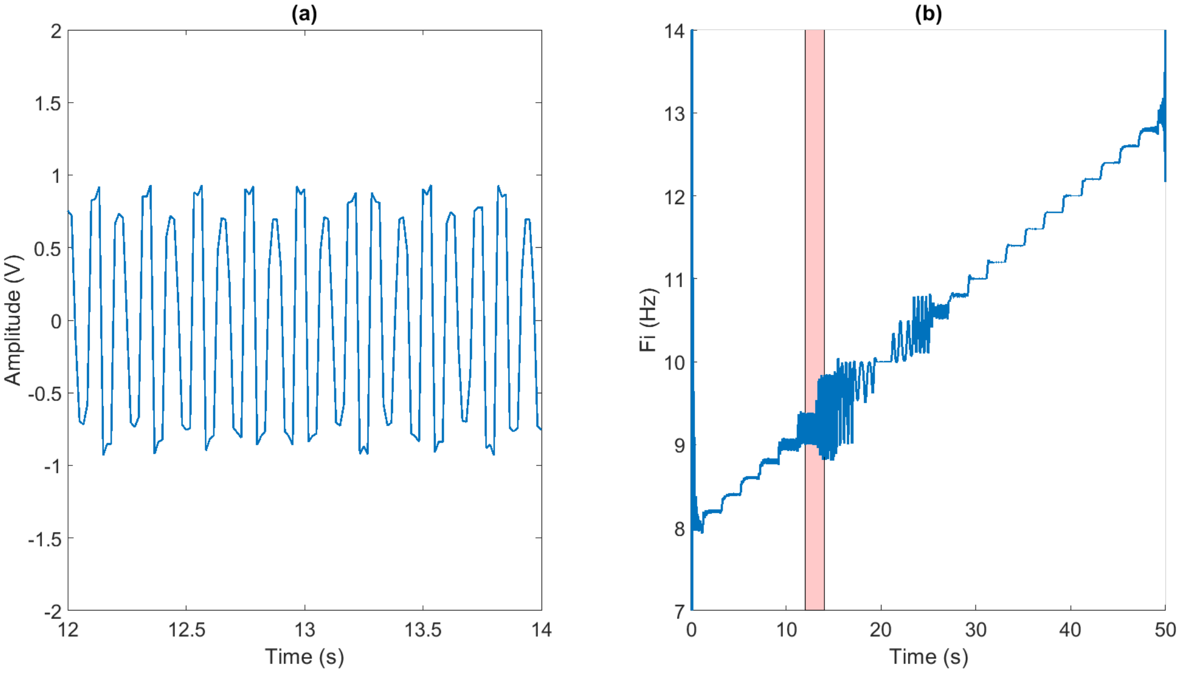

Figure 3.6. (a) Simulated waveform for the square sine method. Frequency changes at t=15 s (0.2 Hz step) (b) Instantaneous frequency F_i of simulated waveform for the square sine method. Shaded area corresponds to the time period displayed in (a).

Results: Offline simulation

As with the square-interpolation method, we first conduct an offline simulation in MatLab, to investigate the effects of varying refresh rates. Just as in Section 3.2.4, we generate a waveform that changed its frequency from 8–13 Hz, in 0.2 Hz steps for a 60 Hz refresh rate, and then repeated the same for a 90 Hz and 120 Hz refresh rates, to see the effect different refresh rates would have on the waveform quality.

As observed from the waveform plot in Fig. 3.6a, we note that for a 60 Hz refresh rate, the waveforms do not look very square like and this is reflected in the instantaneous frequency plots as well, shown in Fig. 3.6b. This is because with a 60 Hz refresh rate, the highest frequency i.e. first harmonic in the generated waveform exceeds the nyquist limit in our band of interest. For example, for the alpha band (8–13 Hz), the first harmonics would range from 24-36 Hz, with the nyquist limit here being 30 Hz (half the sampling frequency). Hence, for the square waves at and above 10 Hz, it does violate this limit and causes aliasing, and for frequencies between 8 and 10 Hz, it is very close to this limit.

Note that the refresh rate also limits the number of harmonics that could be used to obtain a square wave-form. Choosing a higher number of harmonics to approximate a better square wave is only possible when the refresh rate is high enough to accommodate this, and will also incur additional computational complexity. For these reasons, we limited it to just the first harmonic.

Hence, the square-sine method, despite giving signal continuity for a square wave, still fails with a 60 Hz refresh rate due to aliasing and exceeding the nyquist limit. We then hypothesized that this problem is solved with a higher refresh rate, and hence we conducted the

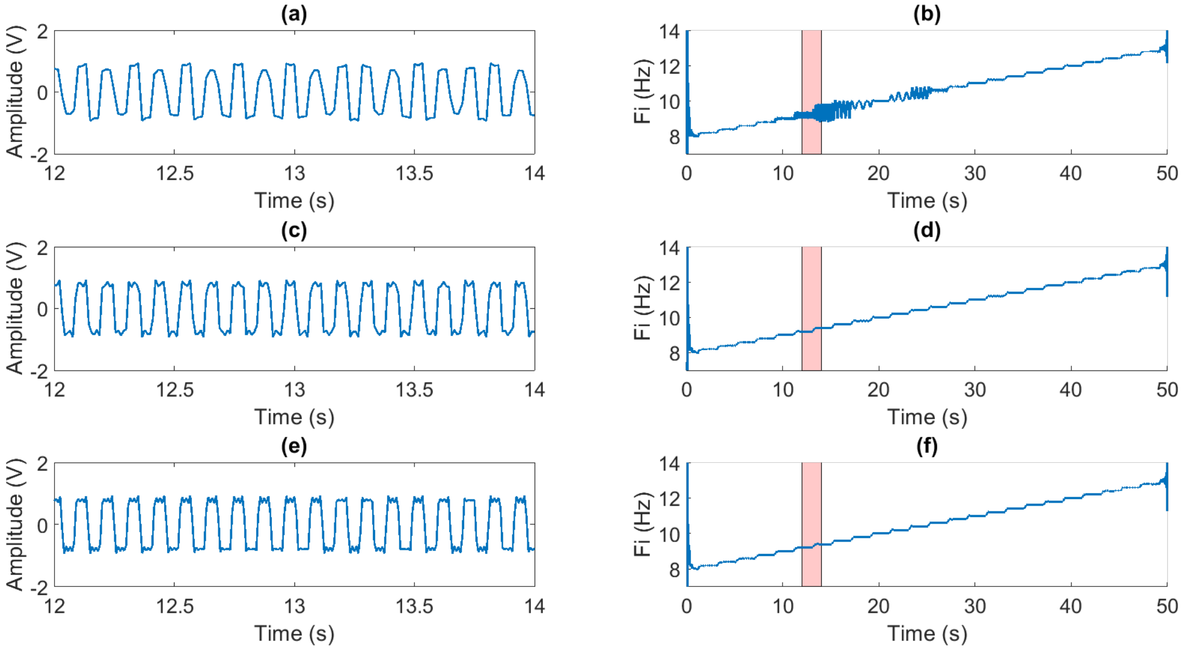

Figure 3.7. Varying sampling frequency F_s (simulated refresh rate) for the square sine method. (a) Raw waveforms for F_s=60 Hz. (b) Instantaneous frequency for F_s=60 Hz. (c) Raw waveforms for F_s=90 Hz. (d) Instantaneous frequency for F_s=90 Hz. (e) Raw waveforms for F_s=120 Hz. (f) Instantaneous frequency for F_s=120 Hz. Shaded area in red for all right column plots correspond to the time period displayed in the left hand plots.

same analysis with a 90 Hz and 120 Hz refresh rates, which may be more common amongst phones of the future. The results for these are shown in Fig. 3.7.

As observed, we see from Fig. 3.7 that for both 90 Hz and 120 Hz refresh rates, the waveforms generated are square like, with the expected instantaneous frequency measurements and sans aliasing effects. This is because for 90 Hz and 120 Hz refresh rates, the nyquist limit is at 45 Hz and 60 Hz, respectively, which is well below the highest frequency contained in our generated waveform. This was not the case for the 60 Hz refresh rate, which is why we observed aliasing there.

In summary then, we note that using the square sine method, we can generate a square waveform with signal continuity at frequency changes. However, this only works when the refresh rate of the phone is high enough (e.g. 90 and 120 Hz) and not for 60 Hz refresh rate ones like the one we use for this thesis, due to the nyquist limit and aliasing. Hence, this method would be suitable for closed-loop applications done with phones of the future with a 90 Hz and 120 Hz refresh rate.

Results: On-phone testing

From the offline simulations, we noted that it is not feasible to generate a square sine waveform for a 60 Hz refresh rate phone, due to the nyquist limit. However, for phones with a 90 Hz or 120 Hz refresh rate, it is possible. Since, we only had access to 60 Hz refresh rate phones for this study, it then made no sense to do an on-phone test, which is the reason why we did not conduct any. However, we do note that this would be possible with phones that have higher refresh rates.

3.2.6 Stimulation method 3: Sine waveform

The previous two methods were used to generate a square waveform, which are known to produce stronger entrainment [139]. However, they were both not optimal, as they either did not satisfy the signal continuity requirement needed for closed-loop applications (square interpolation method) or they did not produce the square-like waveform for a 60 Hz refresh rate phone (square-sine method). Hence, next we explored a third option, which is the use of a sine wave. Sine-waves have been used to generate SSVEP's in literature [135], [136]. For example, Chen et al, used a sine-wave for generating SSVEP's in the range 6–40 Hz, in steps of 2 Hz [135]. They did this on a computer monitor with a refresh rate of 120 Hz, which meant sine waves of up-to 60 Hz in frequency could be generated reliably without exceeding the nyquist limit [135]. However, as with the square-interpolation methods, the target application is open-loop, and not for closed-loop applications, where the frequency is expected to change over time [135].

To overcome this limit, we build upon this method and overcome this limitation via the use of a phase-accumulator. This would have the advantage of satisfying signal continuity via phase memory, whilst also producing a quality waveform. Moreover, unlike the methods in literature, we implement this on a phone, and not computer monitors; more specifically, a phone with a 60 Hz refresh rate, which is much lower than the 120 Hz refresh rate screen monitor used by Chen et al [135], [136]. For a sine wave, at 60 Hz refresh rate, the nyquist limit of 30 Hz is well above the highest frequency component in the alpha band (13 Hz) and hence, we can expect quality signals with no aliasing, generated on a phone.

Implementation

Implementation of a sine wave is straight forward, and we also embedded a phase accumulator for phase memory, which ensures signal continuity at frequency changes. The equation of the waveform is given as,

$$y[i] = \frac{1}{2}\{\sin(\phi[i]) + 1\} \tag{3.4}$$

Where, $y[i]$ is the waveform amplitude at each sample i and $\phi[i]$ is defined as,

$$\phi[i] = \phi[i-1] + \Delta\phi, \quad and \quad \Delta\phi = 2\pi\frac{f}{f_s} \tag{3.5}$$

Here, f and f_s are the stimulation frequency and sampling frequency (refresh rate) respectively. By using a phase accumulator (3.5), phase memory is embedded in the waveform, to ensure signal continuity at frequency changes. Also, note that the amplitude values of $y[i]$ varies between 0 and 1, which is the range accepted by display framework (OpenGL).

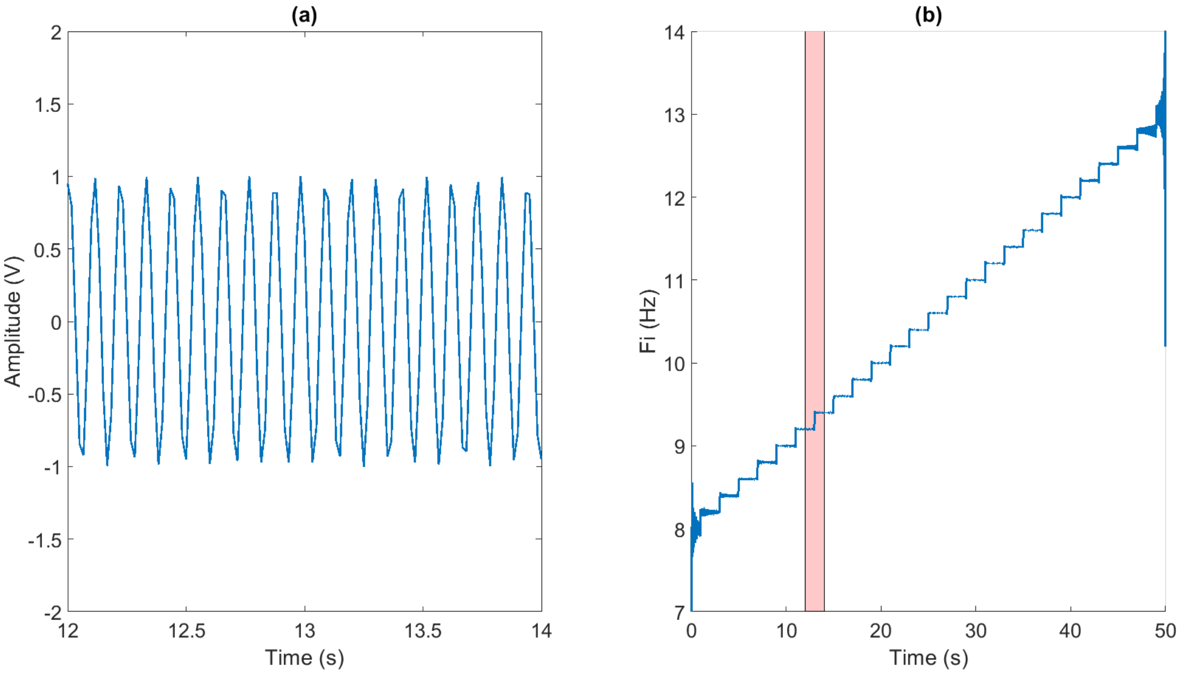

Figure 3.8. (a) Simulated waveform for the sine method. (b) Instantaneous frequency of simulated waveform.Shaded area corresponds to the time period displayed in (a).

Results: Offline simulation

As with the previous two methods, we first conduct an offline simulation in MatLab, to investigate the effects of varying refresh rates. This is done via generating a waveform that changed its frequency from 8–13 Hz, in 0.2 Hz steps for a 60 Hz refresh rate, and then repeating the same for a 90 Hz and 120 Hz refresh rates.

As observed from the raw and instantaneous frequency plots in Fig. 3.8a and 3.8b, respectively, we find that this method produces a waveform that satisfies both requirements: it provides a sinusoidal waveform in the alpha band for a 60 Hz refresh rate, and ensures signal continuity when the frequency changes. The former is possible because the highest frequency components in the alpha band (13 Hz) is well below the nyquist limit of 30 Hz, for a 60 Hz refresh rate. Hence, a quality waveform is generated for the alpha band. Similarly, the signal continuity requirement is satisfied as well, as there is a phase accumulator for phase memory. Hence, implementation wise, the sine method functions the best.

Finally, for completion sake, we also show the effects of varying refresh rates for the sinusoidal method in Fig. 3.9. Here, as expected, no real difference could be observed with increasing refresh rates, as it works well across all. This is because, even for a 60 Hz refresh rate, we know that the highest frequency in the alpha band is still below the nyquist limit, which avoids any degradation of signal quality due to aliasing.

Results: On-phone testing

From the offline analysis, we found that a sine waveform could be generated reliably in the alpha band on a 60 Hz refresh rate phone, whilst ensuring signal continuity. Hence, unlike

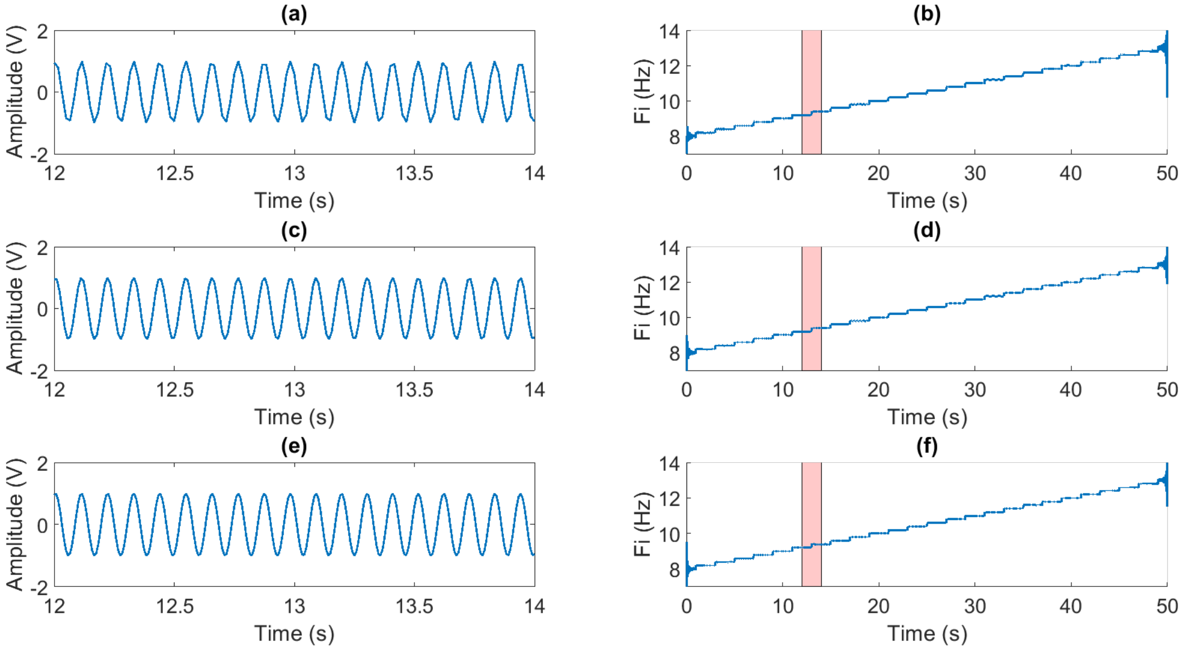

Figure 3.9. Varying sampling frequency (simulated refresh rate) for the sine method. (a) Raw waveforms for Fs=60 Hz. (b) Instantaneous frequency for Fs=60 Hz. (c) Raw waveforms for Fs=90 Hz. (d) Instantaneous frequency for Fs=90 Hz. (e) Raw waveforms for Fs=120 Hz. (f) Instantaneous frequency for Fs=120 Hz. Shaded area in red for all right column plots correspond to the time period displayed in the left hand plots.

for the square interpolation method in Section 3.2.4 where we test using pre-set frequencies on the phone, here we use the stair-case signal (8–13 Hz in 0.2 Hz steps) to test for signal continuity. The stair-case signal would allow us to test for both signal continuity and accuracy. We did not do this for the on-phone implementation of the square interpolation method, as we knew it would not track frequency changes from the offline analysis.

With this, we then ported the sine-method onto the Sony XPeria Z3 phone, and measured the generated waveform and its instantaneous frequency, as detailed in the test methods in Section 3.2.3. The resulting raw waveforms and its instantaneous frequency are shown in Fig. 3.10 and 3.11, respectively.

As observed from both the raw plots and its instantaneous frequency, we note that the app delivers stimulation accurately over the alpha band, as expected. There are no discontinuities when the frequency changes. Here, we also note that the transition jump for the 0.2 Hz step changes in frequencies are not seen in Figure 3.11; instead, they are smoothed out. This is because we used a low pass filter (0.3 Hz cut-off) to smooth out the derivative induced noise in the instantaneous frequency, as detailed in the test methods in Section 3.2.3. On the contrary, for the offline simulations the signals were cleaner without any test-setup induced noise; hence, there was no need to use the smoothing filter, which is the reason why we observe the 0.2 Hz step changes there.

Even though the sine-wave works best for delivering stimulation in the alpha-range on a 60 Hz refresh rate phone, the trade-off made here would be in efficacy; sine-waves has been shown to produce weaker entrainment than square waves [139]. Teng et al. conducted a study where they provided both a square-wave and sine-wave visual stimulus from an LED for frequencies in the range 11-22 Hz [139]. Here, they measured the accuracy of both wave-

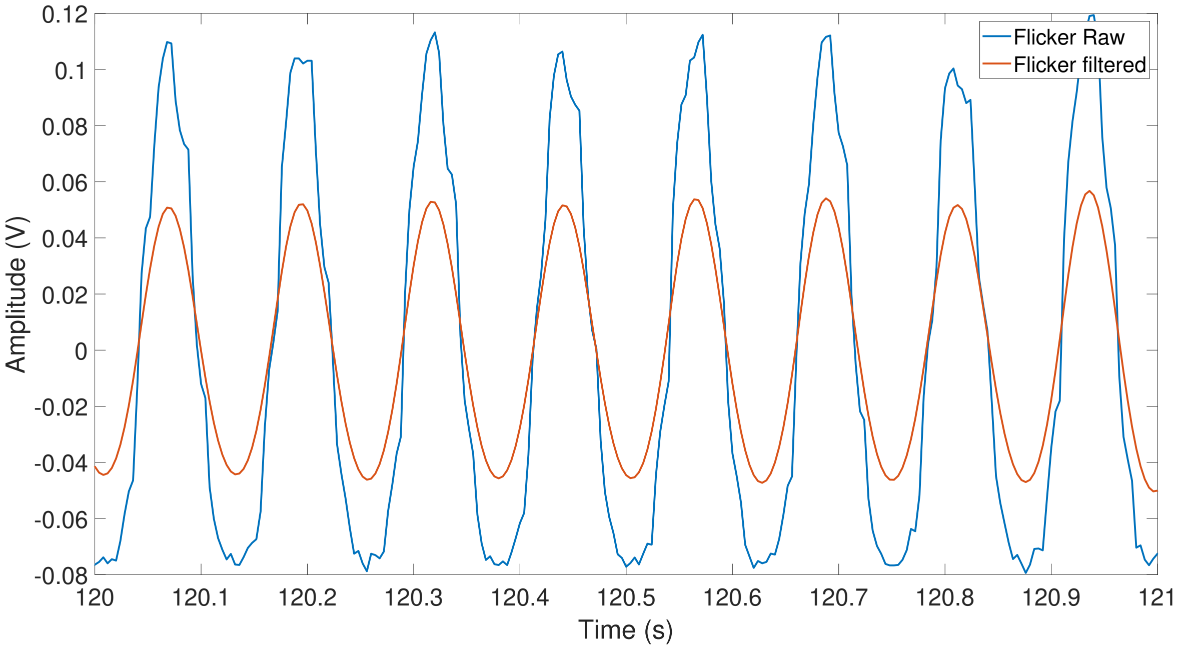

Figure 3.10. On phone testing: sine method raw waveform.

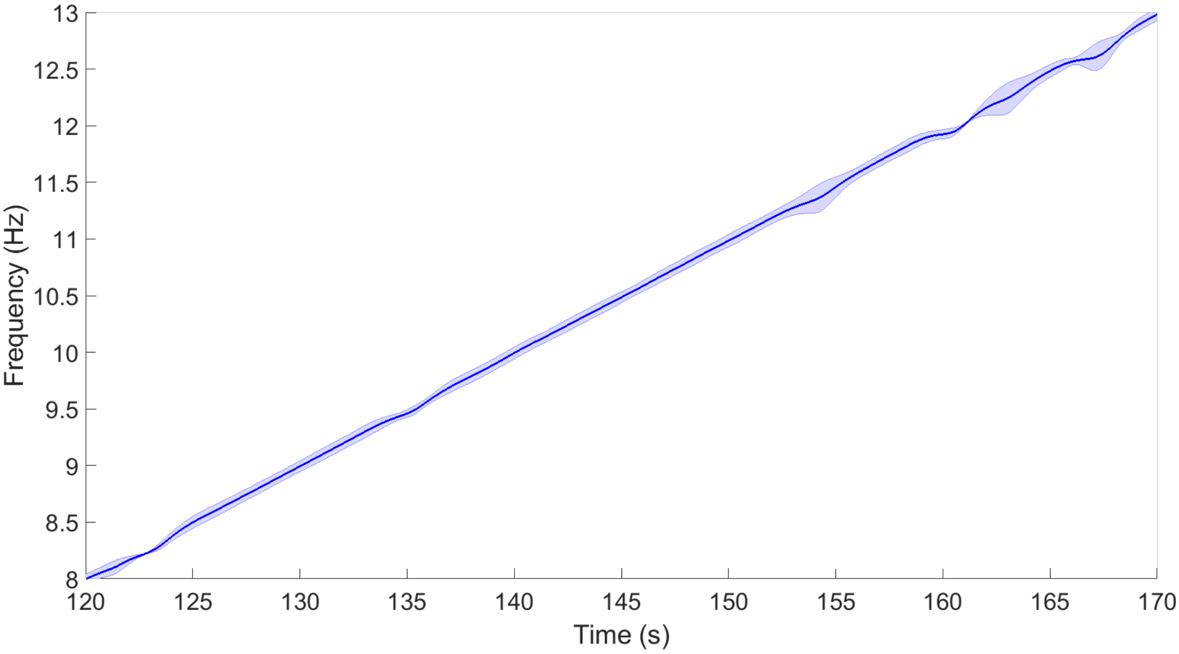

Figure 3.11. On phone testing: sine method instantaneous frequency plot. The shaded area is the inter-trial variance.

forms, with the accuracy being defined as the ratio between the number of trials where the $1f$ component of the SSVEP was detected and the total number of trials. For each trial, 8s of stimulation was provided, followed by a resting period. With this, they reported the square-wave to be better at eliciting SSVEPs (90.8% accuracy), compared to the sine-wave (75% accursacy) [139]. In other words, they showed that the square waveform is more likely to entrain the brain and would be a better modality for entrainment purposes, even though the sine-wave could elicit them too. Hence, this trade-off should be kept in mind.

So, in summary, here we showed that the sine stimulation delivered accurate waveform over the alpha band, whilst preserving signal continuity at frequency changes, on a 60 Hz refresh rate phone. This then makes it a suitable waveform to be used in a closed-loop system, where signal continuity is required.

3.2.7 Discussion

In this Section, we investigated three different waveform generation methods for the visual stimuli: square interpolation, square sine and sine method. The first two produces a square waveform, while the last a sinusoidal one. Here, we compare these methods and inform our choice of method, for use in both the open-loop and closed-loop mode. For open-loop mode, the method should provide accurate waveform over the alpha band, for a 60 Hz refresh rate phone. In addition to this, the closed-loop mode has one additional requirement, which is ensuring signal continuity, at frequency changes, so that the stimulation waveform could be adapted over-time to the on-going EEG. With these requirements in mind, we then compare the three methods.

Comparison of methods

Fig. 3.12 summarizes the performance of the three methods using the stair-case signal, to aid comparison. Out of the first two methods, which are used to generate a square waveform, the square interpolation methods produces a better square waveform, but lacks phase continuity, making it unsuitable for closed-loop applications. These are evident from the spikes observed, when the frequency changes, as seen in Figure 3.12b. On the other hand, the second method (square-sine), overcomes this challenge by generating a square wave via summation of two sine waves (the fundamental frequency and its first harmonic). This however does not work for a 60 Hz refresh rate, since the first harmonics in the alpha band exceeds the nyquist limit, causing aliasing and a poor quality square waveform. This is evident from the noisy instantaneous frequency in the plots. However, we did observe from our offline simulations in Section 3.2.5, that this would not be the case for higher refresh rate phones (90 Hz and 120 Hz), and for these, the square sine method would perform well producing a neat square wave, whilst also satisfying the signal continuity requirement. However, for this project, we are targeting 60 Hz refresh rate phones, as this was the phone that came coupled with the wireless amplifier we used in Chapter 5. In contrast, 90 Hz and 120 Hz refresh rate phones might take a while to be more wide-spread, due to power consumption problems that remain to be solved; even when more readily available, it would still be beneficial turn the refresh rate down as this could reduce power consumption and prolong battery life. Moreover, it must also be noted that for these higher refresh rate, more harmonics may be added to produce a more square like waveform, but this again is limited by the nyquist limit and also may add more computational complexity.

With all these in mind, we then proposed using the square interpolation method for the open-loop mode, as signal continuity is not required for open-loop modes, and this method generates a square waveform accurately over the alpha band on a 60 Hz refresh rate phone, as summarized in the plots in Fig. 3.5, from Section 3.2.4. The authors of the algorithm

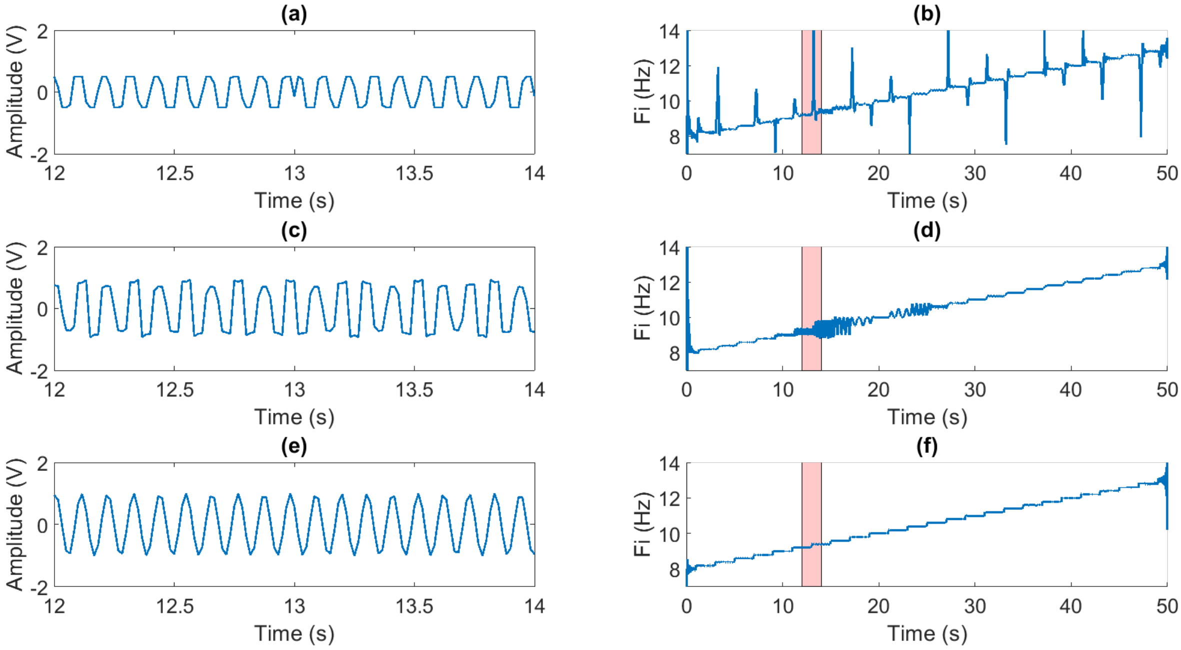

Figure 3.12. Comparison of the 3 methods: Simulated waveforms. (a), (c), and (e) are a small trace of the raw waveform using the square-sine method for sampling frequency F_s of 60Hz, 90Hz and 120Hz. (b), (d) and (f) are the instantaneous frequency F_i for sampling frequency 60Hz, 90Hz and 120Hz, respectively.Shaded area in red for all right column plots correspond to the time period displayed in the left hand plots.

used this method on a 85 Hz and 120 Hz computer monitor to elicit SSVEP's, and we took this further by showing that it works on a 60 Hz refresh rate phone too [133]. Moreover, as square waves have shown to be more likely to elicit entrainment, we then chose this method for the open-loop mode [139].

The third method was a simple sinusoid, which also have been used by other groups, as detailed in Section 3.2.6 [135], [136]. However, unlike other groups where the target application was open-loop SSVEP based BCI's, we advanced this method to be usable for both closed-loop applications and for an on-phone implementation [135], [136]. Here, the highest frequency component in the alpha band (13 Hz) is far below the nyquist limit of 30 Hz, for a 60 Hz refresh rate phone, making it suitable for the on-phone implementation. Moreover, to make it usable in closed-loop applications, we used a phase-accumulator, to ensure signal continuity at frequency changes (see Fig. 3.11). Hence, out of the three methods, this was the one we deemed most fit for the closed-loop mode, as it ensures both signal continuity and a quality waveform, for a 60 Hz refresh rate phone.

It must also be highlighted here that, to the best of our knowledge, only one study exists that used a phone, and not a computer monitor to generate the SSVEP signal [138]. This was a study by Wang et al, where they used a Samsung Galaxy S phone (refresh rate of 55 Hz), to generate a square wave for a 11 Hz signal [138]. They only tested it for a 11 Hz signal, which was an integer divisor of the refresh rate, negating any need for approximation methods [138]. Aside from not covering the entire alpha range, they are also not suited for closed-loop applications, as the target application was opne-loop (SSVEP based BCI).

Hence, in light of these, our contribution here are two folds: firstly, we demonstrated that the square interpolation method works reliably to generate flicker in the alpha range (8–13

Hz) on a 60 Hz refresh rate phone and secondly, we showed that with the use of a phase-accumulator, we can use the sine-method to ensure signal continuity at frequency changes, which makes it fit closed-loop systems.

3.2.8 Summary

In this Section, we investigated three stimulation methods for the visual stimulation, to be used under the constraints of a 60 Hz refresh rate phone. We noted that the square interpolation method is ideal for open-loop applications since its accurate over the alpha band, and provides stronger entrainment than a sine waveform [139]. For the closed-loop method, we chose the sine method, as it is able to satisfy the signal continuity requirement as well, despite providing weaker entrainment. We also note that in the future, if 90 Hz and 120 Hz refresh rate phones that are compatible with real-time streaming EEG hardware will become more common-place, then the square sine method would be the ideal choice for the closed-loop mode, as it will satisfy the signal continuity requirement, whilst avoiding the trade-off in efficacy one would have to make using a sinusoidal waveform. Most importantly, for both modes, we provide a smart-phone based implementation, unlike the many lab-based ones available today [133]–[136].

3.3 Auditory entrainment

3.3.1 Introduction

Auditory stimulation is provided in the form of binaural beats, as detailed in Section 2.3.4. Most binaural beats studies use PC-based softwares to generate the stimuli and are used for open-loop applications. Examples include: Lee et al. who used Gnaural, an open-source binaural beats generator software, to generate 6 Hz binaural beats for inducing sleep; Ross et al. and Lopez et al. both of whom used Matlab for generating 40 Hz binaural beats for their studies in enhancing learning and investigating the efficacy of binaural beats in increasing EEG power (all EEG bands), respectively; and Reedijk et al. who used Audacity to generate 10 Hz and 40 Hz beats for their study on improving creativity via binaural beats [50], [52], [54], [60]. Note that all these are open-loop applications using pre-set beats with fixed frequency and no study exists to the best our knowledge, which uses binaural beats in the context of a closed-loop system, where the frequency is expected to change.

Although most studies use PC-based software to generate the beats and none exists using a phone, there are various apps available in the market for binaural beats. Two such examples are ‘Brain waves - Binaural Beats’ (1m+ downloads) by MynioTech Apps, and ‘Binaural Beats Generator’ (50k+ downloads) by TMSOFT [140], [141]. Both can generate beats in the whole EEG band range (1–50 Hz) but they are all unsuitable for closed-loop applications, as the frequncies are fixed and pre-set.

With these in mind, we then developed our own on-phone implementation of the binaural beats, where the contribution would be usability in closed-loop mode, where the frequency is expected to change. As in Section 3.2, we used a phase-accumulator to achieve this. This then allows us integrate these stimuli into the closed-loop platform in Chapter 5. This would not be possible with the apps available in the market today, as these have pre-set beat frequencies that does not change over-time, in addition to being not modifiable. Hence, in this Section, we discuss our implementation of the binaural beats stimulus and characterize its on-phone accuracy.

3.3.2 Requirements and specifications

With regards to specs, we aimed to stimulate over the alpha band (8–13 Hz) for both the open-loop and closed-loop modes. It must be noted here that frequencies above the alpha band could be implemented unlike the visual case where limits existed due to refresh rate. Regardless, we still narrowed down our focus to just the alpha band, since that's what the chronic pain application driver requires, as detailed in Section 2.6.3. With that in mind, the specs are then summarised as follows:

- **Frequency range: 8–13 Hz** The analgesic effects for chronic pain were observed in the alpha band (8–13 Hz), as detailed in Section 2.6.3 [6], [8], [10]. Hence, the alpha range frequency spec.
- **Signal continuity at frequency changes** Despite this chapter focusing on the open-loop app, the aim is to eventually embed this waveform generation method into the closed-loop app later on in Chapter 5. Here, the Individual Alpha Frequency (IAF) is expected to change over time, which we also showed in Section 4.2.3 [118]–[120]. Hence, we need to ensure that there would be no discontinuity in stimulus waveform when frequency changes in the alpha band, to be fit for closed-loop application later on in Chapter 5.

3.3.3 Test methods

Tone frequency measurements were taken using a National Instruments USB-6212 data acquisition system connected to the phone headphone port via a 3.5 mm stereo audio cable. This setup is shown in Fig. 3.13. The frequency of channel 1 (left channel) was kept constant at 400 Hz, to be the carrier frequency. The frequency of channel 2 (right channel) was then varied from 408-413 Hz, to provide a 8–13 Hz beat range. Note that here the carrier frequency of 400 Hz was chosen, as binaural beats are known to be best percieved when the carrier frequency is between 400 and 500Hz. With this setup, we characterized accuracy for two modes: open-loop and closed-loop.

For the open-loop mode, the beat frequency was pre-set in the range 8–13 Hz, in steps of 1 Hz. 3 trials of 3 minute recordings were then taken for each frequency. To analyse the dom-

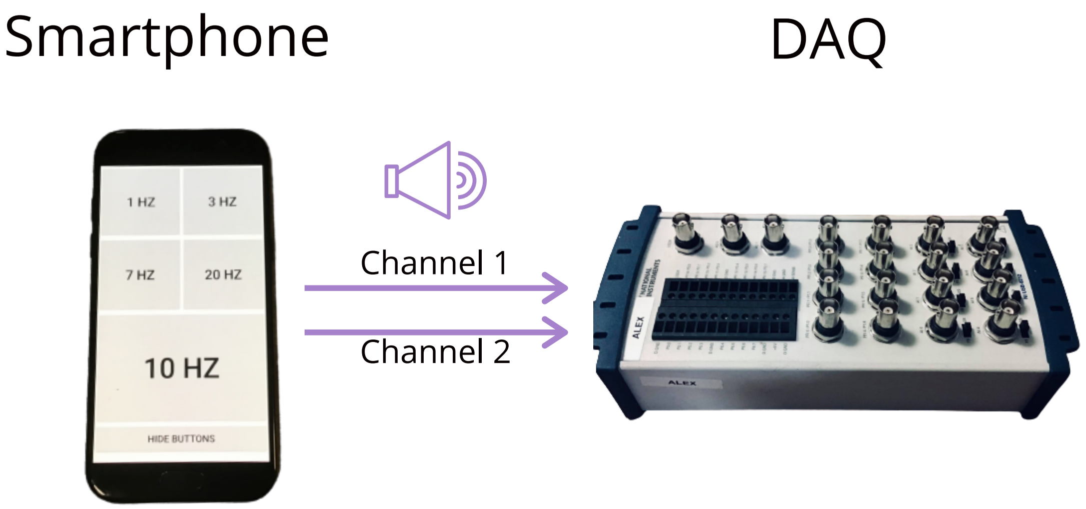

Figure 3.13. Test setup for continuous auditory stimulus.

inant frequency of each 3 minute recordings, we used the welch transform (1.3s hamming window with a 50% overlap) to obtain the power spectrum of the signal from each channel, per recording. The dominant frequency for each channel's waveform was then the frequency with the highest power. With these, the dominant beat frequency could be obtained, which is the difference between the dominant frequencies between the two channels.

For the closed-loop mode, we used a stair-case signal for a 3 trial run, where the frequency varied from 8–13 Hz in 0.2 Hz steps; this allowed us to test for any signal discontinuity, at frequency changes. Here, just like in the visual case, we calculated the instantaneous frequency of each channel via using the hilbert transform to get the instantaneous phase first, after which its derivative is taken to get the instantaneous frequency. We then used a low pass filter (IIR butterworth order 2 with 0.3 Hz cut-off) to filter out the derivative induced noise. With these, we then calculated the beat frequency f_b as:

$$f_b = f_1 - f_2 \tag{3.6}$$

Where f_1 and f_2 are the instantaneous frequencies of channels 1 and channel 2, respectively.

3.3.4 Stimulation method: Binaural beats

Unlike the visual stimulation, where we tested multiple waveform generation methods, binaural beats are by nature sinusoidal; hence, we did not explore various waveform options here. Moreover, no offline simulations were necessary, as we do not operate under the refresh rate constraint here, as was the case for the visual stimuli. Hence, we tested for accuracy starting straight from the phone.

Implementation

To implement binueral beats on a phone, different carrier tones are played to each ear, with the tone of these being adjustable and the difference between the carrier frequencies being the stimulation (beat) frequency. As a repetitive tone it would be possible to pre-save binaural beats for different settings and to simply re-play these. However this introduces discontinuities when looping the pre-saved tones, when used in closed-loop mode, where frequency is expected to change. Instead our tones are generated in real-time using the AudioTrack class in Android Studio. After generating the samples for one cycle for the frequency of interest, they are written to an audio buffer, through which the sound data is streamed. This process is set to loop infinitely, until the user stops the stimulus via the app GUI. Finally, the sampling frequency used to generate the tones were 44100 Hz which is typical in audio applications. Moroever, to ensure signal continuity when the frequency changes, a phase accumulator was embedded into each of the sinusoids generated. The waveforms x_1 and x_2 for channels 1 and 2, respectively, is then given as:

$$x_1[i] = \sin(\phi_1[i]), x_2[i] = \sin(\phi_2[i]) \tag{3.7}$$

Where, $x_1[i]$ and $x_2[i]$ is the waveform amplitude at each sample i for channels 1 and 2, respectively. Moreover, $\phi_1[i]$ and $\phi_2[i]$ are defined as,

$$\phi_1[i] = \phi_1[i-1] + \Delta\phi_1, \quad and \quad \Delta\phi_1 = 2\pi\frac{f_c}{f_s} \tag{3.8}$$

$$\phi_2[i] = \phi_2[i-1] + \Delta\phi_2, \quad and \quad \Delta\phi_2 = 2\pi\frac{f_c + f_b}{f_s} \tag{3.9}$$

Here, f_c, f_b and f_s are the carrier frequency, beat frequency and sampling frequency, respectively, By using a phase accumulator (3.8 and 3.9), phase memory is embedded in the waveform, to ensure signal continuity at frequency changes.

On phone testing

For the on-phone tests, firstly the open-loop mode tests were conducted. This is essentially sweeping the frequency as pre-sets in the range 8–13 Hz, in 1 Hz steps. Results are shown for the measured stimulation frequencies in Fig. 3.14 and a sample trace of the binaural waveform in Figure 3.15.

