

## Development of a mobile-based EMG biofeedback system –

How can EMG-biofeedback training be transferred from the clinic to the patient's home environment?

Master's thesis in Biomedical Engineering

THUNDLUCK SEREEVORAVITGUL



MASTER'S THESIS EX108/2018

## Development of a mobile-based EMG biofeedback system –

How can EMG-biofeedback training be transferred from the clinic to  
the patient's home environment?

THUNDLUCK SEREEVORAVITGUL



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2018

Development of a mobile-based EMG biofeedback system –  
How can EMG-biofeedback training be transferred from the clinic to the patient’s  
home environment?  
THUNDLUCK SEREEVORAVITGUL

© THUNDLUCK SEREEVORAVITGUL, 2018.

Supervisor: Associate Professor Leif Sandsjö and Li Guo, University of Borås  
Examiner: Assistant Professor Stefan Candefjord, Department of Electrical  
Engineering

Master’s Thesis EX108/2018  
Department of Electrical Engineering  
Chalmers University of Technology  
SE-412 96 Gothenburg  
Telephone +46 31 772 1000

Cover: Illustration of the methodology of this thesis.

Typeset in L<sup>A</sup>T<sub>E</sub>X  
Gothenburg, Sweden 2018

Development of a mobile-based EMG biofeedback system –  
How can EMG-biofeedback training be transferred from the clinic to the patient’s  
home environment?

THUNDLUCK SEREEVORAVITGUL

Department of Electrical Engineering  
Chalmers University of Technology

## Abstract

Muscle pain and muscle disorder occur in all ages. The abnormalities could be, for example, pains in office syndrome caused by incorrect working postures, parietic muscles in after stroke patients, or even genetic muscle diseases e.g. Muscular Dystrophies. One common treatment for muscle related problem is a physical therapy using biofeedback technique with a guidance from physical therapist who will introduce the patients to perform muscle training.

This research is a collaboration between universities and hospitals, which aims to bring the clinical application to the home-based treatment. In order to do that, an Electromyography (EMG) biofeedback application has been developed on an Android™ smartphone device with a connection to an EMG hardware unit produced by Shimmer™. The system will acquire real-time EMG signal of the muscle activities by using surface electrodes connected to the EMG unit and transmit EMG data via Bluetooth®. The data will be visualized on the smartphone device as time-series graphs and bar plots with adjustable threshold for the different muscle training levels. The EMG data can optionally be saved as text files in the device storage for further analysis. Pre-processing of the biomedical signal or EMG signal is done by digital filters and average rectified value methods developed in the software application. The prototype system was tested by the research group members and resulted in a possible visual perceptive feedback system with well mobility for home environment usage. The feasibility in Smart Textile was ensured by evaluating the system with Smart Textile electrodes introduced by Scan-to-knit project. The result of the acquired signal from the textile electrodes is as effective as the EMG signal acquired from the traditional contact gel electrodes. The proposed system is mainly beneficial because the patients can do the therapy anytime anywhere based on the guideline program provided by the physical therapist as well as reduce the costs of visit the hospital frequently and enhance the latency of the rehabilitation due to the accessible system.

Keywords: EMG biofeedback, mobile application, Android™ development, Shimmer™.



## Acknowledgements

The thesis was conducted in collaboration with MedTech West. I would like to express my great attitude to my supervisor, associate professor Leif Sandsjö and my examiner assistant professor Stefan Candefjord who motivate, encourage and guide me throughout the whole research and this report. I also particularly appreciate for the assistance in Smart Textile electrodes given by Li Guo, a postdoctoral researcher from University of Borås, and the warm welcome from Cecilia Ehrenborg, an occupational therapist, and her team at Södra Älvsborgs Sjukhus for the visiting. Apart from that, the software development would not be done without the support from Shimmer<sup>TM</sup> service and my colleague, Araya Watthanakaruna. Last but not least my thesis would not be achieved without the great power from my parents who pushed me forward to this point with their greatest love and all support, especially my aunts who gave me an inspiration and motivation for all of my works due to their muscle weakness problem in the elderly.

Thundluck Sereevoravitgul  
Gothenburg, December 2018





# Contents

<b>List of Figures</b>	<b>xi</b>
<b>Abbreviations</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Aim and Limitations . . . . .	3
1.3 Research Questions . . . . .	3
<b>2 Background and Theory</b>	<b>5</b>
2.1 Biofeedback system . . . . .	5
2.2 EMG signal . . . . .	6
2.3 EMG Signal Recording . . . . .	8
2.3.1 Traditional Surface Electrode . . . . .	8
2.3.2 Textile electrode . . . . .	10
2.4 EMG signal Processing . . . . .	10
2.4.1 Artifacts and Noises . . . . .	10
2.4.2 Digital filter . . . . .	11
2.4.3 Signal Analysis . . . . .	11
2.5 EMG biofeedback system at SÄS . . . . .	12
2.5.1 1-Channel EMG biofeedback system . . . . .	12
2.5.2 2-Channel EMG biofeedback system . . . . .	12
2.5.3 Technology integration . . . . .	13
<b>3 Developing process and methods</b>	<b>15</b>
3.1 Interview and Study Visit . . . . .	15
3.2 Design Criteria and Flowchart . . . . .	15
3.3 EMG device - Shimmer3 EMG unit . . . . .	17
3.4 Android Development . . . . .	19
<b>4 Results</b>	<b>23</b>
4.1 System Overview . . . . .	23
4.2 Signal Amplification . . . . .	24
4.3 Signal Processing . . . . .	24
4.3.1 Filtering . . . . .	24
4.3.2 Signal Analysis . . . . .	24
4.3.3 Real-time processing . . . . .	25

4.4	Feedback . . . . .	25
4.4.1	Application Specification . . . . .	25
4.4.2	Application Programming . . . . .	25
4.4.3	Data Presenting and UI . . . . .	26
4.5	Scenarios . . . . .	28
4.5.1	Relaxation training . . . . .	28
4.5.2	Strengthen training . . . . .	29
4.6	System Evaluation . . . . .	29
4.7	Smart Textile Electrode Evaluation . . . . .	31
<b>5</b>	<b>Discussion</b>	<b>33</b>
5.1	EMG unit (hardware) . . . . .	33
5.2	EMG biofeedback System . . . . .	34
5.3	Application of usage . . . . .	34
5.4	Feasibility in Smart Textile Electrode . . . . .	35
5.5	Future Work . . . . .	35
5.5.1	Function . . . . .	35
5.5.2	Application Design . . . . .	36
<b>6</b>	<b>Conclusion</b>	<b>37</b>
	<b>Bibliography</b>	<b>39</b>
<b>A</b>	<b>Appendix 1 - Android code</b>	<b>I</b>
A.1	Main function for graph activity . . . . .	I
A.2	Acquiring and Processing Data . . . . .	II
A.3	Filtering . . . . .	VII
A.4	Plotting . . . . .	XIII

# List of Figures

1.1	An illustration of examples of posture in different situation which may lead to the pain or disorder. (Freepik, 2018) . . . . .	2
2.1	A diagram shows a single motor unit which consists of a motor neuron and a bundle of muscle fiber that it innervates. (redraw from Webster, 2010) . . . . .	6
2.2	An illustration diagram represents an initiative of the muscle action potential at the neuromuscular junction. It starts from a coming nerve impulse from a motor neuron along a synaptic vesicle which triggers a neurotransmitter substance, so called acetylcholine (ACh), to diffuse across a synaptic cleft. After that ACh will activate an ion channel which especially allows sodium ions ( $Na^+$ ) to pass through it. The inflow of $Na^+$ then causes a muscle action potential to further innervate the muscle fibers. (redraw from Gerard J. Tortora, 2010) . . . . .	7
2.3	An illustration shows a signal diagram of an example of measuring the muscle signal at the biceps and triceps muscle by using surface EMG. The electrical potential is recorded by electrodes and amplified by an EMG signal processing unit. (redraw from Vogel, 2016) . . . . .	9
2.4	An example of EMG signal in time-domain. The baseline interval indicates muscle relaxation and the muscle activation indicates muscle contraction or muscle activity level which can be evaluated by the amplitude (PlayRobots, 2018). . . . .	9
2.5	A traditional surface electrode that used in this project produced by Covidien <sup>TM</sup> (bio-medical, 2018). . . . .	10
2.6	An illustration of 1-channel EMG biofeedback system . . . . .	12
2.7	An EMG amplifier unit using at SÄS. Gains can be adjusted by the white knob and threshold level can be selected by the red and black knob for channel A and channel B respectively. The blue knob is used for adjusting the delay in a unit of second(s). . . . .	13
3.1	A flowchart shows the application flow of data. . . . .	16
3.2	Shimmer3 EMG unit . . . . .	17
3.3	A block diagram of Shimmer3 EMG unit is illustrated. . . . .	18
3.4	An example of positioning of electrodes for EMG measurement. . . . .	18
3.5	A user interface of ShimmerCapture application . . . . .	19
3.6	Statistics of mobile operating system market share in the worldwide are indicated as a line and bar graph. (data from StatCounter, 2017). . . . .	20

3.7	At welcome page of Android Studio program, the development can be started from selecting 'Import project' -> 'ShimmerAndroidInstrumentDriver' as indicated in the red boxes, . . . . .	21
4.1	A block diagram shows the overview of the EMG biofeedback system.	23
4.2	An illustration shows how to enable Developer Options for Samsung Galaxy 7 . . . . .	26
4.3	An illustration shows how to turn on a debugging mode and allow the installation from other source. . . . .	26
4.4	Illustrations shows a design of the user interface which can present EMG data as bars or graph. . . . .	27
4.5	Screenshots of two scenarios of muscle relaxation training and strengthen training or activation training mode with a threshold of 25 are presented. The left bar represents EMG values from Channel 1 and the right bar represents EMG values from Channel 2. . . . .	28
4.6	The electrode locations and system connection for shoulder's muscle activity investigation are presented. . . . .	30
4.7	Evaluations of the system have been done by three subjects. They are all healthy adults. . . . .	30
4.8	The evaluation of the EMG biofeedback system with smart textile electrodes was done at the Textile Fashion Center in Borås. . . . .	31

# Abbreviations

<b>ADC</b>	Analog-to-Digital Converter
<b>Ag/AgCl</b>	Silver/Silver Chloride
<b>API</b>	Application Programming Interface
<b>ARV</b>	Average Rectified Value
<b>Ch</b>	Channel
<b>ECG</b>	Electrocardiography
<b>EMG</b>	Electromyography
<b>FIR</b>	Finite Impulse Response
<b>Hz</b>	Hertz (unit)
<b>IIR</b>	Infinite Impulse Response
<b>JDK</b>	Java Development Kit
<b>SÄS</b>	Södra Älvsborgs Sjukhus
<b>SDK</b>	Software Development Kit
<b>UI</b>	User Interface



# 1

## Introduction

How much time do you spend on your smartphone or computer each day? 2 hours? 8 hours? or more? Did you realize which posture do you do when you are using them? Do you think you have a good posture? Have you ever had a neck pain or back pain? Nowadays people usually face with their technology devices, for instance, smartphone, tablet or even computer, and people spend a lot of time with those devices. According to Statista.com, the statistic shows that 76 percent of the respondents spend more than 3 hours a day on their smartphone and one-third of them spend more than 7 hours a day on their device (*Smartphone daily usage time worldwide 2017 / Statistic* 2018). Since December 2014, PwC (Isaac et al., 2014) also referred to the International Data Corporation (IDC) survey (Levitas, 2013) that:

"... 79 percent of us keep our smartphones within arm's reach for all but two hours of our waking day. In fact, one-fourth of us will have our phones within reach at all times."

After keeping an eye on people's posture while using devices, most of them have a wrong posture which is not good for their body structure, especially at the neck and shoulder. On the other side, not only using technology device but you may stay in a wrong posture, even while you are reading this thesis. Some of common incorrect postures are exemplified in Figure 1.1. These could bring a problem with musculoskeletal pain or even mental health such as stress or other kind of symptoms.

One solution for muscle related treatment could be physical therapy using biofeedback technique. The origin of the word "biofeedback" was neologized in the 1960s to describe experimental processes of involuntary physiological function of the body. Then, biofeedback is stated to be a method of measuring a physiological signal from a person and feeding the information back to that person in a perceivable way, e.g. visual or audio, in order to raise the awareness of the subject and be able to control the physiological functions. Generally biofeedback uses various kinds of sensors to detect either mechanical or electrophysiological signals from a body such as kinematics of body movement, temperature, or heart rate (Roberto M., 2016b).

In this thesis, the physiological part that is studied is the muscle in which the sensors are surface electrodes and the information on muscle activities can be fed back as a visual or an audio. It can also be called Electromyography (EMG) biofeedback, which EMG referred to a technique of measuring neuronal activity or myoelectric signals in the muscles. The possible studies of the muscle activities are muscle relax-



**Figure 1.1:** An illustration of examples of posture in different situation which may lead to the pain or disorder. (Freepik, 2018)

ation, muscle activation or contraction, and training on muscle control. Therefore, using biofeedback technique can notice the patient to aware of their unconscious muscle tension or unfavorable postures. In addition, the EMG biofeedback can be applied in other applications, for instance chronic pain disorder, neuromuscular rehabilitation, sport training, stroke patient (Kim, 2017) or bladder dysfunction (Horowitz, 2006).

However, the traditional EMG biofeedback system is accessible only in hospital or clinic where the patient need to frequently visit for rehabilitation. By the time past and developing of technology, it can be found that mobile devices such as a smart-phone or tablet can be an option for visualizing the biofeedback information and it allows the patient to practice at their home environment. These would enhance an efficiency of restoration and reduce the time wasting for transport to the clinic.

### 1.1 Motivation

The rehabilitation clinic at Södra Älvsborgs sjukhus (SÄS) has a long tradition in using EMG signal for biofeedback, but there are several issues that can be developed e.g availability of the EMG acquiring hardware, mobility of the system, new technology integration, etc. Therapists at the clinic have expressed interest in improving the current EMG biofeedback system to be able to move freely and allow the patient to train themselves at home. Therefore, this thesis is motivated from them.



## 1.2 Aim and Limitations

This research aims to develop a **mobile-based** (Android™ compatible) **EMG biofeedback** solution with a concept of bringing the clinical application to the patient's **home environment** as well as investigate the feasibility of home-based biofeedback training utilizing mobile devices (tablet or smartphone) and a set of textile electrodes targeting specific muscles or muscle groups. This thesis emphasizes a system approach, i.e. focusing on demonstrating a complete working system with limited functionality rather than elaborate solutions of the constituent parts.

## 1.3 Research Questions

- Is it possible to transfer a raw EMG signal acquired by an EMG device via wireless communication, i.e. Bluetooth®, and visualize it on a mobile device in real-time?
- Can the real-time EMG signal be presented as biofeedback information on a mobile-based device?
- Can the mobile-based EMG-biofeedback system be used with a smart textile technology targeting specific muscles or muscle groups?
- Can EMG-biofeedback training be transferred from the clinic to the patient's home environment?



# 2

## Background and Theory

This section contains fundamental background and theories behind the EMG biofeedback system. The physiology of muscle activation and myoelectric signal generation will be described. Then, a tool for measuring EMG signal which is an electrode and its type will be discussed as well as essential algorithms used to analyze the signal.

