Delft University of Technology

Faculty of Electrical Engineering, Mathematics and Computer Science

Fourier Analysis of the microvascular function of women with and without migraine

by Nisharani Agnihotri

To obtain the degree of Bachelor of Science in Applied Mathematics at the Delft University of Technology, to be defended publicly on Tuesday August 29, 2023

Student number:	4721519	
Duration:	07/03/2023-29/08/2023	
Thesis committee:	Dr. C. Vuik, supervisor	TU Delft
	Drs. L. Al-Hassany	Erasmus MC
	Dr. M. Keijzer	TU Delft
	Dr. A. Maassen van der Brink	Erasmus MC

Summary

Erasmus MC is conducting a study on the increased risk of cardiovascular disease in women who have migraine. For this purpose, 50 healthy women and 50 women with migraine in age category of 40-60 years were compared using measurements of their dermal blood flow. These measurements are done using two different imaging devices: Laser Speckle Imaging and Laser Doppler Imaging. The result is a recording of the fluctuations in the dermal blood flow over a time period influenced by circumstances and various mechanisms and consists of signal like curves. Erasmus MC have processed these measurements and obtained results. However, they are interested in whether applying Fourier Transform could offer additional insights or new (perhaps faster) methods to process the measurements.
The signal like curves can possibly be decomposed into sine and cosine waves using Fourier Transform. The result of the Fourier Transform consists of a visualisation of the intensity of the various components of such a waveform/curve. Based on that, it is possible to estimate the contribution of the mechanisms to blood perfusion. Existing literature and previous studies provide frequency intervals related to several mechanisms that influence the dermal blood flow.
This thesis will research to what extent the mechanisms contribute to the dermal blood flow across the different phases and how their behaviour differs when the circumstances for the measurement are adjusted. Perhaps there are behaviours that are instantly recognizable across all patients. An interesting question to answer would be whether it is possible to directly visualise the risk groups.

Contents

Chapter 1

Introduction

There are several methods to visualize superficial layers of human dermal skin. This movement can be influenced by several mechanisms. Blood vessels are known to constrict (contract) or dilate (relax), which varies depending on several factors. For example, blood vessels dilate during exercise to allow for more blood to flow.

Migraine is a condition that is possibly linked to micro vascular dysfunction, which is when the small blood vessels get damaged to some extent. Measuring such micro vascular dysfunction is more difficult compared to conventional methods used in general for vascular research. However, one method to measure the micro vascular dysfunction is by doing measurements of the dermal blood flow using Local Thermal Heating (LTH). In this method, areas of human skin are sectioned off using small reservoirs which are filled with water and then heated to increase the dermal flow. Doing this allows certain substances in the blood to break free and therefore, blocking such a substance on one of the sectioned off areas of human skin and comparing it to an area without such blockage can help in understanding the influence of that substance on the passage of blood.

For people with migraine it is believed that the increased risk of cardiovascular disease does not depend on calcification of the larger blood vessels of the large arteries. [1] [2] Instead, the cause seems to be micro vascular dysfunction.[3] The mechanisms that are seemingly linked to micro vascular dysfunction are the Endothelium-Derived Hyperpolarizing Factor (EDHF), nitric oxide (NO) and certain neuropeptides.
In the VASCULAR-study conducted by Erasmus Medical Centre (Erasmus MC) two mechanisms are isolated for further research: nitric oxide and neuropeptides. The way the influence of these mechanisms can be studied is by measuring the dermal blood flow as a response to LTH using the Laser Speckle Imaging (LSI) machine or the Laser Doppler Imaging (LDI) machine. Occasionally, the names Laser Doppler Flowmetry (LDF) and Laser Doppler Perfusion Imaging (LDPI) are used to refer to the method to measure the blood flow using LDI.

These imaging machines return a curve containing the varying amounts of blood flowing through the micro vessels. Currently, the method used for further analysis of the results calculates the area under the curve and the maximum height of the curve. However, there are also other methods for analyzing the curve, such as decomposing it using the Fourier Transform (FT) and the Wavelet Transform. (Qi, 2016) [4] These distributions decompose the original signal for further analysis of the components that make up one such signal. In this way, it might be possible to isolate mechanisms and find the magnitude of their contribution to the signal.
The focus of this project will be to analyse the signal using the Fourier Transform. From

previous analyses of measurements by LDI machines certain lower frequency bands have been isolated from the skin blood flux signals that are thought to represent some mechanisms, including the nitric oxide that is being researched for the VASCULAR-study. [4] [5]

Some variation is expected in the results depending on, amongst other factors, the chosen machine and the physiological tests that are done. Especially the physiological tests cause oscillations of varying frequencies in blood vessels. The idea is that various frequencies correspond to different mechanisms. [6] (Mizeva et al, 2021)

Chapter 2

Background

2.1 Introduction

The LSI and LDI machines are important components of the study. This chapter discusses how these machines are used in practice and what results they produce. Next, the basis of Fourier Analysis will be discussed and the relationship between the two machines and Fourier Analysis.

2.2 LSI in practice

As mentioned previously, the influence of certain physiological mechanisms on the blood flow beneath the skin can be visualised using LSI. By using multiple light scattering, LSI quantifies the fluctuations in the intensity of the scatterings in a large variety of materials. The speckle pattern is caused by differences in the length of the path that the photons have traveled through a material. [7] (Buijs et al, 2019).
The speckle contrast is calculated by the spatial or temporal variance in specific set of speckles. This process is known as LASCA, which stands for Laser Speckle Contrast Analysis, connected to LSCI (Laser Speckle Contrast Imaging). However, the variance in processes that happen at different frequencies is more difficult to quantify since the measurements are done in real-time and the quantification would take some time. To obtain quantitative information, multiple speckle images with several exposure times are required. This happens at the cost of the measurements being real-time. (Buijs et al, 2019).
Another method to calculate the speckle contrast is known as quantitative LSI. This is more commonly used in materials science and measures the differences in processes happening at different frequencies. This method uses the auto-correlation function of the intensity of the speckles as a function of time. The quantitative property does come at the cost of being computationally expensive. (Buijs et al, 2019)

So, two important requirements for the ideal method to calculate speckle contrast follow. It should be fast and quantitative.
Buijs et al. (2019) discuss a new method that uses Fourier transforms to do the required quantification. Related to Fourier transforms is the Fast Fourier transform that makes the processing much faster. [7] According to the Wiener-Khinchin theorem [8] [9] (Wiener, 1930 and Lu and Vaswani, 2009), this method contains the same information as quantitative LSI. [7]

2.3 LDI in practice

For LDI, it was initially believed that the resulting data is highly variable and not reproducible. However, according to Lancaster (2016), if the variations are handled properly then it is possible to derive relevant information from LDI data about the frequencies of the oscillations in the signal by using Fourier Transform.[10] When using Fourier Transform for time-varying data, it is advised to take smaller segments (windows) in time and apply the transform on the segments separately. This can be avoided by using the Wavelet Transform. [5] Applying the Wavelet decomposition leads to information about both the time and frequency domains. (Lancaster, 2016) [10]
The identification of 6 frequency intervals was eventually possible together with the corresponding oscillatory process of the tested mechanisms with the previously described approach. It was concluded that this approach is both reliable and reproducible. (Lancaster, 2016) [10]

2.4 Fourier Analysis

At the basis of Fourier Analysis are the Fourier series. The Fourier series of a function f, which is piece-wise smooth on an interval $[-L, L]$ with $L > 0 \in \mathbb{R}$, converges to the periodic extension of f on the domain on which it is continuous.
Hence, it does not matter whether f is periodic on the interval or not since on an extended domain it is considered to be periodic. The Fourier series representation of a function f can be written as follows:

$$f(t) \sim a_0 + \Sigma_{n=1}^{\infty} a_n \cos\left(\frac{n\pi t}{L}\right) + \Sigma_{n=1}^{\infty} b_n \sin\left(\frac{n\pi t}{L}\right) \tag{2.1}$$

For doing mathematical computations and programming however, it is necessary to discretize the time domain. In other words, the interval for which t is defined will be divided over some N number of steps. This is known as the Discrete Fourier Transform (DTF) and there are two variants of this. For periodic signals it is simply known as the Discrete Fourier Transform (DFT) and for aperiodic signals, which is then assumed to be periodic on an extended domain, there is the Discrete Time Fourier Transform (DTFT). [11]

Applying either one of these transforms on the input signal results in a discrete signal that is given in the frequency domain. In other words, the signal that was originally represented in the time domain, is now represented in the frequency domain. This process is also known as analysis. This can enable the discovery of the most prominent frequencies in a signal, as mentioned earlier. Going back to the time domain from the frequency domain is known as synthesis. [11]

2.4.1 Frequency

The frequency of something is measured in hertz (Hz), where 1Hz corresponds to one rotation in a second. This kind of frequency will be denoted by f, which is different to the f used previously to refer to a function. Hence, the f used from now onward will refer to the frequency, unless stated otherwise.
Related to this is the angular frequency, which is denoted by ω and expressed in radians. The relation between the two can be described as follows: $\omega = 2\pi f$
As mentioned previously, for programming it is necessary to discretize the time domain. Hence, the time interval is divided over N steps. However, this N cannot be decided randomly. It is important to take a certain number of samples to avoid some type of distortion of the signal, when trying to reconstruct it. If not enough samples have been

taken, then it is possible that some important frequencies are missed which results in the distortion of the signal. To determine the number of samples, the sampling frequency is required. According to the Nyquist-Shannon sampling theorem we have: $f_s \geq 2 \cdot f_{max}$, where f_s represents the sampling frequency and f_{max} represents the maximum frequency or the fundamental frequency (in the case the signal consists of several components).
To this end, the digital frequency at index m (from the N samples) is expressed as

$$\hat{f} = m\frac{f_s}{N}$$

Additionally, the digital angular frequency at index m (on an interval ranging from 0 to 2π radians) is expressed as

$$\hat{\omega} = m\frac{2\pi}{N}$$

[11]

2.4.2 Spectral analysis

After applying the Fourier Transform to the signal, the visualisation of the frequency components can be found through the amplitude spectrum. Where previously the signal was represented as a curve on a specific time interval, the amplitude spectrum shows the various frequency bands that the signal is composed of and the extent of their presence in the signal. So, the y-axis represents the amplitude of the frequencies given on the x-axis. Moreover, generally it is assumed that a sinusoidal wave does not have an angular shift. In other words, the wave starts at the origin and has not been shifted prior to starting the measurement. However, this is not always the case. It is possible that the wave is shifted by some angle at the start of the time interval of the measurement. The extent to which a frequency has been shifted already, is represented by the phase plot.

2.4.3 Fast Fourier Transform

There are two ways to apply the the Fourier Transform. The first method would be to, of course, implement it by yourself. The other method would be to use a Fast Fourier Transform function (reffered to as fft) included in a library in Python, such as SciPy or Numpy.
Both ways result in the same solution however, the advantage of using fft becomes clear when used on large data sets. Usually, an implementation has a complexity of N^2 compared to $N \log N$, which is the complexity of fft. Usually, an implementation has a runtime of N^2 compared to $N \log N$, which is the runtime of fft. Implementations of the fft are usually based on the method published in 1965 by James Cooley and John Tukey.

2.5 To summarize

Fourier Analysis is an excellent tool to decompose period data into frequency bands for further study. This method can be extended to aperiodic data as well. In order to do computations, the time domain of the data has to be discretized. Literature suggests that the results using quantitative LSI correspond to results generated using FFT. LDI has previously been used to identify processes in the blood flow, which were initially believed to be highly variable. However, handling the variability properly ensures that relevant information can still be obtained.

Chapter 3

Methods

3.1 Introduction

This chapter discusses the methods employed in the VASCULAR-study and introduces what is expected to be achieved by using Fourier Analysis on the data.

3.2 Methodology of the VASCULAR-study

In addition to the information gathered from the papers on the different processes that use Laser Speckle Imaging (LSI) and Laser Doppler Imaging (LDI), here follows a short account of the process used for the measurements in the VASCULAR-study at Erasmus MC.
The VASCULAR-study was conducted on 100 women in total in age category of 40-60 years who are otherwise healthy with requirements such as no prior history of cardiovascular disease, non smoker, etc. Of the 100 women, 50 suffer from migraine and the other 50 do not. The initial goal of the study is to compare these two groups for an increased risk of cardiovascular disease.
The measurements were conducted on the lower arm of the patients, where three areas on each arm were isolated using small reservoirs and the areas were named Region of Interest 1/2/3 (ROI 1/2/3). For the study, the neuropeptides are blocked on ROI 2, nitric oxide on ROI 1 and the third section (ROI 3) is used to compare the results afterwards. The focus is mainly on the effect of blocking a certain mechanism on the blood flow, which ensures a better understanding of the contribution of that mechanism. The reservoirs are filled with water, which is then heated to ensure an increase in the blood flow, a process known as LTH, as mentioned previously. The five minutes prior to LTH are known as the baseline phase. Roughly 10 minutes later, a first peak is reached, which is appropriately called the peak phase. Roughly 30-35 minutes after the start of the heating process, the plateau phase is reached. In this phase, the absence of the mechanisms is expected to show an influence on the dilation of the blood vessels and the blood flow.
Currently, after obtaining the measurements the processing of the results is done after calculating the area under curve (AUC) and taking the maximum height. The goal now is to research whether the processing of those results can also be done accurately and efficiently by using the Fourier Transform.

For one of the two arms the LSI machine is used. The machine does measurements in real time and maps the blood flow of a blood vessel at any moment. The blood flow can also be influenced due to movement of the patient and similar external factors, hence it is important to note that this method is slightly susceptible to some noise. However, the LSI machine allows for measurements to be done in real-time and is considered to be a reliable

method.

LDI measurements are not done real time. Instead, the machine does measurements in intervals of a few seconds and maps an average value for each interval. Therefore the information given by method is not as accurate as the information given by LSI. Nevertheless, it is a good method to compare/confirm the results of LSI with and therefore its inclusion is necessary. Moreover, this method is less susceptible to noise due to movement of the patients.

3.3 Fourier Analysis how it will be used.

The data sets generated by the LSI and LDI machines are generated by the fluctuations in the amount of blood flowing through the blood vessels over a given period in time. As described previously, these fluctuations can depend on various factors. In the VASCULAR-study, the emphasis lies on the effect of nitric oxide (NO) and certain neuropeptides. However, here the effect of neuropeptides will not be considered. Due to the nature of the neuropeptides, it is recommended to leave them out. Instead, the focus will be on NO and the Endothelium-Derived Hyperpolarizing Factor (EDHF). The fluctuations caused by these mechanisms can possibly be decomposed into sine and cosine waves using Fourier Transform, which enables the visualisation of the intensity of these components. In other words, it is possible to see the contribution of these mechanisms to blood perfusion.

Previous studies ([4] and [10]) have been able to isolate frequency intervals corresponding to the aforementioned mechanisms. According to these studies, the interval 0.005-0.0095Hz could be attributed to EDHF. [4] [10]
Moreover, for the interval 0.0095-0.021Hz is the one where nitric oxide plays an important role in the fluctuations. As phrased by Lancaster, "Interval V, from 0.0095 to 0.021 Hz, was first found to contain oscillations due to endothelial activity through the use of the endothelial dependent vasodilator acetylcholine (ACh) and the endothelial independent vasodilator sodium nitroprusside (SNP) [123]. This endothelial activity was found to be partly mediated by nitric oxide (NO) [102] " [10] Based on this, ACh and SNP also play a role in the blood perfusion caused by NO.

3.4 Conclusion

The Erasmus MC conducted a study, called the VASCULAR-study, on 100 women to find out the role two mechanisms (nitric oxide and neuropeptides) play in the increased risk of cardiovascular disease in women suffering from migraine. To this end, they used two imaging machines (Laser Speckle and Laser Doppler) on each patient to map the blood perfusion. They ran their calculations and obtained results.
The goal now is to use Fourier Transform on their measurements in order to research whether accurate results can be generated efficiently using this method. Due to the nature of the mechanism, neuropeptides are replaced by EDHF. Previous studies have found corresponding frequency bands for both NO (0.0095-0.021Hz) and EDHF (0.0005-0.0095).

Chapter 4

Analyses of the data sets

4.1 Introduction

Initial analysis was done on self-generated data, to gain an intuitive understanding of the method. Several ways exist to generate your own data, depending on the requirements. In this chapter, two such ways will be discussed, followed by an introduction to the actual data set.
As the first step of analysing any type of data, Fourier Transform will be applied.

4.2 Generated data set 1

As mentioned before, the data that will be used here in the future consists of mappings of blood flow through blood vessels over a certain time interval. This type of biological data is rather difficult to imitate. Several processes take place simultaneously and influence the measurements. Moreover, the measurements can also be influenced by factors from outside, making the data susceptible to noise.
Therefore, in order to generate data the simplest way is to generate 30 uniformly distributed random points in the range from 0 to 100. Using uniformly distributed random points is an attempt to obtain points without an underlying pattern in order to avoid the over- or under-representation of certain points. This is to ensure that the data is not periodic, since the final data will not have a strongly periodic appearance. The idea is to "imitate" a measurement of 30 seconds and the range from 0 to 100 seemed reasonable enough. Moreover, the value is in arbitrary units since that holds for the actual data set too. Figure 4.1 contains the data set and figure 4.2 contains the corresponding amplitude spectrum plot. This plot shows the level of presence of the frequencies in the data set. Since the data consists of uniformly distributed random points, there is a huge peak in figure 4.2 for frequency 0.

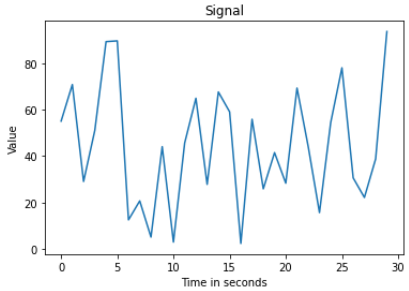

Figure 4.1: Self-generated

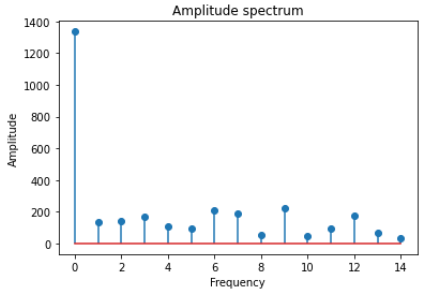

Figure 4.2: Amplitude spectrum

4.2.1 The DC offset

Such a peak is caused by the component wave of the signal that is simply a horizontal line, thus having frequency 0. It describes the amount with which the signal has been shifted from the x-axis. This component is known as the Direct Current (DC) offset. [11] To remedy this, it is possible to subtract the mean value of the data from every data point. This can be seen in figure 4.3

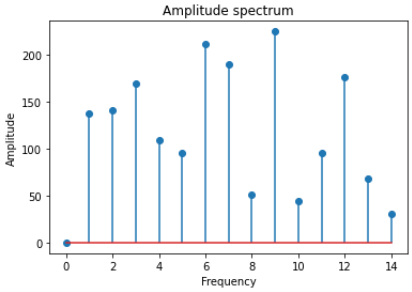

Figure 4.3: Amplitude spectrum after DC offset removal

In this case, the data being randomly generated is causing all the frequencies to have a somewhat equal contribution to the signal. No uniquely significant frequencies seem to be present there, even after the DC offset removal.

