A Decoding algorithm for Non-invasive SSVEP-based Drone Flight Control

Abdelhadi Hireche
Department of Electronics
Hassiba Benbouali University of Chlef
Chlef, Algeria
a.hireche@etu.univ-chlef.dz

Yasmine Zennaia
Department of Electronics
Hassiba Benbouali University of Chlef
Chlef, Algeria
y.zennaia@etu.univ-chlef.dz

Redouane Ayad
Department of Electronics
Hassiba Benbouali University of Chlef
Chlef, Algeria
r.ayad@etu.univ-chlef.dz

Abdelkader Nasreddine Belkacem
Department of Computer and Network Engineering
United Arab Emirates University
Abu Dhabi, United Arab Emirates
belkacem@uaeu.ac.ae

2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 978-1-6654-0126-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/BIBM52615.2021.9669629

***Abstract*—Many advanced researches on natural user interfaces methods based on user-centered design have been using speech, gestures and vision to interact with environment and/or control internet of things (IoT) devices. Brain computer interfaces (BCIs) technology could make this interaction/control more natural, faster, and reliable, and effective. In this paper, we propose a decoding algorithm for controlling a drone in a three-dimensional (3D) space using steady state visually evoked potential (SSVEP)-based BCI modality. SSVEP-based BCI has the great potential for use in virtual reality environment, which enables the user to control the drone using his/her brain activity in an first-person-view mode. Therefore, the user will be in a full control over the flight using BCI system by commanding the drone to take off, land, go forward, stop, and turn right/left. This system yields a super convenient way for normal people with no prior experience to interact with the drone and control a flight mission in a little to no time, over traditional manual control which takes longer time to learn and perfect. in the decoding phase, a various convolutional neural networks (CNN) models were built to accommodate different control criteria such as the generality of the model. This proposed EEG-decode-pipeline has been implemented on an open-source data-set which consists of 8-channel EEG data from 10 subjects performing 12 target SSVEP-based BCI task. A high multi-class BCI classification results were achieved with an accuracy ranging around 80-90% for performing a successful online simulation of the drone control.**

***Index Terms*—Brain Computer Interface (BCI), Electroencephalography (EEG), Drone control, Steady-State Visual Evoked Potential (SSVEP).**

I. Introduction

Electrophysiological and neurophysiologic studies have demonstrated increases in neural activity elicited by gazing at a stimulus. Visual evoked potentials are elicited by sudden visual stimuli and the repetitive visual stimuli would lead to stable voltage oscillations pattern in EEG that is called SSVEP (Steady-State Visual Evoked Potential) [1, 2, 3]. SSVEP is considered as a concept with two different definitions. “Ragan” proposed that SSVEP is a direct response in the primary visual cortex[4]. On the other hand, Silberstein et al. [5] assumed that the SSVEP includes indirect cortical responses via cortical-loops, from the peripheral retina, while a cognitive task is performed. Steady-State Visual evoked potential in this model has a complex amplitude and phase topography across the posterior scalp with considerable inter-subject variability. Although the main mechanism of SSVEP still is unknown, generally SSVEP is considered as a continuous visual cortical response evoked by repetitive stimuli with a constant frequency on the central retina.

As a nearly sinusoidal oscillatory waveform, the SSVEP usually contains the same fundamental frequency as the stimulus and some harmonics of the fundamental frequency. For example, when the retina is excited by a visual stimulus at presentation rates ranging from 3.5 Hz to 75 Hz, the brain generates an electrical activity at the same and different frequency of the visual stimulus. The flickering stimulus of different frequency with a constant intensity can evoke the SSVEP in verity of amplitudes, ranging from (5-12Hz) as low frequencies, (12-25 Hz) as medium ones and (25-50 Hz) as high frequency bands [6]. This type of stimulus is a powerful indicator in the diagnosis of visual pathway function, visual imperceptions in patients with cerebral lesions, loss of multi-focal sensitivity in patients with multiple sclerosis, and neurological abnormalities in patients with schizophrenia and other clinical diagnoses.

In addition to the usual clinical purpose of diagnosing visual pathway and brain mapping impairments, the SSVEP can serve as a basis for brain computer interface (BCI) [7, 8, 9]. Recently, SSVEP based-BCI systems have gained a special place in the BCI paradigms continuum because of having a variety of different possibilities.

SSVEP-based brain computer interfaces are useful in different applications, especially the ones that need some major requirements as follows:[10, 11, 12, 13, 14, 15]

- Large number of BCI commands is necessary (in SSVEP BCI limitations are mostly defined only by the design).