As observed, for the open-loop case in Fig. 3.14, the app functions accurately, with an R squared fit of 1 between the fitted line and the data points. Moreover, there was no differ-

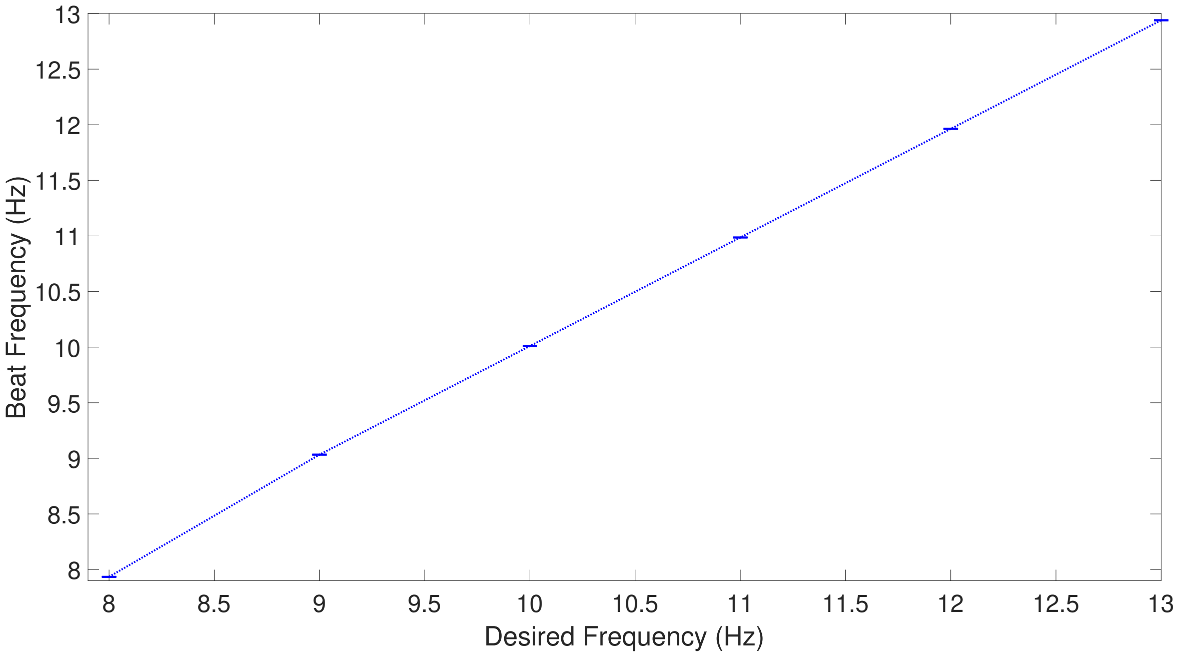

Figure 3.14. Audio stimulation open-loop mode. Each marker represents the pre-set frequencies tested, with an error bar indicating the inter-trial standard deviation, which in this case is 0 Hz.

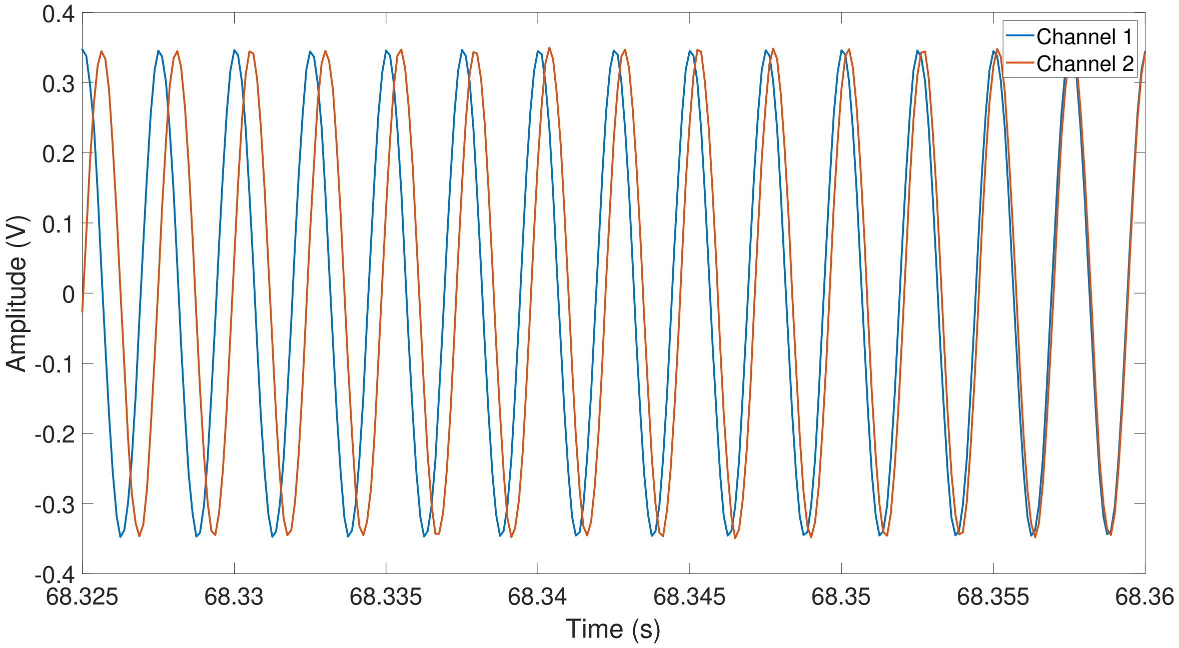

Figure 3.15. Sample binaural waveform trace.

ence in performance between trials, hence the inter-trial standard deviation of 0 Hz. This then confirms accurate functionality over the alpha band.

For the closed-loop mode, we tested for signal continuity at frequency changes using the stair-case signal. The results are presented in Fig. 3.16.

As observed, for the closed-loop mode, the app is able to deliver stimulation accurately, without any discontinuities when the frequency changes. This is achieve via the use of a phase accumulator for phase memory. Hence, we then have confidence that the app can deliver stimulation accurately, for use in the closed-loop mode later on. We also note that we do not observe the 0.2 Hz transition jumps in Figure 3.16 and this is expected since we use

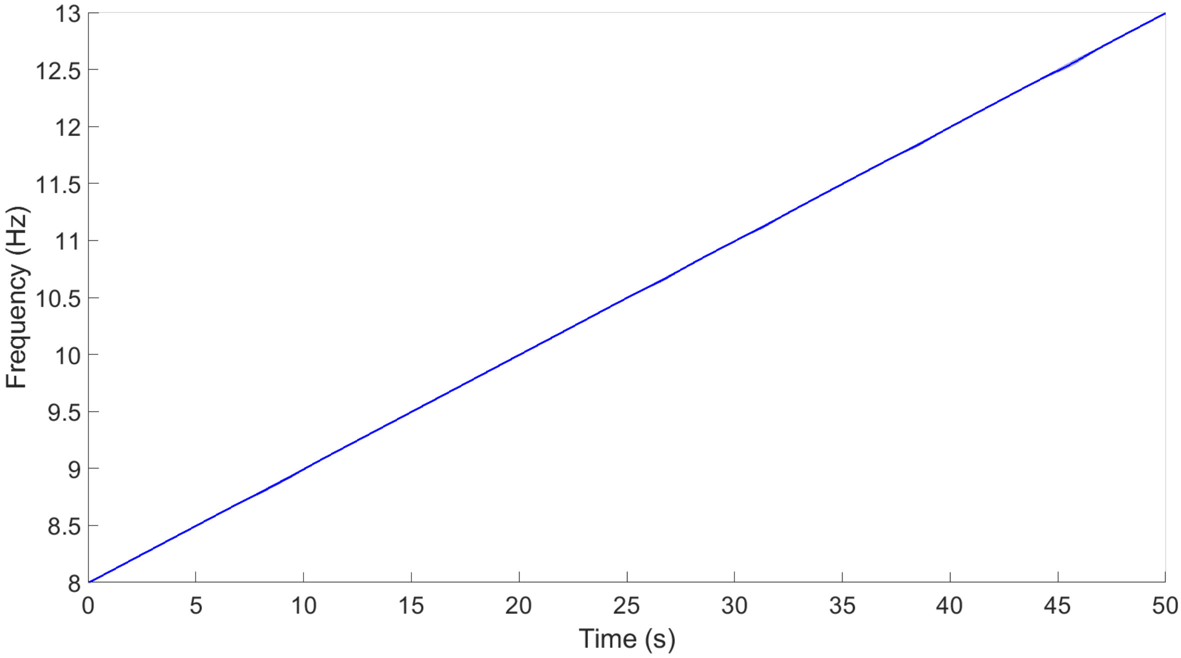

Figure 3.16. Audio stimulation continuous mode. Frequency of signal changing in 0.2 Hz steps from 8–13 Hz. The shaded blue area is the inter-trial standard deviation.

a low pass filter for filtering out the derivative induced noise, which would also then smooth out these 0.2 Hz transition jumps.

As discussed previously, other studies using binaural beats use mostly PC-based software to generate the stimulus and are targeted for open-loop applications: for example, Gnaural (Lee et al.), Matlab (Ross et al. and Lopez et al.) and Audacity (Reedijk et al.) [50], [52], [54], [60]. All these were open-loop applications using pre-set frequencies and to the best of our knowledge, no study exists using binaural beats in the context of a closed-loop system, where the frequency is expected to change. Moreover, smart-phone apps for binaural beats available on the app stores, are un-modifiable and again designed for open-loop stimulus only [140], [141]. In contrast to these, we provide an app-based implementation of binaural beats, which is usable for closed-loop applications, where frequency is expected to change. We achieved this via the use of a phase-accumulator to ensure signal continuity when frequency changes. This then makes the stimulus fit for use in our closed-loop platform, later on in Chapter 5.

3.3.5 Summary

In this Section, we showed that the app can deliver auditory stimulation over the alpha band accurately both for the open-loop and closed-loop modes; for the closed-loop mode, the requirement of signal continuity at frequency changes was achieved, via the use of a phase accumulator.

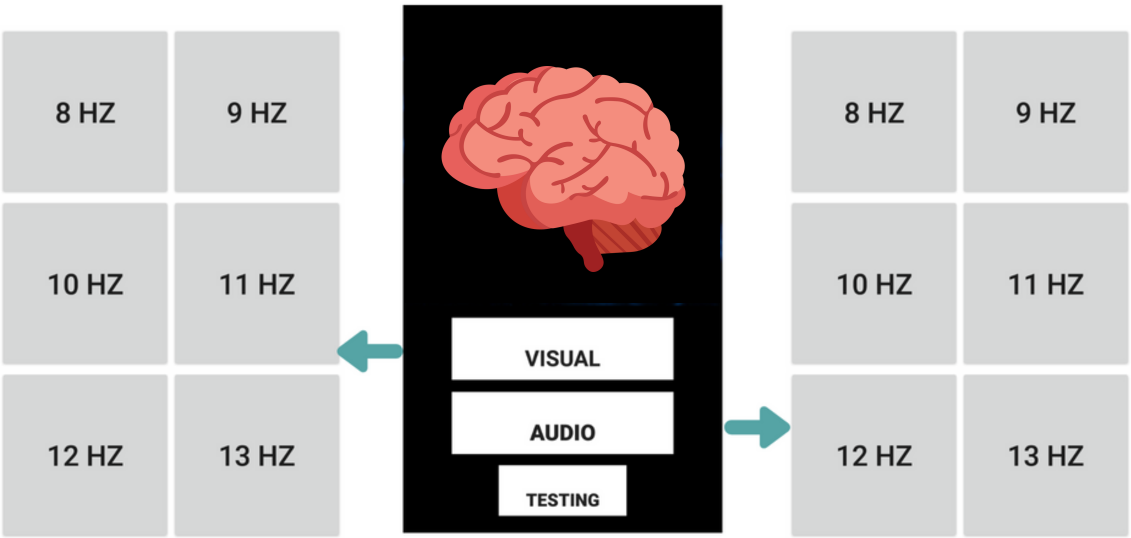

Figure 3.17. Screenshot of the open-loop stimulation app.

3.4 Overall open-loop platform

In this Section, we describe the Graphical User Interface (GUI) for the open-loop app, in addition to briefly highlighting clinical work where our app is being currently used for, by our collaborators.

3.4.1 Graphical User Interface (GUI)

Firstly, screenshots of the app designed is shown in Fig. 3.17. The home page at the centre in Fig. 3.17 contains the options for the different forms of entrainment available with this app and upon clicking on the relevant one, it takes you to the appropriate stimulation mode. For each of the modes, pre-set frequencies for the alpha band range (8–13 Hz) are available for the user to choose. For the audio mode, the user simply has to press the button to start playing the sound, and then can press again to stop. For the visual mode, when the user presses a button, the button layouts will become invisible, after which the phone flicker at the specified frequency will be presented in full screen. To bring back the button view to choose another frequency, the user simply has to tap on the screen again. The user interface design process also involved getting feedback from chronic pain patients via workshops conducted at the Royal Salford Hospital by our collaborators, where the patients were given the opportunity to use the app; features such as big buttons and large texts for the pre-set binaural beats are examples of features added after such feedback.

3.4.2 Current use-cases

In this Chapter, we presented our on-phone audio-visual stimulation methods. In addition to it being usable in our closed-loop platform in Chapter 5, we also highlight here that the open-loop app itself is a contribution. It has been designed to meet technology needs for our collaborators in chronic pain and as such, are being used by them currently for their clinical work [95].

Illustrative Quotes
”Yeah it was so easy to use, you know, you just had to tap the screen . . . it was just brilliant, brilliant”
”1 Right now they don’t have many options for you. They just say we can put you on medication and that’s it really. And that’s quite upsetting”
”It was miraculous. I reckon I’ve gone from, probably from reliant on anti-inflammatories to 80% less . . . Yeah, I’m really keen to know about the science a bit more and how it works, but at the moment I’m just happy feeling confident that it works”
”You could just go off into your own little world with it . . . It was comfortable, you could just relax into it and let time pass by quite happily”
”I did them both [visual and auditory stimulation] lying on a sofa, with my head on a pillow on one arm and spread out across the sofa. So that was fine. In fact, once or twice I fell asleep it was that peaceful”

Table 3.1. Illustrative quotes from patient feedback on the app.

An example of this is a qualitative study done on our app. As all work in the chronic pain application driver has been done using lab-based setups, the first step towards home-based use is to conduct a qualitative study, to see if the app-based approach is appealing to chronic pain patients [6]–[8], [10], [11], [14]. As such, Helen et al. conducted a qualitative study on the acceptability and usability of this app, with results published [95]. To do this, they tested the app on 15 chronic pain patients and asked them to use the app for 10 minutes every day, in their homes for a total of 4 weeks. Telephone interviews were then conducted at the end to discuss feedback; more details on the methodology and frameworks used for analysing and structuring the feedback received, are described in their paper [95]. A sample set of quotes from their feedback, is given in Table 3.1. In general, the patients seemed interested in trying alternative therapies like these, due to their negative attitudes towards pharmacological based solutions [95]. Moreover, they found the user-interface of the app simple and easy to use. Regarding modalities, there were no clear preference for one over the other; both auditory and visual stimulation modes were met with equal reception [95].

We do not go into more detail here, as the intention here is to simply highlight that this app is currently being used for clinical work and is not just a means to an end, for the closed-loop platform.

3.5 Conclusion

In this Chapter, we first detailed the stimulation methods for both the visual and auditory modes. For each of the modes, we aimed to deliver accurate stimulation in the alpha band (8–13 Hz), whilst ensuring signal continuity at frequency changes, for it be used in closed-loop mode, later on.

For the visual mode, we explored three methods: square interpolation, square-sine and sine. With the 60 Hz refresh rate constraint in mind, we then concluded that the square interpolation technique is ideal for open-loop application, as it delivers a clean square wave at a 60 Hz refresh rate. However, for closed-loop applications, we concluded the sine method would be best, as it ensures signal continuity via the use of a phase accumulator. The addition of a phase accumulator here is a new contribution, as other work in literature did not do this, since they were targeting open-loop applications, and not closed-loop ones like ours. However, the use of a sine wave would come at a slight trade-off in efficacy, as square waveforms are more likelier to elicit entrainment, than its sine counterpart [139]. Finally, the square sine method may overcome this trade-off via providing a square wave whilst also ensuring signal continuity but it requires phones with a refresh rate of 90 or 120 Hz, due to the nyquist limit. Hence, it may be a better choice for use in the future, when higher refresh rate phones may become the norm.

For the auditory stimulation, we implemented binaural beats via two separate sinusoidal waveforms provided to each ear. For each of the sinusoids, a phase accumulator was used to ensure signal continuity, which again provided us with accurate stimulation over the alpha band, for both closed-loop and open-loop modes. As with the visual case, the addition of a phase accumulator here is a new contribution, to enable closed-loop applications where frequency is expected to change over time.

With the stimulation functionality complete, these could be now used to build the closed-loop platform. However, before doing so, we next detail the feature extraction algorithms and characterize its accuracy in Chapter 4.

Chapter 4

Pre-requisites to closing the loop: phase-frequency extraction

4.1 Introduction

In this Chapter, we conduct various offline experiments to validate and characterize the feature extraction part of the closed-loop system. Firstly, we characterize the EEG signal for its various properties such as amplitude, frequency and frequency rate-of-change in Section 4.2, after which we feed this data into the tuning methodology of a Phase Locked Loop (PLL) in Section 4.3. After laying out the PLL tuning methodology, we then characterize its performance in tracking EEG phase for all bands in continuous mode (general purpose), for the SO band in discrete mode (sleep engineering application) and for frequency matching in continuous mode (chronic pain application), in Sections 4.4, 4.5 and 4.6, respectively. This then provides a solid foundation to build upon, for the real-time on-phone implementation of the closed-loop platform later on in Chapter 5.

4.1.1 Specifications for closed-loop platform

Here, we define our specifications for three different modes, each of which are application dependent. These are:

- **EEG phase for closed-loop continuous applications (General use-cases)** Here, we aim to show that the PLL can track EEG phase across all bands accurately and better than the state-of-the-art. The state-of-the-art here would be a study by Mansouri et al., where they showed offline that their FFT based algorithm can track phase across all EEG bands, with the specific accuracy values detailed in Table 4.11. Details of this algorithm is given in Section 2.5.4 [44]. We aim to exceed the accuracy values obtained here, via the use of a PLL. This would then pave way for a variety of applications outside the ones we focus on for this thesis: sleep engineering (SO band) and chronic pain (alpha band).
- **EEG phase for closed-loop discrete stimulation (Sleep engineering)** Here, we need to ensure accurate phase tracking functionality over the SO band (0.5–3 Hz), in particular for tracking slow oscillation phase, at discrete points for the sleep engineering

application, as detailed in Section 2.6.2. Other groups have used PLL on lab-based systems for the sleep engineering application; for example, Santostasi et al. achieved a phase error of 0±25.6°, in tracking slow oscillations [2]. We aim to achieve comparable accuracy tuning our PLL offline on a PC. However, it must be highlighted that the contribution is not in the offline analysis (to tune the PLL) in this chapter, but later on in Chapter 5, where we port this PLL onto a phone, to provide the first mobile phone based platform, for the sleep engineering application. More details of the state-of-the art are given in Table 2.1 and the on-phone implementation in Section 5.3 of Chapter 5.

- **EEG frequency for closed-loop continuous stimulation (Chronic pain)** Here, we need to ensure EEG frequency is accurately extracted over the alpha band in a continuous fashion. This would be for use in the chronic pain application, where frequency matched alpha stimulation is hypothesized to improve efficacy, as detailed in Section 2.6.3. Also, as detailed in Section 2.6.3, there are no state-of-the-art to compare the accuracy to. Hence, the results we obtain here would be the first-in kind, which could be used as a bench-mark for future platforms.

4.1.2 Data-sets used

In this Chapter, we make use of two different data-sets to characterize PLL performance: EEG eyes-closed resting state data-set and sleep recordings.

Firstly, the eyes-closed data-set is a public data-set made available by Trujillo et al. in their paper: this was used not for any clinical study but for extraction of information theory based brain connectivity measures [142]. 72 channel EEG data was collected using active Ag/AgCl electrodes and amplified by a BioSemi Active II amplifier system in 24-bit DC mode [142]. The data was collected initially at a sampling rate of 2,048 Hz but then was down-sampled online to 250 Hz. The data was collected from 22 participants (11 female and 11 male), with an age range of 18–26. For the experiment, they had to sit in a comfortable padded chair in a dark room and 4 minutes of eyes open and 4 minutes of eyes closed data were obtained [142]. Of these, we just used the eyes-closed portion of the data-set to aid fair comparison with the state-of-the-art, where they also used the same [44]. In total, this gave 88 minutes of eyes-closed data across all subjects.

Moreover, we use just the frontal channel for analysis: Channel Fp1 was used and this is an arbitrary choice since this is located in the forehead area where alpha power is known to be decent and moreover, this would be a suitable location when translating this technology for at-home use, since a better quality recording could be obtained due to lack of hair. Even though we used channel Fp1 for analysis involving this data-set, it must also be noted that we also investigated performance across all channels to investigate inter-channel variance (Section 4.4.2).

With this, we then used this data-set first to demonstrate the PLL tuning procedure and its

Data-set	Description	Channel	Fs (Hz)	Number of subjects
1	Eyes-closed resting state recording.	Fp1	250	22
2	Slow Oscillations from sleep EEG recording.	Fpz	500	13

Table 4.1. Data-sets used for PLL offline characterisation.

applicability in phase tracking across all EEG bands in Sections 4.4 and 4.3; for this, the experimental paradigm is irrelevant as we are characterizing performance across all bands, and this data-set will not only have strong oscillations in the lower EEG bands, but will also contain oscillations in the higher band, despite being weaker. Moreover, the state-of-the-art study we compare our results to (Mansouri et al.) for performance across all bands, also use eyes-closed resting state data-set, which would facilitate a fair comparison [44]. Secondly, this data-set is also used to characterize frequency matching accuracy for the chronic pain application driver. Here, since the data would contain strong oscillations in the alpha band, it closely mimics a visual entrainment experimental paradigm where strong alpha oscillations are also expected. This then makes it suitable for the chronic pain application driver, where frequency matched alpha stimuli is to be provided and strong alpha oscillations are expected, as detailed in Section 2.6.3.

For the sleep engineering application, we used a sleep recording data-set collected by our collaborators at Cardiff university (Research Ethics approval EC.17.12.12.5187). Clinical outcomes from this data-set is yet to be extracted, as results are not published yet. However, the experiment was designed for the sleep engineering application, which is the same application driver we target: auditory tones were provided during sleep to increase slow oscillations, and in result improve memory consolidation. For this, 2 night recordings (1 night for sham and 1 night for stimulation) from 13 subjects were collected; the participants were healthy young individuals (age range 18–30), with no history of smoking or sleep disturbances. For our analysis, we just used the sham night, to mimic the real-world implementation.

For the experiment, the participants slept for 8–9 hours on a bed at a sleep lab with electrodes wired on to their scalp: the room was temperature controlled and lights were turned off, except for a bed lamp. Before sleeping, they were asked to take a word-pair recall test, which was then again repeated the morning after, as a behavioural test for any improvement in memory consolidation over-night. This, however, is not relevant for our technology validation work, so further details are omitted.

The sleep data was collected using a BrainAmp DC amplifier (Brain Products, Germany), at a sampling rate of 500 Hz: the data was filtered in the slow oscillation (SO) band (0.5–3 Hz) and channel Fpz was used to extract the SO, which were obtained by our collaborators using the algorithm in [127]. A total of 37,440 SO epochs were extracted, each of 5 s length. We then used these epochs to characterize phase tracking performance, as it is on these that phase-locked auditory stimulation is to be provided, for the sleep engineering application. The data-sets used are summarized in Table 4.1.

Delta	Theta	Alpha	Low beta	High beta	Low Gamma	High Gamma
0.5-4 Hz	4-8 Hz	8-13 Hz	13-20 Hz	20-30 Hz	30-40 Hz	40-50 Hz

Table 4.2. EEG frequency band ranges.

4.2 EEG signal characterization

4.2.1 Introduction

In this Section, we characterize the EEG signal for its various properties: the dominant frequency range (Section 4.2.2), frequency rate of change (Section 4.2.3), and the amplitude (Section 4.2.4). All of these would then be fed into the PLL tuning methodology in Section 4.3. As seen in the following sections, these analyses are not available in existing literature to the best of our knowledge, especially for all the EEG bands. Hence, we then found it useful to do our own characterization, for use in the PLL tuning methodology in Section 4.3.

4.2.2 Frequency range

Introduction

Firstly, as we will see in Section 4.3, part of the PLL tuning methodology is to define a lock-range specification and the centre frequency for the PLL i.e. the range over which the PLL should gain lock quickly, as detailed in Section 4.3. For example, if the PLL centre frequency is 10 Hz and the lock-range is ±2Hz, then the PLL is expected to keep good lock over the 8–12 Hz range. To define both these specs, one option would be to use the standard EEG band range. The definition of these bands vary among different studies; for example, Newson et al. reviewed 184 EEG eyes-closed and eyes-open studies and found that the typical definition for these bands are: delta (1.3–3.5 Hz), theta (4–7.5 Hz), alpha (8–13 Hz), beta (12.5–30 Hz) and gamma (30–40 Hz) [143]. However, they also observed that there were variances in these and some groups would use ranges outside these. They reported minimum to maximum ranges (minimum start value (Hz)–maximum end value (Hz)) and showed they were: delta (0–6Hz), theta (2.5–8 Hz), alpha (6–14 Hz), beta (12–50 Hz) and gamma (30–40 Hz) [143]. As another example, zooming in on the alpha band, groups have defined a different range for the alpha band for their studies on how the Individual Alpha Frequency (IAF) varies over time: Haegens et al. (7–14 Hz), Weber et al. (8–12 Hz), and Gutmann et al. (7–13 Hz) [118]–[120]. Keeping these variations in definitions used in mind, we chose to use the ones defined by Mansouri et al., as it is the state-of-the-art for phase tracking across all bands, which we benchmark our results against, as seen in Section 4.4.3. Hence, for our study, the frequency bands are defined the same, and is outlined in Table 4.2.

From these band definitions, the centre values could be taken as the centre frequency and then the range as the lock range; for example, for the alpha band, the centre frequency could be 10 Hz and then the lock range ±2 Hz. Although, this may yield good results as the PLL is adaptive and the gain could be modified to achieve best results, as we see in Section 4.3, the whole point of our PLL tuning methodology was to get a good starting point for the PLL parameters, before optimizing to get best results. Hence, a better approach would be to use the dominant frequency (peak frequency) within each band as the centre frequency, and a measure of its variance for the lock range spec. We do however note that due to our small sample size, our estimate of the dominant frequency here would be biased and for a more general use-case, the standard band-definition maybe more helpful. Regardless, we chose the former route here, to get a good starting point for the tuning parameters, before optimisation. Moreover, even within our small sample size (n=22 for pain and n=13 for sleep), we divided the data-set in a 70–30 tuning-test split, to avoid any in-sample optimisation.

Such a measure for the dominant frequency range is not available in literature for all EEG bands, to the best of our knowledge and of the ones available, they usually focus on changes over a longer time-scale: for example, Weber et al. investigated IAF variations after 30 minutes of exercise and Gutmann et al. after 120 days of isolation [119], [120]. A close one would be a study by Haegens et al., where they calculated the IAF (alpha peak frequency) on 1s epochs, and showed that it varied by 10.3±0.9 Hz within subjects. However, they focused on just the alpha band and with a window size of 1 second, and not sample-by-sample (the theoretical limit), which is the mode of operation for our closed-loop system in Chapter 5.

With such an analysis lacking, we then decided to characterize the EEG instantaneous frequency on a sample-by-sample time frame, for all EEG bands. A sample-by-sample characterization of instantaneous frequency and its range is useful, as this would be the mode of operation for a real-time closed-loop system implemented in Chapter 5. We also have validated in Section 4.6 that the sample-by-sample characterization of the EEG frequency is a good approximation to the window/epoch based one. This metric would then be used for the PLL tuning methodology in Section 4.3, to define the lock range and the centre frequency of the PLL.

Methods

We used the data-set 1 from Table 4.1 to do the frequency characterization; this data-set is chosen, as this is the one used for characterizing phase accuracy in all bands, as detailed in Section 4.1.2. With this, we first appended data from all subjects (n=22) into a single signal which was then band-pass filtered (Order 2 IIR butterworth filter) for each of the EEG bands, with the pass-band frequencies defined in Table 4.2. Then, it was passed through a Hilbert transform to obtain the instantaneous phase values, after which the derivative of the instantaneous phase was taken to obtain the instantaneous frequency values, for each sample of the entire recording. This then gave the dominant instantaneous frequency (mean) and its variation (3×standard deviation). Note that we used 3 times the standard deviation, as we wanted to most of the frequency ranges (99.7%) expected, when tuning the PLL.

	Delta	**Theta**	**Alpha**	**Low beta**	**High beta**	**Low gamma**	**High gamma**
Frequency range	2.18 ±1.44	6.67 ±2.78	10.24 ±2.00	14.90 ±4.03	23.34 ±3.82	33.66 ±3.59	43.99 ±3.13

Table 4.3. Sample-by-sample instantaneous frequency range characterization for all EEG bands.

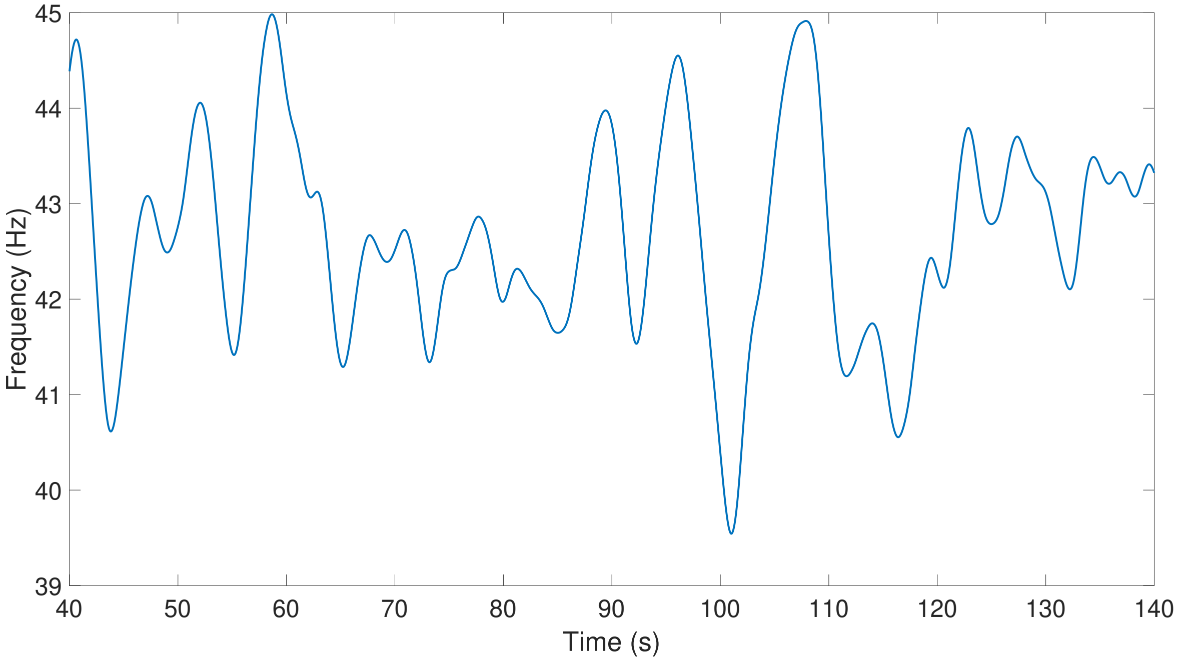

Figure 4.1. Sample trace of instantaneous frequency for the high gamma band.

With these, the mean of the instantaneous frequency and the 3×standard deviation, would be used as the centre frequency and lock-range, respectively, in Section 4.3.

Results and Discussion

For the instantaneous frequency range analysis, the results are depicted in Fig. 4.2 and summarized in Table 4.3. Moreover, a sample trace for the instantaneous frequency of an EEG band (high gamma), is shown in Fig. 4.1.

As observed from Fig. 4.2 and Table 4.3, the mean instantaneous frequency and the 3×standard deviation, falls within the EEG bands defined in Table 4.2. Our aim was to find the instantaneous frequency variations within these band definition, and hence this is what we expect, since we bandpass filtered within these bands defined in Table 4.2. However, it must be noted here that the range is smaller for some bands: for example, according the definitions by Mansouri et al. in Table 4.2, for the high gamma band (40–50 Hz), the instantaneous frequency range within this band is 43.99±3.13, which is smaller than the band definition. This knowledge is then useful in getting specs for the lock ranges required for PLL, as part of the tuning methodology: for the PLL tuning, we can then use the mean as the centre frequency, and the 3×standard deviation as the lock-range.

Moreover, comparing this to the study by Haegens et al., where they showed 10.3±0.9 Hz within subject IAF variation, we see our alpha range (10.24±2.00 Hz) also falls within a similar range, but with a higher variation [118]. The higher variation could be due to a va-

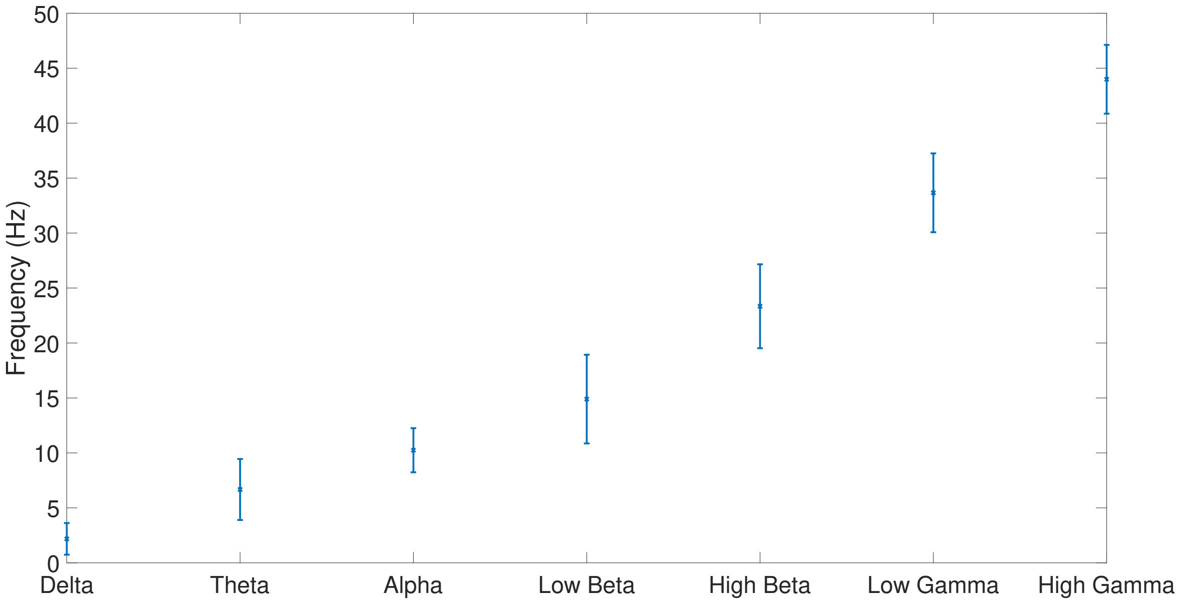

Figure 4.2. Dominant frequency range for all EEG bands, using the instantaneous frequency measure. The error bars represent 3σ.

riety of factors such as: difference in window size (1 sample for ours vs 1s for Haegens et al.) and difference in experimental tasks (eyes-closed for ours vs visual stimuli and working memory task for Haegens et al.). Regardless, we see the values broadly match.

Conclusion

In conclusion, in this Section, we characterized EEG dominant frequency range on a sample-by-sample basis, for all EEG bands, with results summarized in Table 4.3. From these, we could define the centre frequency and the lock-range for the PLL tuning in Section 4.3.

4.2.3 Frequency rate of change

Introduction

For the chronic pain application driver, as detailed in Section 2.6.3, the technology contribution is to match the frequency on a sample-by-sample basis, i.e. as fast as is technologically possible. This can then be easily scaled to any closed loop bandwidth desired. In this section, we aim to find out whether it makes sense to update on a sample-by-sample basis via characterization of the instantaneous frequency rate of change i.e. to find out if the frequency does change at such a small time-scale. This would help judge whether it would be of any benefit to change stimulation at such small time-scales; for example, if the frequency does not change over a sample-by-sample basis, then there would be less benefit in closing the loop here at such small time-scales and perhaps a longer time-scale may be more meaningful, but if it does, then it would be a path worth exploring. Moreover, since this analysis is primarily to motivate the chronic pain application driver, we only analyse the rate-of-change for the alpha band, which is the bio-marker of interest [14].

To date, no characterization of the alpha instantaneous frequency rate of change exists in literature, to the best of our knowledge. This may be because traditionally, the IAF has been considered to be a static trait amongst individuals that does not vary over time, hence the reason why most studies use a fixed IAF determined beforehand, on a per-subject basis [16], [17], [49]. For example, as detailed in Section 2.6.3, for the studies that showed that alpha entrainment is more likely when stimulation frequency is close to the IAF, they all determined the IAF during a base-line EEG recording period, and used that as a fixed pre-set for the remainder of the experiment [16], [17], [49]. However, recent evidence shows that the IAF does vary over time within a subject over time-scales as long as 120 days and 30 minutes, as well as on a second by second basis, as detailed in Section 4.2.2 [118]–[120]. However, these studies do not involve IAF monitoring on a small a time-scale as we need and with no characterization for the rate-of-change, as the nature of their work was an open-loop study, investigating IAF before and after, a certain task [118]–[120].

With such an analysis lacking, we then decided to characterize the IAF rate of change on a sample-by-sample basis (theoretical limit), which will help motivate closing the loop at such small time-scales.

Methods

For the data-set, we used the eyes-closed resting state data-set 1 from Table 4.1 for this analysis, which has a total of 22 subjects, each with 4 minutes of eyes-closed resting state EEG data-set.

To calculate the rate of change, the signal processing pipeline starts with an order 2 IIR butterworth band-pass filter with a passband of 8–13 Hz for alpha band filtering. The Hilbert transform is then taken over the filtered signal for each subject to obtain the instantaneous phase, after which its derivative is taken to obtain the instantaneous frequency. The derivative operation is noise sensitive and amplifies the high frequency components. Hence, we also used an order 2 IIR butterworth low pass filter with a 2.5 Hz cut-off frequency, to remove the high frequency spikes (see example in Fig. 4.3) and smooth out the frequency output. The filter cut-off was chosen through experimentation and was empirically determined to ensure that most of the derivative induced spikes are removed whilst retaining the low frequency trend. Also to clarify, by spikes, we mean those values that far exceed that of the band range (e.g. 8–13 Hz for alpha band), examples of which could be seen in Fig. 4.3. After smoothing the output, the derivative of the filtered instantaneous frequency is taken to get the rate of change. Finally, the Root Mean Square (RMS) of the rate was used as an ‘average’ rate-of-change metric, as it is more or less an AC like waveform with mean at 0. Having calculated the rate of change for each subject, we then report the mean and the standard deviation across the 22 subjects, to characterize the average rate of change and its inter-subject spread.

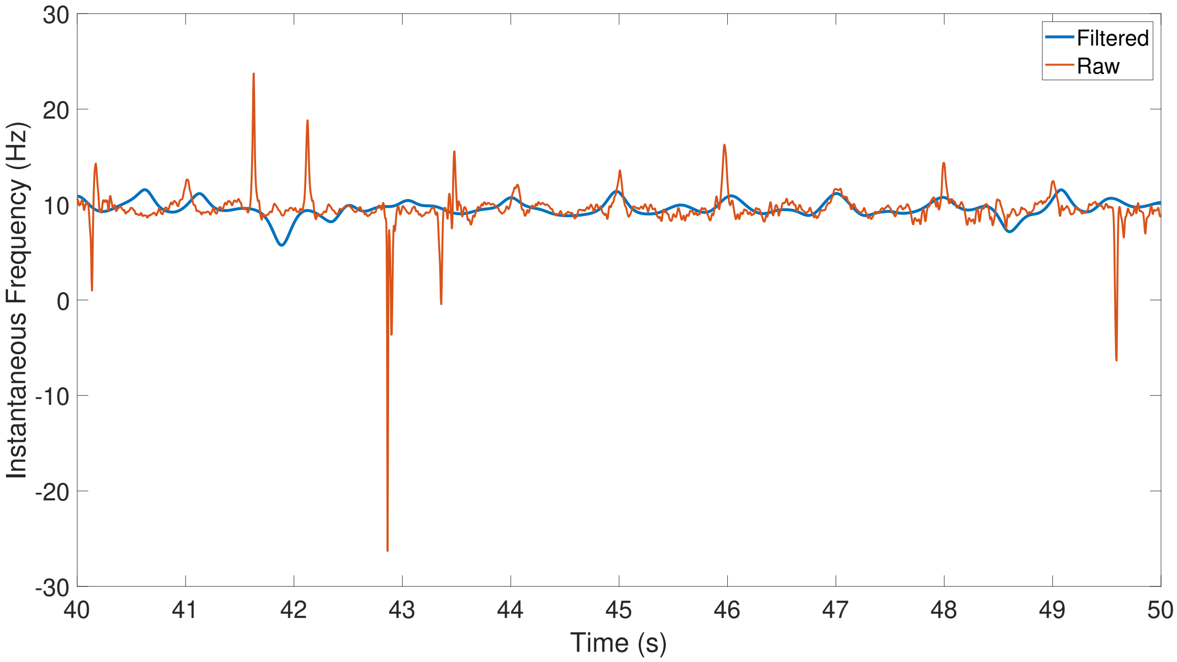

Figure 4.3. A trace of the instantaneous frequency before and after low-pass filtering to remove the derivative induced spikes.

Results and Discussion

As seen from Fig. 4.3, the raw instantaneous frequency plot coming out of the derivative operation is spiky; the spikes are derivative induced, as discussed earlier. Consequently, the filtered waveform successfully removed the spikes, whilst retaining the general trend. However, we do notice that at a few portions e.g. $t = 41s$ to $42s$, there is an unusual down spike deviating from the frequency trend. This would be a filtering induced artefact due to the two neighboring up spikes. Such occurrences are rare and this could have been improved using more sophisticated filters such as a median filter, which are more suitable for removing spikes. However, this would result in a trade-off in computational cost, as median filters are computationally more costly due to the sorting operation required to find the median in a window of data. Hence, we avoided these, as our goal is a smart-phone based closed-loop system, where computational burden must be kept low.

Moreover, we also see from Fig. 4.4 that the frequency varies over time, even for short time spans. More specifically, the RMS value for the rate-of-change was found to be 9.75±2.67 Hz/s, as seen from Fig. 4.4. Note that at first glance, this may seem counter-intuitive i.e. the EEG does not change say by 10 Hz increments every second. Instead, this has to do with the fact that we used the RMS value of the rate-of-change waveform as a measure of the average, which captures mostly the upper-range values of the rate-of-change. It does not capture the fact that the rate varies over-time and a lot of the times is close to zero too, as seen from Fig. 4.4. In other words, bursts of rapid change are present, but not for very long which means that absolute change in frequency is in-line with the results in Section 4.2.2, where we showed the alpha frequency range is 10.24±2.00 Hz. Hence, due to the imperfection of the metric, the 9.75±2.67 Hz/s must be seen as the upper-range value for the rate.

Moreover, it must be highlighted here that no such analysis exists in literature, for us to

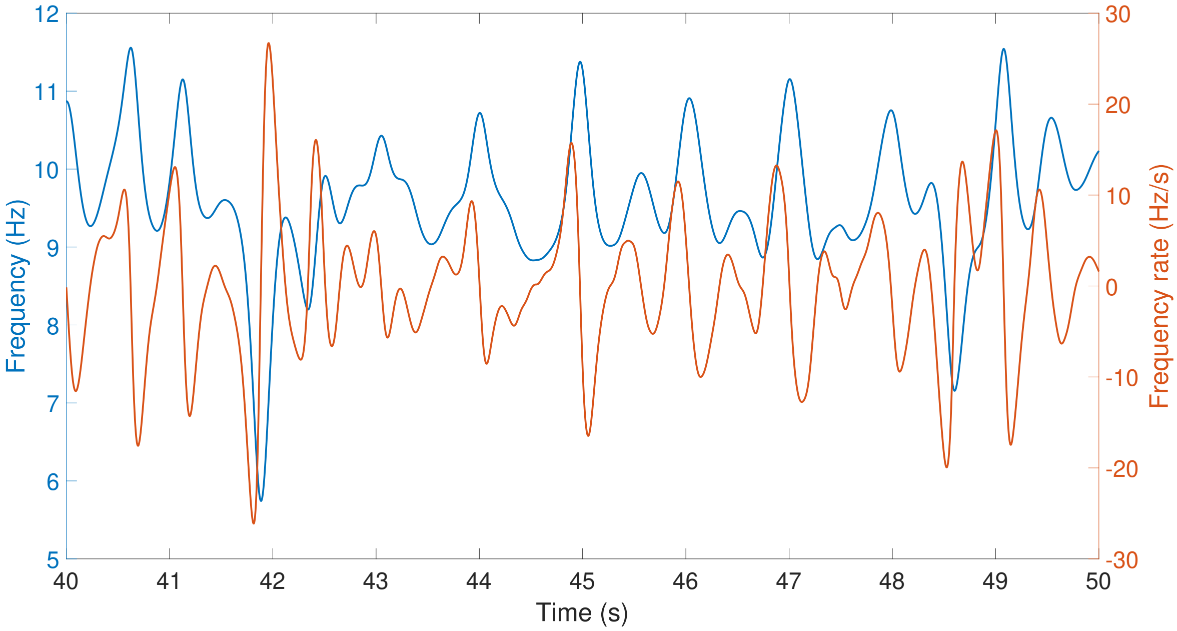

Figure 4.4. Sample trace of the instantaneous frequency and its rate of change.

benchmark our results against. The closest work are those that measure change of IAF over time and that too on a longer time-scale, without a rate-of-change characterization: e.g. the IAF calculation over 1s windows in the study by Haegens et al [118]. The reason for a lack of such an analysis could be that such studies operate either on open-loop paradigmns (investigating changes in IAF before and after certain tasks) or they used a 'semi-closed loop' paradigm where the IAF is calculated over a base-line period before experiment starts on a per-subject basis, and then remains fixed [16], [17], [49], [118]–[120]. For such studies, a rate-of-change characterization is not necessary, since it is not closed loop.