4.3 Self-generated data set 2

Another way to generate a data set on which Fourier Transform can be applied is by defining functions to create sine waves. Using 4 such functions, to which some randomly generated noise has been added, figure 4.4 is obtained. In figure 4.5 the corresponding amplitude spectrum can be seen, where the DC offset (see section 4.2.1) has been removed and the frequency axis has been halved (see section2.4.1). From this plot, it is clear that the 4 sine functions, and therefore this signal, consists of the following frequencies: 19 Hz, 37 Hz, 53 Hz and 67 Hz.

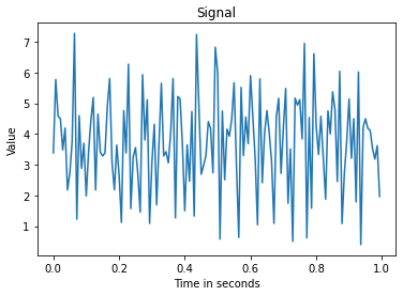

Figure 4.4: Self-generated

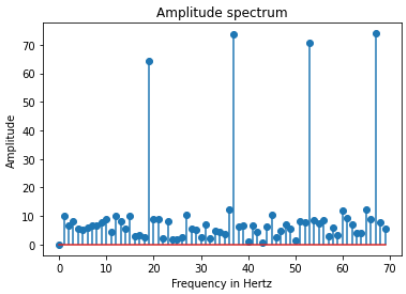

Figure 4.5: Amplitude spectrum

4.4 To summarize

As mentioned previously in section 2.4.3, the libraries SciPy and NumPy contain an implementation of the Fast Fourier Transform which can be used immediately. The emphasis, however, lies on how useful information can be obtained after the transformation has been done. More on that follows in the next chapter.

Chapter 5

Results for the first patient

5.1 Introduction

In total 13 of the 200 measurements have been analysed. In this chapter, the very first one of the 13 will be discussed. For this set, both the Laser Doppler Imaging (LDI) and Laser Speckle Imaging (LSI) measurements will be included.
First, some observations will be described, followed by a deeper analysis of both data sets.

5.2 Description of the sets

Region of interest (ROI) 3 for both LDI and LSI can be seen in figures 5.1 and 5.2. The LSI measurement was done over a time period of 42 minutes and 48 seconds, which is converted to seconds entirely for the visualisation. Similarly, the LDI measurement was done over a period of 42 minutes and 24 seconds. On the x-axis, these have been converted into seconds. The y-axis contains the changes in dermal blood flow. This does not correspond to any units.

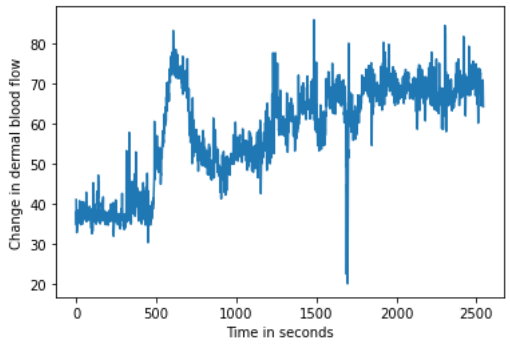

Figure 5.1: LSI measurement

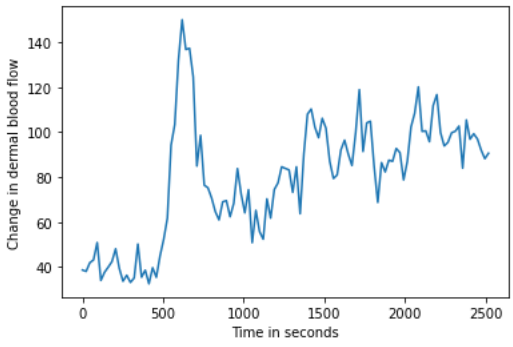

Figure 5.2: LDI measurement

5.3 Initial observations

Some observations can be made from these images. First of all, the LSI measurement, despite its many new advantages for mapping blood flow, is very susceptible to noise and outliers since the measurements are being done in real-time. The outliers can, for example, be caused by a sudden movement of the patient. To a great extent, the outliers have been taken care of for the VASCULAR-study, by taking averages of measurements in the neighbourhood of the outlier, for example. The remaining processing of this data set will

be described in the upcoming sections. Moreover, this measurement is done in real time. There are 2544 data points over a period of 2568 seconds, which comes down to a sampling frequency of 1 Hz. As described in section 2.4.1, the Nyquist frequency becomes 0.5 Hz, meaning that only mechanisms with a frequency below that can be obtained.
Similarly, for LDI there are 111 samples over a period of 2544 seconds. This implies a sampling frequency of rougly 0.04 Hz, therefore recognizing only those mechanisms that have a frequency of at most 0.02 Hz. One important distinction between this measurement and the LSI measurement is that the LDI measurement is less susceptible to noise despite not doing measurements in real time. For example, sudden movements will be averaged out during the measurement interval.

5.4 Smoothing

As observed previously, the LSI measurement contains lots of noise. This can cause irrelevant frequency bands to be over represented and in general give an incorrect idea of what the data is composed of.

To remedy this, one can implement a moving average filter. This filter takes a predefined number x and for every data point, it calculates the average of that point alongside $x-1$ of its neighbouring points. The filter used here, takes the next $x-1$ points to calculate the average for every point. It is important to not take x to be too large or too small. If taken too large, the risk is that the curve will be smoothed too much to the extent that it might look like linear curve. For example, in the most extreme case x is equal to the size of the data set. Then, for every data point the filter changes the value into the average of all the points. Repeatedly doing that, will cause the value of each data point to grow linearly due to summing of the average of the points to its right. Figures 5.3 and 5.4 confirm that behaviour by taking values 500 and 2500 for the LSI measurement.
On the other hand, the problem with taking a small value for x is of course that the noise is barely reduced. This can be seen in figure 5.5.
After considering several values between 10 and 60, the final value for x is set to 40, as can be seen in figure 5.7.

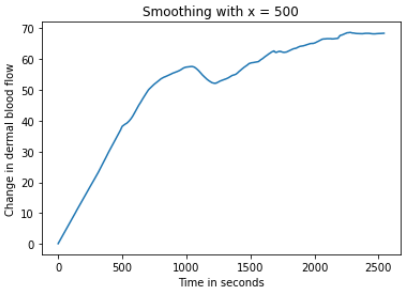

Figure 5.3: Smoothed LSI

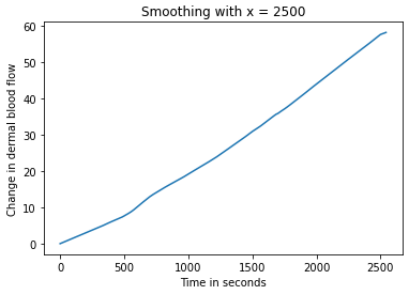

Figure 5.4: Smoothed LSI

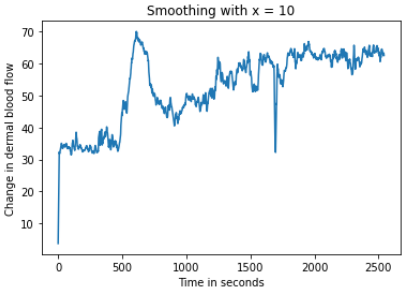

Figure 5.5: Smoothed LSI

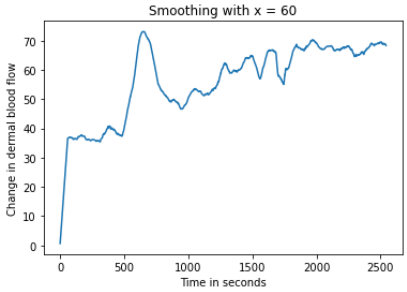

Figure 5.6: Smoothed LSI

Figure 5.7: Final smoothed version

5.5 Initial analysis LSI patient 1

This section continues discussing ROI 3 for patient 1. As discussed in chapter 3.2, there are three significant phases in every measurement: the baseline, the peak phase and the plateau phase. These phases will be considered separately using the smoothed LSI data set. Each of these phases is different in length. The baseline phase is 5 minutes long and starts 5 minutes before the local thermal heating is started. So, this phase consists of 300 points. The peak phase is 10 minutes long and therefore contains 600 points. Finally, the plateau phase is also 5 minutes long (300 points). This explains the differences in the amount of lines that are present in the plots of each phase. To ensure a coherent comparison between the intervals, the boundaries for the x-axes will be kept consistent among the intervals being compared. For each of the phases, the DC offset (see section 4.2.1) was present initially. However, that has also already been taken care of here.

In figure 5.8, the complete amplitude plot of the baseline can be seen. As described in section 2.4.1, this plot at first sight is symmetric due to the periodicity. Hence, it is alright to remove the second half, as shown in figure 5.9. From here on, only the first half of the plot will be shown as the amplitude spectrum. In the same plot, the frequencies with the greatest amplitude for baseline can be identified. Focusing specifically on the first 30 data points, as visible in 5.10, enables a better recognition of these frequencies. It is the range from 0.003 till roughly 0.01Hz that contains the largest amplitudes. Based on literature, frequencies that appear in the range between 0.003 and 0.005Hz are too small to be linked to a specific mechanism. As mentioned in 3.3, literature suggests that the mechanism Endothelium-Derived Hyperpolarizing Factor (EDHF) can be attributed to the frequency interval from 0.005Hz till 0.0095Hz. ([5], [10]).

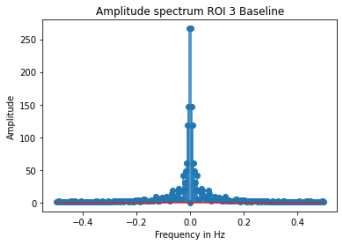

Figure 5.8: Baseline full

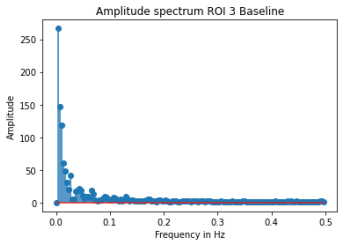

Figure 5.9: 1st half

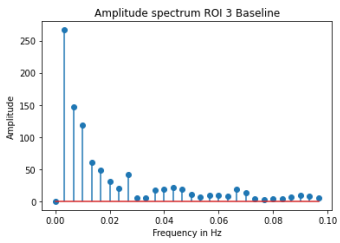

Figure 5.10: 1st 30 points

The amplitude spectra for the peak phase can be seen in figures 5.11 and 5.12. The dominating frequency is between roughly 0.0017 Hz and 0.003 Hz.
As mentioned before, these frequencies are too small to be linked to a specific mechanism. This phase is better suited for comparison with the peak phase of other regions of interest and/or for comparison with other phases from the same region of interest.

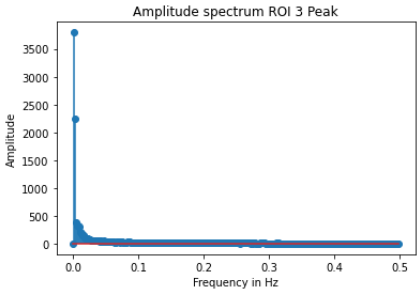

Figure 5.11: Peak phase

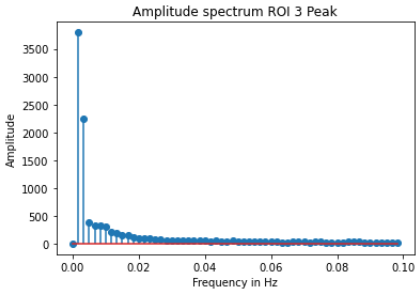

Figure 5.12: Peak phase, zoomed in

For the plateau phase, which can be seen in figures 5.13 and 5.14, the largest contribution seems to be from the frequency of 0.003 Hz. Overall, the dominating frequencies seem to lie in the interval between 0.003 Hz and 0.0167 Hz.
For the frequencies that go up and till 0.0095Hz, the behaviour can be explained as described in the previous phase. For frequencies that fall in the range between 0.0095 and 0.021Hz, the related mechanism is suggested to be nitric oxide mediated endothelial activity based on ACh and SP ([5] [10] and see section 3.3 for more details). In this particular phase, the frequencies between 0.01 and 0.0167 Hz could be related to the previously described activity.

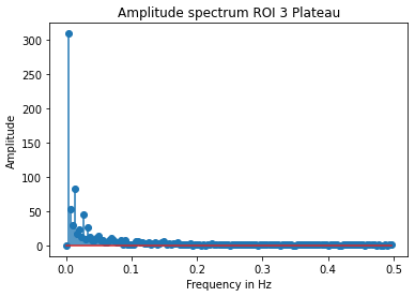

Figure 5.13: Plateau phase

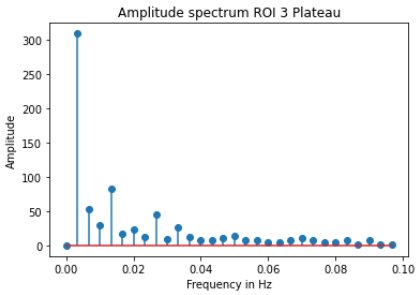

Figure 5.14: Plateau phase, zoomed in

5.6 Analysis LSI patient 1 (continued)

Firstly, it is important to organise some of the information discussed previously by defining the intervals for the upcoming plots. The previous sections in this chapter focussed on ROI 3 only. From this section on, all regions will be discussed and it will be clearly highlighted whenever that is not the case.
Here follows a quick refresher of the difference between the three (more details can be found in section 3.2): Nitric oxide is blocked in ROI 1, neuropeptides are blocked in ROI 2 and ROI 3 is the control region. As mentioned in section 3.3, neuropeptides will not be focused upon. Instead, the frequency interval found in literature for EDHF will be used to research whether any conclusions on its effects can be made.

Mechanism	Interval	Colour	Blocked in ROI
EDHF	0.005-0.0095Hz	Red	Nowhere
NO	0.0095-0.021Hz	Cyan	1

Table 5.1: Definition of mechanisms

Before discussing the images, the average values for NO and EDHF in each region of interest for the first 3 phases will be presented.

	Baseline	Peak	Plateau	SPEP	EPEP
EDHF	78.80	248.00	33.54	561.45	433.55
NO	45.83	98.46	36.29	229.93	211.21
	Baseline	**Peak**	**Plateau**	**SPEP**	**EPEP**
EDHF	88.35	102.38	36.73	318.10	241.19
NO	44.55	54.01	31.51	115.12	102.97
	Baseline	**Peak**	**Plateau**	**SPEP**	**EPEP**
EDHF	146.93	342.06	52.96	832.00	702.60
NO	64.58	174.46	38.07	332.21	249.60

Table 5.2: Values for ROI 1/2/3

In figure 5.15, the baseline phase for the three regions of interest can be seen. This phase is vital for comparison with the behaviour in other phases. Based on the images, the three regions do not differ significantly in the baseline phase. Table 5.2 shows that in ROI 3, the values for both mechanisms are higher compared to the other regions.

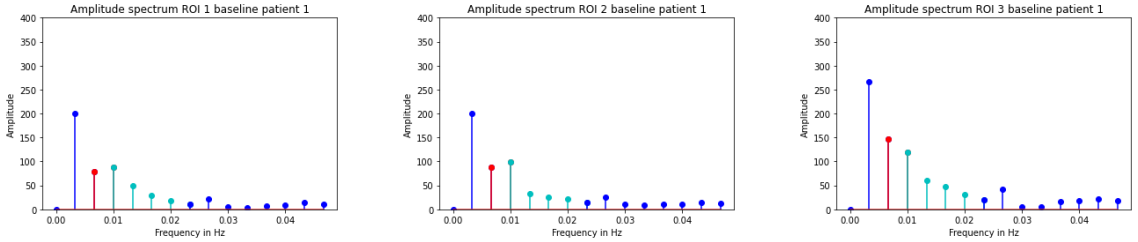

Figure 5.15: Patient 1 baseline phase

The peak phase is essentially the period in which the skin is gradually being heated to 40 degrees. Heating causes several mechanisms to start activating. At first sight, this behaviour might not be visible in figure 5.16. However, putting the baseline next to it and using the same scale shows this behaviour much better (see figure 5.20).
There are still some observations that can be made in figure 5.16. For example, the values

for EDHF and NO are smaller in ROI 1 and ROI 2 compared to ROI 3 (see also table 5.2). Moreover, the first two blue lines (that do not necessarily represent any mechanism) are significantly smaller.

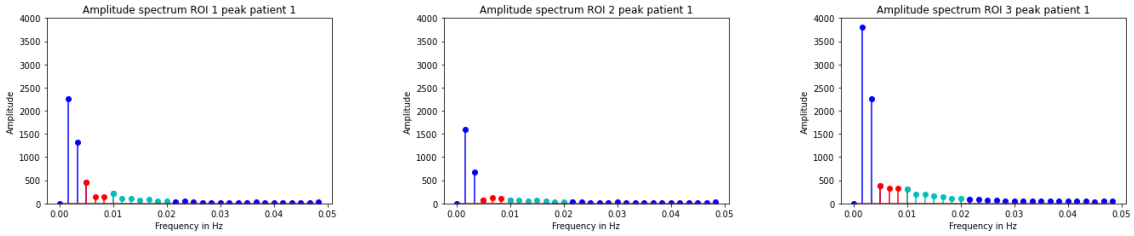

Figure 5.16: Patient 1 peak phase

In figure 5.17, the plateau phase across the three regions can be seen. The plateau phase is where the release of nitric oxide would be the most. Taking a look at the plots shows that there is no significant difference in the values of NO across the three regions. The averages given in table 5.2 are also between 31 and 36.
The peaks in ROI 1 (where NO has been blocked) and ROI 3 even appear to be indentical to some extent and according to table 5.2 the values for NO in both regions are very close to each other. Moreover, again the initial blue lines are significantly smaller in ROI 1 and 2 compared to ROI 3.

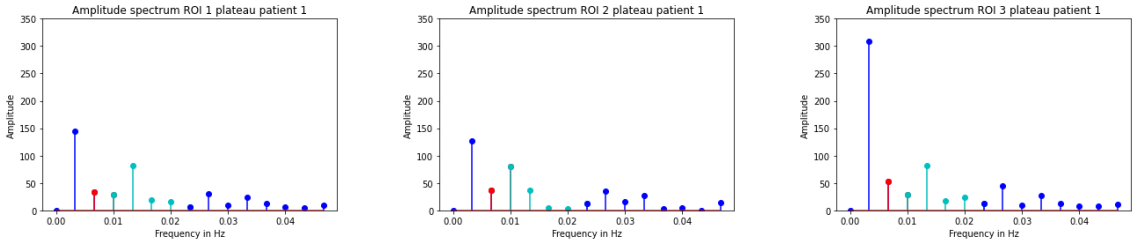

Figure 5.17: Patient 1 plateau phase

Since the plateau phase does not provide any conclusive information, the time interval is expanded to include the peak phase and the range between the peak and plateau phase (so from the start of the peak till the end of the plateau, shortened as SPEP). This can be seen in figure 5.18. It is important to first note that in literature, the effect of or increase in the release of NO after local thermal heating is expected to appear after at least 30 minutes after the heating was started. Hence, from that point of the view, figure 5.17 would be the most important.
There are some differences among each individual plot, the most notable one being ROI 2 (where the neuropeptides were blocked) which appears to have smaller amplitudes in every section. Also, the blue lines at the beginning do not have an outlier compared to ROI 1 and ROI 3. EDHF seems to be a lot less too. However, the difference between ROI 1 and ROI 3 for NO is not noteworthy here either.

