Technical University of Denmark

Matteo Saibene (s192730)
Andreas Hylleberg (s164277)

# BCI speller using SSVEP

January 21, 2020

Biomedical Engineering
Department of Electrical Engineering

# TABLE OF CONTRIBUTORS

| Section | Contributors |
|---|---|
| Abstract | Andreas |
| 1 | Matteo |
| 1.1 | Matteo |
| 1.2 | Matteo |
| 1.3 | Matteo |
| 2 | Matteo |
| 2.1 | Matteo |
| 2.2 | Matteo |
| 2.3 | Matteo |
| 3 | Matteo |
| 3.1 | Andreas |
| 4 | Matteo |
| 4.1 | Matteo |
| 4.2 | Andreas |
| 5 | Andreas |
| 5.1 | Andreas, Matteo |
| 5.2 | Andreas |
| 5.3 | Andreas |
| 5.4 | Andreas |
| 5.5 | Andreas |
| 6 | Andreas |

# Abstract

Brain-Computer-Interface (BCI) systems based on the Steady-State Visual Evoked Potential (SSVEP) paradigm, which have received increased attention in the last decade, due to it's promising capabilities of application. BCI's provide a wide range of functionality to severely debilitated people that have little to no means of interacting with the world. Motor neuron degenerative diseases like Amyotrophic Lateral Sclerosis (ALS) or trauma-induced neurological disabilities like Locked-In Syndrome, have a huge impact on the individual's health and quality of life. Recent studies show that ALS has in recent years suggested a growing tendency in prevalence[1], which makes the subject of investigation even more attractive.

This report reviews, replicates methods and sheds additional light on the work done on the DTU BCI speller from 2013[2]. The method takes basis in an non-bias autocorrelation algorithm for signal filtering, while utilising Fourier Transformations for classification and fundamental and harmonic frequencie differentiation. The paper reviews aims to verify it's accountability in terms of success and accuracy under low data-resolution conditions while appending to it's future development. Overall, the method was reviewed and found to successfully classify frequencies with high accuracy, despite only getting data from one electrode placed on the midline occipital "Oz" position. The dataset provided, consisted of data recorded from five subjects that partook the study experiment. The main objective of the experiment was to spell sentences with the integration of the danish dictionary support to minimize needed task-time, maximize Information Transfer Time (ITR) and Characters Per Minute (CPM) and overall contributing to user-friendliness.

The method of feature extraction and classification showed a total average accuracy of 85% which is relatively lower than expected, which was mostly due to faulty classification of Subject 3. Exclusion of this subject, showed a then total average accuracy of 95% which is very feasible and compares well to the findings in the DTU BCI speller study and other classification methods for the same paradigm. The classification method is reviewed in order to touch upon its ability to adapt and take external factors like noise, subject variety and algorithm short-comings into account. On the same note, other classification methods are reviewed and compared with regard to their respective domains of limitations, strengths and weaknesses.

# Table of Contents

# 1 Introduction

The goal of the study was to create a BCI speller[2] using SSVEP signal recorded using a non-invasive EEG system, to help people with LIS to communicate through usual means. LIS is a condition where the patient is fully conscious, but is unable to speak or moving. In early stages of LIS, patients are able to move their eyes and blink, but when the severity of the disease progresses, LIS patients are unable to blink and move their eyes while keeping their cognitive function.

## 1.1 Steady State Visual Evoked Potentials

Evoked potentials are generated by the brain in response to a particular stimulus. Steady state visually evoked potential (or SSVEP) are common types of evoked potentials and SSVEP is the stimulated response of neuronal activity in the visual cortex when the subject is gazing at a flickering pattern, flickering at a particular frequency. The brain signal was recorded using an EEG system and the signal exhibit characteristic peaks at the flickering frequency the subject is gazing at. Sometimes the power spectrum of the recorded signals shows higher amplitudes at the harmonic frequency of the fundamental which will be covered in the methodology chapter 3.
SSVEP can be used for a variety of applications within the medical field, like control of prosthesis and wheelchairs, verification of of cognitive integrity or within the non-medical field for entertainment etc... .
SSVEP signal can be recorded using a non-invasive BCI system (Fig. 1.1), that allows easy signal recordings among different subjects.

## 1.2 Comparison with p300 speller

P300 evoked potential can be also used to develop a so-called P300-based BCI speller. P300 wave occurs in response to relevant but infrequent stimuli and the width of the P300 is greater the lower the probability of appearance of the stimulus. P300 is considered an endogenous potential because its presence it is not related to a physical attribute of the stimulus (shape, color, etc ...), but because of a person's reaction to the stimulus. Moreover, the reproducibility of P300 signal makes it a common choice for both psychological and clinical testing in the laboratory.
In P300-based speller, the target letter is selected when there's intersection of the row and column. Unlike P300 speller, SSVEP speller produce a precise and detectable frequency of the stimulus and SSVEP-based speller it's capable or reaching very high speed (10-12 words/minute) and high information transfer rate (ITR).