With this in mind, we can then conclude that closing the loop for such short time scales (sample-by-sample) does make sense, as the frequency does vary, as observed both from Fig. 4.4 and the 9.75±2.67 Hz/s RMS value for the rate-of-change waveform. For example, for a sampling frequency of 250 Hz (sampling frequency of our wireless EEG acquisition system in Chapter 5), that's 4 ms between samples received. Then, with a maximum rate of 10 Hz/s, the maximum change we can then capture is 40 mHz, in the closed-loop system. Moreover, as observed from Table 4.3, the values for the frequency range are reported to 2 decimal places, ie. a change of 10 mHz. So, we can see (at least on average) changes in the brain that are this small and a sample-by-sample approach would be needed to potentially follow these small changes in frequency.

Hence, for the chronic pain application driver, we could then develop technology that provides frequency matched stimulation on a sample-by-sample basis, which is the theoretical limit.

Conclusion

In conclusion, in this Section, we characterized the EEG instantaneous frequency rate-of-change and found it to have an RMS value of 9.75±2.67 Hz/s, which we then showed is high enough for us to justify closing the loop on a sample-by-sample basis, for our closed-loop system in Chapter 5.

4.2.4 Amplitude

Introduction

In this Section, we characterize EEG amplitude. More specifically, we need to find a virtual maximum or upper-range value for the EEG signal. This is needed since the EEG amplitude is variable and the PLL performance is amplitude dependant, as detailed in Section 4.4.4. This amplitude dependent performance was also reported by Santostasi et al. in their lab-based PLL platform, and they did not take this into account when tuning the PLL but rather relied on a trial and error approach, to achieve best possible accuracy [2]. Moreover, none of the PLL based platforms do any amplitude normalization procedures such as the use of an automatic gain control, as it may have added additional computational complexity and its own tuning related issues, without adding much benefit i.e. the performance was already good enough without any amplitude normalization [2], [4], [5], [13]. Hence, we follow a similar approach and do not add any amplitude normalization components to our PLL. Regardless, we aimed to tune the PLL taking into account the incoming amplitude of the EEG, as detailed in Section 4.3. Hence, here we aim to approximate the expected EEG amplitude with a constant upper-range value, to take into account the time-varying nature of the EEG amplitude. This procedure is detailed in Section 4.3.

Such an analysis of the amplitude ranges of the EEG within each frequency band, is not available in literature, to the best of our knowledge. The most common approach in EEG literature is to work in the frequency domain and use the band-power as bio-markers for different diseases; Newson et al. reviewed 184 EEG studies where band-power was used as bio-markers for the different psychiatric diseases [143]. For example, they showed that studies involving Attention Deficit Hyperactivity Disorder (ADHD), Obsessive Compulsive Disorder (OCD) and Schizophrenia, typically involved higher power in the lower bands (delta and theta) for the patients, when compared to healthy humans. Such studies focus on just one band and not all bands. Moreover, what we are interested in is the amplitude ranges for all EEG bands in the time-domain after band-pass filtering; this is the input signal expected into the PLL, the amplitude of which we are trying to take into account when tuning the PLL, as detailed in Section 4.3.

Methods

To characterize the amplitude range, we used data-set 1 from Table 4.1; this data-set is chosen as this is the one used for characterizing phase accuracy in all bands, as detailed in Section 4.1.2 and that is what we are tuning the PLL here for. The aim then is to find an amplitude value A, a upper range value for each band which we define as the 75th percentile value. This then approximates the expected input (EEG) amplitude, to the PLL.

	Delta	Theta	Alpha	Low beta	High beta	Low gamma	High gamma
Frequency range (Hz)	0.5–4	4–8	8–13	13–20	20–30	30–40	40–50
Order	2						
Type	Butterworth IIR						

Table 4.4. Band-pass filter settings for all bands.

To achieve this, we first appended data from all subjects and then band-pass filtered for each band, with the filter settings detailed in Table 4.4. We then use this band-pass filtered signal and its box and whisker statistics to obtain the 75th percentile value, which we use as the value for A, as it would be a good measure of the upper-range of the incoming EEG signal. Using this value of A, we can then ensure that the gain value K for the PLL (see equation 4.11 from Section 4.3) is not too high to cause instability. This is because with the A value, we could ensure the gain is scaled down by the appropriate factor and take into account an approximate for the upper-range value of the incoming EEG signal. Moreover, we choose the 75th percentile and not the lower quartiles, to design for worst case scenarios. For example, if we used the lower range, then the PLL would have been tuned so that gain K would be higher than necessary to meet the desired lock range specifications, and as a result, it could break down at points where the EEG amplitude is higher than designed for.

Results and Discussion

The results for the amplitude characterization are summarized with the box and whisker plot in Fig. 4.5 and the virtual 'upper-range' amplitude A used for PLL tuning purposes are summarized in Table 4.5.

	Delta	Theta	Alpha	Low beta	High beta	Low gamma	High gamma
A= 75th percentile	4.72	3.16	4.60	2.20	1.60	0.99	0.74

Table 4.5. Amplitude characterisation for all EEG bands.

As observed from Fig. 4.5, while there are outliers in all of the bands, the core is quite similar, with the amplitudes being relatively small. Despite this, it is clear that in general, amplitude decreases for higher EEG bands: for example, the amplitude A from Table 4.5 is highest for lower bands (e.g. delta and alpha), and lower for higher bands. We expect these results; for example, when a user's eyes are closed and in a resting state, lower bands are expected to have higher power, especially the alpha band [144]. This explains the high alpha and delta amplitudes, as this was from a eyes-closed resting state experiment.

These amplitude values will then be used later in the PLL tuning procedure to take into account the amplitude dependent lock range of PLLs; we do this by equating the EEG signal as a virtual signal with a constant amplitude A (values from Table 4.5).

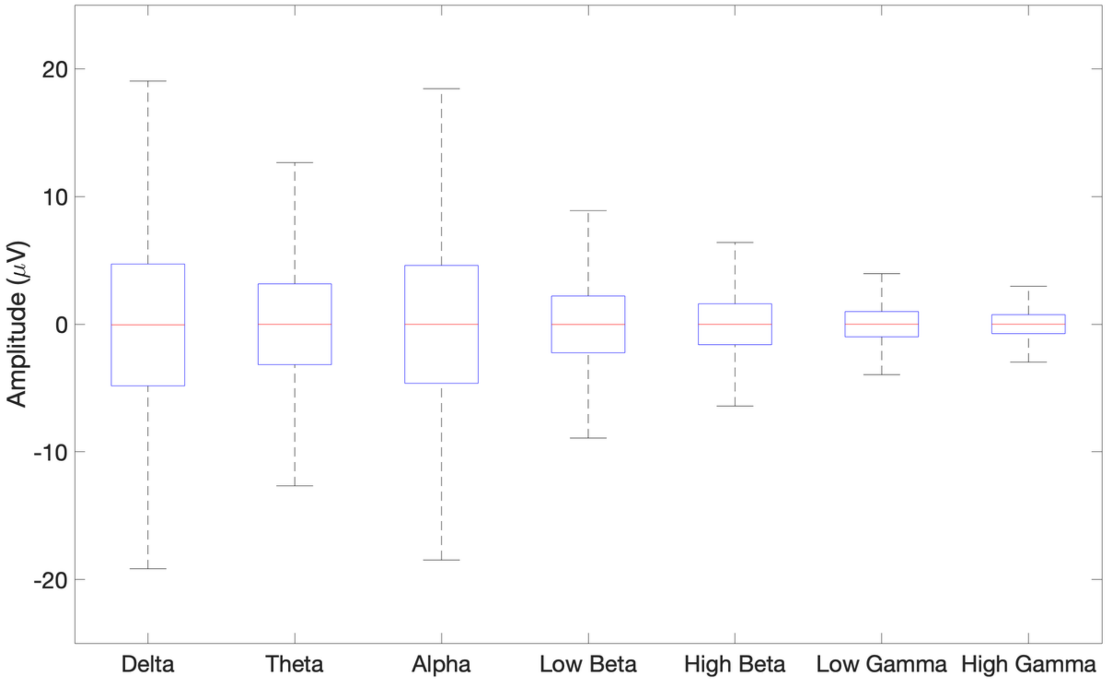

Figure 4.5. Amplitude distribution box plot for all EEG bands. Ends of the whisker plots correspond to: 1.9σ for Delta, Alpha and High Gamma bands; 2.1σ for Theta and Low Gamma bands; and 2.2σ for Low Beta and High Beta bands.

Conclusion

In conclusion, we characterized EEG amplitudes for all EEG bands by using the 75th percentile values of the box and whisker statistics as a proxy for the amplitude A, which will be used to approximate an incoming EEG signal, for the PLL tuning procedure in Section 4.3. The trends we observed were similar to that found in literature: high power in the lower bands (alpha/theta) for an eyes-closed resting state data-set.

4.2.5 Summary

In this Section, we characterized the EEG signal for its frequency range, frequency rate of change and amplitude. For the frequency range, we will use these values for defining the centre frequency (mean) and lock range (3×standard deviation), when tuning the PLL in Section 4.3. For the amplitude, we noticed a general decrease in amplitude for higher EEG bands, alongside high alpha/theta amplitudes and this again would feed into the PLL tuning methodology, to take into account the time-varying nature of EEG amplitude. Finally, we also confirmed that the EEG frequency changes over very short time spans (sample-by-sample), and we reported an RMS of 9.75±2.67 Hz/s for the rate of change; this then confirms the need for closing the loop at such short time-scales on a sample-by-sample basis, in order to follow these changes in frequency, which happen at such short time-scales.

4.3 Phase Locked Loop: tuning procedure

4.3.1 Introduction

A Phase Locked Loop (PLL) was used for real-time feature extraction of both phase and frequency, which we discussed in the literature review in Section 2.5.5. PLLs are attractive due to its adaptive nature, low computational complexity and its ability to operate on a sample-by-sample basis. This last factor could drive down loop latency and provide immunity against edge artefacts that is common with epoch-based methods like the Hilbert transform and FFT based methods [121]. PLLs have previously been used on PC based platforms for the sleep engineering application [2], [4], [5] but not for frequency matched stimulation in pain. For sleep, we provide the first smartphone based implementation to enable out-of-the-lab research, as detailed in Section 2.6.2.

Background

The field of Phase Locked Loops is a well matured field and has been a topic of research for more than 80 years. The first PLL was introduced at around 1932 by French engineer de Bellescize, also known as the inventor of coherent communication [121]. It later found various applications in industry and has since been used for mostly high frequency applications such as clock synchronization, clock recovery, frequency synthesis and phase and frequency demodulation [145]. It is also a building block of many modern day technologies such as television, motor control, disk-drive control and wireless communication systems [145].

The different classes of PLL can be broadly categorized into 4: Linear PLLs, Digital PLLs, All-Digital PLLs (ADPLL) and Software PLL (SPLL). Linear PLLs were the first kinds available as IC chips; introduced at around 1965, these were made up of purely analogue parts with a 4-quadrant multiplier as the phase detector, active/passive RC filters as the low pass filters, and a voltage controlled oscillator [121]. Later, in the 1970s, DPLLs came into the market, which unlike its name is not made up of all digital components; rather, a digital XOR phase detector is used, while the remaining components remained analogue [121]. It is only later that pure digital PLLs were made available, also known as All Digital Phase Locked Loops (ADPLL) [121]. Finally, with the advent of fast micro-processors and Digital Signal Processors (DSPs), PLLs crossed over into the software domain, where all electronic components were replaced by lines of code [121]. The obvious benefits of these are the easy re-reconfigurability and on-the-fly modifications that could be made, which is not possible with its hardware counter parts. Moreover, these were the most generalizable type of PLL, as all the other 3 kinds of PLLs (ADPLL, DPLL, LPLL) are realizable in software. Despite these, the one factor that limits the use of SPLLs in many applications is that the algorithm must run at a frequency at least twice that of the reference signal; this is to satisfy the Nyquist theorem and only becomes an issue for high frequency applications where the

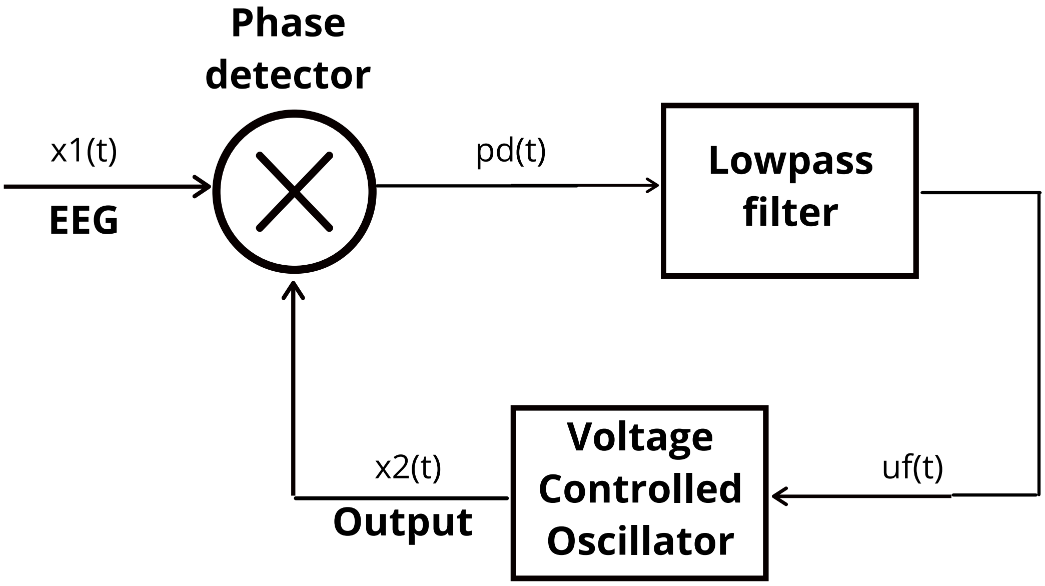

Figure 4.6. A block diagram of the PLL. The phase detector coupled with the low pass filter outputs the phase difference between the two signals, which is then used as the control signal for the voltage controller oscillator (VCO). The VCO then increases or decreases its running oscillator frequency till phase lock is achieved.

signal frequencies exceed the micro-processor's sampling frequency capabilities [121]. For EEG signals, which have a frequency range of 1-100 Hz, this would not be an issue since modern day micro-processors sample signals at much higher frequencies.

For this project, the focus is the SPLL, as the ultimate goal is to provide Smart-Phone based closed-loop therapies; hence, an SPLL would provide the benefit of quick reconfigurability and more options of trying different settings and parameters out, with the entire loop running on the phone. Main benefit here is that no extra hardware will be added to the phone to run the PLL.

A basic PLL and how it functions

The PLL mathematics are well documented in literature and here, we provide a digital implementation of an analogue PLL with a first order low pass filter (active lead lag), similar to [2]. A block diagram of the PLL is shown in Figure 4.6.

We use a classical PLL with a simple multiplier as the phase detector, as this has shown to work well in previous studies [2], [4], [5]. We could have used a digital XOR phase-detector based PLL to account for amplitude variations, but we did not go this route, as the conversion of the EEG to a constant amplitude signal would add additional complexities, due to signal info lost during the conversion. Moreover, since previous work has proved the classical PLL to be effective, we deemed this the most suitable choice for the constraints of the situation.

Let x$_1$(t) be the input and x$_2$(t) the output from the PLL i.e. the voltage controlled oscillator (VCO) output. The ultimate goal then is to reduce the phase error between x$_1$(t) and x$_2$(t), which leads to a locked state. Let,

$$x_1(t) = A_1 \sin[w_1 t + \theta_1(t)] \tag{4.1}$$

$$x_2(t) = A_2 \sin[w_2 t + \theta_2(t)] \tag{4.2}$$

Firstly, the phase detector which in our case is an analogue multiplier takes the product of signals $x_1(t)$ and $x_2(t)$, which gives us the phase detector output $p_d(t)$,

$$p_d(t) = K_d A_1 A_2 \sin[w_1 t + \theta_1(t)] \sin[w_2 t + \theta_2(t)] \tag{4.3}$$

Where, K_d is the phase detector gain. Next, assuming the PLL is in lock, the two frequencies w_1 and w_2 are identical. Then, using trigonometric identities, (4.3) is rearranged as,

$$p_d(t) = \frac{K_d A_1 A_2}{2} \sin[\theta_2(t) - \theta_1(t)] + \frac{K_d A_1 A_2}{2} \sin[2wt + \theta_2(t) + \theta_1(t)]. \tag{4.4}$$

We notice here that the output of the phase detector contains both a DC term containing information about the phase difference between the 2 signals and a high frequency term. This high frequency term needs to be removed and this is done by the low pass filter in the PLL. The filtered output then is a representation of the phase error between the input and output. It is a sinusoid of the error and hence, the phase detector here is non-linear. To linearize this, we assume that the signal is tracking and that the phase difference between the two is small, which gives us the linearized phase detector equation [121],

$$p_d(t) = \frac{K_d A_1 A_2}{2} [\theta_2(t) - \theta_1(t)] + \frac{K_d A_1 A_2}{2} sin[2wt + \theta_2(t) + \theta_1(t)]. \tag{4.5}$$

From here, the output is filtered using a low pass filter, and assuming perfect filtering, we get the filter output as,

$$u_f(t) = \frac{K_d K_a A_1 A_2}{2} [\theta_2(t) - \theta_1(t)] \tag{4.6}$$

Where K_a is the filter gain. This filtered output is then used as the control input to the Voltage Controlled Oscillator, which increases or decreases its running oscillator frequency, based on the input presented to it. Hence, first when the phase error is large, the larger control voltage will cause the VCO to compensate by increasing its oscillator frequency, which will cause the error to decrease in the next iteration and this process is repeated until both the input and output signals are locked.

There are various parameters of the PLL that needs tuning such as the low pass filter coefficients, loop gain and the VCO centre frequency. To tune these, other PLL-based platforms reported in Table 2.1 have done this through trial and error [2], [4], [5], as opposed to using a more systematic approach which is what we aim to do. For this, one option would be to use a top-down approach using optimization algorithms; for example, finding a global minimum for a given cost function which minimizes phase error. The other option is to use a bottom-up approach; using control systems theory. Since, the PLL is a closed-loop feedback system, we could characterize it via its closed-loop transfer function and then tune and optimize its parameters for best performance.

In our study, we use the bottom-up approach to get a good first estimate, and then hand-tuned the values to optimize performance. This way, we get more control over the tuning procedure.

4.3.2 Methods

The PLL tuning process starts with defining an approximate specification for the lock range Δf_l. The lock-range is one of many metrics used to define the operating range of a PLL; it defines the frequency range over which the PLL would quickly lock on to an incoming signal [121]. Other ranges include the pull range and hold range, which defines the slow lock-in range and the absolute range over which the PLL would lose lock, respectively [121]. For our work, we choose the lock-range as the metric of choice, as we want to design for quick lock-in whenever the PLL loses lock. The lock-range Δf_l is then defined as,

$$\Delta f_l \approx \frac{\zeta \omega_n}{\pi}[121] \tag{4.7}$$

Where, ζ is the damping factor and ω_n is the natural frequency, of the closed-loop system.

With this at hand, we can then design the closed-loop system and tune it to achieve the ω_n and ζ values, for the desired approximate lock-range. To do this, we first need the transfer function of the PLL, which could be obtained from the transfer function of the individual components of the PLL. For this, the transfer functions of the lead-lag controller (in this case a low pass filter) $F(s)$ and that of the VCO $G(s)$ are,

$$F(s) = K_a \frac{s\tau_2 + 1}{s\tau_1 + 1}, \quad G(s) = \frac{K_o}{s} \tag{4.8}$$

Where, K_0 and K_a are the VCO and filter gains, respectively and τ_1, τ_2 the filter coefficients. From this, the phase transfer function for the entire PLL $H(s)$ is derived as,

$$H(s) = \frac{\theta_2}{\theta_1} = \frac{KF(s)G(s)}{s + KF(s)G(s)} \tag{4.9}$$

And in terms of the loop gain $K1$ and filter coefficients, it is:

$$H(s) = \frac{\Theta_2}{\Theta_1} = \frac{K\frac{1+s\tau_2}{\tau_1}}{s^2 + s\frac{1+K\tau_2}{\tau_1} + \frac{K}{\tau_1}}, K = K_0 K_d K_a \tag{4.10}$$

where Θ_2 and $\Theta 1$ are the output and input phase, respectively; K_0, K_d and K_a are the VCO, phase detector and low pass filter gains, respectively; and τ_1 and τ_2 are the filter coefficients.

It must be noted here that for the tuning process, we replace the phase detector gain as $K_d = A$, where A is the amplitude of incoming EEG and it is used only for offline tuning (transfer function gain) because the gain of the phase detector could be conceptualized as having A; this is because the output of the phase detector will be amplified by EEG's varying amplitude since it is a multiplier detector. Hence, we need to take this into account when tuning the PLL to account for amplitude variations. However, after tuning, when ported on to the phone, there won't be any A in the code for the PLL, since for the real system, the EEG will already be incoming and we just used A as an approximate amplitude for this incoming EEG, for tuning purposes only. With this, the gain for tuning purposes then becomes,

$$K = AK_0 K_a \tag{4.11}$$

where, A is the assumed maximum EEG amplitude discarding the outliers, as estimated in Section 4.2.4.

Comparing the denominator of (4.10) to that of the normalized form of a second order transfer function, we can then obtain the ω_n and ζ values in terms of the PLL components i.e. the filter coefficients and loop gain. They are then given as:

$$\omega_n = \frac{K}{\tau_1}, \zeta = \frac{\omega_n}{2}(\tau_2 + \frac{1}{K}) \tag{4.12}$$

From (4.12), we observe that tuning a PLL to get a desired lock range is not straight-forward. Both ω_n and ζ which define the lock-range are inter-dependent, with the loop gain K and filter coefficients effecting both at the same time. For example, to increase lock-range, both ω_n and ζ could be increased but we want to limit ζ close to 0.707 i.e. critically damped: this is done to avoid an over-damped ($\zeta < 1$) or an under-damped system ($\zeta > 1$), which are characterized by heavy oscillations and a sluggish response, respectively; both these would not be ideal in a PLL. Hence, while tuning we need to ensure ζ is close to 0.707 and ω_n high enough to meet the desired lock range spec.

Now, to increase ω_n, first option is to increase gain K. However, as seen from (4.12), increasing gain not only increases ω_n but also both increases ζ due to increased ω_n but also reduces ζ at the same time, due to the 1/K term.

The other parameters that could be tuned are the filter coefficients. Here, it must be noted that lowering τ_1 i.e. increasing poles of the filter $F(s)$ in (4.8) (also increases cut-off frequency and band-width of filter) would increase ω_n and ζ, which would result in an increased lock range, since the lock range is proportional to ω_n and ζ (see (4.7)). Hence, care must be taken not to increase τ_1 too much, such that ζ gets too high, which would result in an under-damped system. It maybe tempting to think then that one could combat the increased ζ by reducing τ_2, which would in theory reduce ζ, while also increasing high frequency attenuation, as τ_2 controls the zero. However, this should be done with caution as there is a limit to how much τ_2 could be reduced before the ratio between τ_1 and τ_2 would exceed or violate the low pass filter conditions ($\tau_1 > \tau_2$), and would in turn result in a high pass filter because of τ_1, τ_2 ratios. Hence, a limit exists to how much τ_2 could be reduced.

For these reasons, we note then that tuning a PLL is not straightforward due to the interdependence between parameters: changing gain K effects both ζ and ω_n at the same time, and tuning the filter coefficients has its limits due to the pole zero ratio constraints to satisfy conditions for a low pass filter. Hence, both ζ and ω_n cannot be tuned independently.

With these complexities in mind, we then approached the tuning problem with the following procedure: first we define the operating range specification which would be the lock range f_l; second, we derive the required ω_n and ζ values, based on the lock range; third, we design the loop gain K and the filter coefficients τ_1 and τ_2, based on the required ζ and ω_n values; finally, with this as the starting point, we then fine tune and optimize the filter coefficients and the loop gain, to achieve maximum accuracy. The data-set used to demonstrate this is data-set 1 from Table 4.1; this data-set is chosen, as this is the one used for characterizing phase accuracy in all bands, as detailed in Section 4.1.2 and that is what we are tuning the PLL here for. It must also be noted that we used a 70–30 training-test split to avoid in-sample optimization; hence, the first 15 subjects data was used to tune the PLL here and the remaining 7 for testing the accuracy in Section 4.4. . The steps are summarized below:

1. **Defining the operating range** f_l: Here, we use the lock range f_l defined as the ±standard deviation of the EEG dominant frequency for each band, which was obtained in Section 4.2.2 and summarized in Table 4.3.

2. **Derive** ω_n **and** ζ: Here, firstly we set $\zeta = 0.707$ to aim for a critically damped system. With ζ and f_l defined, we can then find ω_n as (rearranging 4.7),

$$\omega_n \approx \frac{\Delta f_l \pi}{\zeta} \tag{4.13}$$

3. **Design loop gain K and filter**: Here, first we design the loop filter coefficients by arbitrarily choosing the cut-off frequency to be 0.46 Hz, with $\tau_1 = 0.4$s and $\tau_2 = 0.1408$s, through trial and error. We chose this frequency, so that the cut-off is low enough to reduce jitter by attenuating the high frequency component of the phase detector output, whilst not being too low as to limit the lock-range. For example, designing for the worst case, assuming the both inputs to the phase detector is a 0.74 Hz signal (lower delta band range from Table 4.2), then the phase detector output would have a high frequency component twice that i.e. 1.48 Hz which needs to be filtered out, which a low pass filter with 0.46 Hz cut-off

would be able to do.

With this, we then manually varied gain K while keeping the filter coefficients constant till the resulting ω_n and ζ values were close enough to those defined in Step 2. This will then give our initial starting point for our PLL parameters.

4. **Fine tune and optimize**: Following this, we then varied gain K from the initial starting point in Step 3 till the best accuracy was achieved. For evaluating the phase tracking accuracy, we used the Phase Locking Value (PLV) as the metric of choice, to measure how well the PLL tracks the incoming EEG. Moreover, this would also facilitate comparison to state-of-the-art work by [44], where they used the same metric for characterizing accuracy over all bands, with their FFT-based algorithm. The PLV is defined as:

$$PLV = \left\| \frac{1}{N} \sum_{i=1}^{N} e^{i(\phi_{EEG} - \phi_{PLL})} \right\| \tag{4.14}$$

where, ϕ_{EEG} and ϕ_{PLL} are the EEG and PLL instantaneous phase at sample i and N is the number of samples over which we are calculating the PLV over. In our case, N is the entire data for a single subject, which gives us a single PLV value for each subject. We then use the mean and standard deviation over all subjects, to characterize PLL accuracy.

With this, we kept optimizing performance by varying only the gain K and not the filter coefficients, because it was more straight forward to do the former, whilst the latter had more constraints to it. For example, the τ_1, τ_2 ratios needed to be maintained for low pass filter conditions to be satisfied, in addition to the inter-dependence detailed earlier. For these reasons, it was more straightforward to optimize by varying gain K.

Finally, it must also be noted here that when coding this PLL, we converted the analogue low pass filter designed here to a digital low pass filter using the bilinear transform, as the PLL is run in the digital domain.

4.3.3 Results and Discussion

Here, we present the results for the final PLL tuning parameters, for all bands both pre and post optimization. The results are summarized in Table 4.7 and 4.8. It must be noted here that f_c is the PLL centre frequency, for which we use the mean value for the dominant frequency range from Section 4.2.2, and the standard deviation being the lock range f_l. Moreover, the combined gain $K = AK_0K_a$ (see (4.11)) is the gain after taking into account the amplitude variation in EEG through replacing it with a virtual signal with constant amplitude A for tuning purposes, as discussed in Section 4.3. We also summarized and recap the essential pre-requisites in Table 4.6; these are the values we obtained from our EEG characterization analyses in Section 4.2.

As observed from Table 4.7 and 4.8, optimization did give us better performance in PLV. More specifically, it reduced the PLV variance for all bands.

	Delta	**Theta**	**Alpha**	**Low beta**	**High beta**	**Low gamma**	**High gamma**
A	4.72	3.16	4.60	2.20	1.60	0.99	0.74
f_i (Hz)	2.18 ±1.44	6.67 ±2.78	10.24 ±2.00	14.90 ±4.03	23.34 ±3.82	33.66 ±3.59	43.99 ±3.13

Table 4.6. Summary of pre-requisites used for tuning puproses: 75th percentile amplitude A and the frequency range f_i for the lock range.

One thing we observe from these, is that in general, there is a trend towards higher loop gains required for higher EEG bands. The lock ranges is also higher for higher bands. These could be explained by the fact that higher bands have lower power or amplitude with less dominant oscillations, and hence, a higher gain would be needed for satisfactory lock. At the same time, it must also be noted that despite the higher gains for the higher bands, the PLV is slightly lower. This is because the gain cannot be increased indefinitely before the system becomes unstable; hence, there is always a limit. Due to these reasons, we see that the gain K is in general higher for the higher bands, and lower for the lower bands, where dominant oscillations are already present.

Regarding the final ζ and ω_n values achieved in Table 4.8, we see that the final values are slightly different to that of the original specs. For example, for the delta band, the specification is $\omega_n = 6.22$ and $\zeta = 0.707$, while the final values are $\omega_n = 9.15$ and $\zeta = 0.780$. The impact of this would be negligible because at the end of the day, the important thing is achieving the lock range specification as close as possible and not the individual ω_n and ζ values. Initially, the $\zeta = 0.707$ was chosen for all bands as the spec to start with, and as a good starting point. As we had seen earlier, the tuning parameters are inter-dependent and cannot be tuned independently and hence, an exact match to the specification is not expected. Instead, what we aimed for was a close enough match, with importance given to achieving the desired lock-range and more so the post-optimized accuracy, as measured by the PLV. Hence, these discrepancies are not surprising.

		Delta	**Theta**	**Alpha**	**Low beta**	**High beta**	**Low gamma**	**High gamma**
Specifications	f_c (Hz)	2.18	6.67	10.24	14.90	23.34	33.66	43.99
	f_l (Hz)	±1.44	±2.78	±2.00	+/4.03	±3.82	±3.59	±3.13
	ω_n	6.40	12.35	8.89	17.91	16.97	15.95	13.91
	ζ	0.707	0.707	0.707	0.707	0.707	0.707	0.707
Derived Parameters	τ_1(s)	0.4	0.4	0.4	0.4	0.4	0.4	0.4
	τ_2(s)	0.1408	0.1408	0.1408	0.1408	0.1408	0.1408	0.1408
	f_{cut} (Hz)	0.46	0.46	0.46	0.46	0.46	0.46	0.46
	K	3.5	20	7	55	65	92	95
	A	4.72	3.16	4.60	2.20	1.60	0.99	0.74
	K_{tot}	16.52	63.20	32.20	121	104	91.08	70.3
Resulting performance	ω_n	6.43	12.57	8.972	17.39	16.12	15.10	13.26
	ζ	0.647	0.984	0.771	1.296	1.213	1.145	1.028
	f_l (Hz)	±1.32	±3.94	±2.20	±7.18	±6.22	±5.50	±4.34
	PLV	0.87±0.02	0.89 ±0.03	0.89±0.05	0.75±0.16	0.77±0.10	0.75 ±0.08	0.74 ±0.08

Table 4.7. PLL Parameters obtained before optimization. f_c is the PLL centre frequency, f_{cut} the low pass filter cut-off frequency and K_{tot} is the combined gain.

		Delta	**Theta**	**Alpha**	**Low beta**	**High beta**	**Low gamma**	**High gamma**
Specifications	f_c	2.18	6.67	10.24	14.90	23.34	33.66	43.99
	f_l (Hz)	±1.44	±2.78	±2.00	+/4.03	±3.82	±3.59	±3.13
	ω_n	6.40	12.35	8.89	17.91	16.97	15.95	13.91
	ζ	0.707	0.707	0.707	0.707	0.707	0.707	0.707
Derived Parameters	τ_1 (s)	0.4	0.4	0.4	0.4	0.4	0.4	0.4
	τ_2 (s)	0.1408	0.1408	0.1408	0.1408	0.1408	0.1408	0.1408
	f_{cut} (Hz)	0.46	0.46	0.46	0.46	0.46	0.46	0.46
	K	7.5	14	11	25	35	52	65
	A	4.72	3.16	4.60	2.20	1.60	0.99	0.74
	K_{tot}	16.52	63.20	32.20	121	104	91.08	70.3
Resulting performance	ω_n	9.14	10.21	10.90	11.54	11.76	11.35	10.97
	ζ	0.780	0.84	0.882	0.92	0.93	0.909	0.886
	f_l (Hz)	±2.27	±2.73	±3.10	±3.38	±3.50	±3.28	±3.10
	PLV	0.88±0.01	0.88±0.01	0.91±0.04	0.85±0.03	0.83±0.03	0.80±0.01	0.78± 0.01

Table 4.8. PLL Parameters obtained after optimization.

	Delta	Theta	Alpha	Low beta	High beta	Low gamma	High gamma
f_l spec (Hz)	±1.44	±2.78	±2.00	+/4.03	±3.82	±3.59	±3.13
f_l achieved (Hz)	±2.27	±2.73	±3.10	±3.38	±3.50	±3.28	±3.10
Error (Hz)	$+0.83$	-0.05	$+1.1$	-0.65	-0.32	-0.31	-0.03

Table 4.9. PLL operating ranges specifications compared to the ones achieved post optimization, along with the error between both.

Finally, Table 4.9 summarizes the lock range spec and the achieved lock range, alongside the error. As observed, for the alpha and delta band, the lock range specs were exceeded by ±0.87 Hz and ±1.20 Hz respectively. This could be explained by the fact that these bands had the most dominant EEG power, and strongest oscillations as evidenced by the higher A for these bands, which makes locking easier for the PLL. This is also expected since it is an eyes-closed resting state data-set where these EEG bands are known to have higher power.

For the remaining bands the achieved lock range was lower than that of the specs, with the error increasing for higher EEG bands. This again is because for the higher EEG bands the amplitudes are lower with less dominant oscillations, and hence locking becomes harder here, as there is a limit to how much the gain could be increased before the PLL breaks down. This may explain the poorer performance for these bands, both with regards to the lock range achieved and its locking accuracy, as measured by the PLV.

4.3.4 Summary

In this Section we detailed our PLL tuning methodology: we use a bottom up tuning approach, where we first get a good initial starting point for the PLL parameters using PLL maths and control theory, after which we fine tuned and optimized, to achieve the best accuracy. We demonstrated this procedure for all EEG bands using the eyes-closed resting state EEG data-set, with results summarized in Table 4.8. These would be then used in the following Section, to characterizing PLL accuracy.

4.4 Phase Locked Loop for continuous stimulation: EEG phase

4.4.1 Introduction

In this Section, we aim to demonstrate EEG phase tracking performance across all bands, using the tuning parameters obtained in Section 4.3.

For this, we conduct three different experiments: inter-channel accuracy (Section 4.4.2), inter-subject accuracy (Section 4.4.3) and amplitude dependent performance (Section 4.4.4). The first two characterizes the PLV and its variance between channels and subjects, respectively, to ensure it works consistently without any need for channel specific or subject specific tuning. The third demonstrates how the PLL accuracy is amplitude dependent.

Most importantly, in this Section, we conduct these analysis for all EEG bands, for a wider range of potential applications. Our aim is to show that the PLL is superior to the state-of-the art over all bands, and hence, could be easily adapted to other applications, where their bands of interest may be different from ours.

4.4.2 Inter-channel analysis

Introduction

In this Section, we characterize the inter-channel accuracy of the PLL, to see if the PLL performs well consistently across all channels. This is important as the channel choice is application-dependent and more so since this Section is about the wider applicability of the PLL across all bands for a wider variety of applications. We do not want to tune individually for each channel.

Methods

To assess phase tracking accuracy and inter-channel spread, we used data-set 1 from Table 4.1; this data-set was used as this is an eyes-closed resting state data-set that would contain not only strong delta, theta and alpha waves, but as with any other EEG data, would also contain power in the higher bands. Moreover, we use the same tuning parameters obtained for this data-set in Section 4.3; the parameters are summarized in Table 4.8 from Section 4.3, where we had demonstrated the tuning procedure for this data-set, with values obtained for all EEG bands. Note that here the tuning parameters were obtained for channel Fp1; although it is possible to individually tune for each channel to optimize, this would be cumbersome and impractical for various applications, especially those involving lots of channels (e.g. 72 channels used in our data-set). Hence, we aim to assess if the performance obtained over the entire head would be good enough, despite using the parameters obtained from tuning channel Fp1. Finally, we use a 70–30 training-test split, to avoid any in-sample optimisation; this results in data from 7 subjects used for testing.

For evaluating the phase tracking accuracy, we used the PLV (see 4.14) as the metric of choice to facilitate comparison to state-of-the-art work by [44], where they used the same metric for characterizing accuracy over all bands. With this, for each channel, we append data from all subjects into one single signal, and then calculate the PLV over the signal for each channel. The accuracy is then defined as:

$$\text{Accuracy} = PLVChannel_{mean} \pm PLVChannel_{std} \tag{4.15}$$

where, $PLVChannel_{mean}$ is the mean of the PLV values obtained across all channels (1 PLV value per channel) and the $PLVChannel_{std}$ is the standard deviation of the PLV val-

	Delta	**Theta**	**Alpha**	**Low beta**	**High beta**	**Low gamma**	**High gamma**
PLV	0.87 ±0.01	0.83 ±0.06	0.90 ±0.04	0.81 ±0.06	0.78 ±0.06	0.73 ±0.06	0.70 ±0.06
PLV State-of-the-art [44]	0.71 ±0.03	0.66 ±0.03	0.77 ±0.08	0.63 ±0.04	0.63 ±0.04	0.62 ±0.04	0.62 ±0.05
Improvement	16%	17%	13%	18%	15%	11%	8%

Table 4.10. PLL inter-channel accuracy compared to the state-of-the-art.

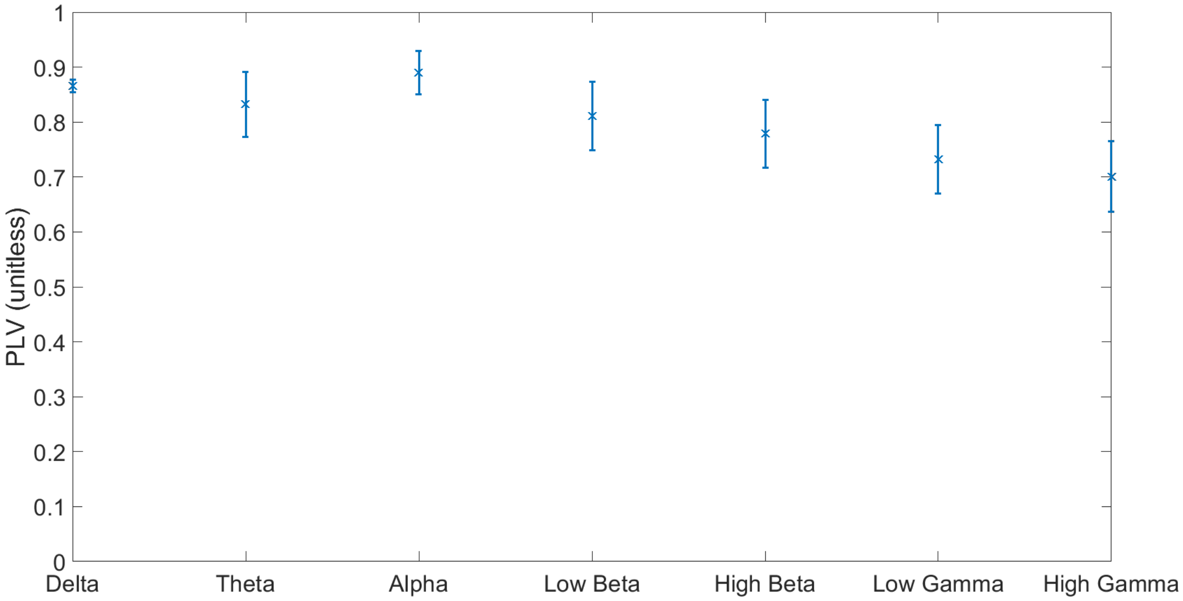

Figure 4.7. PLL inter-channel variance plots.

ues for all channels, which shows the inter-channel PLV spread. With this, we can then characterize the PLLs inter-channel variability, along with its phase-tracking performance.

Results and Discussion

For the results, firstly, we see from Fig. 4.7 and Table 4.10 that the inter-channel variance is smallest for the alpha band and the delta band. This would be because these bands have the most dominant oscillations, as evidenced earlier in Section 4.2.4 and hence, the variance in error would be smaller for these. We also depict this in the head-plots in Fig. 4.9 where we plot the 75th percentile amplitude from the EEG characterization done in Section 4.2.4; the topo-plot here gives the amplitude distribution for the different EEG channels. Here, we see that the EEG amplitude for the alpha and delta band are much higher compared to the others (in the range 2 and 4) across all channels. This then confirms the lower variance observed, as the PLL is better able to track signals which have higher amplitudes. Section 4.4.4 details this amplitude dependent performance of the PLL, in more detail.

We also observe from the topo-plot in Fig. 4.8 is that there appears to be a localized region of lower PLV, which is consistent across all bands, around the parietal and occipital areas. This we found is again due to the amplitude dependent performance of the PLL. The lower amplitude regions in Fig. 4.9 coincides with the low PLV areas in Fig. 4.8. Similarly,

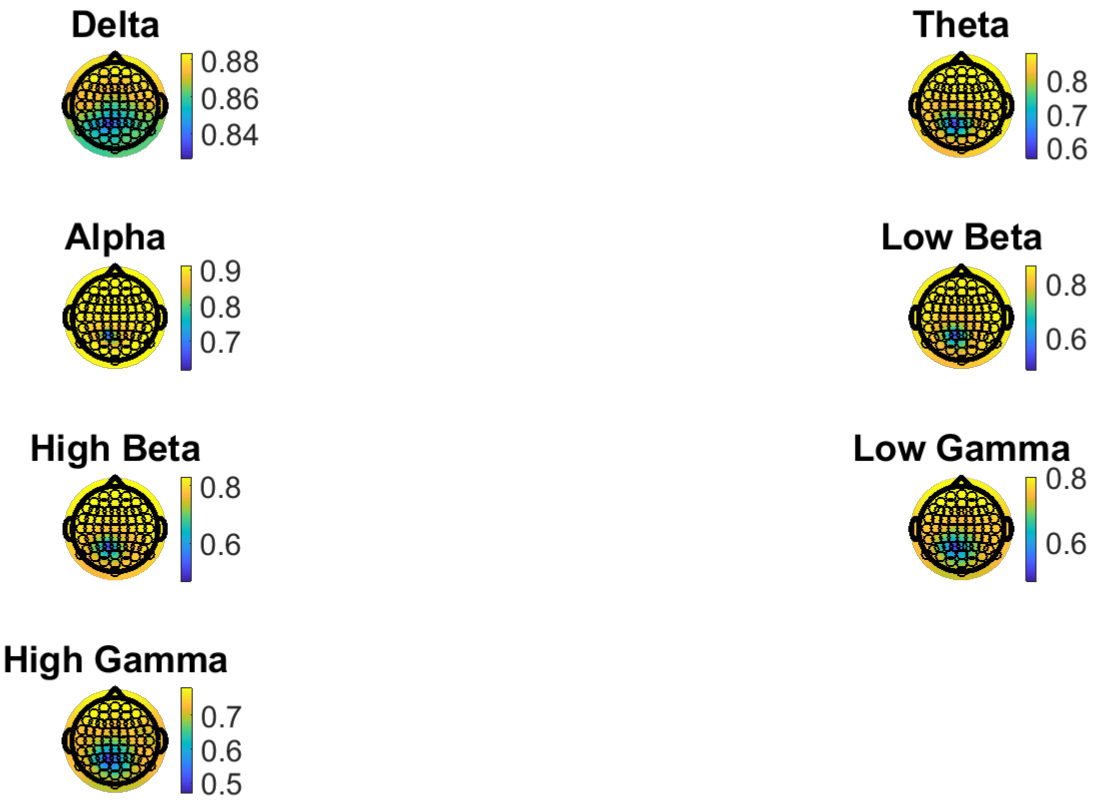

Figure 4.8. Topoplot of PLV values across all channels.

