ResearchGate

Experience designing a BCI for flying a drone based on the use of Empirical Mode Decomposition

Technical Report · November 2016

DOI: 10.13140/RG.2.2.16858.13761

3 authors:

Sumit Kumar Ram
Indian Institute of Science Education and Res…

Marta Molinas
Norwegian University of Science and Techno…

Geir Kulia
Norwegian University of Science and Technol…

Sumit Kumar Ram

Experience designing a BCI for flying a drone based on the use of Empirical Mode Decomposition

Project report
Supervisor: Prof. Marta Molinas

Trondheim, November 2016

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Innovation and Creativity

NTNU

Norwegian University of Science and Technology

Project report

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Engineering Cybernetics

ISBN N/A (printed version)
ISBN N/A (electronic version)
ISSN 1503-8181

Project report at NTNU, 2016:N/A

Contents

List of Tables

List of Figures

List of Symbols

Acknowledgements

This is a research project done under the supervision of Prof. Marta Molinas, Department of Engineering Cybernetics, Norwegian University of Science and Technology(NTNU).

It is with immense gratitude that I acknowledge my Professor for her constant support and guidance, without which this project could not have been realised.

I am indebted to my colleagues Geir Kulia and Atle Rygg for their constant support and help in many aspects of the project through out.

This project would not have been possible unless the financial support from the Department of Engineering Cybernetics as Jens Balchen Scholarship. I would like to extend my gratitude to Department of Engineering Cybernetics and late Jens Glad Balchen.

And lastly, I also wish to acknowledge the hospitality and warm reception of all the staffs and faculty members at Norwegian University of Science and Technology, for their guidance and encouragement.They have been really supportive during my stay at the Institute.

Abstract

Abstract

A brain computer interface (BCI) or brain machine interface (BMI), is a direct communication pathway between the brain and an external device. This report describes a Brain Computer Interface(BCI) system based on the EEG recordings of the Emotiv EPOC headset and the post recording procedures with python in Linux environment. The BCI system processes the F8 electrode recordings of the EEG device and applies an eye blink search algorithm to classify the eye blink events in real time. The algorithm takes the amplitude of F8 electrode recording as the feature and classifies eye blink events using Empirical Mode Decomposition, normalizing function and a cut off amplitude level. Every time the system detects two eye blink in a time interval of two seconds it gives take of/land command to a drone which is connected to the BCI system via WiFi and each time it detects one eye blink it gives commands for moving forward and backward.

A set of experiments were conducted and results were noted down to verify the accuracy of the system with the subjects covering a wide age group and an accuracy rate of 84% was noted for the system.

Chapter 1

An introduction to BCI

1.1 Introduction

Brain Computer Interface (BCI) is a relatively new field of research which consists of signal acquisition from brain, analysis, and translation into commands which can be related to output devices to carry out a prespecified desired action. BCI design and development involves several disciplines and its applications ranges from computer games, to device and robot actuation to clinical diagnosis of patients and clinical research. BCI creates a new non-muscular channel for relaying a person's intentions to external devices such as computers, speech synthesizers, assistive appliances, and neural prostheses(1), which is particularly attractive for individuals with severe motor disabilities. Such an interface would improve their quality of life and would, at the same time, reduce the cost of intensive care. In other way a BCI can be considered as an artificial intelligence system that can recognize a certain set of patterns in brain signals following five consecutive stages: signal acquisition, preprocessing or signal enhancement, feature extraction, classification, and the control interface(2).

In terms of signal acquisition, the BCI systems are classified into invasive (intracranial) and noninvasive. Noninvasive systems primarily exploit EEGs to control external devices. The techniques in developing such systems have been under development recently due to their hazard less nature and flexibility. Due to the invasive nature, BCI technology has traditionally been unattractive for serious scientific investigation. However due to development of non-invasive techniques it has acquired significant attraction in scientific community as well as general audience. Even though the technique has the advantage of not exposing the patient to the risks of brain surgery, EEG-based techniques provide limited information mainly because of the existence of system and physiological noise and the interfering undesired signals and artifacts. However, despite these shortcomings, EEG-based methods can detect modulations of brain activity that correlate with visual stimuli, gaze angle, voluntary intentions, and cognitive states (3) .These advantages led to development of several classes of EEG-based systems and has attracted a significant attention towards developing the field.

1.2 Recent developments

Even though the early works in this field started in the beginning of the seventies by JJ Vidal(4) using EEG which was first used by H. Berger(5) for recording human brain activity, its use in research has not become wide spread due to cost and the limitations of the embedded mathematical tools to analyze the obtained signals. More recently there has been significant improvement and increased interest in BCI research due to development of affordable EEG recording devices like Emotiv EPOC which are today massively used in commercial as well as research sectors. Due to these developments and increased attention from various other fields the traditional BCI research has undergone a radical change over the last two decades (6). BCI research, which was confined to only three groups 20 years ago and only six to eight groups 10 years ago, is now a flourishing field with more than 100 active research groups all over the World studying the topic (3). The number of articles published regarding neural interface technology has increased exponentially over the past decade (3). Some of the recent advantages in this fields towards developing a BCI system using spontaneous EEG activity are Linear classification of imagined hand movements(7), A spelling device by Birbaumer et al. (9) , Communication between man and sorroundings (10)EEG-based communication(11); A new communication device for handicapped persons (12); EEG-based communication: prospects and problems (13); EEG-based brain-computer interface for cursor control (14).

Generation of the EEG activity used to drive BCI systems is usually achieved, at least initially, by the subject performing certain cognitive tasks. Much BCI research has involved the development of powerful signal processing techniques (e.g. Spatial filter (15); Bayesian neural networks (16); pattern recognition approach (17)).

Even though there have been notable development in the analysis techniques for the professional EEG devices However, things changes while dealing with the affordable commercial headset like EPOC. The problem with these is the low signal to noise ratios and non professional recording techniques in outdoor conditions(6), which has prompted the implementation of new algorithms to extract the relevant information and discard the noise from the EEG recordings.

Chapter 2

An introduction to EEG

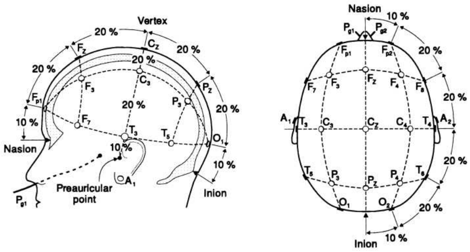
2.1 A brief history of EEG

The presence of electrical activity in the brain was discovered by Richard Caton, who was an English physician, in 1875(8). His findings were about electrical phenomena on rabbits' and monkeys' brains. Hans Berger, a German neurologist, started to study on electrical activity of human brain. In 1924, he used his ordinary radio equipment to amplify the brain's electrical activity so that he could record it on graph paper(5). He realized that rhythmic changes in brain waves varied with the individual's state of consciousness. In other words, he made the first EEG device. Thus, he is known as the inventor of EEG. His findings about the first human electroencephalogram were published in 1929(5).