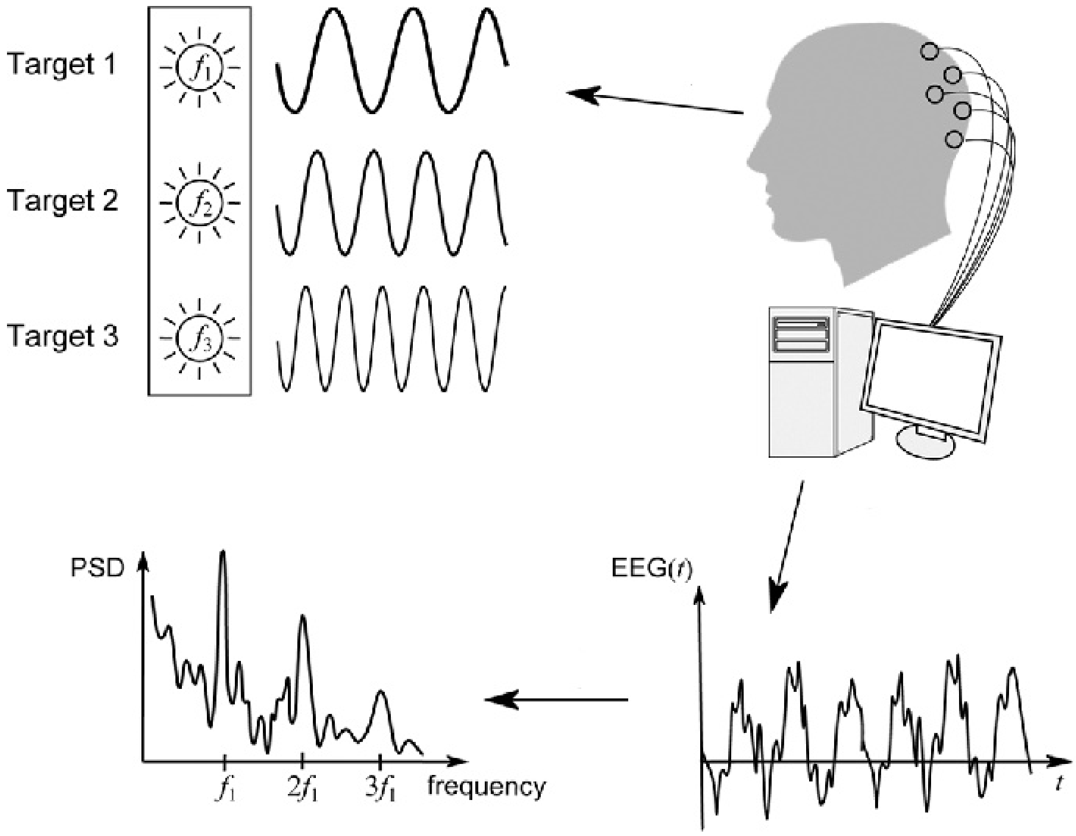


***Figure 1.1.*** Schema of SSVEP decoding approach

## 1.3 Related work

Another approach to BCI-based speller is addressed by Erwei Yin [3] used which claimed to increase the accuracy and information transfer rate of a conventional P300 speller using SSVEP. They integrated periodic and random flickering patterns to evoke both P300 and SSVEP. The results showed 100% accuracy level in 5 of the 12 subjects, while the average accuracy among 12 subjects was 93.85% . By integrating P300 speller with SSVEP, this pilot study along with the proposed BCI speller, could achieve better and more stable performance compared with conventional P300 speller.
Another way of using SSVEP signal was proposed by A. Saboor [4] who used SSVEP signal generated with Epson Moverio BT-200 augmented reality glasses and recorded with a portable EEG, in a smart home scenario. He calmed to reach a high accuracy level which suggests that AR (augmented reality) may be used along with SSVEP in a smart home scenarios.

# 2 Experiment setup

## 2.1 Description of the data

For the investigation and detection of Steady-State Visually Evoked Potential, the free available dataset from Adnan Vilic[5] was used. The dataset was procured and recorded for a master thesis and made available for the purpose of non-commercial projects to get started with SSVEP response detection.
The dataset was acquired from recordings of nine total subjects being exposed to respectively single- [Section 2.3] and multi-target [Section 2.2] stimulation. [Fig. 2.1] shows the setup used for the experiments. All data was recorded using three electrode: the signal electrode was placed at Oz, reference is at Fz ,and ground electrode is in position Fpz. Electrodes were placed using standard 10-20 electrode placement system . The only filtering applied on the data was an analogue notch filter at cutoff frequency of 50 Hz to counter main hum noise. Neither the circuit schematic and bode diagram of the filter was included in the documentation.

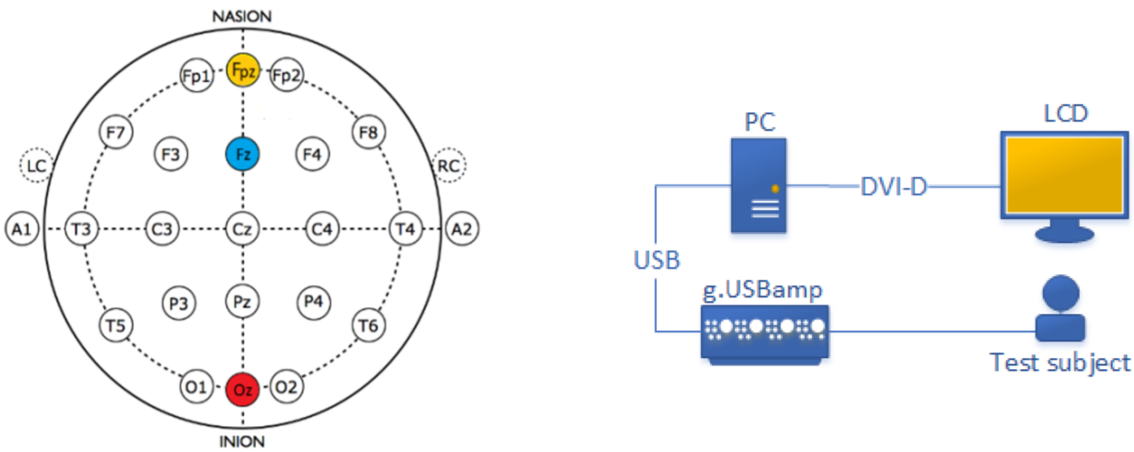


***Figure 2.1.*** Experimental setup and electrode position

## 2.2 Multi-target stimulation

In the multi-target stimulation setup, SSVEP signals were recorded from five different subjects with different gender and age [Fig. 2.3], looking at a specific target frequency, while other flickering targets were present on the screen. The stimulation matrix consisted of seven flickering targets, that would flicker at the frequencies 6Hz, 6.5Hz, 7Hz, 7,5Hz, 8,2Hz, 9,3Hz and 10Hz. For each target frequency the subject was gazing at, 16 seconds of EEG data was recorded with a sampling frequency of 512 Hz. The placement and dimensions of the targets in the multi-target

setup is shown in figure [Fig.2.2].
Each subject underwent to two different trials:

- Trial 1) The subject was asked to gaze at one target frequency for a total time of 16 seconds. Flickering frequencies are sequenced as follow: $trial\_1 = [7.5, 8.2, 7, 8.2, 6, 7.5, 6, 6, 8.2, 8.2]$ .
- Trial 2) The subject was asked to gaze at one target frequency for a total time of 16 seconds. Flickering frequencies are sequences as follow: $trial\_2 = [9.3, 6, 6, 8.2, 6, 6.5, 7.5, 7, 6, 6]$ .

In this experiment, the subject is seated 60cm away from an LCD Benq XL2420T 24" monitor with a refresh rate of 120Hz. The reason why this monitor with high refresh rate was used for this experiment is not clearly stated in the documentation of the experiment, but the author in his website [cite website... ], mentioned that a flickering frequency of 7.2Hz easier to produce in a 120 Hz monitor, rather than 30 Hz on a 60 Hz monitor.
According to the author, there is also a great difference in how LCD (liquid crystal display) and CRT (cathode ray tube) monitors work, and what refresh rate on monitors means. A CRT monitor updates the whole screen in every frame so you are certain that everything is refreshed, and hence, the refresh rate indicates how fast the whole screen can be updated. On the other hand, LCD monitors instead of updating the whole screen in every frame, update only the part that the monitor knows has changed.
Dataset was compiled as a structure containing a multi-dimensional matrix. [Fig. A.4].

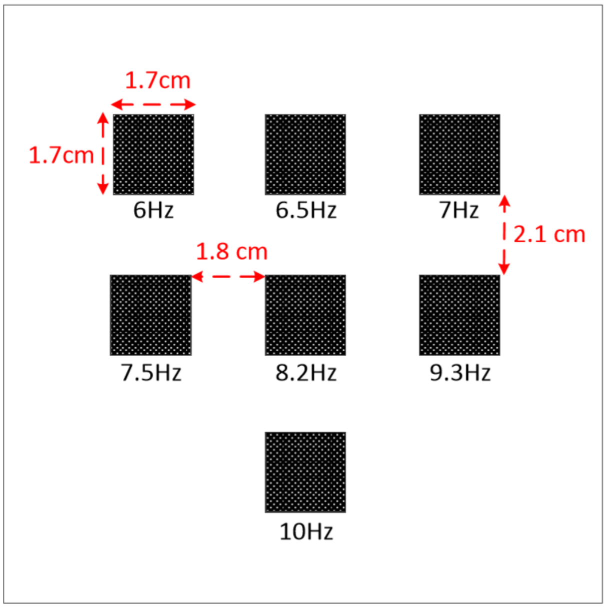


***Figure 2.2.*** Target Placement

| Subject | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Gender | Male | Female | Male | Female | Male |
| Age | 32 | 27 | 27 | 27 | 26 |

***Figure 2.3.*** List of participants for multi-target flickering

## 2.3 Single-target stimulation

The author of the experiment also included in the data set a single-target stimulation [Fig. A.2], where four different subjects were gazing at one flickering target in the screen. The setup of the experiment was the same as for multi-target stimulation, except that there was only one flickering target in the screen, and the resulting EEG signal was recorded for 30 seconds for each trial.