### 2.1 Biofeedback system

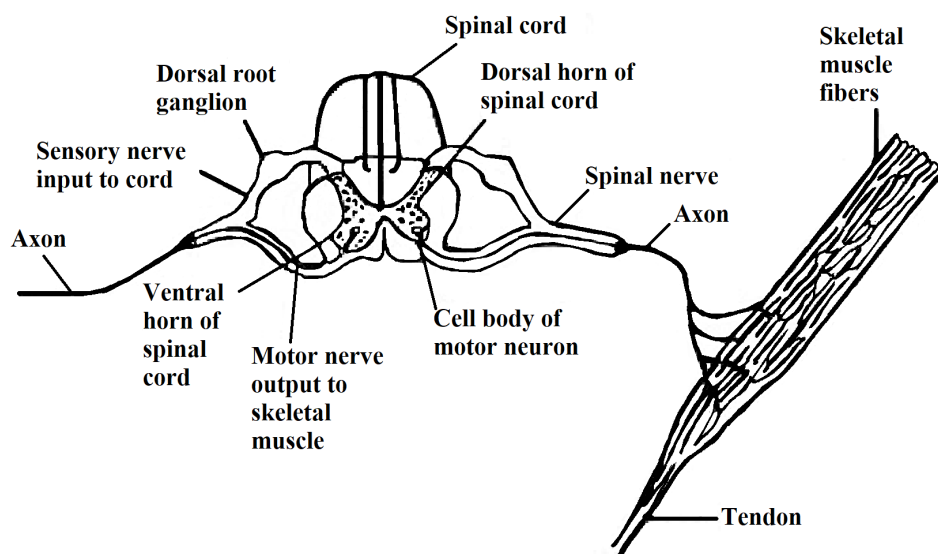
In a review of biofeedback in rehabilitation by Giggins et al. (Giggins et al., 2013), the biofeedback has been used in rehabilitation application for more than 50 years to help the patient in the physical control after the injury. The review has categorized the different biofeedback in rehabilitation into 2 main categories. One is the physiological biofeedback and another one is biomechanical biofeedback. The physiological system of the body that can be investigated and provide biofeedback can be divided into 3 types. The first type is the physiological biofeedback which includes neuromuscular biofeedback, EMG biofeedback, and real-time ultrasound imaging biofeedback. The second type is cardiovascular biofeedback which can be a heart rate biofeedback and heart rate variability or respiratory sinus arrhythmia. The last type is the respiratory system, for instance breathing measurement or pressure. Meanwhile, the biomechanical biofeedback can be divided into 3 main types which are movement, postural control and force. Within the study of rehabilitation, a postural correction is also another utilize of biofeedback in the neuromuscular system. A research written by Gaffney BM. et al.(Gaffney et al., 2015) has shown an experiment of adopting a neutral posture by a typing task. There is an evidence of the ability to modify posture using their biofeedback system to improve posture and prevent neck/shoulder pain during computer work .

Another study of biofeedback shows that biofeedback was used in after injury, for example, after a stroke (Kim, 2017) or after whiplash-associated disorders (Ehrenborg and Archenholtz, 2010). A case study by Cecilia Ehrenborg at SÄS expressed an evaluation of the effectiveness of surface EMG biofeedback training to an interdisciplinary rehabilitation program about the consequences of activities in daily life. It resulted in no support of the effectiveness of surface EMG biofeedback training for people with long-lasting pain after whiplash after four weeks. However, the study shows a promising consequence as the surface EMG biofeedback has increased awareness of the treatment group's working posture in daily life. With the inconclusive result, the interest in the surface EMG biofeedback system is still expressed.

There is also a biofeedback system that represents the feedback in an entertaining computer game environment. A research made by Madusha J., et al. (Jayarathne et al., 2015) describes the development of EMG biofeedback in the game environment. The subject was asked to control the Flappy Bird in the game to fly and avoid the obstacle by controlling the bicep muscle. The level of the bird is proportional to the level of the muscle flexion. Another application related to a game control proposed by Jovanov (Jovanov and Popović, 2014) is a use of EMG biofeedback system for the humans after brain injury. A car race computer gaming was used as a feedback of the movement of fingers extension and flexion, and radial and ulnar flexion. The system was proven as a fun and motivating solution for the exercise in rehabilitation of the patients.

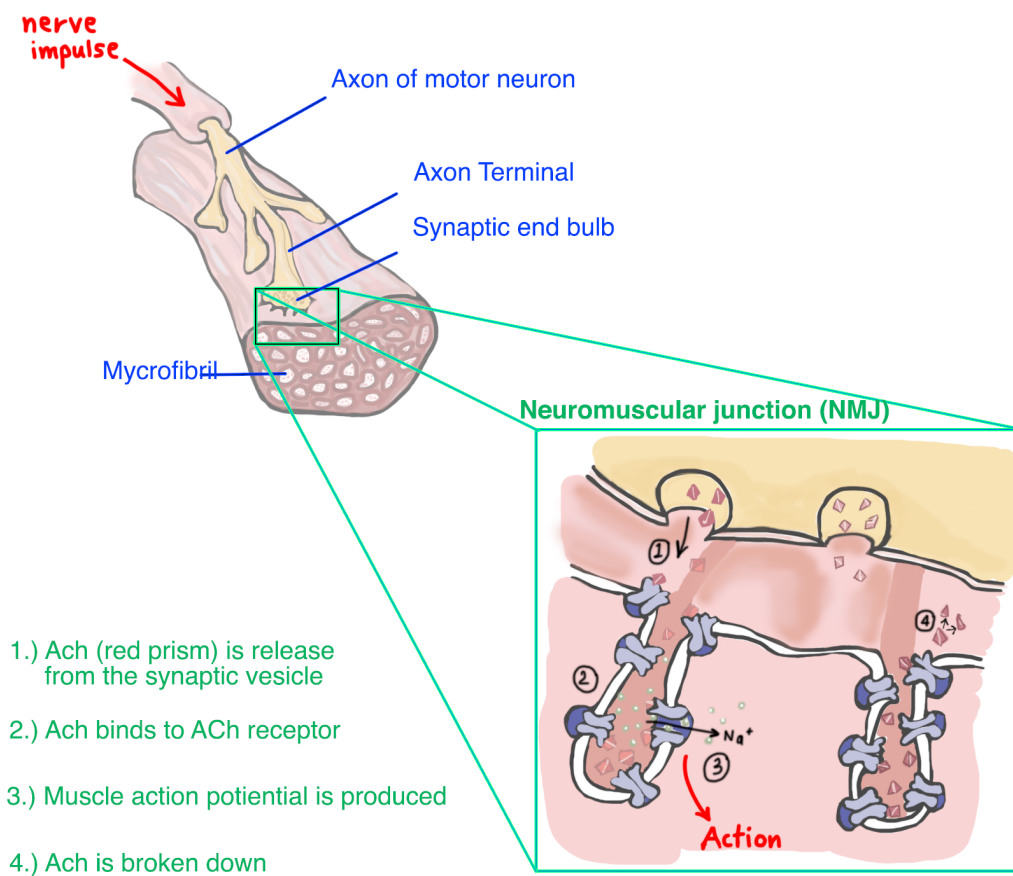
Similarly to this thesis, Richer (Richer et al., 2014) had developed an Android™ application for enabling a real-time analysis of ECG and EMG for biking athlete. He used Shimmer™ sensors for acquiring electrical signals and transferred the data via Bluetooth® to the mobile device. The system is mainly used to monitor the user's training parameters by evaluating the user's training status for live training supervision from heart beat detection (via ECG) and treadle detection (via EMG). Advantages of using Shimmer™ in this research are compact and lightweight as well as wireless communication accessible for providing real-time data. However, a constrain from the user's movement during biking affected the accuracy of the algorithm due to motion artifacts and other artifact i.e. the friction between electrodes and clothes.

## 2.2 EMG signal



**Figure 2.1:** A diagram shows a single motor unit which consists of a motor neuron and a bundle of muscle fiber that it innervates. (redraw from Webster, 2010)

EMG signal is an electrical signal which represents muscle activities. The fundamental process underlying the muscle activation is a transmission of a force generated by individual muscle fibers through the neuromuscular system. The origin of EMG or myoelectric signal simply starts from the basic unit of the neuromuscular system called motor unit. A motor unit consists of a motor neuron in spinal cord, its dendrites, axon, and the bundle of muscle fibers as shown in Figure 2.1. When the motor neuron receives sensory or input from other parts of the nervous system, an electrical impulse or the action potential is generated by a motor neuron through all the axons and innervates muscle fibers in order to response an action (Roberto M., 2016a). The stimulation of the motor neuron then causes a group of muscle fibers to contract or relax. Figure 2.2 represents how the muscle action potential can be produced at a neuromuscular junction where the axon terminal of the motor neuron join with the motor end plate of the muscle fiber (Gerard J. Tortora, 2010).



**Figure 2.2:** An illustration diagram represents an initiative of the muscle action potential at the neuromuscular junction. It starts from a coming nerve impulse from a motor neuron along a synaptic vesicle which triggers a neurotransmitter substance, so called acetylcholine (ACh), to diffuse across a synaptic cleft. After that ACh will activate an ion channel which especially allows sodium ions ( $Na^+$ ) to pass through it. The inflow of  $Na^+$  then causes a muscle action potential to further innervate the muscle fibers. (redraw from Gerard J. Tortora, 2010)

The force of muscle activities depend on the size and number of small muscle unit in the target muscles fibers. The field potential of the active muscle fibers generally has a triphasic form of brief duration between 3 to 15ms and an amplitude of 20 to 2000  $\mu\text{V}$ . The frequency of discharge can be varied from 6 to 30 per second (Webster, 2010). Figure 2.3 exemplifies a diagram of recording EMG signal with electrodes at the biceps and triceps muscle. These two muscles are an example of antagonist muscle, for when one contracts, another is relaxed. They are reciprocally innervated in order to avoid from working against each other (Bahill, 1981). An example of EMG signal can be illustrated as shown in Figure 2.4. If muscle fibers, e.g. biceps muscle, are contracted a high amplitude will be detected as indicated as "muscle activation". Meanwhile, the relaxation phase of the muscle group can be determined as the "baseline" interval.

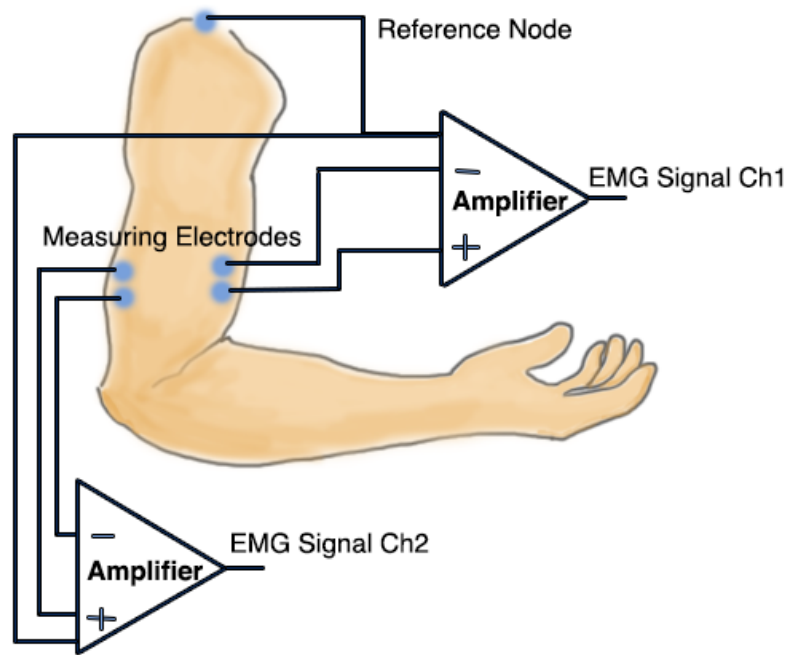
There are many factors that could affect the EMG signal, i.e. the thickness of subcutaneous tissue layer or the electrode size which affects the EMG amplitude and bandwidth. The accuracy of the signal also depends on the location of surface electrodes that should be placed related to the innervation zone in order to acquire the best result. However, in the real measurement, there would be a risk of having crosstalk from other muscles which is a signal detected over a target muscle but actually generated by another or nearby muscle (Roberto M., 2016c).

### 2.3 EMG Signal Recording

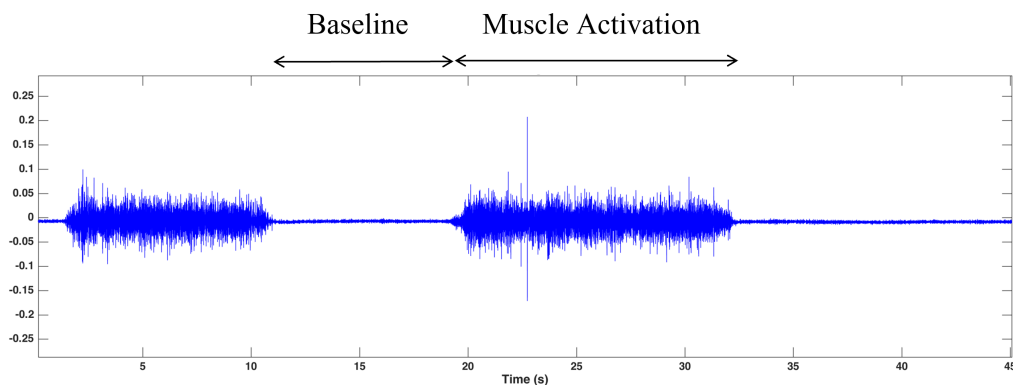
EMG signal can be recorded by applying needle electrodes (intramuscular recording) or surface electrodes (surface recording) to the target muscle and measure the electrical potential field of the muscle (Webster, 2004). The needle electrodes provide more precise measurement since the needle directly contact the innervated muscle, but this invasive technique needs an expert physician and it causes pain. Meanwhile, the surface electrode is a non-invasive method. Its result is not as accurate as the invasive technique due to body artifact caused by surface condition, the improper attached area or other factors. However, the non-invasive method is easy to handle and painless. With these benefits, the studied EMG system uses surface electrodes as a medium of measurement which can be divided into 2 categories, traditional surface electrode and textile electrode.

#### 2.3.1 Traditional Surface Electrode

The traditional surface electrode that used in this project is silver/silver chloride (Ag/AgCl) Latex free foam-pad electrodes which consist of a metal-disk electrode and non-irritating electrolyte gel (hydrogel) at the middle surrounded by adhesive tack on surface of foam pad (Webster, 2010), see Figure 2.5. The electrodes should be positioned by placing two electrodes across the target muscle fibers in order to measure a different potential of the muscle and placing the reference electrode at the bony prominence area such as ankle, wrist, or vertebra prominence C7 at the neck. The center to center distance between the two electrodes should be 2 cm and the size of the metal-disk electrode where the signal is picked up should not exceed



**Figure 2.3:** An illustration shows a signal diagram of an example of measuring the muscle signal at the biceps and triceps muscle by using surface EMG. The electrical potential is recorded by electrodes and amplified by an EMG signal processing unit. (redraw from Vogel, 2016)



**Figure 2.4:** An example of EMG signal in time-domain. The baseline interval indicates muscle relaxation and the muscle activation indicates muscle contraction or muscle activity level which can be evaluated by the amplitude (PlayRobots, 2018).

1 cm (Shimmer, 2017). There are two main cautions regarding the location of the electrodes. Firstly, the electrodes must not be located on or near the tendon of the muscle because the muscle fibers near the tendon become fewer which results in a lower amplitude of the EMG signal. Secondly, they must not be placed on the motor point of the muscle since there will be a summation of nerve signal in the recording EMG signal. Therefore, the inaccurate EMG signal might be acquired (Luca, 2002).

Advantages of using the traditional surface electrode are that it is a skin-contact electrode which results in less motion artifacts and the hydrogel also has a good

property of electron transport across the interface that will enhance the recording of the EMG signal (Tavakoli and Tang, 2017). On the other side, the traditional surface electrode is a consumable element due to the disposable material. So, it would be great if there is a solution for reusable electrode.



**Figure 2.5:** A traditional surface electrode that used in this project produced by Covidien<sup>TM</sup> (bio-medical, 2018).

### 2.3.2 Textile electrode

Textile-based electrode is a surface electrode made of garment with conductive yarn. It has advantages such as soft, flexible, washable, create less skin irritation to the user and be able to use for long-time monitoring in electrophysiological measurement (Guo, Eriksson, and Sandjö, 2014). Using the textile electrode, the patient may be able to do a movement easier and able to use the EMG biofeedback system anytime. Constrains of using textile electrode are the motion artifact and the accuracy of EMG signal since the textile electrode is not sticky contacted to the skin.

## 2.4 EMG signal Processing

The raw signal from EMG devices cannot be easily interpreted without processing. It has to be analyzed in order to extract the useful information before presenting back to the patient in comprehensible way.