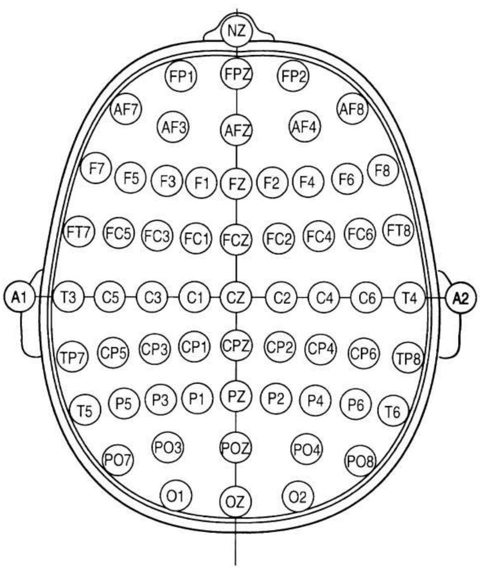
2.2 Generation of EEG signals

The electrical signals, in other words potentials, generated by the brain's neural activity can be observed at the scalp by using suitable amplification methods. The measured signal is called EEG which stands for electroencephalogram. It is used to examine global brain function of the person whose brain signals are recorded. However, brain function related to the performance of specific cognitive tasks is not evaluated using this method. Therefore, EEG serves to provide initial information about global brain condition(18).

Electroencephalography is a medical imaging technique that reads scalp electrical activity generated by brain structures. The electroencephalogram (EEG) is defined as electrical activity of an alternating type recorded from the scalp surface after being picked up by metal electrodes and conductive media (22). Thus electroencephalographic reading is a completely non-invasive procedure that can be applied



(a) Left and above the head



(b) Location and nomenclature of electrodes

Figure 2.1: The international 10-20 system seen from (a) left and above the head. A = Ear lobe, C = central, Pg = nasopharyngeal, P = parietal, F = frontal, Fp = frontal polar, O = occipital. (b) Location and nomenclature of electrodes

repeatedly to patients, normal adults, and children with virtually no risk or limitation. When brain cells (neurons) are activated, local current flows are produced.

EEG measures mostly the currents that flow during synaptic excitations of the dendrites of many pyramidal neurons in the cerebral cortex. Differences of electrical potentials are caused by summed post synaptic graded potentials from pyramidal cells that create electrical dipoles between soma (body of neuron) and apical dendrites (neural branches). Brain electrical current consists mostly of Na+, K+, Ca++, and Cl- ions that are pumped through channels in neuron membranes in the direction governed by membrane potential (22). The detailed microscopic picture is more sophisticated, including different types of synapses involving variety of neurotransmitters. Only large populations of active neurons can generate electrical activity recordable on the head surface.

Between electrode and neuronal layers current penetrates through skin, skull and several other layers. Weak electrical signals detected by the scalp electrodes are massively amplified, and then displayed on paper or stored to computer memory(23). Due to capability to reflect both the normal and abnormal electrical activity of the brain, EEG has been found to be a very powerful tool in the field of neurology and clinical neurophysiology.

From the anatomical point of view, the brain can be divided into three sections: cerebrum, cerebellum, and brain stem. The cerebrum consists of left and right hemisphere with highly convoluted surface layer called cerebral cortex. The cerebrum obtains centres for movement initiation, conscious awareness of sensation, complex analysis, and expression of emotions and behaviour. The cerebellum coordinates voluntary movements of muscles and balance maintaining. The brain stem controls respiration, heart regulation, biorhythms, neuro hormone and hormone secretion, etc.(21). The highest influence to EEG comes from electric activity of cerebral cortex due to its surface position.

2.3 Advancements in inexpensive EEG devices

Even though EEG technology has been being used traditionally mainly for medical purposes, recent advancements in the field of brain computer interface has attracted the use of EEG technology as a recording tool for brain activity. Motivated by this the technology has gained boost from many other aspects of commercialisation and hence more and more low end EEG devices are being available from the manufacturers for BCI systems. Emotiv EPOC is one of such low end EEG instrument which has gained popularity in research community as a recording tool for the BCI system. For our BCI system we are using 16 channel Emotiv EPOC device. As Fig.(2.4) shows it is a easy use and easy wearable device which is very

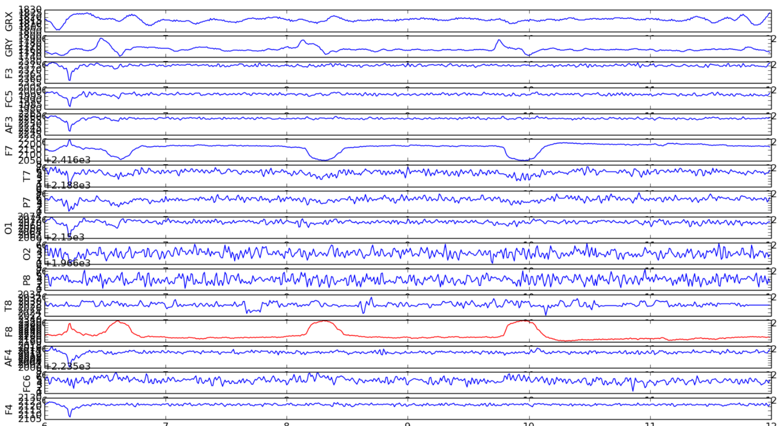


Figure 2.2: A sample EEG recording collected from all channels of the Emotiv EPOC headset

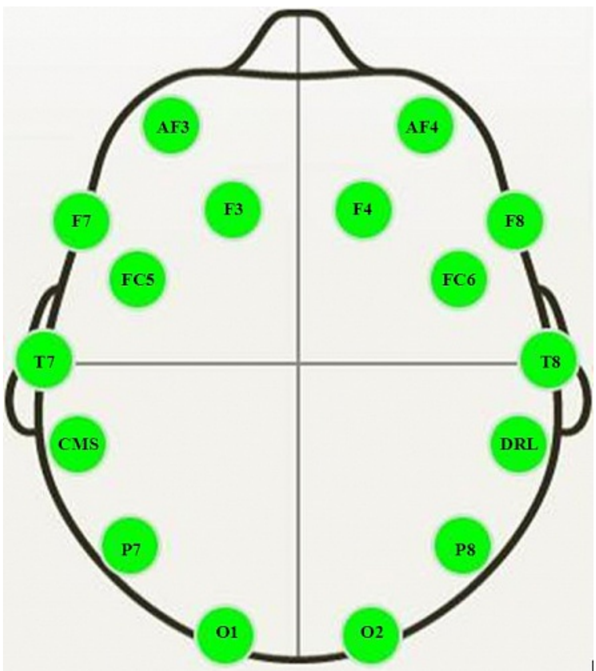


Figure 2.3: The electrode positions for the Emotiv EPOC Headset

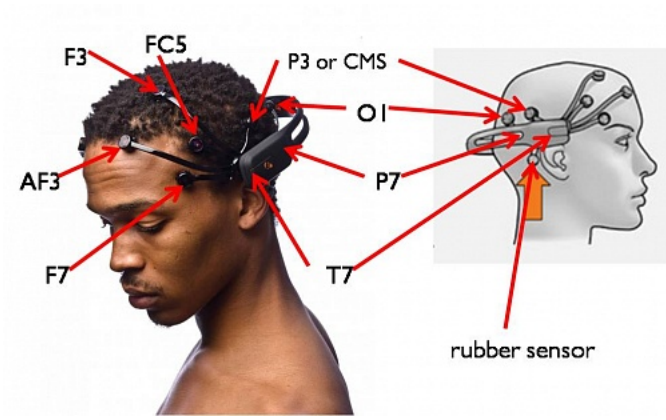


Figure 2.4: Showing the easy handling of the EEG device

convenient to use for building BCI systems. Fig.(2.3) represents the electrode positions of the device according to international 10-20 system. As the electrodes cover most of the active area of the brain it is vary convenient to build simple BCI systems with this device. Fig.(2.2) shows the EEG recordings collected from all channels of the emotiv EPOC system. The recording from F8 electrode is marked red as this is the signal of interest for our BCI system.