This dataset was not used to test the classifier, but it was used to prove that the flickering frequency was detected in the power spectrum of the signal. As the flickering frequency was successfully detected in this test, we decided to carry on the experiment with multi-target stimulation. If not, further investigations should be made.

# 3 Methods

A BCI speller was made using the provided SSVEP data and the proposed classification method described in [2].
Since the dataset contains a total of 10 trials (5 subject x 2 trials for each), and each trial contains 10 signals with 16 seconds length, the subject, trial and target frequency are selected by the user, prior to signal processing and classification. This makes the code lighter and faster because the dataset is not loaded entirely at the beginning of the session.
After loading the sample data into MATLAB, auto-correlation is applied to reduce background noise. The function autom(x) was declared at the bottom of the script. The auto-correlation function[6] of a signal describes the dependence of the values of the samples at one time on the values of the samples at another time.
The auto-correlation function was written based on equation [Equation: 3.1], $for\ m = 1, 2, \ldots, M+1$

$$R_{xx}(m) = \frac{1}{N} \sum_{n=1}^{N-m+1} x(n)x(n+m+1) \tag{3.1}$$

From this function, the properties of maximum value, periodicity, and symmetry can be derived. It should be considered that the auto-correlation function of a periodic signal is also periodic. Figure [Fig. A.1] in the annexes shows how auto-correlation function reduces the noise in the signal prior to the classification process.

***Listing 3.1.*** Auto-correlation function

```matlab

function [Rxx]=autom(x)
    % [Rxx]=autom(x)
    % This function Estimates the autocorrelation of the sequence of
    % random variables given in x as: Rxx(1), Rxx(2),...,Rxx(N), where N is
    % Number of samples in x.
    N=length(x);
    Rxx=zeros(1,N);
    for m=1: N+1
        for n=1: N-m+1
            Rxx(m)=Rxx(m)+x(n)*x(n+m-1);
        end;
    end;
end
```

After autocorrelation, the FFT was applied to the signal using Welch's method, instead of using a periodogram to estimate the spectral density of the signal. Welch's method aims to estimate the Power Spectral density of a signal from a sequence of time-samples. In order to implement

this method in matlab, the pwelch function was used.
According to matlab documentation, the pwelch function is defined as:

***Listing 3.2.*** pwelch function in Matlab

```
[pxx ,f]= pwelch(x,window,n_overlap,nfft,Fs);
```

Where:

- x is the given signal.
- Window: if this input is empty, the default Hamming window is used. In our case, we used a rectangular window defined as $rectwin(nfft)$, because the default Hamming window was distorting the signal.
- *N_overlap* is an integer number that defines the number of overlapping samples.
- Nfft is an integer positive number that defines the number of points in the DFT.
- Fs is the sampling frequency.

After auto-correlation and utilising Welch's method, the frequency spectrum was plotted as reference, with a frequency range from 4 to 21 Hz.
Unlike the proposed method ([2]), zero-padding is not used for our purposes, because the length of the signal was enough to have a good accuracy of the DTF. After processing the signal, classes $C_x$ were identified. Each class represents a target frequency and the value of each class, $C_x$, is the sum of power amplitudes, |Y|, around the relevant frequencies.
The equation below [Equation: 3.2], shows how each class was identified:

$$c_x = \sum_{F1-0.1}^{F1+0.1} |Y| + \sum_{2\cdot F1-0.1}^{2\cdot F1+0.1} |Y| \tag{3.2}$$

Where F1 is the fundamental frequency, while $2 \cdot F2$ is the first harmonic of the fundamental frequency F1. The reason why this method also accounts for the first harmonic, is because on some subjects, the amplitude of the harmonic frequency is higher than the amplitude of the dominant frequency. This method ensures that each class $Cx$ is always detected. Each class is then normalized, so the dominant class will have a value of 1.
The results are then plotted using *bar* function in matlab.

### 3.0.1 Implementation

The proposed method was implemented using MATLAB, where the dataset was loaded into matlab environment. In order to make the selection of the subject, trial, and target frequency easier, the user can insert these data in the command window [Fig. 3.1] via the implementation of a simple console user interface, after running various trials. Moreover this makes the code lighter and enhances the computational speed.
The full code can be found in the annexes (A).

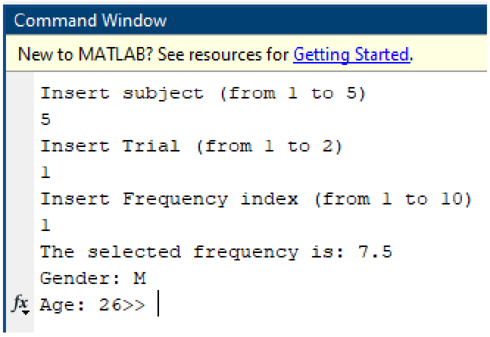


***Figure 3.1.*** Example of input in the command window

## 3.1 Performance evaluation

The verification of detected frequencies correlating with the expected frequencies was done by false/true positive evaluation. Due to the small datasets as well as the classification used, it was deemed inexpedient to do Receiver Operation Characteristics (ROC) curves to showcase the accuracy and performance. The classifier being developed doesn't train data in order to approximate parameters for best-fitting detection for each subject, making the evaluation near-impossible by ROC and epochs. On a related note, ITR was also deemed inexpedient, as the dataset does not provide information on tasks or time spent, but merely aims to detect SSVEP. In order to evaluate the performance of the classifier, an algorithm was developed that takes basis in false/true positives and the output [Fig. A.3] of the algorithm was than analysed in section [4.2]. The performance algorithm compared all detected frequencies to it's expected target frequency and was done for all subjects and trials. If the expected correlates with the detected, a true boolean is returned and returns a false boolean for non-correlation. By doing this, we're able to investigate trends among subject accuracy and detection accuracy and recognise possible errors within the system. The approach for the algorithm looks as following,

- For 'n' subjects (n=5), start iterative performance test on 'm' trials
- For each trial 'm' (m=2), check correlation between expected vs detected frequency for all trial frequencies 'b' (b=10). Correlation here implies true equality between values.
- If correlation is true, append boolean '1' to it's respective trial row in a matrix
- If non-correlation is true, append boolean '2' to it's respective trial row in a matrix
- Iterate all 'b' frequencies for all 'm' trials for 'n' subjects
- Console outputs binaries, subject and total accuracy for each iteration to give the operator the possibility of closer investigation.
- A final matrix is produced containing all binary results of the performance test
- Compute matrix to plot statistics of subject accuracy and target frequency accuracy

# 4 Results

As described in the previous metholodogy section, the different methods to extract and classify the EEG signal was developed and a performance evaluation was done to test it's viability and accuracy. Results were as following:

## 4.1 Detection classification

The figure below [Fig. 4.2] shows an example of successful classification, where right frequency was identified despite the amplitude of the harmonic frequency being higher, than the amplitude of the fundamental frequency. The subject was gazing at the flickering box 1 that has a target frequency of 6 Hz. As can be seen from the bar chart below, the classifier identifies the target frequency of 6 Hz, despite the harmonic frequency 12 Hz has the highest amplitude in the SSVEP signal.