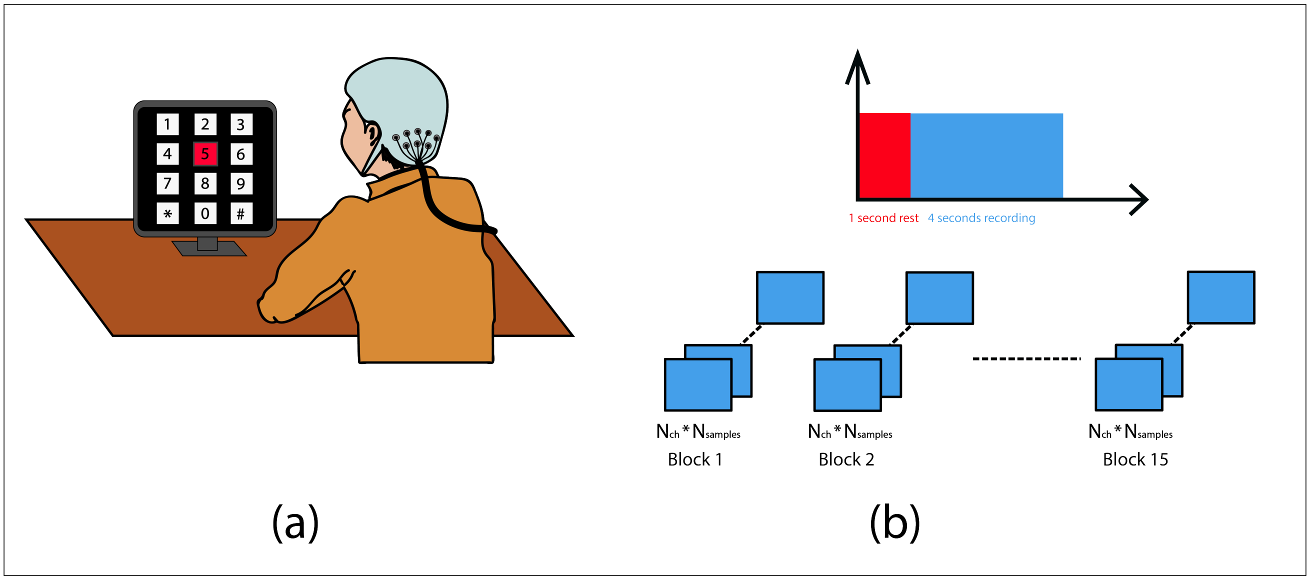

Fig. 1. **Experimental paradigm:** a) brain computer interface hardware set-up. b) data structure of the EEG signals.

- High reliability of recognition is necessary (in SSVEP BCI, patterns are clearly distinguishable by frequency).
- No training (or just short time training for classifier training) is allowed.
- Self-paced performance is required.

A typical SSVEP-based BCI system uses a light-emitting diode (LED) for flickering. SSVEP responses can be measured within narrow frequency bands (e.g. around the visual stimulation frequency.

Several numbers of stimuli can be implemented by using not necessarily a wide range of flickering frequencies, as the minimum detectable difference between frequencies is 0.2 Hz.

The occipital region is the area where this feature is generated more prominently.[16] The most wide-spread signal processing technique to extract the SSVEP responses of the brain from the raw EEG data is based on power spectral density (PSD) using FFT of a sliding data window with a fixed length.[17]

Figure 1 shows the control loop for our BCI system that we intend to implement, it start with a computer screen that shows the flickering stimuli at the specific frequencies of choice.

Each frequency stimulus maps to a specific command for drone control, the user gazes at the intended stimulus which he/she wants eventually to be transformed into the specific command, the brain activity changes accordingly. the EEG channels at the occipital area captures these changes and sends it to our computing device (computer in this case), it gets filtered first to a specific bandpass.

The filtered signal gets passed to a function that computes the feature, in this case we used the complex spectrum feature as the input to our SSVEP classifier. This classifier which is built and via convolutional neural network (CNN) algorithm makes a probability calculation and figures out the class with the highest probability to be the resulted choice of prediction.

Finally, the predicted class gets translated to the desired output command, this whole process repeats over and over, each second a new command is updated and get sent to the drone.

II. Methods

A. Pipeline of Work

To implement our SSVEP based-BCI we followed a pipeline of work that is consisted of three main phases, we began the first phase by searching for, selecting and exploring a particuler public SSVEP dataset, after that in the second phase we'll do the actual heavy lefting work where we will be building and training the model according to different changing scenarios, and testing the performance in terms of accuracy prediction in an offline mode and response speed in a simulated online environement. Finally, in the last phase, we'll be evaluating the results and having some disscusion about it.