### 2.4.1 Artifacts and Noises

In all signals acquired by a medical device, noise and artifact are considered as a disturbance of the system which affect the accuracy of the information carried by the signal. The information may be distorted or the system may be unreliable. Noise can be classify as external or internal. External Noise could be a power line interference (50Hz or 60Hz), noise from improper contact of electrode to the patient's skin or electromagnetic interference from wireless communication. On the other side, internal noise would be generated from the patient for instance motion artifacts, breathing artifact and ECG interference (Chatterjee and Miller, 2010). From those types of noise, a signal processing should be done in order to eliminate the noises and enhance the signal. A digital signal processing such as a digital filter could be fruitful to reduce the noise.



### 2.4.2 Digital filter

Digital Filter is a part of signal processing which mainly used for separate interference signal from the acquired signal and restore a distorted signal in some way (Smith, 1997a). As mentioned in the section 2.2 and 2.4.1, there are many factors that could affect EMG data. The acquired EMG data can be disturbed by noise from the environment and motion artifacts. Thus, EMG data should be filtered out the noises. The filter design depends on the specification of the filters which may rely on constraints of frequency response, unit or step response, filter type (e.g. FIR or IIR), and filter order (Hayes, 2012). In EMG application, the muscle activities are generally within the range of 15-500Hz (Luca, 1997 and Boxtel, 2001). Therefore, a high-pass filter with a 3dB cut-off frequency at 10-20 Hz and a low-pass filter with a 3dB cut-off frequency at 400-450Hz should be applied in order to extract the useful information and avoid loss of information. However, if the muscles under a very thin subcutaneous fat layer are investigated, some exceptions could be found. Then, the bandwidth can be increased more than 450Hz. Moreover, the movement artifact can be eliminated by applying a high-pass filter at 25-30 Hz for movement or gait analysis (Merletti, 2009, Sandsjö, 2004, Assambo and Burke, 2012).

In this thesis, BlackMan-Windowed-Sinc filters (Smith, 1997b) are used to separate the noise from the EMG signal. Advantages of using this type of the filter are stable, produce few overshoot, good stopband attenuation and easy to program. The ideal filter kernel of the sinc function, written in Equation 2.1, has a limitation of non-zero amplitude dropping to both negative and positive infinity and the abruptness of the truncated ends. A modification of Blackman window is then added to flatten the stopband, which is given by Equation 2.2. The multiplication of the sinc-filter kernel and the Blackman window will result in a perfect EMG signal for the system.

$$h[i] = \frac{\sin(2\pi f_c i)}{i\pi} \quad (2.1)$$

$$w[i] = 0.42 - 0.5\cos\left(\frac{2\pi i}{M}\right) + 0.08\cos\left(\frac{4\pi i}{M}\right) \quad (2.2)$$

Where  $M$  = length of filter, and  $f_c$  = cut-off frequency.

### 2.4.3 Signal Analysis

The EMG raw signal can be biphasic depends on the connection of the EMG channels and it has a sampling rate of 1024 Hz. The average rectified value (ARV) is computed in order to estimate the signal and make it becomes easier to be interpreted. A chunk of 256 ( $N=256$ ) data or 32 ( $N=32$ ) data of each second will be used for calculating the average of its absolute value according to Equation 2.3, where  $\bar{X}$  is the mean value of the chunk of interested data. The difference in the number of samples ( $N$ ) depends on the different applications which will be explained in section 4.3.2.

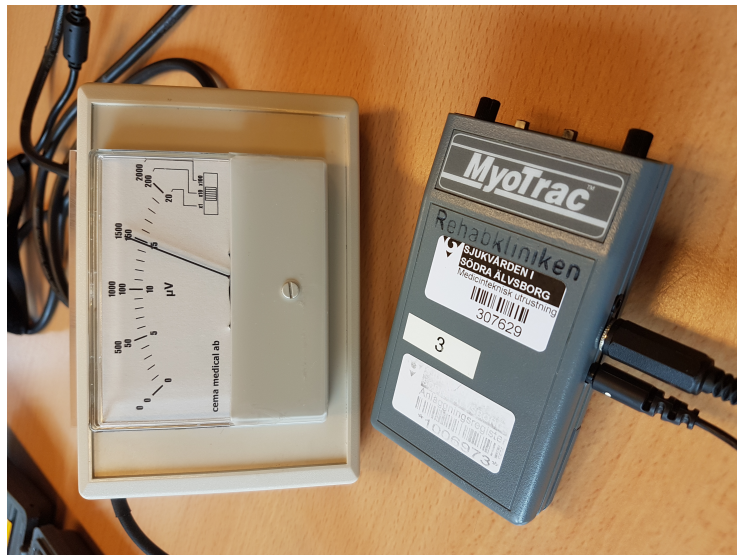
$$[h]ARV = \frac{1}{N} \sum_{i=1}^N |x_1 - \bar{X}| \quad (2.3)$$

### 2.5 EMG biofeedback system at SÄS

At SÄS, there are two traditional EMG biofeedback systems. One is a 1-channel EMG biofeedback system with audio feedback or visual feedback on an analog meter. Another system is a 2-channel EMG biofeedback system with a visual feedback on PC screen, including time-graph and bar.

#### 2.5.1 1-Channel EMG biofeedback system

The 1-channel system can measure only a muscle group. The system using at SÄS is shown in Figure 2.6. This system can result in audio feedback or visual feedback as an analog meter. Even though the electrode are applied at the muscle groups on both left and right areas of the shoulder, but the meter represents only one value which is the summation of those EMG values. The threshold and audio level can be manually adjusted at the hardware.



**Figure 2.6:** An illustration of 1-channel EMG biofeedback system

#### 2.5.2 2-Channel EMG biofeedback system

The 2-channel EMG biofeedback system is a system where 5 electrodes are used to measure the EMG signal. It requires 2 electrodes per channel and the last electrode act as a reference which should be applied at the bony area. The advantage of using 2-channel at SÄS is to monitor the muscle group on both left and right side for instance the muscle groups in the shoulder for posture correction. An amplifier unit using at SÄS is shown in Figure 2.7. It can adjust various parameter by this hardware i.e. gain, range, etc. The EMG signal will be processed by this hardware and result in visual feedback on the screen. There are two visual feedback option. One is bar chart and another one is time-graph. On the screen, a threshold of the muscle activation can be set. If the EMG value is higher than the threshold, the green color will be present. Otherwise, red color is shown.



**Figure 2.7:** An EMG amplifier unit using at SÄS. Gains can be adjusted by the white knob and threshold level can be selected by the red and black knob for channel A and channel B respectively. The blue knob is used for adjusting the delay in a unit of second(s).

### 2.5.3 Technology integration

Nowadays, technology becomes a big role in our life. The traditional biofeedback system which mostly consists of sensors, processor modules, and monitors or sound generator is not freely moved while performing a training especially for muscle training. With a wireless communication technology such as a smartphone, the possibility of having a wireless biofeedback system would be a good solution for EMG biofeedback. The availability of Bluetooth® in most of the smartphone would be a benefit in communicating with a processor hardware. But that means a wireless processor hardware must be available.

By searching for a processor module, there are many current available solution on the market. Most of the solutions provide Bluetooth® communication and Android™ device compatibility. These are a good sign of developing a mobile-base EMG biofeedback system.



# 3

## Developing process and methods

This chapter describes the project's development processes which are a combination of methodologies for researching and the development of the EMG system. The thesis is performed by designing criteria of the system based on the study visit and searching for an available commercial EMG amplifier. The work then proceeded by developing an Android™ application in order to present real-time EMG-signals in a biofeedback manner on a mobile device.

### 3.1 Interview and Study Visit

With a collaboration at SÄS, a study visit and interview at the hospital were performed to understand the traditional EMG biofeedback system at clinical. A discussion with occupational therapists who experience working with target patients has been done in order to learn the training process of the muscle relaxation training which is one of the EMG biofeedback system used at the rehabilitation clinic. The visit was done on the 26th of April 2018 at the Rehabilitation clinic at SÄS in Borås.

At the visit, a big picture of the EMG biofeedback system has been clearly presented as well as comments from the therapists who are familiar with the system was expressed. The system at SÄS consists of 3 main parts which are electrodes for measuring EMG signal, an EMG amplifier unit for acquiring and processing the signal, and an analog meter or a computer for monitoring the signal. It can be found that the traditional system is not easy to carry by the user due to the wire connections of the computer and the EMG device. It also does not allow the patients to have the system at their home. The patient needs to visit the clinic frequently to perform the rehabilitation training for a period of time.

### 3.2 Design Criteria and Flowchart

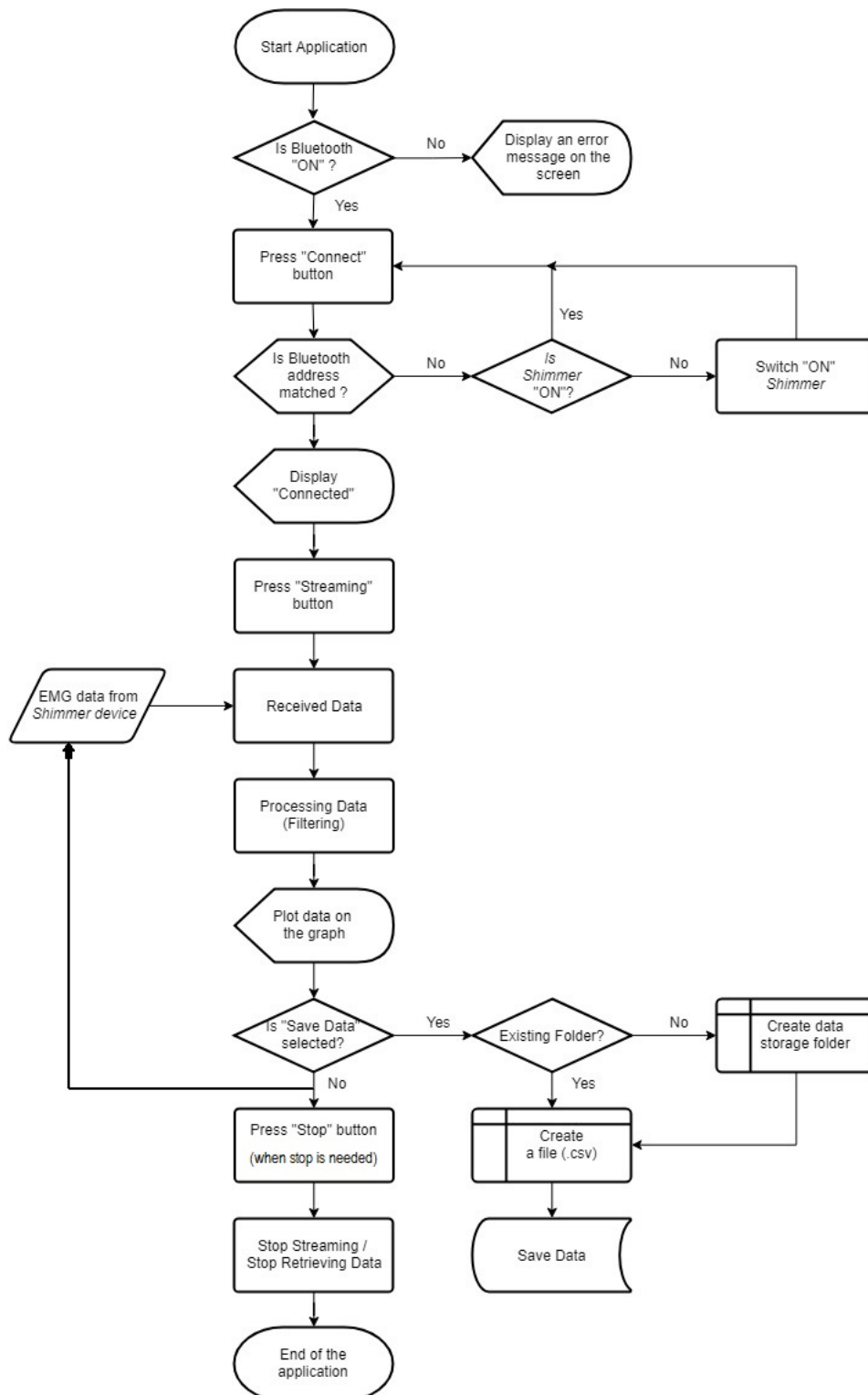
From the study visit, interview, and literature reviews, a design criteria of the developing system is then summarized. The implemented prototype of EMG biofeedback system is developed based on criteria :

- **simple**, i.e. easy to use and understand by low-technology usage, easy to set up and operate.
- **portable**, i.e. the system should be moved freely so the patient can be used at home environment (Koga et al., 2013).

### 3. Developing process and methods

- **practical**, i.e. reliable result / data.
- **real-time**, i.e. the EMG data can be acquired and visualized real-time.

The implemented Android software should work following a procedure simplified in Figure 3.1.



**Figure 3.1:** A flowchart shows the application flow of data.

### 3.3 EMG device - Shimmer3 EMG unit

By acquiring the EMG data, an electronic hardware for data acquisition is needed. Since the research aims to develop an EMG biofeedback system that applicable for home-based use and feasible to monitor the data on a mobile device, the important features would be a wireless module with a wireless communication. One solution is an EMG unit, called Shimmer3 EMG unit (Figure 3.2), produced by Shimmer<sup>TM</sup>. It is a 16-bit and 24-bit 2-channel system with Bluetooth<sup>®</sup> 4.0 communication and compatible with both PC and Android<sup>TM</sup> device. With a provided Java<sup>TM</sup> library function, Shimmer3 is selected to be the EMG measurement for this research.



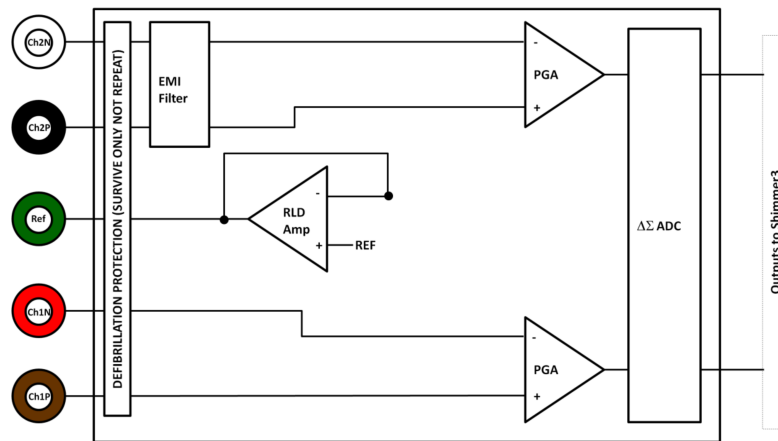
**Figure 3.2:** Shimmer3 EMG unit

Shimmer3 EMG unit is a five-wire EMG unit that can be configured to record the electrical potential of muscle activity. It consists of two channels input and one reference. The EMG Unit uses a low-power, multichannel analog front-end especially designed for biopotential measurements, consisting of delta-sigma analog-to-digital converters and programmable configurable gain amplifiers (1, 2, 3, 4, 6, 8, 12). It also provides a wireless solution for measuring EMG signal with an ergonomically valuable arrangement. The basic circuit diagram of the module is shown in Figure 3.3. It offers a Bluetooth<sup>®</sup> communication for connecting to mobile phone or other electronic devices. These advantages fit the research goal that aims to develop an EMG biofeedback system for a home-based training.