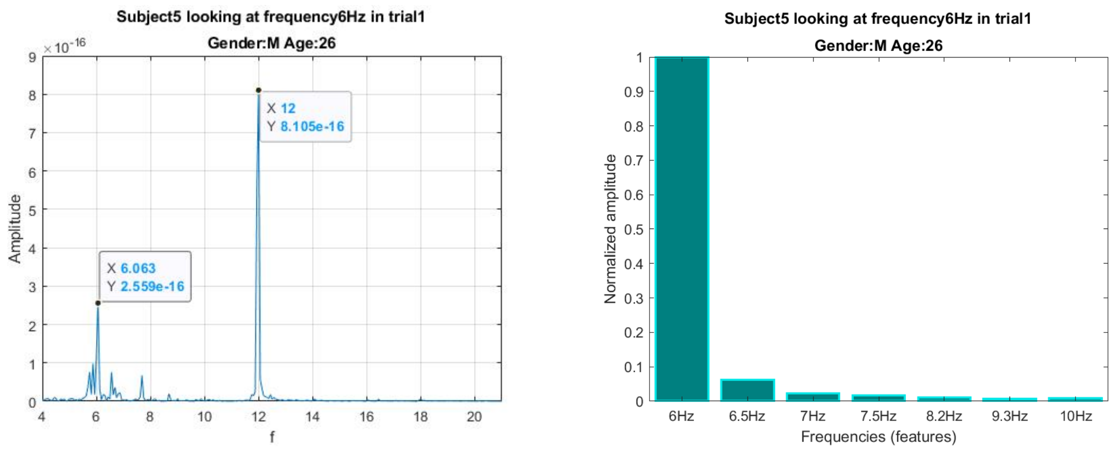


***Figure 4.1.*** Subject 5, Trial 1, frequency 6Hz

Even though the classifier as shown in figure [Fig. 4.2] classified frequency 7.5Hz as dominant, it can be seen in investigation of the signal in the frequency spectrum of the SSVEP signal, that other frequencies corresponding to flickering patterns close to the dominant frequency are present. This relevant result shows that the classifier discriminates unwanted frequencies.

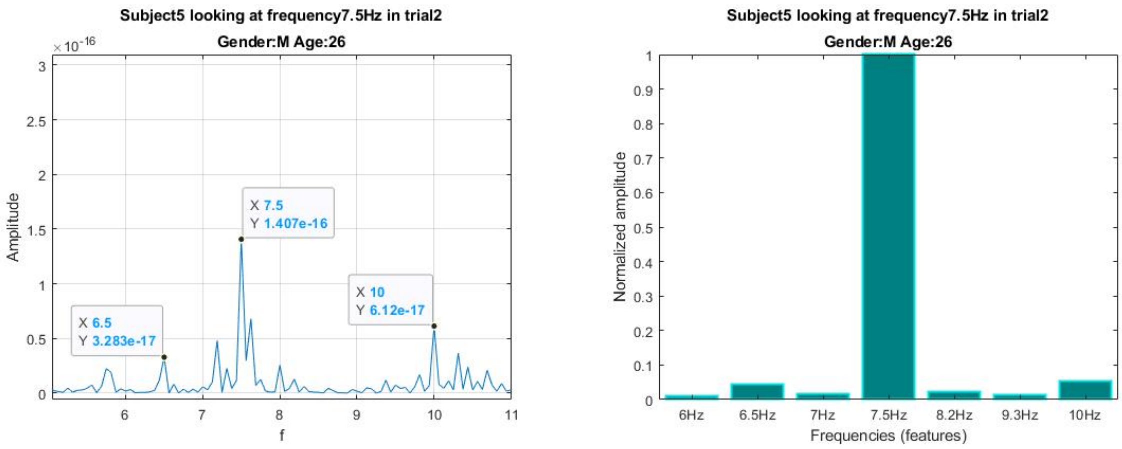


***Figure 4.2.*** Subject 5, Trial 2, frequency 7.5 Hz

Figure [Fig. 4.3] shows a good example of successful classification in which the proposed method classifies the frequency 7.5 Hz as the flickering frequency of the pattern the subject is focusing at. The spectrum of the signal, clearly shows two clear and distinct peaks corresponding to the fundamental frequency and its harmonic (7.5Hz and 15 Hz).

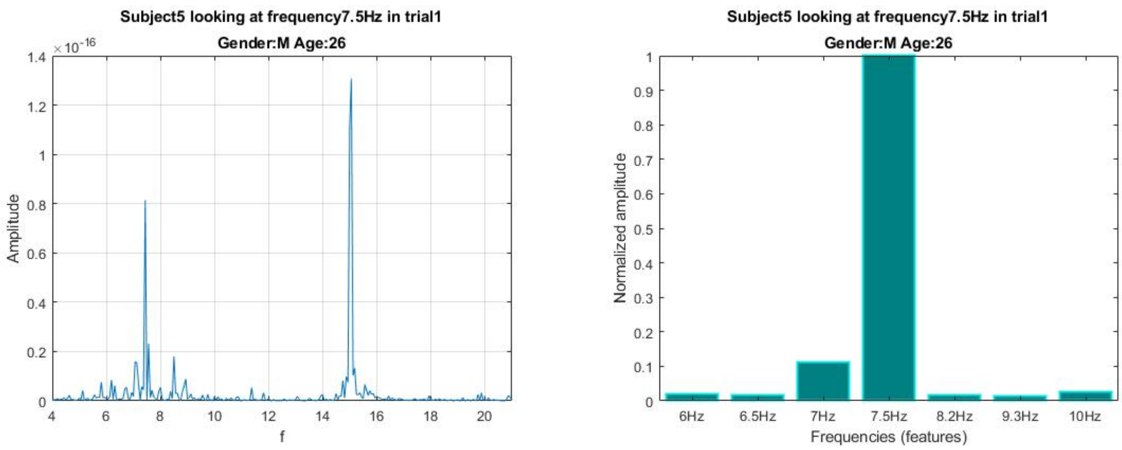


***Figure 4.3.*** Subject 5, Trial 1, frequency 7.5 Hz

## 4.2 Performance

The total average accuracy of the algorithm was found to be 85% which is close to the accuracy of the done work with the same proposed classification algorithm[2].

| **Subject** | **Average accuracy on two trails** |
|---|---|
| Subject 1 | 85% |
| Subject 2 | 100% |
| Subject 3 | 45% |
| Subject 4 | 95% |
| Subject 5 | 100% |
| **Total** | 85% |

***Table 4.1.*** Target frequency accuracy

It's already apparent by looking at table 4.1 and figure 4.4 that subject 3 had performance issues. This could be an indication that an underlying issue with the setup, the pre-processing and/or classification could be present. Due to the small population size and limited data size, it impacts the performance quite a bit and the resultant accuracy is relative to it's accountability. If subject 3 was to be excluded, the total accuracy would be 95%. A way to investigate if any target frequencies had classification issues, correlation between expected and detected was evaluated over the different target frequencies for all subjects as described in 3.1. Subject 3's misclassifcations had a bias towards the 6-7Hz range, but without a bigger data size it's not safe to say for certain, as the trend isn't evidently shared among the other subjects.