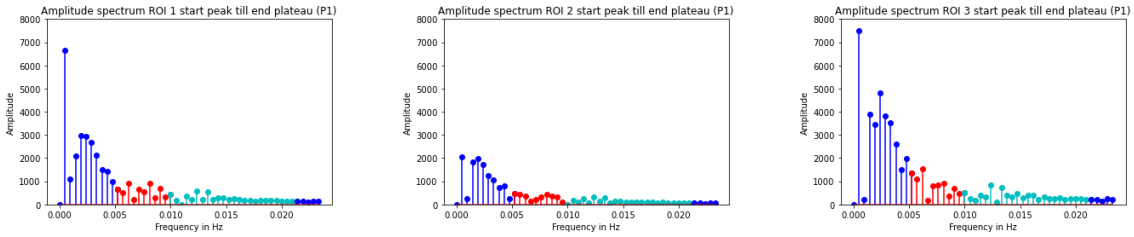

Figure 5.18: Patient 1 from start of peak till end of plateau

In figure 5.19, a smaller interval is considered. The peak phase is left out, as that is the part where the skin is still being heated (so from the end of the peak till the end of the plateau, shortened as EPEP). It can be argued that this region is still too big, however it still might contain some interesting observations.
Firstly, the overall behaviour of the three regions appears to mimic the behaviour seen in figure 5.18. ROI 2 again exhibits the most different behaviour and EDHF is significantly reduced compared to both ROI 1 and ROI 3. The initial blue lines again do not contain outliers.
Overall, by table 5.2, it can be confirmed that ROI 1 on average has smaller amplitudes for EDHF in all the different phases compared to ROI 3. The shape of the blue lines in ROI 1 does show some similarity with the blue lines in ROI 3. Moreover, after 0.015Hz the NO amplitudes in ROI 1 seem to be smaller than the ones in ROI 3.

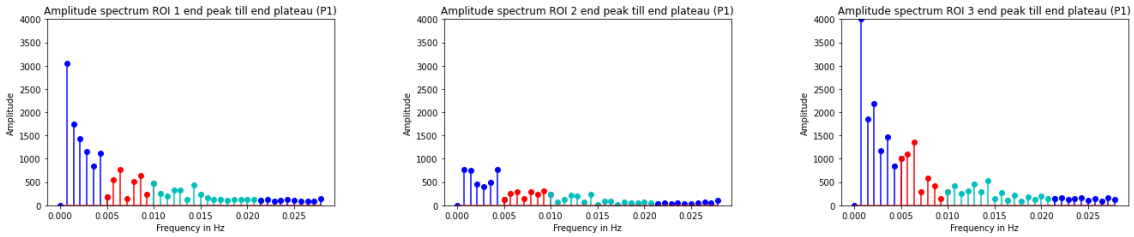

Figure 5.19: Patient 1 from end of peak to end of plateau

Figure 5.20 shows the most interesting plots from ROI 3 side by side. The peak phase shows increased amplitudes for all the sections. The plateau phase would be where increased release of NO is expected. However, that is not visible here. Table 5.4 shows that the value for NO in the plateau phase of ROI 1 is 36.29, whereas in ROI 3 it is 38.07. So, very slightly higher in ROI 3 compared to ROI 1 (in which NO was blocked). Again based on table 5.4, both EDHF and NO have a much smaller amplitude in the plateau phase compared to the baseline.

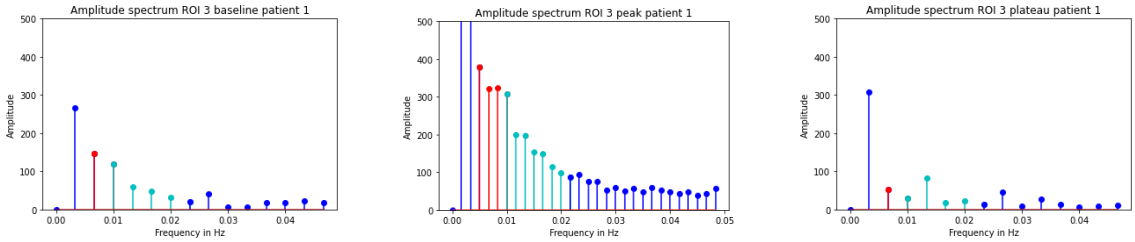

Figure 5.20: Patient 1. Base Peak Plateau

5.7 Analysis LDI patient 1

Similar to the decomposition of LSI, the LDI measurement can be split into the three phases of interest (baseline, peak and plateau).
No smoothing was done for this measurement, just the DC offset was removed. Smoothing after the removal would be redundant. The amplitudes in the peak phase are much higher, as visible in 5.21, behaviour that is similar to the LSI measurement. However, overall behaviour of ROI 1 and ROI 3 is quite similar here as well.

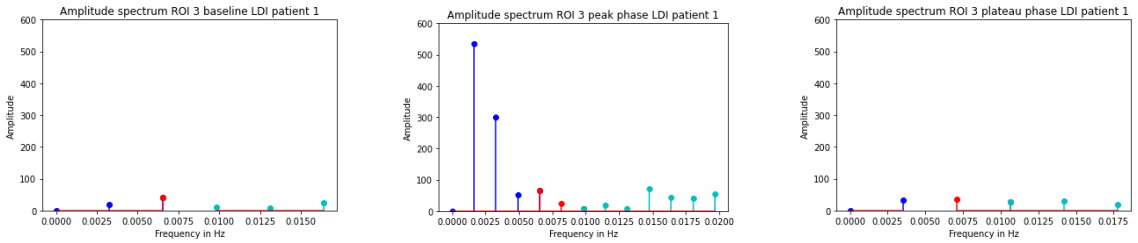

Figure 5.21: Patient 1. LDI. Base Peak Plateau

As mentioned previously, these visualisations show much more variation in the frequency amplitudes due to the smaller size of the set. Figure 5.22 shows the variations (after zooming in on the baseline and plateau). However, the LDI measurement will not be used from hereon since it does not offer much more useful information. Moreover, due to the Nyquist rate, the only mechanisms that can be reliably considered are the ones that are below 0.02Hz (which is half of the sampling frequency of the LDI measurement). Therefore, the LDI measurements will not be considered for future patients.

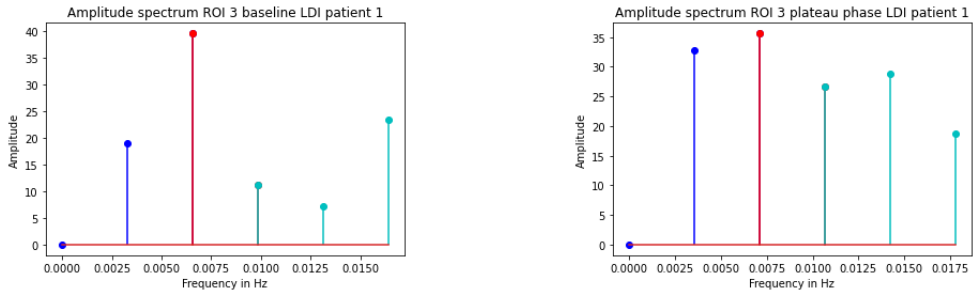

Figure 5.22: Zoomed in baseline and plateau.

5.8 To summarize

Given a data set, some processing is required before obtaining useful results. This includes, but is not limited to, smoothing of the data set and DC offset removal.
For every phase of the LSI measurement, dominating frequencies and/or frequency intervals can be obtained based on literature. This can then be used to analyse the behaviour of the mechanisms in every part of the measurement.
The first measurement appears to show quite varying and counter-intuitive behaviour. The next chapter includes the decomposition of the LSI measurements of two more patients.
The idea to consider larger intervals (SPEP and EPEP) did not offer additional insights and therefore shall not be considered from here on, just like the LDI measurements.

Chapter 6

Results patients 2 and 3

6.1 Introduction

For patients 2 and 3, only the peak and plateau phases will be discussed as the additional phases from the last chapter were not a significant addition. Finally, there will be a scaled comparison between the baseline phase and the peak and plateau phases in ROI 3.

6.2 LSI patient 2

Before discussing the plots, the average values for EDHF and NO in the three phases of all regions of interest will be shown.

	Baseline	**Peak**	**Plateau**
EDHF	123.97	159.50	85.66
NO	42.63	41.78	39.20

	Baseline	**Peak**	**Plateau**
EDHF	140.39	65.56	170.84
NO	51.86	38.43	25.08

	Baseline	**Peak**	**Plateau**
EDHF	150.39	272.55	144.04
NO	58.09	102.92	37.08

Table 6.1: Values for ROI 1/2/3

Starting with the peak phases, as seen in figure 6.1, the blue lines in the beginning appear to retain their behaviour despite decreasing in amplitudes in both ROI 1 and ROI 2 (compared to ROI 3). ROI 2 (where the neuropeptides were blocked) overall shows smaller amplitudes for both mechanisms. For ROI 1 (where NO was blocked), this holds largely too.
In the plateau phase of ROI 1, where NO is being blocked, EDHF decreases significantly compared to NO.

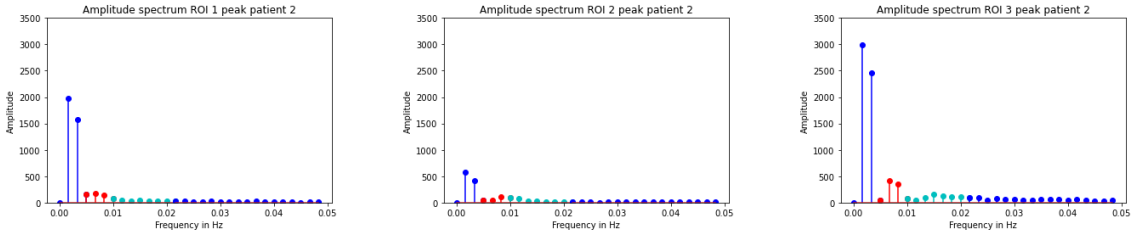

Figure 6.1: Patient 2 peak phase

The plateau phases shown in figure 6.2 across the three regions all show 3 larger peaks at the beginning, which is a new behaviour compared to patient 1. Again, the difference between NO in ROI 1 and ROI 3 does not appear to be significant and in ROI 2 the amplitude values for it are much lower. EDHF is lower in ROI 1, whereas in ROI 2 it is much higher. This behaviour is again confirmed by table 6.1.

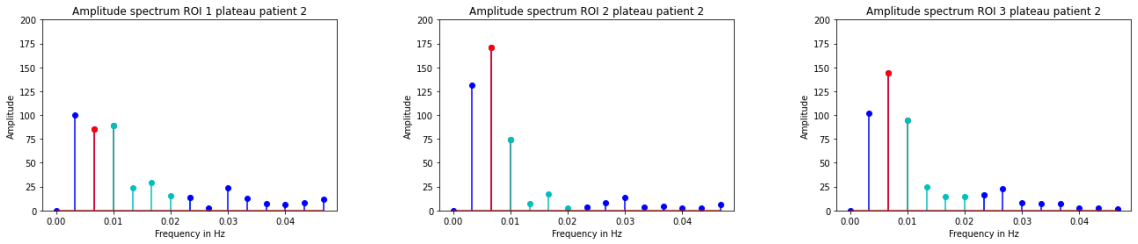

Figure 6.2: Patient 2, plateau phase

In 6.3, a comparison between the three phases in ROI 3 results in similar conclusions as patient 1. EDHF retains similar amplitudes in the baseline and plateau phases (see table 6.1) and NO decreases in the plateau phase compared to the baseline.

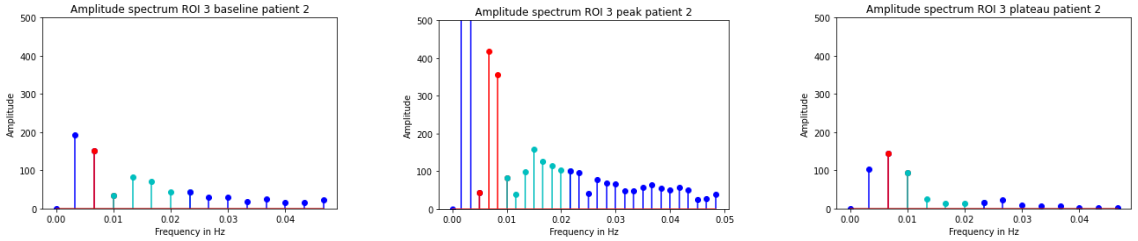

Figure 6.3: Patient 2. Base Peak Plateau

6.3 LSI patient 3

Two patients might not be enough to properly draw conclusions. Therefore, another measurement will be discussed broadly.
Again, before discussing the plots, there will be an overview of the average values first.

	Baseline	**Peak**	**Plateau**
EDHF	151.82	197.90	35.91
NO	65.86	150.27	32.70
	Baseline	**Peak**	**Plateau**
EDHF	103.99	171.42	56.61
NO	62.12	73.27	34.24
	Baseline	**Peak**	**Plateau**
EDHF	197.44	630.34	25.85
NO	74.27	269.42	50.57

Table 6.2: Values for ROI 1/2/3

For patient 3 in figure 6.4, in ROI 2 the blue outliers that are visible in ROI 1 and 3 are immediately much lower in the peak phase already. This is different compared to the previous two patients, where the amplitudes for the blue lines in the peak phase were proportionally smaller for both ROI 1 and 2.

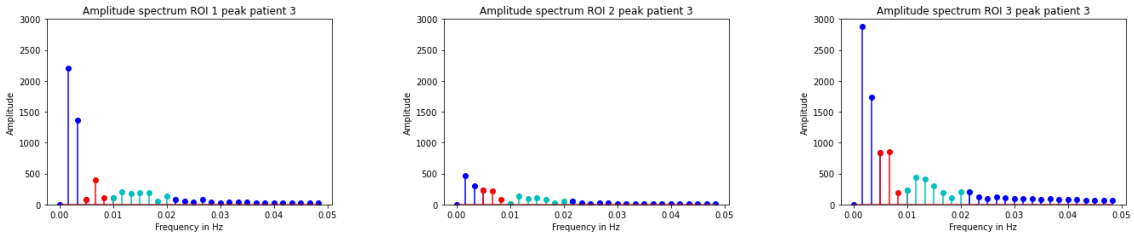

Figure 6.4: Patient 3 peak phase

In the plateau phase shown in figure 6.5, the amplitudes of the different intervals in ROI 1 and ROI 3 are quite similar. The NO related amplitudes, however, are lower in ROI 1 (32.70) compared to ROI 3 (50.57). In the same way as the previous patient, the value of EDHF is much higher in ROI 2 compared to the other two phases.

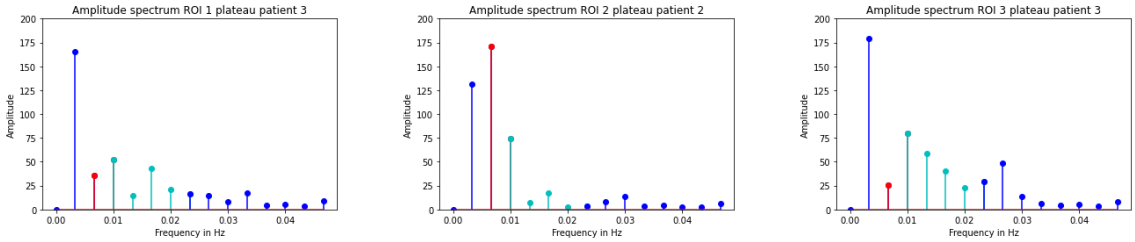

Figure 6.5: Patient 3, plateau phase

Finally, figure 6.6 again confirms that in the peak phase, all the mechanisms start being active and have larger amplitudes. Contrary to what the literature suggests, the plateau phase again contains decreased values of the amplitudes of all mechanisms.

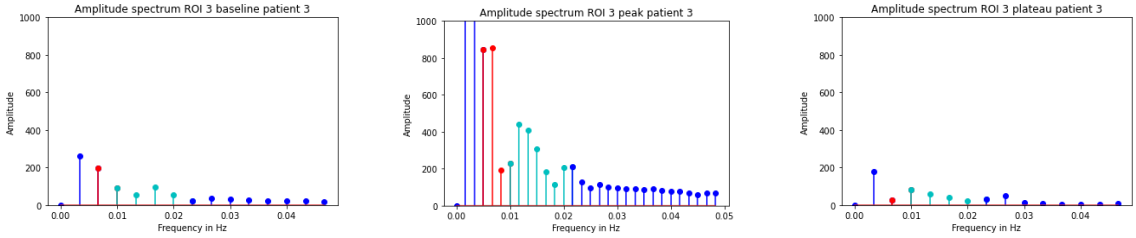

Figure 6.6: Patient 3. Base Peak Plateau

6.4 Conclusion

Overall, the behaviour of the patients seems to be a bit unexpected. The three patients together provide a large amount of material for interesting comparisons.
The next chapter contains an overview of 10 new patients.

Chapter 7

Global observations 3rd set of patients.

7.1 Introduction

This chapter broadly discusses a totally new different set of 10 patients, which will be referred to as the 3rd set of patients (as it was the 3rd set to be received). The corresponding images can be found in appendix A. It is important to note that the y-axes have the same scale per phase (which was also the case for the three previous patients). For example, the y scale for the baseline phase for all regions of interest for patient 1 is 200. This is different from the y scale used for the plots of the peak phases for patient 1.

7.2 Observations

Overall, the baseline plots are quite useful for comparisons with the other phases which is similar to the method used for the first 3 patients. Another point worth noting is that the y limits for the peak phases are mostly higher, which is consistent with the activation of several mechanisms after local thermal heating.
For patients 2 and 7, EDHF seems to be present much more strongly in the baseline compared to NO. For the remaining patients, EDHF and NO do not seem to differ as significantly as in patients 2 and 7.

In the peak phase, the behaviour of all mechanisms seems to go into several directions in all regions of interest. For most patients, the values for both mechanisms in ROI 3 are the highest, which aligns with the expectation since ROI 3 is being measured under standard circumstances. It is difficult to say more than this, as there is a strong variation amongst the results for the peak phase all over.
One remark can be made about the behaviour of the blue outlying lines. For many patients, these lines seem to behave in a more restricted way in ROI 2 compared to ROI 1 and 3.

NO should have a stronger amplitude in the plateau phase of ROI 3 compared to ROI 1 (where this mechanism is being blocked). Looking at the average values in appendix C, this seems to be the case for all patients except for patient 10. In patient 6, NO is significantly stronger in ROI 3 than ROI 1.

7.3 Conclusion

All in all, the behaviour of the mechanisms throughout the different phases and different regions of interest does show rather large variations. It is difficult to find some significant pattern or points of agreement throughout all of the patients. Certain behaviours are confirmed, such as the increased values for all mechanisms in the peak phase and the slightly higher values for NO in the plateau phase of ROI 3 compared to ROI 1.

Chapter 8

Conclusion and discussion

8.1 To what extent do the mechanisms differ across the phases and regions

The values for NO seem to stay in similar ranges for both the baseline and plateau phases in ROI 3. It would be reasonable to not see much activity for NO in the baseline phase as the local thermal heating has not yet started. However, the release of NO is expected in the plateau phase. For that reason, the values for NO in ROI 1 (where NO is blocked) can be compared to ROI 3. Again, for most of the 13 patients the values stay in similar ranges. However, it is interesting to note that for patient 1 (chapter 5) and patient 2 (chapter 6) the values for NO in the plateau phase in ROI 3 seem to be the lowest.