B. Brief about the Dataset

We chose an offline SSVEP dataset that it's publicly available in a github repository[18], in order to build our classifier and test it, our selection criteria was based on several facts that aligns closely with our goals. For controlling the quadcopter in a three dimensional space we need at least to do nine distinct commands, which are takeoff, land, move forward, move backward, turn right, turn left, go up, go down and stop, another fact was that the need to test the classifier across different subjects to see how the classifier performs both dependent and independent from each subject, so we set the selection criteria to that the dataset needs at least to be taken on 6 subjects and performed on at least 9 classes.

this dataset was used originally in this study[19], data were collected on ten healthy subjects (9 males and 1 females, mean

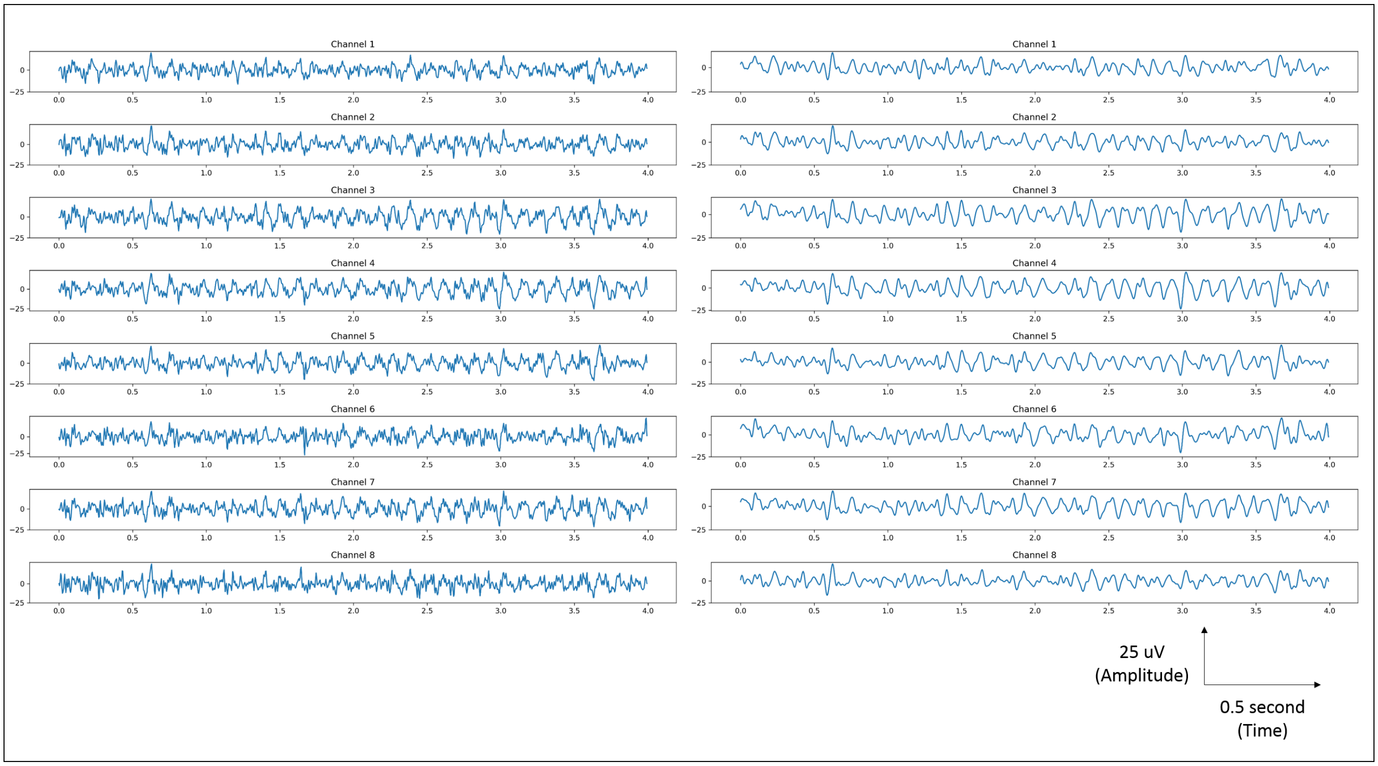

Fig. 2. Raw and filtered EEG signals

age: 28 years old), twelve flickering stimuli were displayed on a screen a 27-inch LCD monitor (ASUS VG278) with a refresh rate of 60Hz and a resolution of 1280×800 pixels.

The BioSemi ActiveTwo EEG (Biosemi B.V., Netherlands) system with a sampling rate of 2048 Hz was used to acquire the EEG data. Eight active electrodes were placed over the occipito-parietal area.

At the start of each trial, the subject was directed by a red square cue to gaze at a specific stimulus. This cuing period was 1 second. A stimulation period of 4 seconds was followed by the cue period and the subjects were asked to focus on the targeted stimulus for the entire duration with instruction to not move or eye blink to avoid artifacts presence in the signal.

One block consisted of 12 trials with one trial for each of the 12 stimuli on the screen presented in random order. A total of 15 blocks were presented leading to a total of 180 trials, data was saved in a .mat file, each file has a four-way tensor electroencephalogram (EEG) data for each subject and it's represented as the following multi-dimensional array: $[N.\,targets,\ N.\,channels,\ N.\,sampling\ points,\ N.\,trials]$ with:

- Number of targets : 12
- Number of channels : 8
- Number of sampling points : 1114
- Number of trials : 15
- Sampling rate (Hz) : 256

C. Preprocessing

All active eight channels were used from this dataset, Temporal filtering was performed using a 4th order Butterworth band-pass filter between 6 Hz and 80 Hz. Additionally, each 4 second trial was divided into 1 second non-overlapping segments to get more epochs.