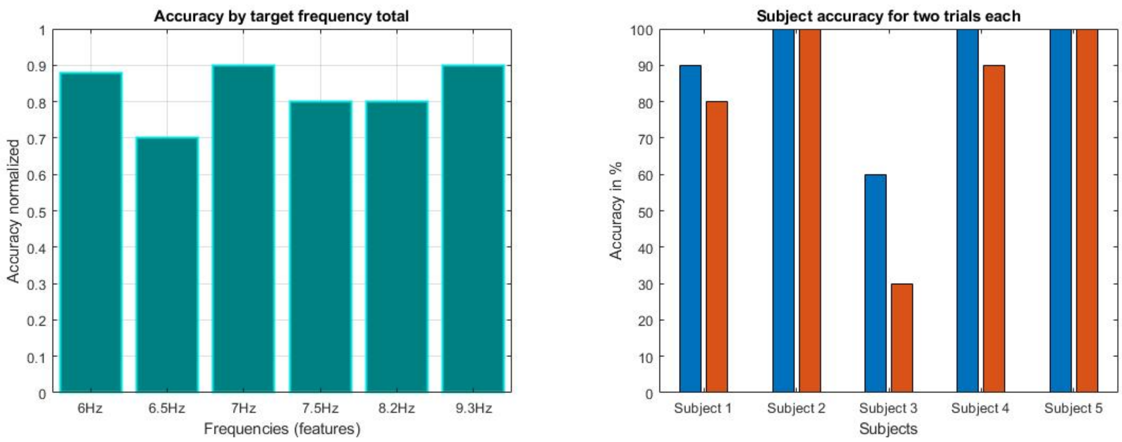


***Figure 4.4.*** Subject and target accuracy respectively

Overall does the performance looks pretty successful for the setup and compares well to work done with similar setup and classification.

# 5 Discussion

The work done verifies the proposed method's viability and validity. Overall, the developed classifier is very satisfactory and provided high accuracy from one single electrode signal source. The biggest critical point for the method evaluation was the low population size and small data-set, making the resultant evaluation accountability a bit relative. Generally to test a method, a larger population size is preferable, as a five subject experiment study doesn't provide an upright picture of the system's robustness. The investigation did find flaws and errors within the system, and opened up for possible future implementations and improvements, which will try be addressed in the following chapter.

## 5.1 Sources of error

Wrongful detection was found under performance evaluation and preliminary spectral analysis, which came to show as artefacts with higher amplitude than the expected frequency.

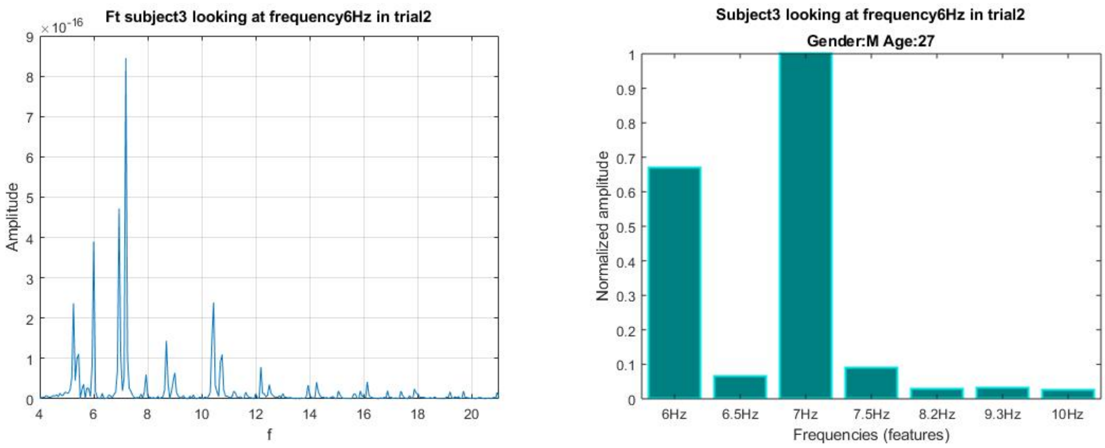


***Figure 5.1.*** Detection of wrong target frequency, 6Hz was stimulated, 7 Hz detected

As seen here above in 5.1, the system showed tendencies to misclassify features, with a slight bias towards around 7Hz $\pm 0,5Hz$ band as shown previously in 4.4.

### 5.1.1 Natural artefacts

Different types of perturbations can have a huge impact on the performance as the the artefacts can change the shape of the neurological phenomenon during intentional control resulting in a decrease in true-positive ratio or by mimicking the shape and properties of the neurological phenomenon during non-control periods, resulting in an increase of false-positive rates[7]. A 2018 paper [8] systematically evaluated natural artefacts for SSVEP-based BCI systems, where

various perturbations were induced and tried countered by different methods. Subjects were instructed to speak, think and listen to perturbate the signal. It was here found that detection of perturbations wasn't related to a specific classifier. Other types of natural artefacts would i.e. be EOG or EMG, where little EMG interference were to be expected due to the 1-30Hz analogue band-pass filter and though the EOG frequency bands lies within the target frequencies, it shouldn't be an expected prominent artefact at the midline occipital electrode position. This could imply that if the subject is not being fully concentrated on the task at hand, it could make it difficult for the system as cognitive tasks start interfering with the signal.
Other interference that could be presented in the recorded signal could be caused when the subject is not focused on the specific flickering target, causing other neighbour frequencies to be shown in the signal.

### 5.1.2 Flickering interference

The BCI setup used a LCD monitor to produce stimulus from several target frequencies. A factor to take into consideration that could affect the signal and classification, are inconsistencies when refreshing the LCD monitor. For fixed precision of the stimulus frequency, control of the refresh rate is important. There's the option of either refreshing via frames or time. If high precision in refresh rate is sought after, the system should have complete control over the graphics card and constantly verify and check frames. This would mean that the system should be as distributed as possible, so the spelling matrix is the only thing that needs updated to avoid background processes interfering with the refresh rate. LCD monitors only update the part of the screen that is different compared to the previous frame, so it should be optimal for this study. It's all dependent on the technology used, but could be a possible culprit for faulty classification. To summarise, you cannot assume that the setup will always produce accurate frequency flickering and some frequencies will be easier to produce, where other frequencies require bigger sampling to detect proper. A method from a paper in 2012[9] addresses this issue by a proposed method based on Multiple Frequencies Sequential Coding and was found to be more efficient than current traditional protocols.

## 5.2 Detection optimisation

The better the stimulus performs, the bigger spike and steepness in the feature to be classified. As noted in performance there was observed issues with faulty classification. As described in the previous section, a variety of error sources could be affecting the classifier. Though, if we know the target frequency to classify, an implementation of adaptive resolution would increase the accuracy of classification. In example, looking to classify a 6,5Hz stimulated frequency the algorithm could implement means of training a parameter to determine resolution under calibration, as the stimulated response will vary from subject to subject. This means, if the stimulus has a high performance, a low resolution would drastically improve accuracy and classification. It's basically a further development of the zero-padding used to obtain a 0,1Hz resolution in the DTU BCI speller paper.

## 5.3 Scalability and ability to adapt

Testing the proposed method on data procured from a different setup, would've been a great way to cross-reference it's viability. Since the method doesn't include parameter approximation like other classifiers do, it's safe to assume that it would under-perform when being scaled up. The method doesn't own the the ability to adapt to subject and setup differences, which is why choosing an adaptive classifier that utilises training of hyper-parameters is proved to be superior on terms of result validity[10]. If the system were to become more autonomous and less rigid, a more intelligent and adaptable method of classification should be investigated.

## 5.4 Comparison of classifiers

As covered in the report, the primary methodology used for feature extraction was doing auto-correlation on the signals, while feeding Welch's Power Spectral Density (PSDA) estimation function with the auto-correlated Fourier transformed signal to detect and classify frequencies. A custom classifier that sought after the biggest magnitude of frequency spikes, was built to detect the features, both natural and harmonic.