For EDHF, there seems to be some interesting behaviour too. The baseline phase contains more EDHF in ROI 3 than in the plateau phase of ROI 3 for the largest part of all 13 patients. Looking at a comparison of the baseline amongst the regions of interest in 5.15 (note: the scale of the baseline plots in section 5.5 is different than the scale used for the plots in figure 5.15), the value for EDHF seems to be quite stable. EDHF could simply be a mechanism that is more present than other mechanisms during standard circumstances.

Both mechanisms do have stronger amplitudes in the peak phases across the different regions, which aligns with the expectation of the peak phase. This is the phase in which the local thermal heating of the skin is started and therefore the mechanisms all start activating.
For some patients, in the peak phases there is difference visible between ROI 1 and 2 compared to ROI 3. These patients show overall smaller values in peak phases of ROI 1 and 2 compared to the peak phases of ROI 3, which aligns with the fact that in ROI 1 and 2 certain mechanisms have been blocked.

ROI 2 was the region in which several neuropeptides are blocked, which also contribute to the dilation of the vessels. Generally ROI 2 seems to show lower values in the peak phase. There are even a few patients for which the blue lines and overall peaks for the two mechanisms in the peak phase are much smaller compared to the other two regions.

Generally, it appears that all patients that have been analysed so far show largely varying results in the important sections of the measurements. Moreover, the differences between NO do not appear to be as strong as expected and sometimes even smaller amounts of NO appear in regions where they would not be expected to be smaller (such as in ROI 3).

8.2 CGRP

Calcitonin gene-related peptide (CGRP) is one of the neuropeptides that was blocked in ROI 2. Due to its relation to cardiovascular disease and migraine, this is an especially interesting neuropeptide to research using Fourier Transform. However, it appears that so far there are no LSI or LDI generated measurements available that can be used for further analysis. The existing data consists of EEG measurements (for example), which is outside the scope of this thesis. There are also other methods that have been used to research the relationship with migraine, however those methods are entirely different as well.

8.3 Relation of the sampling frequency with the maximum attainable frequency bands

The sampling frequency of the LSI data sets is roughly around 1Hz for every set. Which means that, due to the Nyquist-Shannon sampling theorem, the largest accurate frequency that we can obtain reliably from the data sets is 0.5Hz. This in turn means that we can reliably research the mechanisms that have their range below that number. For LDF measurements, the Nyquist rate is even lower (namely 0.02 Hz). For the mechanisms that are being studied in this thesis, this number is near the border. Moreover, due to the results shown for the first patient, the LDI measurements for the other patients were left out.

8.4 Other methods to filter the data

There are several methods to filter the data and to make sure reliable results can be obtained. To this end, some window functions were tried on ROI 3 of patient 1. In appendix B, these window functions and the corresponding results are discussed in more detail. The main conclusion here is that for these measurements, the use of window functions does not appear to add anything to the existing analysis.

8.5 Recommendations

For the future, it would be interesting to firstly finish processing all the measurements using Fourier Transform. Moreover, there are still some ideas that can be explored in the direction of window functions.
Alternatively, Wavelet Decomposition would be a strong method to analyse these measurements with (as was proposed in the project description). Several studies (such as [5]) use Wavelet Decomposition to obtain the frequency intervals that relate to the mechanisms. Perhaps even the LDI measurements could be processed using the Wavelet Decomposition. Another possibility would be to try to obtain frequency bands related to CGRP using Wavelet Decomposition.