Chapter 3

Eye blinks in EEG

3.1 A simple model describing eye blink EEG signals

The blinking of eye and movement of eye is mainly observed in frontal electrode recordings of the EEG device how ever its effects extends to the posterior electrodes. The potential change in the EEG recordings due to eye movement can be understood using a simple model(19) in which we can consider the cornea of the eye as a positive charge that either moves towards or away from the recording electrodes. Hence the electrodes where most prominent effect of eye movement can be observed during vertical eye movements are Fp1 and Fp2 because they are placed directly above the eyes, similarly in horizontal direction F7 and F8 electrodes records the largest potential change because of their lateral position. In a typical upward vertical eye movement(eg, eye closure, eye blink) will produce a downward deflection in Fp1-F3 and Fp1-F8 because the positively charged cornea is moving towards Fp1 making it increasingly more positive. If the eye move to the left, in a lateral direction. Therefore the pen will deflect up in Fp1-F7 and down in F7-T3. The opposite will occur in Fp1-F8 and F8-T4.

Eye movement artifacts have long been believed to be due to movement of the eye which carries a steady electrical charge, the cornea being about 100mV positive with respect to the retina. However, under experimental conditions movement of the corneo-retinal dipole is not necessary to produce blink artifacts(20): movement of the eyelid across the eyeball is sufficient. Moreover, some eye movement artifact can be recorded even after removal of the eye, including the cornea and retina, suggesting that movement of residual deep in orbit can cause artifacts(20).

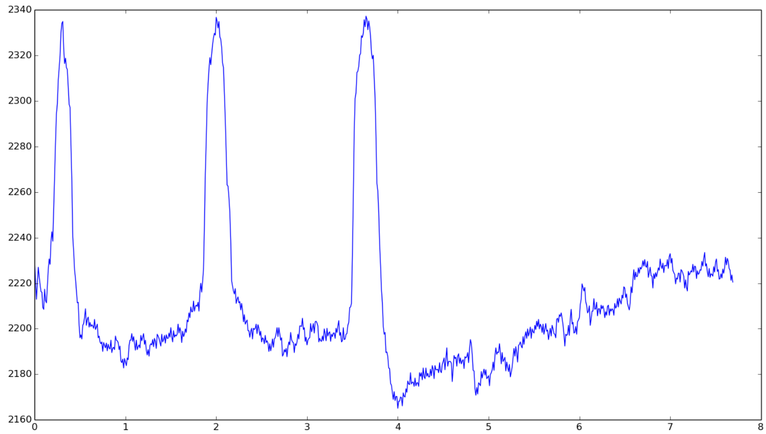


Figure 3.1: Showing the Eye blink EEG signals recorded from F8 channel of Emotiv EPOC Headset.

3.2 Shape of eye blink artifacts

Eye blink artifacts can be easily distinguished from other signals by their frontal distribution, their bilateral symmetry and their characteristic shape. Fig.(3.1) shows three sample eye blink EEG signals from the emotive EPOC headset. As it is clear from the figure that the eye blink artifacts can be easily distinguished from the other EEG signals due to its higher potential value which can be described using the dipole model for cornea movement which is described in the previous section. For our BCI system we are particularly interested in extracting these high amplitude events from other EEG activities.

Chapter 4

Empirical Mode Decomposition

4.1 Nonlinear and Non-stationary time series

Before going into the concept of non stationary time series let's have a look at the concept of stationary time series first. Stationary signal is a stochastic process for which the parameters such as the mean and variance, if they are present, do not change over time and do not follow any trends. In other words its joint probability distribution does not vary over time when sifted in time(26). This means if one looks at a stationary signal for a few moments and then wait an hour and looks at it again, it would look essentially the same. Mathematically a time series $X(t)$ can be considered as a stationary time series if it satisfies the following criteria:

$$E(|X(t)^2|) < \infty \tag{4.1}$$

$$E(X(t)) = m \tag{4.2}$$

$$C(X(t_1), X(t_2)) = C(X(t_1 + \tau), X(t_2 + \tau)) = C(t_1 - t_2) \tag{4.3}$$

Where the symbols have their usual meanings. But in reality most of the time series we encounter are non linear and non stationary in nature(25) (24). There are many analysis techniques, which includes the wavelet analysis , Heisenberg wavelet analysis, the empirical orthogonal function expansion etc, developed over time to deal with these kinds of time series but most of them fail to fulfil the adaptivity criteria. Hilbert Huang Transformation (24) proposed by Huang et al. serves as a major break through in this field due to its adaptive nature. Empirical Mode Decomposition(EMD) used in the process of HHT decompose the signal into finite number of Intrinsic Mode Functions(IMFs). Since the decomposition is based on the local characteristic time scale of the data, and applicable to nonlinear and non-stationary processes, this technique does not requires any apriory input parameters for the decomposition. Hence it is adaptive in nature.

4.2 Empirical Mode Decomposition

Empirical Mode Decomposition(EMD) is a method for breaking a signal to its fundamental components known as Intrinsic Mode Functions (IMFs). Unlike Fourier decomposition it decomposes any non linear and non stationary waves into finite frequency and amplitude modulated signal bases known as IMFs which are complete and nearly orthogonal. The advantage of EMD against Fourier decomposition is its applicability in analyzing Non linear and non stationary signals. In this process EMD filters out the fundamental components of the original waves on the basis of their instantaneous frequency i.e, the IMFs with higher values of instantaneous frequencies come out first compared to low frequency components. Some of the advantages of EMD over other decomposition techniques is described below.

- The functions(IMFs) into which a signal is decomposed are all in the time-domain and of the same length as the original signal allows for varying frequency in time to be preserved.
- IMFs from real world signals is important because natural processes often have multiple causes, and each of these causes may happen at specific time intervals. This type of data is evident in an EMD analysis, but quite hidden in the Fourier domain or in wavelet coefficients.

Due to the above facts EMD method serves a great analysis tool in analysing nonlinear and non stationary signals of seismic data, results of neuroscience experiments, electrocardiograms, gastroelectrograms, and sea-surface height (SSH) readings.

4.2.1 Methodology

For getting the IMFs from a given signal, the signal should satisfy fallowing criteria.

- The signal has at least two extrema-one maximum and one minimum
- The characteristic time scale is defined by the time lapse between the extrema

- If the data were totally devoid of extrema but contained only inflection points, then it can be differentiated once or more times to reveal the extrema

Keeping in mind the properties of IMFs

- Number of extreme values and zero crossings differ at most by one.
- The 'local mean' value of the mode is zero.