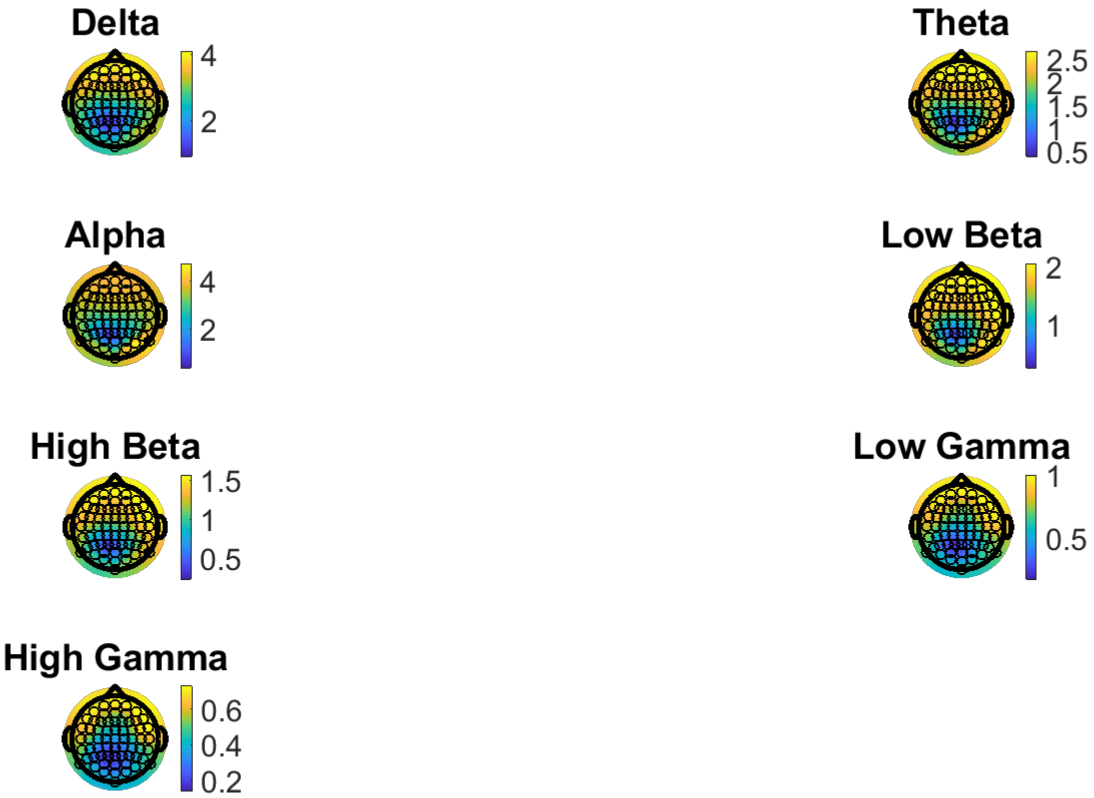

Figure 4.9. Topoplot of 75th percentile amplitude A for all channels.

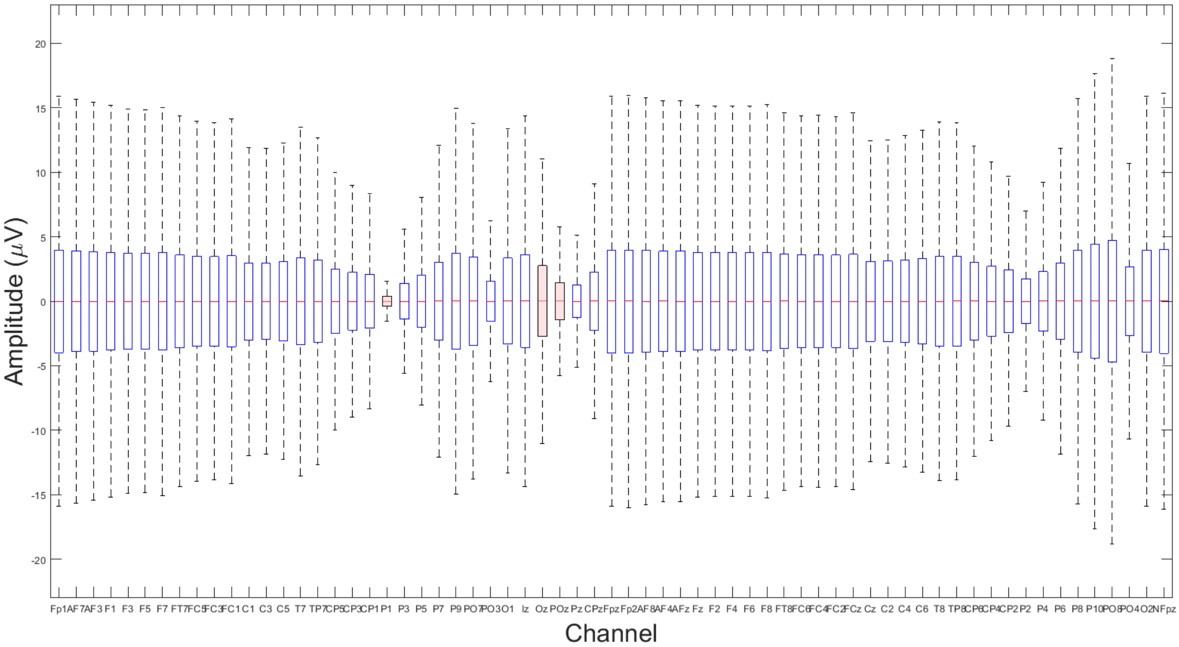

Figure 4.10. EEG Amplitude distribution for all channels. P1, POz and Oz highlighted in red to show examples of channels that have lower amplitudes.

the amplitude box-whisker plot in Fig. 4.10 for the alpha band, shows that the parietal and occipital channels such as P1, POz and Oz (highlighted in red), all have lower amplitude, compared to the other head regions. Both these pieces of evidences then suggest that the localized region of lower PLV in the back of the head, is caused by weaker oscillations in the region.

Finally, comparing our results to the state-of-the-art in Table 4.10, we see that the PLL performs on average 14% better (average of improvements made across all bands) than the state-of-the-art, where they also characterized the inter-channel variance using the PLV metric (see Table 4.10) [44]. For their work, they characterized inter-channel accuracy on 5 subjects (5 minutes of EEG per subject) for a 64 channel setup; this equates to a total of 1,600 minutes of eyes-closed data. Ours is also an eyes-closed data-set and we use data from 7 subjects (4 minutes of EEG per subject), for a 74 channel setup; this then equates to a total of 2,072 minutes of data [142]. Hence, the data-set, type and amount used in our case is similar to the state-of-the-art, with ours being slightly higher [44]. Moreover, we calculated inter-channel spread over 74 channels, while they did over 64 channels [44], [142]. This may explain the slightly higher inter-channel variance in our case for the theta, low-high beta and low-high gamma bands. However, for the alpha and delta bands, the inter-channel variance was lower in our case, compared to the state-of-the-art, and this may again be due to the dominant oscillations present here, which improves PLL performance, as detailed in Section 4.4.4. Moreover, this is what we want, as the alpha and delta bands are our primary application drivers. Regardless, for all bands, we note that the PLL is superior to the state-of-the-art, with a mean PLV value that is on average 14% better than the state-of-the-art [44].

Conclusion

In summary, for the inter-channel analysis, we found that the PLL performs consistently across all channels, with low inter-channel variance. The alpha and delta band inter-channel variance was the lowest, which is expected due to the stronger oscillations present in these bands. On the contrary, channels with lower amplitudes were shown to have higher variance, again due to the amplitude-dependent performance of the PLL. Regardless, over-all, the low inter-channel variance suggests that the PLL does not have to be tuned separately for each channel, making it application agnostic. More importantly, we also show that the PLL performs on average 14% better than the state-of-the-art, in tracking EEG phase.

4.4.3 Inter-subject analysis

Introduction

For the inter-subject analysis, we aim to find out the variance in PLL phase tracking performance over different subjects, for a single channel. The inter-subject variance would then help us understand how well the PLL adapts to EEG from different subjects, without needing to be tuned on a subject-by-subject basis. In contrast to these, other phase targeting algorithms like the one by Ngo et al. involves tuning on a subject-by-subject basis, as detailed in Section 2.6.2 [63]. This would be not ideal, as the aim is to have a 'fit-and-forget' solution.

Methods

For characterizing performance, we used data-set 1 from Table 4.1, to keep it similar to the inter-channel analysis done in Section 4.4.2. Hence, the tuning parameters and the number of subjects used (n=7) after the 70–30 train-test split, all remain the same.

Similarly, we used the PLV (see 4.14) as the metric of choice; the PLV is calculated over each subject's recording i.e. 4 minutes of data. The accuracy is then defined as:

$$\text{Accuracy} = PLV_{mean} \pm PLV_{std} \tag{4.16}$$

where, PLV_{mean} is the mean of the PLV values of all subjects and the PLV_{std} is the standard deviation of the PLV for all subjects, which shows the inter-subject PLV spread. With this, the higher the PLV_{mean} and the lower the PLV_{std}, the better the phase tracking performance would be across subjects.

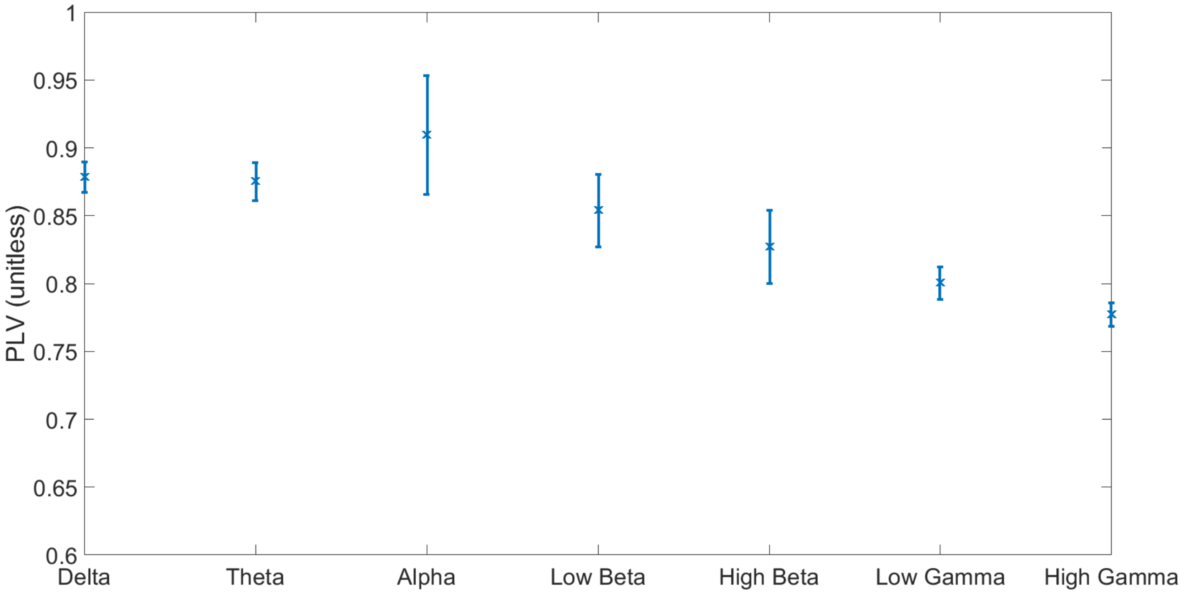

Figure 4.11. EEG Phase tracking accuracy for all bands using the PLV metric.

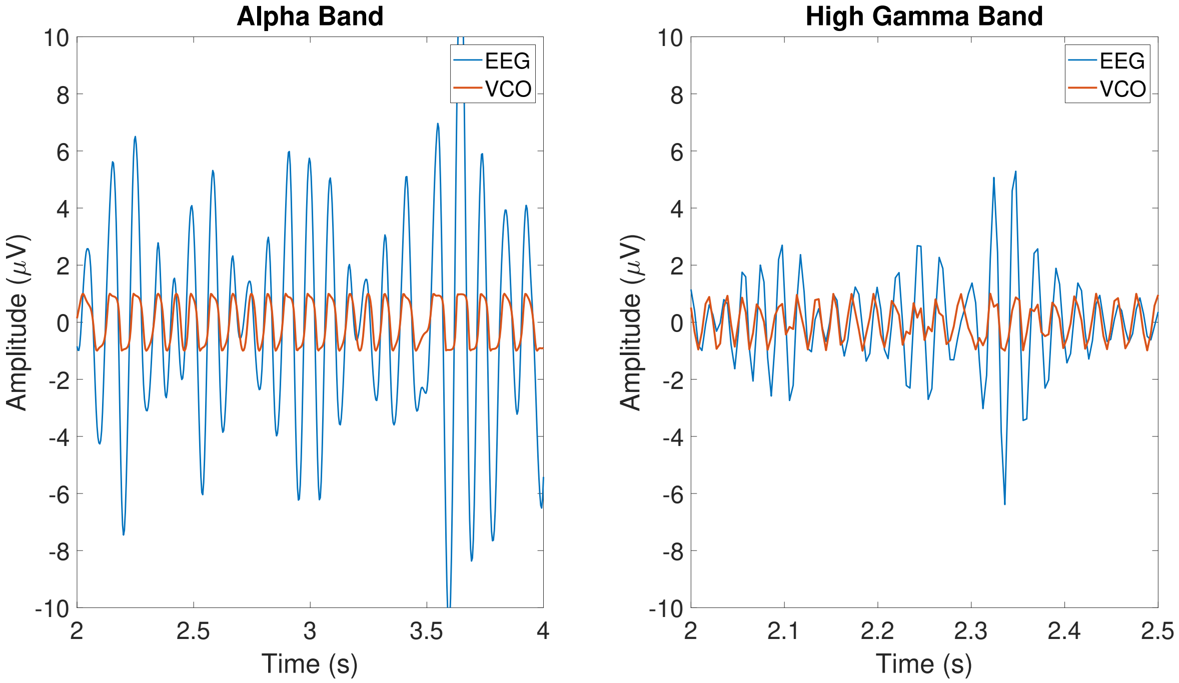

Figure 4.12. Sample trace of PLL output vs the band-pass filtered EEG.

Results and Discussion

Here, we present the results for the inter-subject phase tracking performance. As observed from Fig. 4.11 and Table 4.11, the PLV performance is best for the alpha and delta band, and in general worsens for higher bands, which we also observed in the inter-channel analysis done in Section 4.4.2.

This is further evident from the raw plots in Fig. 4.12, where we see that the PLL output is less oscillatory for the higher bands. As mentioned earlier in Section 4.4.2, better performance in the lower bands are expected since we used a resting-state EEG data-set, where these bands are known to have stronger oscillations, which we also saw from our amplitude characterization analysis in Section 4.2.4 and is evident from the raw EEG plot in Fig. 4.12.

	Delta	**Theta**	**Alpha**	**Low beta**	**High beta**	**Low gamma**	**High gamma**
PLV	0.88 ±0.01	0.88 ±0.01	0.91 ±0.04	0.85 ±0.03	0.83 ±0.03	0.80 ±0.01	0.78 ±0.01

Table 4.11. PLL phase tracking performance across subjects.

Most importantly, higher accuracy for the alpha and delta bands, as observed here, is what we want, as these are the bands of focus for our application drivers.

Finally, we note that the inter-subject spread is low (<4% in most bands), which is even lower than the inter-channel spread (<6% in most bands). We also note here that the inter-subject spread was not reported in the work by Masnsouri et al. and hence, we cannot make a fair comparison here.

Conclusion

In conclusion, in this Section, we characterized the inter-subject spread of the PLL phase-tracking performance; we showed that the inter-subject spread is low (<4% for most bands), which suggests that subject-specific tuning may not be necessary.

4.4.4 EEG amplitude dependant performance

Introduction

In this Section, we explore in more detail how the PLL performance is amplitude dependent, an observation we found consistently in earlier analyses. Traditionally, PLLs are used in applications where the input signal to be tracked has constant amplitude such as in clock-synchronization [145]. However, in our case, the non-stationary EEG has amplitude that varies both over-time and across subjects, which makes it more difficult to tune the PLL in a deterministic way, since the amplitude would have an effect on the effective gain of the PLL (see equation 4.11 from Section 4.3). Hence, it would be useful to conduct an analysis on how amplitude variations affect PLL performance.

Methods

To demonstrate the amplitude dependent performance, we run the PLL both on a synthetic signal and the EEG. The former will help augment our confidence for observations made on the latter.

For the synthetic signal, we simply vary amplitude of a sinusoid and observe the resulting change in performance. For this, we used 5 different sinusoids with amplitudes $1V$, $4V$, $7V$, $10V$ and $13V$. For each amplitude, we then conducted a frequency sweep from 0–25

Hz and then measured the lock range for a PLL designed to track a 10 Hz signal; we define the lock range as the frequencies for which the PLV is close to 1 i.e. perfect phase lock, which is expected around the 10 Hz mark. For frequencies outside the lock-range, the PLV is expected to be low.

The PLL here was tuned to track a 10 Hz signal perfectly well (PLV=1) i.e. the alpha band; we are just using this frequency as an example, as the purpose here is to demonstrate amplitude dependence, which would be the same regardless of the frequency band of choice. For this, we used the same PLL for the alpha band from Table 4.8 but with a gain of 1 to account for the different amplitudes here for the synthetic signals. The gain was optimized so that the PLL would perfectly track the 10 Hz signal within the lock range, with a PLV of 1. Using these, we can then compare the lock ranges for each of the different signals with its different amplitude, and observe any amplitude dependent variations in accuracy.

After demonstrating the behaviour on synthetic signals, we then do the same with EEG data, this time using data-set 1 from Table 4.1 and we use the training set (n=15). Fp1 channel was used and we appended data from all subjects into one signal before processing it. Then, as with the synthetic signal, we use the alpha band and hence, filter the raw EEG with a band-pass filter (IIR butterworth order 2 with pass band of 8–13 Hz). Then we passed the filtered signal into the PLL to get the PLL based sample-by-sample instantaneous phase measures Θ_{PLL} and compare this phase to the same obtained using the gold standard Hilbert based method $\Theta_{Hilbert}$. Also note that 1s of data was discarded from the end and beginning, to remove edge artefacts associated with the Hilbert based method. Moreover, for the PLL based measure, the PLL parameters used for the alpha band are summarized in Table 4.8 of Section 4.3. With this, to demonstrate the amplitude dependent performance, we plotted the phase error against the amplitude for each sample. The phase error at each sample is defined as:

$$\text{error} = \Theta_{Hilbert} - \Theta_{PLL} \tag{4.17}$$

where, $\Theta_{Hilbert}$ is the instantaneous phase obtained using the gold standard Hilbert transform and Θ_{PLL} is the phase obtained using the PLL.

Results and Discussion

Firstly, for the synthetic analysis, the lock ranges for varying amplitudes is shown in Fig. 4.13. We observe that as the amplitude increases, so does the lock range of the PLL. In essence, this then means that when amplitude is higher (lock range higher), the PLL tracking performance improves, which translates to lower phase error. This is what we observed in previous Sections; for example, in the inter-channel analysis, we saw that for channels with higher amplitude, the phase tracking accuracy was higher and vice-versa for channels with lower amplitudes.

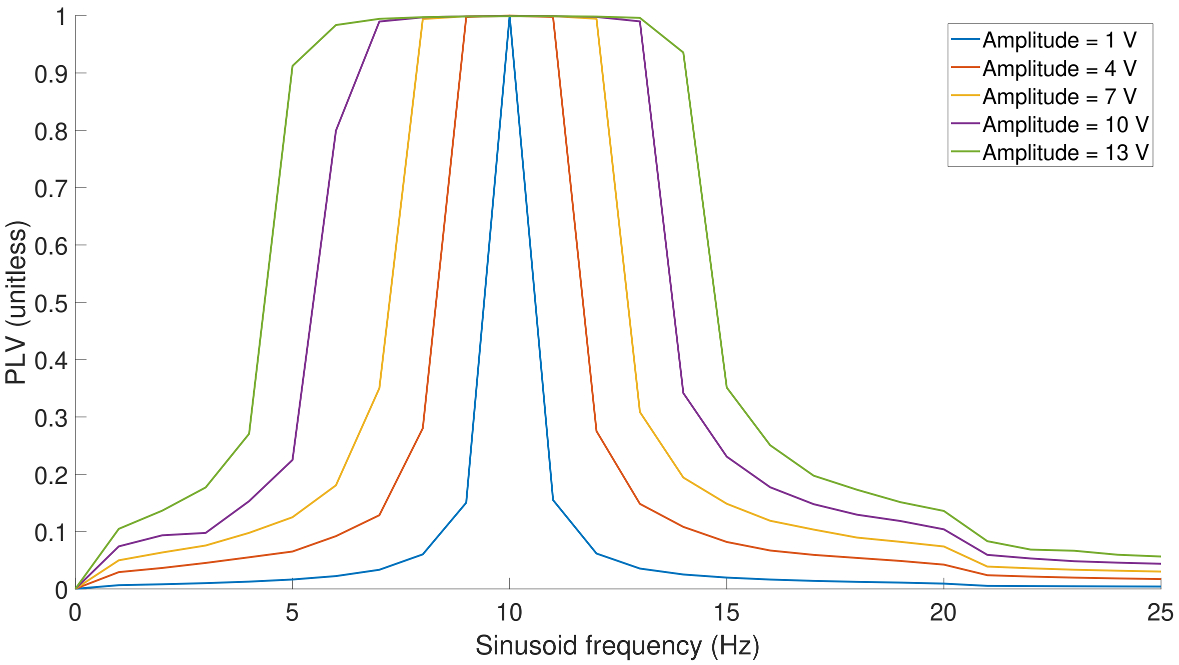

Figure 4.13. PLL amplitude dependent performance using synthetic signals.

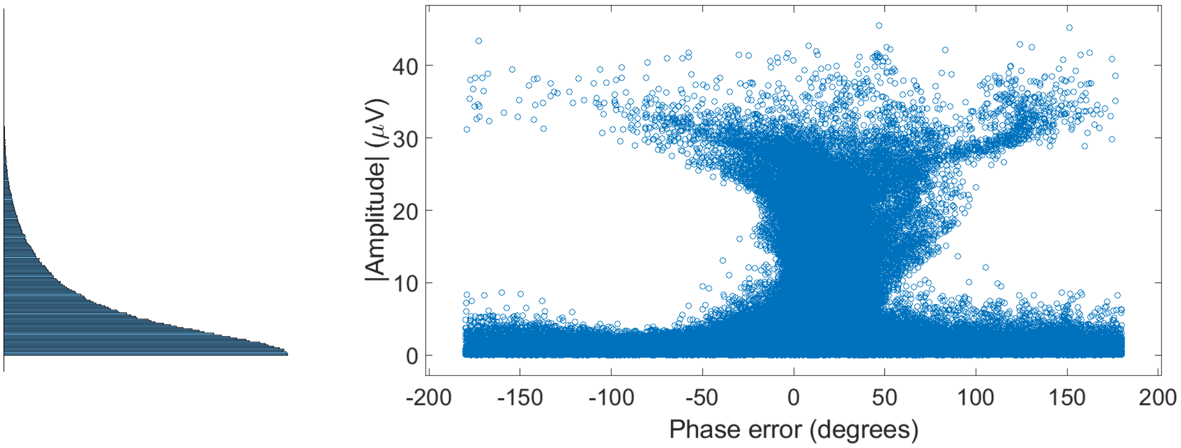

Figure 4.14. PLL amplitude dependent performance using EEG signals. Phase error per sample plotted against its amplitude at that sample. Marginal histogram of the amplitude values plotted along y axis.

For the EEG signal, when plotting the phase error at each point against its amplitude, we observe the same phenomena as seen in Fig. 4.14. Here, we see that when the phase error is smaller, the EEG amplitude tends to be highest and vice-versa for large phase errors, which again confirms observations made in previous Sections. Moreover, we note that there are some outliers; sometimes, for higher amplitudes the phase error is large but these are not the majority and is most likely due to periods when the PLL is out of lock. In other words, at points when the PLL is out of lock, even if the amplitude is large, it would take longer than a sample to get back in lock, which may be the cause of these outliers.

Overall, this amplitude dependence then means that deterministic tuning of the PLL is not very straightforward, due to the time-varying nature of EEG amplitude. This is exactly why we did EEG amplitude characterization in Section 4.2.4, and used the virtual constant amplitude A (75th percentile EEG amplitude) for PLL tuning purposes in Section 4.3. This allowed us to capture and take into account at least some of this amplitude variability, into the PLL tuning process.

Moreover, it must be noted here that in theory, one could add in some sort of amplitude normalization such as an automatic gain control, or an XOR-based phase detector. However, both these would add its own complications; the automatic gain control would incur additional computational complexity, in addition to its own tuning parameters. Similarly, the XOR based phase detector would again dilute the signal information of the EEG, as there would be a signal conversion stage from EEG to square-waves, which is then fed to the XOR phase detector. This conversion, say using a Schmidt trigger, would involve setting upper and lower thresholds, and hence, adds again additional tuning parameters. These additional complexities are unnecessary, as other groups using a PLL have reported clinical effects, without any of these [2], [4], [5]. For example, for the sleep engineering application, Santostasi et al., Ong et al. and Palambros et al., all used a classical PLL without any amplitude normalization, with which they reported clinical effects i.e. enhancement of slow oscillations, as detailed in Section 2.6.2 [2], [4], [5]. Thus, given our aim is to have a real-time implementation of the closed-loop system on a power-resource constrained device such as the phone, we considered the added computational and tuning complexities associated with amplitude normalisation procedures not to be beneficial as this point. Moreover, we also highlight here that even without any amplitude normalization, our phase-tracking performance is on average 14% better than the state-of-the-art, for all EEG bands. Hence, for these reasons, we chose to use the classical PLL by itself without any amplitude normalisation.

Conclusion

In conclusion, we have shown in this Section that the PLL performance is amplitude dependent, confirming observations made in previous Sections. We have shown through our synthetic analysis that the PLL lock range increases with increasing amplitude and vice versa. This in turn means that higher amplitude would equate to better PLL tracking accuracy, which is exactly what we found in the EEG analysis, where regions of high amplitude were characterized by low phase error.

4.4.5 Summary

In this Section, we aimed to demonstrate the PLLs ability to track EEG phase well across all bands, for wider applicability beyond our application drivers. Here, we showed that the PLL tracks well across all bands and performs on average 14% better than the state-of-the-art, for all bands. We also showed that the inter-channel and inter-subject spread is low, which negates any need for channel or subject specific tuning. Moreover, we also showed that the PLL performance is amplitude dependent, where performance improves for higher amplitudes, which also explained the improved performance for the lower EEG bands in the inter-channel and inter-subject variance experiments.

4.5 Phase Locked Loop for discrete stimulation: EEG phase

4.5.1 Introduction

In the previous Sections, we detailed the PLLs performance across all EEG bands, for general applicability. In this Section, we are going to zoom in on just the delta band, and more specifically, the SO band (0.5–3 Hz) within it, which contains the slow oscillations. This is for the sleep engineering application driver, as detailed in Section 2.6.2. There, we noted that the optimal target phase for highest efficacy is being explored by various groups as part of the clinical work being done in this field. In addition to this, people are also looking at the technology side of things, developing new platforms that are faster and more accurate. We are aiming for a technology contribution via providing a smart-phone based mobile platform that could be used for *in-the-wild* experiments. All these are discussed in more depth in Section 2.6.2.

Here, in this Section, we then characterize the PLLs accuracy in tracking slow oscillation phase on a PC, before moving onto the real-time on-phone implementation in Chapter 5. It must be noted that for this application, we are targeting discrete phase points for one-off stimulation, and not continuous matching, as done in Section 4.4 and 4.6. This reflects the two modes of stimulation: one continuous where we react to every change in the EEG on a sample by sample basis, and one discrete where we want to detect a particular event and react to that, which in the sleep-engineering application is the phase of a SO, as detailed in Section 2.6.2

4.5.2 Methods

For this analysis, since we are focusing on the sleep engineering application driver, we used the sleep data-set which is data-set 2 from Table 4.1. As detailed in Section 4.1.2, the raw EEG data was filtered in the SO band (0.5-3 Hz) and 5 s slow oscillations epochs were extracted by our collaborators using the algorithm in [127], which resulted in a total of 37,440 5 s SO epochs over the entire 13 subjects. Moreover, just like in Section 4.4, we used a 70–30 training-test split, to avoid in-sample optimisation. With this, we then used these epochs to characterize phase tracking performance, aiming to target the peak of the slow oscillation.

For the PLL, it must be noted here that the same PLL settings from Section 4.4 was not used, as these were tuned for the delta band (0.5–4 Hz) and not the SO band (0.5–3 Hz) specifically, which is defined differently from the delta band. Slow oscillations typically have higher amplitude and hence, the settings used in Section 4.4 for the delta band would be sub-optimal here, especially considering the amplitude dependence of PLL performance. For these reasons, we chose to tune the PLL separately, despite using the same methodology. First, we characterized the EEG data via its virtual amplitude A and the frequency range to take into account the amplitude dependent performance and lock-range settings, respectively. Using these, we tuned the PLL using the same procedure as in Section 4.3:

defining the operating range (lock range), deriving the needed ω_n and ζ from the specs, designing the gain K and filtering coefficients τ_1 and τ_2 to meet the ω_n and ζ spec as close as possible, and finally optimizing these initial parameters to get best possible phase-targeting accuracy.

Moreover, in previous Sections, we used the PLV as the metric for accuracy, which is relevant for continuous stimulation. In contrast, here, since we are interested in discrete stimulation, we need to find phase error at a discrete time-point, and not over the entire epoch. For this reason, the metric we used was the phase error, which is also the metric used by other sleep engineering platforms [2], [4], [5], [12], [13], [20]. More specifically, we use phase error at the peak, where the error is,

$$\text{error} = \theta_{Hilbert} - \theta_{PLL} \tag{4.18}$$

where, $\theta_{Hilbert}$ is the instantaneous phase extracted using the gold standard Hilbert transform and θ_{PLL} is the instantaneous phase extracted by the PLL. This metric would give an error value for each sample point, and we then measure the error at the peak. This is consistent with the characterization done by other work in this field, despite different target phases; some target the peak and other's the negative peak to test various hypothesis. Regardless of the target phase, this error metric would help compare our work against other's in the field.

4.5.3 Results and Discussion

Here, we present the results for the phase tracking accuracy. We first start with the pre-requisites, which is used to obtain the tuning parameters, after which we present the phase accuracy results. This is in line with the tuning methodology in Section 4.3, but this time done for the SO band for a sleep data-set, to target the sleep engineering application driver.

Pre-requisites

For the pre-requisites, the virtual amplitude A, alongside the dominant frequency range were obtained first (see Section 4.3 for recap). The results are shown in Fig. 4.16, 4.17 and summarized in Table 4.12.

Firstly, Fig. 4.15 shows a sample SO trace, to give an idea of slow oscillation morphology. As expected, we see larger than usual EEG amplitudes ($> 40 - 70\mu V$), which is characteristic of SO oscillations during sleep [124].

With the box and whisker plot in Fig. 4.16 for the amplitude distribution, we get the virtual amplitude $A = 40\mu V$ as the 75th percentile value; note that these are much higher than the ones obtained from the data-set used in Section 4.2.4, for the delta band waveform ($A = 4.72\mu V$). This is because the current one is slow oscillation from sleep data, which

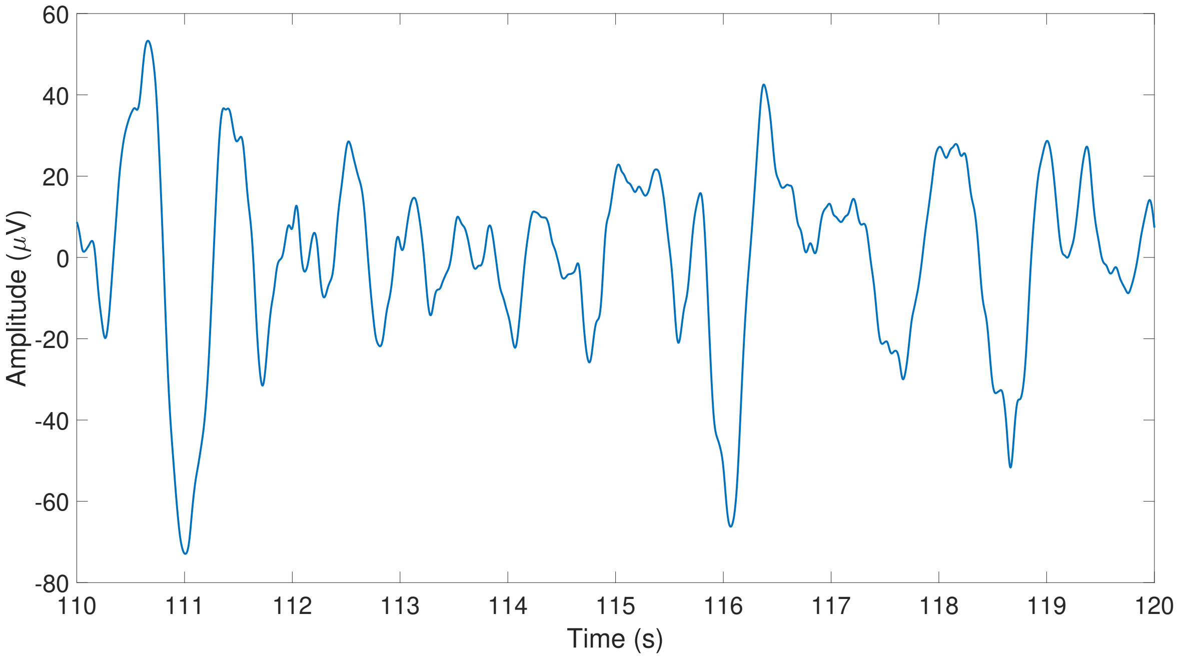

Figure 4.15. Raw trace of a sample slow oscillation. The figure shows 2 slow epochs concatenated, each of 5 s length.

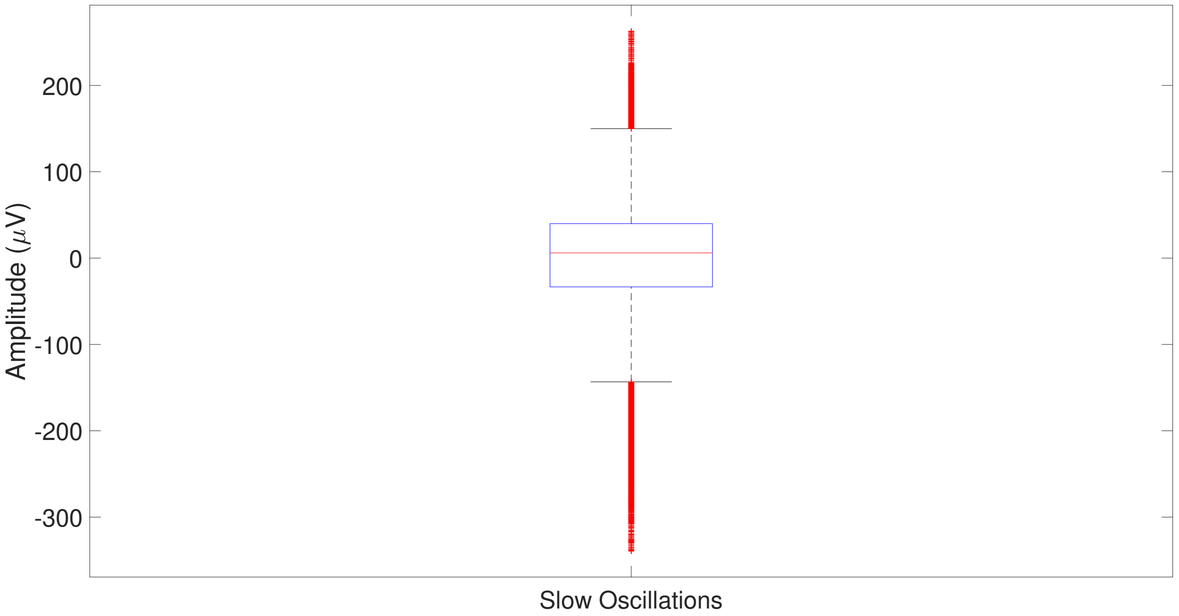

Figure 4.16. Amplitude distribution of the slow oscillations.

is known to have larger oscillation, while the data-set used in the previous sessions were taken when awake (eyes-closed resting state), which explains the lower delta-band amplitude. This further confirms the need to tune the PLL separately for the sleep engineering application, primarily due to the amplitude dependent performance of the PLL.

Finally, we also obtained the dominant frequency range, which is used to define the specs for the lock range for tuning purposes. For this, we used the same procedure as in Section 4.2.2: band-pass filtering the raw data in the SO band; using the Hilbert transform to get instantaneous phase and subsequently the instantaneous frequency from its derivative; and finally smoothing the raw instantaneous frequency (low pass order 2 IIR butterworth filter with a 2.5 Hz cut-off) to remove the derivative induced noise. The resulting lock-range spec is

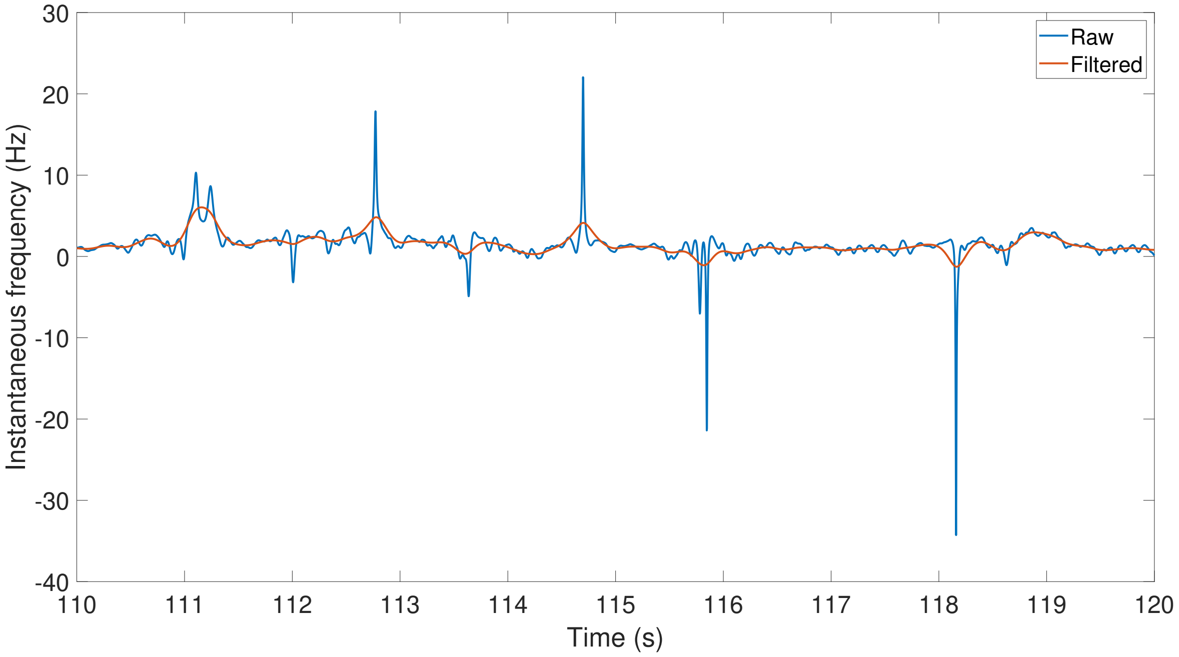

Figure 4.17. Instantaneous frequency for the slow oscillations.

Pre-requisites		Specifications				Derived Parameters						Resulting performance		
A	Frequency range (Hz)	f_c (Hz)	f_l (Hz)	ω_n	ζ	τ_1 (s)	τ_2 (s)	f_{cut} (Hz)	K	A	K_{tot} $(K \times A)$	ω_n	ζ	f_l (Hz)
40	1.17 ±1.99	1.17	1.99	8.84	0.707	0.4	0.1408	0.46	0.65	40	26	8.06	0.722	1.85

Table 4.12. Tuning parameters for the sleep engineering application.

summarized in Table 4.12. Moreover, Fig. 4.17 shows the raw instantaneous frequency plotted against the filtered one; note again that here, the spikes in the raw waveform are the derivative induced noise, hence the need for the smoothing filter.

Tuning parameters obtained

The final tuning parameters obtained using the procedure in Section 4.3 are summarized in Table 4.12. As noted, the filter coefficients are same but the gain $K = 0.65$ is much lower than the one obtained for the data-set in Section 4.3, where the gain for the delta band was $K = 7.5$. This is expected since we saw that the virtual amplitude $A = 40\mu V$ for the slow oscillations here are much higher than the one for the data-set in Section 4.2.4, where $A = 4.72\mu V$. Hence, to compensate for the larger amplitudes, we get a much lower gain value for the sleep data-set.

Phase accuracy

Next, we present the phase accuracy obtained with these tuning parameters. When tracking the slow oscillations, we get a phase error of 19.7±25.2 degrees over the entire 5 s epoch and 11.2±11.2 degrees for the phase error at the peak. This is further depicted in Fig. 4.18a and 4.18b.

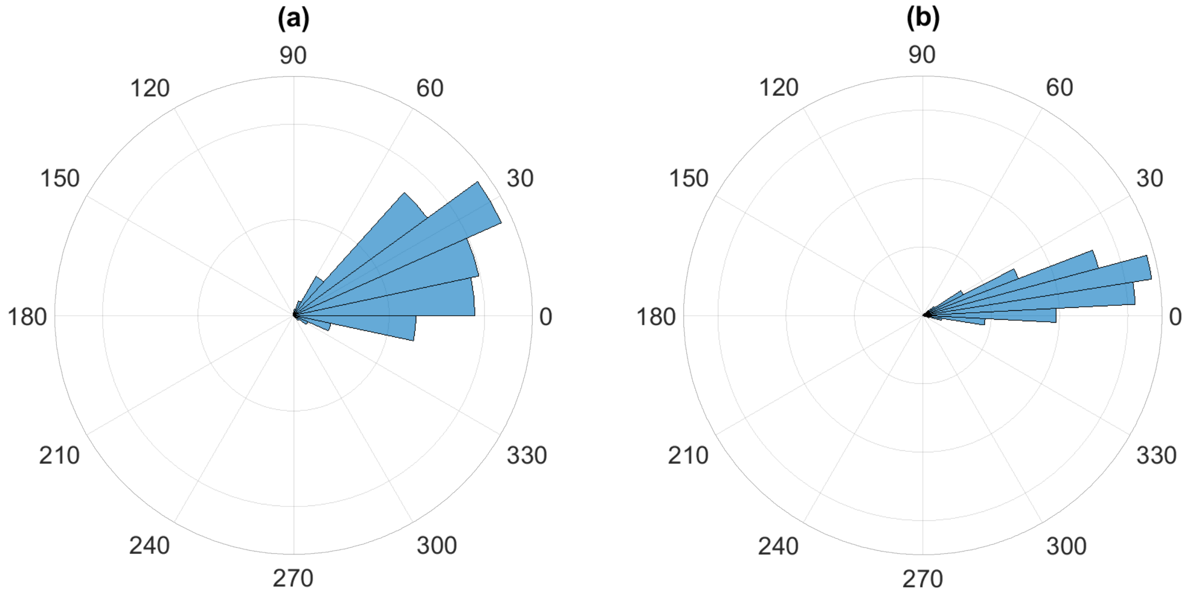

Figure 4.18. Phase error rose plot: Error over epoch in the left, and error at peak on the right.