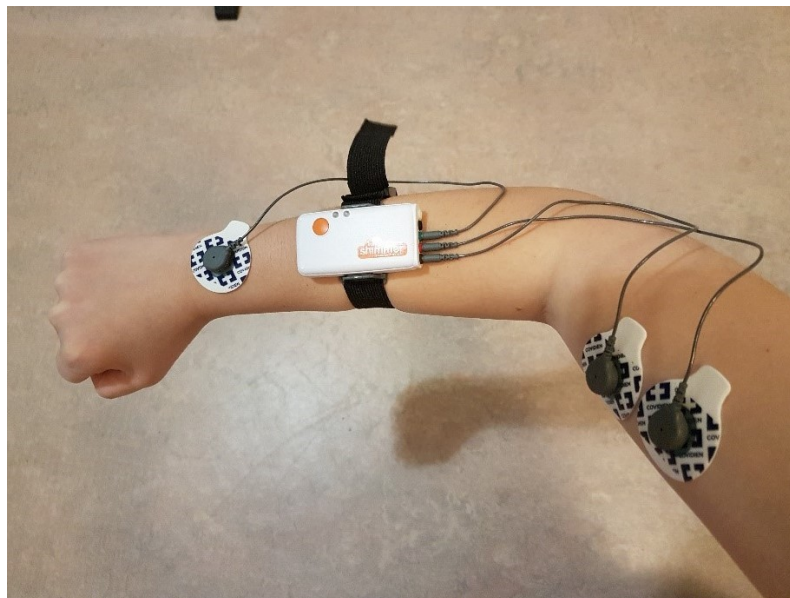
An example of using the EMG unit is to measure the muscle activity of biceps and triceps by connecting the 5 electrodes to channel 1 negative (Ch1N – brown), channel 1 positive (Ch1P – red), channel 2 negative (Ch2N – white), channel 2 positive (Ch2P – black), and reference (Ref – green). Then, the electrodes should be positioned on the body, for example on the right arm. Two electrodes should be placed across the muscle fibers of the target muscle, in this case are biceps and triceps. The reference electrode should be placed at an electrically neutral point of the body, as far away as possible from the target muscle. Bony prominences such as those at the elbow and the wrist are recommended Shimmer, 2017. Figure 3.4 shows an example of positioning electrodes.

### 3. Developing process and methods

---



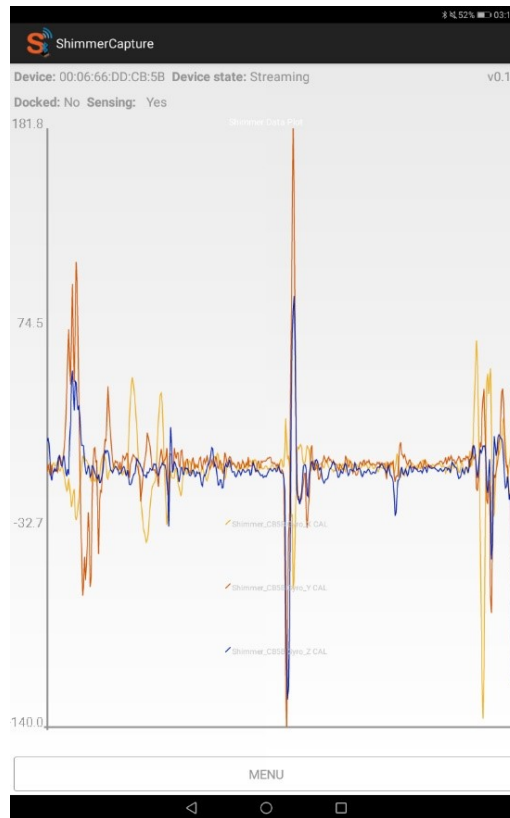
**Figure 3.3:** A block diagram of Shimmer3 EMG unit is illustrated.



**Figure 3.4:** An example of positioning of electrodes for EMG measurement.

In addition, Shimmer<sup>TM</sup> also has several available Android<sup>TM</sup> applications on Google<sup>TM</sup> Play which can be used for monitoring EMG data on a smartphone or a tablet. One of the available application that is used at the start of the project is called ShimmerCapture application. This application has simple useful features for EMG data acquisition such as streaming, logging and plotting the signals to graph. However, this application is not limited to monitoring EMG-signal, but also allows other kind of sensors included in Shimmer3 module to be explored such as ECG, gyroscope, etc. Several configurations, such as gain, sampling rate, resolution, are provided. Figure 3.5 represents the user interface of ShimmerCapture application.





**Figure 3.5:** A user interface of ShimmerCapture application

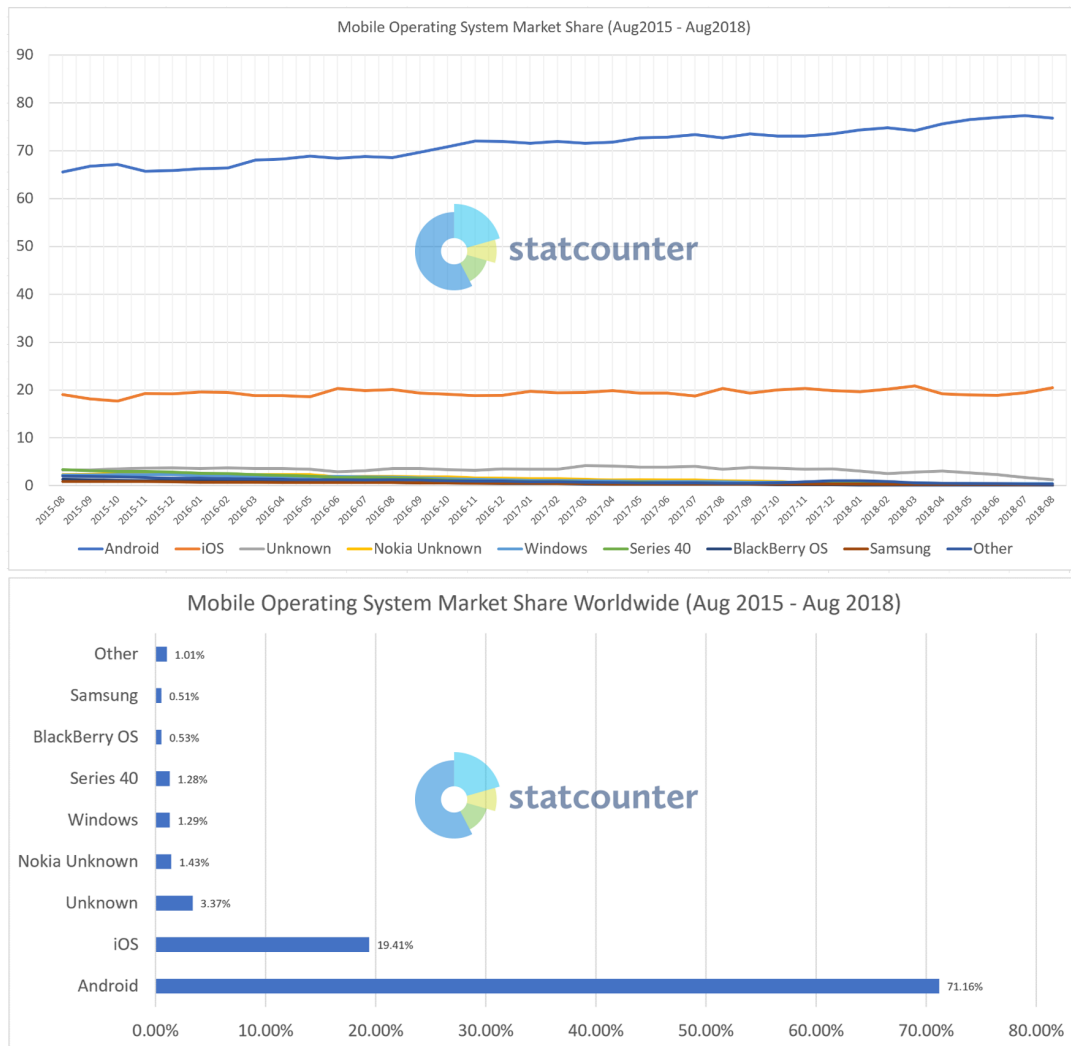
### 3.4 Android Development

One of the benefits of using the EMG unit by Shimmer<sup>TM</sup> is the supportive Android API and useful Java<sup>TM</sup> libraries for developers. The mobile application in this research is developed by using Android Studio program together with provided Shimmer's Java<sup>TM</sup> libraries. Android operating system also has the most worldwide market share for the mobile operating system according to the Statcounter<sup>®</sup> (Statcounter, 2017).

In order to develop a mobile application, there are several pre-requisites of hardware and software for using Android Studio program together with the Shimmer Java/Android API library which are:

- **Java Development Kit (JDK)** - a software environment used for developing Java applications and applets.
- **Android<sup>TM</sup> Software Development Kit (SDK)** - a software environment helps developers to create applications for the Android<sup>TM</sup> platform.
- **Android<sup>TM</sup> device** - a device which has a minimum Android API version of 17 or Android 4.2 (Jelly-Bean) and has any version of Bluetooth<sup>TM</sup> for wireless communication.
- **Shimmer Java/Android API** - supported library from Shimmer<sup>TM</sup>
- **Shimmer3 EMG unit** - an EMG device

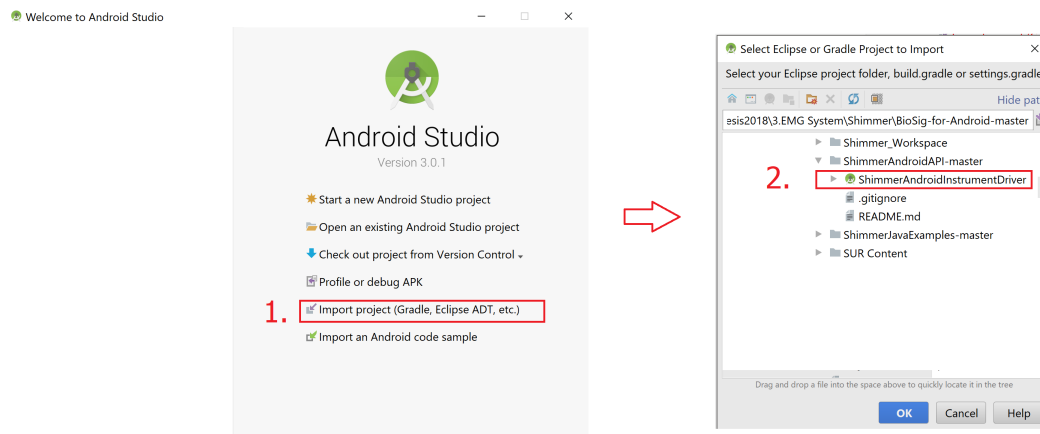
### 3. Developing process and methods



**Figure 3.6:** Statistics of mobile operating system market share in the worldwide are indicated as a line and bar graph. (data from StatCounter, 2017).

The application development then starts from import project. The supported Shimmer Java/Android API library, named *ShimmerAndroidInstrumentDriver*, is selected at this state, see Figure 3.7. If the application already exists and the modification is needed, then select 'Open an existing Android project' option. For a new project, a new module should be created. For existing projects, the change can be done by editing the code in the Java folder.

The development on Java programming language is an object-oriented programming which then the development of this project is based on ShimmerObject class. The acquired data from the Shimmer unit is collected in an object cluster via Bluetooth manager functioned by ShimmerBluetoothManagerAndroid class. Applying filters to the EMG signal in order to eliminate noise can be done by using a support from Filter class. Common functions that are used in the developing EMG biofeedback system can be listed as below:



**Figure 3.7:** At welcome page of Android Studio program, the development can be started from selecting 'Import project' -> 'ShimmerAndroidInstrumentDriver' as indicated in the red boxes,

- **btManager.connectShimmerThroughBTAddress(shimmerBtAdd);**  
- attempt to connect to the Android device via Bluetooth connection, where *shimmerBtAdd* is a bluetooth address of the Shimmer unit.
- **btManager.startStreaming(shimmerBtAdd);**  
- start streaming, acquiring data from the Shimmer unit.
- **shimmerDevice.stopStreaming();**  
- stop streaming over Bluetooth.
- **Collection<FormatCluster> f = objc.getCollectionOfFormatClusters(Configuration.Shimmer3.ObjectClusterSensorName.channelName);**  
**FormatCluster c = ((FormatCluster)ObjectCluster.returnFormatCluster(f, ChannelDetails.channelType));**  
**dataEMG = c.mData;**  
- retrieve data from a data structure.
- **filterType = new Filter(filterType, samplingRate, cutOffFrequency);**  
- initialize a filter
- **dataEMG = filterType.filterData(rawData);**  
- apply the filter to the data

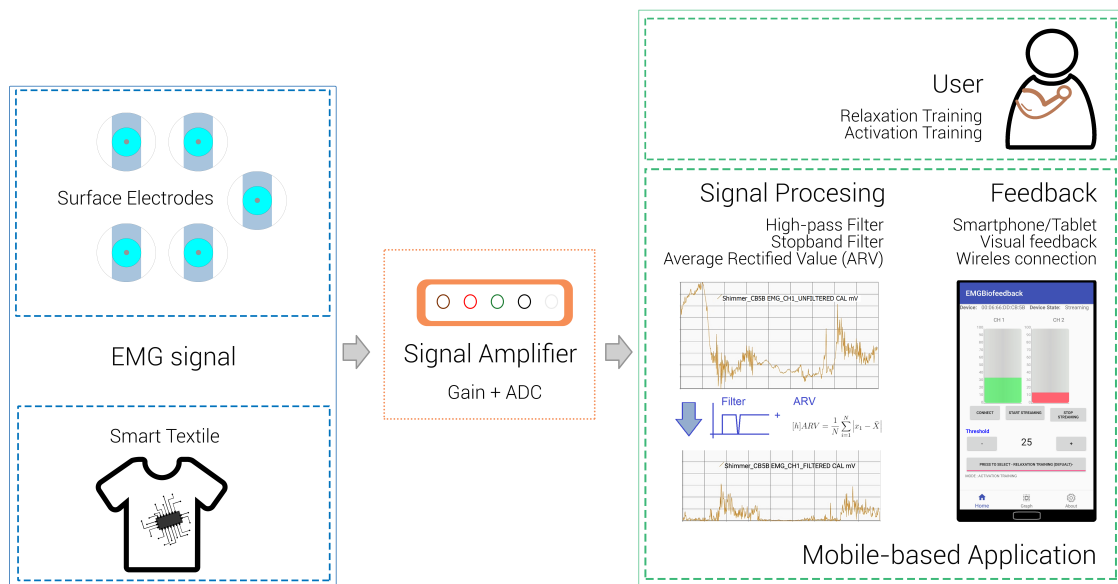


# 4

## Results

Since this thesis is a development of a system. This chapter will represent the EMG biofeedback system and its specification. Two scenarios that the system could be involved will be shown. Then the thesis work was evaluated by testing the system with subjects and the possibility in smart textile electrodes was investigated together with the Scan-to-Knit project at the University of Borås. The supported idea and other comments will be discussed in the next chapter.

### 4.1 System Overview



**Figure 4.1:** A block diagram shows the overview of the EMG biofeedback system.

An EMG biofeedback system should have at least 3 main parts as indicated in Figure 4.1. First part is a sensor that can be used to measure EMG signal. In this thesis, the non-invasive electrode is used and it can be represented as either conventional surface electrode (contact gel) or smart textile electrodes (garment). Second, the electrodes must be connected to an EMG module, so called Signal Amplifier in Figure 4.1, which can amplify the electronic signal and convert it to useful data. Lastly, the EMG signal need to be processed and fed back to the user in order to complete the biofeedback system. In this case, a mobile-based application such as smartphone or

tablet is a mean of feedback that uses to visualize the EMG information and can be developed to warning as an audio feedback.

### 4.2 Signal Amplification

According to the system overview, the second part of the system is a signal amplification. In this section, a signal processing by the available hardware is mentioned. A data processing by the mobile application will be discussed in the next section.

In Chapter 2, the basic background of EMG signal has been mentioned. The electronic signal generated by the muscle fiber has a very small voltage potential as low as microvolt ( $\mu V$ ). In order to represent the EMG signal, an amplifier is needed. In this project, a solution for EMG signal amplifier is the Shimmer3 EMG unit which has a default gain as 12 and a build-in delta-sigma analog-to-digital (ADC) converters. However, in order to present the data as clear as possible the gain of 100 is added in the software to amplify the acquired data and display on the graph view explicitly.

### 4.3 Signal Processing

The acquired EMG data from the Shimmer3 unit needs to be processed before feedback to the user via visual or audio medium. The EMG signal is then filtered and computed ARV.