We can follow the following steps to obtain the IMFs from a given time series data.

Sifting

The process of sifting can be carried out in following steps.

1. Compute all maximum (minimum) values of time series($X(t)$).
2. Perform cubic spline interpolation to construct $S_{max}(S_{min})$ splines, for these values.
3. Estimate the local mean of the time series as $m(t) = \frac{1}{2}(S_{max} + S_{min})$
4. Subtract $m(t)$ from $X(t)$ to obtain the first sifted time series h_1.

Fig.(4.1) provides a graphical illustration of the sifting process.

Getting IMFs using iteration of the sifting process

Once we are done with the first sifting process we will have to repeat this process again and again to get the subsequent sifting as in equation(4.4).

$$h_{1(k-1)} - m_{1k} = h_{1k} \tag{4.4}$$

Here $h_{1(k-1)}$ represents the $k-1^{st}$ sift and h_{1k} represents the k^{th} sift

And once the $k-1^{st}$ and k^{th} sift satisfy equation (4.5) we can call h_{1k} as our first IMF c_1

$$SD = \sum_{t=0}^{T} \left[\frac{|h_{1(k-1)}(t) - h_{1(k)}(t)|^2}{h^2_{1(k-1)}(t)} \right] \tag{4.5}$$

This process is graphically described in Fig.(4.2)

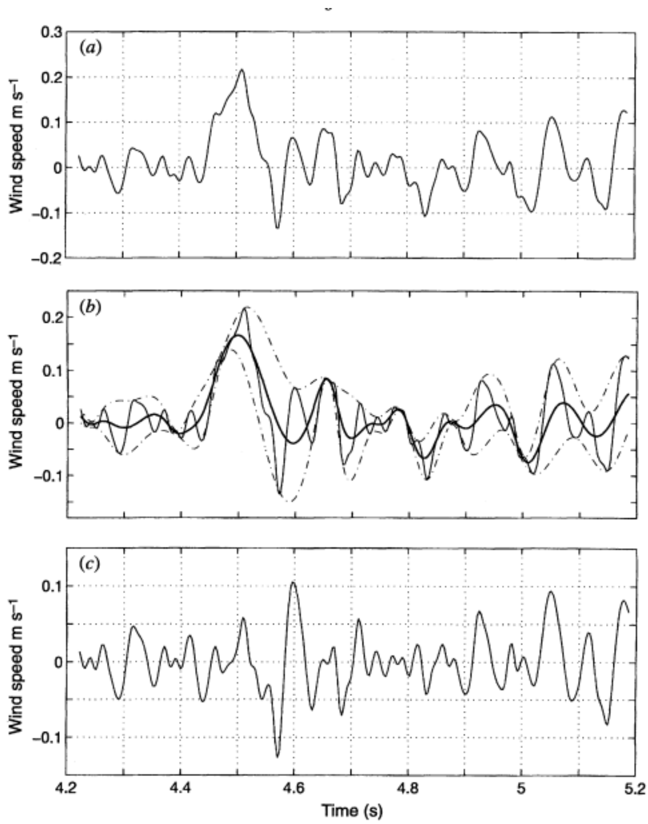


Figure 4.1: Illustration of the sifting processes: (a) the original data; (b) the data in thin solid line, with the upper and lower envelopes in dot-dashed lines and the mean in thick solid line; (c) the difference between the data and ml. This is still not an IMF, for there are negative local maxima and positive minima suggesting riding waves(24)

Getting subsequent IMFs

Once first IMF is obtained we will have to subtract it from the original signal.

$$r_1 = X(t) - c_1 \tag{4.6}$$

and repeat the sifting and iterations as described in previous steps on r_1 till getting an IMF.

And this process can be repeated to obtain all the IMFs and the monotone residue. Thus we have the following equation which describes the whole EMD process.

$$r_1 - c_2 = r_2, ..., r_{n_1} - c_n = r_n \tag{4.7}$$

And according to the completeness property of Empirical Mode Decomposition

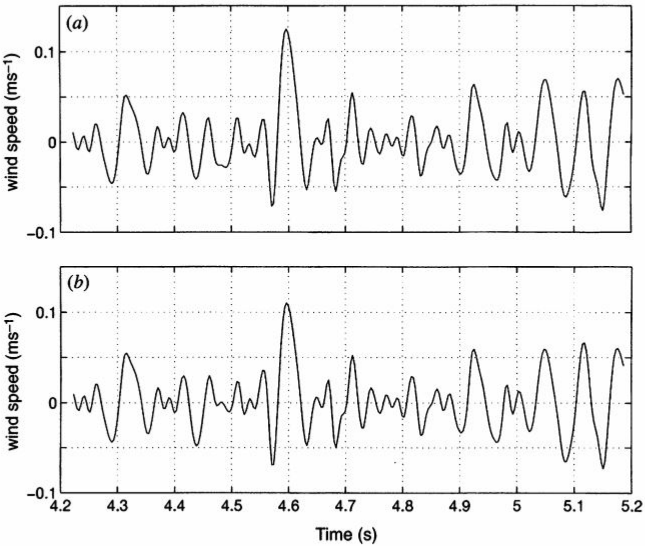


Figure 4.2: Illustration of the effects of repeated sifting process: (a) after one more sifting of the result in figure 3c, the result is still asymmetric and still not a IMF; (b) after three siftings, the result is much improved, but more sifting needed to eliminate the asymmetry. The final IMF is shown in figure 2 after nine siftings.

we will have

$$X(t) = \sum_{i=1}^{n} c_i + r_n \tag{4.8}$$

4.2.2 Algorithm in a nutshell

1. Start with a discrete time series $X(t)$ with sampling period T_s.
2. Compute all maximum (minimum) values of $X(t)$, perform cubic spline interpolation, $S_{max}(S_{min})$, for these values.
3. Estimate the local mean of the time series as $m(t) = \frac{1}{2}(S_{max} + S_{min})$
4. Subtract $m(t)$ from $X(t)$ and repeat step-1, 2, 3 with the resulting time series, until a stopping criterion is reached.
5. Once the stopping criterion is reached name the function as first IMF (c_1). Subtract this IMF from the original signal and repeat the steps-1, 2, 3, 4 until getting the residue which does not have enough extremas for interpolation.

Chapter 5

The BCI system

5.1 Brain Computer Interface

5.1.1 Introduction

Most of the BCI systems consist of a recording device which records the brain activity, a processing unit: which processes the recorded brain activity to useful instruction for mechanical devices and an external device which uses the brain signals to carry out various desired functions.

5.1.2 Recording devices

Recording can be categorised into invasive and noninvasive based on the technique used during the measurement of the brain activity. However generally noninvasive measurement is preferred over invasive one compromising the resolution due to convenience in performing experiments without putting the subject into life threat. The non invasive recording devices can be categorised based on the technology they use for recording the brain activity. For the time being only EEG, MEG and fMRI are vastly used for this purpose to record electrical activity, magnetic activity and blood flow change. However EEG is the most popular recording technique used in this field due to its cost effectiveness and other advantages. Further more due to recent advancements in the manufacture of low cost EEG recording devices like Emotiv EPOC, the BCI research has gained a remarkable interest amongst students and scientific community.