Comparing this to the state-of-the art platforms for sleep engineering: Santostasi at el. obtained 12.5±29 and 13.6±9.88 degrees phase error over the entire epoch and the peak, respectively for their offline analysis using a PLL [2]. They used data from 5 subjects (2 night recordings per subject), while we used single night recordings from 4 subjects for testing. Moreover, both ours and their studies used the Fpz channel for the analysis. Hence, the results are comparable and our performance is similar and does not deviate too far off; however, we highlight here that our differentiation here is not in the offline results but in the on-phone implementation in Chapter 5. The current PLL based platforms are lab based and not suitable for *out-of-the-lab* use [2], [4], [5]. Moreover, it must be noted here that other sleep engineering platforms do not conduct an offline analysis before porting it for real-time implementation, and hence a comparison could not be made at this point [4], [5], [12], [13]. However, we do compare accuracy of our on-phone implementation against all these platforms, in Section 5.3 of Chapter 5.

4.5.4 Summary

In this Section, we tuned a new PLL for the sleep engineering application where discrete phase-locked auditory stimulation is the requirement, and target phase was the peak of the slow oscillations. We showed that using the PLL resulted in a phase error of 11.2±11.2 degrees when targeting the peak, which is comparable to state-of-the art platform in this field. This then lays the foundation for the on-phone implementation in Chapter 5, which is where the technology contribution lies.

4.6 Phase Locked Loop for continuous stimulation: EEG frequency

4.6.1 Introduction

So far, we have looked at the phase feedback measure in both continuous mode (general applications) and discrete (sleep engineering) mode. In this Section, we explore EEG instantaneous frequency, as the feedback measure. As discussed in Section 2.6.3, the aim is to provide frequency-matched continuous stimulation for the chronic pain application driver and provide an ambulatory platform, to enable ‘out-of-the-clinic’ research.

Since we are focusing on just the chronic pain application driver for frequency, we only consider the alpha band and not the others, as done for the continuous phase in Section 4.4. For the latter, there were state-of-the art algorithms to compare to, whilst the same does not exist for frequency-matched stimulation. Hence, for these two reasons, we chose to focus just on the alpha band and limit scope.

Other studies doing frequency matched stimulation are detailed in Section 2.6.3; for example, the work by Pino et al. where they used frequency matched visual stimulation to reduce anxiety [115]. However, as detailed in Section 2.6.3: they do not do any technology validation characterizing its accuracy, as a consumer grade EEG device (Neurosky) was used with the frequency values obtained directly from the headset SDK, on a second-by-second basis [115]. Moreover, they did not have an option for binaural beats, but just visual stimuli. In contrast, we aim to track frequency on a sample-by-sample basis (theoretical limit) and match both audio and visual stimuli to these frequency changes, all on a smartphone. In this section, we characterize the accuracy of our frequency matching algorithm offline, before porting it onto a phone in Section 5.4.

When it comes to EEG alpha frequency, most literature refer to it as the Individual Alpha Frequency (IAF), which is the dominant frequency calculated over a time-period: typically, the IAF is calculated for a single subject [16], [49], [119], [120] and at times over an epoch length [115], [118]. The IAF is usually calculated by taking the power spectrum over an epoch and choosing the peak frequency as the IAF over that epoch [16], [49], [118]–[120]. However, we are not interested in such epoch based calculations. Instead, we focus on getting the instantaneous frequency on a sample-by-sample basis, as this would be ideal case for a closed-loop platform, where latency needs to be driven down. Also, it must be noted that it is possible to get sample-by-sample measures using epoch-based calculations; for example, calculating over a window starting from the current time-point and a set of past values. However, this would increase computation time and hence increase latency, in addition to the edge effects that would be incurred at window end (current-time point) and start, which would have to be discarded. Hence, a sample-by-sample measure, calculated on a sample-by-sample basis would be ideal to drive latency down. Moreover, it is also worth noting that once we have a sample-by-sample measure of instantaneous frequency honed, then by default this could be used later on for epoch based measures, as needed.

This then raises the question as to whether the IAF and the sample-by-sample approach we are using, are equivalent. To answer this, we first conduct an analysis to see how similar the results are for an epoch based measure, using both the sample-by-sample and the traditional IAF approach. Then, we characterize the sample-by-sample instantaneous frequency extraction accuracy using the PLL.

It must also be noted here that we use the PLL as opposed to the Hilbert transform for real-time implementation, as the later has various disadvantages such as: increased computational complexity; it requires an epoch of data to be accurate, making it sub-optimal for driving latency down; and also incurs edge effects which would either have to be discarded (increased latency) or compensated for by mirroring data at the edges (increased run-time). In contrast, the PLL is optimal for driving latency down as we could operate on one sample at a time whilst avoiding any edge effects.

4.6.2 Methods

For the frequency characterisation, we use the eyes-closed resting state data-set 1 from Table 4.1. From this, we use channel Fp1; as detailed in Section 4.1.2, this would be a good channel for the chronic pain application driver as it contains decent alpha power. Moreover, this being a non-hair region makes it more usable for at-home use.

Sample-by-sample instantaneous frequency vs epoch based

Firstly, we aim to show here that the sample-by-sample instantaneous frequency is equivalent to the traditional epoch based measure. To do this we divide the alpha band pass filtered EEG signal for each subject into 12 s epochs, which is mostly an arbitrary choice but also one taken keeping in mind that 12 s would ensure that enough windows are present for the welch's transform to average over. We used a welch transform (1.3s hamming window with 50% overlap). Hence, with a window size of 1.3s, and the total epoch being of 12 s length, there will be around 18 windows for the welch transform to average the power spectrum over; this will allow us to get a smoother power spectrum for each 12 s epoch, which would not be the case if we had used an FFT over the whole epoch. With this, the dominant frequency in each epoch (the IAF), is calculated as the frequency with the highest power in the epoch.

After this, we obtain an IAF measure over an epoch again, but this time using a PLL. Here, we use the PLL to first get a sample-by-sample instantaneous frequency measure over the 12 s epoch (using the same methodology as in 4.2.2), after which we take the mean of the instantaneous frequency over the 12 s epoch to get the equivalent 'epoch based' measure $f_{PLLEpoch}$. From this we can then calculate the error f_{error} as:

$$f_{error} = f_{Epoch} - f_{PLLEpoch} \tag{4.19}$$

With this, we can then judge whether the epoch averaged sample-by-sample instantaneous frequency calculations obtain with the PLL, is equivalent to the traditional epoch based IAF measures.

Instantaneous frequency accuracy using a PLL

Next, once this equivalence is established, we then characterize accuracy for the PLL-based instantaneous frequency measure, on a sample-by-sample basis. For this, we first band-pass filtered in the Alpha range using an order 2 IIR butterworth filter with 8–13 Hz pass band frequencies. Then, we compared the instantaneous frequency measured by the PLL f_{PLL} with that obtained by the gold standard Hilbert transform $f_{Hilbert}$.

For the f_{PLL} measure, we used the same PLL settings as in Section 4.3, where it was already tuned for this data-set. The instantaneous frequency is obtained by taking the derivative of the instantaneous phase and then smoothing the output using the same low pass filter as in Section 4.2.2, to remove the derivative induced spikes. This then results in the PLL-based instantaneous frequency measure f_{PLL}, which is then compared against the Hilbert-based one $f_{Hilbert}$. The error between both f_{error} is defined as:

$$f_{error} = f_{PLL} - f_{Hilbert} \tag{4.20}$$

4.6.3 Results and Discussion

Sample-by-sample instantaneous frequency vs epoch based

First, we present the equivalence between the sample-by-sample (using a PLL) and the traditional epoch-based calculations, to see how close the former is to the latter. Fig. 4.19 gives a raw trace of the frequency calculated over each epoch using both methods and as observed, the sample-by-sample version gives a close approximate to the traditional epoch based measure; the error between both was -0.10±0.28 Hz, as observed from the histogram in Fig. 4.20. Note that with the welch settings (1.3s window size and sampling frequency of 250 Hz), the FFT resolution is 0.8 Hz (sampling frequency/number of samples). In other words, our error is lower than the frequency resolution of the welch transform. Hence, we can conclude that the sample-by-sample approach gives a close approximate to the traditional epoch based calculations and hence, can use it for the closed-loop platform. Compared to epoch-based methods, a sample-by-sample approach is more desirable as it drives down latency, in addition to allowing finer time-resolutions for closing the loop at.

Instantaneous frequency accuracy using a PLL

Next, we present results for the PLL-based instantaneous frequency measure. As observed from the raw plots in Fig. 4.21, the PLL-based implementation closely follows the Hilbert-

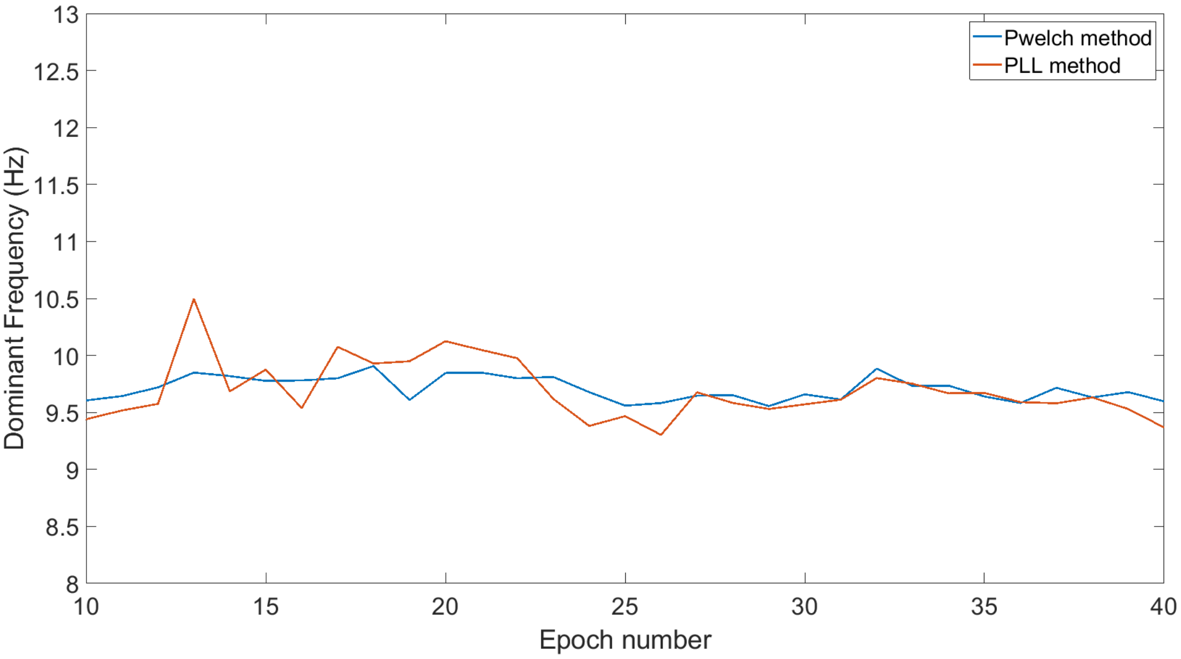

Figure 4.19. Sample-by-sample method vs epoch-based equivalence validation.

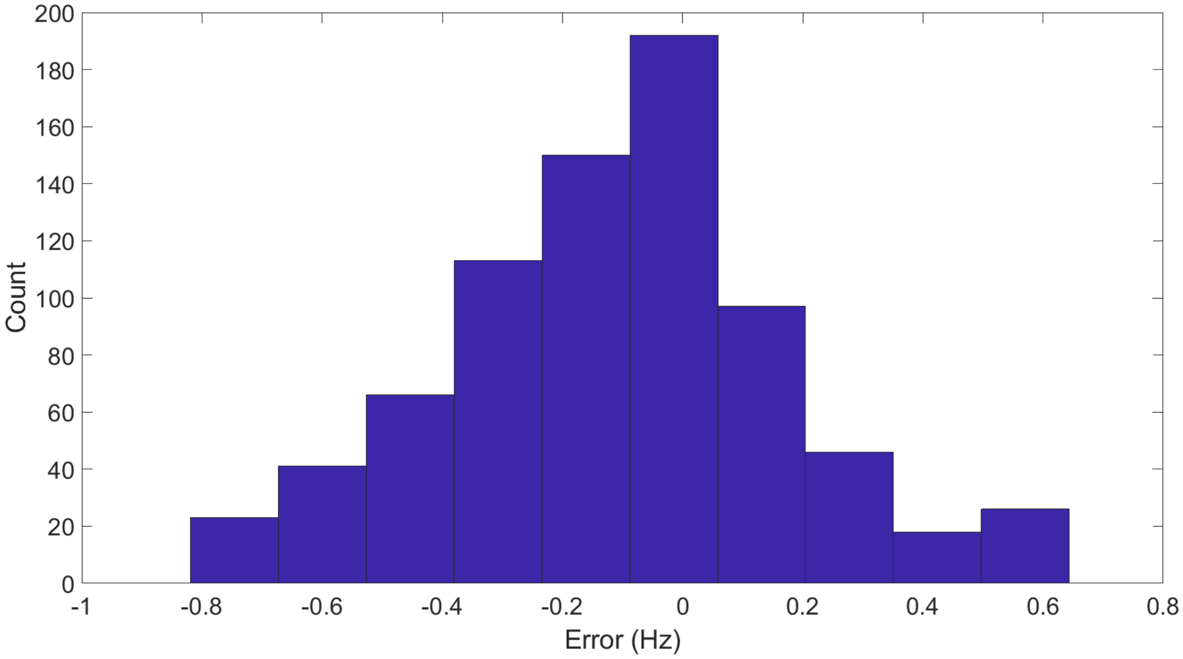

Figure 4.20. Hisogram of error (outliers removed) between the epoch based and sample-by-sample based method of calculating frequency.

based one. However there exists few areas, where the Hilbert-based measure deviates from the PLL based ones, for example time=56s and time=79s in Figure 4.21. This may be due to the noise-sensitive nature of the Hilbert method as it expects a narrow band signal and not a multi-component one, as is the case for EEG. In contrast, the PLL essentially 'fly-wheels' over such noise due to its implicit low pass filter and does not respond as instantaneously as the Hilbert, which makes it less affected by these.

Moreover, it must be noted that these areas then affects the error values obtained, since it is the Hilbert-based method that we use as the benchmark. Any outlier in the benchmark would spill over to the overall error measure and to account for this, we then removed the outliers using Matlab's 'rmoutliers' function which classes as outlier's values that are more

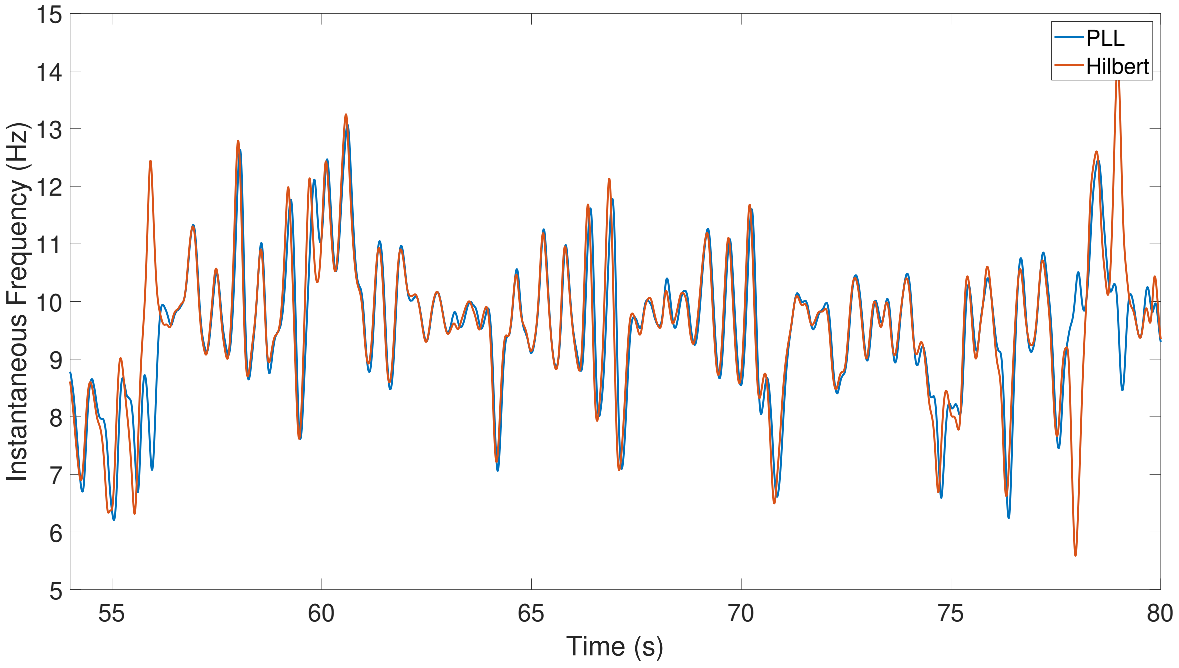

Figure 4.21. Instantaneous frequency obtained using the PLL and the Hilbert method.

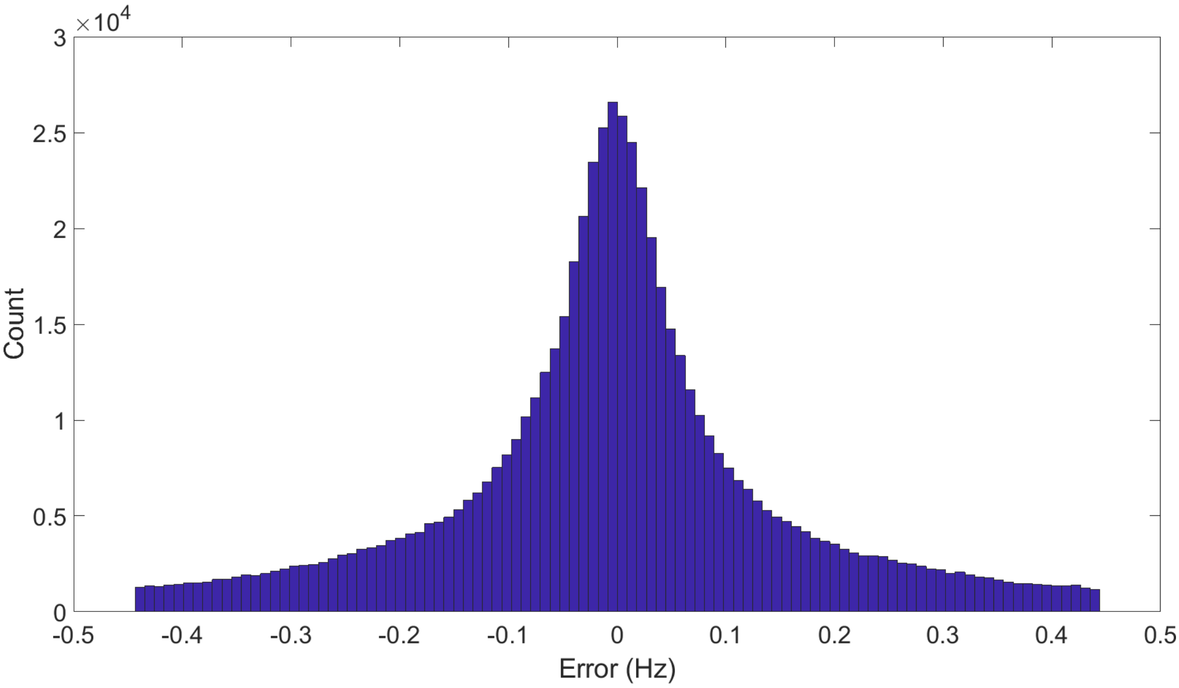

Figure 4.22. Error histogram for PLL based instantaneous frequency calculation (with outliers removed).

than three scaled median absolute deviations away. With this, the frequency matching errors are then 0.08 ±0.79 Hz with outliers and -0.004 ±0.15 Hz sans outliers. These results with outliers removed are shown in 4.22.

With this, we can then conclude that the PLL tracks the instantaneous EEG frequency well, giving a close approximate to the gold-standard Hilbert based method. The key advantage of using the PLL, as opposed to epoch-based methods like the Hilbert transform is that the former allows for sample-by-sample operation without any edge-effects, which makes it ideal for driving latency down. Moreover, if one needs to calculate over an epoch for a different application, we have demonstrated functionality at the finest time-scale (sample-by-sample), and hence, by definition, the same could be extended for an epoch based measure.

4.6.4 Summary

In this Section, we aimed to show accurate extraction of the instantaneous EEG frequency, using a PLL. We first showed that a sample-by-sample measure is equivalent to that of an epoch-based one, when extrapolated over an epoch, which confirmed its validity. Following this, we then characterized the PLLs frequency matching accuracy on a sample-by-sample basis, resulting in a 0.004 ±0.15 Hz error, when compared to the gold-standard. This then confirmed that the PLL could be used to get a good estimate of the instantaneous EEG frequency on a sample-by-sample basis, making it an excellent choice for low-latency closed-loop systems.

4.7 Conclusion

In this Chapter, we conducted offline analyses to characterize the accuracy of the feature extraction part of the closed-loop system, namely frequency and phase extraction using the PLL.

To do this, we first characterized the EEG in Section 4.2 according to its amplitude, frequency and frequency rate of change; the first two feeds into the PLL tuning methodology and the last motivates closing the loop on a sample-by-sample basis. Then, in Section 4.3, we detailed the PLL tuning methodology taking a bottom up approach: first, we used PLL maths and control theory to get a initial starting point for the tuning parameters, and then we fine-tuned this to get optimal accuracy values. Once optimized, we then divided the remaining accuracy characterization into three separate analyses, based on the operation mode: Continuous phase locked stimulation (general use), Discrete phase-locked stimulation (sleep engineering) and continuous frequency matched stimulation (chronic pain).

Firstly, for the continuous phase-locked stimulation in Section 4.4, we characterized phase targeting accuracy across all EEG bands, to show its relevance for a wider range of applications. Here, using the PLL resulted in around 14% percent improvement in accuracy, when compared to the state-of-the-art. Furthermore, we also showed that the inter-channel and inter-subject variance was low, negating any need for channel or subject specific tuning. Here, the contribution in showing that the PLL performs better than the state-of-the-art across all bands.

Secondly, for the discrete phase-locked stimulation mode in Section 4.5), we obtained comparable accuracy to other lab-based platforms, with a phase error of 11.2±11.2 degrees, when targeting the peak of the slow oscillation. Here, the contribution is not in the use of a PLL, but rather in its on-phone implementation, which will be detailed later in Chapter 5.

Finally, for the continuous frequency matched stimulation mode in Section 4.6, we obtained a frequency matching error of 0.004±0.15 Hz using the PLL. Here, there were no state-of-the art platforms to compare to, as ours is the first to provide sample-by-sample frequency matched stimulation for the chronic pain application driver. This was so because we are responding to a clinical need and building technologies that have been requested but not previously available. Hence, here our contribution is in providing this novel platform, with the on-phone implementation in Chapter 5.

With the accuracy characterized for the feedback measures, we can then move onto the next Chapter, where we tie all of these different components together, to characterize the on-phone implementation.

Chapter 5

Closed-loop light and sound based neuromodulation on a phone

5.1 Introduction

In the previous Chapter, we tuned the PLL for phase-frequency extraction. In this Chapter, we port these algorithms onto a phone, which when combined with the stimulation component in Chapter 3 would then gear us towards the complete on-phone implementation. Most closed-loop platforms available today are lab-based ones, and not suited for *in-the-wild* research. Here, we then provide a mobile app based platform for closed-loop audio-visual stimulation. For the chronic pain application driver, we provide the first on-phone platform for continuous frequency matched audio-visual stimulation. On the other hand, for the sleep engineering application, only three portable platforms exist to our knowledge here: two of them run on embedded hardware [12], [13] and the third on a laptop-tablet hybrid [20]. In contrast, we provide the first smartphone-based platform, with the advantage here being the ubiquitous nature of smartphones.

The app operates in either continuous or discrete mode, depending on the application driver, as shown in Figure 5.1. For chronic pain, the app operates in continuous mode, where the audio and visual stimulus waveforms are provided in a continuous fashion, whilst adapting to the frequency of the on-going EEG. On the other hand, for the sleep engineering application, the app operates in discrete mode: here, phase-locked auditory stimulation is provided in a one-off manner, when the target phase is reached.

We first start the Chapter discussing the system design and implementation in Section 5.2, where we characterize the latency and accuracy for the individual components of the system i.e. the sensing and signal processing components. It must be noted here that we do not characterize the stimulation component, as we have done that already in Chapter 3. With individual components designed, we then tie the pieces together and we also introduce a test phantom based setup for validating the technology in a controlled environment using pre-recorded EEG data; this negates the need for on-person tests for technology validation. With this, we then characterize the accuracy and latency of the complete system, starting with the frequency-matched continuous audio-visual stimulation (continuous mode) in Section 5.4 and then the phase-locked auditory stimulation (discrete mode) in Section 5.3.

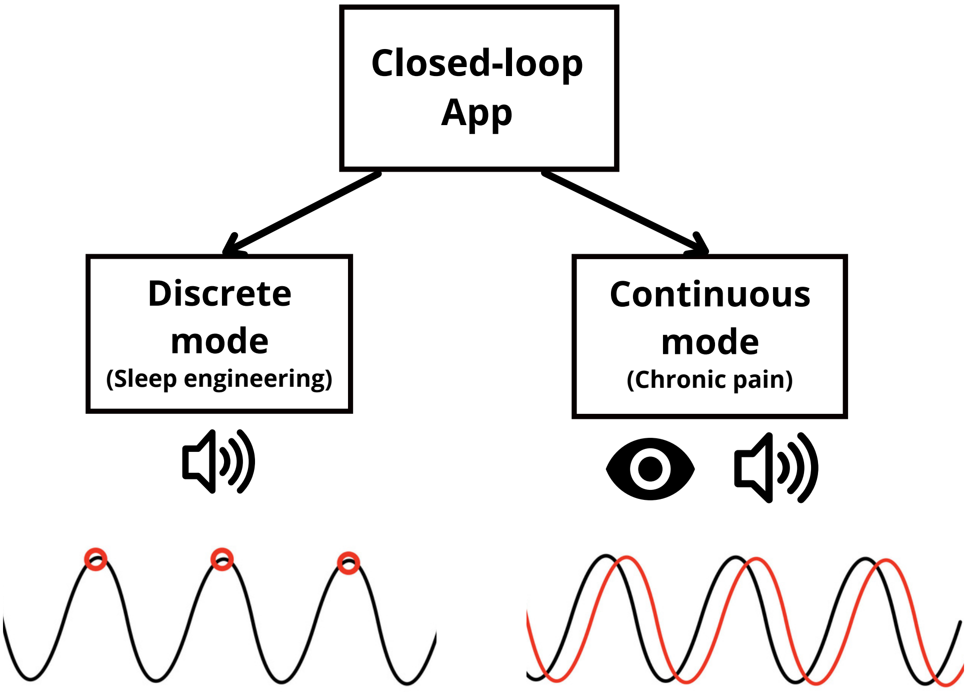

Figure 5.1. Closed loop app overview showing different modes of operation.

These would be designed for the chronic pain and sleep engineering application drivers, respectively.

5.2 System design and implementation

5.2.1 Introduction

The closed-loop platform consists of three parts: EEG wireless sensing, signal processing for phase-frequency extraction, and the audio-visual stimulation, all of which runs on an Android app. In this Section, we start by characterizing these individual components separately, before combining them to form the whole. For sensing, we characterize the wireless communication latency and transmission accuracy to ensure low latency transmission with minimal data loss. For signal processing, we port the PLL tuned in Chapter 4 onto a smartphone and compare the on-phone accuracy and latency, to that of the PC. For the stimulation, we do not need to characterize separately here, as this was already done in Chapter 3.

5.2.2 Methods

Sensing

First, we characterized the wireless transmission accuracy and latency, of the EEG signal. To do this, we used the smarting amplifier by mbraintrain. At the time of doing the work, this was the only medical grade wireless EEG device, that was designed to stream to a phone. As the focus of this PhD was software, this was the most suitable for use rather than making a new system from scratch. An API is available for programming in Android, and we use this to stream EEG data to our phone.

The smarting amplifier is a wireless 24 channel EEG amplifier, which transmits using Bluetooth v2.1 Enhanced Data-Rate (EDR), which is a class 2 bluetooth device (range 10m), optimised for short-range wireless transmission. The signal could be sampled at either 250 Hz or 500 Hz, and each channel has its own 15k samples sized buffer, where the signal is added sample-by-sample as they arrive. The received signal could then by accessed from this buffer one sample at a time without needing the buffer to be full, which is ideal since we would want to operate on a sample, as soon as it arrives to minimize latency.

To characterize the accuracy and latency, we first needed a test setup to stream an EEG signal onto the phone, as shown in Fig. 5.2. To do this, we used pre-recorded EEG data, a NI DAQ data-acquisition system (NI-USB6212) and a Digitimer D171 100 dB attenuator. Here, pre-recorded EEG signals (from data-set 2 in Section 4.1) was streamed from a PC via the DAQ, where the output of the DAQ is the EEG signal represented as an analog voltage (range from 0 to 10V). This signal was then attenuated using the 100 dB attenuator, with an output in the micro-volts range, which mimics EEG voltages, as measured directly from the head. The data is then streamed from the smarting amplifier to a Sony Xperia Z3 smartphone running on Android 6.0.1, using wireless communication. With this, we could simulate an on-person EEG data-collection setup, without having to conduct an on-person test. It must also be noted here that the airplane-mode was turned on, with wifi and location services turned off. Not doing so resulted in a higher transmission latency, which can be anywhere in the range of 100-500 ms.

With this setup, we first measure the transmission accuracy by calculating the root mean squared error (RMSE) between the received and streamed signal. 30 minutes of pre-recorded EEG data (subject 1 of data-set 2 in Table 4.1) was used, taking only the Fpz channel. 3 trials of these 30 minute runs were taken. The similarity i.e. the RMSE error between the streamed and received signal would then help us judge signal transmission quality, and any signs of transmission induced signal attenuation or distortion.

For the latency characterization, the smarting amplifier SDK comes with a time-stamp for each sample; these time-stamps are estimates (made on the phone) that the SDK makes for when the data was acquired by the amplifier. Details of this algorithm were proprietary to the company, so were not shared. Once the sample is received on our app, it was then time-stamped using the ‘SystemClock.elapsedRealtimeNanos()’ method which returns the time since the system was booted, including deep sleep. It uses the on-board real-time clock (not the internet clock) and hence gives more precise time-stamps. It must be noted that the timestamps used by the smarting SDK is synced to this same clock and hence, both are accurate to the nanoseconds and uses the on-board real-time clock in the processor of the phone. The latency was then calculated for each sample of the 30 minute EEG signal recording, for a 3 trial run. The mean of the latency for all samples and its standard devia-

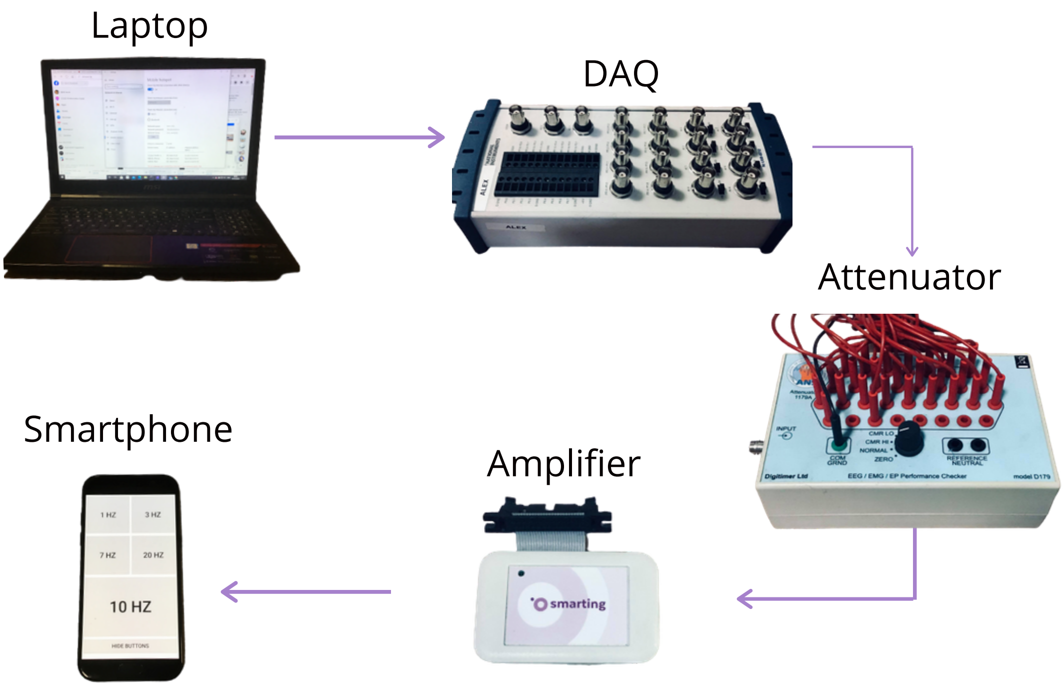

Figure 5.2. Test setup for characterizing transmission accuracy and latency.

tion were then calculated for each of the 3 trials.

Signal processing

For the signal processing component, we validate the on-phone signal processing via porting the PLL designed in Chapter 4 onto a phone. We then compare the on-phone phase-frequency accuracy and run-time latency, to that of the PC. The accuracy is expected to be the same, all else being equal and the latency is expected to be higher for the phone, due to limited computational power. Characterizing this per sample run-time latency is important, as it would help us judge whether a sample-by-sample operation is feasible: for example, a sample is received every 4 ms for the 250 Hz sampling rate and if the signal processing run-time is higher than this, then a sample-by-sample approach would not make sense.

To validate the phase targeting accuracy, we used the sleep data-set (data-set 2 from Table 4.1) and to validate frequency accuracy, we used the eyes-closed resting state data-set 1 from Table 4.1. We used the frontal channels Fp1 and Fpz for the eyes-closed and sleep data-set, respectively. These settings were kept consistent to those used in the offline analysis in Chapter 4, as the purpose of this Section is to validate accuracy of the same system, when ported to a phone.

The signal processing on the phone is coded in Java, and the pipeline remains the same, as those in Chapter 4; more specifically, the alpha PLL in Section 4.3 (chronic pain) and the SO band PLL in Section 4.5 (sleep engineering). The only difference here would be the use of non-causal filtering for all filters in Chapter 4, which is not possible for real-time implementation. Here, the filtering is causal. As expected, this would yield a slightly different error value on the phone and this was done intentionally, to aid comparison and see the effects of non-causal vs causal filtering i.e. we could check if the lack of non-casual filtering would cause a significantly higher error on a phone. Hence, to aid comparison, we present results using both non-causal and causal filtering.

The error metrics remain the same as those used in Chapter 4. For phase, we measure both the error over the entire 5 s epoch and the error at peak, where the error is defined as:

$$\phi_{error} = \phi_{Hilbert} - \phi_{PLL}. \tag{5.1}$$

Here, $\phi_{Hilbert}$ is the instantaneous phase extracted using the gold standard Hilbert transform and ϕ_{PLL} is the instantaneous phase extracted by the PLL. This metric would then give a per sample error values, which we can use for the on-phone vs PC comparison.

Similarly, the frequency error f_{error} is the error between the instantaneous frequency obtained using the PLL f_{PLL} and the one obtained using the gold standard Hilbert transform $f_{Hilbert}$:

$$f_{error} = f_{PLL} - f_{Hilbert}. \tag{5.2}$$

Finally, the per sample computation time is measured both on a PC and the phone, to compare computational cost. The PC used had a Intel Core i7-6700 CPU running at 3.4 GHz, with 16 GB of RAM and the Sony Xperia Z3 phone used has a Qualcomm Snapdragon 801 processor running at 2.6 GHZ, with 3GB RAM. We present these latency for both frequency and phase extraction, and this would help judge whether the run-time latency is good enough i.e. lower than 4ms (250 Hz sampling rate), to enable sample-by-sample operation.

5.2.3 Results and Discussion

Sensing

Firstly, for the transmission accuracy, we observe from the raw plots in Fig. 5.3 that the received and streamed signal match perfectly well. Here, we also see that the spread over three trials is minimal. We also show this quantitatively with a Root Mean Squared Error (RMSE) of 3.24±0.13 μV between the streamed and received signals, for the 3 trial run. These then gives us confidence that the signal is being transmitted accurately, with minimal data-loss or distortion.

Next, we move onto the wireless latency characterization. The latency for the 3 trials are depicted in the raw traces and histogram in Fig. 5.4 and 5.5, respectively. From the raw trace, we see that the latency fluctuates over time and from the histograms, we observe outliers. Both these are expected since the smartphone Operating System is not a real-time

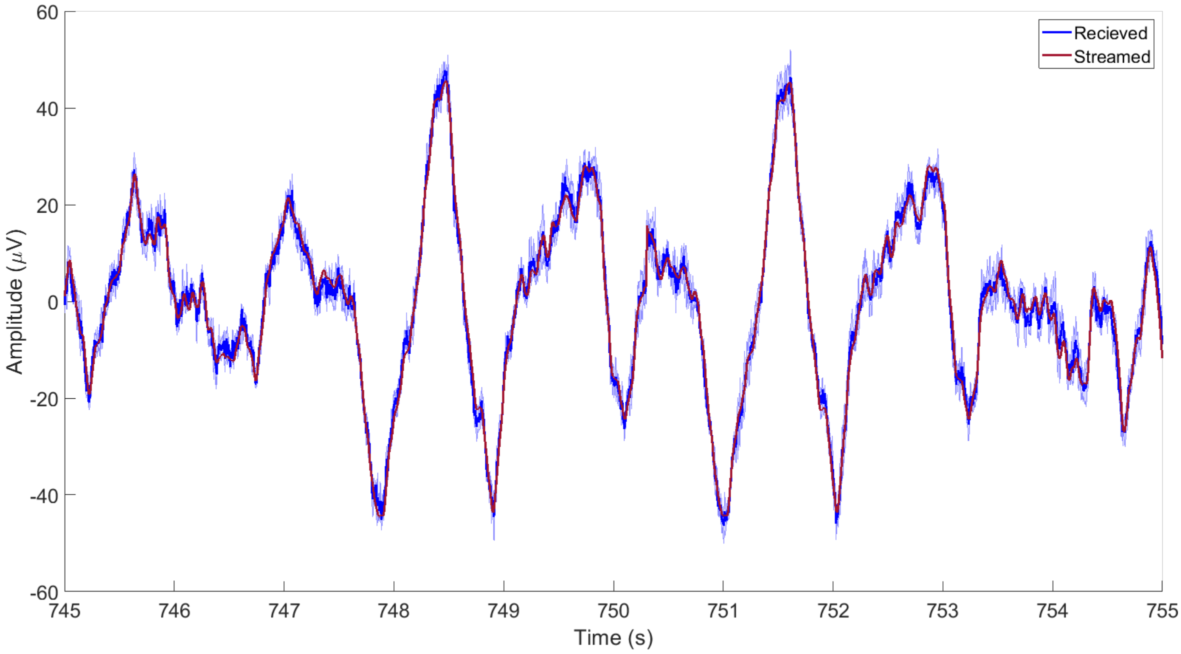

Figure 5.3. Received signal on phone (3 trials) vs streamed signal.For received signal in blue, the solid line is the mean of 3 trials and the shaded line represents the spread over three trials, as measured by the standard deviation.

	Trial 1	**Trial 2**	**Trial 3**
Latency (ms)	53.8 ±23.7	54.8±24.9	61.7±24.1

Table 5.1. Per sample latency for the three trials.

operating system, due to its non-deterministic nature. In other words, the per sample delay could be influenced by factors outside of our control such as various phone background processes. This would then make the latency unpredictable and outside our control.

With this in mind, the measured latency for each trial is summarized in Table 5.1. Firstly, we see that the mean and standard deviation varies across trial, which as we mentioned previously is due to the non-deterministic nature of the phone operating system. Other than that, we observe in general that the latency is in the 50–60ms range. This is a limitation that we would have to work with and future hardware improvements could improve the latency here. To recall, the current amplifier uses a class 2 Bluetooth device (Bluetooth v2.1 EDR), which is an older class introduced in 2007. However, the latest model of the amplifier Smarting Pro uses Bluetooth 5, and claims to have brought down transmission latency by a factor of 5 [99]. Assuming this to be the case, using the Smarting Pro, we could then expect the latency to lower down in 10–12 ms range. However, at the time of this study, the Smarting Pro was not available yet both in the market, and in our lab, and hence, we had to use the older version.

Finally, it must also be noted here that one of the limitations of this study is that we did not characterize performance in a dynamic environment; for example, investigating the effects of incoming calls and notifications. The performance were not characterized under these settings, instead we assumed a static setup. Such a study may be helpful for future work, especially if mapping this technology to other clinical applications where the user is encour-

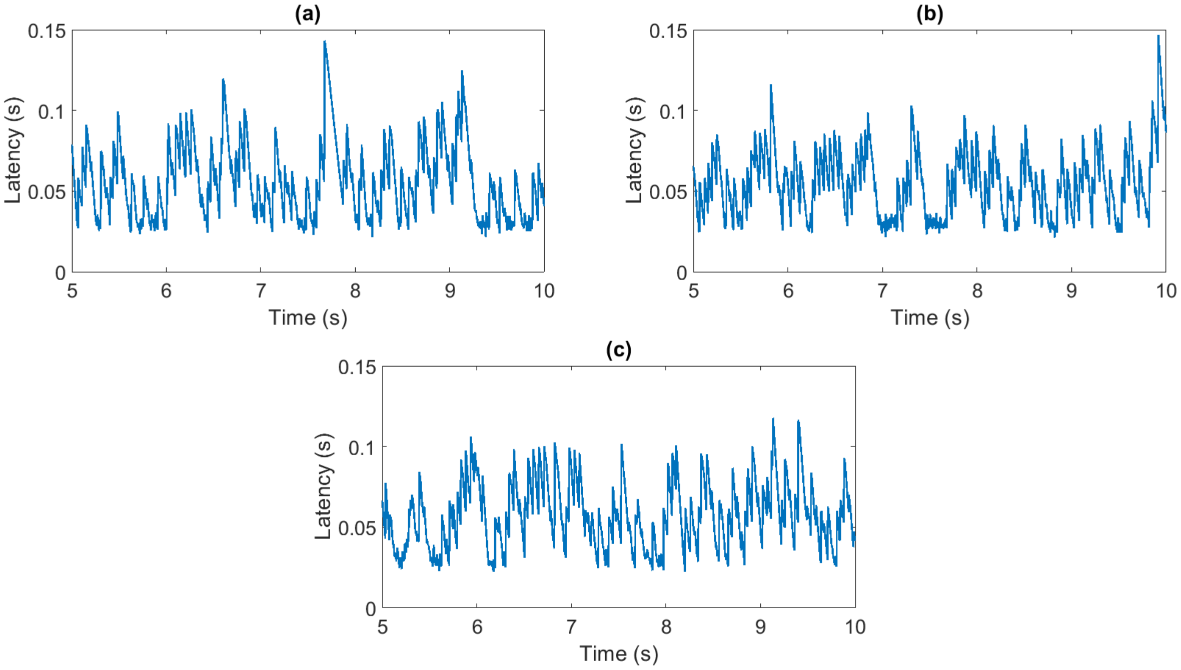

Figure 5.4. Raw plot of communications latency. 3 trials marked (a), (b) and (c).