#### 4.3.1 Filtering

The acquired EMG data need to be filtered to eliminate unwanted signal. Then, the application is programmed to filter the EMG signal by using a provided Java<sup>TM</sup> library named *Filter()*. In this thesis, a high-pass filter with a cut-off frequency at 10Hz is applied in order to eliminate the low frequency noise e.g. wandering baseline. After that a stopband filter with cut-off frequencies at 49 Hz and 51 Hz is applied to stop the 50 Hz powerline interference. The given filtering function is implemented based on the BlackMan-Windowed-Sinc Filter technique by Steven (Smith, 1997b).

#### 4.3.2 Signal Analysis

After filtering, the average rectified EMG signal was computed. In this thesis, there will be two different views developed on the smartphone which will be explained in the next section. In brief, the bar view which is mainly used by the patient will visualize ARV of the samples of each quarter of second or 256 data samples ( $N=256$ ). With this average, it can reduce the fluctuation of the data visualization which may occur from a quick move. Another view is a graph which is implemented for the specialist who can identify the EMG signal. The ARV of 32 data samples were estimated and plotted on the graph. This number of sample results in a proper rate of the graph visualization. The processed data will then be used for monitoring.

### 4.3.3 Real-time processing

The thesis work aims to develop a real-time system. Since the Shimmer3 unit offers Bluetooth® connection and most of mobile devices in the market provides Bluetooth® connection, so a real-time system is not too difficult to implement. The EMG data is processed to be presented on the smartphone as close as real-time. The EMG biofeedback application is programmed to set the frequency of the Shimmer3 unit to be 1024Hz. Then, a chunk of 256 samples or 32 samples, depends on views, will be further processed. The method of signal analysis will result in a small delay about a quarter of second or 0.25 seconds due to the data collection which is acceptable.

## 4.4 Feedback

EMG biofeedback can be represented in many way such as visual or audio. From many literature, there are many solution to express EMG biofeedback but this project chooses a smartphone Samsung Galaxy 7 as a mobile device to represent the EMG biofeedback because it could both visualize and make a sound feedback as well as the Android™ operating system has the most market share in the world. It also fit the requirement of the project that would like to transfer the clinical usage to the home environment.

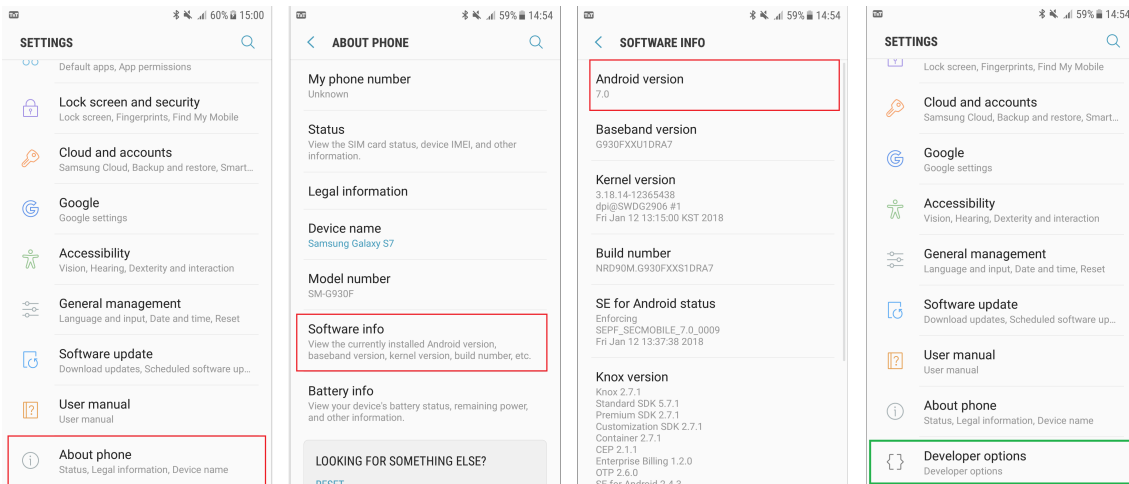
### 4.4.1 Application Specification

The developing mobile-application in this project is an Android-based application with Android API 17 or equivalent to Android 4.2 (Jelly-bean) or newer version. The smartphone or tablet must be able to use a Bluetooth® connection in order to connect the device to the EMG unit, in this case Shimmer3 EMG unit. The device should have at least 30MB free space for the application. At this stage, the application is a prototype and does not available on the market. So, the installation of the application must be done by debugging-mode. Using a debugging-mode, a developer option should be activated. For instance, this project uses Samsung Galaxy S7 as a mobile device, the developer option can be enabled by going to 'Setting' > 'About phone' > 'Software Info' > Android version, then click on Android version tab for 7 times (See Figure 4.2). After that an installation allowance from other sources must be turned on as shown in Figure 4.3.

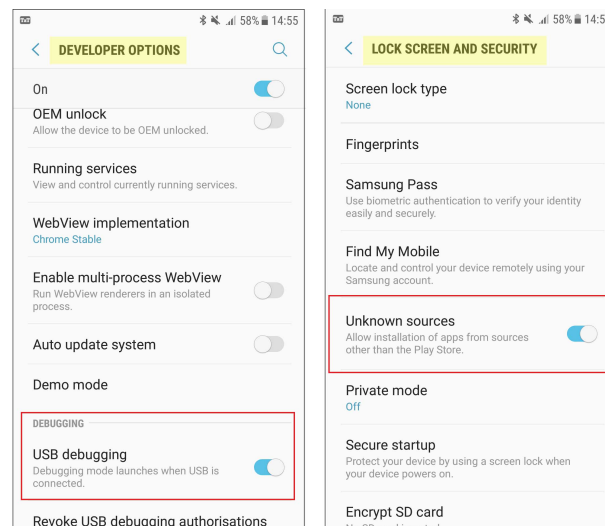
### 4.4.2 Application Programming

Since the application is Android-based system, the Android Studio version 3.0 is used for programming the developing application. It allows developers to design the user interface and program the application by Java™ programming language. The provided Java™ libraries from Shimmer™ Engineering are used together with the Shimmer Java/Android API User Manual revision 2.11. Some of the significant implemented functions and source codes are provided in Appendix.

## 4. Results



**Figure 4.2:** An illustration shows how to enable Developer Options for Samsung Galaxy 7



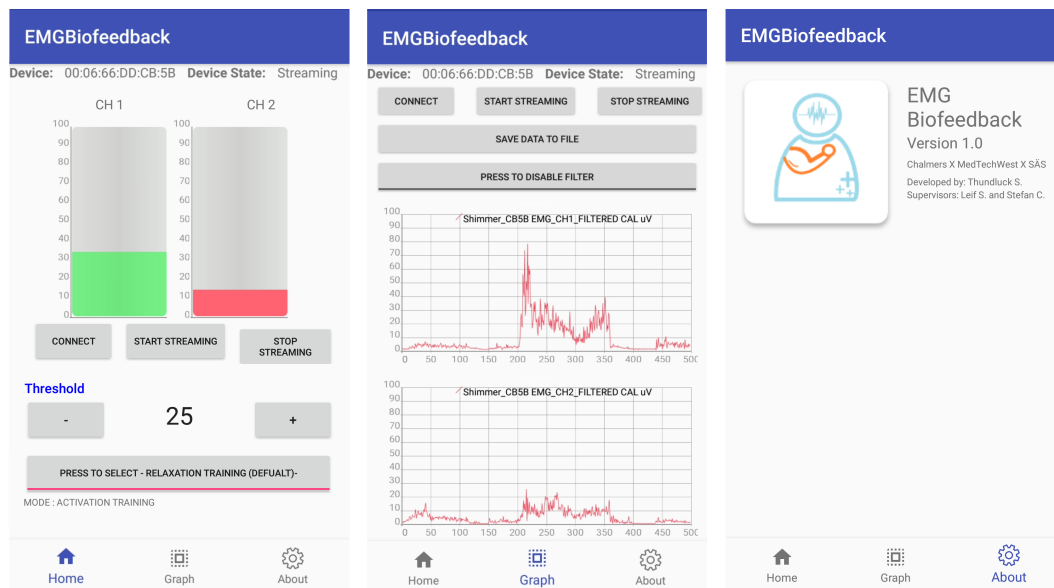
**Figure 4.3:** An illustration shows how to turn on a debugging mode and allow the installation from other source.

The application uses the fundamental of a bottom navigation bar of the Android development which consists of three destinations. In principle, the bottom navigation bar is ergonomic and consistent (*Bottom navigation* 2018). Using this design, the user is easy to reach the navigation on a handheld mobile device.

### 4.4.3 Data Presenting and UI

Figure 4.4 shows screenshots of the user interface of the EMG biofeedback application. There are three navigation which are Home, Graph and About. The Home view is mainly implemented for the patient or the user who does not familiar with the EMG signal. The color indication is easy to interpret. In the graph mode, the EMG signal can be clearly seen and used for further analysis. A specialist is needed to interpret the graph. The last view is an information about the application.





**Figure 4.4:** Illustrations shows a design of the user interface which can present EMG data as bars or graph.

For the bar view, the EMG data is presented as 2 bars, for 2 channels. A threshold value can be adjusted as well as an option for different training modes which can be selected by toggling the button indicated 'PRESS TO SELECT - '. These two options can be selected while the device is streaming. There will be a text indicates the mode of training under the mode button. The threshold can be adjusted by pressing + or - button to increase or decrease the threshold value respectively.

For the graph view, the application allows the user to see the data as a time-series graph with either processed data (default) or unprocessed data by toggling the button indicated 'PRESS TO DISABLE FILTER'. The raw data can be saved by pressing 'SAVE DATA TO FILE' button for further investigation. These two buttons are able to choose while the device is streaming. The save data button is only designed for enabling saving data but not function for stopping saving. To stop saving data, a stop streaming button must be used. The option of disable filter is designed for showing a raw data on the graph in order to evaluate the filter as well as for investigating any significant detail in the raw signal.

To use the application, there are four main steps.

**Step1** - Turn on Bluetooth

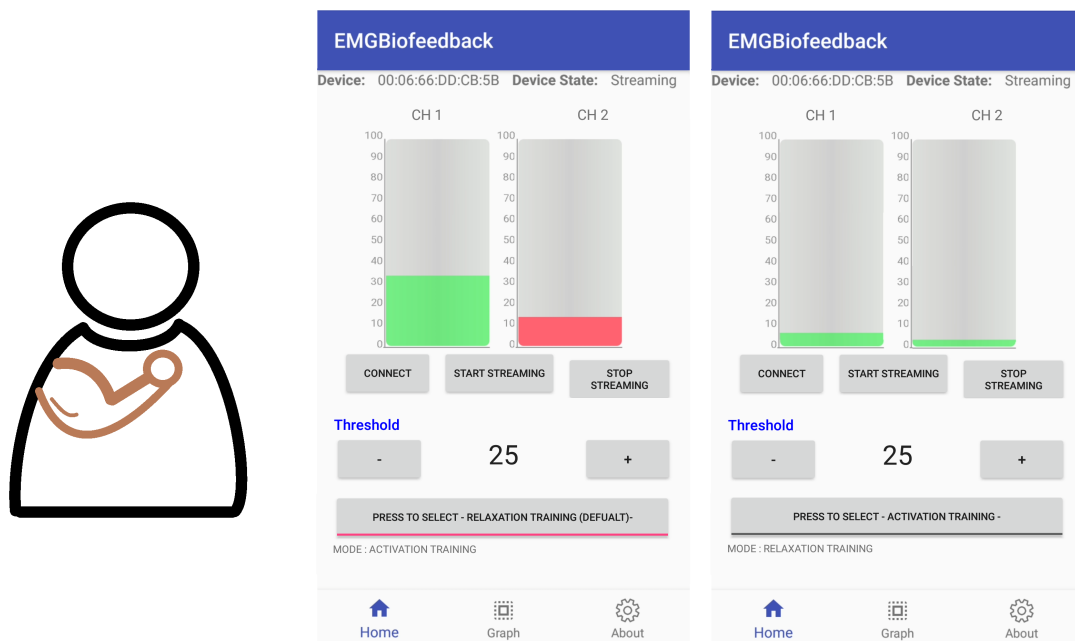
**Step2** - Select 'CONNECT' button to attempt connecting to the Shimmer unit. If the connection fails, select 'CONNECT' again until the device status shows 'Connected'. If the device is connected, a small pop-up box will appear at the bottom of the screen and state that the device is ready to stream

**Step3** - Select 'START STREAMING' to acquire data.

**Step4** - Select 'STOP STREAMING' to stop acquiring data.

## 4.5 Scenarios

Two Examples of using the system EMG biofeedback system are training of muscle relaxation and muscle activation which in this case can be selected by pressing the toggle button 'PRESS TO SELECT - ' mode as shown in Figure 4.5. By setting a threshold, the patient can train themselves to relax or strengthen their muscles and investigate their movement by the bars or graph visualization. There will be a green color means successfully done the muscle training while a red color indicates a fault in muscle training.



**Figure 4.5:** Screenshots of two scenarios of muscle relaxation training and strengthen training or activation training mode with a threshold of 25 are presented. The left bar represents EMG values from Channel 1 and the right bar represents EMG values from Channel 2.

### 4.5.1 Relaxation training

The relaxation training can be applied to the patients who has a problem of muscle relaxing. The patients can train themselves by this EMG biofeedback system. One example of the relaxation training is a training at shoulder-neck area. First, four electrodes should be attached to the muscle area on left and right shoulder, two electrodes for each side. Second, a reference electrode should be attached to the vertebra prominence of C7. Then, all electrodes should be connected to the EMG unit. It can be assumed that Channel 1 is the left side and Channel 2 is the right side. A threshold should be adjusted before the observation by clicking on decrease or increase button. From the rightmost of Figure 4.5, it can be assumed that the left bar is an EMG value from EMG unit's Channel 1 which assumed to represent left shoulder and the right bar is an EMG value from EMG unit's Channel 2 which

assumed to represent right shoulder. An example of the result in the Figure 4.5 is a relaxation of both left and right muscle groups of shoulder which could be a posture of put your arm freely downward and sit comfortably on a chair. The left and right muscle group should be relaxed which result in green color as it is lower than the threshold. If the patient has a problem of muscle relaxation, for example an office syndrome patient, it may result in red color indicated on the bar.

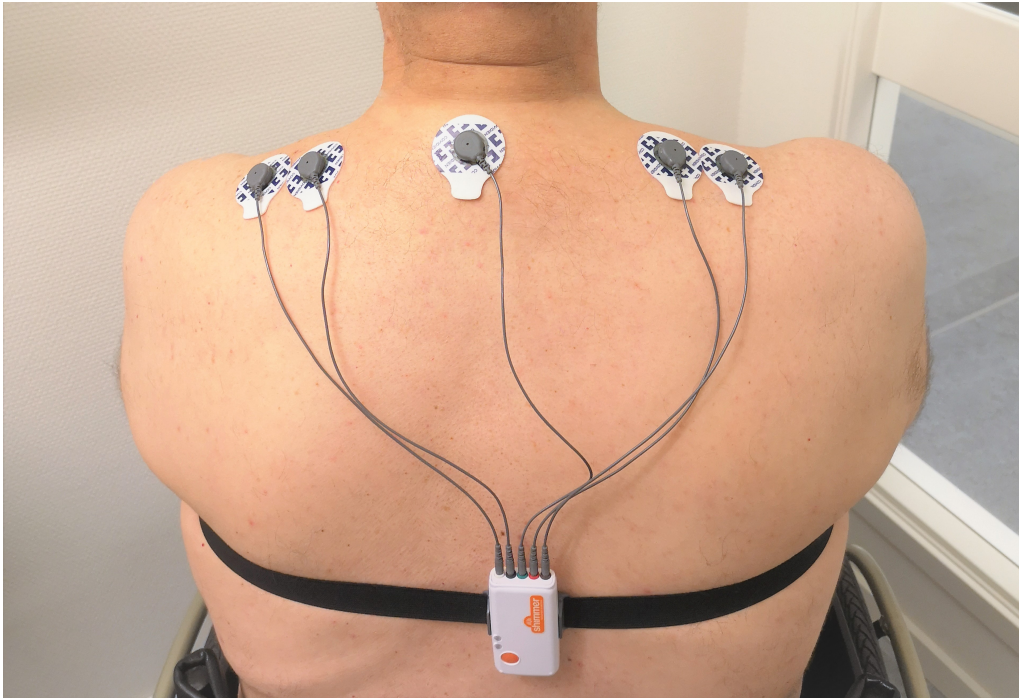
### 4.5.2 Strengthen training

An example of the strengthen training can be applied to the patient who has a muscle contraction problem e.g. after stroke patient. It could be a training at the biceps and triceps muscles. Two electrodes should be attached to the biceps muscle area and another two electrodes should be attached at the triceps muscle area. Then, a reference electrode should be attached to the elbow or any bony part. After that the electrodes should be connected to the EMG unit. It can be assumed that Channel 1 is the biceps muscle and Channel 2 is the triceps muscle. An example of muscle strengthening training is a posture of grasp the hand and curl the arm as a drawing in the most left of the Figure 4.5. A setting of threshold level at 25 is applied. The red color of the bars shows EMG values that are lower than the threshold which mean that the muscle is not strength enough. On the other hand, the green color represents a successful strengthen muscle over the target threshold.