The most popular linear classifiers for SSVEP-based BCI systems are currently Linear Discriminant Analysis (LDA) or Support Vector Machines (SVM). Most reported systems use combinations or modifications of these type of classifiers, of which chosen classification depends on the shape of the acquired signals, level of pre-processing and methods used for feature extraction. LDA is not always the most suitable method as it does not tackle problems with missing class labels very well. Since LDA is a binary classifier, only spatial/temporal association to the data samples indicates class membership. Canonical Correlation Analysis (CCA) which seeks correlated projections between two views of data, has proved to be equivalent in terms of efficiency to LDA[11]. A paper from 2011 investigated performance for different types of canonical correlation methods[12], showing Multiway CCA (MCCA) to have high accuracy with low detection time in comparison to standard CCA and PSDA and combined types of PSD+LDA within the SSVEP paradigm. CCA is currently being considered the state-of-the-art feature extraction method in comparison to other well-known methods of feature extraction and classification. The comparison study also showed MCCA to be superior to standard CCA, PSDA and PSDA+LDA.

## 5.5 Future work

In order to make BCI spellers more available for end-users, both in terms of cost and usability, the system design should aim to maximise the ITR (Information Transfer Rate), minimise data size and transfer rate, while maintaining an acceptable accuracy. As reviewed in this report, high accuracy was proven to be possible with recordings from one electrode source. Expanding on the classifier system developed, it would be relatively simple to implement a BCI with merely the developed code, similar to the DTU BCI speller. Since the system is lightweight in terms of equipment and setup in comparison to most BCI systems, a variety of applications are imaginable. People with severe motor disabilities are stripped away of their ability to interact with their environment, where this type of BCI speller could easily be integrated and modified into a "smart-house"[13], as an user interface with multi-purposes, giving freedom and control back to the implicated person.

As mentioned in the introduction section, related work done combining SSVEP with the P300 paradigm[3], showed a hybrid system with a even greater accuracy. This opens up the possibility of other type of hybrid BCI's. Gaze tracking is far superior to SSVEP-based BCI's in terms of CPM, ITR and user-friendliness[14], and since both methods require same conditions, a future possibility would be developing a hybrid BCI consisting of eye/gaze tracking and combining this with P300 or SSVEP based paradigms. It's imaginable that eye- or gazetracking could work as the main controller for various smart-home user interfaces, while utilising the major BCI paradigms to control selections by performing mind-clicks. This would increase the system's usability by lowering the cognitive workload for the user, making the application more usable and efficient.

# 6 Conclusion

In conclusion, the proposed classification method was processed and found to be quite efficient in it's simplistic system design and methodology, with relatively high accuracy results. The work gave insight as to how you can approach classification through different channels and provided inspiration towards possible system expansions. The project work shed some important light on the different computational approaches available in order to build a robust classifier for SSVEP-detection. Investigation of these approaches included a review of some popular digital signal-processing techniques currently used for feature extraction and event classification within the linear domain, while suggesting how these differentiates in conjunction with one another.

Further development of the system should focus on optimising the build classifier, i.e. an implementation of a linear type classifier, while keeping emhpasis on maintaining acceptable ITR rates and high precision.

# Bibliography


[1] Elisa Longinetti and Fang Fang. *Epidemiology of amyotrophic lateral sclerosis: an update of recent literature*. Oct. 2019. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6735526/.

[2] Adnan Vilic, Troels W. Kjær, Carsten E. Thomsen, S. Puthusserypady, and Helge B. D. Sørensen. *DTU BCI Speller: An SSVEP-based Spelling System with Dictionary Support*. July 2013. URL: https://orbit.dtu.dk/en/publications/dtu-bci-speller-an-ssvep-based-spelling-system-with-dictionary-su/.

[3] Erwei Yin, Zongtan Zhou, Jun Jiang, Fanglin Chen, Yadong Liu, and Dewen Hu. "A novel hybrid BCI speller based on the incorporation of SSVEP into the P300 paradigm". In: (Feb. 2013).

[4] Abdul Saboor, Aya Rezeika, Piotr Stawicki, Felix Gembler, Mihaly Benda, Thomas Grunenberg, and Ivan Volosyak. "SSVEP-Based BCI in a Smart Home Scenario". In: *Advances in Computational Intelligence*. Ed. by Ignacio Rojas, Gonzalo Joya, and Andreu Catala. Cham: Springer International Publishing, 2017, pp. 474–485. ISBN: 978-3-319-59147-6.

[5] Adnan Vilic. *Adnan Vilic BCI dataset*. July 2013. URL: https://www.setzner.com/avi-ssvep-dataset/.

[6] S R Taghizadeh. *Digital Signal Processing Part 3. Discrete-Time Signals & SystemsCase Studies*. Jan. 2000. URL: https://www.academia.edu/11185613/Taghiszadeh_2000_-_Digital_signal_processing.

[7] M Fatourechi, R K Ward, and G E Birch. *A self-paced brain–computer interface system with a low false positive rate*. Dec. 2007. URL: https://iopscience.iop.org/article/10.1088/1741-2560/5/1/002.

[8] Zafer İşcan and Vadim V. Nikulin. *Steady state visual evoked potential (SSVEP) based brain-computer interface (BCI) performance under different perturbations*. Jan. 2018. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5779700/.

[9] Yangsong Zhang, Peng Xu, Tiejun Liu, Jun Hu, Rui Zhang, and Dezhong Yao. *Multiple Frequencies Sequential Coding for SSVEP-Based Brain-Computer Interface*. Mar. 2012. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0029519.

[10] Andreas Schulze Bonhage, Joern Rickert, Julie Blumberg, Stephan Waldert, Ad Aertsen, and Carsten Mehring. *Adaptive Classification for Brain Computer Interfaces*. Feb. 2007. URL: https://www.researchgate.net/publication/5843502_Adaptive_Classification_for_Brain_Computer_Interfaces.

[11] Olcay Kursun, Ethem Alpaydin, and Oleg V. Favorov. *Canonical correlation analysis using within-class coupling*. Nov. 2009. URL: https://www.cmpe.boun.edu.tr/~ethem/files/papers/olcay_cca_prl.pdf.

[12] Yu Zhang, Qibin Zhao, and Guoxu Zhou. *Multiway Canonical Correlation Analysis for Frequency Components Recognition in SSVEP-Based BCIs*. Nov. 2011. URL: https://www.researchgate.net/publication/221139405_Multiway_Canonical_Correlation_Analysis_for_Frequency_Components_Recognition_in_SSVEP-Based_BCIs.

[13] Nataliya Kosmyna, Franck Tarpin-Bernard, Nicolas Bonnefond, and Bertrand Rivet. *Feasibility of BCI Control in a Realistic Smart Home Environment*. Aug. 2016. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4999433/#!po=83.3333.

[14] Pasqualotto 1, Matuz T, Federici S, Ruf CA, Bartl M, Olivetti Belardinelli M, Birbaumer N, and Halder S. *Usability and Workload of Access Technology for People With Severe Motor Impairment: A Comparison of Brain-Computer Interfacing and Eye Tracking*. Dec. 2015. URL: https://journals.sagepub.com/doi/full/10.1177/1545968315575611.