D. Feature Extraction

Magnitude spectrum features computed using fast fourier transform (FFT) always considered to be one of the fundamental and old ways to extract features in SSVEP based-BCIs.

To get magnitude spectrum features, the prepossessed time-domain EEG segments $x(n)$ are transformed into the frequency domain $X(k)$ using FFT. This resulted in a sequence of complex numbers $Re(X(k)) + j.Im(X(k))$ from which the magnitude spectrum will be calculated: $|X(k)| = \sqrt{Re(X(k))^2 + Im(X(k))^2}$.

Figure shows frequency response computed using fast-fourier transform for subject 8 in the first 3 frequencies stimuli. The flicker frequencies can be readily identified from the magnitude response of the signal.

Both magnitude and phase related information are extracted from the signal, first step was to transform the input time-domain signal into the complex FFT representation using the standard FFT, with a resolution setting of 0.293 Hz, then the frequency components of the real part and imaginary part across each channel were taken between 3 Hz and 35 Hz resulting in two vectors of length 110 each, These two

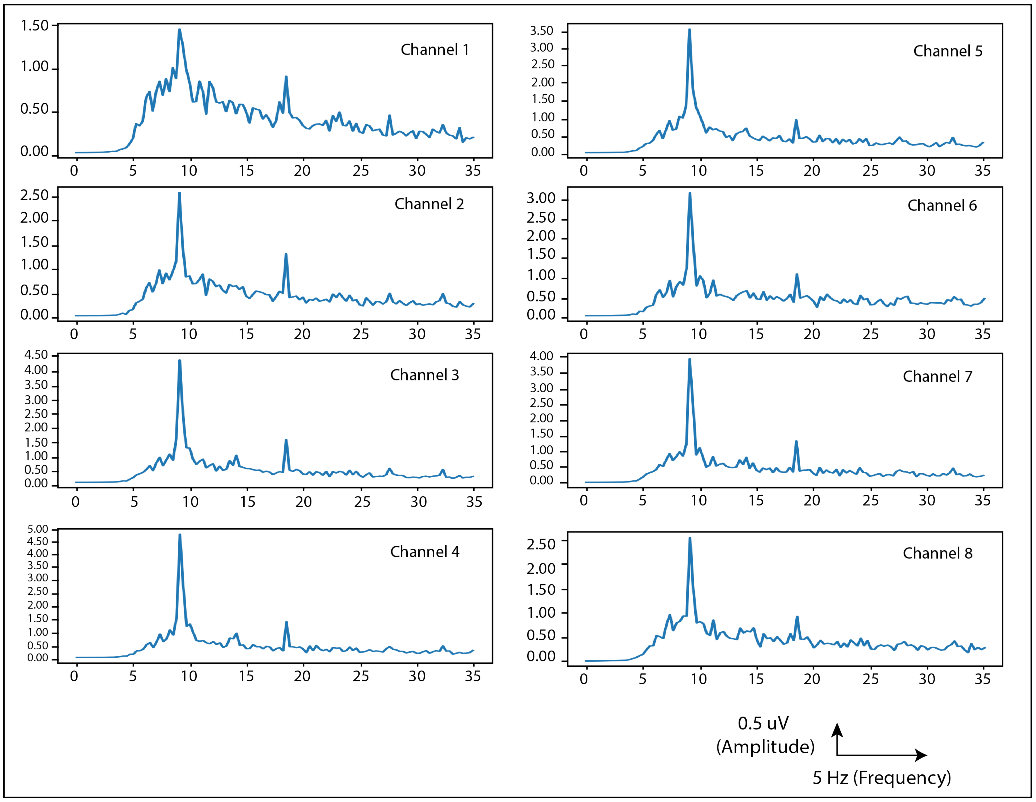

Fig. 3. Frequency spectral density results using fast Fourier transform

vectors were concatenated into a single feature vector as pairs $Re(X)||Im(X)$, where the first half contained the real part and the second half contained the imaginary part of the complex FFT resulting in a vector of length 220. The resultant feature vector was stacked together to form a matrix which gonna be considered the input to our CNN algorithm, with dimensions $N_{ch} \times N_{fc}$ where $N_{fc} = 220$ is approach of using the complex FFT as input to the CNN is referred to as the C-CNN method.

E. Classification

The network architecture is inspired by the work from these papers [20, 21].