## 4.6 System Evaluation

The EMG biofeedback system has been evaluated by testing on a volunteer subject. The system was used for investigating the muscle activity at the shoulder-neck area. The electrodes were placed on the muscle fiber on both left and right shoulders of the subject and the reference electrode was placed at the vertebra prominence C7 as shown in Figure 4.6. The evaluation was done by asking the subject to use the application with advice from the creator. The subject was guided to start turn on Bluetooth, connect to the Shimmer unit and start streaming data. The subject was asked to stretch out his left arm and right arm separately alternately to activate the muscle group at the shoulder area and watched the result on the screen for a period of time. The test was done in the graph view first. Then, the same test was moved to the bar view. The subject can successfully perceive the feedback from both graph view and bar view. After the test, the subject was allowed to play around the application by himself in order to evaluate the user interface. At the end, the subject was asked for a cognition on the presented results. The subject answered that the system works well and allows him to train his muscle with a good perceivable presentation.

There were also evaluations on other two subjects at extensor muscle group. Two electrodes were place across the extensor muscle area and the reference electrode was placed at the wrist as shown in Figure 4.7, leftmost. The subjects were asked to bend back their hand in order to activate the extensor muscle group and see the result on the smartphone. These two subject expressed that the system is easy to



**Figure 4.6:** The electrode locations and system connection for shoulder’s muscle activity investigation are presented.

use and it successfully shows the feedback of their movement. It should be valuable for the muscle training. Another evaluation was a usage of the system with the bicep muscle group for strengthening training, see Figure 4.7, rightmost. For this evaluation, the subject was asked to raise her arm against the specialist’s hand which then result in activating the bicep muscle. The subject can perceive the feedback by the graph and bar visualization on the smartphone.

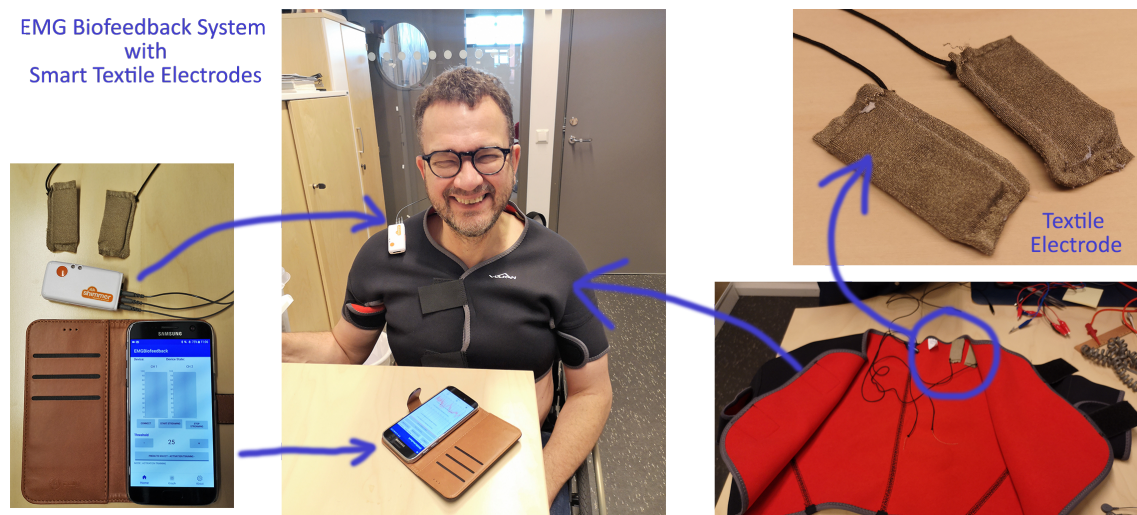


**Figure 4.7:** Evaluations of the system have been done by three subjects. They are all healthy adults.

All evaluations were done by the healthy adult persons. They were diversified, i.e. age, gender, nationality and experience in using the system. The evaluations indicate that the system is practical, portable and easy to use by any experience level.

## 4.7 Smart Textile Electrode Evaluation

An evaluation of the system with textile electrodes was done by a support from Scan-to-knit project at Textile Fashion Center at University of Borås. The system was evaluated by using textile electrodes instead of the conventional electrodes. Two textile electrodes were attached to an elastic jacket at the left shoulder area as shown in Figure 4.8 and the reference electrode was located at the vertebra prominence C7. The subject was guided to use the application by connecting the EMG device with the smartphone and streaming data. The subject was asked to sit in a relax posture by putting his arms down. Then he was asked to stretch his left arm to activate the muscle group at the left part of his shoulder and lay it down to relax the muscle group for a couple of times. The EMG signal was presented on the screen of the smartphone. The acquired signal from the textile electrodes was as good quality as the signal that was acquired from the contact gel electrodes. The subject also confirmed that the system works well as it works with the traditional electrodes. The subject was able to perceive his muscle activities from the visual feedback of the system.



**Figure 4.8:** The evaluation of the EMG biofeedback system with smart textile electrodes was done at the Textile Fashion Center in Borås.



# 5

## Discussion

The thesis is purposed to develop an EMG biofeedback system which can transfer the clinical application to the home environment. From literature reviews and study visit at SÄS, understanding of the system, problems and ideas as well as interests in the development of the system were fully expressed. It is found that the main problems are the limitation in movement of the system and the need of using the system at the patient's home. Since most of the EMG biofeedback applications are used for muscle training, it is better for the patient to be able to train at their home circumstance in order to enhance the restoration. Therefore, an integration of the wireless communication as Bluetooth™ has been involved to make the system be portable. A wireless EMG unit was then chosen to be an EMG signal acquisition and a mobile device as smartphone was selected to be a solution for EMG signal monitoring. During the developing process, an idea of color identification on the bar visualization is established for easier comprehension. Most of the thesis work was contributed to the software development which is implemented on Android™ platform. The signal processing and analyzing were digitally done in the software and the processed data was then presented on Android™ devices. At the end, the developed system was evaluated by four volunteer healthy adults which results in a successful EMG biofeedback system with portable, reliable and practical qualifications. This solution can fulfill the need of transfer the clinical system to the home environment. It can also increase the capability of rehabilitation of the patient due to feasible to access the system anywhere anytime. However, the developed system is a prototype and the developed software is not available in the market. It has to be developed in the future work before launch.

### 5.1 EMG unit (hardware)

To develop the EMG biofeedback system, an EMG device is needed. From the result, the Shimmer3 EMG Unit is used because it has 2-channel of electrodes and it comes together with the Android API solution, a support for Android™ development. By using 2-channel EMG system, it allows the system to be applied to bilateral parts of the body (i.e. left and right side training simultaneously). The Android API solution is also a good start on developing a Android™ application. Moreover, the Shimmer3 unit also consists of other kind of useful sensors such as IMU, ECG, and GSR which can be a solution for EMG biofeedback in the future work.

However, after using the Shimmer unit for all the development. Some disadvantages of the system are found. The provided cables for the electrode connection is too small and too short. A usage of the system where the target muscle groups are far from each other or far from the reference point is hard to be done. With the small cable, there is a risk of broken cable which is very difficult to investigate. The elastic strap that used to hold the Shimmer unit is also not firm. The Shimmer unit can be dropped off if the strap is too tight.

### 5.2 EMG biofeedback System

Regarding the data presented, many ideas were encourage by the EMG biofeedback system at SÄS. The bar graph is available on the traditional system at SÄS but SÄS has a difficulty in wireless accessible. The SÄS system needs to be wired by the computer power which is not comfortable to do a movement training. With the wireless communication technology, this research proposes a Bluetooth communication to the system. By using the wireless EMG Device together with a smartphone, the muscle training can be done easier. The system become smaller in size. The EMG device can be attached to any part of body by a stretch strap and the user can access to the biofeedback though a smartphone with an available software on its. There is one problem that the research's team found on the SÄS's data presenting. It is a color of the bar which contrasts to the general meaning. It is the case that a user performs muscle relaxation training and the EMG value should be lower than the threshold. Then, the bar should be a green color but in the existing program, the bar is red when the EMG value is lower than the threshold. This might make a confusion. Another point is the 1-channel system of SÄS which normally gives feedback as a sound. The device always alarm when the user has a muscle relaxation but the alarm itself can make the user be stressed. These are two main points that can be inspected from the visit at the hospital.

Therefore, the developed EMG biofeedback system allows the user to choose the mode of training and the colors will be changed regarding the selected mode. The green color means a successful in training and the red color means a failure in the training for both relaxation and activation training. So, the problem of color recognition can be solved by this system. However, this system is mainly developed for visual feedback. The problem in audio system is still unsolved.

### 5.3 Application of usage

The EMG biofeedback system of this project is developed for the feasibility in utilizing for both muscle relaxation and muscle stress training. The system also introduces the mobility of the treatment by using Bluetooth wireless communication of the mobile device.

According to the test during the development and the evaluation, the system can be used for different muscle groups i.e bicep, tricep, and shoulder-neck muscle group.



The volunteer subjects were diversifying. Some of them has no background using the EMG biofeedback system. The evaluation can confirm that the system is stable, reliable, and also easy to use by every level of experience. The availability of saving also benefits for the specialist in further study of the patient's result.

## 5.4 Feasibility in Smart Textile Electrode

The result presents the feasibility in using smart textile electrodes with the developed EMG biofeedback system. In the evaluation, the reference electrode was a conventional contact gel electrode instead of the textile electrode because of a limited number of the textile electrode at the moment. However, the Scan-to-knit project has ensured that there is no difference in the result if the reference electrode is a textile electrode. This guarantees that the textile-based electrode can be used with the developed system.

One obstacle would be the elastic jacket which is not well fitted to the subject. So, the electrodes were not attached to the body skin properly. A tape needed to fasten the electrodes to the skin during the test. In addition, there was an assumption that the humid electrode results in a better signal acquisition. This could be proven in the future work of the Scan-to-knit project.

From the evaluation, it is possible to produce a garment with textile electrodes fixed at the different position of the body for the different muscle group training. This solution allows the patient to do the training without the occupational therapist. Therefore, it additionally supports the idea of bringing the clinical system to the home environment.

## 5.5 Future Work

Even though the system has been evaluated by subjects but the system itself is still a prototype of the EMG biofeedback system. There are several outstanding points that could be further developed in order to enhance the capability of the system.

### 5.5.1 Function

At this moment, the three different views of the user interface is not connected to each other. The change in the view results in a reset of the screen. Even though the device is still connected to the EMG unit but the information on the screen is disappeared and no status is shown after changing the view. To solve this problem, fragment bundle should be added in the software.

Moreover, the data presented on the bar is still flickering. A median average window can be applied in order to flatten the overshoot of the data due to a twitch of the muscle. In addition, an option for retrieving the saved data and present on the

application could be done. So, the therapist or even the patient can recall for the result that has been recorded without open the file on the computer.

### **5.5.2 Application Design**

A review from one of the subjects is that the UI design is easy to understand and provide feedback on the action made by the user. Other subjects also agree that the UI is easy to handle. One problem would be a UI on tablet devices which is not proper in the ratio of the screen. The UI is not well implemented for the tablet use at this moment. A screen rotation may also cause a problem which UI is not fit to the screen. The application should be redesigned based on the UX/UI design criteria including the ergonomic standard for instance a comprehensive of the button.

# 6

## Conclusion

It can be concluded that the developed EMG biofeedback system in this project has solved most of the problem from the system at SÄS which are the possibility of moving the clinical treatment to home environment with the use of mobile device and the better recognition in different training mode of relaxation and activation. The system can be used freely as it has no wire connection between the monitor device, i.e. Android<sup>TM</sup> smartphone, and the EMG unit, i.e. Shimmer3 EMG unit. With the evaluation results, the system can be proved that it is stable, reliable and portable. It also answers the research questions:

- **Is it possible to transfer a raw EMG signal acquired by an EMG device via wireless communication, i.e. Bluetooth<sup>®</sup>, and visualize it on a mobile device in real-time?**

Yes, it is possible. From the evaluation results, the raw EMG signal acquired by the Shimmer<sup>TM</sup> EMG device can be transferred via Bluetooth<sup>®</sup> and visualized on the mobile-based device in real-time. The communication bandwidth and real-time data management are not issues.

- **Can the real-time EMG signal be presented as biofeedback information on a mobile-based device?**

Yes, the real-time EMG signal can be presented as biofeedback information, i.e. bar graphs, on the mobile device. However, the signal pre-processing and signal analysis are needed to make biofeedback information. There is no problem related to the mobile device capacity in order to present the processed data in real-time.

- **Can the mobile-based EMG-biofeedback system be used with a smart textile technology targeting specific muscles or muscle groups?**

Yes, the developed system can be used with the smart textile electrodes. The smart textile technology can be a solution for recording EMG-signal by the patient him/herself as well as performing the training outside the clinic.

- **Can EMG-biofeedback training be transferred from the clinic to the patient's home environment?**

Yes, EMG biofeedback training can be done at the patient's home by using the developed system in this project.

However, the developed system is a prototype and mainly express the idea of the solution. So, the system can be developed more in an aspect of mobile application, ergonomics usage and more. The EMG device can be further developed as well in order to get lower cost of the hardware.



# Bibliography

- Assambo, Cédric and Martin J. Burke (2012). “Low-Frequency Response and the Skin-Electrode Interface in Dry-Electrode Electrocardiography”. In: *Advances in Electrocardiograms*. Ed. by Richard M. Millis. Rijeka: IntechOpen. Chap. 2. DOI: 10.5772/22777. URL: <https://doi.org/10.5772/22777>.
- Bahill, A. Terry (1981). “Chapter 4 : Close loop”. In: *BIOENGINEERING — Biomedical, Medical, and Clinical Engineering*. Englewood Cliffs, N.J: Prentice-Hall, Inc.
- bio-medical (2018). *Covidien Kendall Disposable Surface EMG/ECG/EKG Electrodes 1 3/8" (35mm) 50pkg*. URL: <https://bio-medical.com/covidien-kendall-disposable-surface-emg-ecg-ekg-electrodes-1-3-8-35mm-50pkg.html>.
- Bottom navigation (2018). URL: <https://material.io/design/components/bottom-navigation.html#implementation>.
- Boxtel, A. (2001). “Optimal signal bandwidth for the recording of surface EMG activity of facial, jaw, oral, and neck muscles”. In: *Psychophysiology* 38.1, pp. 22–34. DOI: 10.1111/1469-8986.3810022.
- Chatterjee, Shakti and Aubert Miller (2010). *Biomedical instrumentation systems*. Delmar Cengage Learning.
- Ehrenborg, Cecilia and Birgitha Archenholtz (2010). “Is surface EMG biofeedback an effective training method for persons with neck and shoulder complaints after whiplash-associated disorders concerning activities of daily living and pain — a randomized controlled trial”. In: *Clinical Rehabilitation* 24.8, pp. 715–726. DOI: 10.1177/0269215510362325.
- Freepik (2018). *Correct and incorrect postures*. URL: <http://www.freepik.com>.
- Gaffney, Brecca M. et al. (2015). “Evaluation of Novel EMG Biofeedback for Postural Correction During Computer Use”. In: *Applied Psychophysiology and Biofeedback* 41.2, pp. 181–189. DOI: 10.1007/s10484-015-9328-3.
- Gerard J. Tortora, Bryan Derrickson (2010). “Chapter 8 : The Muscular System”. In: *Introduction to the Human body - the essentials of anatomy and physiology*. 8th ed. John Wiley & Sons, Inc., pp. 184–193.
- Giggins, Oonagh M. et al. (2013). “Biofeedback in rehabilitation”. In: *Journal of NeuroEngineering and Rehabilitation* 10.1, p. 60. DOI: 10.1186/1743-0003-10-60. URL: <https://doi.org/10.1186/1743-0003-10-60>.
- Guo, Li, Siw Eriksson, and Leif Sandjö (2014). *Surface Resistivity of Textile-Based Electrodes*.
- Hayes, Monson H (2012). “Chapter 9 : Filter Design”. In: *Schaum's outlines Digital Signal Processing*. 2nd ed. McGraw-Hill, pp. 358–379.