Appendix A

Plots related to the 3rd data set

A.1 Baseline

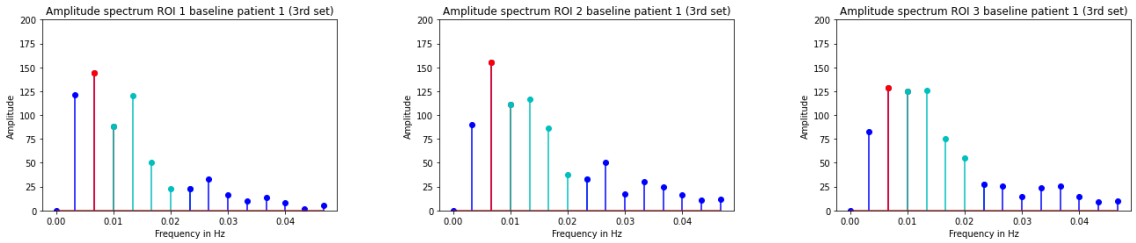

Figure A.1: Patient 1, baseline phase

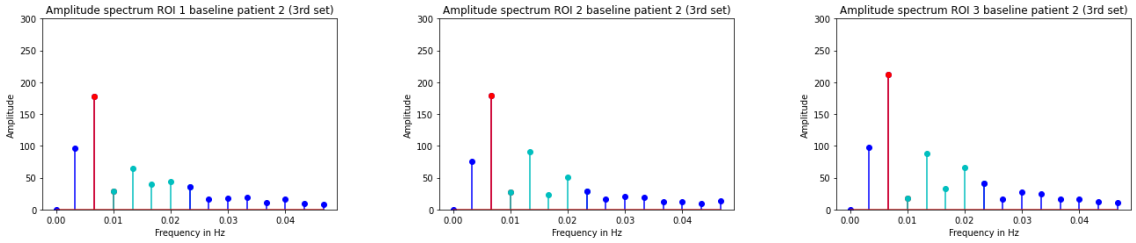

Figure A.2: Patient 2, baseline phase

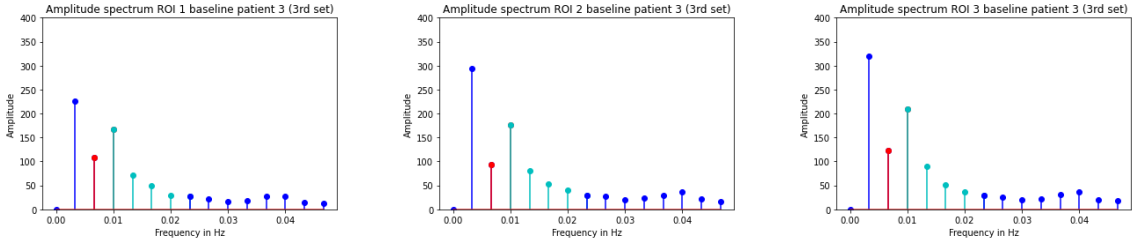

Figure A.3: Patient 3, baseline phase

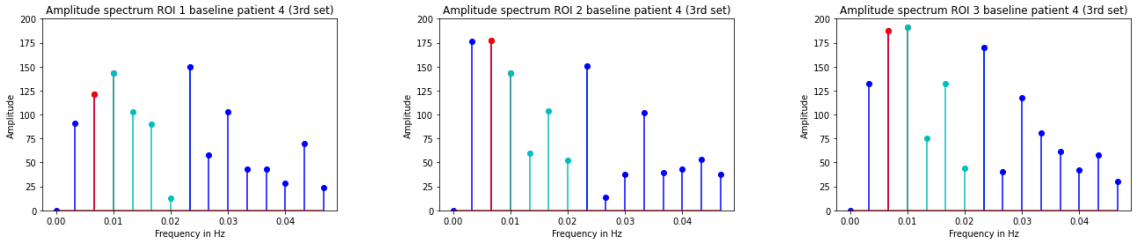

Figure A.4: Patient 4, baseline phase

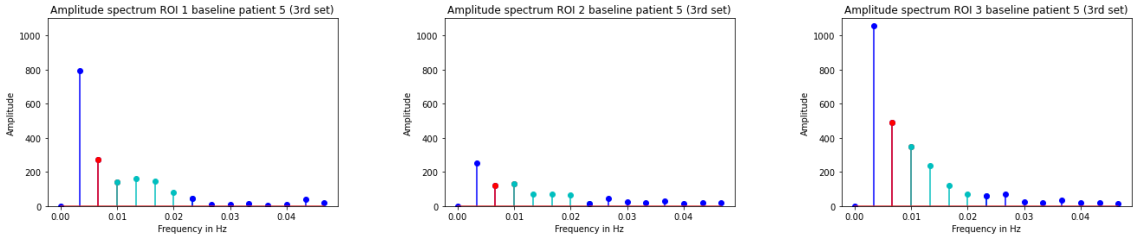

Figure A.5: Patient 5, baseline phase

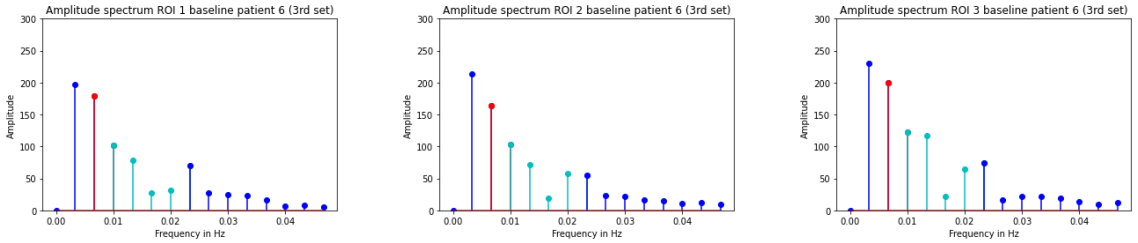

Figure A.6: Patient 6, baseline phase

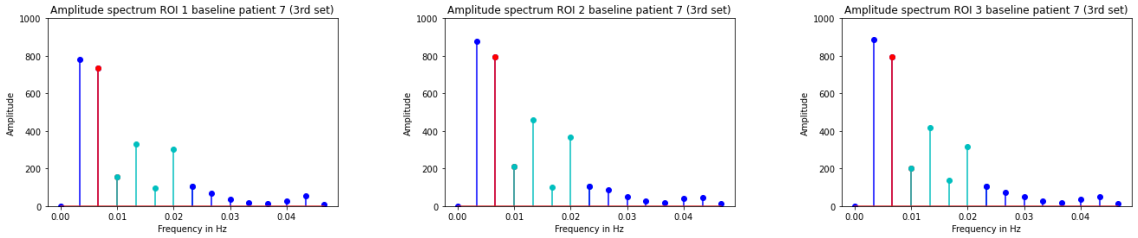

Figure A.7: Patient 7, baseline phase

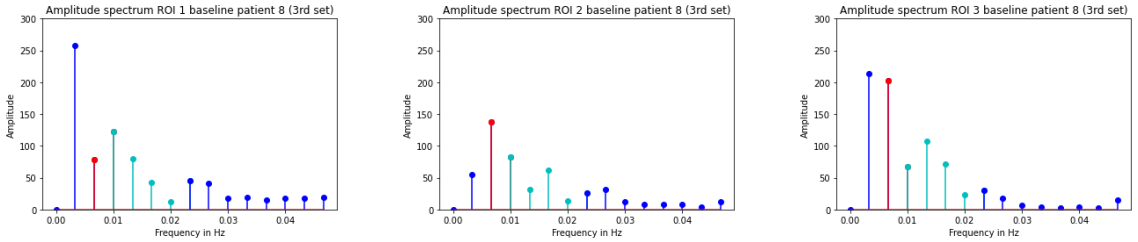

Figure A.8: Patient 8, baseline phase

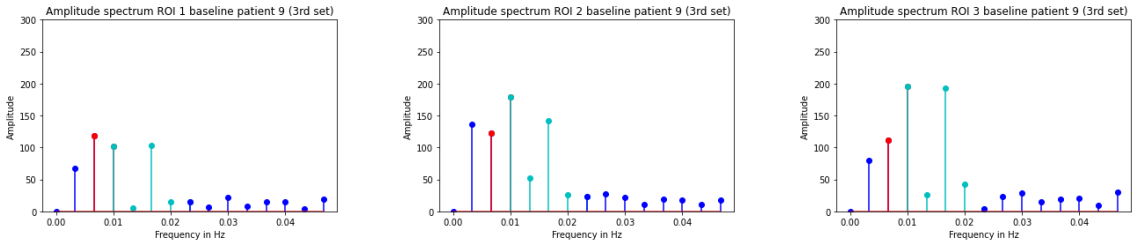

Figure A.9: Patient 9, baseline phase

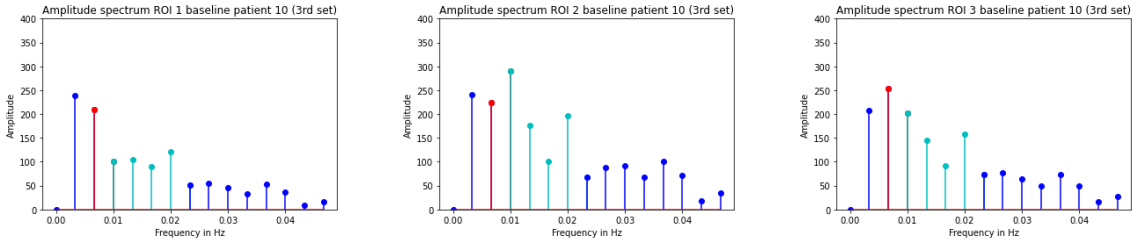

Figure A.10: Patient 10, baseline phase

A.2 Peak

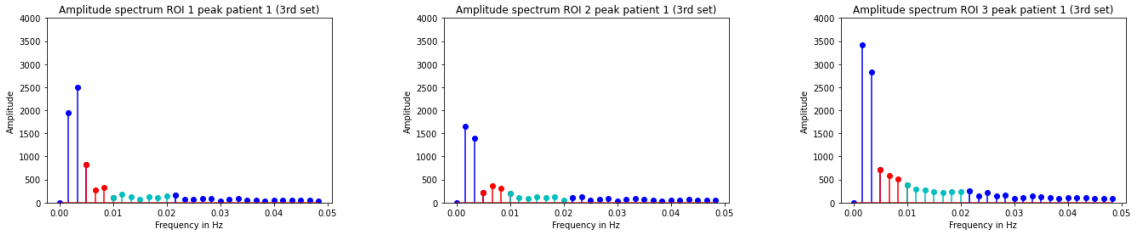

Figure A.11: Patient 1, peak phase

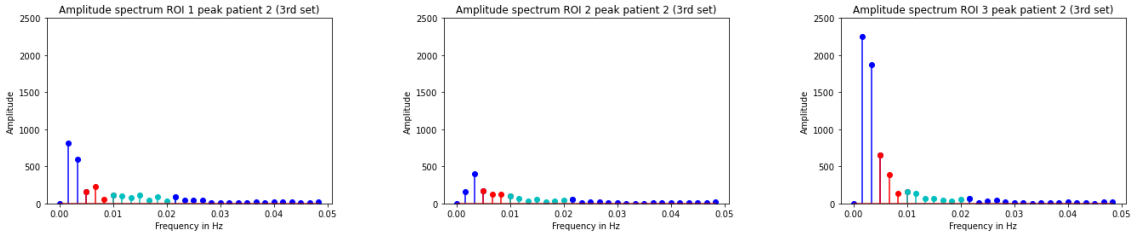

Figure A.12: Patient 2, peak phase

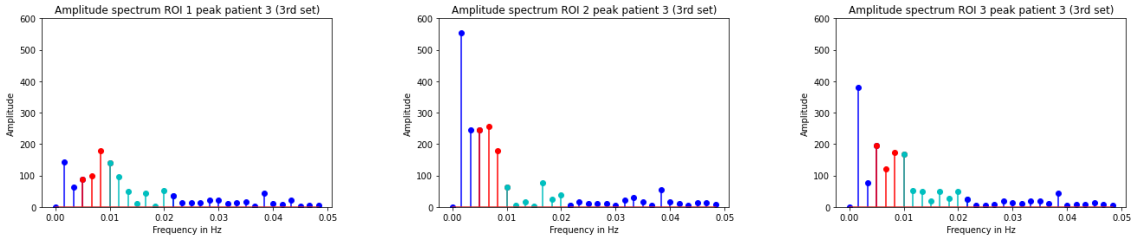

Figure A.13: Patient 3, peak phase

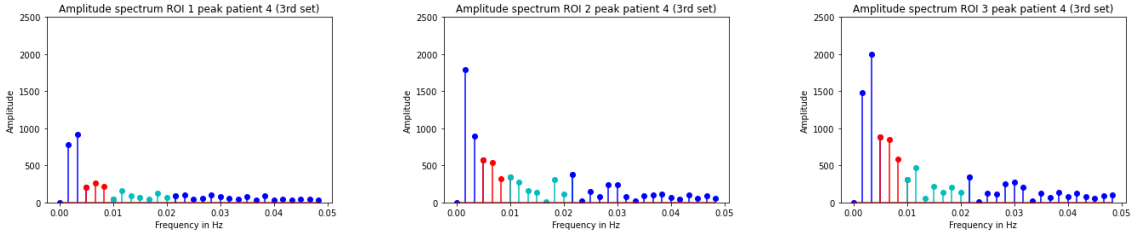

Figure A.14: Patient 4, peak phase

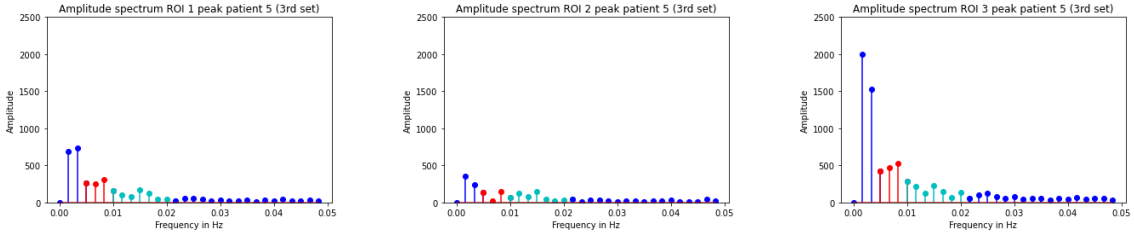

Figure A.15: Patient 5, peak phase

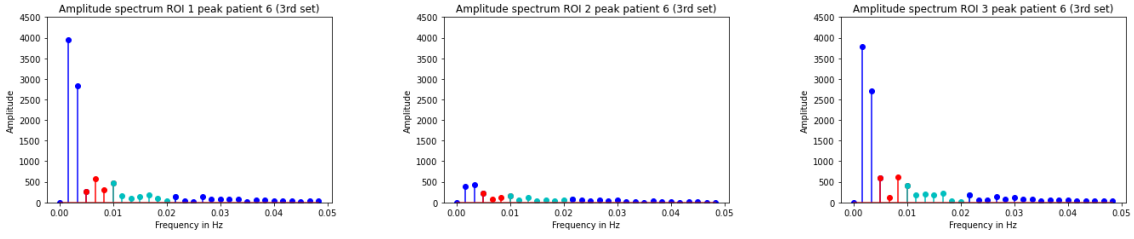

Figure A.16: Patient 6, peak phase

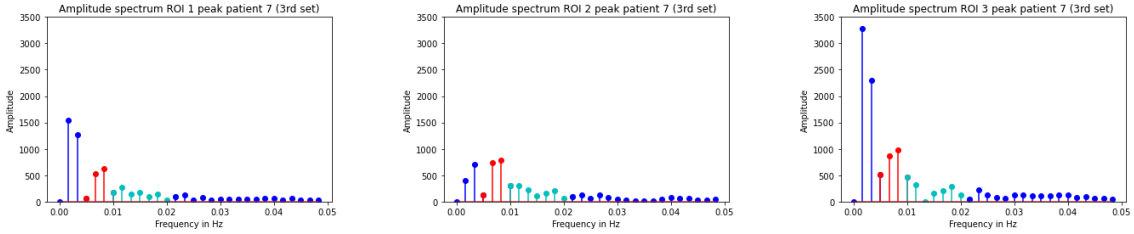

Figure A.17: Patient 7, peak phase

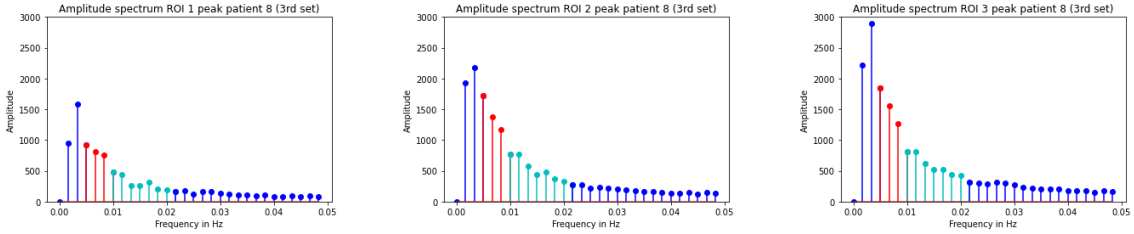

Figure A.18: Patient 8, peak phase

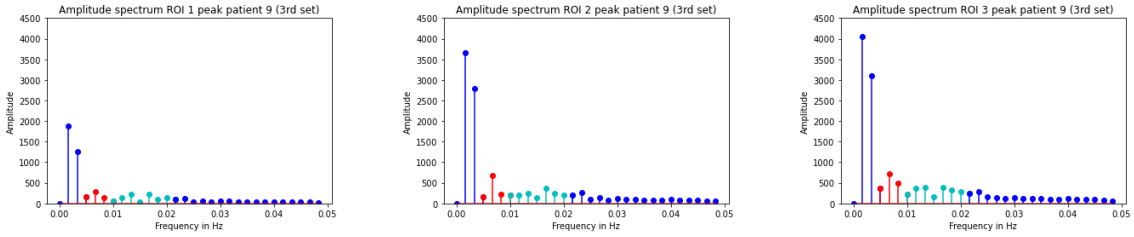

Figure A.19: Patient 9, peak phase

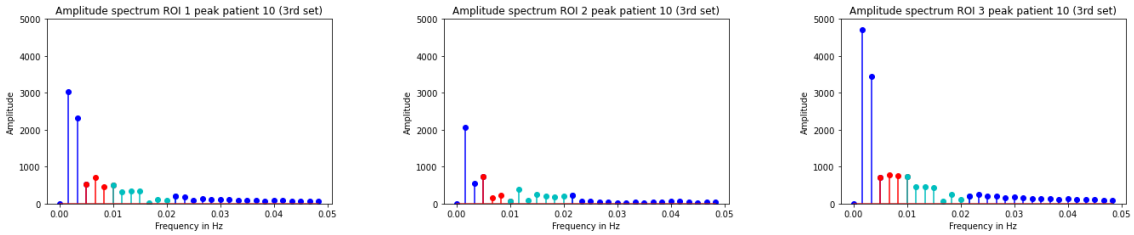

Figure A.20: Patient 10, peak phase

A.3 Plateau

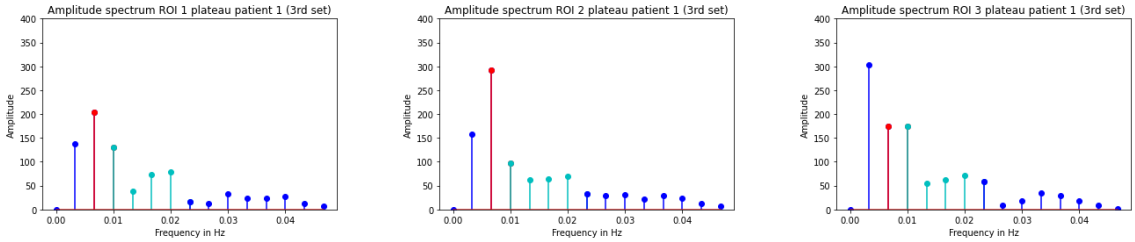

Figure A.21: Patient 1, plateau phase

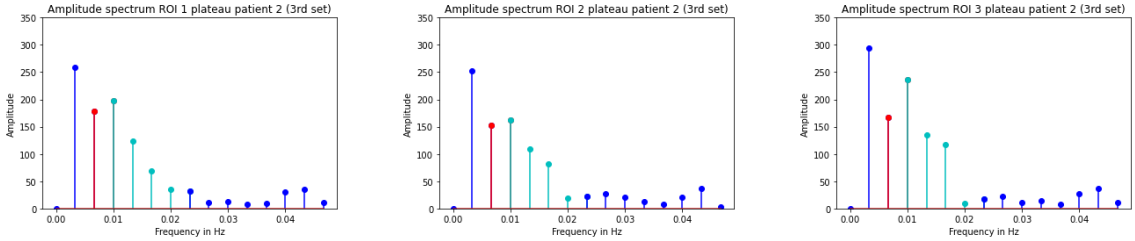

Figure A.22: Patient 2, plateau phase

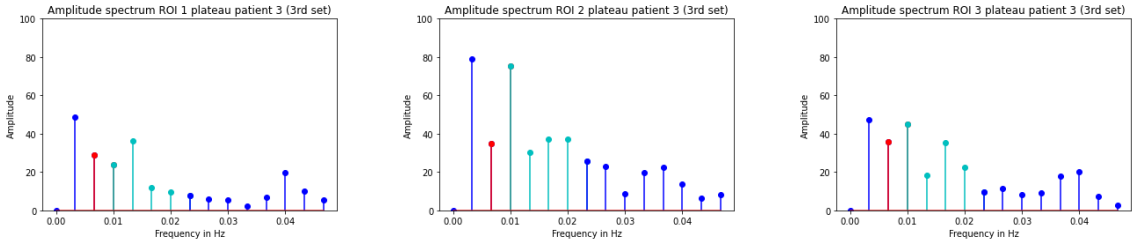

Figure A.23: Patient 3, plateau phase

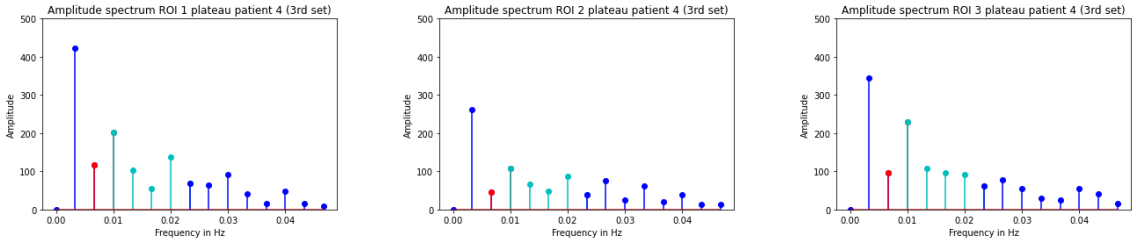

Figure A.24: Patient 4, plateau phase

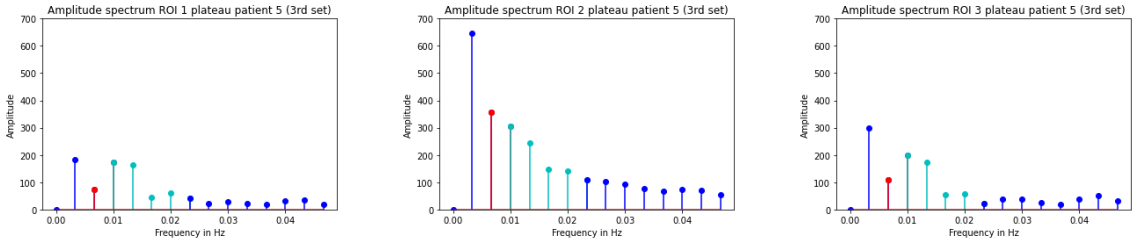

Figure A.25: Patient 5, plateau phase

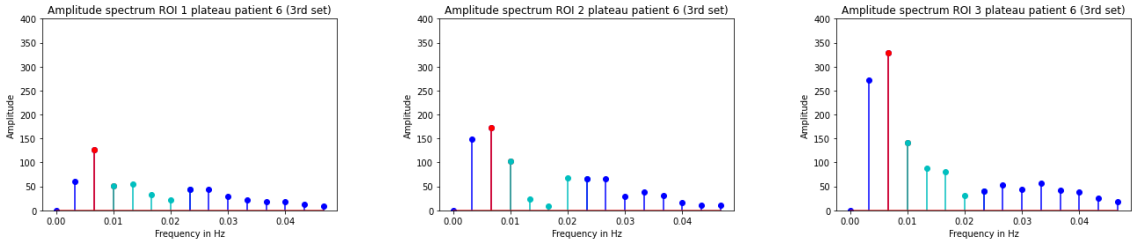

Figure A.26: Patient 6, plateau phase

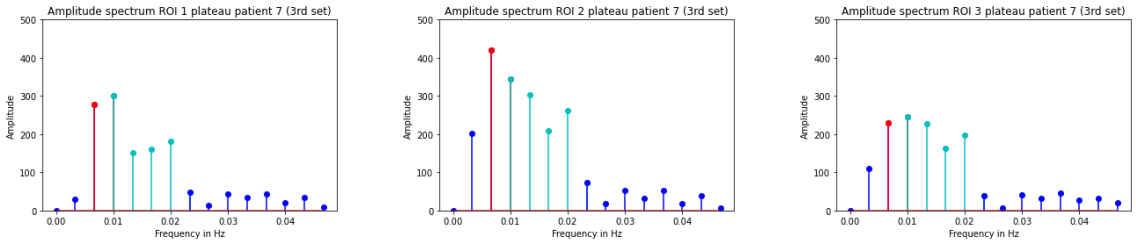

Figure A.27: Patient 7, plateau phase

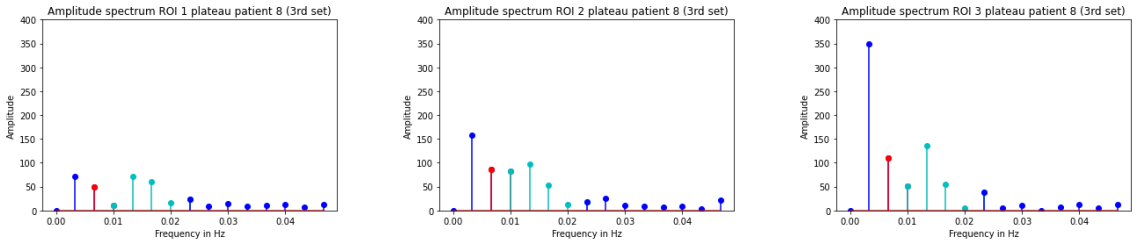

Figure A.28: Patient 8, plateau phase

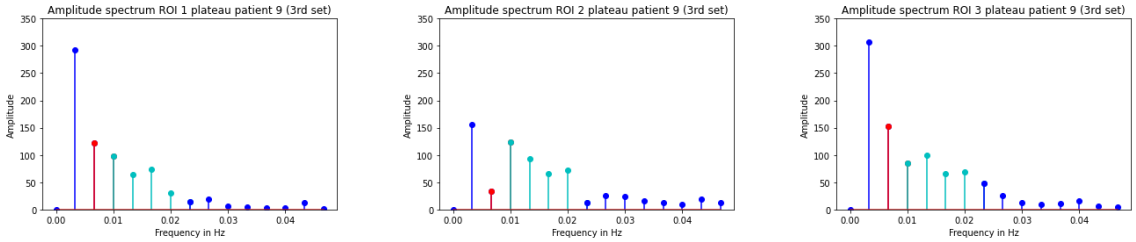

Figure A.29: Patient 9, plateau phase

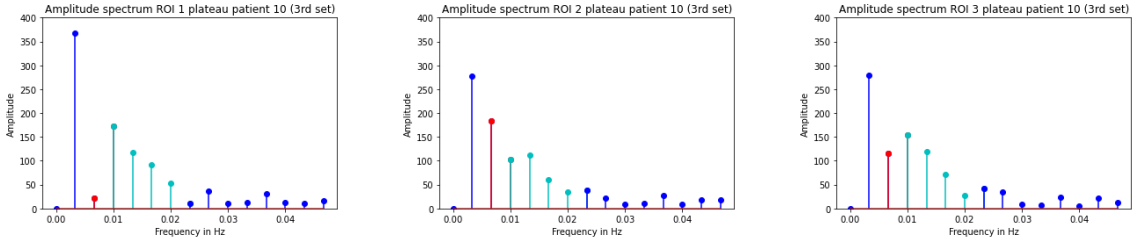

Figure A.30: Patient 10, plateau phase

Appendix B

Window functions

As it has been established that the frequency intervals of interest range between 0.005 and 0.021 Hz, one could consider using certain window functions in order to more specifically filter out these frequencies or to zero out certain parts of the signal.
Window functions attenuate a specific part of the frequencies whereas some other pre-defined interval of frequencies is let through. This can be defined by the user as their needs might require. If the user requires a filter that allows the lower frequencies and attenuates the higher frequencies, they could use a low-pass filter. Similarily, there are high pass filters. Finally, there are also band-pass filters, which allow frequencies from a pre-defined interval and attenuate the rest. It is possible to apply filters before applying the Fourier Transform, but there are also some filters that are used in the frequency domain. [11]
Since the LSI and LDI measurements do not start measuring from 0, it is possible to use a window function to make the measurement behave like it starts and ends at 0.

B.1 Hanning

The first window function to be tried is the Hanning function. [12] The Hann window is defined as:

$$w[n] = 0.5 - 0.5\cos\frac{2\pi n}{N}, \;\; 0 \leq n \leq N \tag{B.1}$$

where N is the length of the signal. The application of this function is shown in figure B.1 where the original ROI 3 signal for patient 1 is also visible for comparison.
In figure B.2, the plateau phase of the Hanning signal can be seen. Here, neither the DC offset has been removed nor has the Hanning signal been smoothed.
Figure B.3 shows the plateau phase of the Hanning signal where only the DC offset has been removed and figure B.4 shows the phase when it has only been smoothed with the same factor as was used for the regular ROI 3 of patient 1.
Figure B.5 shows the phase after DC offset removal and smoothing, and next to it in figure B.6 the original ROI 3 is shown (after smoothing and DC offset removal) for comparison. There does not seem to be a huge difference between variations of the plateau phase. It would be alright to continue without the use of any window function.

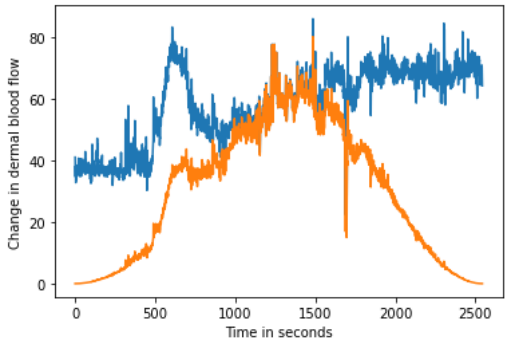

Figure B.1: Original and Hanning

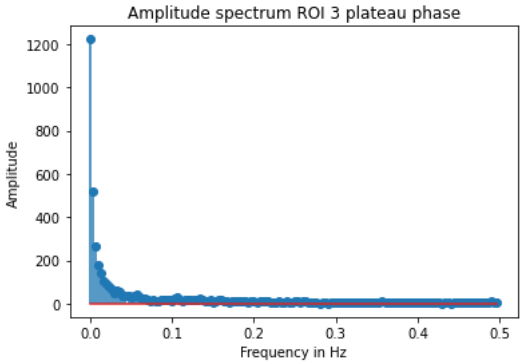

Figure B.2: Plateau phase Hanning, basic

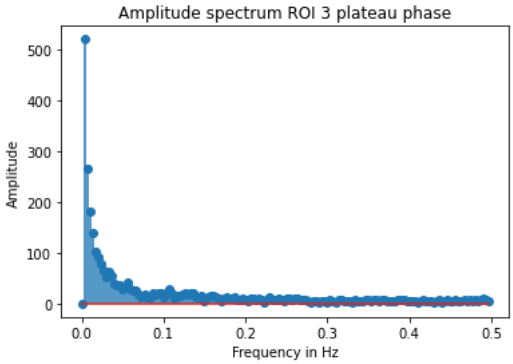

Figure B.3: Hanning DC removed

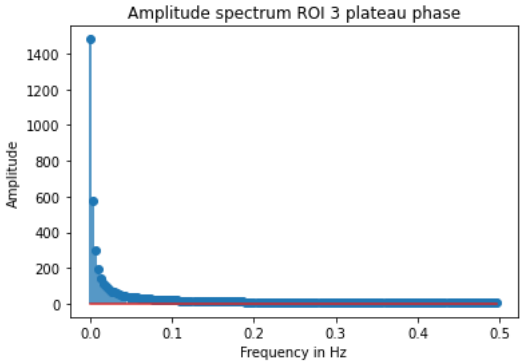

Figure B.4: Hanning smooth

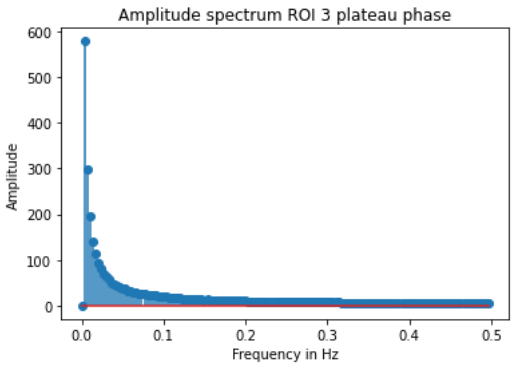

Figure B.5: Hanning smooth and DC removed

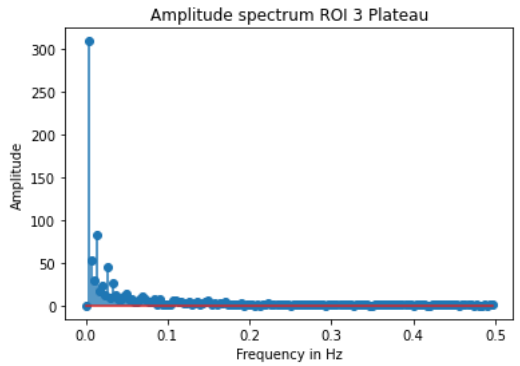

Figure B.6: Original smooth and DC removed

B.2 Hamming

Related to the Hanning window function is the Hamming window function. This filter uses the value 0.54 for the first coefficient in equation B.1 and 0.46 in the second coefficient. [11]
Judging by figure B.7, it is clear that there is no significant difference in the application of Hanning and Hamming windows.

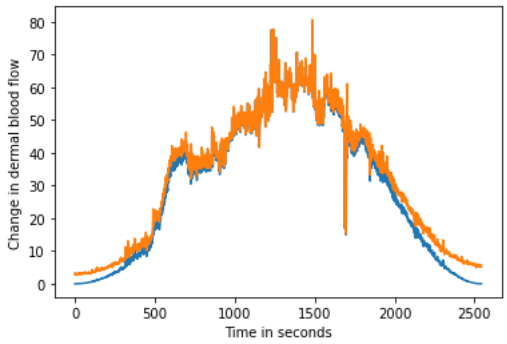

Figure B.7: Hanning and Hamming

B.3 Butterworth

Finally, the Butterworth filter was also tried. The implementation used for the LSI measurement is a low-pass filter. [13] [14] This filter does not require any smoothing, however the DC offset was still removed. Moreover, several number of orders and cutoff frequencies were tried. For complex data sets, higher orders are required. For the LSI measurement order 5 was finalized. Moreover, the cut off frequency (so the frequencies below this number are attenuated) was set to 0.005. The results can be seen in the next figures. In figure B.8, the Butterworth curve follows the actual measurement almost exactly and the curve itself is smooth. However, for the plateau phase again it does not create a huge difference.