The CNN consists of four main layers, an input layer, two convolutional layers, and a fully connected output layer. The features extracted in the previous preprocessing step were provided as input to the CNN. The input layer of the CNN had dimensions $N_{ch} \times N_{fc}$, This was followed by the convolutional layer Conv-1, designed based on the spatial filtering concept. The kernel dimensions of $N_{ch} \times 1$ were used for this layer and performed 1 dimensional convolutions across the channel dimension (N_{ch}) of the input. The objective for this layer was to learn representations as a result of applying different weights to each channel in the input. The number of feature maps in the Conv-1 layer was $2 \times N_{ch}$ with dimensions $1 \times N_{fc}$.

The Conv-2 layer was designed to operate on the spectral representation dimension (N_{fc}) of the previous layer. The kernel dimension for this layer was 1×10. The number of feature maps in this layer were $2 \times N_{ch}$. As a result of the convolution, the feature maps in this layer had the dimensions equal to $1 \times (N_{fc} - 10 + 1)$.

Batch normalization was performed on the outputs of layers Conv-1 and Conv-2. Batch normalization has been shown to reduce the internal covariance within input samples resulting in the samples having zero mean and unit variance.

The rectified linear unit (ReLU) was used as the activation function. to prevent overfitting. Dropout was added to the network as a regularization technique. Dropout and batch-normalization have been shown to improve the generalization performance and training speed of neural networks.

The number of units in the output layer (K) is equal to the number of SSVEP classes in the input data. The output layer was equipped with the softmax function to output the probability that a given input segment belonged to a particular class.

F. Offline Training and Testing Procedure

1) User-Dependent Training: A user-dependent model was developed as follows: data from a single participant was used to train a model and was validated using the same participant's data. To achieve this, a 10-fold cross-validation procedure was performed on 1 second segments of the epochs. The total number of 1 s segments in the training fold were 648 and 72 in the testing fold, since the method is UD (User-Dependent), it's referred to as UD-C-CNN.

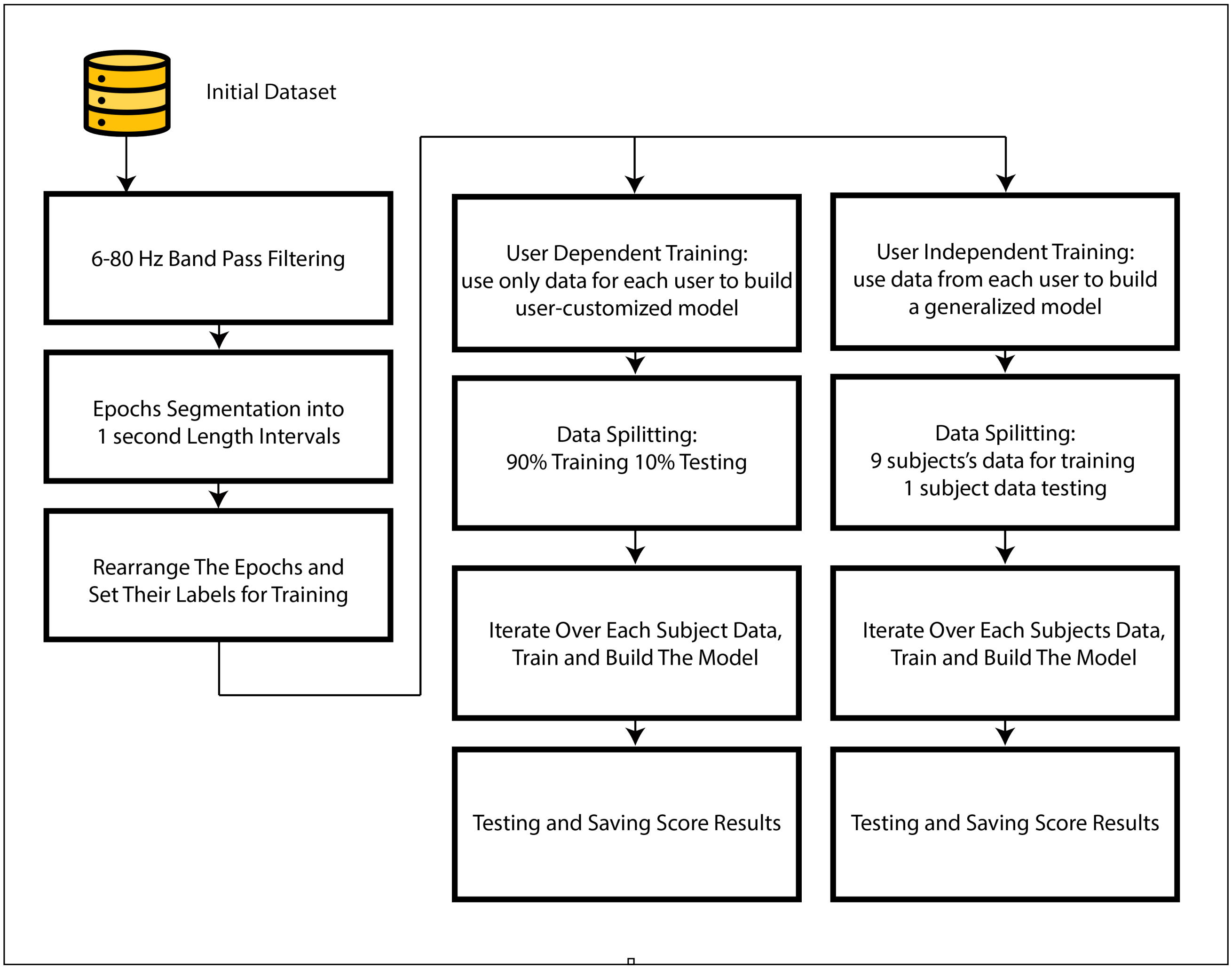

Fig. 4. Pipeline of the proposed decoding algorithm.