- Horowitz, Sala (2006). “Biofeedback Applications: A Survey of Clinical Research”. In: 12, pp. 275–281.
- Isaac, Chris et al. (2014). *Mobile data analytics: not just for consumers any more*. URL: <https://www.pwc.com/id/en/publications/assets/ticepublications/data-analytics.pdf>.
- Jayarathne, M. et al. (2015). “EMG based biofeedback system using a virtual reality method”. In: *2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS)*, pp. 111–116. DOI: 10.1109/ICIINFS.2015.7398995.
- Jovanov, I. M. and D. B. Popović (2014). “EMG based biofeedback with the smarting system”. In: *12th Symposium on Neural Network Applications in Electrical Engineering (NEUREL)*, pp. 163–166. DOI: 10.1109/NEUREL.2014.7011494.
- Kim, Ju-Hong (2017). “The effects of training using EMG biofeedback on stroke patients upper extremity functions”. In: *Journal of Physical Therapy Science* 29.6, pp. 1085–1088. DOI: 10.1589/jpts.29.1085.
- Koga, Kenichiro et al. (2013). “Portable biological signal measurement system for biofeedback and experiment for functional assessment”. In: pp. 412–416. ISBN: 978-89-93215-05-2. DOI: 10.1109/ICCAS.2013.6704003.
- Levitas, Danielle (2013). *Always Connected How Smartphones And Social Keep Us Engaged*.
- Luca, Carlo J. De (1997). “The Use of Surface Electromyography in Biomechanics”. In: *Journal of Applied Biomechanics* 13.2, pp. 135–163. DOI: 10.1123/jab.13.2.135.
- (2002). *SURFACE ELECTROMYOGRAPHY: DETECTION AND RECORDING*.
- Merletti, Roberto et al. (2009). “Technology and instrumentation for detection and conditioning of the surface electromyographic signal: State of the art”. In: 24 (2), pp. 122–134. DOI: 10.1016/j.clinbiomech.2008.08.006.
- PlayRobots (2018). *Bio-Potential Measurement Basics*. URL: <http://www.playrobots.co.uk/bio-potential-monitoring/bio-potential-measurement-basics/>.
- Richer, R. et al. (2014). “Real-Time ECG and EMG Analysis for Biking Using Android-Based Mobile Devices”. In: pp. 104–108. ISSN: 2376-8886. DOI: 10.1109/BSN.2014.20.
- Roberto M., DarioF. (2016a). “Chapter 1 : Phusiology of muscle activation and force GENERATION”. In: *SURFACE ELECTROMYOGRAPHY Physiology, Engineering, and Applications*. John Wiley & Sons, Inc., pp. 1–29.
- (2016b). “Chapter 18 : Surface EMG Biofeedback”. In: *SURFACE ELECTROMYOGRAPHY Physiology, Engineering, and Applications*. John Wiley & Sons, Inc., pp. 485–500.
- (2016c). “Chapter 2 : Biophysics of the generation of EMG signals”. In: *SURFACE ELECTROMYOGRAPHY Physiology, Engineering, and Applications*. John Wiley & Sons, Inc., pp. 30–53.
- Sandsjö, Leif (2004). *Ambulatory monitoring and analysis of surface electromyographic signals in ergonomic field studies : methods and applications*. Doktoravhandlingar vid Chalmers tekniska högskola: Ny serie 2127. Göteborg : Chalmers tekniska högsk., 2004. ISBN: 91-7291-445-9. URL: <http://proxy.lib.chalmers>.

- se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=cat06296a&AN=clc.b1294819&site=eds-live&scope=site.
- Shimmer (2017). *EMG User Guide Revision 1.12*.
- Smartphone daily usage time worldwide 2017 | Statistic (2018). URL: <https://www.statista.com/statistics/781692/worldwide-daily-time-spent-on-smartphone/> (visited on 08/21/2018).
- Smith, Steven W. (1997a). "Chapter 14 : Introduction to Digital Filters". In: *The Scientist and Engineer's Guide to Digital Signal Processing*. San Diego, CA, USA: California Technical Publishing, pp. 261–276.
- (1997b). "Chapter 16 : Windowed-sinc filters". In: *The Scientist and Engineer's Guide to Digital Signal Processing*. San Diego, CA, USA: California Technical Publishing, pp. 285–296.
- StatCounter (2017). *Mobile operating system market share worldwide*. URL: <http://gs.statcounter.com/os-market-share/mobile/worldwide> (visited on 09/10/2018).
- Tavakoli, Javad and Youhong Tang (2017). "Hydrogel Based Sensors for Biomedical Applications: An Updated Review". In: *Polymers* 9.12, p. 364. DOI: 10.3390/polym9080364.
- Vogel, Richard (2016). *EMG*. URL: <https://neurologiclabs.com/neuromonitoring/emg/>.
- Webster, John G. (2004). "Chapter 7 : Nervous System". In: *Bioinstrumentation*. New York: John Wiley & Sons, Inc, pp. 255–257.
- (2010). "Chapter 4 : The origin of Biopotentials". In: *Medical Instrumentation : Application and Design*. 4th ed. Hoboken, New Jersey: John Wiley & Sons, Inc., pp. 144–146.





# A

## Appendix 1 - Android code

### A.1 Main function for graph activity

```
public final Handler graphHandler = new Handler() {
    public void handleMessage(Message msg) {
        for(Handler handler : mHandlerList) {
            //Rebroadcast the message received to the List of Handlers
            handler.obtainMessage(msg.what, msg.arg1, msg.arg2, ...
            msg.obj).sendToTarget();
        }
        // handlers have a what identifier which identify the type of msg
        switch (msg.what) {
            case ShimmerBluetooth.MSG_IDENTIFIER_DATA_PACKET:
                /*----- Handle the data packet message -----*/
                handleMsgDataPacket(msg);
                break;
            case Shimmer.MESSAGE_TOAST:
                Log.d("toast",msg.getData().getString(Shimmer.TOAST));
                Toast.makeText(graphView.getContext(), ...
                msg.getData().getString(Shimmer.TOAST),
                Toast.LENGTH_SHORT).show();
                if (msg.getData().getString(Shimmer.TOAST).equals...
                ("Device connection was lost")){
                }
                break;
            case ShimmerBluetooth.MSG_IDENTIFIER_STATE_CHANGE:
                /*----- Handle the state change message -----*/
                mShimmer = new Shimmer(graphHandler);
                handleMsgStateChange(msg);
                break;
        }
    }
};
}
```

## A.2 Acquiring and Processing Data

```
...
// Data Analysis Variables
final double samplingRate = 1024;
double shimmerSamplingRate = 0;
final int sampleData = 32;
int count1 = 0;
int count2 = 0;
double sumdataArray1 = 0;
double sumdataArray2 = 0;
double sumARV1 = 0;
double sumARV2 = 0;
double meandataArray1 = 0;
double meandataArray2 = 0;
double[] dataArrayEMG_CH1 = new double[sampleData];
double[] dataArrayEMG_CH2 = new double[sampleData];
double dataEMG_CH1, dataEMG_CH2 = 0;
double dataRawEMG_CH1, dataRawEMG_CH2 = 0;
String dateData = "NaN";
String timeData = "NaN";
public boolean saveToFile = false;

Filter hpf, bsf;

// Write to CSV variables
private FileWriter fw;
private BufferedWriter bw;
private File file;
boolean firstTimeWrite = true;
...

public void handleMessageDataPacket(Message msg) {
    if ((msg.obj instanceof ObjectCluster)) {
        // within each msg an object can be include, objectclusters are
        // used to represent the data structure of the shimmer device

        ObjectCluster objc = (ObjectCluster) msg.obj;
        if (mConfigureOnFirstTime) {
            // ----- High-pass Filter initialize -----
            try {
                double[] cutoffHP = {10};
                double[] cutoffBS = {49, 51};
                shimmerSamplingRate = shimmerDevice.getSamplingRateShimmer();
                hpf = new Filter(Filter.HIGH_PASS, shimmerSamplingRate, cutoffHP);
                bsf = new Filter(Filter.BAND_STOP, shimmerSamplingRate, cutoffBS);
            } catch (Exception e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }
        }
        mConfigureOnFirstTime = false;
    }
}
```

```

//-----Filter Signal and plot -----
// Channel 1
// first retrieve all the possible formats for the current sensor device
Collection<FormatCluster> dataFormats1 = objc.getCollectionOfFormat ...
    Clusters(Configuration.Shimmer3.ObjectClusterSensorName.EMG_CH1_24BIT);
// retrieve the calibrated data
FormatCluster ch1Cluster = ((FormatCluster) ObjectCluster.returnFormat ...
    Cluster(dataFormats1, ChannelDetails.CHANNEL_TYPE.CAL.toString()));

if (ch1Cluster != null) {
dataEMG_CH1 = ch1Cluster.mData;
dataRawEMG_CH1 = dataEMG_CH1;
//Log.d(LOG_TAG, "Read value CH1 : " + dataEMG_CH1);
if (toggleFilter == 0){
    try {
        dataEMG_CH1 = hpf.filterData(dataEMG_CH1);
        dataEMG_CH1 = bsf.filterData(dataEMG_CH1);
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    dataArrayEMG_CH1[count1] = dataEMG_CH1;
    sumDataArray1 += dataArrayEMG_CH1[count1];
    count1++;
    if (count1 == sampleData){
        meanDataArray1 = sumDataArray1 / sampleData;
        for (int i=0; i<dataArrayEMG_CH1.length; i++){
            sumARV1 += abs(dataArrayEMG_CH1[i] - meanDataArray1);
        }
        dataEMG_CH1 = sumARV1 / sampleData;
        dataEMG_CH1 = dataEMG_CH1*100;
        Log.d(LOG_TAG, "Read value CH1 : " + dataEMG_CH1 + "\n");
        objc.addData("EMG_CH1_FILTERED", ...
            ChannelDetails.CHANNEL_TYPE.CAL, "uV", dataEMG_CH1);
        String[] yprop = {objc.getShimmerName(), ...
            "EMG_CH1_FILTERED", "CAL", "uV"};
        try {
            if (plotUtil.checkIfPropertyExist(yprop, 1) == false) {
                plotUtil.addSignal1(yprop, dynamicPlot1);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            plotUtil.filterDataAndPlot(objc, dynamicPlot1, 1);
        } catch (Exception e) {
            e.printStackTrace();
        }
        count1 = 0;
        sumDataArray1 = 0;
        sumARV1 = 0;
    }
}
else{
Log.d(LOG_TAG, "No filter is applied.");
}
}

```

## A. Appendix 1 - Android code

---

```
count1++;
if (count1 == sampleData){
    Log.d(LOG_TAG, "Read value CH1 : " + dataEMG_CH1 + "\n");
    objc.addData("EMG_CH1_FILTERED", ...
        ChannelDetails.CHANNEL_TYPE.CAL, "mV", dataEMG_CH1);
    String[] yprop = {objc.getShimmerName(), ...
        "EMG_CH1_FILTERED", "CAL", "mV"};
    try {
        if (plotUtil.checkIfPropertyExist(yprop, 1) == false) {
            plotUtil.addSignal1(yprop, dynamicPlot1);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    try {
        plotUtil.filterDataAndPlot(objc, dynamicPlot1, 1);
    } catch (Exception e) {
        e.printStackTrace();
    }
    count1 = 0;
}
} else {
    Log.d(LOG_TAG, "FormatCluster Channel 1 is null.-----");
}
// Channel 2 =====
Collection<FormatCluster> dataFormats2 = objc.getCollectionOf ...
    FormatClusters(Configuration.Shimmer3.ObjectClusterSensor ...
    Name.EMG_CH2_24BIT);
FormatCluster ch2Cluster = ((FormatCluster) ObjectCluster. ...
    returnFormatCluster(dataFormats2, ...
    ChannelDetails.CHANNEL_TYPE.CAL.toString()));

if (ch2Cluster != null) {
    dataEMG_CH2 = ch2Cluster.mData;
    dataRawEMG_CH2 = dataEMG_CH2;
    if (toggleFilter == 0){
        try {
            dataEMG_CH2 = hpf.filterData(dataEMG_CH2);
            dataEMG_CH2 = bsf.filterData(dataEMG_CH2);
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    dataArrayEMG_CH2[count2] = dataEMG_CH2;
    sumdataArray2 += dataArrayEMG_CH2[count2];
    count2++;
    if (count2 == sampleData){
        meandataArray2 = sumdataArray2 / sampleData;
        for (int i=0; i<dataArrayEMG_CH2.length; i++){
            sumARV2 += abs(dataArrayEMG_CH2[i] - meandataArray2);
        }
        dataEMG_CH2 = sumARV2 / sampleData;
        dataEMG_CH2 = dataEMG_CH2*100;
        Log.d(LOG_TAG, "Read value CH2 : " + dataEMG_CH2);
        objc.addData("EMG_CH2_FILTERED", ...
```

```

        ChannelDetails.CHANNEL_TYPE.CAL, "uV", dataEMG_CH2);
String[] yprop2 = {objc.getShimmerName(), ...
    "EMG_CH2_FILTERED", "CAL", "uV"};
try {
    if (plotUtil.checkIfPropertyExist(yprop2, 2) == false) {
        plotUtil.addSignal2(yprop2, dynamicPlot2);
    }
} catch (Exception e) {
    e.printStackTrace();
}
try {
    plotUtil.filterDataAndPlot(objc, dynamicPlot2, 2);
} catch (Exception e) {
    e.printStackTrace();
}
count2 = 0;
sumdataArray2 = 0;
sumARV2 = 0;
}
}
else{
Log.d(LOG_TAG, "No filter is applied.");
count2++;
if (count2 == sampleData){
    Log.d(LOG_TAG, "Read value CH2 : " + dataEMG_CH2);
    objc.addData("EMG_CH2_FILTERED", ...
        ChannelDetails.CHANNEL_TYPE.CAL, "mV", dataEMG_CH2);
String[] yprop2 = {objc.getShimmerName(), ...
    "EMG_CH2_FILTERED", "CAL", "mV"};
try {
    if (plotUtil.checkIfPropertyExist(yprop2, 2) == false) {
        plotUtil.addSignal2(yprop2, dynamicPlot2);
    }
} catch (Exception e) {
    e.printStackTrace();
}
try {
    plotUtil.filterDataAndPlot(objc, dynamicPlot2, 2);
} catch (Exception e) {
    e.printStackTrace();
}
count2 = 0;
}
}
} else {
    Log.d(LOG_TAG, "FormatCluster Channel 2 is null.-----");
}
dat = DateFormat.getDateInstance(DateFormat.MEDIUM).format(new Date());
tim = DateFormat.getTimeInstance(DateFormat.MEDIUM).format(new Date());
/**
 * ----- Writing all channels of CAL data to CSV file -----
 */
if(saveToFile){
if(firstTimeWrite) {
    //Construct a file
    String baseDir = android.os.Environment.getExternalStorage ...