5.1.3 Processing unit

The processing of the brain activity recording is mainly a signal processing task and can be divided into pre processing, feature extraction, classification and post processing. For our case pre processing is mainly done in EEG device itself, it eliminates unnecessary signal component which is irrelevant from an EEG signal point of view. Meaning in this process very high frequency and amplitude components are eliminated which may have been caused due to the effect of other electrical and magnetic activity of the environment. In feature extraction part we mainly concentrate on the feature of desired activity in EEG signals. EEG signal can have a wide variety of features that can be classified in the next step. Some of the features of EEG signals are amplitude values of EEG signals, Band Powers (BP), Power Spectral Density (PSD) values, Auto Regressive (AR) and Adaptive Auto Regressive (AAR) parameters, Time-frequency features and inverse model-based features etc. These EEG features of a certain activity is classified from the other activities based on the classifiers. This classification technique is mainly a machine learning technique which can use Linear Classifiers, Neural Networks, Nonlinear Bayesian classifiers, Nearest Neighbor Classifiers and a wide variety of other classification methodology to extract the desired features from the EEG signal. After the feature classification process the processing unit assigns various parameter values to the classified features and sends it to the next step. This process is called post processing.

5.1.4 Communication with external device

The Processing unit communicates with the external device based on the post processing parameters extracted from the EEG signal. The communication system can be considered as the integration of the software system of the external device with the processing unit output system. The output of the post processing step is fed into the application of the external device and hence the external device performs the functions according to the signals coming from the brain. In this way a Brain Computer Interface(BCI) directly communicates with the brain to carry out desired functions with the external device and hence it enables a direct communications pathway between the brain and the object to be controlled.

5.2 Working model of a Brain Computer Interface(BCI) system

5.2.1 Introduction

Various components of the present Brain Computer Interface system is described in this section. As discussed in the previous section for a general BCI system we have divided our BCI system into three components for a better understanding.

- Brain activity recording device
- BCI Algorithm

- External device.

Fig(5.2) shows a block diagram for the working model. As figure shows the EEG signals from the headset is collected in a computer with the help of data acquisition system and is processed with the BCI algorithm and the processed data is again sent to the Drone API(external device software system) which controls the Drone.

Let us look at the system in more details.

5.2.2 Brain activity recording device

Various brain activities are recorded in a two stage process. In first stage the EEG headset collects data with much high sampling rate, eliminates the noise and send it to a computer with relatively low sampling rate.

EEG headset

As discussed in the previous section for the non invasive mode of BCI system EEG is the most popular amongst the BCI researchers and most of the BCI literature is based on the EEG based BCI systems. So its more convenient to chose the EEG mode of recording for a quick start in this field. Apart from all, recent advancements in the commercialization of low end EEG devices serves as a great tool for the very purpose. Hence we have chosen Emotiv EPOC head set as our brain activity recording device for the BCI system. It is a 16 channel(including gyro sensors) EEG device. The 14 EEG channel locations of this device according to 10-20 EEG system are AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4. Fig(2.3) and Fig(2.4) shows the electrode position and easy wear headset. Table(5.1) shows technical details for EPOC. More details about the headset can be found in (https://emotiv.com).

The headset collects electrical signals from various parts of the brain with a sampling rate 128 SPS (2048 Hz internal). The 2048 Hz internal sampling is being filtered and sent at a rate of 128 SPS as output via blue tooth to a computer for further processing and desired usage.

Data acquisition

The data acquisition software which in our case python-emotiv receives data from the headset through blue tooth connection, stores it to a temporary file in computer from where the data can be visualised using graphical interface and can be utilised for further processing. More details about python-emotiv can be found in (https://github.com/ozancaglayan /python-emotiv). It is a Python module for accessing Emotiv EPOC EEG headset on Linux. It can be easily cloned from its github repository and installed in Linux computer for accessing the raw data from

	EEG HEADSET
Number of channels	14 (plus CMS/DRL references, P3/P4 locations)
Channel names (International 10-20 locations)	AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4
Sampling method	Sequential sampling. Single ADC
Sampling rate	128 SPS (2048 Hz internal)
Resolution	14 bits 1 LSB = 0.51μV (16 bit ADC, 2 bits instrumental noise floor discarded)
Bandwidth	0.2 - 45Hz, digital notch filters at 50Hz and 60Hz
Filtering	Built in digital 5th order Sinc filter
Dynamic range (input referred)	8400μV (pp)
Coupling mode	AC coupled
Connectivity	Proprietary wireless, 2.4GHz band
Power	LiPoly
Battery life (typical)	12 hours
Impedance Measurement	Real-time contact quality using patented system

Figure 5.1: Showing detailed specification of the Emotiv EPOC headset

Emotiv EPOC headset.

5.2.3 BCI system

The BCI system is basically a translating and communicating system which establishes a communication pathway between the data acquisition system and the external device software system. BCI system consists of

1. BCI Algorithm

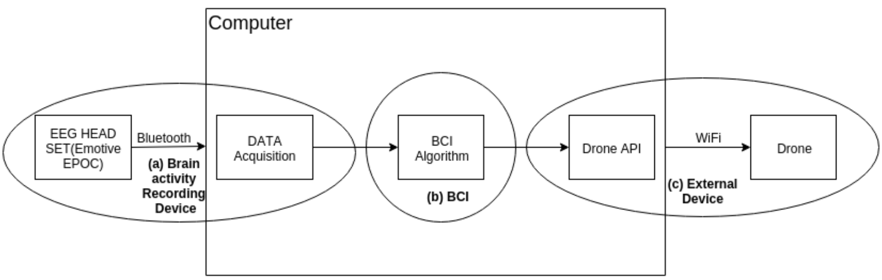


Figure 5.2: Illustration of the BCI system (a) Brain activity measuring device(EEG headset, Data acquisition device), (b) BCI Algorithm, (c) External Device(Drone API, Drone)

2. Communication System

BCI algorithm

The BCI algorithm is a classifier which receives raw data from the data acquisition system and processes it to detect the eye blink events in real time from the data sample. And this detection algorithm converts the raw data to a binary output(1 for every blinking events and 0 for no blinks).

Communication system

Based on the output of the BCI Algorithm(Classifier) it converts the EEG signals to useful command for the external device which in our case is the Drone API. For every two blinks in a given interval of time(2 seconds) it translates the EEG signal to Take off/Land and for every one blink in between (while the Drone is in air) it translates to commands: move forward/backward.

More details about the BCI algorithm and Communication system is provided in subsequent chapters.

5.2.4 External device

In our BCI system we are using a Drone(Parrot AR Drone) to carry out desired works from the EEG signals. This external device can be understood in two parts.

- Drone API
- Drone

The Drone API is a software that maintains a communication between the drone and computer.

Drone API

We are using python-ardrone program for this purpose. This API is the python module for controlling AR.Drone with a computer. This program takes keyboard inputs and from the computer keyboard and makes it drone commands and sends it to the Drone via the computer WiFi to make communication between the computer and the Drone. However we have modified the program to take the inputs from our BCI system in place of the key board inputs. More details about this software package can be found in (https://github.com/venthur/python-ardrone) github repository.