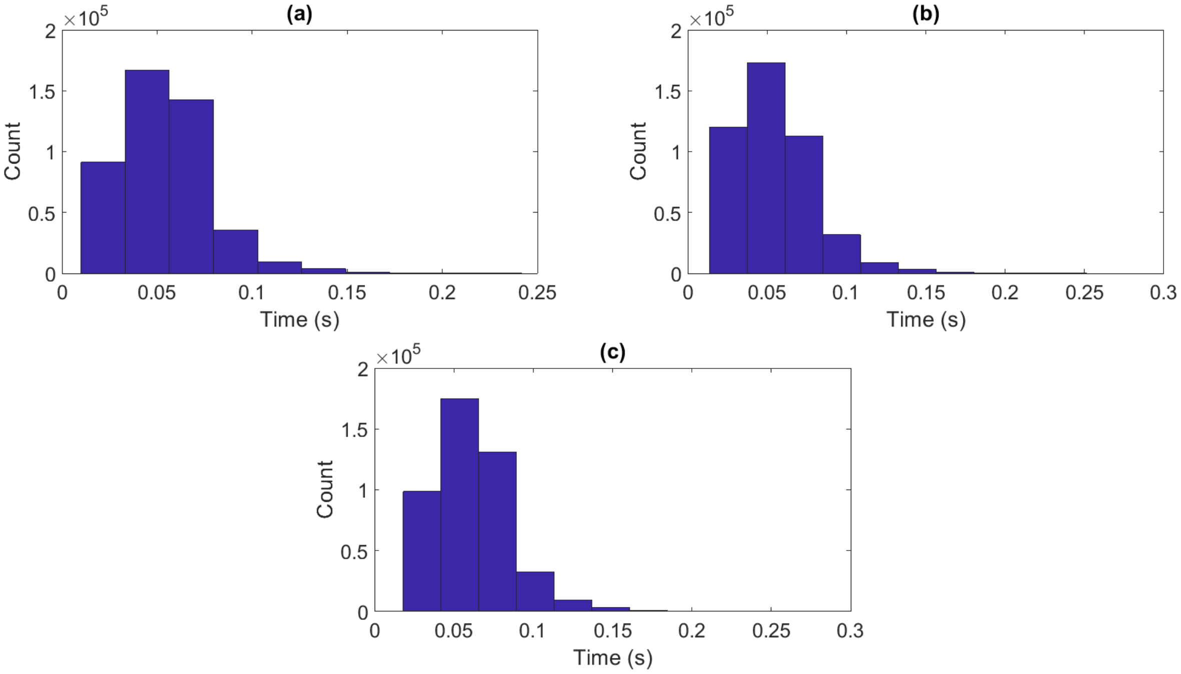

Figure 5.5. Histogram of communications latency. 3 trials marked (a), (b) and (c).

aged to browse the phone during stimulation. However, for chronic pain and sleep engineering, it will be reasonable to ask the users to use the app under silent mode, with minimal background processes and services running. For example, for chronic pain, the user will be experiencing the visual stimuli with a VR headset and the auditory binaural beats without engaging in other activities, as it may interfere with alpha entrainment. Hence, since the user won't be browsing the phone or engaging in other activities during the session, it will be reasonable to ask them to turn notifications off. Similarly, the user will be sleeping for the sleep engineering application, and hence, the same argument applies. For these reasons, we did not find it necessary to characterize performance under load.

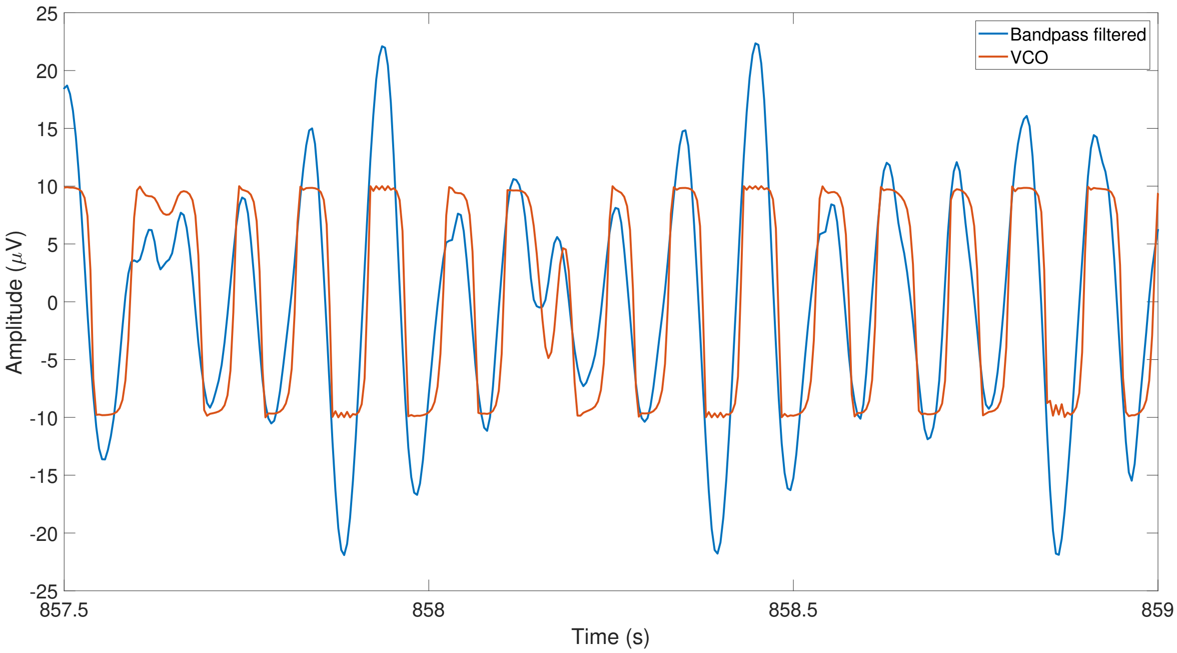

Figure 5.6. VCO output vs band-pass filtered EEG, for the on-phone implementation.

	PC: non-causal filtering	PC: causal filtering	Phone
Frequency error (Hz)	0.004±0.15	0.006 ±0.38	0.006 ±0.38
Phase error peak (degrees)	11.2±112	13.69 ±13.31	13.69 ±13.31
Phase error epoch (degrees)	19.7±25.2	20.48 ±30.25	20.48 ±30.25

Table 5.2. Summary of accuracy metrics for both phase and frequency, done both on PC and phone. Moreover, a comparison is given to the same results obtained using non-causal filtering.

Signal processing

For the signal processing component, we first present results for the frequency tracking accuracy. A sample trace of the PLL output plotted against the input EEG is shown in Fig. 5.6, when implemented on a phone. As expected, the PLL tracks the EEG well and adapts to its time-varying nature. Moreover, the raw frequency output is shown in Fig. 5.7 for both the PC and on-phone implementation. As observed, the difference between the two is indistinguishable, suggesting that the on-phone frequency extraction matches the PC based one. To quantify these, a summary of the error values are shown in Fig. 5.8 and Table 5.2. As evident, the errors are the same (0.006±0.38 Hz) for both the PC and the on-phone implementation, further confirming accurate implementation. Moreover, the effects of non-causal vs causal filtering is evident from the discrepancies in the on-PC error: for non-causal filtering, the error is lower by a factor of 1.5 and this is expected since the zero-phase filtering distorts the signal less, in contrast to the causal filters used in real-time applications. Other than that, we confirm that the on-phone signal processing module is working as expected, matching the PC-based results.

Next, we present results for the phase extraction. The on-phone phase error over the epoch and the peak is summarized in Fig. 5.9 and 5.10, respectively. Moreover, Table 5.2 compares the on-phone errors to the PC based results, again showing exact match. As with the frequency, we also note that non-causal filtering decreased the phase error values whilst the

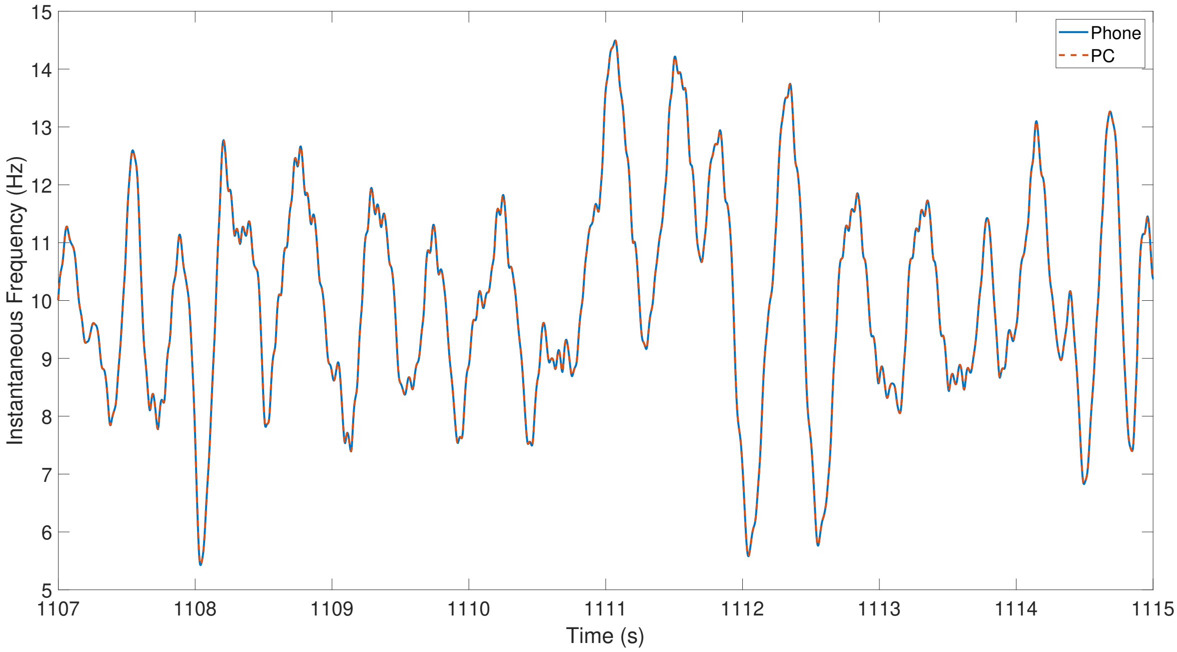

Figure 5.7. Instantaneous frequency raw plot: Phone vs PC.

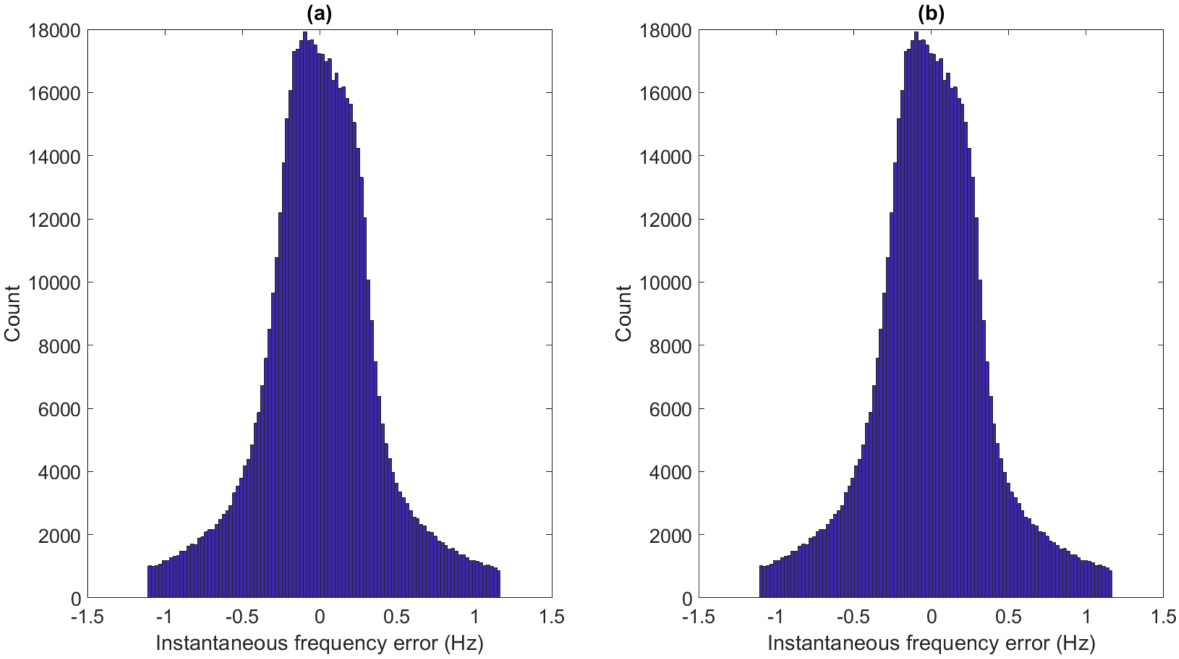

Figure 5.8. Instantaneous frequency error with outliers removed: Phone (a) vs PC (b). The abrupt clipping off towards the end is due to outliers removed.

real-time implementation increased errors. Overall, we can then confirm that the on-phone phase extraction is accurate and working as expected.

Having characterized on-phone phase-frequency accuracy, we now move on to characterizing computation time for both. The results are summarized in Table 5.3. As expected, the PC outperforms the phone for both phase and frequency, with a smaller variance. Nevertheless, the smartphone still offers real-time processing capability: with a 250 Hz sampling rate, a new sample is collected every 4 ms. Consequently, since the run-time latency is in the sub-millisecond range, we can then rest assured that a sample is processed quick enough before the next one arrives. This then allows for sample-by-sample operation, which will aid in driving loop latency down.

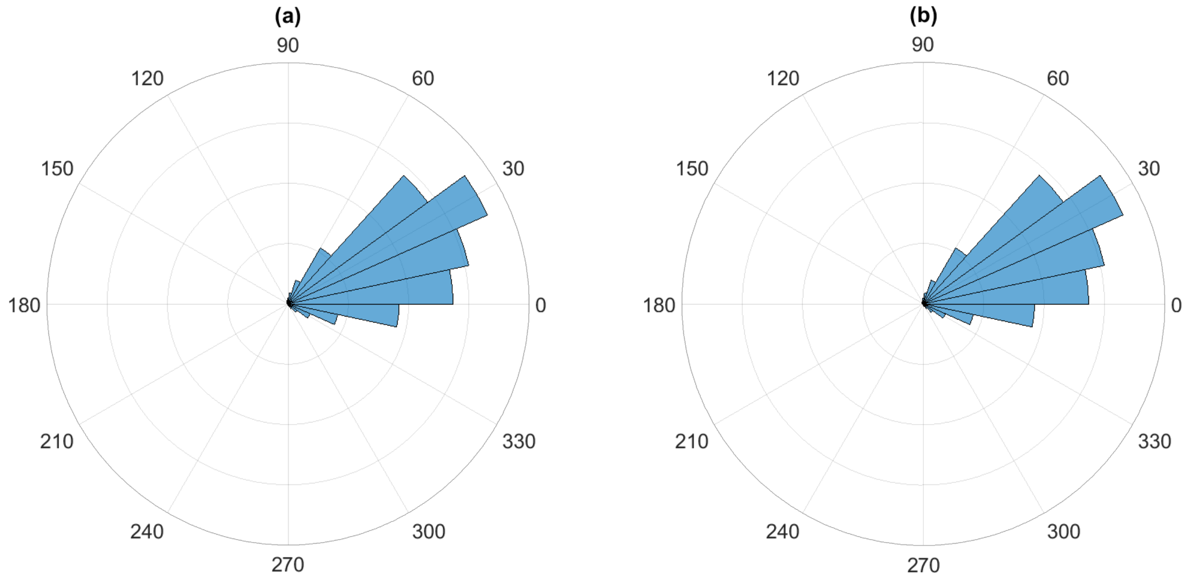

Figure 5.9. Phase error over epoch: Phone (a) vs PC (b).

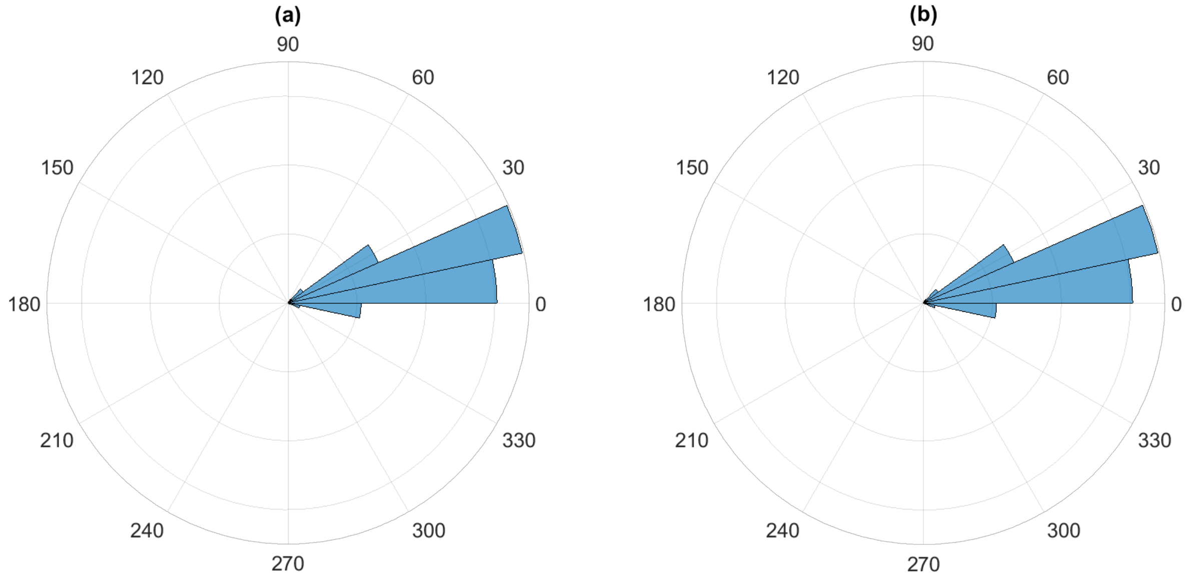

Figure 5.10. Phase error at peak: Phone (a) vs PC (b).

	Phone latency (ms)	**PC latency (ms)**
Phase	0.041 ±0.334	0.001 ±0.165
Frequency	0.037 ±0.220	0.001 ±0.019

Table 5.3. Summary of signal processing latency for both phase and frequency extraction, when done on both a PC and phone.

5.2.4 Summary

In summary, in this section, we characterised the individual components in isolation i.e. the sensing and signal processing components, before combining them to form the complete loop. For this, we first characterized the sensing component: here, we reported accurate transmission, as evidenced by a RMSE of 3.24±0.13μV between the received and transmitted signal and a per sample latency of 62±24 ms. Similarly, for the signal processing component, we showed that the phase-frequency targeting accuracy were exactly the same both on a PC and phone, in addition to a per-sample processing run-time obtained in the sub-millisecond range.

5.3 Test phantom based validation: Discrete mode

5.3.1 Introduction

In this Section, we characterize the performance of the closed-loop system in discrete mode, as depicted in Figure 5.1. This would be a system which can deliver discrete phase-locked auditory stimulation, in the form of short auditory tones, which has applications in the field of sleep engineering. As discussed in Section 2.6.2, lot of recent work has been going into optimizing the accuracy of sleep engineering platforms, mostly done on a PC and restricted to lab-based settings. Here, we provide a smart-phone based platform using the PLL, which would enable 'out-of-the-lab' experiments to be conducted.

5.3.2 Methods

Firstly, to characterize the complete loop in discrete mode (see Fig. 5.1), we used a test-setup akin to that of in Section 5.2, where we simulate a system to represent an on-person testing scenario, but in a more controlled manner. The test-setup remains the same, with the difference here being that the audio stimulation from the phone was also captured by the DAQ to complete the loop; the DAQ was used to record the audio output from the phone, as if it were a pair of headphones. This was done using a 3.5 mm stereo audio to BNC cable. Using this setup, we were then able to simulate the complete loop from sensing to stimulation, as if the data was being collected from the head, using just the pre-recorded EEG data. Data-set 2 from Table 4.1 was used for streaming and the test setup is depicted in Fig. 5.11.

Characterizing latency

With this, firstly, we characterize the complete loop latency using a synthetic signal and then we use real EEG signals to characterize the accuracy.

Firstly, for the loop latency, we streamed a simple 1 Hz (within the 0.5–3 Hz SO band) sine-wave of 1 hour duration, to the phone, where the app was running the PLL in discrete mode and programmed to stimulate at the peak of the sine-wave. The auditory stimulation is a short 10 ms audio pulse (480 Hz sine pulse similar to [2], [5]) and this was then received at the DAQ. Note that the PLL parameters on the phone remained same as in Section 4.12, where we tuned the PLL for the sleep engineering data-set, as this would be the application driver for the discrete mode. It is for this reason also that we chose a 1 Hz synthetic sine wave, since it is close to the dominant frequency in the SO band (0.5–3 Hz). Also, the reason for choosing a synthetic single frequency sinewave and not the EEG for characterizing

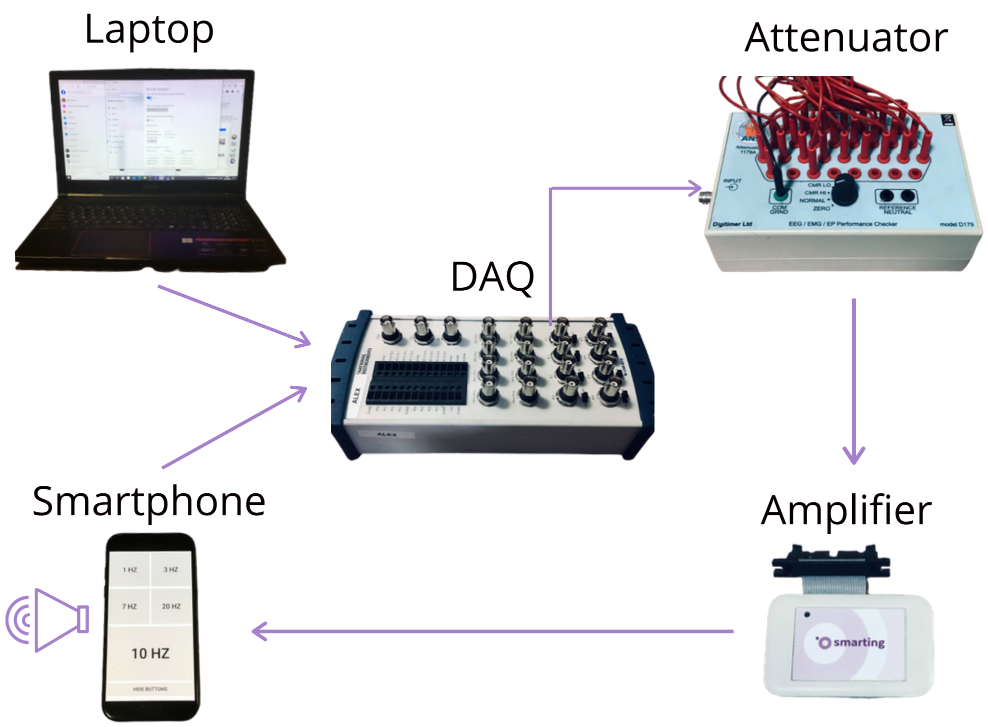

Figure 5.11. Test setup for the audio mode. EEG data streamed from the laptop is converted to an analog voltage, which is then attenuated via the attenuator before passing onto the amplifier. Note that the audio output from the phone is collected at the DAQ's input channel for recording, and is not passed onto the attenuator.

loop latency is that, once both the streamed signal and the auditory stimulation is collected on the DAQ, and the phase error between the peak and the phase at the stimulated point is calculated, we would need the frequency of the signal to calculate latency. Since the EEG is a multi-component signal this is harder to do but on the other hand, with a single frequency sinusoid, we know the frequency and hence, can calculate latency. It is for this reason, that we chose a mono-component synthetic sinusoid with frequency close to the dominant one in the SO band for the latency characterization; moreover, for latency, the waveform used does not have any effect but it does so only for the accuracy.

With the streamed signal and the auditory stimulation waveform then collected at the DAQ, we then analyse latency offline by first getting the phase error between the peak of the sinusoid (target phase) and the stimulaton points (stimulation delivered). With this, we can then get the complete loop latency t_{loop} as:

$$t_{loop} = \frac{\phi_e}{360 \times f} \tag{5.3}$$

where, ϕ_e is the phase error and f is the frequency of the sinusoid, which in our case is 1 Hz.

Characterizing accuracy

To characterize characterize accuracy, we use the same setup as described in Fig. 5.11 but this time instead of streaming a synthetic signal, we stream actual pre-recorded EEG data. We use the sleep data-set here i.e. Data-set 2 from Table 4.1. Just like in the previous Section, we then collect the streamed signal and the auditory stimulation signal at the DAQ, which we then analyse offline, to obtain the phase error.

To calculate the phase error, we band-pass filter the EEG data in the SO band using an IIR butterworth filter with a passband of 0.5–3 Hz, after which we apply the Hilbert transform on the band-pass filtered signal to obtain instantaneous phase measures for the entire recording. We then calculate the phase error ϕ_e as:

$$\phi_e = \phi_{target} - \phi_{stim} \tag{5.4}$$

where, ϕ_{stim} is the EEG phase at the stimulation point, where actual stimulation was delivered and ϕ_{target} the target phase, which in our case is the peak i.e. 0 degrees, as we are using a cosine to generate the stimulation from the PLL.

5.3.3 Results and Discussion

Latency

For the latency characterization using the synthetic signal, we present the raw plots of the sinusoidal waveform, and the auditory stimulation, where the target was to provide stimulation at the peak of the waveform, in Fig. 5.12. As observed, the stimulation is working as expected, as it is observed to be just after the peak, obviously due to inherent delays.

The delays were then measured to be 70±24 ms, as shown in Fig. 5.13. As a comparison, the lab-based platform by Santaostasi et al. running on a PC was shown to give a delay of 70±5 ms [2]. As evident, the mean delays are exactly same and is dominated by the wireless communication in our case (62±24 ms from Table 5.1) which you can't do anything further about while staying on a smartphone platform. In contrast, Santostasi et al. used a wired setup, where a Matlab API communicated with the EEG amplifier via a TCP/IP port. The transmission latency for them was then limited by the data packet delivery of the API (sent every 20 ms). Hence, their communications latency was much lower (20 ms) than ours (62 ms) due to the wired setup.

Moreover, the variance in our case (±24 ms) is higher than the one by Santostasi et al. (±5 ms) and this is again dominated by the wireless communications in our system, which is about ±24 ms (see Table 5.1). Again not so much could be done here as at the time of this study, the amplifier represented the state of the art for wireless transmission onto a smartphone. However, recently, a newer version of the same amplifier (Smarting Pro) has been

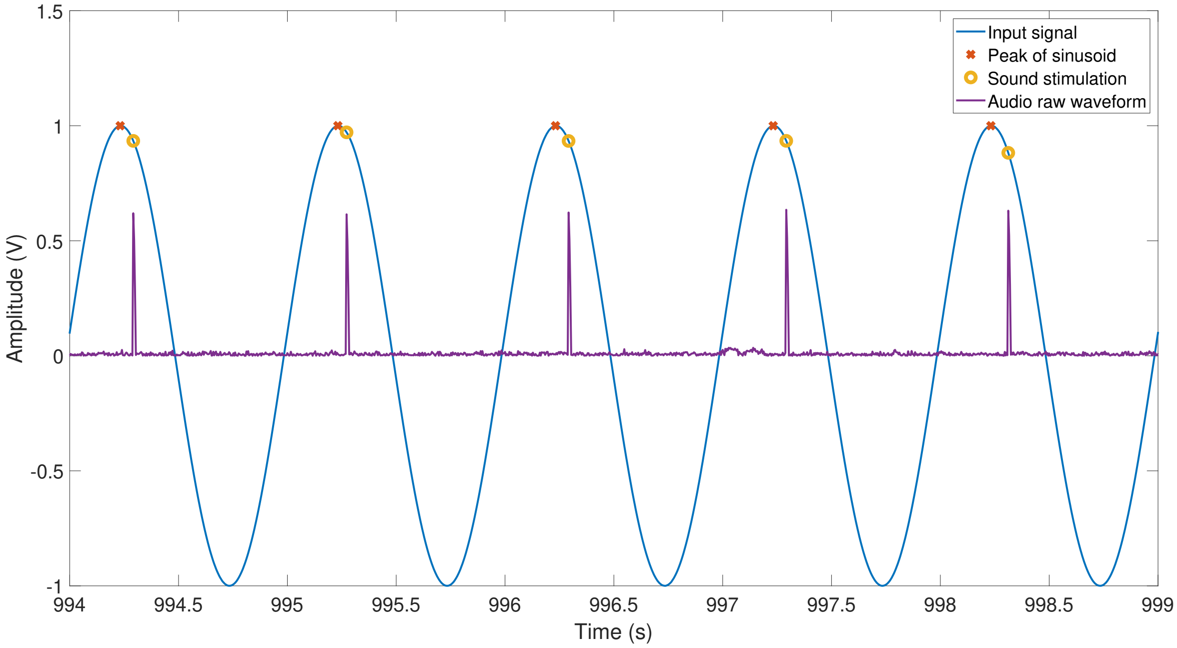

Figure 5.12. Raw waveforms of the streamed signal and the output sound stimulation.

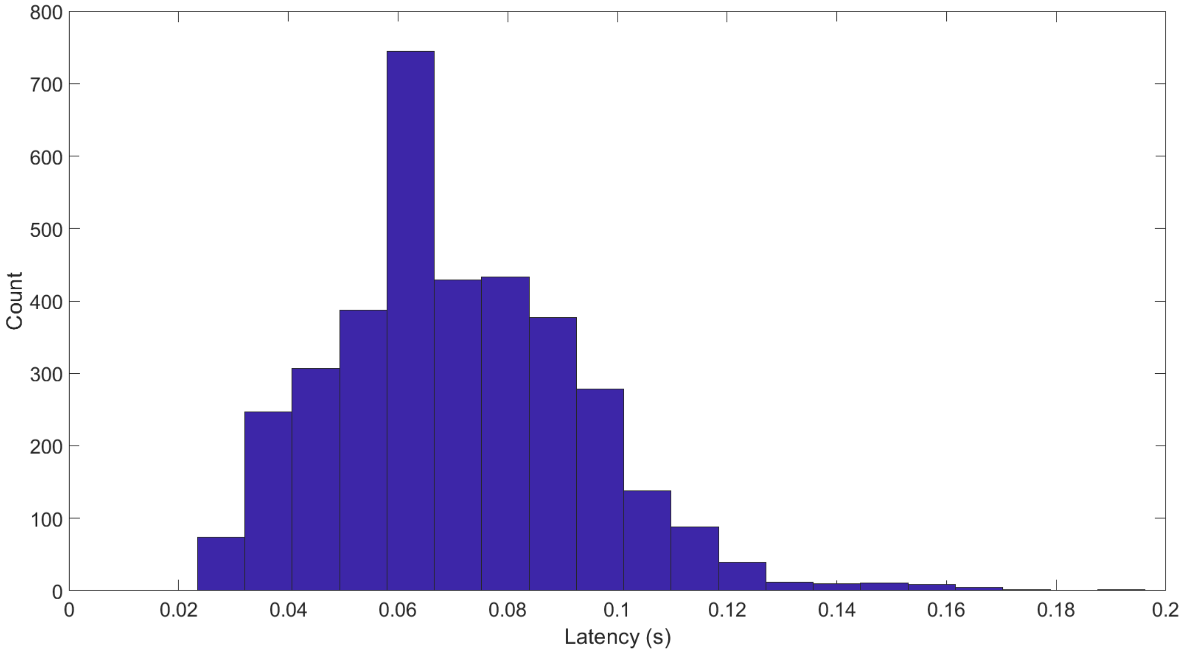

Figure 5.13. Histogram of overall loop latency.

made available, which uses Bluetooth 5 for wireless transmission; they claim it to have reduced the transmission latency by a factor of 5 [99]. Hence, for future work, using this amplifier may reduce the wireless communications latency significantly.

With this delay, one may then ask if it is possible to compensate for it, say by stimulating 70 ms earlier, to compensate for the mean. Since we know the frequency of the sinusoid (1 Hz), we then known that 70 ms for a 1 Hz signal is equivalent 25.2 degrees delay, and then we could change the stimulation control to stimulate at -25.2 degrees (0 degrees target), to compensate for this delay. For a sinusoid with a known frequency, this is possible. However, for EEG, its trickier, as its not a mono-component signal, and hence, we would need to assume a frequency, to make this kind of delay compensation. Although we could assume the frequency to be the dominant frequency in the SO band, and use that to do the

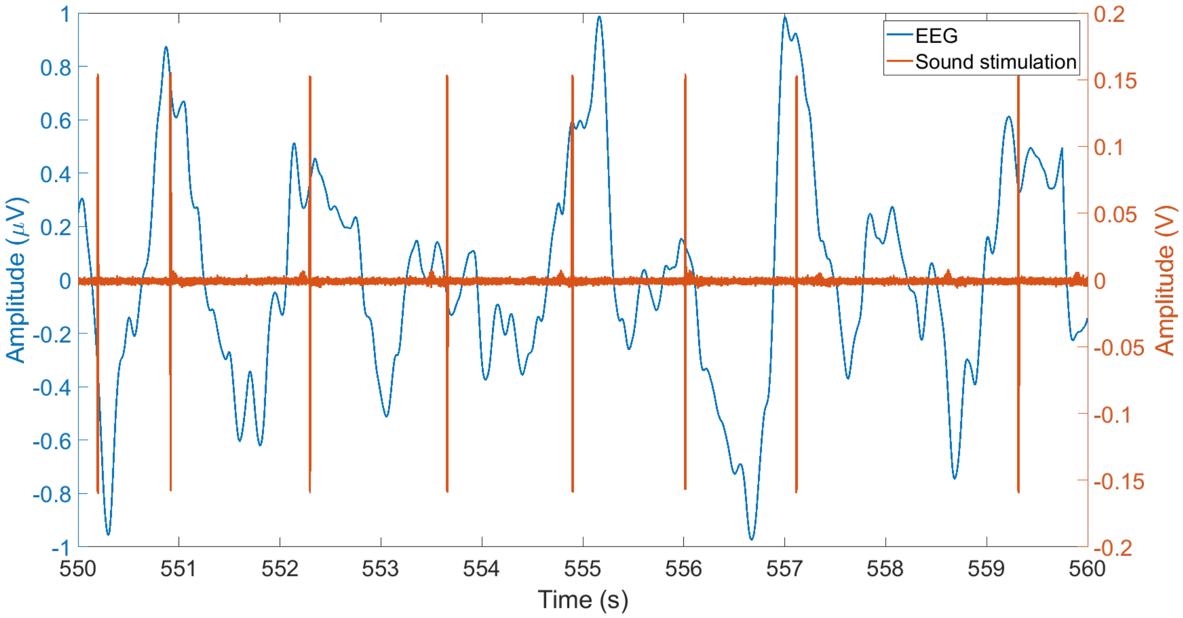

Figure 5.14. Band-pass filtered slow oscillations and the delivered sound stimulation.

delay compensation, we did not take this approach as it still dilutes the result, and may misrepresent the actual peak. Instead, we have provided the delay measurements as it is, and other groups could compensate for it using these assumptions if they wish to, and results are presented here, to bench mark against.

Accuracy

For the accuracy characterization, we first show the recorded EEG signal and the discrete stimulation waveform in Fig. 5.14. As observed, the stimulation is being delivered on target mostly, which is the peak, and the PLL adapts to the time-varying EEG signal. With this, we obtained an overall phase error of 12±60 degrees, as observed from Fig. 5.15.

Our performance is good and is similar to the performance of most other sleep engineering systems currently available, which was previously summarized in Table 2.1 of Section 2.6.2.

Firstly, there are lab-based systems which don't use a PLL. For example, Ngo et al. used a non-adaptive method, where you first detect the start of SO negative wave based on a predetermined threshold and then apply stimulation after a set time-interval to hit the peak [63]. With these, they reported an error of 11±66 degrees when targeting the peak [63]. Similarly, Cox et al. used an FFT based algorithm, resulting in an error of 11±66 degrees and 33±76 degrees in targeting the positive and negative peaks, respectively [15]. In comparison to these, our system which is a mobile based platform achieved an 12±60 degrees error which is better than these lab-based platforms based on non-PLL methods. Our performance is better here, despite running on a phone primarily because of the PLL we used, which we have shown earlier to be superior to the FFT based methods, which assume signal stationarity and does not take into account the time-varying nature of the EEG frequency.

Secondly, there are lab-based platforms which use a PLL and in result have much lower er-

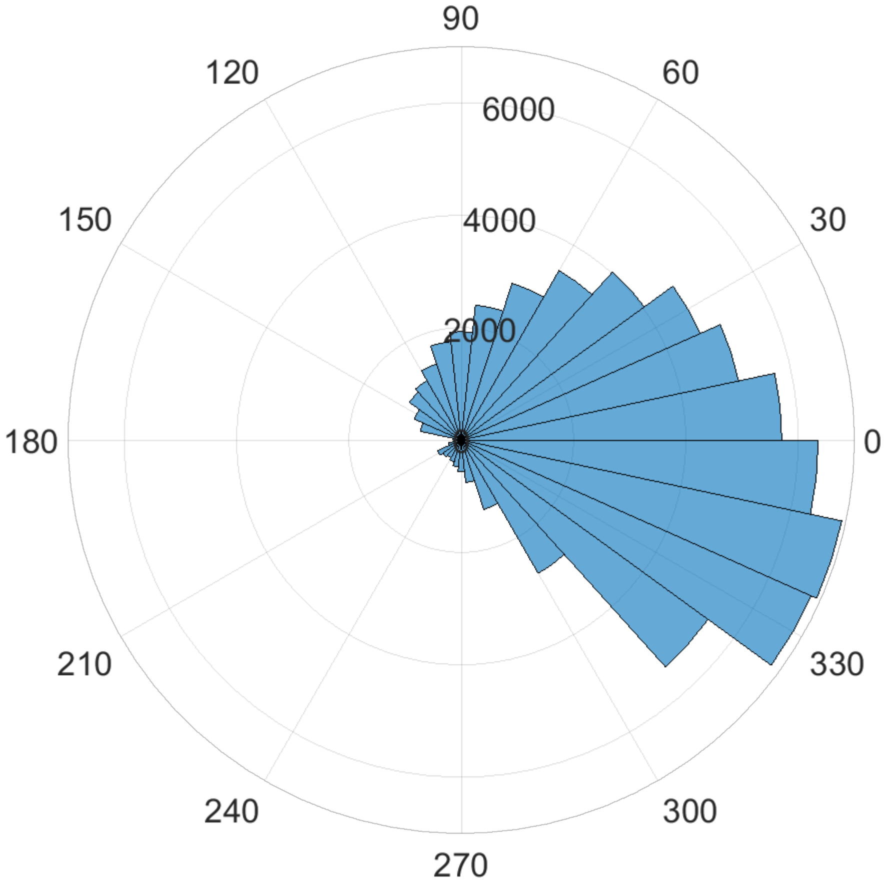

Figure 5.15. Phase error at peak.

ror variance: for example, 0±26 degrees and 6±47 degrees reported by Santostasi et al., and Ong et al., respectively [2], [5]. In contrast, Palambros et al., reported a higher variance of 12±73 degrees, and this is most likely due to their data, where they used older subjects: it is known from literature [2], [4], [5] and from our own analysis we conducted in Section 4.4.4, that the PLLs performance is amplitude dependent, where the performance improves with increasing amplitudes. Hence, since older subjects are known to have lower Slow Oscillation amplitudes, a relatively higher variance in error, is then expected. Then, in general, we note that the PLL is a superior algorithm and in comparison to these lab-based PLL implementations, our mobile based system has a slightly lower performance, and this would be due to the wireless communications latency (62±24 ms) which dominated our total loop latency (70±24 ms), as seen previously. Hence, we are trading off performance for 'out-of-the-lab' use, portability and flexibility.

Finally, we need to compare against other portable mobile sleep engineering platforms. Of these, Debellemeniere et al. used an FFT based algorithm and reported an error of 0±52 degrees, using a commercial grade sleep EEG device which runs on an embedded system with on-board signal processing locked down, that others can't access or modify [12]. This makes it inflexible and unfitting for research purposes. On the contrary, Ferster et al. used a PLL and provides a more research oriented device as an embedded system, with options to modify protocols and settings, and reported an error of 0.4±47 degrees [13]. Comparing the performance of our system against these two, we note that our error of 12±60 degrees is not too far off from these, despite running on a phone. It must be recalled here that both the platforms by [13] and [12] run on custom embedded chips, meaning the delays and latencies could be controlled in a deterministic manner. Moreover, there would not be any wireless communications latency for these embedded systems. This is not the case with our smart-phone based implementation, where the communications latency (62±24 ms) dominated the over-all loop latency of 70±24 ms. The trade-off though is worth it as our platform runs on a ubiquitous mobile platform such as a smart-phone. Finally, in contrast to these embedded paltforms, Leminen et al. used a Surface Pro laptop-tablet hybrid in conjunction with the Enobio wireless EEG headset to provide an ambulatory system. However, they used external speakers placed above the user's head to deliver the sounds and the surface pro laptop-tablet hybrid is not ideal for ambulatory purposes, as it is not as ubiquitous as say a smart-phone. More importantly, they use the same algorithm as Ngo et al., which stimulates based on a fixed time-delay once the negative peak of the SO is detected, to reach the peak (the time-delay being tuned on a subject-by-subject basis offline). This non-adaptive nature of the algorithm again explains the higher error they reported of –18±67 degrees. In contrast to this, we achieved a lower phase error of 12±60 degrees and this would be due to the adaptive nature of the PLL. Moreover, ours is a smart-phone based implementation, which is more ubiquitous than laptop-tablet hybrids, which not all people would have.

It must also be noted that we may be able to compensate for the mean phase delay in a predetermined fashion, by stimulating say 12 degrees earlier but here we would have to make the assumption that the EEG is a mono-component signal, whereas in reality, it is multi-component, and even in the SO band, the frequency may vary anywhere between 0.5 and 3 Hz. Hence, we did not compensate for this mean delay but leave it open to anyone who may wish to do so, as they could use the dominant frequency value to predict the signal forward.

Finally, we also note here that despite not having done any clinical tests to see if our system enhances slow oscillations, we could reasonably assume that it is fit for purpose. This is because our phase-targeting performance is similar to portable platforms like [12] and better than ones like [20] and [3]; all these conducted clinical studies in addition to technology development, and showed that their platforms were able to enhance slow oscillations, and achieve the intended clinical effect. Hence, we asusme our system would also be able to do the same, given the similar performance.

With these, overall, our contribution here then is in a Phase Locked Loop based platform for phase locked auditory stimulation, and we differentiate by providing the first on-phone implementation, through a dedicated Android App, making it more accessible for the wider population.

5.3.4 Summary

In this Section, for the discrete mode, we have provided a closed-loop system which provides phase-locked auditory stimulation, targeting the sleep engineering application driver. We characterized the platform by measuring both the over-all loop latency using a synthetic mono-component signal, and then the phase-targeting accuracy on actual EEG signals. As a result, we obtained an over-all loop latency of 70±24 ms and a phase error 12±60 degrees, both of which are comparable to existing platforms available in literature [2], [4], [5], [12], [13], [15], [20], [63]. Our contribution here is in providing the first PLL-based on-phone implementation in the form of an Android app, which achieves similar performance to the state-of-the-art portable platforms. This then would be more suitable for out-of-lab use-cases, with the ubiquitous nature of smart-phones.

5.4 Test phantom based validation: Continuous mode

5.4.1 Introduction

In this Section, we focus on the continuous mode of operation: frequency matched audio-visual stimulation for the chronic pain application driver, as shown in Figure 5.1. The phone screen flicker and binaural beats would be used for the visual and auditory stimuli respectively, and both these would be matched to the alpha frequency.

It must also be noted that here we only characterize loop accuracy and not latency. For the chronic pain application driver, there is only one metric of interest which is accuracy. This is the difference between the current IAF and the current stimulation frequency. Increased latency might mean the IAF is correctly followed, but with the lag. For the purposes here, this is still an error in the wanted stimulation frequency and so is accounted for in the accuracy metric.