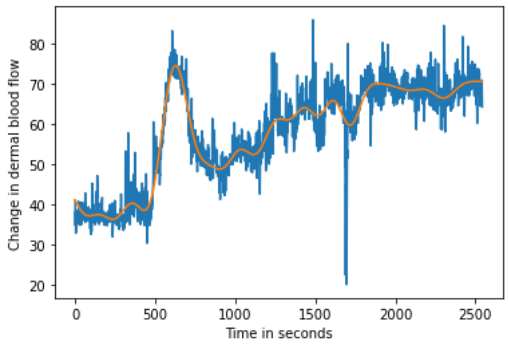

Figure B.8: Butterworth compared to original

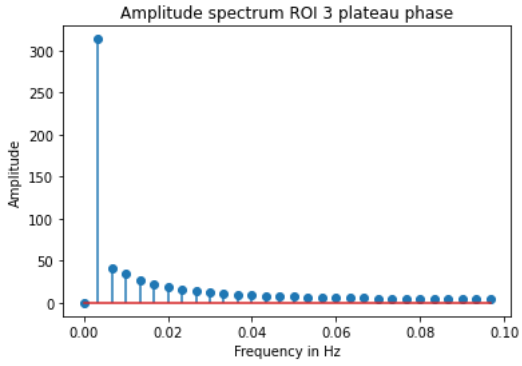

Figure B.9: Butterworth plateau DC removed

Appendix C

Average values NO and EDHF in baseline, peak and plateau 3rd set of patients

C.1 Patient 1

	Baseline	**Peak**	**Plateau**
EDHF	144.68	473.88	203.16
NO	70.38	120.46	80.40

	Baseline	**Peak**	**Plateau**
EDHF	155.13	297.82	292.36
NO	87.84	114.96	73.03

	Baseline	**Peak**	**Plateau**
EDHF	128.77	603.61	173.51
NO	95.33	268.86	90.88

Table C.1: Values for ROI 1/2/3

C.2 Patient 2

	Baseline	**Peak**	**Plateau**
EDHF	177.88	149.12	178.74
NO	44.31	82.83	106.05

	Baseline	**Peak**	**Plateau**
EDHF	179.49	139.96	153.27
NO	48.07	49.64	93.30

	Baseline	**Peak**	**Plateau**
EDHF	212.05	393.45	167.544
NO	51.26	80.26	124.49

Table C.2: Values for ROI 1/2/3

C.3 Patient 3

	Baseline	Peak	Plateau
EDHF	108.35	121.74	28.94
NO	78.83	56.65	20.48

	Baseline	Peak	Plateau
EDHF	93.96	226.49	34.77
NO	87.76	32.23	44.91

	Baseline	Peak	Plateau
EDHF	123.32	162.50	35.97
NO	96.22	58.71	30.26

Table C.3: Values for ROI 1/2/3

C.4 Patient 4

	Baseline	Peak	Plateau
EDHF	121.57	230.17	116.27
NO	87.20	87.15	123.46

	Baseline	Peak	Plateau
EDHF	176.99	477.32	44.57
NO	89.88	190.09	77.81

	Baseline	Peak	Plateau
EDHF	187.25	770.63	96.33
NO	110.71	217.71	131.20

Table C.4: Values for ROI 1/2/3

C.5 Patient 5

	Baseline	Peak	Plateau
EDHF	270.00	276.22	72.31
NO	132.25	101.12	110.65

	Baseline	Peak	Plateau
EDHF	118.06	102.90	356.91
NO	82.99	75.38	210.07

	Baseline	Peak	Plateau
EDHF	488.73	475.46	110.10
NO	193.52	169.63	120.84

Table C.5: Values for ROI 1/2/3

C.6 Patient 6

	Baseline	Peak	Plateau
EDHF	178.61	377.06	126.02
NO	58.72	169.91	40.16

	Baseline	Peak	Plateau
EDHF	163.32	133.18	173.32
NO	62.69	78.64	50.72
	Baseline	**Peak**	**Plateau**
EDHF	200.40	448.83	328.96
NO	81.49	180.67	85.15

Table C.6: Values for ROI 1/2/3

C.7 Patient 7

	Baseline	Peak	Plateau
EDHF	735.85	407.48	278.82
NO	220.98	147.58	197.98
	Baseline	**Peak**	**Plateau**
EDHF	793.66	549.45	419.20
NO	285.14	195.91	279.29
	Baseline	**Peak**	**Plateau**
EDHF	793.76	783.67	229.21
NO	268.11	219.07	208.08

Table C.7: Values for ROI 1/2/3

C.8 Patient 8

	Baseline	Peak	Plateau
EDHF	78.41	829.78	49.88
NO	64.23	308.01	39.67
	Baseline	**Peak**	**Plateau**
EDHF	138.06	1424.42	85.89
NO	47.27	533.78	60.90
	Baseline	**Peak**	**Plateau**
EDHF	202.73	1558.23	109.93
NO	67.31	592.58	61.75

Table C.8: Values for ROI 1/2/3

C.9 Patient 9

	Baseline	Peak	Plateau
EDHF	117.85	198.56	122.40
NO	56.15	133.24	66.69
	Baseline	**Peak**	**Plateau**
EDHF	122.22	351.85	33.49
NO	99.27	229.29	88.42
	Baseline	**Peak**	**Plateau**
EDHF	111.90	528.71	152.27
NO	114.26	304.67	79.74

Table C.9: Values for ROI 1/2/3

C.10 Patient 10

	Baseline	Peak	Plateau
EDHF	209.67	566.73	22.18
NO	103.95	244.41	108.54
	Baseline	**Peak**	**Plateau**
EDHF	223.47	375.84	183.89
NO	191.27	197.05	77.15
	Baseline	**Peak**	**Plateau**
EDHF	253.00	748.76	115.32
NO	149.06	353.23	92.80