Drone

We are using Parrot AR.Drone as an external device to showcase the function of our BCI system. Parrot AR.Drone is a remote controlled flying quad copter helicopter built by the French company Parrot. The drone is designed to be controlled by mobile or tablet operating systems such as the supported iOS or Android within their respective apps however we are using python module to run the drone as discussed previously. More details about this drone can be found in (http://ardrone2.parrot.com).

Chapter 6

EEG analysis methodology

6.1 Introduction

We know from our previous chapter that every BCI system has a processing unit which processes the preprocessed recorded signal and converts it useful commands for the desired function to be carried out. In our case we are choosing the amplitude of the EEG signal for the eye-blink event as the feature and hence a eye-blink detection algorithm is developed to classify the desired features for the eye-blink event from the other events. Fig(6.1) illustrates the eye-blink detection algorithm. This feature extraction classifier algorithm can be understood in following steps

1. Empirical Mode Decomposition of EEG signal
2. Normalisation of the residue
3. Classification using cut off amplitude value

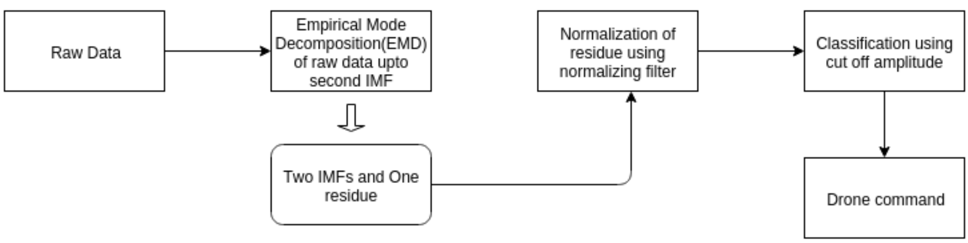


Figure 6.1: Illustration of the blink detection algorithm using block diagrams

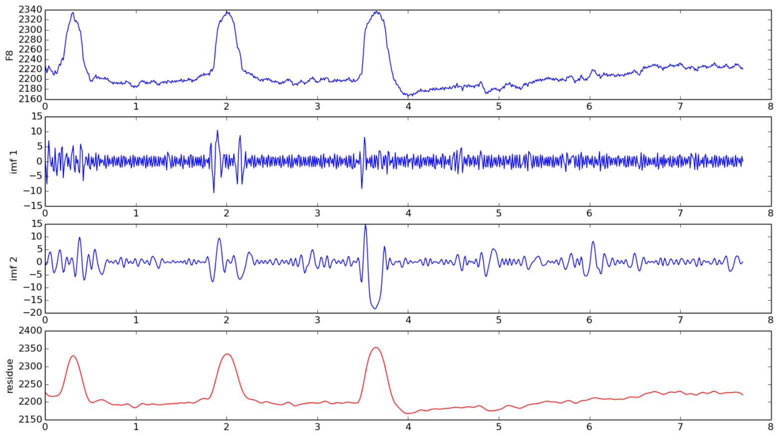


Figure 6.2: Empirical Mode Decomposition of the EEG signal collected from the F8 electrode of the Emotiv EPOC headset

6.2 Empirical Mode Decomposition of EEG signal

EEG signals are nonlinear and non-stationary in nature as it satisfies all the requirements for that. Hence Empirical Mode Decomposition is a great tool for analysing these signals. For our purpose we are collecting the EEG signals from the F8 electrode of the Emotiv EPOC headset and as the first step of the processing we are decomposing it using the Empirical Mode Decomposition till we obtain the first two IMFs of the original signal. As we know IMFs are frequency and amplitude modulated narrow band signals and the IMFs come in the decreasing order of their frequency value narrow bands. The first two IMFs remove most of the fluctuations from the original signal which otherwise lead to problematic situations in subsequent steps. Once the first two IMFs are removed from the original signal the residue is sent to the next step for further processing. Fig(6.2) demonstrates the Empirical Mode Decomposition of the F8 EEG signals. The red curve in the figure represents the residue after second IMF which is the desired signal for the next step. It can be easily observed that residue after second IMF only contains the high amplitude values without any high frequency fluctuations through out.

6.3 Normalisation of the residue

The humps in terms of higher amplitude values in the residue represents the eye blink events and we are particularly interested in classifying these events from the other possible events. For this purpose the first step would be to bring the whole

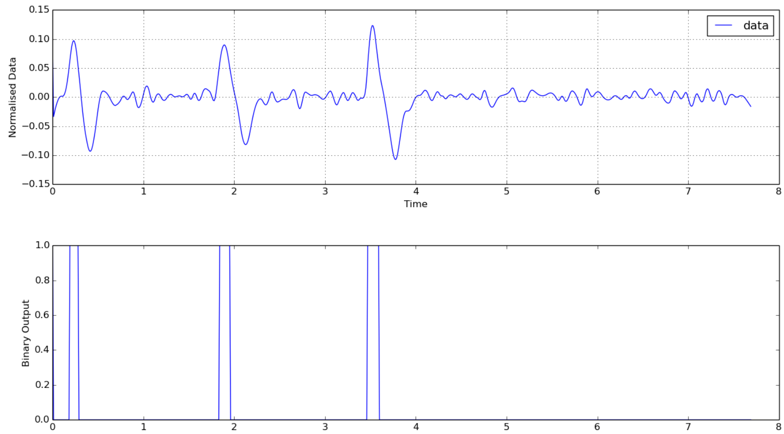


Figure 6.3: Conversion of normalised EEG signals to binary output signals using the cut off amplitude. First figure shows the normalised residue and the second figure shows the corresponding binary output

wave form to zero level and we can do this using the function in eq.(6.1). Where we transform each point of the residue and assign a new value around zero. In eq.(6.1) r_i represents the i^{th} point of the residue time series and y_i represents the i^{th} value of the transformed new transformed time series and the multiplication factor comes from the empirical analysis.

$$y_i = (0.0089) \times (r_i - r_{i-1} + y_{i-1}) \tag{6.1}$$

6.4 Classification using cut off amplitude value

Once normalization of the residue is completed the eye blink events can be easily distinguished from the other events and can be easily extracted by setting a cut off amplitude level. Fig(6.3) represents the normalised signal along with the translated binary signal using the cut off amplitude level. It can be easily seen that the normalised signal falls around zero and the high amplitude pulses can be easily extracted from the rest of the signals by setting a cut off value of 0.05. Hence for each high amplitude pulse we can get a 1 value and for rest of the signal we will have 0. Thus we can translate the normalised residue to a string of binary digits where each 1 represents an eye blink event. This binary array signal is then sent to the communication system which assigns various commands to the external device based on the temporal position of the value 1.

Chapter 7

The BCI communication system

7.1 Introduction

We have already described the function of various parts of the processing unit of a general Brain Computer Interface(BCI) system. In this chapter we will discuss the post processing part of the processing unit. The over all function of this part is to make a communication pathway between the EEG device and the external device based on the classified features from the signal. For the sack of simplicity let's understand it in two layers.