5.4.2 Methods

For characterizing this mode, we used data-set 1 from Table 4.1 since it is an eyes-closed resting state data-set, with high alpha power. Hence, for both audio and visual modes, we use the alpha band PLL in Table 4.8.

Visual stimulation

For the visual stimulation, we match the frequency of the stimulus (sine-method from Section 3.2.6) to that of the on-going EEG. The stimulus implementation details are explained in depth in Chapter 3, and there we had concluded that a sinusoidal waveform would be optimal for the closed-loop mode, keeping in mind the 60 Hz phone refresh rate constraint.

The test-setup used is same as that of Section 5.3 but this time the visual stimulus was measured using a PDA10CS-InGaAs photo-detector, which is an integrated chip containing a photo-diode, operational amplifier, and associated circuitry to provide accurate light intensity measurements. The complete test-setup is depicted in Fig. 5.16.

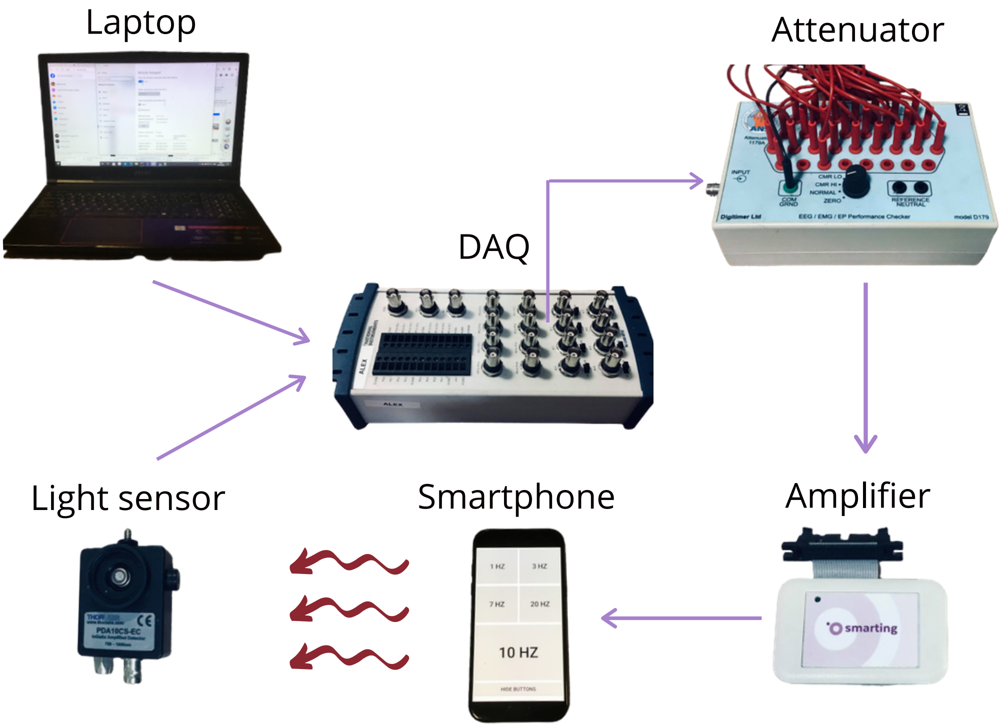

Figure 5.16. Test setup for visual stimulus. Note that the phone flicker output measured via the photodiode, is collected at the input channel of the DAQ. It is not passed onto the attenuator.

We then collect the flicker waveform, alongside the streamed EEG signal at the DAQ. From this, we then obtain instantaneous frequency for both using same methods as in Section 3.2 and 4.6: this involves first band-pass filtering the signals (Order 2 IIR bandpass filter with 8–13 Hz pass-band for the EEG); then we used the Hilbert transform to get instantaneous phase and the frequency as its derivative; finally, we used a low pass filter (Order 2 IIR Butterworth low pass filter with a 2.5 Hz cut-off frequency) to smooth out the derivative induced spikes. It must be noted here that for the band-pass filtering of the visual flicker, we used an order 8 instead of order 2 filter; the reason for this is noise reduction in the flicker waveforms obtained via the photo-diode, which could affect the instantaneous frequency measures of the flicker, due to the derivative function which is noise sensitive. Higher order filters will attenuate out-of-band components more, at the cost of time-domain distortions. Such higher order filters were then not used for the on-phone implementation precisely because of the delays they would incur. These could be compensated offline using forward-backward filtering but not online where no future data is available. With these, the frequency matching errorf_e is then defined as:

$$f_e = f_{eeg} - f_{stim} \tag{5.5}$$

where, f_{eeg} and f_{stim} is the instantaneous frequencies of the EEG and stimulation waveforms, respectively.

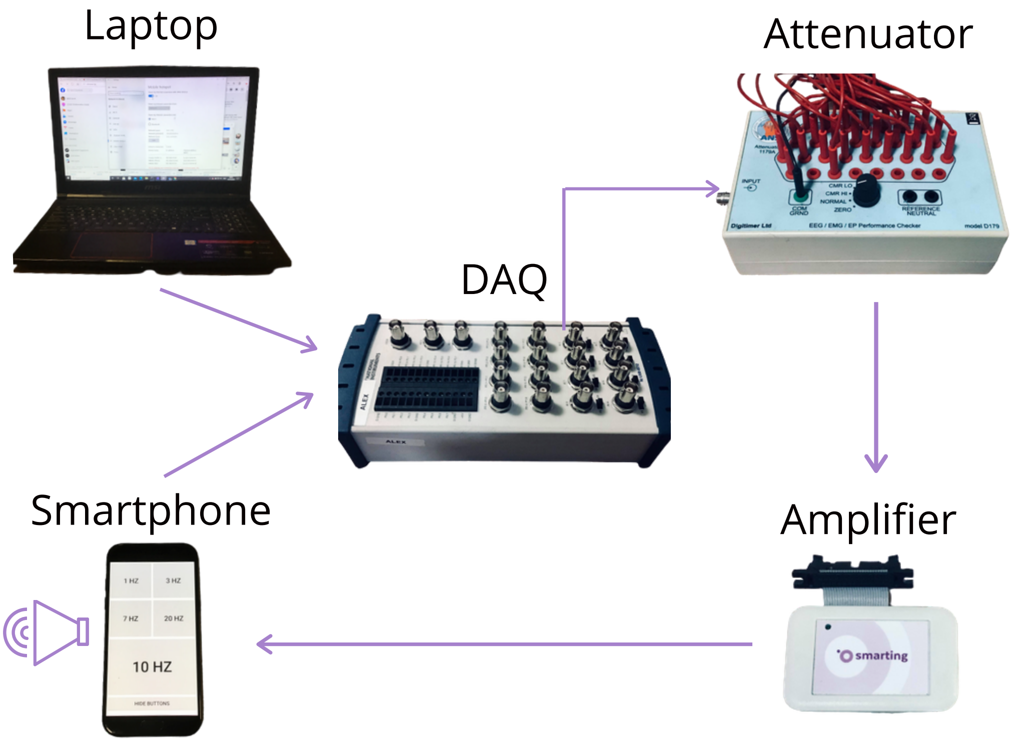

Figure 5.17. Test setup for auditory mode. Note that the phone audio output is measured at the input channel of the DAQ. It is not passed onto the attenuator.

Auditory stimulation

Next, for the auditory mode, binaural beats was used (see Section 3.3). The test-setup remains the same but this time we replace the photo-diode with an audio jack, to measure the beats. This setup is shown in Fig. 5.17.

We then analyse these offline, by first calculating the beat frequency f_b as:

$$f_b = f_{ch1} - f_{ch2} \tag{5.6}$$

where, f_{ch1} and f_{ch2} are the instantaneous frequencies of the audio waveforms from channel 1 (left) and 2 (right), respectively. The instantaneous frequency was calculated just like in the visual case for both EEG and the stimulus waveform. For the latter, the only difference here is that we did not band-pass filter the binaural beats, since the measurement was already clean and mono-component, as expected of a measurement using an audio jack cable.

We then characterize the accuracy via the frequency error f_{err}:

$$f_{err} = f_b - f_{eeg} \tag{5.7}$$

where, f_b and f_{eeg}, are the instantaneous frequencies of the binaural beat and EEG, respectively.

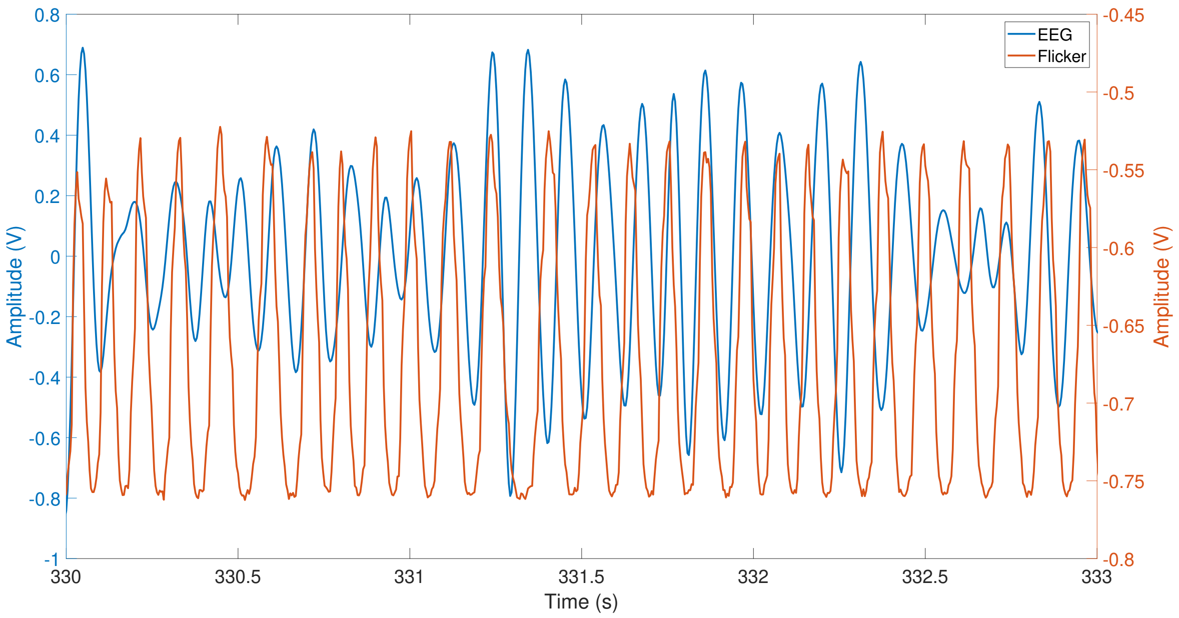

Figure 5.18. Flicker waveform plotted against the band-pass filtered EEG.

5.4.3 Results and Discussion

Visual

Firstly, for the visual mode, we see the flicker waveform and bandpass filtered EEG in Fig. 5.18. It is evident here that the stimulus waveform is not phase-locked to the on-going EEG and this is expected, since the application driver requires frequency matching and not phase-locking.

With this, we obtained a frequency matching error of –0.17±1.29 Hz, with an error histogram shown in Fig. 5.20. We also see this match visually in Fig. 5.19 where we see the flicker frequency matching the EEG frequency, despite being slightly delayed. This delay is expected, as the system has an overall loop latency. Aside from this, the two filters used in the signal processing pipeline (both order 2 filters) would inherently shift the signal. With all these combined, a shift and delay in the flicker frequency is expected and we argue these to be the main cause of the error obtained, and not necessarily the PLLs tracking capability. To demonstrate this, in the offline analysis in Section 4.6, there were no communications latency or delays due to filtering (zero-phase filtering used) and there, we obtained much lower error of 0.004 ±0.15 Hz. This then shows again that the major contributor to the error is not the PLL, but the system induced latency and lack of zero-phase filtering, all of which are constraints of a real-time system.

The question then remains, as to whether this error could be lowered and if this accuracy is good enough. Firstly, regarding improvements, we have done our best to reduce delays: for example, we used the lowest order filters (order 2) to ensure minimal filter induced delay; the PLL was tuned offline to achieve best possible accuracy (0.004 ±0.15 Hz error); and we process each sample as soon as it arrives without any buffer, avoiding any buffering induced delay. All in all, apart from optimizing even more on the tuning parameters, there is noth-

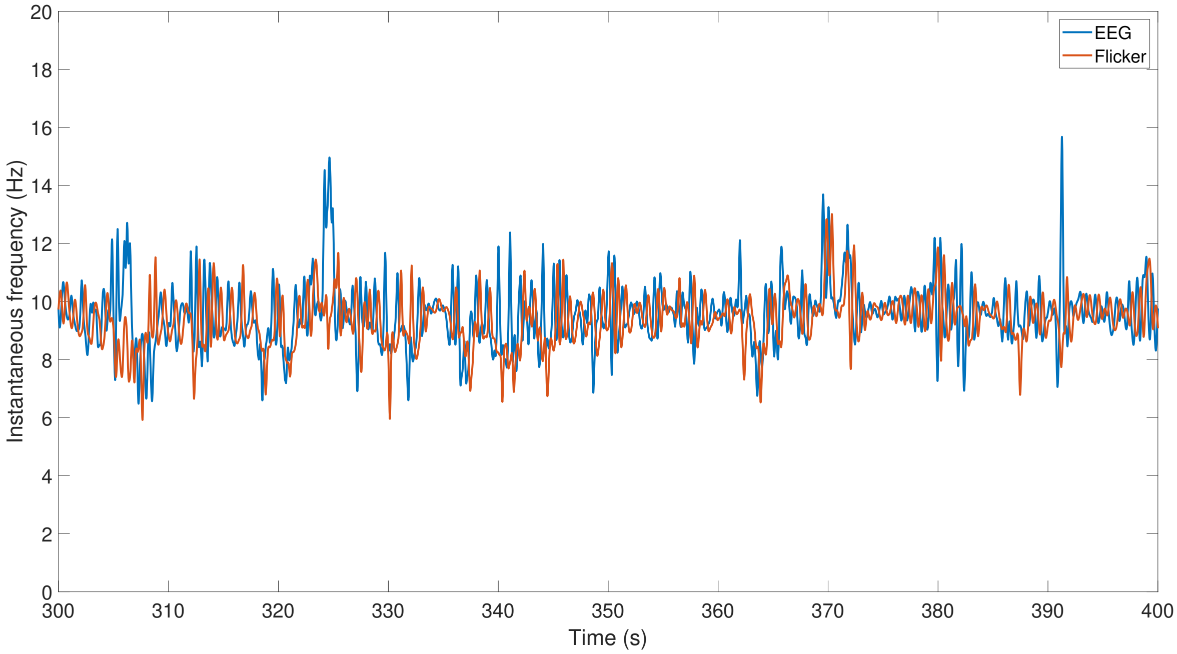

Figure 5.19. Instantaneous frequency of the flicker and the EEG.

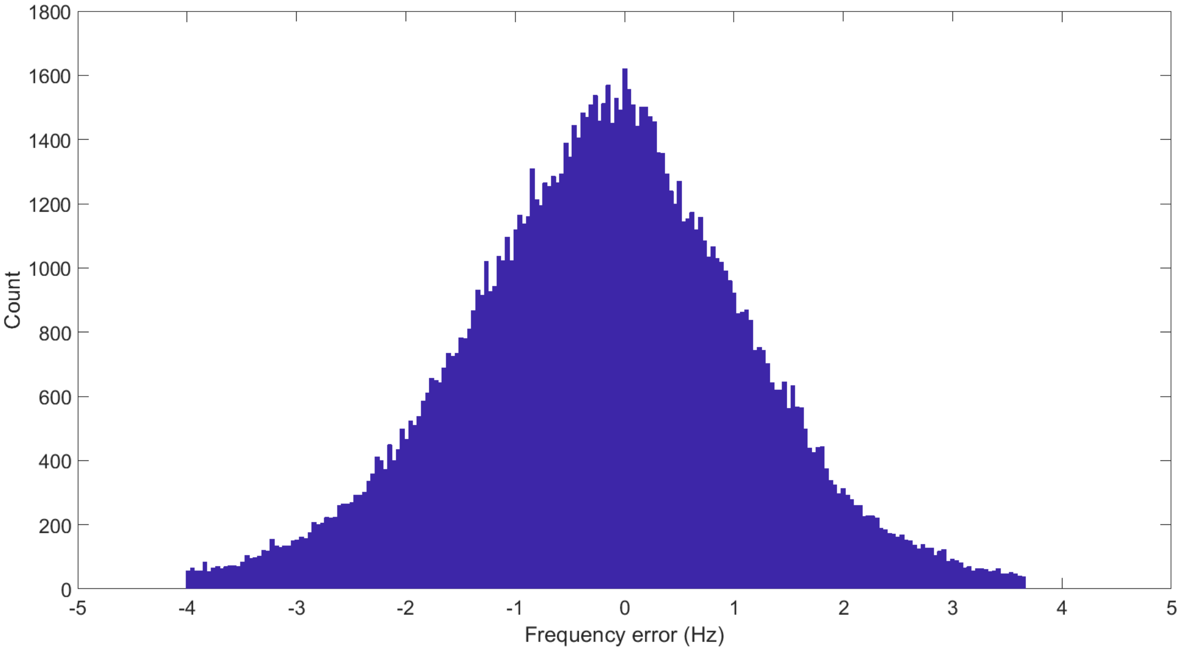

Figure 5.20. Instantaneous frequency error. Clips towards the end because outliers removed.

ing that could be done to improve performance, whilst maintaining a wireless Bluetooth link for compatibility with a smartphone. As observed in Section 5.2 and 5.3, the wireless communications latency (62 ms) dominated the overall loop latency (70 ms). It must be noted here then that using the latest Smarting Pro amplifier which uses Bluetooth 5, is expected to drive down latency by a factor of 5 [99]. This may help improve the frequency matching error obtained here.

We note here that this is now the state of the art for others to compare against, as no similar work exists (details in Section 2.6.3). Hence, we also cannot judge as to whether the performance is good enough to achieve clinical efficacy, as no similar works exist. However, with regards to our -0.17±1.29 Hz error, we do note that this is lower than the resolution

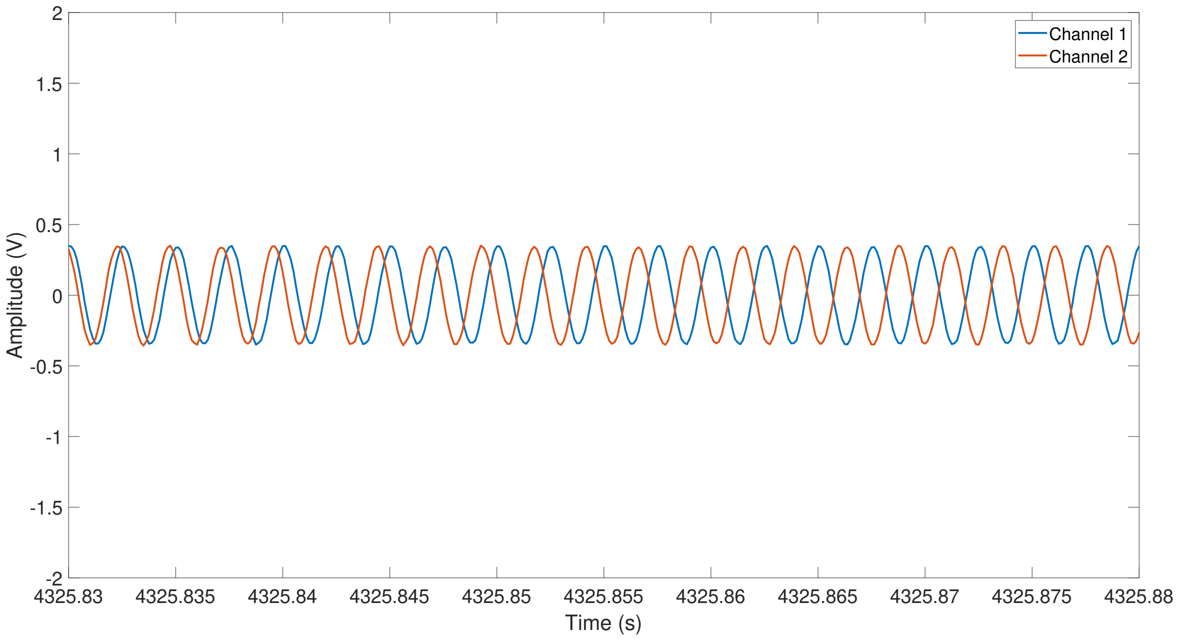

Figure 5.21. Raw binaural waveforms plotted for each channel audio. EEG not overlayed as its frequency too low compared to binaural (400+ Hz), making it hard to visualize both together.

of classical frequency measuring tools. For example, the FFT based methods which most neuroscientists use has a frequency resolution f_{res} defined as:

$$f_{res} = \frac{f_s}{N} \tag{5.8}$$

where, f_s is the sampling frequency and N the window size. This means that the smaller the window size, the poorer the frequency resolution would be. For example, in our case of a 250 Hz sample rate and taking the worse case in our error i.e. a 1.5 Hz error, this corresponds to a window size of 167 samples or 0.67s. Any smaller window size would result in a frequency resolution higher than the error we obtained. Moreover, in our case, we update on a sample by sample basis and each update takes around 70 ms (see Section 5.3); for such small time windows, the frequency resolution would be much higher than the error we obtained. More importantly, in the absence of any state-of-the-art to compare to, we consider our results here the state-of-the-art for future work to compare to.

Audio

For the auditory mode, we see the raw audio waveforms in Fig. 5.21 and its instantaneous frequency in Fig. 5.23. As observed from Figure 5.23, channel 1 frequency is constant at 400 Hz, while channel 2 frequency tracks the frequency of the on-going EEG, oscillating from a centre frequency of 410 Hz. The difference in both is then the beat frequency, which tracks the alpha rhythm of the on-going EEG, as shown in Fig. 5.22. Note here the slight delay in the beat frequency is again due to the on-phone filtering and loop latency, which we would expect as explained in the previous Section.

The performance of the auditory mode is similar to that of the visual mode, with an error of

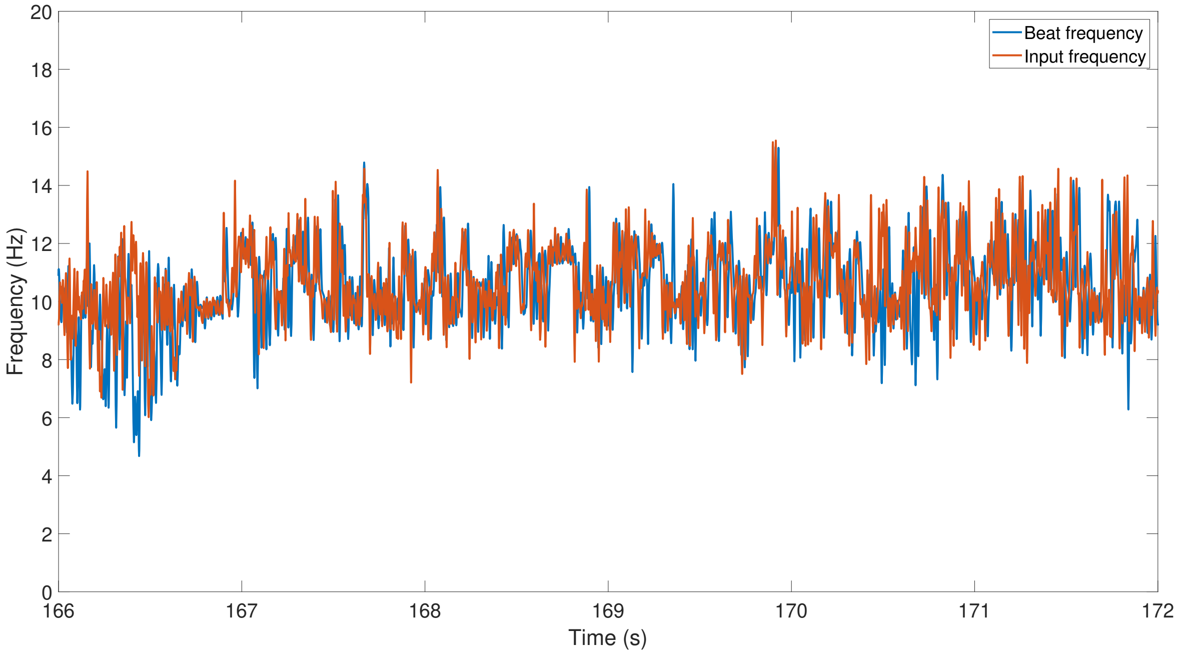

Figure 5.22. Instantaneous frequency of the binaural beat and the EEG.

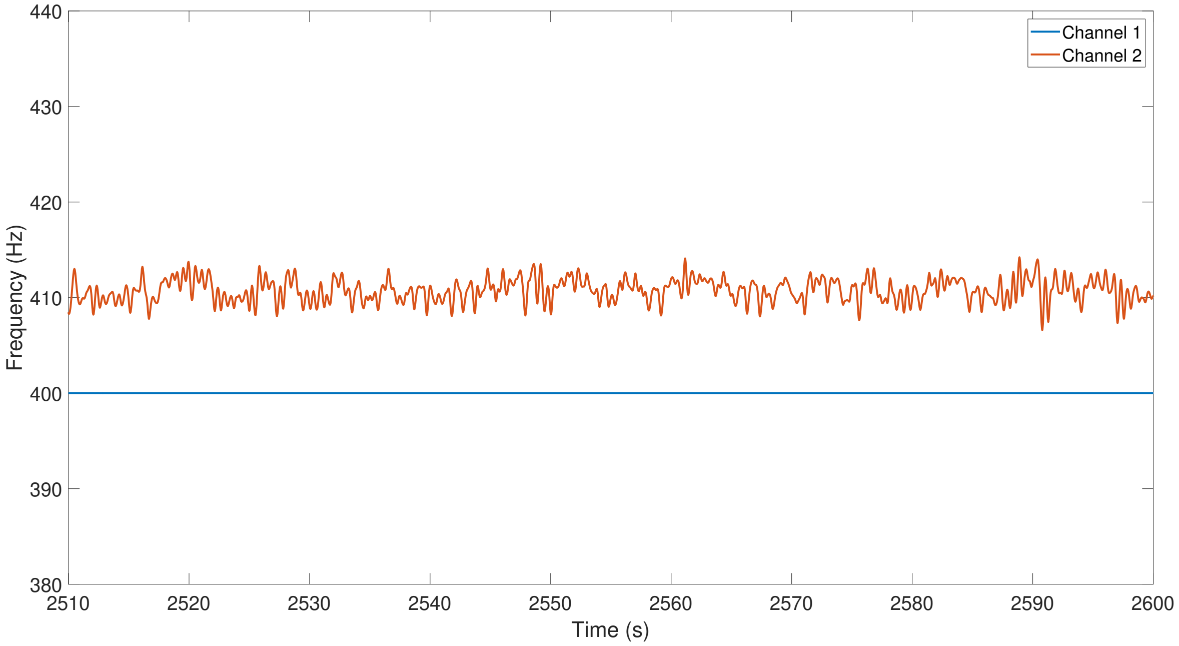

Figure 5.23. Instantaneous frequency of each of the audio channels.

0.20±1.30 Hz, as depicted in Fig. 5.24. Again, no state-of-the-art platforms exists to compare to and we provide the first-in kind, for future platforms to compare against. Also, as is the case for the visual mode, the error could potentially be reduced even further using the latest Smarting Pro amplifier, which is advertised to have 5 times lower latency [99]. Given that the wireless communications latency dominated the overall loop latency, this could potentially reduce the frequency matching error even further.

5.4.4 Summary

In this Section, we characterized performance for the frequency matched mode, for both audio and visual stimulation. A frequency matching error of -0.17±1.29 Hz and 0.20 ±1.3

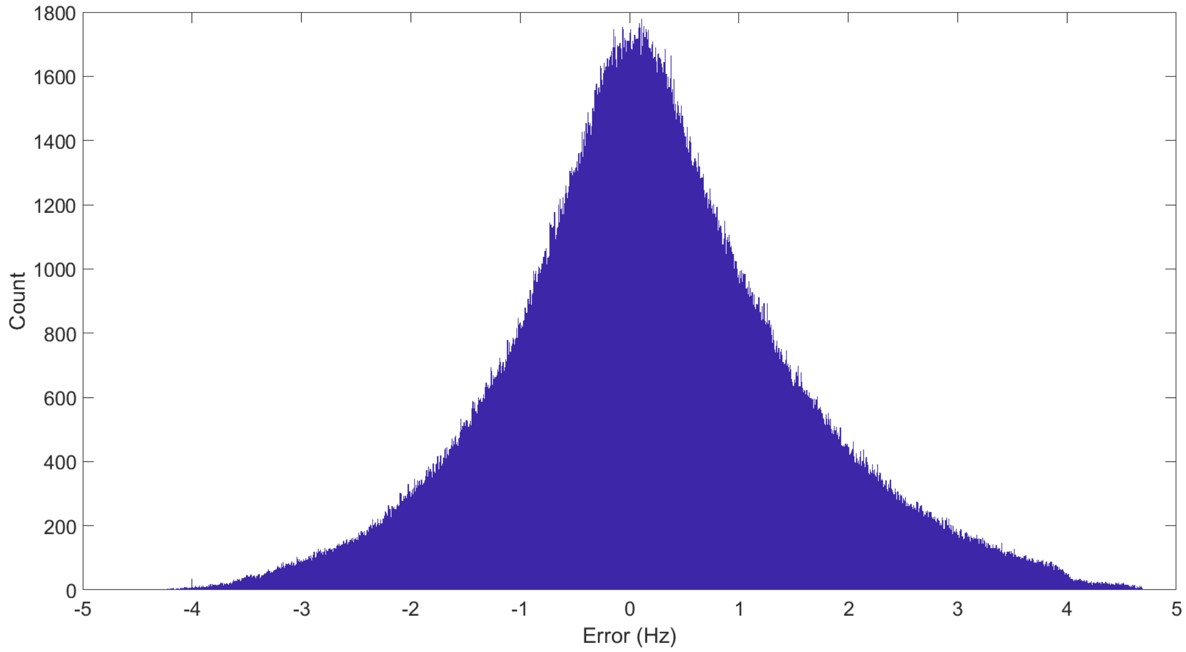

Figure 5.24. Instantaneous frequency error. Clips towards the end because outliers removed.

Hz error were obtained for the visual and auditory modes, respectively, confirming accurate functionality. Moreover, to our knowledge, this is the first-in kind of a platform and we provide results for future platforms to compare to. Such platforms then enable the currently hypothesis driven experiments in chronic pain; namely to investigate whether IAF matched stimulation could result in a more pronounced entrainment in chronic pain patients, and in result increased analgesic effects [6]–[8], [10].

5.5 Conclusion

In summary, this Chapter focused on the on-phone implementation and characterization of the closed-loop platform. These come in two different flavors: one which provides discrete phase-locked auditory stimulation for the sleep engineering application driver and the other for continuous frequency-matched audio-visual stimulation, for the chronic pain application driver. Both these used the PLL we tuned in Chapter 4 for feature extraction.

To characterize performance, we provided a test-setup, which allows for controlled experiments without the need for on-person tests to do technology validation. This involved streaming pre-recorded EEG via a DAQ and attenuating it to mimic an actual EEG signal, which then is picked up by an amplifier and streamed to the phone. The phone then processes the signal and provides stimulation, which is then measured back at the DAQ, alongside the streamed signal. This complete loop setup then enabled quicker technology validation within a controlled environment.

To characterize the system performance, we first validated the individual components of the platform in isolation i.e. the sensing and signal processing components in Section 5.2. Note that the stimulation component was already validated in Chapter 3 and hence is not included here. For the sensing component, we demonstrated good transmission accuracy with an RMSE of 3.24±0.13μV between the streamed and received signal, in addition to a per sample transmission latency of 62±24 ms. For the signal processing component, we achieved a phase error of 14±13 degrees when targeting the peak and a frequency matching error of 0.006±0.38 Hz, both of which matched the PC based results, under the same causal filter settings. In contrast, the latency was different for both platforms: for the phone, it was 0.04±0.33 ms and 0.04±0.22 ms for the phase and frequency, respectively. On the PC, the variance was slightly smaller with 0.001±0.165 ms and 0.001±0.019 ms, for the phase and frequency, respectively. This was expected due to the relatively lower computational power of the phone. Either way, the phone latency is in the sub-millisecond range, which is more than good enough. Given the per sample communications latency of 62 ms, a sub-millisecond signal processing latency then ensures that each sample is processed fast enough before the next sample is acquired, ensuring feasibility of a sample-by-sample approach.

After we had validated the individual components in isolation, we then proceeded to characterizing the complete platforms as a whole in Sections 5.3 and 5.4. For the discrete mode (sleep engineering) in Section 5.3, a phase targeting error of 12±60 degrees was reported, alongside an over-all loop-latency of 70±24 ms, both of which were comparable to other platforms available. Here, our contribution was providing the first on-phone implementation of the sleep engineering platform, enabling out-of-lab research on a more ubiquitous mobile platform such as the smartphone.

We then concluded the Chapter characterizing the second platform in Section 5.4, i.e. the one for continuous frequency matched audio-visual stimulation (chronic pain). For this, we reported a frequency matching error of -0.17±1.29 Hz and 0.20±1.3 Hz error for the visual and auditory modes, respectively. Here, our contribution is in providing the first-in kind of a platform, for a sample-by-sample frequency-matched audio-visual stimulation. The results presented here then could be used as a benchmark for future platforms to compare to.

Chapter 6

Conclusions

6.1 Contributions

In this thesis, we provide a multi-purpose light and sound based closed-loop neuromodulation platform. More importantly, unlike most other systems available today, ours is a smartphone based platform that runs on an Android app, making it fit for ambulatory use-cases and *in-the-wild* experiments. Even though this technology is aimed at the sleep engineering (phase-locked auditory stimulation) and chronic pain application drivers (frequency-matched audio-visual stimulation), it could be easily adapted for other applications. To create this platform, we first started with the open-loop stimulation in Chapter 3, after which we validated the feature extraction algorithms offline in Chapter 4. Both these then paved the way for the complete on-phone implementation in Chapter 5.

We started the thesis with the design of the open-loop stimulation app in Chapter 3. Here, our aim was to provide light and sound stimulation in the alpha-band (8–13 Hz), whilst ensuring signal continuity at frequency changes, to make it fit for closed-loop applications. We then explored three waveform generation methods for the visual stimuli: square interpolation, square-sine and sine. With the 60 Hz refresh rate constraint in mind, we then concluded that the square interpolation technique is best for open-loop application which don't require signal continuity. In contrast, for the closed-loop mode, the sine-method with a phase-accumulator was proposed to ensure signal continuity. For both, we showed the app delivered stimulation accurately over the alpha band, as evidenced by the R-squared error of 1 between the fitted line and the measured values. Finally, we also noted that the square sine method may overcome the trade-off between efficacy (square-wave) and signal continuity (sine-wave) via providing a square wave whilst also ensuring signal continuity but it requires phones with a refresh rate of 90 or 120 Hz, due to the nyquist limit. This makes it a better choice for use in the future, where smart-phones with higher refresh rate may become more ubiquitous. Next, for the auditory stimulation, we implemented binaural beats with a phase accumulator for signal continuity. Here, since refresh rates were not an issue, we just used a simple sinusoid and did not explore other methods. The stimulus was again delivered accurately, as evidenced by an R-squared error of 1 between the fitted line and measured values. For both auditory and visual stimulus, our key contribution is in providing stimulus generation that is usable in closed-loop applications, where the frequency is expected to change. To do this, we used a phase-accumulator to ensure signal continuity at frequency changes, and this was not something considered by current stimulus generation methods, which are mostly for open-loop applications, for example SSVEP-based BCI's. Moreover, we implemented both these stimulus methods on a phone and the app developed is currently being used in clinical work, by our collaborators. Hence, the open-loop app in itself is a contribution, and not just a means to an end for the closed-loop platform. After having characterized the stimulation platform, we then moved on-to closing the loop.

For closing the loop, we first start in Chapter 4, where we characterize the PLL performance offline before implementing it on the phone. To do this, we first started with characterizing the EEG amplitude, frequency and frequency rate-of-change in Section 4.2: the first two feeds into the PLL tuning methodology and the latter motivates frequency matched closed-loop stimulation on a sample-by-sample basis i.e. if the EEG frequency does not vary, there is no point closing the loop. Parts of this analysis such as the frequency rate-of-change have never been done before and we show that for the alpha band, it is 9.75±2.67 Hz/s. Then, in Section 4.3, we detailed the PLL tuning methodology, where we took a bottom up approach using PLL maths and control theory to get a initial starting point, after which the parameters were fine-tuned for optimal accuracy.

With the PLL tuned, we then moved onto characterizing accuracy for three different use-cases: Continuous phase locked stimulation (general applications), Discrete phase locked stimulation (Sleep engineering) and continuous frequency matched stimulation (chronic pain). Firstly, we characterized phase-locking performance across all bands in Section 4.4, to show that the PLL could work across all bands, and hence, be applicable for a wide variety of application drivers. Here, for all EEG bands, we showed that the PLL performed on average 14% better than the state-of-the-art FFT based methods, confirming its superiority in tracking phase-across all bands [44]. Compared to epoch-based methods like the FFT, the PLL does not assume signal stationarity over short periods of time and is able to adapt to the time-varying EEG, making it more accurate across all bands. In addition to this, we also showed that the inter-channel variance is low, negating the need for channel-specific tuning. After showing general applicability to all bands, we then narrowed down to our specific application drivers. Firstly, for the phase-locked auditory stimulation mode (sleep engineering application) in Section 4.5, we showed that the PLL can track slow oscillation phase accurately, as evidenced by a 11.2±11.2 degrees phase error, a result that was comparable to similar works [2]. Then, for the continuous frequency-matched alpha stimulation (chronic pain), we showed in Section 4.6 that the PLL tracks the instantaneous frequency in the alpha band well on a sample-by-sample basis, as evidenced by its 0.004±0.15 Hz error. Hence, having characterized the phase-frequency extraction for all these modes, we then moved onto the on-phone implementation in Chapter 5, which ties in the audio-visual stimulation in Chapter 3 and the signal processing in 4 all into one, to form the complete system.

For the on-phone implementation in Chapter 5, We provided two separate platforms: one which provides discrete phase-locked auditory stimulation for the sleep engineering application driver and the other that provides continuous frequency-matched audio-visual stimulation, for the chronic pain application driver. Both these used the PLL we tuned in Chapter 4, for extracting their respective features. To characterize performance, we provided a 'in-the-loop' testing platform, which streamed pre-recorded EEG data to an amplifier from a DAQ, whilst collecting the resulting stimulation waveforms emitted from the phone. This allowed us to validate the technology in a controlled environment, without having to do on-person tests. Using this test-setup, we first validated the individual components of the platform in isolation i.e. the sensing and signal processing components in Section 5.2, with the stimulation component already validated in Chapter 3. For sensing, we demonstrated good transmission accuracy with an RMSE of 3.24±0.13 μV between the streamed and received signal and a per sample latency of 62±24 ms. Similarly, for the signal processing component, we achieved a 14±13 degrees and a 0.006±0.38 Hz phase and frequency matching error, respectively, both of which matched the PC-based results. In contrast, the on-phone signal-processing run-time was 0.04±0.33 ms and 0.04±0.22 ms for the phase and frequency, respectively, whilst on the PC, the variance was lower with 0.001±0.165ms and 0.001±0.019 ms for the phase and frequency, respectively. Either way, the on-phone signal processing latency is in the sub-millisecond range and with our 250 Hz sensing sampling rate on the phone, the system could process each sample quick enough before the next one is acquired, which is ideal for driving down the loop latency. After validating the individual components in isolation, we then tied them together to form the whole. First, we started with the frequency matched stimulation for the chronic pain application driver in Section 5.4, where we reported a frequency matching error of -0.17±1.29 Hz and 0.20±1.30 Hz, for the visual and auditory modes, respectively. This is first-in-kind of a platform for the chronic pain application driver and we provide these values as a bench-mark for future platforms to compare to. Similarly, for the phase-locked auditory stimulation platform (sleep engineering) in Section 5.3, a phase targeting error of 12±60 degrees was reported, alongside an over-all loop-latency of 70±24 ms, both of which were comparable to other platforms available [2], [4], [5], [12], [13], [20]. Here, our contribution was providing the first on-phone implementation of the sleep engineering platform. Both these platforms would then enable *in-the-wild* research to be conducted on a ubiquitous mobile platform such as the smart-phone. Moreover, using our results from Section 4.4.3, these could easily be modified for other applications which require phase-frequency matched stimulation.

6.2 Limitations

Firstly, one of the limitations of this work is that we tested the platform on only one phone model: the Sony Xperia Z3, as this was the phone that came bundled with the smarting EEG amplifier. Moreover, for the on-phone stimulation, we showed in Chapter 3 that the sine-square algorithm would be preferable due to it being a square wave and at the same time satisfying signal continuity; however, we showed in Section 3.2.5 that the square sine-method can accurately deliver stimuli with continuity, when the refresh rate is at 90 Hz and 120 Hz, but not for 60 Hz, as is the case with the phone we used. Hence, due to lack of phone hardware with 90 Hz and 120 Hz refresh rates, we could not confirm the simulations on actual phones. Hence, an ananlysis on multiple phone models would be useful, both for the visual stimulation methods and the complete system.

Secondly, with regards to the PLL used, we have highlighted at various points the amplitude dependent performance of the PLL, with higher amplitudes producing lower errors, as detailed in Section 4.4.4. However, we have not considered any amplitude normalisation procedures in the real-time platform. It would be worth examining amplitude normalization procedures and whether it improves performance significantly. To tackle this, one could add in an automatic gain control component, or use an XOR phase detector instead of the simple multiplier detector. Both these would come with their own trade-offs such as increased complexity and adding more parameters to tune. This combined with the already good results obtained with the classical PLL then led us to stick with it and focus more on the on-phone implementation and the complete platform. Regardless, it would be useful to know if adding in an amplitude normalisation component before the PLL, would significantly improve performance.

Thirdly, another limitation of our work is that we did not include any artefact removal procedures in our signal processing pipeline. Our assumption was that the PLL is robust to noise and this can be intuitively understood by noting that the input signal buried in noise, will cause the signal to be delayed or advanced in a random manner, which in turn causes the phase detector output before the low pass filtering, to jitter around an average value. This jitter would then be smoothed out by the low pass filter in the PLL, which in turn translates to the VCO responding to error differences in the signal of interest and not the noise. Regardless, it would be useful to do an artefact robustness analysis, to check if the PLL performs as well in the presence of artefacts. That said, our current application drivers are not motion heavy i.e. the user is expected to be sitting still in a relaxed pose for the chronic pain application driver and sleeping for the sleep engineering application. However, for motion heavy applications with say lots of movements and walking, then this would serve as a useful reference to judge how robust the PLL is to noise, and if extra artefact removal steps are necessary.