2) User-Independent Training: A user-dependent model was developed as follows: a leave-subject-out-cross-validation is used, so if the given dataset contains N subjects, then the training is carried out on the data of the N-1 subjects and tested on the data of the left subject, the UI training leads to the development of a calibration free system where the user can directly use the system without getting the algorithm to see and learn from his data, The total number of 1 second segments of epochs in the training fold were 6480 and 720 in the testing fold, since the method is UD (User-Dependent), it's referred to as UI-C-CNN.

G. Online Simulation Procedure

For the online model prediction testing and drone simulation we used $sjtu\_drone$ [22] (sjtu refers to the Shanghai Jiao Tong University), which is a quadcopter simulation program for Parrot AR. Drone, forked from $tum\_simulator$ [23], it's developed with ROS and Gazebo, It is used for testing visual SLAM algorithms aiding with different sensors, such as IMU, sonar range finder and laser range finder. And also for testing algorithms for UAV contests in SJTU.

We used a UI-C-CNN model built from subject 6 data, as a representative data, because its high decoding accuracy to avoid miss-classification during online simulation phase. We have picked 9 classes which maps to the 9 commands of the quadcopter, as it is shown in figure 5.

We regard the testing fold of subject 6 data to be the input to the mode, each second an epoch of 1 second is sent, which means a command is updated and sent to the drone each second, we picked the epochs in such a way that drone moving path will be in a kind of a square shape, where the drone first take-off, it keep going forward for 2 seconds then turns left, it repeats this for 4 times, then it stops for 2 seconds, then land.

III. Results and Discussion

A. Offline Test Results

The following table shows the results from the user-dependent and user-independent training scenarios using com-

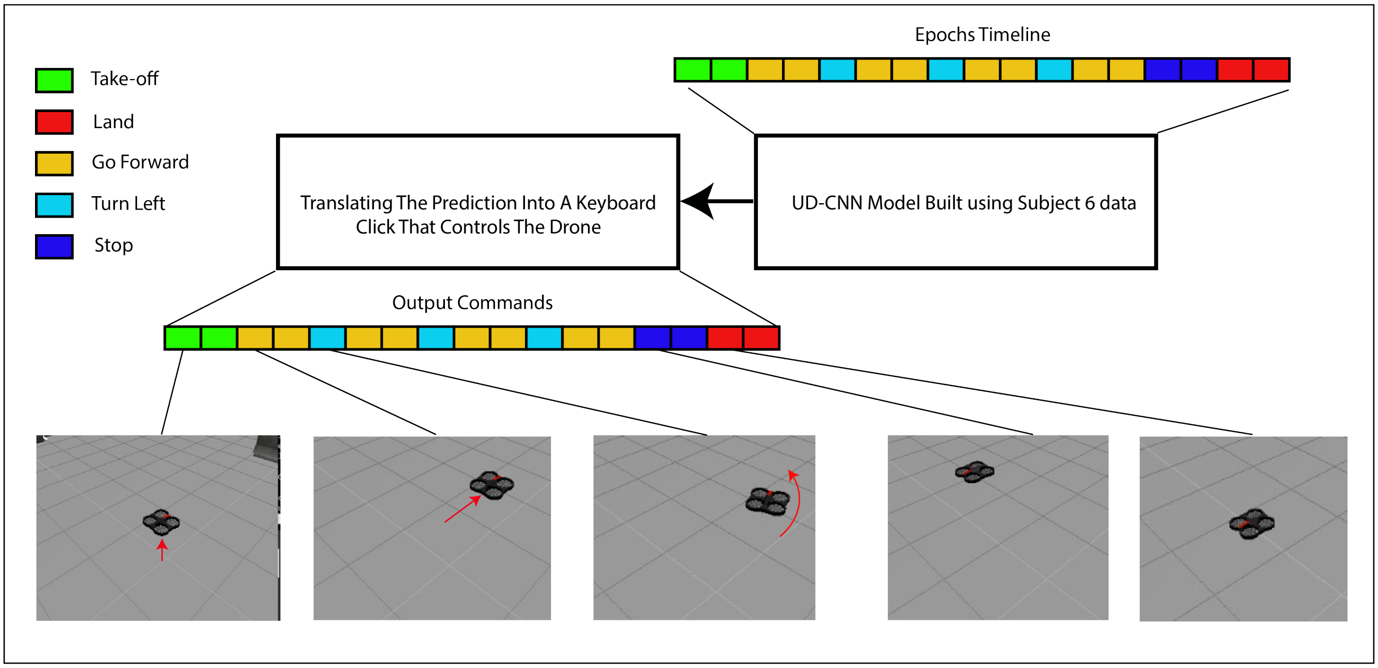

Fig. 5. Online simulation diagram.