- Assignment of characters to the classified features based on their temporal position.
- Assignment of drone commands to the characters collected from the previous step.

Lets look into the whole process one by one.

7.2 Assignment of characters to the classified features based on their temporal position

The data acquisition API stores the data in a temporary file inside the computer. Once stored as a temporary file the function of this part begins. In this process the data from the temporary file is read by the program. Every time the program takes the latest 256 (two sec) data and analyse it with the help of the BCI algorithm to detect if the data sample contains any eye blink events within this. Let's say the program is in mode 'y' while it has not detected two blinks in a given time interval.

While in mode 'y' each time it detects an eye blink in the sample it raises a flag for the blink and starts a timer but the process goes on and once it detects the next blink within two seconds it assigns it a value let say 'a' and it turns its mode to 'x' and the process goes on. While in mode 'x' when ever it detects one blink it raises a flag for the blink starts the timer again and if it detects another blink within the given frame it assigns a value 'b' and comes to mode 'y' and if it doesn't detect any blink in the given frame it assigns a value let say 's' which can take dual values 'u' and 'v' based on the previous value chosen by the program.

7.3 Assignment of drone commands to the characters collected from the previous step

In this step the program assigns various drone commands to the characters obtained from the previous step. And the commands are directly fed into the drone API in the computer so that the drone can be directed according to the EEG signals. Following are the drone commands assigned to the characters from the previous step.

1. a= Take off
2. u= Move forward
3. v= Move backward
4. b= Land

Fig(7.1) represents the post processing part of our brain computer interface system with the help of a flow chart.

Fig(7.2) represents the screen shot of the computer screen while the BCI system is in action.

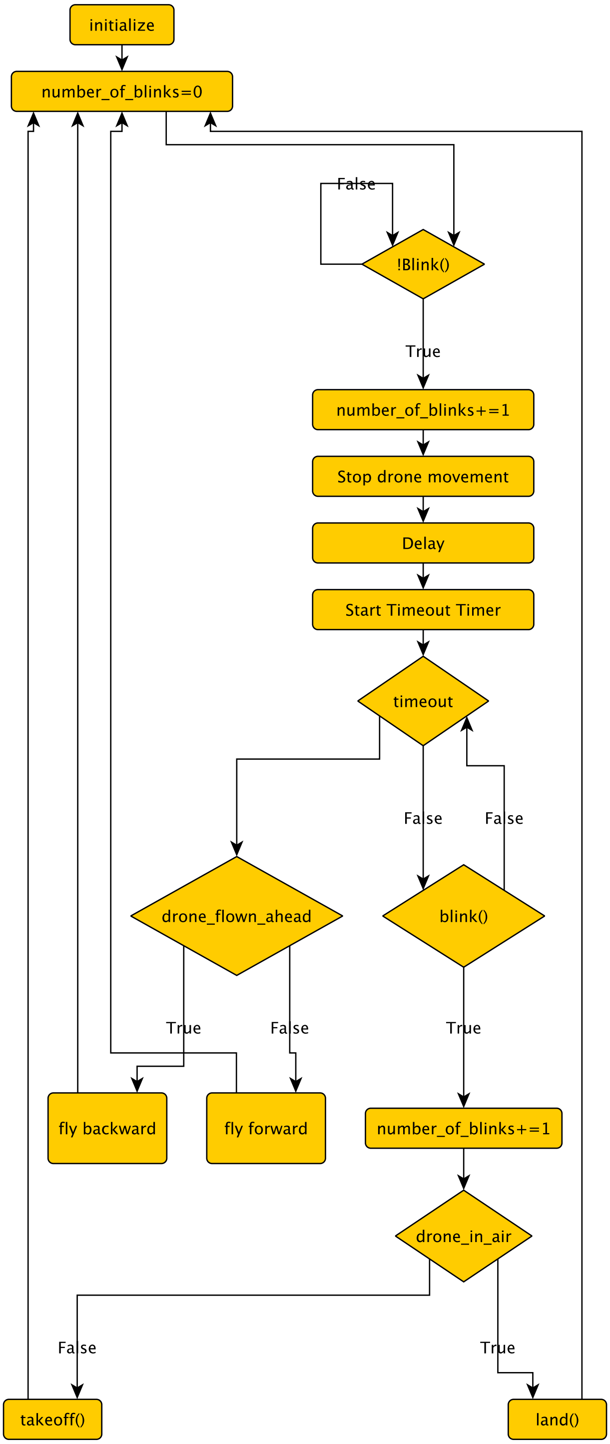


Figure 7.1: Flowchart showing the complete communication algorithm that reads the binary signals from the EEG Drone Interface and converts it to drone commands

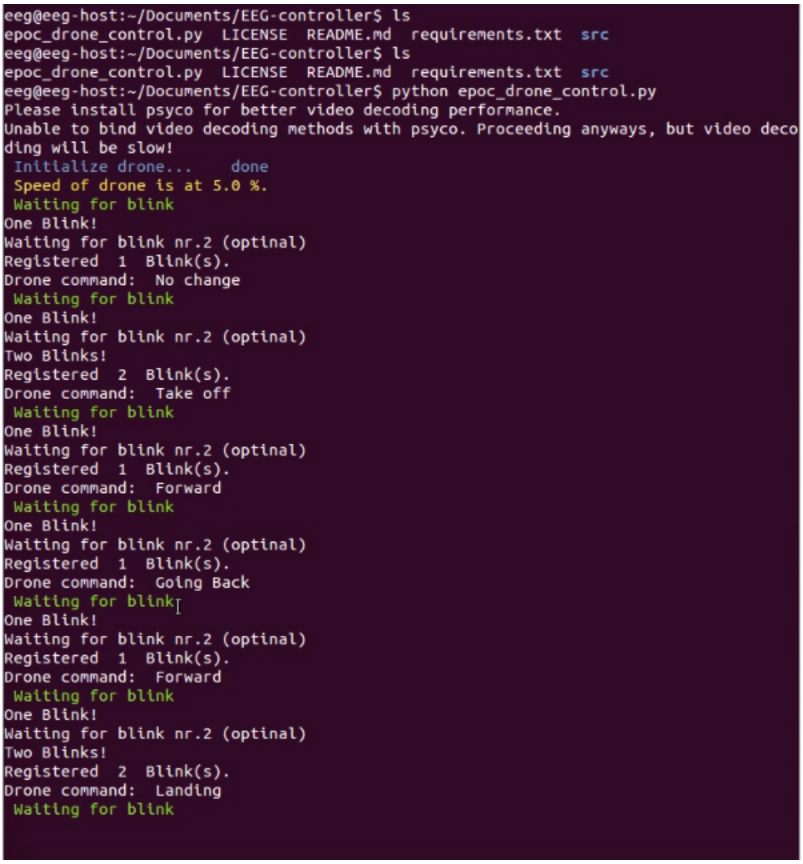


Figure 7.2: Screen shot of the computer screen while the Brain Computer Interface system in action

Chapter 8

Results and Conclusion

8.1 Results

Four subjects including two females (12, 36 years) and 2 males (20, 44 years) participated in the tests and verify the system. The results for various events and drone commands are noted down in Table(8.1). It was found that 84% of the time the system seems to work perfectly which is a significant accuracy the BCI system to be EEG based and the experiments were performed in outdoor environment. Fig.(8.1) mimics the experiments which are performed for testing and verifying system.