Table C.10: Values for ROI 1/2/3

Appendix D

Miscellaneous information

D.1 Values for smoothing

	Patient 1	Patient 2	Patient 3
x	50	50	25

	1	2	3	4	5	6	7	8	9	10
x	20	30	40	15	30	20	30	30	30	30

Appendix E

Code

E.1 Patient 1, set 1

Codes for patient 2 and 3 are similar.

```
#%%
import numpy as np
from matplotlib import pyplot as plt
from scipy.fft import fft, fftfreq
import pandas as pd
from math import ceil
import scipy.signal
import openpyxl
#import random

#%%
def movavg(sig, val):
    init = np.zeros(len(sig))
    for i in range(val):
        init[i] = np.array([1/(val+1)])
    outp = np.convolve(sig, init)[:len(sig)]
    return outp

#%%
df = pd.read_excel(r"C:\Users\Nisha\Documents\BEP\LSCImeting.xlsx",
                   sheet_name = 1, header = 29, usecols="A:F", nrows = 2544)

#%% To find the row numbers where of the baseline, peak and plateau
wb = openpyxl.load_workbook(r'C:\Users\Nisha\Documents\BEP\LSCImeting.xlsx',
                            data_only=True)

# get the worksheet
ws = wb.worksheets[1]

# check first column in all columns for a different colour
rows = []
colours = []
for row in range(31,len(df)+31):
    cell = ws.cell(column=1, row=row)
```

```
    fgColor = cell.fill.fgColor.index
    if fgColor != '00000000':
        rows.append(row)
        colours.append(fgColor)

print(rows)
print(colours)

#%% In excel file the data starts from row 31 and here from 0.
#Hence the acquired row numbers are "corrected" by 31
rown = [k-31 for k in rows]
print(rown)

#%% Find entries at which we expect the intervals of mechanisms to start
vals = [0.005, 0.0095, 0.021]
p3 = []
for v in vals:
    p3.append(ceil(v*300))
print(p3)

p6 = []
for v in vals:
    p6.append(ceil(v*600))
print(p6)

p21 = []
for v in vals:
    p21.append(ceil(v*2100))
print(p21)

p14 = []
for v in vals:
    p14.append(ceil(v*1400))
print(p14)

#%% Different regions of interest
ROI1 = np.array(df['1. ROI'])
ROI2 = np.array(df['2. ROI'])
ROI3 = np.array(df['3. ROI'])
#%% Properties
N = len(df)
H = N//2
n = np.arange(N)

#%% To determine a smoothing factor
# f1 = [10,20,30,40,50,60,70]
# for elts in f1:
#     ROI3s = movavg(df['3. ROI'], elts)
#     plt.plot(ROI3s)
#     plt.title("ROI 3 patient 1 smooth with x = "+ str(elts))
#     plt.xlabel("Time in seconds")
```

```
#       plt.ylabel("Change in dermal blood flow")
#       plt.show()

#%% Smoothing with a factor of 50
f = 50
ROI3s = movavg(ROI3, f)
ROI1s = movavg(ROI1, f)
ROI2s = movavg(ROI2, f)

#%% ROI3 after smoothing
# plt.plot(ROI3s)
# plt.title("ROI 3 patient 1 smooth with x = "+str(f))
# #plt.title("ROI3 patient 1 smoothed with x =" + str(f))
# plt.xlabel("Time in seconds")
# plt.ylabel("Change in dermal blood flow")
# plt.show()

#%% Amplitude spectrum ROI3 entirely after smoothing and DC offset removal
# plt.stem(fftfreq(N)[:1250],np.abs(fft(ROI3s - np.mean(ROI3s)))[:1250])
# plt.title("Amplitude ROI 3 patient 1")
# plt.xlabel("Digital frequency in hertz")
# plt.ylabel("Amplitude")
# plt.show()

#%% limits for the y-scale per phase
y1 = 2000 #400
y2 = 500 #4000
y3 = 500 #350
y4 = 2000#8000
y5 = 2000#4000

#%% Images initial smoothing 50 ROI 3
# plt.stem(fftfreq(300)[:30], np.abs(fft(ROI3s[rown[3]:rown[4]]
#       -np.mean(ROI3s[rown[3]:rown[4]])))[:30])
# plt.title("Amplitude spectrum ROI 3 Plateau")
# plt.xlabel("Frequency in Hz")
# plt.ylabel("Amplitude")

#%% Baseline ROI 1, 300 points
plt.stem(fftfreq(300)[:(p3[0]+1)], np.abs(fft(ROI1s[rown[0]:rown[1]]
    -np.mean(ROI1s[rown[0]:rown[1]])))[:(p3[0]+1)], linefmt = "b")
plt.stem(fftfreq(300)[p3[0]:(p3[1]+1)], np.abs(fft(ROI1s[rown[0]:rown[1]]
    -np.mean(ROI1s[rown[0]:rown[1]])))[p3[0]:(p3[1]+1)], linefmt = "r")
plt.stem(fftfreq(300)[p3[1]:(p3[2]+1)], np.abs(fft(ROI1s[rown[0]:rown[1]]
    -np.mean(ROI1s[rown[0]:rown[1]])))[p3[1]:(p3[2]+1)], linefmt = "c")
plt.stem(fftfreq(300)[p3[2]:(p3[2]+8)], np.abs(fft(ROI1s[rown[0]:rown[1]]
    -np.mean(ROI1s[rown[0]:rown[1]])))[p3[2]:(p3[2]+8)], linefmt = "b")
plt.title("Amplitude spectrum ROI 1 baseline patient 1")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y1)
```

```
plt.show()

#%% Baseline ROI 2, 300 points
plt.stem(fftfreq(300)[:(p3[0]+1)], np.abs(fft(ROI2s[rown[0]:rown[1]]
    -np.mean(ROI2s[rown[0]:rown[1]])))[:(p3[0]+1)], linefmt = "b")
plt.stem(fftfreq(300)[p3[0]:(p3[1]+1)], np.abs(fft(ROI2s[rown[0]:rown[1]]
    -np.mean(ROI2s[rown[0]:rown[1]])))[p3[0]:(p3[1]+1)], linefmt = "r")
plt.stem(fftfreq(300)[p3[1]:(p3[2]+1)], np.abs(fft(ROI2s[rown[0]:rown[1]]
    -np.mean(ROI2s[rown[0]:rown[1]])))[p3[1]:(p3[2]+1)], linefmt = "c")
plt.stem(fftfreq(300)[p3[2]:(p3[2]+8)], np.abs(fft(ROI2s[rown[0]:rown[1]]
    -np.mean(ROI2s[rown[0]:rown[1]])))[p3[2]:(p3[2]+8)], linefmt = "b")
plt.title("Amplitude spectrum ROI 2 baseline patient 1")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y1)
plt.show()

#%% Baseline ROI 3, 300 points
plt.stem(fftfreq(300)[:(p3[0]+1)], np.abs(fft(ROI3s[rown[0]:rown[1]]
    -np.mean(ROI3s[rown[0]:rown[1]])))[:(p3[0]+1)], linefmt = "b")
plt.stem(fftfreq(300)[p3[0]:(p3[1]+1)], np.abs(fft(ROI3s[rown[0]:rown[1]]
    -np.mean(ROI3s[rown[0]:rown[1]])))[p3[0]:(p3[1]+1)], linefmt = "r")
plt.stem(fftfreq(300)[p3[1]:(p3[2]+1)], np.abs(fft(ROI3s[rown[0]:rown[1]]
    -np.mean(ROI3s[rown[0]:rown[1]])))[p3[1]:(p3[2]+1)], linefmt = "c")
plt.stem(fftfreq(300)[p3[2]:(p3[2]+8)], np.abs(fft(ROI3s[rown[0]:rown[1]]
    -np.mean(ROI3s[rown[0]:rown[1]])))[p3[2]:(p3[2]+8)], linefmt = "b")
plt.title("Amplitude spectrum ROI 3 baseline patient 1")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y1)
plt.show()

#%% Peak phase ROI1 from index 411 till 1011
plt.stem(fftfreq(600)[:(p6[0]+1)], np.abs(fft(ROI1s[rown[1]:rown[2]]
    -np.mean(ROI1s[rown[1]:rown[2]])))[:(p6[0]+1)], linefmt = "b")
plt.stem(fftfreq(600)[p6[0]:(p6[1]+1)], np.abs(fft(ROI1s[rown[1]:rown[2]]
    -np.mean(ROI1s[rown[1]:rown[2]])))[p6[0]:(p6[1]+1)], linefmt = "r")
plt.stem(fftfreq(600)[p6[1]:(p6[2]+1)], np.abs(fft(ROI1s[rown[1]:rown[2]]
    -np.mean(ROI1s[rown[1]:rown[2]])))[p6[1]:(p6[2]+1)], linefmt = "c")
plt.stem(fftfreq(600)[p6[2]:(p6[2]+17)], np.abs(fft(ROI1s[rown[1]:rown[2]]
    -np.mean(ROI1s[rown[1]:rown[2]])))[p6[2]:(p6[2]+17)], linefmt = "b")
plt.title("Amplitude spectrum ROI 1 peak patient 1")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y2)
# plt.xticks(np.arange(0.004,0.01, 0.001))
plt.show()

#%% Peak phase ROI2 from index 411 till 1011
plt.stem(fftfreq(600)[:(p6[0]+1)], np.abs(fft(ROI2s[rown[1]:rown[2]]
    -np.mean(ROI2s[rown[1]:rown[2]])))[:(p6[0]+1)], linefmt = "b")
```

```
plt.stem(fftfreq(600)[p6[0]:(p6[1]+1)], np.abs(fft(ROI2s[rown[1]:rown[2]]
    -np.mean(ROI2s[rown[1]:rown[2]])))[p6[0]:(p6[1]+1)], linefmt = "r")
plt.stem(fftfreq(600)[p6[1]:(p6[2]+1)], np.abs(fft(ROI2s[rown[1]:rown[2]]
    -np.mean(ROI2s[rown[1]:rown[2]])))[p6[1]:(p6[2]+1)], linefmt = "c")
plt.stem(fftfreq(600)[p6[2]:(p6[2]+17)], np.abs(fft(ROI2s[rown[1]:rown[2]]
    -np.mean(ROI2s[rown[1]:rown[2]])))[p6[2]:(p6[2]+17)], linefmt = "b")
plt.title("Amplitude spectrum ROI 2 peak patient 1")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y2)
plt.show()

#%% Peak phase ROI3 from index 411 till 1011
plt.stem(fftfreq(600)[:(p6[0]+1)], np.abs(fft(ROI3s[rown[1]:rown[2]]
    -np.mean(ROI3s[rown[1]:rown[2]])))[:(p6[0]+1)], linefmt = "b")
plt.stem(fftfreq(600)[p6[0]:(p6[1]+1)], np.abs(fft(ROI3s[rown[1]:rown[2]]
    -np.mean(ROI3s[rown[1]:rown[2]])))[p6[0]:(p6[1]+1)], linefmt = "r")
plt.stem(fftfreq(600)[p6[1]:(p6[2]+1)], np.abs(fft(ROI3s[rown[1]:rown[2]]
    -np.mean(ROI3s[rown[1]:rown[2]])))[p6[1]:(p6[2]+1)], linefmt = "c")
plt.stem(fftfreq(600)[p6[2]:(p6[2]+17)], np.abs(fft(ROI3s[rown[1]:rown[2]]
    -np.mean(ROI3s[rown[1]:rown[2]])))[p6[2]:(p6[2]+17)], linefmt = "b")
plt.title("Amplitude spectrum ROI 3 peak patient 1")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y2)
plt.show()

#print(fftfreq(600)[:10])

#%% Plateau ROI1 from index 2211 till 2511
plt.stem(fftfreq(300)[:(p3[0]+1)], np.abs(fft(ROI1s[rown[3]:rown[4]]
    -np.mean(ROI1s[rown[3]:rown[4]])))[:(p3[0]+1)], linefmt = "b")
plt.stem(fftfreq(300)[p3[0]:(p3[1]+1)], np.abs(fft(ROI1s[rown[3]:rown[4]]
    -np.mean(ROI1s[rown[3]:rown[4]])))[p3[0]:(p3[1]+1)], linefmt = "r")
plt.stem(fftfreq(300)[p3[1]:(p3[2]+1)], np.abs(fft(ROI1s[rown[3]:rown[4]]
    -np.mean(ROI1s[rown[3]:rown[4]])))[p3[1]:(p3[2]+1)], linefmt = "c")
plt.stem(fftfreq(300)[p3[2]:(p3[2]+8)], np.abs(fft(ROI1s[rown[3]:rown[4]]
    -np.mean(ROI1s[rown[3]:rown[4]])))[p3[2]:(p3[2]+8)], linefmt = "b")
plt.title("Amplitude spectrum ROI 1 plateau patient 1")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y3)
plt.show()

#%% Plateau ROI2 from index 2211 till 2511
plt.stem(fftfreq(300)[:(p3[0]+1)], np.abs(fft(ROI2s[rown[3]:rown[4]]
    -np.mean(ROI2s[rown[3]:rown[4]])))[:(p3[0]+1)], linefmt = "b")
plt.stem(fftfreq(300)[p3[0]:(p3[1]+1)], np.abs(fft(ROI2s[rown[3]:rown[4]]
    -np.mean(ROI2s[rown[3]:rown[4]])))[p3[0]:(p3[1]+1)], linefmt = "r")
plt.stem(fftfreq(300)[p3[1]:(p3[2]+1)], np.abs(fft(ROI2s[rown[3]:rown[4]]
    -np.mean(ROI2s[rown[3]:rown[4]])))[p3[1]:(p3[2]+1)], linefmt = "c")
```

```
plt.stem(fftfreq(300)[p3[2]:(p3[2]+8)], np.abs(fft(ROI2s[rown[3]:rown[4]]
    -np.mean(ROI2s[rown[3]:rown[4]])))[p3[2]:(p3[2]+8)], linefmt = "b")
plt.title("Amplitude spectrum ROI 2 plateau patient 1")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y3)
plt.show()

#%% Plateau ROI3 from index 2211 till 2511
plt.stem(fftfreq(300)[:(p3[0]+1)], np.abs(fft(ROI3s[rown[3]:rown[4]]
    -np.mean(ROI3s[rown[3]:rown[4]])))[:(p3[0]+1)], linefmt = "b")
plt.stem(fftfreq(300)[p3[0]:(p3[1]+1)], np.abs(fft(ROI3s[rown[3]:rown[4]]
    -np.mean(ROI3s[rown[3]:rown[4]])))[p3[0]:(p3[1]+1)], linefmt = "r")
plt.stem(fftfreq(300)[p3[1]:(p3[2]+1)], np.abs(fft(ROI3s[rown[3]:rown[4]]
    -np.mean(ROI3s[rown[3]:rown[4]])))[p3[1]:(p3[2]+1)], linefmt = "c")
plt.stem(fftfreq(300)[p3[2]:(p3[2]+8)], np.abs(fft(ROI3s[rown[3]:rown[4]]
    -np.mean(ROI3s[rown[3]:rown[4]])))[p3[2]:(p3[2]+8)], linefmt = "b")
plt.title("Amplitude spectrum ROI 3 plateau patient 1")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y3)
plt.show()

#print(fftfreq(300)[:10])

#%% From start peak until end of plateau ROI1
plt.stem(fftfreq(2100)[:(p21[0]+1)], np.abs(fft(ROI1s[rown[1]:rown[4]]
    -np.mean(ROI1s[rown[1]:rown[4]])))[:(p21[0]+1)], linefmt = "b")
plt.stem(fftfreq(2100)[p21[0]:(p21[1]+2)], np.abs(fft(ROI1s[rown[1]:rown[4]]
    -np.mean(ROI1s[rown[1]:rown[4]])))[p21[0]:(p21[1]+2)], linefmt = "r")
plt.stem(fftfreq(2100)[p21[1]+1:(p21[2]+1)], np.abs(fft(ROI1s[rown[1]:rown[4]]
    -np.mean(ROI1s[rown[1]:rown[4]])))[p21[1]+1:(p21[2]+1)], linefmt = "c")
plt.stem(fftfreq(2100)[p21[2]:(p21[2]+5)], np.abs(fft(ROI1s[rown[1]:rown[4]]
    -np.mean(ROI1s[rown[1]:rown[4]])))[p21[2]:(p21[2]+5)], linefmt = "b")
plt.title("Amplitude spectrum ROI 1 start peak till end plateau (P1)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y4)
#plt.xticks(np.arange(0.021,0.023,0.002))
plt.show()

#%% From start peak until end of plateau ROI2
plt.stem(fftfreq(2100)[:(p21[0]+1)], np.abs(fft(ROI2s[rown[1]:rown[4]]
    -np.mean(ROI2s[rown[1]:rown[4]])))[:(p21[0]+1)], linefmt = "b")
plt.stem(fftfreq(2100)[p21[0]:(p21[1]+2)], np.abs(fft(ROI2s[rown[1]:rown[4]]
    -np.mean(ROI2s[rown[1]:rown[4]])))[p21[0]:(p21[1]+2)], linefmt = "r")
plt.stem(fftfreq(2100)[p21[1]+1:(p21[2]+1)], np.abs(fft(ROI2s[rown[1]:rown[4]]
    -np.mean(ROI2s[rown[1]:rown[4]])))[p21[1]+1:(p21[2]+1)], linefmt = "c")
plt.stem(fftfreq(2100)[p21[2]:(p21[2]+5)], np.abs(fft(ROI2s[rown[1]:rown[4]]
    -np.mean(ROI2s[rown[1]:rown[4]])))[p21[2]:(p21[2]+5)], linefmt = "b")
plt.title("Amplitude spectrum ROI 2 start peak till end plateau (P1)")
```

```
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y4)
plt.show()

#%% From start peak until end of plateau ROI3
plt.stem(fftfreq(2100)[:(p21[0]+1)], np.abs(fft(ROI3s[rown[1]:rown[4]]
    -np.mean(ROI3s[rown[1]:rown[4]])))[:(p21[0]+1)], linefmt = "b")
plt.stem(fftfreq(2100)[p21[0]:(p21[1]+2)], np.abs(fft(ROI3s[rown[1]:rown[4]]
    -np.mean(ROI3s[rown[1]:rown[4]])))[p21[0]:(p21[1]+2)], linefmt = "r")
plt.stem(fftfreq(2100)[p21[1]+1:(p21[2]+1)], np.abs(fft(ROI3s[rown[1]:rown[4]]
    -np.mean(ROI3s[rown[1]:rown[4]])))[p21[1]+1:(p21[2]+1)], linefmt = "c")
plt.stem(fftfreq(2100)[p21[2]:(p21[2]+5)], np.abs(fft(ROI3s[rown[1]:rown[4]]
    -np.mean(ROI3s[rown[1]:rown[4]])))[p21[2]:(p21[2]+5)], linefmt = "b")
plt.title("Amplitude spectrum ROI 3 start peak till end plateau (P1)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y4)
plt.show()

#%% From end of peak until end of plateau ROI1
plt.stem(fftfreq(1400)[:(p14[0]+1)], np.abs(fft(ROI1s[rown[2]:rown[4]]
    -np.mean(ROI1s[rown[2]:rown[4]])))[:(p14[0]+1)], linefmt = "b")
plt.stem(fftfreq(1400)[p14[0]:(p14[1]+1)], np.abs(fft(ROI1s[rown[2]:rown[4]]
    -np.mean(ROI1s[rown[2]:rown[4]])))[p14[0]:(p14[1]+1)], linefmt = "r")
plt.stem(fftfreq(1400)[p14[1]:(p14[2]+1)], np.abs(fft(ROI1s[rown[2]:rown[4]]
    -np.mean(ROI1s[rown[2]:rown[4]])))[p14[1]:(p14[2]+1)], linefmt = "c")
plt.stem(fftfreq(1400)[p14[2]:(p14[2]+10)], np.abs(fft(ROI1s[rown[2]:rown[4]]
    -np.mean(ROI1s[rown[2]:rown[4]])))[p14[2]:(p14[2]+10)], linefmt = "b")
plt.title("Amplitude spectrum ROI 1 end peak till end plateau (P1)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
#plt.xticks(np.arange(0.015,0.022,0.002))
plt.ylim(0,y5)
plt.show()

#%% From end of peak until end of plateau ROI2
plt.stem(fftfreq(1400)[:(p14[0]+1)], np.abs(fft(ROI2s[rown[2]:rown[4]]
    -np.mean(ROI2s[rown[2]:rown[4]])))[:(p14[0]+1)], linefmt = "b")
plt.stem(fftfreq(1400)[p14[0]:(p14[1]+1)], np.abs(fft(ROI2s[rown[2]:rown[4]]
    -np.mean(ROI2s[rown[2]:rown[4]])))[p14[0]:(p14[1]+1)], linefmt = "r")
plt.stem(fftfreq(1400)[p14[1]:(p14[2]+1)], np.abs(fft(ROI2s[rown[2]:rown[4]]
    -np.mean(ROI2s[rown[2]:rown[4]])))[p14[1]:(p14[2]+1)], linefmt = "c")
plt.stem(fftfreq(1400)[p14[2]:(p14[2]+10)], np.abs(fft(ROI2s[rown[2]:rown[4]]
    -np.mean(ROI2s[rown[2]:rown[4]])))[p14[2]:(p14[2]+10)], linefmt = "b")
plt.title("Amplitude spectrum ROI 2 end peak till end plateau (P1)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y5)
plt.show()
```

```
#%% From end of peak until end of plateau ROI3
plt.stem(fftfreq(1400)[:(p14[0]+1)], np.abs(fft(ROI3s[rown[2]:rown[4]]
    -np.mean(ROI3s[rown[2]:rown[4]])))[:(p14[0]+1)], linefmt = "b")
plt.stem(fftfreq(1400)[p14[0]:(p14[1]+1)], np.abs(fft(ROI3s[rown[2]:rown[4]]
    -np.mean(ROI3s[rown[2]:rown[4]])))[p14[0]:(p14[1]+1)], linefmt = "r")
plt.stem(fftfreq(1400)[p14[1]:(p14[2]+1)], np.abs(fft(ROI3s[rown[2]:rown[4]]
    -np.mean(ROI3s[rown[2]:rown[4]])))[p14[1]:(p14[2]+1)], linefmt = "c")
plt.stem(fftfreq(1400)[p14[2]:(p14[2]+10)], np.abs(fft(ROI3s[rown[2]:rown[4]]
    -np.mean(ROI3s[rown[2]:rown[4]])))[p14[2]:(p14[2]+10)], linefmt = "b")
plt.title("Amplitude spectrum ROI 3 end peak till end plateau (P1)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y5)
plt.show()

#%% Average values
print("Patient 1, 1st data set")

print("Average values for baseline: ROI 1, 2 and 3")

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI1s[rown[0]:rown[1]]
    -np.mean(ROI1s[rown[0]:rown[1]])))[2])))
print("Avg val NO = "+str(np.sum(np.abs(fft(ROI1s[rown[0]:rown[1]]
    -np.mean(ROI1s[rown[0]:rown[1]])))[3:7])/4))

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI2s[rown[0]:rown[1]]
    -np.mean(ROI2s[rown[0]:rown[1]])))[2])))
print("Avg val NO = "+str(np.sum(np.abs(fft(ROI2s[rown[0]:rown[1]]
    -np.mean(ROI2s[rown[0]:rown[1]])))[3:7])/4))

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI3s[rown[0]:rown[1]]
    -np.mean(ROI3s[rown[0]:rown[1]])))[2])))
print("Avg val NO = "+str(np.sum(np.abs(fft(ROI3s[rown[0]:rown[1]]
    -np.mean(ROI3s[rown[0]:rown[1]])))[3:7])/4))


print("Average values for peak: ROI 1, 2 and 3")

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI1s[rown[1]:rown[2]]
    -np.mean(ROI1s[rown[1]:rown[2]])))[3:6])/3))
print("Avg val NO = "+str(np.sum(np.abs(fft(ROI1s[rown[1]:rown[2]]
    -np.mean(ROI1s[rown[1]:rown[2]])))[6:13])/7))

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI2s[rown[1]:rown[2]]
    -np.mean(ROI2s[rown[1]:rown[2]])))[3:6])/3))
print("Avg val NO = "+str(np.sum(np.abs(fft(ROI2s[rown[1]:rown[2]]
    -np.mean(ROI2s[rown[1]:rown[2]])))[6:13])/7))

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI3s[rown[1]:rown[2]]
    -np.mean(ROI3s[rown[1]:rown[2]])))[3:6])/3))
print("Avg val NO = "+str(np.sum(np.abs(fft(ROI3s[rown[1]:rown[2]]
```

```
    -np.mean(ROI3s[rown[1]:rown[2]]))))[6:13])/7))



print("Average values for plateau: ROI 1, 2 and 3")

print("Avg val EDHF = "+str(np.abs(fft(ROI1s[rown[3]:rown[4]]
    -np.mean(ROI1s[rown[3]:rown[4]])))[2]))
print("Avg val NO = "+str((np.sum(np.abs(fft(ROI1s[rown[3]:rown[4]]
    -np.mean(ROI1s[rown[3]:rown[4]])))[3:7])/4)))

print("Avg val EDHF = "+str(np.abs(fft(ROI2s[rown[3]:rown[4]]
    -np.mean(ROI2s[rown[3]:rown[4]])))[2]))
print("Avg val NO = "+str((np.sum(np.abs(fft(ROI2s[rown[3]:rown[4]]
    -np.mean(ROI2s[rown[3]:rown[4]])))[3:7])/4)))

print("Avg val EDHF = "+str(np.abs(fft(ROI3s[rown[3]:rown[4]]
    -np.mean(ROI3s[rown[3]:rown[4]])))[2]))
print("Avg val NO = "+str((np.sum(np.abs(fft(ROI3s[rown[3]:rown[4]]
    -np.mean(ROI3s[rown[3]:rown[4]])))[3:7])/4)))

#%%
print("Average values for SPEP: ROI 1, 2 and 3")

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI1s[rown[1]:rown[4]]
    -np.mean(ROI1s[rown[1]:rown[4]])))[11:21])/10))
print("Avg val NO = "+str((np.sum(np.abs(fft(ROI1s[rown[1]:rown[4]]
    -np.mean(ROI1s[rown[1]:rown[4]])))[21:46])/25)))

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI2s[rown[1]:rown[4]]
    -np.mean(ROI2s[rown[1]:rown[4]])))[11:21])/10))
print("Avg val NO = "+str((np.sum(np.abs(fft(ROI2s[rown[1]:rown[4]]
    -np.mean(ROI2s[rown[1]:rown[4]])))[21:46])/25)))

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI3s[rown[1]:rown[4]]
    -np.mean(ROI3s[rown[1]:rown[4]])))[11:21])/10))
print("Avg val NO = "+str((np.sum(np.abs(fft(ROI3s[rown[1]:rown[4]]
    -np.mean(ROI3s[rown[1]:rown[4]])))[21:46])/25)))

#%%
print("Average values for EPEP: ROI 1, 2 and 3")

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI1s[rown[2]:rown[4]]
    -np.mean(ROI1s[rown[2]:rown[4]])))[7:14])/7))
print("Avg val NO = "+str((np.sum(np.abs(fft(ROI1s[rown[2]:rown[4]]
    -np.mean(ROI1s[rown[2]:rown[4]])))[14:30])/16)))

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI2s[rown[2]:rown[4]]
    -np.mean(ROI2s[rown[2]:rown[4]])))[7:14])/7))
print("Avg val NO = "+str((np.sum(np.abs(fft(ROI2s[rown[2]:rown[4]]
    -np.mean(ROI2s[rown[2]:rown[4]])))[14:30])/16)))
```

```
print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI3s[rown[2]:rown[4]]
    -np.mean(ROI3s[rown[2]:rown[4]])))[7:14])/7))
print("Avg val NO = "+str((np.sum(np.abs(fft(ROI3s[rown[2]:rown[4]]
    -np.mean(ROI3s[rown[2]:rown[4]]))))[14:30])/16)))
```