plex spectrum features and CNN, along with the results from the combination method proposed on the original paper of the dataset. [19]

	UD-C-CNN	UI-C-CNN	CM
Subject 1	76.53	58.47	78.89
Subject 2	55.97	31.94	71.67
Subject 3	94.86	78.33	94.44
Subject 4	98.06	93.89	99.44
Subject 5	99.86	90.14	100.00
Subject 6	99.72	92.78	99.44
Subject 7	94.44	85.28	98.33
Subject 8	99.31	97.50	100.00
Subject 9	97.92	94.31	98.89
Subject 10	90.28	84.58	86.67
Avg/STD	$90.69 \pm 13.36\%$	$80.72 \pm 19.46\%$	$92.78 \pm 10.22\%$

TABLE I
TEST SCORES.

Individual classification accuracy for each subject scores are fairly close between UD-C-CNN and the combination method, with some exceptions, as the case for subject 2 where the combination outperformed the UD-C-CNN with a fairly tangible percentage, and for subject 10 where UD-C-CNN outperformed the combination method with a significant percentage.

On the other hand UI-C-CNN was good, but still performed poorly on some subjects, especially with subject 2 where it yields 31.94%, and also on subject 1 with 58.47%, nevertheless it did very good and competed well with the other methods on the remaining subjects with an accuracy score ranging from 78.33% to 94.31%.

The overall average accuracies of both classification methods across all 10 subjects compared with the original combination method proposed on the original paper, Among CNN training scenarios, UD method achieved the highest accuracy with $90.69 \pm 13.36\%$, compared to the combination method which achieved $92.78 \pm 10.22\%$ it did very well and in general it was very close to it with a difference below 2%.

UI-C-CNN which achieved $80.72 \pm 19.46\%$ also did well but UD-C-CNN and Combination Method outperformed it with a significant difference.

B. *Online Simulation Test Results*

The online drone simulation worked pretty well, and the model was able to respond instantaneously and with accuracy after receiving the input. Fig.**??** shows scenes captured through all the phases of the simulation, where each a second a new epoch of data is received and get processed and classified, then translated into an output command to the drone which responds back with a particular action movement.

IV. CONCLUSION

The accuracy of prediction is great in user-dependant scenario across all subjects, which signify how much important is a costume trained classifier, because everyone is different, each one of us a slight difference in brain structure which can effect brain activity significantly.

User-independent classifier on the hand did well across most of the subjects, with exceptions for subject 1 and 2, which back up the the first argument mentioned in the previous paragraph which state the our brain still has much difference though they're look alike, but still the user-independent classifier did pretty well, and the classification could be generalized across many different subjects in most cases, which allows for a direct plug and play mode with the system, with no prior calibration to each new user.

We achieved our main goal from this work, and that is to to control a quadcopter in a three dimensional space using an SSVEP-based BCI, which surely will serve as a great tool to interface and control the drone even for people with little to no experience on how manual drone control works.

Future work and development may consist developing a virtual reality application towards the use of our BCI System in a potential realistic world situation, exploring other BCI modalities and figures new BCI system designs like hybrid BCIs that could achieve more and ease the user-experience to make it more efficient and reliable.

Acknowledgment

This work was supported by the United Arab Emirates University (UAEU grant no G00003270 “31T130”).

References

[1] Ramon A. Suárez Fernández et al. “Natural user interfaces for human-drone multi-modal interaction”. In: *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2016, pp. 1013–1022. DOI: 10.1109/ICUAS.2016.7502665.

[2] Meng Wang et al. “A Wearable SSVEP-Based BCI System for Quadcopter Control Using Head-Mounted Device”. In: *IEEE Access* PP (Apr. 2018), pp. 1–1. DOI: 10.1109/ACCESS.2018.2825378.

[3] Jaehoon Choi and Sungho Jo. “Application of Hybrid Brain-Computer Interface with Augmented Reality on Quadcopter Control”. In: *2020 8th International Winter Conference on Brain-Computer Interface (BCI)*. 2020, pp. 1–5. DOI: 10.1109/BCI48061.2020.9061659.

[4] David Regan. “Steady-state evoked potentials”. In: *J. Opt. Soc. Am.* 67.11 (Nov. 1977), pp. 1475–1489. DOI: 10.1364/JOSA.67.001475. URL: http://www.osapublishing.org/abstract.cfm?URI=josa-67-11-1475.