```

## A. Appendix 1 - Android code

---

```
        Directory().getAbsolutePath();
String fileName = "EMGbiofeedback Data " + DateFormat.getDate ...
        TimeInstance().format(new Date()) + ".csv";
String path = baseDir + "/EMGBiofeedBack";
String filePath = path + File.separator + fileName;
file = new File(path);
if(!file.exists()){
    file.mkdir();
}
else {
    file = new File(filePath);
    try {
        if (!file.exists()) {
            file.createNewFile();
        }
        fw = new FileWriter(file.getAbsolutePath());
        bw = new BufferedWriter(fw);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
//Write headers on first-time
try {
    bw.write("DATE" + "," + "TIME" + "," + ...
            "EMG Channel 1 (Raw Data)" + "," + ...
            "EMG Channel 2 (Raw Data)");
    bw.write("\n");
} catch (IOException e) {
    e.printStackTrace();
}
firstTimeWrite = false;
}
else {
    try {
        bw.write(dat + "," + tim + "," + ...
                dataRawEMG_CH1 + "," + dataRawEMG_CH2);
        bw.write("\n");
    } catch (IOException e3) {
        e3.printStackTrace();
    }
}
}
}
}
```

## A.3 Filtering

```

package com.shimmerresearch.algorithms;

import android.widget.Toast;

import java.util.Arrays;
import java.util.List;

public class Filter {

    public static int LOW_PASS = 0;
    public static int HIGH_PASS = 1;
    public static int BAND_PASS = 2;
    public static int BAND_STOP = 3;

    // filter parameters
    private int filterType;
    private double samplingRate = Double.NaN;
    private double[] cornerFrequency;
    private int nTaps;
    private double minCornerFrequency, maxCornerFrequency;

    // buffered data (for filtering streamed data)
    private double[] bufferedX;

    // filter coefficients {h}
    private double[] coefficients;

    // input parameters are invalid
    private boolean validParameters = false;

    // default parameters
    private double defaultSamplingRate = 512;
    private double[] defaultCornerFrequency = { 0.5 };
    private int defaultNTaps = 200;

    public Filter() throws Exception{
        filterType=LOW_PASS;

        SetFilterParameters(LOW_PASS, defaultSamplingRate, ...
            defaultCornerFrequency, defaultNTaps);
    }

    public Filter(int filterType) throws Exception{

        this.filterType=filterType;

        SetFilterParameters(filterType, defaultSamplingRate, ...
            defaultCornerFrequency, defaultNTaps);
    }

    public Filter(int filterType, double samplingRate, ...

```

## A. Appendix 1 - Android code

---

```
        double[] cornerFrequency) throws Exception{

        this.filterType=filterType;

        SetFilterParameters(filterType, samplingRate, ...
            cornerFrequency, defaultNTaps);
    }

    public Filter(int filterType, double samplingRate, ...
        double[] cornerFrequency, int nTaps) throws Exception{

        this.filterType=filterType;

        SetFilterParameters(filterType, samplingRate, ...
            cornerFrequency, nTaps);
    }

    public void SetFilterParameters(int LoHi, ...
        double samplingRate, double[] cornerFrequency, int nTaps) ...
        throws Exception{

        //reset the buffers
        this.bufferedX = null;

        if(cornerFrequency.length!=1){
            if(cornerFrequency[0] > cornerFrequency[1]){
                minCornerFrequency = cornerFrequency[1];
                maxCornerFrequency = cornerFrequency[0];
            }
            else{
                minCornerFrequency = cornerFrequency[0];
                maxCornerFrequency = cornerFrequency[1];
            }
        }
        else
            minCornerFrequency=maxCornerFrequency=cornerFrequency[0];

        if (maxCornerFrequency > samplingRate / 2)
        {
            this.validparameters = false;
            throw new Exception("Error: cornerFrequency is greater
                than Nyquist frequency. Please choose valid parameters.");
        }
        else
        {
            if (nTaps % 2 != 0)
            {
                nTaps--;
                Toast.makeText(null, "Warning: nPoles is not an
                    even number. nPoles will be rounded to " + ...
                    Integer.toString(nTaps), Toast.LENGTH_LONG).show();
            }

            // High pass or Low pass filter

```



```

if (LoHi == LOW_PASS || LoHi == HIGH_PASS)
{
    this.samplingRate = samplingRate;
    this.cornerFrequency = cornerFrequency;
    this.nTaps = nTaps;

    double fc = (cornerFrequency[0] / samplingRate);
    // calculate filter coefficients
    coefficients = new double[nTaps];
    coefficients = calculateCoefficients(fc, LoHi, nTaps);
    this.validparameters = true;
}
else if (LoHi == BAND_PASS || LoHi == BAND_STOP)
{
    if (cornerFrequency.length != 2)
        throw new Exception("Error. Bandpass or bandstop
            filter requires two corner frequencies to be
            specified");

    this.samplingRate = samplingRate;
    this.nTaps = nTaps;

    double fcHigh = maxCornerFrequency / samplingRate;
    double fcLow = minCornerFrequency / samplingRate;

    // calculate filter coefficients
    double[] coefficientHighPass = ...
        calculateCoefficients(fcHigh, HIGH_PASS, nTaps);
    double[] coefficientLowPass = ...
        calculateCoefficients(fcLow, LOW_PASS, nTaps);

    coefficients = new double[coefficientHighPass.length];
    for(int i=0; i<coefficientHighPass.length;i++){
        if(LoHi == BAND_PASS)
            // sum of HPF and LPF for bandstop filter,
            // spectral inversion for bandpass filter
            coefficients[i] = - (coefficientHighPass[i] + ...
                coefficientLowPass[i]);
        else
            // sum of HPF and LPF for bandstop filter
            coefficients[i] = coefficientHighPass[i] + ...
                coefficientLowPass[i];
    }
    if(LoHi == BAND_PASS){
        coefficients[(nTaps/2)] = coefficients[(nTaps/2)] +1;
    }
    this.validparameters = true;
}
else
    throw new Exception("Error. Undefined filter type:
        use 0 - lowpass, 1 - highpass, 2- bandpass,
        or 3- bandstop");
}
}

public double filterData(double data) throws Exception

```

## A. Appendix 1 - Android code

---

```
{
    double dataFiltered = Double.NaN;

    if (!this.validparameters)
        throw new Exception("Error. Filter parameters are invalid.
            Please set filter parameters before filtering data.");
    else
    {
        int nSamples = 1;
        int bufferSize = this.nTaps;
        if(bufferedX==null){
            // buffers are initiliazed to 0 by default
            bufferedX = new double[bufferSize+nSamples];
            // fill the buffer X with the first data
            Arrays.fill(bufferedX, data);
        }
        else{
            // all the elements in the buffer are shifted
            // one position to the left
            System.arraycopy(bufferedX, 1, bufferedX, 0, ...
                bufferedX.length-1);
            bufferedX[bufferedX.length-1] = data;
        }
        double Y = filter(bufferedX);
        dataFiltered = Y;
    }
    return dataFiltered;
}

public double[] filterData(double[] data) throws Exception
{
    if (!this.validparameters)
        throw new Exception("Error. Filter parameters are
            invalid. Please set filter parameters before
            filtering data.");
    else
    {
        double[] dataFiltered = new double[data.length];

        for(int i=0; i<data.length; i++){
            double individualDataFiltered=filterData(data[i]);
            dataFiltered[i] = individualDataFiltered;
        }
        return dataFiltered;
    }
}

private double filter(double[] X)
{
    int nTaps = coefficients.length;
    double Y = 0;

    for(int i=0; i<nTaps; i++)
        Y += X[nTaps-i]*coefficients[i];
    return Y;
}
```

```

private double[] calculateCoefficients(double fc, int LoHi, ...
    int nTaps) throws Exception
{
    if (!(LoHi == LOW_PASS || LoHi == HIGH_PASS))
        throw new Exception("Error: the function
            calculateCoefficients() can only be called for
            LPF or HPF.");

    //Initialization
    int M = nTaps;
    double[] h = new double[M];
    for(int i=0;i<M; i++)
        h[i] = 0;

    for(int i=0;i<M;i++){
        h[i] = 0.42 - 0.5 * Math.cos((2*Math.PI*i)/M) + ...
            0.08*Math.cos((4*Math.PI*i)/M);
        if(i!=M/2)
            h[i] = h[i] * (Math.sin(2*Math.PI*fc*...
                (i-(M/2))))/(i-(M/2));
        else
            h[i] = h[i] * (2*Math.PI*fc);
    }
    double gain = 0;
    for(int i=0;i<h.length;i++)
        gain += h[i];

    for(int i=0;i<h.length;i++){
        if(LoHi == HIGH_PASS){
            h[i] = - h[i]/gain;
        }
        else
            h[i] = h[i]/gain;
    }
    if(LoHi == HIGH_PASS){
        h[M/2] = h[M/2] + 1;
    }
    return h;
}

public double GetSamplingRate(){
    return samplingRate;
}

protected void SamplingRate(double samplingrate){
    samplingRate = samplingrate;
}

public double[] GetCornerFrequency(){
    return cornerFrequency;
}

protected void SetCornerFrequency(double[] cf){
    cornerFrequency = cf;
}

```

```
public static double[] fromListToArray(List<Double> list) {  
  
    double [] array = new double[list.size()];  
    for(int i=0;i<list.size();i++)  
        array[i] = list.get(i);  
  
    return array;  
}  
  
public void resetBuffers() {  
  
    bufferedX = null;  
}  
  
public int getFilterType() {  
    return filterType;  
}  
}
```

## A.4 Plotting

```

package com.tng_asus.emgbiofeedback;

import android.graphics.Color;

import com.androidplot.xy.LineAndPointFormatter;
import com.androidplot.xy.XYPlot;
import com.shimmerresearch.driver.FormatCluster;
import com.shimmerresearch.driver.ObjectCluster;
import com.shimmerresearch.tools.PlotManagerAndroid;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
import java.util.List;

import pl.flex_it.androidplot.XYSeriesShimmer;

public class PlotUtil extends PlotManagerAndroid {

    List<XYSeriesShimmer> mListofSeries1 = ...
        new ArrayList<XYSeriesShimmer>();
    List<XYSeriesShimmer> mListofSeries2 = ...
        new ArrayList<XYSeriesShimmer>();
    public List<int[]> mListofTraceColorsCurrentlyUsed = ...
        Collections.synchronizedList(new ArrayList<int[]>());
    //below is used to identify he signals coming into the filter
    public List<String[]> mListofPropertiestoPlot1 = ...
        Collections.synchronizedList(new ArrayList<String[]>());
    // below is used to identify he signals coming into the filter
    public List<String[]> mListofPropertiestoPlot2 = ...
        Collections.synchronizedList(new ArrayList<String[]>());

    public void addSignal1(String [] signal, XYPlot chart) ...
        throws Exception{
        if (!checkIfPropertyExist(signal)){
            // Add the series first, cause the filter might find
            // the signal before the series is added
            int index = mListofSeries1.size();
            String name = signal[0]+ " " + signal[1]+" " + ...
                signal[2] + " " + signal[3];
            //List<Number> data = new ArrayList<Number>();
            List<Number> d = new ArrayList<Number>();
            XYSeriesShimmer xys = new XYSeriesShimmer(d);
            xys.setTitle(name);
            xys.setXAxisLimit(mXAxisLimit);
            xys.setClearGraphatLimit(mClearGraphatLimit);
            mListofSeries1.add(xys);
            addSignalGenerateRandomColor(signal, 1);
            int [] colorrgbarray = ...
                mListofTraceColorsCurrentlyUsed.get(index);
            LineAndPointFormatter lapf = ...

```

## A. Appendix 1 - Android code

---

```
        new LineAndPointFormatter(colorrngbaray[0], ...
        colorrngbaray[1], colorrngbaray[2]);
lapf = new LineAndPointFormatter(Color.rgb(colorrngbaray[0], ...
        colorrngbaray[1], colorrngbaray[2]), null, null);
chart.addSeries(mListofSeries1.get(index),lapf);
mDynamicPlot = chart;

    for (XYSeriesShimmer xy: mListofSeries1){
        xy.clearData();
    }
}else {
    throw new Exception("Error: " + ...
        joinChannelStringArray(signal) + ...
        " Signal/Property already exist.");
}
}

public void addSignal2(String [] signal, XYPlot chart) ...
throws Exception{
    if (!checkIfPropertyExist(signal)){
        // Add the series first, cause the filter might find
        // the signal before the series is added
        int index = mListofSeries2.size();
        String name = signal[0]+ " " + signal[1]+" " + ...
            signal[2] + " " + signal[3];
        //List<Number> data = new ArrayList<Number>();
        List<Number> d = new ArrayList<Number>();
        XYSeriesShimmer xys = new XYSeriesShimmer(d);
        xys.setTitle(name);
        xys.setXAxisLimit(mXAxisLimit);
        xys.setClearGraphatLimit(mClearGraphatLimit);
        mListofSeries2.add(xys);
        addSignalGenerateRandomColor(signal, 2);
        int [] colorrngbaray = ...
            mListofTraceColorsCurrentlyUsed.get(index);
        LineAndPointFormatter lapf = ...
            new LineAndPointFormatter(colorrngbaray[0], ...
                colorrngbaray[1], colorrngbaray[2]);
        lapf = new LineAndPointFormatter(Color.rgb(colorrngbaray[0], ...
            colorrngbaray[1], colorrngbaray[2]), null, null);
        chart.addSeries(mListofSeries2.get(index),lapf);
        mDynamicPlot = chart;

        for (XYSeriesShimmer xy: mListofSeries2){
            xy.clearData();
        }
    }else {
        throw new Exception("Error: " + ...
            joinChannelStringArray(signal) + ...
            " Signal/Property already exist.");
    }
}

public void filterDataAndPlot (ObjectCluster ojc, XYPlot chart, ...
    int channelNo) throws Exception {
    // TODO Auto-generated method stub
```

```

List<String[]> toPlot = null;
mDynamicPlot = chart;
if (channelNo == 1) {
    toPlot = mListofPropertiestoPlot1;
}else{
    toPlot = mListofPropertiestoPlot2;
}
for (int i=0;i<toPlot.size();i++){
    String[] props = toPlot.get(i);
    if (ojc.getShimmerName().equals(props[0])){

        FormatCluster f = ...
        ObjectCluster.returnFormatCluster(ojc.get...
        CollectionOfFormatClusters(props[1], props[2]);
        if (f!=null){
            if (channelNo == 1) {
                mListofSeries1.get(i).addData(f.mData);
            }else{
                mListofSeries2.get(i).addData(f.mData);
            }
        } else {
            throw new Exception("Signal not found");
        }
    }
}
if (mDynamicPlot!=null){
    mDynamicPlot.redraw();
}
}

protected void addSignal(String[] channelStringArray, ...
int channelNo){
    if(channelNo == 1 ) {
        mListofPropertiestoPlot1.add(channelStringArray);
    }else{
        mListofPropertiestoPlot2.add(channelStringArray);
    }
}

protected void addSignalGenerateRandomColor(String [] ...
channelStringArray, int channelNo){

    addSignalAndUseFixedColor(channelStringArray, ...
generateRandomColor(), channelNo);
}

protected void addSignalAndUseFixedColor(String [] ...
channelStringArray, int[] rgb, int channelNo){

    addSignal(channelStringArray, channelNo);
    mListofTraceColorsCurrentlyUsed.add(rgb);
}

public boolean checkIfPropertyExist(String[] ...
channelStringArray, int channelNo){

```

## A. Appendix 1 - Android code

---

```
List<String[]> toPlot = null;
if (channelNo == 1) {
    toPlot = mListofPropertiestoPlot1;
}else{
    toPlot = mListofPropertiestoPlot2;
}
synchronized(toPlot){
    Iterator<String[]> entries = toPlot.iterator();
    while (entries.hasNext()) {
        String[] propertiestoPlot = entries.next();
        boolean test = true;
        int lengthToUse = channelStringArray.length > ...
            4? 4:channelStringArray.length;
        for (int p=0; p<lengthToUse; p++){
            if((propertiestoPlot[p]==null) && ...
                (channelStringArray[p]==null)){
                test = true;
            }
            else if((propertiestoPlot[p]==null) || ...
                (channelStringArray[p]==null)){
                test = false;
            }
            else if (!propertiestoPlot[p].equals...
                (channelStringArray[p])){
                test = false;
            }
        }
        if (test==true){
            return true;
        }
    }
}
return false;
}
```