# List of Figures

# A Annexes

| Field | Type | Size | Description |
| --- | --- | --- | --- |
| AmpSamlingFrequency | Double | | Sampling rate of amplifier. |
| EEG | Double[] | 89120x10 | Each column contains 16 seconds of EEG data from which SSVEP responses can be derived. |
| ElectrodePlacement | Char[] | 1x2 | Electrode position for obtaining SSVEP |
| FlickeringFrequencies | Double[] | 1x7 | All frequencies for targets that are flickering at the same time. |
| FzImpedance | Double | | Impedance at reference in kΩ |
| OzImpedance | Double | | Impedance at signal electrode in kΩ |
| RecordedDate | Char[] | 1x20 | Date of recording. |
| StimulusDevice | Char[] | 1x25 | LCD monitor used for flickering |
| SubjectAge | Char[] | 1x2 | Age of subject. |
| SubjectDistance | Char[] | 1x5 | Distance from monitor |
| SubjectGender | Char[] | 1x1 | Gender of subject |
| TargetFrequency | Double[] | 1x10 | Each column contains the flickering frequency of the target that the subject is supposed to look at. |
| TargetSize | Char[] | 1x5 | Size of flickering target. |

***Figure A.2.*** Single channel flickering entiity design

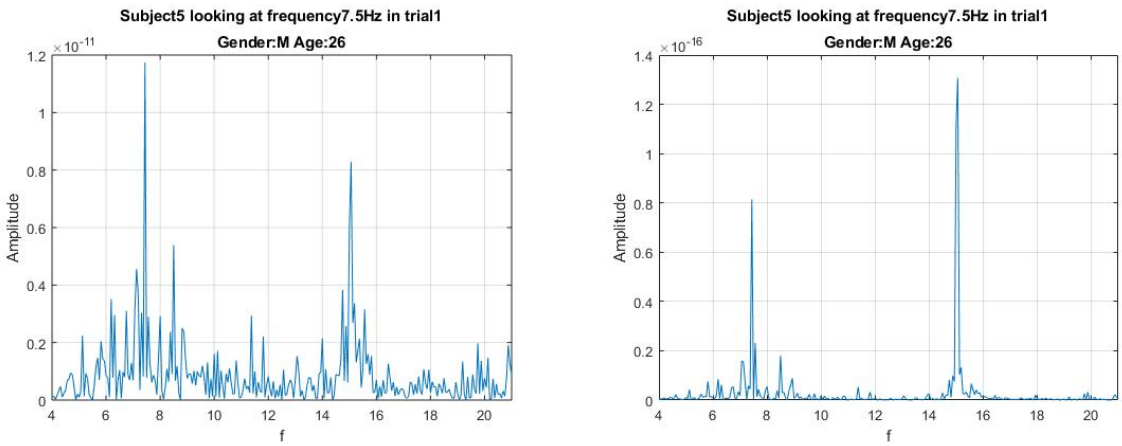


***Figure A.1.*** before/after auto-correlation

```
-----------------------------------------
1110111111
 Accuracy is 90% for Subject 1, Trial 1
0111101111
 Accuracy is 80% for Subject 1, Trial 2
>>Subject 1 accuracy for two trials is 85.000000%<<
-----------------------------------------
1111111111
 Accuracy is 100% for Subject 2, Trial 1
1111111111
 Accuracy is 100% for Subject 2, Trial 2
>>Subject 2 accuracy for two trials is 100.000000%<<
-----------------------------------------
1101000111
 Accuracy is 60% for Subject 3, Trial 1
1010000100
 Accuracy is 30% for Subject 3, Trial 2
>>Subject 3 accuracy for two trials is 45.000000%<<
-----------------------------------------
1111111111
 Accuracy is 100% for Subject 4, Trial 1
1111111011
 Accuracy is 90% for Subject 4, Trial 2
>>Subject 4 accuracy for two trials is 95.000000%<<
-----------------------------------------
1111111111
 Accuracy is 100% for Subject 5, Trial 1
1111111111
 Accuracy is 100% for Subject 5, Trial 2
>>Subject 5 accuracy for two trials is 100.000000%<<
```

***Figure A.3.*** Command window output of the evaluation algorithm

| Field | Type | Size | Description |
|---|---|---|---|
| AmpSamlingFrequency | Double | | Sampling rate of amplifier. |
| EEG | Double[] | 89120x10 | Each column contains 16 seconds of EEG data from which SSVEP responses can be derived. |
| ElectrodePlacement | Char[] | 1x2 | Electrode position for obtaining SSVEP |
| FlickeringFrequencies | Double[] | 1x7 | All frequencies for targets that are flickering at the same time. |
| FzImpedance | Double | | Impedance at reference in kΩ |
| OzImpedance | Double | | Impedance at signal electrode in kΩ |
| RecordedDate | Char[] | 1x20 | Date of recording. |
| StimulusDevice | Char[] | 1x25 | LCD monitor used for flickering |
| SubjectAge | Char[] | 1x2 | Age of subject. |
| SubjectDistance | Char[] | 1x5 | Distance from monitor |
| SubjectGender | Char[] | 1x1 | Gender of subject |
| TargetFrequency | Double[] | 1x10 | Each column contains the flickering frequency of the target that the subject is supposed to look at. |
| TargetSize | Char[] | 1x5 | Size of flickering target. |