[5] Richard B Silberstein et al. “Steady state visually evoked potential (SSVEP) topography in a graded working memory task”. In: *International Journal of Psychophysiology* 42.2 (2001), pp. 219–232. ISSN: 0167-8760. DOI: https://doi.org/10.1016/S0167-8760(01)00167-2. URL: https://www.sciencedirect.com/science/article/pii/S0167876001001672.

[6] Zhenghua Wu et al. “Stimulator selection in SSVEP-based BCI”. In: *Medical Engineering Physics* 30.8 (2008). Special Issue (part): Bioengineering in Taiwan, pp. 1079–1088. ISSN: 1350-4533. DOI: https://doi.org/10.1016/j.medengphy.2008.01.004. URL: https://www.sciencedirect.com/science/article/pii/S1350453308000234.

[7] Abdelkader Nasreddine Belkacem et al. “Real-time control of a video game using eye movements and two temporal EEG sensors”. In: *Computational intelligence and neuroscience* 2015 (2015), p. 1.

[8] Nuraini Jamil et al. “Cognitive and Affective Brain–Computer Interfaces for Improving Learning Strategies and Enhancing Student Capabilities: A Systematic Literature Review”. In: *IEEE Access* (2021).

[9] Nuraini Jamil et al. “Noninvasive Electroencephalography Equipment for Assistive, Adaptive, and Rehabilitative Brain–Computer Interfaces: A Systematic Literature Review”. In: *Sensors* 21.14 (2021), p. 4754.

[10] Xiaorong Gao et al. “A BCI-based environmental controller for the motion-disabled”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11.2 (2003), pp. 137–140. DOI: 10.1109/TNSRE.2003.814449.

[11] Abdelkader Nasreddine Belkacem and Abderrahmane Lakas. “A Cooperative EEG-based BCI Control System for Robot–Drone Interaction”. In: *2021 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE. 2021, pp. 297–302.

[12] Fatima Ali Al-Nuaimi et al. “Mind drone chasing using EEG-based Brain Computer Interface”. In: *2020 16th International Conference on Intelligent Environments (IE)*. IEEE. 2020, pp. 74–79.

[13] Abdelkader Nasreddine Belkacem et al. “Brain computer interfaces for improving the quality of life of older adults and elderly patients”. In: *Frontiers in Neuroscience* 14 (2020), p. 692.

[14] Chao Chen et al. “Quadcopter robot control based on hybrid brain–computer interface system”. In: *Sensors and Materials* 32.3 (2020), pp. 991–1004.

[15] L Shao et al. *EEG-Controlled Wall-Crawling Cleaning Robot Using SSVEP-Based Brain-Computer Interface. J Healthc Eng 2020*. 2020.

[16] Christian Breitwieser, Christoph Pokorny, and Gernot R Müller-Putz. “A hybrid three-class brain–computer interface system utilizing SSSEPs and transient ERPs”. In: *Journal of Neural Engineering* 13.6 (Oct. 2016), p. 066015. DOI: 10.1088/1741-2560/13/6/066015. URL: https://doi.org/10.1088/1741-2560/13/6/066015.

[17] Setare Amiri et al. “A Review of P300, SSVEP, and Hybrid P300/SSVEP Brain- Computer Interface Systems”. In: *Brain-Computer Interface Systems*. Ed. by Reza Fazel-Rezai. Rijeka: IntechOpen, 2013. Chap. 10. DOI: 10.5772/56135. URL: https://doi.org/10.5772/56135.

[18] Nakanishi et al. *12-class joint frequency-phase modulated steady-state visual evoked potentials (SSVEPs)*. 2015. URL: https://github.com/mnakanishi/12JFPM_SSVEP.

[19] Masaki Nakanishi et al. “A Comparison Study of Canonical Correlation Analysis Based Methods for Detecting Steady-State Visual Evoked Potentials”. In: *PLOS ONE* 10.10 (Oct. 2015), pp. 1–18. DOI: 10.1371/

journal.pone.0140703. URL: https://doi.org/10.1371/journal.pone.0140703.

[20] Richard B Silberstein et al. “Steady state visually evoked potential (SSVEP) topography in a graded working memory task”. In: *International journal of psychophysiology* 42.2 (2001), pp. 219–232.

[21] Aravind Ravi. “Enhancing the Decoding Performance of Steady-State Visual Evoked Potentials based Brain-Computer Interface”. In: (2019). URL: https://uwspace.uwaterloo.ca/handle/10012/14881.

[22] *sjtu_drone - A quadrotor simulation program based gazebo+ROS, which is forked from the tum$_s$$imulator$..* 2015. URL: https://bitbucket.org/dannis/sjtu_drone/src/master/.

[23] Juergen Sturm Hongrong Huang. *tum_simulator - The implementation of a gazebo simulator for the Ardrone 2.0.* 2015. URL: http://wiki.ros.org/tum_simulator.