Finally, it must also be noted that we have not bench-marked our PLL-based phase extraction method against Quantativave EEEG (qEEG), though we have done so for our PLL based frequency extraction method. According to the American Academy of Neurology, qEEG is the mathematical processing of EEG which results in the transformation of the raw EEG signals into domains that reveal more information; these are a set of quantitative metrics typically used in clinical routines alongside the raw EEG data [146]–[149]. Examples include source analysis, frequency analysis, topographic analysis, and higher order measures such as the Phase Lag Index (PLI), which measures the extent of synchronisation between different brain regions [147]. It is then useful to benchmark our PLL based phase-frequency extraction algorithm, against some of these qEEG metrics used in clinical routines. For this, two obvious qEEG metrics of relevance would be frequency based metrics such as the FFT power-spectrum and phase-based metrics such as thee Phase Lag Index (PLI) [147], [150]. With regards to the frequency domain, algorithms such as the Fourier transform and the Welch method is typically used in qEEG routines, to get the power spectrum of the EEG data [147], [150]. Here, we have showed in Section 4.6.2 that our PLL based frequency measures are good approximates of FFT based ones, in determining the dominant alpha frequency. However, for the phase based qEEG measures like the PLI, we have not conducted any similar bench-marking, at least explicitly. The PLI is defined as,

$$PLI = | < sign[sin(\Delta\Phi(t_k))))]] > |[147] \tag{6.1}$$

Here, $\Delta\Phi$, which is the phase difference between two different time-series, is used as a measure of synchronisation between 2 distinct brain regions and the signum function discards phase differences of 0, to avoid any spurious results due to volume conduction [147]. We have not bench-marked our PLL based phase-extraction method against this, say via obtaining PLI measures where the instantaneous phase is calculated using the PLL. Typical qEEG routines use the Hilbert transform to obtain the instantaneous phase measures for the PLI metric and we have already bench-marked our PLL-based phase measures, against the Hilbert based ones in Chapter 4 [147]. Hence, we did not see any further benefit of doing any PLL-based PLI measures, as this would be a whole different higher order analysis, looking into the brain connectivity measures between two different brain regions and does not sit well with our first-order measures such as single channel phase, which is what is needed for the application driver at hand i.e. sleep engineering. This maybe also the reason why all state-of-the-art platforms for sleep engineering did not benchmark their results against such qEEG metrics such as the PLI, and still achieved the desired clinical outcome whilst using the Hilbert transform as the benchmark for their phase-measures [2], [4], [5], [13]. this, we have also done.

6.3 Future work

For future work, firstly, one thing we did not do due to time-limitations and COVID, is to conduct on-person tests. It would be useful especially for the chronic pain application driver, where no target bio-marker exists. Hence, an important piece of future work would be to determine through on-person tests as to whether the closed-loop frequency matched stimulation we provide would result in stronger entrainment, and in turn increased analgesic effects. This would be a big study in itself and we chose to not do the clinical work here, but instead focused on creating technology to enable such work.

Another useful piece of future work would be to investigate the effects of using different EEG amplifiers on the system performance. We observed from Chapter 5 that the major contributor to the loop latency is the communications latency and hence it would be useful to know how different the latency is for other amplifiers, both research-grade and consumer-grade, and judge the effects on the system performance. For example, the recently announced Smarting Pro, which is advertised to have 5 times lower latency than the current system, which may mean that our system latency could be reduced by a factor of 5 with this.

Finally, the focus of this thesis has been on first order uni-variate features such as the phase and frequency, primarily due to the application drivers at hand. However, this does not give any information on the interaction between different brain regions, and their relation to brain-functions and behaviours. Recent evidence suggests that brain functions are not localized to separate regions, but a result of interactions between different brain regions, which is what brain connectivity analysis is all about [91], [151]–[155]. Out of the various connectivity measures available, it would be interesting to use the ones based on instantaneous phase measures such as the Phase Locked Value (PLV) and the Phase Lag Index (PLI) ti see if the PLL could be applied here to any benefit for real-time applications; perhaps it could be used to stimulate at finer time resolutions i.e. on a sample-by-sample basis [155]. This would be a natural extension to the current work we are doing.

References

[1] R. Rao, *Brain-computer interfacing: An introduction*. Cambridge University Press, Oct. 2013.

[2] G. Santostasi, R. Malkani, B. Riedner, M. Bellesi, G. Tononi, K. A. Paller, and P. C. Zee, “Phase-locked loop for precisely timed acoustic stimulation during sleep,” *J Neuroscience Methods*, vol. 259, pp. 101–114, 2016.

[3] H.-V. Ngo, J. C. Claussen, J. Born, and M. Mölle, “Induction of slow oscillations by rhythmic acoustic stimulation,” *Journal of sleep research*, vol. 22, Aug. 2012.

[4] N. Papalambros, G. Santostasi, R. Malkani, R. Braun, S. Weintraub, K. Paller, and P. Zee, “Acoustic Enhancement of Sleep Slow Oscillations and Concomitant Memory Improvement in Older Adults,” *Front. Hum. Neurosci.*, vol. 11, p. 109, 2017.

[5] J. Ong, J. Lo, N. Chee, G. Santostasi, K. Paller, P. Zee, and M. Chee, “Effects of phase-locked acoustic stimulation during a nap on EEG spectra and declarative memory consolidation,” *Sleep Med*, vol. 20, pp. 88–97, 2016.

[6] K. Ecsy, C. Brown, and A. Jones, “Cortical nociceptive processes are reduced by visual alpha-band entrainment in the human brain,” *Eur. J. Pain*, vol. 22, no. 3, pp. 538–550, 2018.

[7] L. Arendsen, S. Hugh-Jones, and D. Lloyd, “Transcranial alternating current stimulation at alpha frequency reduces pain when the intensity of pain is uncertain,” *J. Pain*, vol. 19, Mar. 2018.

[8] L. Arendsen, J. Henshaw, C. Brown, M. Sivan, J. Taylor, N. Trujillo-Barreto, A. Casson, and A. Jones, “Entraining alpha activity using visual stimulation in patients with chronic musculoskeletal pain: A feasibility study,” *Front. Neurosci.*, vol. 14, p. 828, Aug. 2020.

[9] N. J. Davis and M. G. van Koningsbruggen, ““Non-invasive” brain stimulation is not non-invasive,” *Frontiers in Systems Neuroscience*, vol. 7, pp. 1–4, 2013.

[10] K. Ecsy, A. Jones, and C. Brown, “Alpha-range visual and auditory stimulation reduces the perception of pain,” *European Journal of Pain*, vol. 21, no. 3, pp. 562–572, 2017.

[11] S. Ahn, J. Prim, M. Alexander, K. Mcculloch, and F. Frohlich, “Identifying and engaging neuronal oscillations by transcranial alternating current stimulation in patients with chronic low back pain: A randomized, crossover, double-blind, sham-controlled pilot study,” *J. Pain*, vol. 20, Sep. 2018.

[12] E. Debellemaniere, S. Chambon, C. Pinaud, V. Thorey, D. Dehaene, D. Léger, M. Chennaoui, P. Arnal, and M. Galtier, “Performance of an Ambulatory Dry-EEG Device for Auditory Closed-Loop Stimulation of Sleep Slow Oscillations in the Home Environment,” *Front. Hum. Neurosci.*, vol. 12, p. 88, 2018.

[13] M. Ferster, C. Lustenberger, and W. Karlen, “Configurable Mobile System for Autonomous High-Quality Sleep Monitoring and Closed-Loop Acoustic Stimulation,” *IEEE Sens. Lett.*, vol. 3, no. 5, pp. 1–4, 2019.

[14] K. Ecsy, “Analgesic effects of EEG alpha-wave entrainment on acute and chronic pain,” PhD thesis, University of Manchester, 2014.

[15] R. Cox, I. Korjouko, M. Boer, and L. Talamini, “Sound Asleep: Processing and Retention of Slow Oscillation Phase-Targeted Stimuli,” *PloS one*, vol. 9, Jul. 2014.

[16] A. Notbohm, J. Kurths, and C. Herrmann, “Modification of brain oscillations via rhythmic light stimulation provides evidence for entrainment but not for superposition of event-related responses,” *Front. Neurosci.*, vol. 10, Feb. 2016.

[17] R. Gulbinaite, T. Viegen, M. Wieling, M. Cohen, and R. VanRullen, “Individual alpha peak frequency predicts 10 hz flicker effects on selective attention,” *J. Neurosci.*, vol. 37, pp. 1163–17, Sep. 2017.

[18] E. Raffin and H. R. Siebner, “Transcranial brain stimulation to promote functional recovery after stroke,” *Current Opinion in Neurology*, vol. 27, no. 1, pp. 54–60, 2014.

[19] R. Nardone, Y. Höller, S. Leis, P. Höller, N. Thon, A. Thomschewski, S. Golaszewski, F. Brigo, and E. Trinka, “Invasive and non-invasive brain stimulation for treatment of neuropathic pain in patients with spinal cord injury: A review,” *The Journal of Spinal Cord Medicine*, vol. 37, no. 1, pp. 19–31, 2014.

[20] M. Leminen, J. Virkkala, E. Saure, T. Paajanen, P. Zee, G. Santostasi, C. Hublin, K. Müller, T. Porkka-Heiskanen, M. Huotilainen, and T. Paunio, “Enhanced Memory Consolidation Via Automatic Sound Stimulation During Non-REM Sleep,” *Sleep*, vol. 40, no. 3, Jan. 2017.

[21] A. O. Hebb, J. J. Zhang, M. H. Mahoor, C. Tsiokos, C. Matlack, H. J. Chizeck, and N. Pouratian, “Creating the feedback loop: closed-loop neurostimulation,” *Neurosurg Clin N Am*, vol. 25, no. 1, pp. 187–204, 2014.

[22] T. Ferree, M. Clay, and D. Tucker, “The spatial resolution of scalp eeg,” *Neurocomputing*, vol. 38-40, pp. 1209–1216, 2001.

[23] WikiPedia, *10-20 system (eeg)*, `https://en.wikipedia.org/wiki/10-20_system_(EEG)`, Accessed: 2021-10-3.

[24] T. Deer, “History of Neurostimulation,” in. Springer, 2011, ch. Atlas of Implantable Therapies for Pain Management, pp. 3–6.

[25] D. A. Malone, D. D. Dougherty, A. R. Rezai, L. L. Carpenter, G. M. Friehs, E. N. Eskandar, S. L. Rauch, S. A. Rasmussen, A. G. Machado, C. S. Kubu, A. R. Tyrka, L. H. Price, P. H. Stypulkowski, J. E. Giftakis, M. T. Rise, P. F. Malloy, S. P. Salloway, and B. D. Greenberg, “Deep Brain Stimulation of the Ventral Capsule/Ventral Striatum for Treatment-Resistant Depression,” *Biological Psychiatry*, vol. 65, no. 4, pp. 267–275, 2009.

[26] I. O. Bergfeld, M. Mantione, M. L. Hoogendoorn, H. G. Ruhé, P. Notten, J. Van Laarhoven, I. Visser, M. Figee, B. P. De Kwaasteniet, F. Horst, A. H. Schene, P. Van Den Munckhof, G. Beute, R. Schuurman, and D. Denys, “Deep brain stimulation of the ventral anterior limb of the internal capsule for treatment-resistant depression,” *JAMA Psychiatry*, vol. 73, no. 5, pp. 456–464, 2016.

[27] S. Little, A. Pogosyan, S. Neal, B. Zavala, L. Zrinzo, M. Hariz, T. Foltynie, P. Limousin, K. Ashkan, J. FitzGerald, A. L. Green, T. Z. Aziz, and P. Brown, “Adaptive deep brain stimulation in advanced Parkinson disease,” *Annals of Neurology*, vol. 74, no. 3, pp. 449–457, 2013.

[28] G. E. Gmel, T. J. Hamilton, M. Obradovic, R. B. Gorman, P. S. Single, H. J. Chenery, T. Coyne, P. A. Silburn, and J. L. Parker, “A new biomarker for subthalamic deep brain stimulation for patients with advanced Parkinson’s disease—a pilot study,” *Journal of Neural Engineering*, vol. 12, no. 6, 2015.

[29] B. Rosin, M. Slovik, R. Mitelman, M. Rivlin-Etzion, S. N. Haber, Z. Israel, E. Vaadia, and H. Bergman, “Closed-loop deep brain stimulation is superior in ameliorating parkinsonism,” *Neuron*, vol. 72, no. 2, pp. 370–384, 2011.

[30] S. Little and P. Brown, “What brain signals are suitable for feedback controzl of deep brain stimulation in Parkinson’s disease?” *Annals of the New York Academy of Sciences*, vol. 1265, no. 1, pp. 9–24, 2012.

[31] H. S. Mayberg, “Targeted electode-based modulation of neural circuits for depression,” *Journal of Clinical Investigation*, vol. 119, no. 4, pp. 717–725, 2009.

[32] B. D. Greenberg, L. A. Gabriels, D. A. Malone, A. R. Rezai, G. M. Friehs, M. S. Okun, N. A. Shapira, K. D. Foote, P. R. Cosyns, C. S. Kubu, P. F. Malloy, S. P. Salloway, J. E. Giftakis, M. T. Rise, A. G. MacHado, K. B. Baker, P. H. Stypulkowski, W. K. Goodman, S. A. Rasmussen, and B. J. Nuttin, "Deep brain stimulation of the ventral internal capsule/ventral striatum for obsessive-compulsive disorder: Worldwide experience," *Molecular Psychiatry*, vol. 15, no. 1, pp. 64–79, 2010.

[33] A. M. Lozano, H. S. Mayberg, P. Giacobbe, C. Hamani, R. C. Craddock, and S. H. Kennedy, "Subcallosal Cingulate Gyrus Deep Brain Stimulation for Treatment-Resistant Depression," *Biological Psychiatry*, vol. 64, no. 6, pp. 461–467, 2008.

[34] B. H. Bewernick, R. Hurlemann, A. Matusch, S. Kayser, C. Grubert, B. Hadrysiewicz, N. Axmacher, M. Lemke, D. Cooper-Mahkorn, M. X. Cohen, H. Brockmann, D. Lenartz, V. Sturm, and T. E. Schlaepfer, "Nucleus Accumbens Deep Brain Stimulation Decreases Ratings of Depression and Anxiety in Treatment-Resistant Depression," *Biological Psychiatry*, vol. 67, no. 2, pp. 110–116, 2010.

[35] S. H. Kennedy, P. Giacobbe, S. J. Rizvi, F. M. Placenza, Y. Nishikawa, H. S. Mayberg, and A. M. Lozano, "Deep brain stimulation for treatment-resistant depression: follow-up after 3 to 6 years," *American Journal of Psychiatry*, vol. 168, no. 5, pp. 502–510, 2011.

[36] A. M. Lozano, P. Giacobbe, C. Hamani, S. J. Rizvi, S. H. Kennedy, T. T. Kolivakis, G. Debonnel, A. F. Sadikot, R. W. Lam, A. K. Howard, M. Ilcewicz-Klimek, C. R. Honey, and H. S. Mayberg, "A multicenter pilot study of subcallosal cingulate area deep brain stimulation for treatment-resistant depression," *Journal of Neurosurgery*, vol. 116, no. 2, pp. 315–322, 2012.

[37] R. Shamir, A. Noecker, and C. McIntyre, *Cc by 3.0 <https://creativecommons.org/licenses/by/3.0>*, [Online; accessed June 29, 2018], 2014. [Online]. Available: `https://commons.wikimedia.org/wiki/File:Typical_deep_brain_stimulation_setup.jpg`.

[38] A. T. Barker, R. Jalinous, and I. L. Freeston, "Non-Invasive Magnetic Stimulation of Human Motor Cortex," *The Lancet*, vol. 325, no. 8437, pp. 1106–1107, 1985.

[39] E. Dayan, N. Censor, E. R. Buch, M. Sandrini, C. Physiology, and N. Section, "Non-invasive brain stimulation: from physiology to network dynamics and back," *Nature Neuroscience*, vol. 16, no. 7, pp. 838–844, 2016.

[40] A. Karabanov, A. Thielscher, and H. R. Siebner, "Transcranial brain stimulation: closing the loop between brain and stimulation," *Curr Opin Neurol*, vol. 29, no. 4, pp. 397–404, 2016.

[41] Saiote, Turi, Paulus, and Antal., *Cc by 3.0 <https://creativecommons.org/licenses/by/3.0>*, [Online; accessed June 1, 2020], 2013. [Online]. Available: `https://commons.wikimedia.org/wiki/File:Fnhum-07-00435-g001.jpg`.

[42] R. C. Kadosh, *The Stimulated Brain - Cognitive Enhancement Using Non-Invasive Brain Stimulation*. 2014, p. 38.

[43] A. Antal and W. Paulus, "Transcranial alternating current stimulation (tACS)," *Frontiers in Human Neuroscience*, vol. 7, no. June, pp. 1–4, 2013.

[44] F. Mansouri, K. Dunlop, P. Giacobbe, J. Downar, and J. Zariffa, "A Fast EEG Forecasting Algorithm for Phase-Locked Transcranial Electrical Stimulation of the Human Brain," *Front Neuroscience*, vol. 11, p. 401, 2017.

[45] Z. Tianbao, Z. Hong, and T. Zheng, "A study of brainwave entrainment based on eeg brain dynamics," *Computer and Information Science*, vol. 2, Apr. 2009.

[46] X. Gao, H. Cao, D. Ming, H. Qi, X. Wang, X. Wang, R. Chen, and P. Zhou, "Analysis of EEG activity in response to binaural beats with different frequencies," *Int J Psychophysiology*, vol. 94, no. 3, pp. 399–406, 2014.

[47] C. S. Herrmann, "Human EEG responses to 1-100Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena," *Experimental Brain Research*, vol. 137, no. 3-4, pp. 346–353, 2001.

[48] Y. T. Wang, Y. Wang, C. K. Cheng, and T. P. Jung, "Measuring Steady-State Visual Evoked Potentials from non-hair-bearing areas," *Conf Proc IEEE Eng Med Biol Soc.*, pp. 1806–1809, Aug. 2012.

[49] J. Frederick, D. Timmermann, H. Russell, and J. Lubar, "Eeg coherence effects of audio-visual stimulation (avs) at dominant and twice dominant alpha frequency," *J. Neurother.*, vol. 8, pp. 25–42, May 2005.

[50] M. Lee, C. Song, G. Shin, and S. Lee, "Possible Effect of Binaural Beat Combined With Autonomous Sensory Meridian Response for Inducing Sleep," *Front. Hum. Neurosci.*, vol. 13, p. 425, 2019.

[51] U. Kirk, A. Wieghorst, C. Nielsen, and W. Staiano, "On-the-Spot Binaural Beats and Mindfulness Reduces Behavioral Markers of Mind Wandering.," *J Cogn Enhanc*, vol. 3, pp. 186–192, 2019.

[52] S. Reedijk, A. Bolders, and B. Hommel, "The impact of binaural beats on creativity," *Front. Hum. Neurosci.*, vol. 7, no. 786, 2013.

[53] R. Le Scouarnec, R. Poirier, J. Owens, J. Gauthier, A. Taylor, and P. Rodeheaver, "Use of binaural beat tapes for treatment of anxiety: A pilot study of tape preference and outcomes," *Altern Ther Health Med.*, vol. 7, pp. 58–63, 2001.

[54] B. Ross and M. Lopez, "40 Hz Binaural beats enhance training to mitigate the attentional blink.," *Sci Rep*, vol. 10, p. 7002, 2020.

[55] M. Garcia-Argibay, M. Santed, and J. Reales, "Efficacy of binaural auditory beats in cognition, anxiety, and pain perception: a meta-analysis.," *Psychol Res.*, vol. 83, pp. 357–372, 2 2019.

[56] J. Nantawachara and Y. W., "Brain responses to 40-Hz binaural beat and effects on emotion and memory," *Int. J. Psychophysiol.*, vol. 120, pp. 96–107, 2017.

[57] H. Tang, M. Vitiello, M. Perlis, and R. B., "Open-Loop Neurofeedback Audiovisual Stimulation: A Pilot Study of Its Potential for Sleep Induction in Older Adults.," *Appl Psychophysiol Biofeedback.*, vol. 40, pp. 183–188, 3 2015.

[58] M. Garcia-Argibay, M. Santed, and J. Reales, "Binaural auditory beats affect long-term memory.," *Psychol Res.*, vol. 83, pp. 1430–2772, 2019.

[59] Harshvardhan Kamat, *Bineural beats*, [Online; accessed June 29, 2018]. [Online]. Available: `https://www.quora.com/What-are-the-best-binaural-beats-that-had-a-great-effect-on-you`.

[60] F. López-Caballero and C. Escera, "Binaural Beat: A Failure to Enhance EEG Power and Emotional Arousal," *Front. Hum. Neurosci.*, vol. 11, no. 557, 2017.

[61] P. Afshar, A. Khambhati, S. Stanslaski, D. Carlson, R. Jensen, D. Linde, S. Dani, M. Lazarewicz, P. Cong, J. Giftakis, P. Stypulkowski, and T. Denison, "A translational platform for prototyping closed-loop neuromodulation systems," *Frontiers in Neural Circuits*, vol. 6, no. January, pp. 1–15, 2013.

[62] C. Zrenner, P. Belardinelli, F. Muller-Dahlhaus, and U. Ziemann, "Closed-Loop Neuroscience and Non-Invasive Brain Stimulation: A Tale of Two Loops," *Front Cell Neuroscience*, vol. 10, p. 92, 2016.

[63] H. V. Ngo, T. Martinetz, J. Born, and M. Molle, "Auditory closed-loop stimulation of the sleep slow oscillation enhances memory," *Neuron*, vol. 78, no. 3, pp. 545–553, 2013.

[64] S. Jensen, G. Molnar, J. Giftakis, W. Santa, R. Jensen, D. Carlson, M. Lent, and T. Denison, "Information, energy, and entropy: Design principles for adaptive, therapeutic modulation of neural circuits," *ESSCIRC 2008 - Proceedings of the 34th European Solid-State Circuits Conference*, pp. 32–39, 2008.

[65] T. V. Salomons, K. Dunlop, S. H. Kennedy, A. Flint, J. Geraci, P. Giacobbe, and J. Downar, “Resting-state cortico-thalamic-striatal connectivity predicts response to dorsomedial prefrontal rTMS in major depressive disorder,” *Neuropsychopharmacology*, vol. 39, no. 2, pp. 488–498, 2014.

[66] C. Liston, A. C. Chen, B. D. Zebley, A. T. Drysdale, R. Gordon, B. Leuchter, H. U. Voss, B. J. Casey, A. Etkin, and J. Marc, “Default Mode Network Mechanisms of Transcranial Magnetic Stimulation in Depression,” vol. 76, no. 7, pp. 517–526, 2015.

[67] M. George, J. Taylor, and E. Short, “The Expanding Evidence Base for rTMS Treatment of Depression,” *Current opinion in psychiatry*, vol. 26, no. 1, pp. 13–18, 2013.

[68] M. C. Lo and A. S. Widge, “Closed-loop neuromodulation systems: next-generation treatments for psychiatric illness,” *International Review of Psychiatry*, vol. 29, no. 2, pp. 191–204, 2017.

[69] D. Cyron, “Mental side effects of deep brain stimulation (dbs) for movement disorders: The futility of denial,” *Front Integr Neurosci.*, vol. 10, no. 17, 2016.

[70] K. Famm, “Drug discovery: A jump-start for electroceuticals (vol 496, pg 159, 2013),” *Nature*, vol. 496, pp. 300–300, Apr. 2013.

[71] A. Berenyi, M. Belluscio, D. Mao, and G. Buzsaki, “Closed-Loop Control of Epilepsy by Transcranial Electrical Stimulation,” *Science*, vol. 337, no. 6095, pp. 735–737, 2012.

[72] T. Schmidt and D. Henrich, “Patient Adaptive Neurofeedback for ADHD Therapy,” *Biomedical Engineering*, vol. 57, no. 1, 2012.

[73] J.-S. Brittain, P. Probert-Smith, T. Z. Aziz, and P. Brown, “Tremor suppression by rhythmic transcranial current stimulation,” *Current Biology*, vol. 23, no. 5, pp. 436–440, 2013.

[74] K. S. Cha, D. Yeo, and K. H. Kim, “Neural signal processing for closed-loop neuromodulation,” *Biomedical Engineering Letters*, vol. 6, no. 3, pp. 113–122, 2016.

[75] C. N. Heck, D. King-Stephens, A. D. Massey, D. R. Nair, B. C. Jobst, G. L. Barkley, V. Salanova, A. J. Cole, M. C. Smith, R. P. Gwinn, C. Skidmore, P. C. Van Ness, G. K. Bergey, Y. D. Park, I. Miller, E. Geller, P. A. Rutecki, R. Zimmerman, D. C. Spencer, A. Goldman, J. C. Edwards, J. W. Leiphart, R. E. Wharen, J. Fessler, N. B. Fountain, G. A. Worrell, R. E. Gross, S. Eisenschenk, R. B. Duckrow, L. J. Hirsch, C. Bazil, C. A. O’Donovan, F. T. Sun, T. A. Courtney, C. G. Seale, and M. J. Morrell, “Two-year seizure reduction in adults with medically intractable partial onset epilepsy treated with responsive neurostimulation: Final results of the RNS System Pivotal trial,” *Epilepsia*, vol. 55, no. 3, pp. 432–441, 2014.

[76] F. T. Sun and M. J. Morrell, “The RNS System: Responsive cortical stimulation for the treatment of refractory partial epilepsy,” *Expert Review of Medical Devices*, vol. 11, no. 6, pp. 563–572, 2014.

[77] I. Osorio, M. G. Frei, S. Sunderam, J. Giftakis, N. C. Bhavaraju, S. F. Schaffner, and S. B. Wilkinson, “Automated seizure abatement in humans using electrical stimulation,” *Annals of Neurology*, vol. 57, no. 2, pp. 258–268, 2005.

[78] C. Zuo, X. Yang, Y. Wang, C. E. Hagains, A. L. Li, Y. B. Peng, and J. C. Chiao, “A digital wireless system for closed-loop inhibition of nociceptive signals,” *Journal of Neural Engineering*, vol. 9, no. 5, 2012.

[79] S. Zanos, “Closed-Loop Neuromodulation in Physiological and Translational Research,” *CSH Perspect. Med.*, vol. 9, 2018.

[80] F. Sun and M. Morrell, “Closed-loop Neurostimulation: The Clinical Experience,” *Neurotherapeutics*, vol. 11, pp. 553–563, 2014.

[81] I. Iturrate, M. Pereira, and J. Millán, “Closed-loop electrical neurostimulation: Challenges and opportunities,” *Current Opinion in Biomedical Engineering*, vol. 8, pp. 28–37, 2018.

[82] K. Mirza, C. Golden, K. Nikolic, and C. Toumazou, “Closed-Loop Implantable Therapeutic Neuromodulation Systems Based on Neurochemical Monitoring,” vol. 13, p. 808, 2019.

[83] Y. Jin, S. G. Potkin, A. S. Kemp, S. T. Huerta, G. Alva, T. M. Thai, D. Carreon, and W. E. Bunney, *Therapeutic effects of individualized alpha frequency transcranial magnetic stimulation (αTMS) on the negative symptoms of schizophrenia*, 2006.

[84] T. R. Mullen, C. A. Kothe, Y. M. Chi, A. Ojeda, T. Kerth, S. Makeig, T. P. Jung, and G. Cauwenberghs, “Real-time neuroimaging and cognitive monitoring using wearable dry EEG,” *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 11, pp. 2553–2567, 2015.

[85] A. Stopczynski, C. Stahlhut, M. K. Petersen, J. E. Larsen, C. F. Jensen, M. G. Ivanova, T. S. Andersen, and L. K. Hansen, “Smartphones as pocketable labs: Visions for mobile brain imaging and neurofeedback,” *International Journal of Psychophysiology*, vol. 91, no. 1, pp. 54–66, 2014.

[86] C. Zrenner, J. Tünnerhoff, C. Zipser, F. Müller-Dahlhaus, and U. Ziemann, “Brain-state dependent non-invasive brain stimulation using closed-loop real-time EEG signal analysis to trigger a TMS pulse with millisecond accuracy,” *Clinical Neurophysiology*, vol. 126, no. 8, p. 85, 2015.

[87] F. Mansouri, P. Fettes, L. Schulze, P. Giacobbe, J. Zariffa, and J. Downar, “A real-time phase-locking system for non-invasive brain stimulation,” *Front Neurosci*, vol. 12, p. 877, 2018.

[88] A. Widmann, E. Schroger, and B. Maess, “Digital filter design for electrophysiological data–a practical approach,” *J Neurosci Methods*, vol. 250, pp. 34–46, 2015.

[89] J. McIntosh and P. Sajda, “Estimation of phase in EEG rhythms for real-time applications.,” *J Neural Eng.*, vol. 17, 3 2020.

[90] Z. Ding, G. Ouyand, H. Chen, and X. Li, “Closed-loop transcranial magnetic stimulation of real-time EEG based on the AR mode method.,” *Biomedical Physics & Engineering Express*, vol. 6, 3 2020.

[91] L. L. Chen, R. Madhavan, B. I. Rapoport, and W. S. Anderson, “Real-time brain oscillation detection and phase-locked stimulation using autoregressive spectral estimation and time-series forward prediction,” *IEEE Trans Biomed Eng*, vol. 60, no. 3, pp. 753–762, 2013.

[92] C. Zrenner, D. Galevska, J. Nieminen, D. Baur, M. Stefanou, and U. Ziemann, “The shaky ground truth of real-time phase estimation,” *NeuroImage*, vol. 214, p. 116 761, 2020.

[93] S. Shirinpour, I. Alekseichuk, K. Mantell, and A. Opitz, “Experimental evaluation of methods for real-time eeg phase-specific transcranial magnetic stimulation,” *J. Neural Eng.*, 2020.

[94] L. Pion-Tonachini, S. H. Hsu, S. Makeig, T. P. Jung, and G. Cauwenberghs, “Real-time EEG Source-mapping Toolbox (REST): Online ICA and source localization,” *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 4114–4117, 2015.

[95] H. N. Locke, J. Brooks, L. J. Arendsen, N. K. Jacob, A. J. Casson, A. K. Jones, and M. Sivan, “Acceptability and usability of smartphone-based brainwave entrainment technology used by individuals with chronic pain in a home setting.A Systematic Review of Closed-Loop Feedback Techniques in Sleep Studies—Related Issues and Future Directions,” *Br J Pain.*, vol. 14, pp. 161–170, Aug. 2020.

[96] W. Flemons, N. Douglas, S. Kuna, D. Rodenstein, and J. Wheatley, “Access to Diagnosis and Treatment of Patients with Suspected Sleep Apnea,” *American Journal of Respiratory and Critical Care Medicine*, vol. 169, no. 6, 2004.

[97] J. Choi, M. Kwon, and S. Jun, “A Systematic Review of Closed-Loop Feedback Techniques in Sleep Studies—Related Issues and Future Directions,” *Sensors*, vol. 20, p. 2770, May 2020.

[98] D. Looney, P. Kidmose, C. Park, M. Ungstrup, M. Rank, K. Rosenkranz, and D. Mandic, “The in-the-ear recording concept: User-centered and wearable brain monitoring,” *IEEE Pulse*, vol. 3, no. 6, pp. 32–42, 2012.

[99] *Smarting Pro*, [Online; accessed June 1, 2021]. [Online]. Available: `https://mbraintrain.com/smarting-pro/`.

[100] A. B. L. Tort, M. A. Kramer, C. Thorn, D. J. Gibson, Y. Kubota, A. M. Graybiel, and N. J. Kopell, “Dynamic cross-frequency couplings of local field potential oscillations in rat striatum and hippocampus during performance of a T-maze task,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 51, pp. 20 517–20 522, 2008.

[101] R. T. Canolty, E. Edwards, S. Dalal, M. Soltani, S. Nagarajan, H. Kirsch, M. S. Berger, N. M. Barbaro, and R. T. Knight, “High gamma power is phase-locked to theta oscillations in human neocortex,” *Science*, Jan. 2006.

[102] C. de Hemptinne, E. S. Ryapolova-Webb, E. L. Air, P. A. Garcia, K. J. Miller, J. G. Ojemann, J. L. Ostrem, N. B. Galifianakis, and P. A. Starr, “Exaggerated phase–amplitude coupling in the primary motor cortex in Parkinson disease,” *Proceedings of the National Academy of Sciences*, vol. 110, no. 12, pp. 4780–4785, 2013.

[103] S. Lukas, E. Tahnée, T. Sanne, S. Teresa, D. Beatrice, A. Tom, and T. Alexander, “Phase of beta-frequency tACS over primary motor cortex modulates corticospinal excitability,” *Cortex*, vol. 103, pp. 142–152, 2018.

[104] P. Gordon, B. Zrenner, S. Dörre, P. Belardinelli, C. Zrenner, and U. Ziemann, “Influence of theta phase on EEG synchronized TMS to the dorsolateral prefrontal cortex,” *Brain Stimul.*, vol. 12, p. 526, 2019.

[105] D. Desideri, C. Zrenner, U. Ziemann, and P. Belardinelli, “Phase of sensorimotor μ-oscillation modulates cortical responses to TMS of the human motor cortex,” *J. Physiol.*, vol. 597, 2019.

[106] K. E. Mathewson, C. Prudhomme, M. Fabiani, D. M. Beck, A. Lleras, and G. Gratton, “Making Waves in the Stream of Consciousness: Entraining Oscillations in EEG Alpha and Fluctuations in Visual Awareness with Rhythmic Visual Stimulation,” *Journal of Cognitive Neuroscience*, vol. 24, no. 12, pp. 2321–2333, 2012.

[107] S. Hanslmayr, J. Gross, W. Klimesch, and K. Shapiro, “The role of α oscillations in temporal attention,” *Brain research reviews*, vol. 67, pp. 331–43, Jun. 2011.

[108] E. Somer, J. Allen, J. Brooks, V. Buttrill, and A. Javadi, “Theta Phase-dependent Modulation of Perception by Concurrent Transcranial Alternating Current Stimulation and Periodic Visual Stimulation,” *J Cogn Neurosci.*, vol. 32, pp. 1142–1152, 6 2020.

[109] R. VanRullen, N. Busch, J. Drewes, and J. Dubois, “Ongoing EEG Phase as a Trial-by-Trial Predictor of Perceptual and Attentional Variability,” *Front. Psychol.*, vol. 2, p. 60, 2011.

[110] R. E. Dustman and E. C. Beck, “Phase of alpha brain waves, reaction time and visually evoked potentials,” *Electroencephalography and Clinical Neurophysiology*, vol. 18, pp. 433–440, May 1965.

[111] F. J. Valera, A. Toro, E. Roy John, and E. Schwartz, “Perceptual framing and cortical alpha rhythm,” *Neuropsychologia*, vol. 19, pp. 675–686, Feb. 1981.

[112] S.-H. Oh, Y.-R. Lee, and H.-N. Kim, “A novel EEG feature extraction method using hjorth parameters,” *International Journal of Electronics and Electrical Engineering*, vol. 2, no. 2, pp. 106–110, 2014.

[113] W. Klimesch, P. Sauseng, and C. Gerloff, “Enhancing cognitive performance with repetitive transcranial magnetic stimulation at human individual alpha frequency,” *European Journal of Neuroscience*, vol. 17, no. 5, pp. 1129–1133, 2003.

[114] D. Veniero, A. Vossen, J. Gross, and G. Thut, “Lasting EEG/MEG Aftereffects of Rhythmic Transcranial Brain Stimulation: Level of Control Over Oscillatory Network Activity,” *Front. Cell. Neurosci.*, vol. 9, p. 477, 2015.

[115] O. Pino, “Neuro-Upper, a Novel Technology for Audio-Visual Entrainment. A Randomized Controlled Trial on Individuals with Anxiety and Depressive Disorders,” *BAOJ Medical & Nursing*, vol. 3, Aug. 2017.

[116] P. Albouy, S. Baillet, and R. Zatorre, “Driving working memory with frequency-tuned noninvasive brain stimulation,” *Ann N Y Acad Sci.*, vol. 1423, p. 184, 2018.

[117] A. Widge, M. Boggess, A. Rockhill, A. Mullen, S. Sheopory, R. Loonis, D. Freeman, and E. Miller, “Altering alpha-frequency brain oscillations with rapid analog feedback-driven neurostimulation,” *PLoS One*, vol. 13, no. 12, pp. 1–18, 2018.

[118] S. Haegens, H. Cousijn, G. Wallis, P. Harrison, and A. Nobre, “Inter- and intra-individual variability in alpha peak frequency,” *NeuroImage*, vol. 92, Feb. 2014.

[119] J. Weber, T. Klein, and V. Abeln, “Shifts in broadband power and alpha peak frequency observed during long-term isolation,” *Sci. Rep.*, vol. 10, Oct. 2020.

[120] B. Gutmann, A. Mierau, T. Hulsdunker, C. Hildebrand, A. Przyklenk, W. Hollmann, and H. K. Struder, "Effects of physical exercise on individual resting state eeg alpha peak frequency," *Neural Plast.*, vol. 2015, Feb. 2015.

[121] R. E. Best, *Phase-Locked Loops, Design, Simulation and Applications*. McGraw-Hill, 1999.

[122] J. Encinas, *Phase Locked Loops*. Chapman & Hall, 1993.

[123] P. Brennan, *Phase-Locked Loops, Principles and Practice*. Macmillan Press Ltd., 1996.

[124] R. Berry, R. Budhiraja, D. Gottlieb, D. Gozal, C. Iber, V. Kapur, C. Marcus, R. Mehra, S. Parthasarathy, S. Quan, S. Redline, K. Strohl, W. S. Davidson, and M. Tangredi, "Rules for Scoring Respiratory Events in Sleep: Update of the 2007 AASM Manual for the Scoring of Sleep and Associated Events," *Journal of Clinical Sleep Medicine*, vol. 8, no. 5, pp. 579–619, 2012.

[125] J. R. Deatherage, R. D. Roden, and K. Zouhary, "Normal Sleep Architecture," *Seminars in Orthodontics*, vol. 15, no. 2, pp. 86–87, 2009.

[126] S. Binder, K. Berg, F. Gasca, B. Lafon, L. C. Parra, J. Born, and L. Marshall, "Transcranial slow oscillation stimulation during sleep enhances memory consolidation in rats," *Brain Stimulation*, vol. 7, no. 4, pp. 508–515, 2014.

[127] M. Navarrete, J. Schneider, H.-V. V. Ngo, M. Valderrama, A. J. Casson, and P. A. Lewis, "Examining the optimal timing for closed-loop auditory stimulation of slow-wave sleep in young and older adults," *Sleep*, vol. 43, no. 6, Dec. 2019.

[128] V. Legrain, G. D. Iannetti, L. Plaghki, and A. Mouraux, "The pain matrix reloaded: A salience detection system for the body," *Progress in Neurobiology*, vol. 93, no. 1, pp. 111–124, 2011.

[129] H. Mano and B. Seymour, "Pain: A distributed brain information network?" *PLOS Biology*, vol. 13, pp. 1–4, Jan. 2015.

[130] R. Sitaram, T. Ros, L. Stoeckel, S. Haller, F. Scharnowski, J. Lewis-Peacock, N. Weiskopf, M. Laura Blefari, M. Rana, E. Oblak, N. Birbaumer, and J. Sulzer, "Closed-loop brain training: The science of neurofeedback," *Nature Reviews Neuroscience*, vol. 18, Dec. 2016.

[131] E. Elbogen, A. Alsobrooks, S. Battles, K. Molloy, P. Dennis, J. Beckham, S. McLean, J. Keith, and C. Russoniello, "Mobile Neurofeedback for Pain Management in Veterans with TBI and PTSD.," *Pain Med.*, vol. 22, no. 2, pp. 329–337, 2021.

[132] C. Herrmann, M. Murray, S. Ionta, A. Hutt, and J. Lefebvre, “Shaping intrinsic neural oscillations with periodic stimulation,” *J. Neurosci.*, vol. 36, Apr. 2016.

[133] S. K. Andersen and M. M. Muller, “Driving steady-state visual evoked potentials at arbitrary frequencies using temporal interpolation of stimulus presentation,” *BMC Neurosci*, vol. 16, p. 95, 2015.

[134] M. Nakanishi, Y. Wang, Y. Wang, Y. Mitsukura, and T. Jung, “Generating visual flickers for eliciting robust steady-state visual evoked potentials at flexible frequencies using monitor refresh rate,” *PLOS ONE*, vol. 9, no. 6, pp. 1–12, Jun. 2014.

[135] X. Chen, Y. Wang, S. Zhang, S. Xu, and X. Gao, “Effects of stimulation frequency and stimulation waveform on steady-state visual evoked potentials using a computer monitor,” *J. Neural Eng*, vol. 16, no. 6, 2019.

[136] X. Chen, Z. Chen, S. Gao, and X. Gao, “A high-ITR SSVEP-based BCI speller,” *Brain-Computer Interfaces*, vol. 1, no. 3, pp. 181–191, 2014.

[137] Y. Wang, Y. Wang, and T. Jung, “Visual stimulus design for high-rate ssvep bci,” *IET Digital Library*, pp. 1057–1058, 2010.

[138] Y. Wang, Y. Wang, C. Cheng, and T. Jung, “Developing stimulus presentation on mobile devices for a truly portable ssvep-based bci,” *Annu Int Conf IEEE Eng Med Biol Soc*, pp. 5271–5774, 2013.

[139] F. Teng, Y. Chen, A. Choong, S. Gustafson, C. Reichley, P. Lawhead, and D. Waddell, “Square or sine: Finding a waveform with high success rate of eliciting ssvep,” *Comput Intel Neurosc*, vol. 2011, p. 364 385, Sep. 2011.

[140] *Brain Waves - Binaural Beats*, [Online; accessed June 1, 2021]. [Online]. Available: `https://play.google.com/store/apps/details?id=net.luizmello.brainwaves`.

[141] *Binaural Beats Generator*, [Online; accessed June 1, 2021]. [Online]. Available: `https://play.google.com/store/apps/details?id=com.tmsoft.whitenoise.generator.beats`.

[142] L. T. Trujillo, C. T. Stanfield, and R. D. Vela, “The effect of electroencephalogram (eeg) reference choice on information-theoretic measures of the complexity and integration of eeg signals,” *Front Neurosci*, vol. 11, p. 425, 2017.

[143] J. J. Newson and T. C. Thiagarajan, “Eeg frequency bands in psychiatric disorders: A review of resting state studies,” *Front Hum Neurosci.*, vol. 12, Jan. 2018.

[144] C. S. Nayak and A. C. Anilkumar, *Eeg normal waveforms*, `https://www.ncbi.nlm.nih.gov/books/NBK539805/`, Accessed: 2021-04-18.

[145] D. Abramovitch, “Phase-locked loops: a control centric tutorial,” *Proceedings of the 2002 American Control Conference*, vol. 1, pp. 1–15, 2003.

[146] P. Kanda, R. Anghinah, M. Smidth, and J. Silva, “The clinical use of quantitative eeg in cognitive disorders.,” *Dement Neuropsychol.*, vol. 3, pp. 195–203, 2009.

[147] M. Chaturvedi, J. Bogaarts, C. Kozak, F. Hatz, U. Gschwandtner, A. Meyer, P. Fuhr, and V. Roth, “Phase lag index and spectral power as qeeg features for identification of patients with mild cognitive impairment in parkinson’s disease.,” *Clin Neurophysiol.*, vol. 130, pp. 1937–1944, 2019.

[148] M. Bellesi, B. Riedner, G. Garcia-Molina, C. Cirelli, and G. Tononi, “Enhancement of sleep slow waves: Underlying mechanisms and practical consequences,” *Frontiers in systems neuroscience*, vol. 8, p. 208, Oct. 2014.

[149] L. L. Popa, H. Dragos, C. Pantelemon, O. V. Rosu, and S. Strilciuc, “The role of quantitative eeg in the diagnosis of neuropsychiatric disorders.,” *J Med Life.*, vol. 13, pp. 8–15, 2020.

[150] L. Billeci, F. Sicca, K. Maharatna, F. Apicella, A. Narzisi, G. Campatelli, S. Calderoni, G. Pioggia, and F. Muratori, “On the application of quantitative eeg for characterizing autistic brain: A systematic review.,” *Front Hum Neurosci.*, vol. 7, p. 442, 2013.

[151] M. Hassan and F. Wendling, “Electroencephalography Source Connectivity,” *IEEE Signal Processing Magazine*, pp. 81–96, 2018.

[152] B. He, L. Yang, C. Wilke, and H. Yuan, “Electrophysiological Imaging of Brain Activity and Connectivity:Challenges and Opportunities,” *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 7, pp. 1918–1931, Jul. 2011.

[153] J.-M. Schoffelen and J. Gross, “Source connectivity analysis with MEG and EEG,” *Human brain mapping*, vol. 30, pp. 1857–65, Jun. 2009.

[154] S. Tirdad, M. A.A.P, and A. M.N., “Cumulative effects of theta binaural beats on brain power and functional connectivity,” *Biomed. Signal. Proces.*, vol. 42, pp. 242–252, 2018.

[155] J. García-Prieto, R. Bajo, and E. Pereda, “Efficient computation of functional brain networks: towards real-time functional connectivity,” *Front. Neuroinform.*, vol. 11, p. 8, 2017.