***Figure A.4.*** Multi channel flickering entity design

***Listing A.1.*** Matlab code (without evaluation part)

```
clc
close all
clear all


%% Loading multi target for each subject
% length of each trial = 16 seconds

%% Subject selection....
retry = 1;
while retry == 1
    subject = input('Insert subject (from 1 to 5) \n', 's'); %from 1 to 5
    if str2num(subject) > 5 || str2num(subject) < 1
        fprintf ('Retry.....\n');
    else
        retry = 0;
    end
end
%% Trial selection....
retry = 1;
while retry == 1
    trial = input('Insert Trial (from 1 to 2) \n', 's'); % 1 or 2
    if str2num(trial) > 2 || str2num(trial) < 1
        fprintf ('Retry.....\n');
    else
        retry = 0;
    end
end

%% Frequency selection....
retry = 1;
while retry == 1
    targetFrequencyIndex = input('Insert Frequency index (from 1 to 10) \n', 's');
        %select the frequency that corresponds to index X
    if str2num(targetFrequencyIndex) > 10 || str2num(targetFrequencyIndex) < 1
        fprintf ('Retry.....\n');
    else
        retry = 0;
    end
end

targetFrequencyIndex = str2num(targetFrequencyIndex);


%target freq trial 1 = [7.5,8.2,7,8.2,6,7.5,6,6,8.2,8.2]
%target freq trial 2 = [9.3,6,6,8.2,6,6.5,7.5,7,6,6]
if trial == '1'
    freqVectorTrial = [7.5,8.2,7,8.2,6,7.5,6,6,8.2,8.2];
else
```

```
    freqVectorTrial= [9.3,6,6,8.2,6,6.5,7.5,7,6,6];
end

fprintf ('The selected frequency is: ');
frequency = freqVectorTrial(:,targetFrequencyIndex); %return the frequency of the
    selected index
fprintf (num2str(frequecy),'\n');
str = strcat('Dataset_1/multi/Sub',subject, '_', trial ,'_multitarget.mat');
%loading selected trial
Sub = load(str);

fprintf('\nGender: ');%gender
fprintf(Sub.Data.SubjectGender());
fprintf('\nAge: ');%Age
fprintf(Sub.Data.SubjectAge());

%% subject x trial x
%% autocorrelation

for i=1:10
    Sub.Data.EEG(:,i) = autom(Sub.Data.EEG(:,i)); %autom function explained below
    i = i+1;
end;
%% fft using pwelch
% since N is large, we don't need to perform zero padding to improve the ft.

Fs = 512;                        % sampling frequency
L = length(Sub.Data.EEG(:,targetFrequencyIndex)); % the length of the signal is the
    same for each experiment
nfft = L;                        % number of points in DFT
n_overlap = 1000;                % number of overlapped samples (used for DFT
    averaging)
window= rectwin(nfft);           % we tried using hanning window but the second signal
    gave us spikes at 5Hz


% subject x looking at frequency X in trial X
[pxxFt,f]= pwelch(Sub.Data.EEG(:,targetFrequencyIndex),window,n_overlap,nfft,Fs);
% f = Fs*linspace(0,1,L/2+1);%(0:(L/2))/L;
% ft = fft(autom( Sub1_1.Data.EEG(:,1) ),L)/L;
figure
    plot (f,pxxFt);
    title({strcat('Subject ',subject,'_ looking at frequency ',
        num2str(frequency),'Hz', '_ in trial ',trial ),
        strcat('Gender:',num2str(Sub.Data.SubjectGender()),' Age:',
        Sub.Data.SubjectAge())});
    ylabel('Amplitude'); xlabel ('f'); xlim([4 21]); grid on;

%% ---------------------------------------------------------------
%% Classification
% 7 flickering patterns with different frequencies are used
% 6 , 6.5 , 7 ,7.5 , 8.2 , 9.3 , 10 Hz
```

```
% corresponding to feature 1, ...., 7
% Cx == feature x , where x ranges from 1 to 7

%% feature 1
% for the first frequency (6Hz), the feature is:
fundFreq1 = 6; %fundamental frequency
harmFreq1 = 2*fundFreq1; % harmonic of the fundamental freq.
fun_index1 =find (f==fundFreq1);
harm_index1 = find (f==harmFreq1);

C1_fund = 0;
for i = fun_index1 - 2: fun_index1 + 2 %the range is fun_index +- 0.1 Hz
    C1_fund = C1_fund + abs(pxxFt(i,:));
end;
C1_harm = 0;
for i = harm_index1 - 2: harm_index1 + 2 %the range is fun_index +- 0.1 Hz
    C1_harm = C1_harm + abs(pxxFt(i,:));
end;
C1 = C1_fund + C1_harm;


%% feature 2
% for the 2nd frequency (6.5Hz), the feature is:
fundFreq2 = 6.5; %fundamental frequency
harmFreq2 = 2*fundFreq2; % harmonic of the fundamental freq.
fun_index2 =find (f==fundFreq2);
harm_index2 = find (f==harmFreq2);

C2_fund = 0;
for i = fun_index2 - 2: fun_index2 + 2 %the range is fun_index +- 0.1 Hz
    C2_fund = C2_fund + abs(pxxFt(i,:));
end;
C2_harm = 0;
for i = harm_index2 - 2: harm_index2 + 2 %the range is fun_index +- 0.1 Hz
    C2_harm = C2_harm + abs(pxxFt(i,:));
end;
C2 = C2_fund + C2_harm;

%% feature 3
% for the 3rd frequency (7Hz), the feature is:
fundFreq3 = 7; %fundamental frequency
harmFreq3 = 2*fundFreq3; % harmonic of the fundamental freq.
fun_index3 =find (f==fundFreq3);
harm_index3 = find (f==harmFreq3);

C3_fund = 0;
for i = fun_index3 - 2: fun_index3 + 2 %the range is fun_index +- 0.1 Hz
    C3_fund = C3_fund + abs(pxxFt(i,:));
end;
C3_harm = 0;
for i = harm_index3 - 2: harm_index3 + 2 %the range is fun_index +- 0.1 Hz
    C3_harm = C3_harm + abs(pxxFt(i,:));
```

```
end;
C3 = C3_fund + C3_harm;

%% feature 4
% for the 4th frequency (7.5Hz), the feature is:
fundFreq4 = 7.5; %fundamental frequency
harmFreq4 = 2*fundFreq4; % harmonic of the fundamental freq.
fun_index4 =find (f==fundFreq4);
harm_index4 = find (f==harmFreq4);

C4_fund = 0;
for i = fun_index4 - 2: fun_index4 + 2 %the range is fun_index +- 0.1 Hz
    C4_fund = C4_fund + abs(pxxFt(i,:));
end;
C4_harm = 0;
for i = harm_index4 - 2: harm_index4 + 2 %the range is fun_index +- 0.1 Hz
    C4_harm = C4_harm + abs(pxxFt(i,:));
end;
C4 = C4_fund + C4_harm;

%% feature 5
% for the 5th frequency (8.2Hz), the feature is:
fundFreq5 = 8.1875; %fundamental frequency
harmFreq5 = 2*fundFreq5; % harmonic of the fundamental freq.
fun_index5 =find (f==fundFreq5);
harm_index5 = find (f==harmFreq5);

C5_fund = 0;
for i = fun_index5 - 2: fun_index5 + 2 %the range is fun_index +- 0.1 Hz
    C5_fund = C5_fund + abs(pxxFt(i,:));
end;
C5_harm = 0;
for i = harm_index5 - 2: harm_index5 + 2 %the range is fun_index +- 0.1 Hz
    C5_harm = C5_harm + abs(pxxFt(i,:));
end;
C5 = C5_fund + C5_harm;

%% feature 6
% for the 6th frequency (9.3Hz), the feature is:
fundFreq6 = 9.3125; %fundamental frequency
harmFreq6 = 2*fundFreq6; % harmonic of the fundamental freq.
fun_index6 =find (f==fundFreq6);
harm_index6 = find (f==harmFreq6);

C6_fund = 0;
for i = fun_index6 - 2: fun_index6 + 2 %the range is fun_index +- 0.1 Hz
    C6_fund = C6_fund + abs(pxxFt(i,:));
end;
C6_harm = 0;
for i = harm_index6 - 2: harm_index6 + 2 %the range is fun_index +- 0.1 Hz
    C6_harm = C6_harm + abs(pxxFt(i,:));
end;
```

```
C6 = C6_fund + C6_harm;

%% feature 7
% for the 7th frequency (10Hz), the feature is:
fundFreq7 = 10; %fundamental frequency
harmFreq7 = 2*fundFreq7; % harmonic of the fundamental freq.
fun_index7 =find (f==fundFreq7);
harm_index7 = find (f==harmFreq7);

C7_fund = 0;
for i = fun_index7 - 2: fun_index7 + 2 %the range is fun_index +- 0.1 Hz
    C7_fund = C7_fund + abs(pxxFt(i,:));
end;
C7_harm = 0;
for i = harm_index7 - 2: harm_index7 + 2 %the range is fun_index +- 0.1 Hz
    C7_harm = C7_harm + abs(pxxFt(i,:));
end;
C7 = C7_fund + C7_harm;

%% Classifier
C =[C1 C2 C3 C4 C5 C6 C7]; %vector of amplitudes
C = C./ max(abs(C)); %normalized vector. Amplitude values from 0 to 1
%Cnorm = norm(C,1) can't do the norm.....
%features = [6 6.5 7 7.5 8.2 9.3 10];%vector of features (from 1 to 7)
features = categorical({'6Hz','6.5Hz','7Hz','7.5Hz','8.2Hz','9.3Hz','10Hz'});
features = reordercats(features,{'6Hz','6.5Hz','7Hz','7.5Hz','8.2Hz','9.3Hz','10Hz'});
figure
    bar (features,C, 'FaceColor',[0 .5 .5],'EdgeColor',[0 .9 .9],'LineWidth',1.5);
    title({strcat('Subject ',subject,'_ looking at frequency ',
        num2str(frequecy),'Hz', '_ in trial ',trial ),
        strcat('Gender:',num2str(Sub.Data.SubjectGender()),' Age:',
        Sub.Data.SubjectAge())});
    xlabel('Frequencies (features)');
    ylabel('Normalized amplitude')

%% autocorrenlation function
function [Rxx]=autom(x)
    % [Rxx]=autom(x)
    % This function Estimates the autocorrelation of the sequence of
    % random variables given in x as: Rxx(1), Rxx(2),,Rxx(N), where N is
    % Number of samples in x.
    N=length(x);
    Rxx=zeros(1,N);
    for m=1: N+1
        for n=1: N-m+1
            Rxx(m)=Rxx(m)+x(n)*x(n+m-1);
        end;
    end;
end
```