E.2 3rd set of patients

```
#%%
import numpy as np
from matplotlib import pyplot as plt
from scipy.fft import fft, fftfreq
import pandas as pd
from math import ceil
import scipy.signal
import openpyxl
#import random

#%%
def movavg(sig, val):
    init = np.zeros(len(sig))
    for i in range(val):
        init[i] = np.array([1/(val+1)])
    outp = np.convolve(sig, init)[:len(sig)]
    return outp

#%%
i = 9
df = pd.read_excel(r"C:\Users\Nisha\Documents\BEP\Deel2LSCI.xlsx",
                   sheet_name = i, header = 29, usecols="A:F")

df = df.dropna()

time = ceil(df["Time min"].iloc[-1]*60)
N = len(df) #N + 30 gives the final row of the data
fs = N/time
H = N//2
n = np.arange(N)
#ROIN3 = np.array(df1['3. ROI'])

#%% Original signal full
# plt.plot(n, df["3. ROI"])
# plt.title("ROI 3 patient "+str(i+1)+" (3rd set)")
# plt.xlabel("Time in seconds")
# plt.ylabel("Change in dermal blood flow")
# plt.show()

#%% Smoothing
# f1 = [10, 20, 30, 40, 50, 60]
# for elts in f1:
```

```
#       smooth3 = movavg(df["3. ROI"], elts)
#       plt.plot(n, smooth3)
#       plt.title("ROI 3 patient "+str(i+1)+" (3rd set) smooth with x
#       = "+str(elts))
#       plt.xlabel("Time in seconds")
#       plt.ylabel("Change in dermal blood flow")
#       plt.show()

#%%
f2 = [20,30,40,15,30,20,30,30,30,30]
f = f2[i]
ROI1s = movavg(df["1. ROI"], f)
ROI2s = movavg(df["2. ROI"], f)
ROI3s = movavg(df["3. ROI"], f)

#%%
wb = openpyxl.load_workbook(r'C:\Users\Nisha\Documents\BEP\Deel2LSCI.xlsx',
                            data_only=True)

ws = wb.worksheets[i]

rows = []
colours = []
for row in range(31,len(df)+31):
    cell = ws.cell(column=1, row=row)
    fgColor = cell.fill.fgColor.index
    if fgColor != '00000000':
        rows.append(row)
        colours.append(fgColor)

print(rows)
print(colours)


#%%
rown = [k-31 for k in rows]

#%%
vals = [0.005, 0.0095, 0.021]
p3 = []
for v in vals:
    p3.append(ceil(v*300))
print(p3)

p6 = []
for v in vals:
    p6.append(ceil(v*600))
print(p6)

p21 = []
for v in vals:
```

```
    p21.append(ceil(v*2100))
print(p21)

p14 = []
for v in vals:
    p14.append(ceil(v*1400))
print(p14)

#%%
y11 = [200,300,400,200,1100,300,1000,300,300,400] #Baselines
y22 = [4000,2500,600,2500,2500,4500,3500,3000,4500,5000] #Peak
y33 = [400,350,100,500,700,400,500,400,350,400] #Plateau
y44 = [10000]*10
y55 = [5000]*10
y1 = y11[i]
y2 = y22[i]
y3 = y33[i]
y4 = y44[i]
y5 = y55[i]

#%% Baselines
# Baseline, 300 points
plt.stem(fftfreq(300)[:(p3[0]+1)], np.abs(fft(ROI1s[rown[0]:rown[1]]
    -np.mean(ROI1s[rown[0]:rown[1]])))[:(p3[0]+1)], linefmt = "b")
plt.stem(fftfreq(300)[p3[0]:(p3[1]+1)], np.abs(fft(ROI1s[rown[0]:rown[1]]
    -np.mean(ROI1s[rown[0]:rown[1]])))[p3[0]:(p3[1]+1)], linefmt = "r")
plt.stem(fftfreq(300)[p3[1]:(p3[2]+1)], np.abs(fft(ROI1s[rown[0]:rown[1]]
    -np.mean(ROI1s[rown[0]:rown[1]])))[p3[1]:(p3[2]+1)], linefmt = "c")
plt.stem(fftfreq(300)[p3[2]:(p3[2]+8)], np.abs(fft(ROI1s[rown[0]:rown[1]]
    -np.mean(ROI1s[rown[0]:rown[1]])))[p3[2]:(p3[2]+8)], linefmt = "b")
plt.title("Amplitude spectrum ROI 1 baseline patient "+str(i+1)+" (3rd set)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y1)
plt.show()

# Baseline, 300 points
plt.stem(fftfreq(300)[:(p3[0]+1)], np.abs(fft(ROI2s[rown[0]:rown[1]]
    -np.mean(ROI2s[rown[0]:rown[1]])))[:(p3[0]+1)], linefmt = "b")
plt.stem(fftfreq(300)[p3[0]:(p3[1]+1)], np.abs(fft(ROI2s[rown[0]:rown[1]]
    -np.mean(ROI2s[rown[0]:rown[1]])))[p3[0]:(p3[1]+1)], linefmt = "r")
plt.stem(fftfreq(300)[p3[1]:(p3[2]+1)], np.abs(fft(ROI2s[rown[0]:rown[1]]
    -np.mean(ROI2s[rown[0]:rown[1]])))[p3[1]:(p3[2]+1)], linefmt = "c")
plt.stem(fftfreq(300)[p3[2]:(p3[2]+8)], np.abs(fft(ROI2s[rown[0]:rown[1]]
    -np.mean(ROI2s[rown[0]:rown[1]])))[p3[2]:(p3[2]+8)], linefmt = "b")
plt.title("Amplitude spectrum ROI 2 baseline patient "+str(i+1)+" (3rd set)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y1)
plt.show()
```

```
# Baseline ROI 3
plt.stem(fftfreq(300)[:(p3[0]+1)], np.abs(fft(ROI3s[rown[0]:rown[1]]
    -np.mean(ROI3s[rown[0]:rown[1]])))[:(p3[0]+1)], linefmt = "b")
plt.stem(fftfreq(300)[p3[0]:(p3[1]+1)], np.abs(fft(ROI3s[rown[0]:rown[1]]
    -np.mean(ROI3s[rown[0]:rown[1]])))[p3[0]:(p3[1]+1)], linefmt = "r")
plt.stem(fftfreq(300)[p3[1]:(p3[2]+1)], np.abs(fft(ROI3s[rown[0]:rown[1]]
    -np.mean(ROI3s[rown[0]:rown[1]])))[p3[1]:(p3[2]+1)], linefmt = "c")
plt.stem(fftfreq(300)[p3[2]:(p3[2]+8)], np.abs(fft(ROI3s[rown[0]:rown[1]]
    -np.mean(ROI3s[rown[0]:rown[1]])))[p3[2]:(p3[2]+8)], linefmt = "b")
plt.title("Amplitude spectrum ROI 3 baseline patient "+str(i+1)+" (3rd set)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y1)
plt.show()

#print(fftfreq(300)[:10])

#%% Peak phases
# Peak phase ROI1
plt.stem(fftfreq(600)[:(p6[0]+1)], np.abs(fft(ROI1s[rown[1]:rown[2]]
    -np.mean(ROI1s[rown[1]:rown[2]])))[:(p6[0]+1)], linefmt = "b")
plt.stem(fftfreq(600)[p6[0]:(p6[1]+1)], np.abs(fft(ROI1s[rown[1]:rown[2]]
    -np.mean(ROI1s[rown[1]:rown[2]])))[p6[0]:(p6[1]+1)], linefmt = "r")
plt.stem(fftfreq(600)[p6[1]:(p6[2]+1)], np.abs(fft(ROI1s[rown[1]:rown[2]]
    -np.mean(ROI1s[rown[1]:rown[2]])))[p6[1]:(p6[2]+1)], linefmt = "c")
plt.stem(fftfreq(600)[p6[2]:(p6[2]+17)], np.abs(fft(ROI1s[rown[1]:rown[2]]
    -np.mean(ROI1s[rown[1]:rown[2]])))[p6[2]:(p6[2]+17)], linefmt = "b")
plt.title("Amplitude spectrum ROI 1 peak patient "+str(i+1)+" (3rd set)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y2)
# plt.xticks(np.arange(0.004,0.01, 0.001))
plt.show()

# Peak phase ROI2
plt.stem(fftfreq(600)[:(p6[0]+1)], np.abs(fft(ROI2s[rown[1]:rown[2]]
    -np.mean(ROI2s[rown[1]:rown[2]])))[:(p6[0]+1)], linefmt = "b")
plt.stem(fftfreq(600)[p6[0]:(p6[1]+1)], np.abs(fft(ROI2s[rown[1]:rown[2]]
    -np.mean(ROI2s[rown[1]:rown[2]])))[p6[0]:(p6[1]+1)], linefmt = "r")
plt.stem(fftfreq(600)[p6[1]:(p6[2]+1)], np.abs(fft(ROI2s[rown[1]:rown[2]]
    -np.mean(ROI2s[rown[1]:rown[2]])))[p6[1]:(p6[2]+1)], linefmt = "c")
plt.stem(fftfreq(600)[p6[2]:(p6[2]+17)], np.abs(fft(ROI2s[rown[1]:rown[2]]
    -np.mean(ROI2s[rown[1]:rown[2]])))[p6[2]:(p6[2]+17)], linefmt = "b")
plt.title("Amplitude spectrum ROI 2 peak patient "+str(i+1)+" (3rd set)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y2)
plt.show()

# Peak phase ROI3
plt.stem(fftfreq(600)[:(p6[0]+1)], np.abs(fft(ROI3s[rown[1]:rown[2]]
```

```
    -np.mean(ROI3s[rown[1]:rown[2]])))[:(p6[0]+1)], linefmt = "b")
plt.stem(fftfreq(600)[p6[0]:(p6[1]+1)], np.abs(fft(ROI3s[rown[1]:rown[2]]
    -np.mean(ROI3s[rown[1]:rown[2]])))[p6[0]:(p6[1]+1)], linefmt = "r")
plt.stem(fftfreq(600)[p6[1]:(p6[2]+1)], np.abs(fft(ROI3s[rown[1]:rown[2]]
    -np.mean(ROI3s[rown[1]:rown[2]])))[p6[1]:(p6[2]+1)], linefmt = "c")
plt.stem(fftfreq(600)[p6[2]:(p6[2]+17)], np.abs(fft(ROI3s[rown[1]:rown[2]]
    -np.mean(ROI3s[rown[1]:rown[2]])))[p6[2]:(p6[2]+17)], linefmt = "b")
plt.title("Amplitude spectrum ROI 3 peak patient "+str(i+1)+" (3rd set)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y2)
plt.show()

#print(fftfreq(600)[:10])

#%% Plateaus
#Plateau ROI1
plt.stem(fftfreq(300)[:(p3[0]+1)], np.abs(fft(ROI1s[rown[3]:rown[4]]
    -np.mean(ROI1s[rown[3]:rown[4]])))[:(p3[0]+1)], linefmt = "b")
plt.stem(fftfreq(300)[p3[0]:(p3[1]+1)], np.abs(fft(ROI1s[rown[3]:rown[4]]
    -np.mean(ROI1s[rown[3]:rown[4]])))[p3[0]:(p3[1]+1)], linefmt = "r")
plt.stem(fftfreq(300)[p3[1]:(p3[2]+1)], np.abs(fft(ROI1s[rown[3]:rown[4]]
    -np.mean(ROI1s[rown[3]:rown[4]])))[p3[1]:(p3[2]+1)], linefmt = "c")
plt.stem(fftfreq(300)[p3[2]:(p3[2]+8)], np.abs(fft(ROI1s[rown[3]:rown[4]]
    -np.mean(ROI1s[rown[3]:rown[4]])))[p3[2]:(p3[2]+8)], linefmt = "b")
plt.title("Amplitude spectrum ROI 1 plateau patient "+str(i+1)+" (3rd set)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y3)
plt.show()

# Plateau ROI2
plt.stem(fftfreq(300)[:(p3[0]+1)], np.abs(fft(ROI2s[rown[3]:rown[4]]
    -np.mean(ROI2s[rown[3]:rown[4]])))[:(p3[0]+1)], linefmt = "b")
plt.stem(fftfreq(300)[p3[0]:(p3[1]+1)], np.abs(fft(ROI2s[rown[3]:rown[4]]
    -np.mean(ROI2s[rown[3]:rown[4]])))[p3[0]:(p3[1]+1)], linefmt = "r")
plt.stem(fftfreq(300)[p3[1]:(p3[2]+1)], np.abs(fft(ROI2s[rown[3]:rown[4]]
    -np.mean(ROI2s[rown[3]:rown[4]])))[p3[1]:(p3[2]+1)], linefmt = "c")
plt.stem(fftfreq(300)[p3[2]:(p3[2]+8)], np.abs(fft(ROI2s[rown[3]:rown[4]]
    -np.mean(ROI2s[rown[3]:rown[4]])))[p3[2]:(p3[2]+8)], linefmt = "b")
plt.title("Amplitude spectrum ROI 2 plateau patient "+str(i+1)+" (3rd set)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y3)
plt.show()

# Plateau ROI3
plt.stem(fftfreq(300)[:(p3[0]+1)], np.abs(fft(ROI3s[rown[3]:rown[4]]
    -np.mean(ROI3s[rown[3]:rown[4]])))[:(p3[0]+1)], linefmt = "b")
plt.stem(fftfreq(300)[p3[0]:(p3[1]+1)], np.abs(fft(ROI3s[rown[3]:rown[4]]
    -np.mean(ROI3s[rown[3]:rown[4]])))[p3[0]:(p3[1]+1)], linefmt = "r")
```

```
plt.stem(fftfreq(300)[p3[1]:(p3[2]+1)], np.abs(fft(ROI3s[rown[3]:rown[4]]
    -np.mean(ROI3s[rown[3]:rown[4]])))[p3[1]:(p3[2]+1)], linefmt = "c")
plt.stem(fftfreq(300)[p3[2]:(p3[2]+8)], np.abs(fft(ROI3s[rown[3]:rown[4]]
    -np.mean(ROI3s[rown[3]:rown[4]])))[p3[2]:(p3[2]+8)], linefmt = "b")
plt.title("Amplitude spectrum ROI 3 plateau patient "+str(i+1)+" (3rd set)")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.ylim(0,y3)
plt.show()

#print(fftfreq(300)[:10])

#%% Average values
print("Patient "+str(i+1))

print("Average values for baseline")

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI1s[rown[0]:rown[1]]
    -np.mean(ROI1s[rown[0]:rown[1]])))[2])))
print("Avg val NO = "+str(np.sum(np.abs(fft(ROI1s[rown[0]:rown[1]]
    -np.mean(ROI1s[rown[0]:rown[1]])))[3:7])/4))

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI2s[rown[0]:rown[1]]
    -np.mean(ROI2s[rown[0]:rown[1]])))[2])))
print("Avg val NO = "+str(np.sum(np.abs(fft(ROI2s[rown[0]:rown[1]]
    -np.mean(ROI2s[rown[0]:rown[1]])))[3:7])/4))

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI3s[rown[0]:rown[1]]
    -np.mean(ROI3s[rown[0]:rown[1]])))[2])))
print("Avg val NO = "+str(np.sum(np.abs(fft(ROI3s[rown[0]:rown[1]]
    -np.mean(ROI3s[rown[0]:rown[1]])))[3:7])/4))


print("Average values for peaks")

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI1s[rown[1]:rown[2]]
    -np.mean(ROI1s[rown[1]:rown[2]])))[3:6])/3))
print("Avg val NO = "+str(np.sum(np.abs(fft(ROI1s[rown[1]:rown[2]]
    -np.mean(ROI1s[rown[1]:rown[2]])))[6:13])/7))

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI2s[rown[1]:rown[2]]
    -np.mean(ROI2s[rown[1]:rown[2]])))[3:6])/3))
print("Avg val NO = "+str(np.sum(np.abs(fft(ROI2s[rown[1]:rown[2]]
    -np.mean(ROI2s[rown[1]:rown[2]])))[6:13])/7))

print("Avg val EDHF = "+str(np.sum(np.abs(fft(ROI3s[rown[1]:rown[2]]
    -np.mean(ROI3s[rown[1]:rown[2]])))[3:6])/3))
print("Avg val NO = "+str(np.sum(np.abs(fft(ROI3s[rown[1]:rown[2]]
    -np.mean(ROI3s[rown[1]:rown[2]])))[6:13])/7))
```

```
print("Average values for plateau")

print("Avg val EDHF = "+str(np.abs(fft(ROI1s[rown[3]:rown[4]]
    -np.mean(ROI1s[rown[3]:rown[4]])))[2]))
print("Avg val NO = "+str((np.sum(np.abs(fft(ROI1s[rown[3]:rown[4]]
    -np.mean(ROI1s[rown[3]:rown[4]])))[3:7])/4)))

print("Avg val EDHF = "+str(np.abs(fft(ROI2s[rown[3]:rown[4]]
    -np.mean(ROI2s[rown[3]:rown[4]])))[2]))
print("Avg val NO = "+str((np.sum(np.abs(fft(ROI2s[rown[3]:rown[4]]
    -np.mean(ROI2s[rown[3]:rown[4]])))[3:7])/4)))

print("Avg val EDHF = "+str(np.abs(fft(ROI3s[rown[3]:rown[4]]
    -np.mean(ROI3s[rown[3]:rown[4]])))[2]))
print("Avg val NO = "+str((np.sum(np.abs(fft(ROI3s[rown[3]:rown[4]]
    -np.mean(ROI3s[rown[3]:rown[4]])))[3:7])/4)))
```

E.3 Windows

```
#%%
import numpy as np
from matplotlib import pyplot as plt
from scipy.fft import fft, fftfreq
import pandas as pd
from math import ceil
import scipy.signal
#import random

#%%
df = pd.read_excel(r"C:\Users\Nisha\Documents\BEP\LSCImeting.xlsx",
                   sheet_name = 1, header = 29, usecols="A:F", nrows = 2544)

#%%
ROI3 = np.array(df['3. ROI'])
ROITHREE = fft(ROI3)

#%% Properties
time = 42*60+48
N = len(df)
H = N//2

n = np.arange(N)

#%%
def movavg(sig, val):
    init = np.zeros(len(sig))
    for i in range(val):
        init[i] = np.array([1/(val+1)])
    print(init)
    outp = np.convolve(sig, init)[:len(sig)]
```

```
    return outp

#%%
ROI3smooth = movavg(ROI3, 50)
ROI3smooth1 = ROI3smooth - np.mean(ROI3smooth)

#%%
a1 = ROI3 * np.hanning(N)
a2 = movavg(a1,50)

#nothing new for a3 4 and 5
a3 = (ROI3-np.mean(ROI3)) * np.hanning(N)
a4 = movavg(a3,50)

a5 = ROI3 * np.hamming(N)

#%%
#plt.plot(n, ROI3, label = 'ROI3')
#plt.plot(n, a1, label = 'Hanning window')
#plt.plot(n, ROI3smooth, label = 'Smooth')
plt.plot(n, a2, label = 'Smooth Hann window')
plt.plot(n, a3, label = "Mean removed and Hann window")
plt.xlabel("Time in seconds")
plt.ylabel("Change in dermal blood flow")
plt.show()

#%%
plt.stem(fftfreq(300)[:150], np.abs(fft(a4[2211:2511]))[:150])
    #-np.mean(a2[2211:2511])))[:150])
plt.title("Amplitude spectrum ROI 3 plateau phase")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.show()
```

E.4 Butterworth filter

```
#%%
import numpy as np
from matplotlib import pyplot as plt
from scipy.fft import fft, fftfreq
import pandas as pd
from math import ceil
import scipy.signal
#import random

#%%
def movavg(sig, val):
    init = np.zeros(len(sig))
    for i in range(val):
        init[i] = np.array([1/(val+1)])
    print(init)
```

```
    outp = np.convolve(sig, init)[:len(sig)]
    return outp

#%%
df = pd.read_excel(r"C:\Users\Nisha\Documents\BEP\LSCImeting.xlsx",
                   sheet_name = 1, header = 29, usecols="A:F", nrows = 2544)

ROI3 = np.array(df['3. ROI'])

#%% Properties
time = 42*60+48
N = len(df)
H = N//2
fs = N/time

n = np.arange(N)

#%%
ROI3smooth = movavg(ROI3, 50)
#60 is enough?
ROI3smooth1 = ROI3smooth - np.mean(ROI3smooth)

#%%
order = 5
cutoff = 0.005
analog = False
b,a = scipy.signal.butter(order,cutoff, "low", analog , "ba", fs)
c = scipy.signal.filtfilt(b,a,ROI3)

#%%
plt.plot(n, ROI3, label = 'ROI3')
plt.plot(n, c, label = "Butterworth high pass")
plt.xlabel("Time in seconds")
plt.ylabel("Change in dermal blood flow")
plt.show()

#%%
plt.stem(fftfreq(300)[:30], np.abs(fft(c[2211:2511]
    -np.mean(c[2211:2511])))[:30])
plt.title("Amplitude spectrum ROI 3 plateau phase")
plt.xlabel("Frequency in Hz")
plt.ylabel("Amplitude")
plt.show()
```

Bibliography

[1] Cédric Gollion et al. 'Migraine and large artery atherosclerosis in young adults with ischemic stroke'. In: *Headache: The Journal of Head and Face Pain* 62.2 (2022), pp. 191–197. DOI: https://doi.org/10.1111/head.14265. URL: https://headachejournal.onlinelibrary.wiley.com/doi/abs/10.1111/head.14265.

[2] Hendrikus J.A. van Os et al. 'Migraine and Cerebrovascular Atherosclerosis in Patients With Ischemic Stroke'. In: *Stroke* 48.7 (2017), pp. 1973–1975. DOI: 10.1161/STROKEAHA.116.016133. URL: https://www.ahajournals.org/doi/abs/10.1161/STROKEAHA.116.016133.

[3] Simona Sacco et al. 'Peripheral vascular dysfunction in migraine: a review'. In: *The Journal of Headache and Pain* 14.80 (1 2013). DOI: 10.1186/1129-2377-14-80. URL: https://doi.org/10.1186/1129-2377-14-80.

[4] Wei Qi. 'Fourier and Wavelet Analysis of Skin Laser Doppler Flowmetry Signals'. MA thesis. University of Southampton: CED Group, School of Engineering Sciences, 2011. URL: https://eprints.soton.ac.uk/334190/.

[5] A. Stefanovska, M. Bracic and H. D. Kvernmo. 'Wavelet analysis of oscillations in the peripheral blood circulation measured by laser Doppler technique'. In: *IEEE transactions on bio-medical engineering* 46 (10 1999), pp. 1230–1239. URL: https://doi.org/10.1109/10.790500.

[6] Irina Mizeva et al. 'Spatial heterogeneity of cutaneous blood flow respiratory related oscillations quantified via laser speckle contrast imaging.' In: *PLoS ONE* 16.e0252296 (2021). DOI: 10.1371/journal.pone.0252296.

[7] J. Buijs, J.v.d. Gucht and J. Sprakel. 'Fourier transforms for fast and quantitative Laser Speckle Imaging'. In: *Scientific Reports* 9.13279 (2019). DOI: 10.1038/s41598-019-49570-7.

[8] Norbert Wiener. 'Generalized harmonic analysis'. In: *Acta Mathematica* 55.None (1930), pp. 117–258. DOI: 10.1007/BF02546511.

[9] W. Lu and N. Vaswani. *The Wiener-Khinchin theorem for non-wide sense stationary random processes.* 2009. arXiv: 0904.0602 [math.ST].

[10] Gemma Lancaster. 'Oscillations in microvascular flow: their relationship to tissue oxygenation, cellular metabolic function and their diagnostic potential for detecting skin melanoma'. PhD thesis. Department of Physics, Lancaster University, 2016. URL: https://eprints.lancs.ac.uk/id/eprint/78134/1/2016lancasterphd.pdf.

[11] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing.* California Technical Publishing, 1998. URL: www.DSPguide.com.

[12] *Fast Hann Function implementation.* https://medium.com/@lalesena/what-you-should-know-about-hann-function-790270460a75. Accessed 30/06/2023.

[13] *Signal filtering in Python.* https://swharden.com/blog/2020-09-23-signal-filtering-in-python/. Accessed 30/06/2023.

[14] *Digital Low Pass Butterworth Filter in Python*. https://www.geeksforgeeks.org/digital-low-pass-butterworth-filter-in-python/. Accessed 30/06/2023.