8.2 Conclusion

Although BCI research is relatively young, many advances have been achieved in a little over two decades, because many of these methods are based on previous signal processing and pattern recognition research. Many studies have demonstrated the valuable accuracy of BCIs and provided acceptable information bit rate, despite the inherent major difficulties in brain signal processing.

Table 8.1: Experimentation using the BCI system

Subjects	Take off	Move Forward	Move Backward	Land
12(F)	Y,Y	Y,Y	Y,Y	Y,N
36(F)	Y,Y	Y,N	Y,Y	Y,Y
20(M)	Y,N	Y,Y	Y,Y	Y,Y
44(M)	Y,Y	Y,N	Y,Y	Y,N

Figure 8.1: Verification of the BCI system

This report has described the fundamental aspects of a BCI system along with a working model which uses the Emotiv EPOC EEG recordings and process it to give instruction to a drone for performing various actions according to the users desire. The BCI has been tested and verified by subjects covering a wide age group and both genders in outdoor conditions. The accuracy of the BCI has been found to be quite high in comparison to the training period.

Presented BCI system is a module of the BCI system which is being developed at Department of Cybernetics, NTNU. Further improvements to the BCI system can be carried out by adding the features of imagined motor movement detection using the Emotiv EPOC headset.

Bibliography

[1] Akilandeswari, K., and G. M. Nasira. "Swarm Optimized Feature Selection of EEG Signals for Brain-Computer Interface."

[2] Mallick, Abhijeet, and Deepak Kapgate. "DEVELOPMENT OF EFFECTIVE SIGNAL PRE-PROCESSING ALGORITHMIC TECHNIQUE FOR HYBRID BCI."

[3] Lebedev, Mikhail A., and Miguel AL Nicolelis. "BrainâĂŞmachine interfaces: past, present and future." TRENDS in Neurosciences 29.9 (2006): 536-546.

[4] Vidal, Jean-Jacques. "Toward direct brain-computer communication." Annual review of Biophysics and Bioengineering 2.1 (1973): 157-180.

[5] Berger, Hans. "ÃIJber das elektrenkephalogramm des menschen." European Archives of Psychiatry and Clinical Neuroscience 87.1 (1929): 527-570.

[6] Duvinage, Matthieu, et al. "A P300-based quantitative comparison between the Emotiv Epoc headset and a medical EEG device." Biomedical Engineering 765 (2012): 2012-764.

[7] Babiloni, F., et al. "Linear classification of low-resolution EEG patterns produced by imagined hand movements." Rehabilitation Engineering, IEEE Transactions on 8.2 (2000): 186-188.

[8] Caton, Richard. "Electrical Currents of the Brain." The Journal of Nervous and Mental Disease 2.4 (1875): 610.

[9] Birbaumer, Niels, et al. "A spelling device for the paralysed." Nature 398.6725 (1999): 297-298.

[10] Keirn, Zachary, and Jorge I. Aunon. "A new mode of communication between man and his surroundings." Biomedical Engineering, IEEE Transactions on 37.12 (1990): 1209-1214.

[11] Penny, William D., et al. "EEG-based communication: a pattern recognition approach." IEEE Transactions on Rehabilitation Engineering 8.2 (2000): 214-215.

[12] Pfurtscheller, Gert, Doris Flotzinger, and Joachim Kalcher. "Brain-computer interfaceâĂŤa new communication device for handicapped persons." Journal of Microcomputer Applications 16.3 (1993): 293-299.

[13] Vaughan, Theresa M., Jonathan R. Wolpaw, and Emanuel Donchin. "EEG-based communication: prospects and problems." Rehabilitation Engineering, IEEE Transactions on 4.4 (1996): 425-430.

[14] Wolpaw, Jonathan R., et al. "An EEG-based brain-computer interface for cursor control." Electroencephalography and clinical neurophysiology 78.3 (1991): 252-259.

[15] McFarland, Dennis J., et al. "Spatial filter selection for EEG-based communication." Electroencephalography and clinical Neurophysiology 103.3 (1997): 386-394.

[16] Penny, William D., and Stephen J. Roberts. "Bayesian neural networks for classification: how useful is the evidence framework?." Neural Networks 12.6 (1999): 877-892.

[17] Penny, William D., et al. "EEG-based communication: a pattern recognition approach." IEEE Transactions on Rehabilitation Engineering 8.2 (2000): 214-215.

[18] World Health Organization. Neurological disorders: public health challenges. World Health Organization, 2006.

[19] Berg, Patrick, and Michael Scherg. "Dipole models of eye movements and blinks." Electroencephalography and clinical neurophysiology 79.1 (1991): 36-44.

[20] Harlan, William L., Philip T. White, and Reginald G. Bickford. "Electric activity produced by eye flutter simulating frontal electroencephalographic rhythms." Electroencephalography and clinical neurophysiology 10.1 (1958): 164-169.

[21] Borer, Katarina T. Exercise endocrinology. Human Kinetics, 2003.

[22] Teplan, Michal. "Fundamentals of EEG measurement." Measurement science review 2.2 (2002): 1-11. bibbibb

[23] Prior, Pamela F. "Fundamentals of EEG Technology. Vol. 1 Basic Concepts and Methods." Journal of neurology, neurosurgery, and psychiatry 47.3 (1984): 326.

[24] Huang, Norden E., et al. "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis." Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences. Vol. 454. No. 1971. The Royal Society, 1998.

[25] Huang, Norden E., Zheng Shen, and Steven R. Long. "A new view of nonlinear water waves: The Hilbert Spectrum 1." Annual review of fluid mechanics 31.1 (1999): 417-457.

[26] Bendat, Julius S., and Allan G. Piersol. Random data: analysis and measurement procedures. Vol. 729. John Wiley Sons, 2011.

Appendix A

Python codes for the BCI

A.1 Data acquisition for BCI

```
from colour import c

def read_sensor_data
(filename, number_of_samples, column_of_interest):
        """
␣␣␣␣␣␣␣␣Reads␣data␣from␣the␣buffer␣generated␣by
␣␣␣␣␣␣␣␣the␣python-emotiv␣software.
␣␣␣␣␣␣␣␣It␣calls␣the␣function
␣␣␣␣␣␣␣␣xtract_relevant_sensors_data
␣␣␣␣␣␣␣␣to␣format
␣␣␣␣␣␣␣␣the␣raw␣data␣to␣usable␣data,
␣␣␣␣␣␣␣␣reading␣only
␣␣␣␣␣␣␣␣from␣one␣sensor.
␣␣␣␣␣␣␣␣This␣function␣does␣not␣change
␣␣␣␣␣␣␣␣the␣frame␣buffer.
␣␣␣␣␣␣␣␣"""
        global c
        try:
                f = open(filename, 'r')
        except Exception, e:
                print c.red, 'Unable␣to␣open␣the␣file
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣%s,check␣the␣sensors' % (filename), c.end

        sensor_data = []
```

```
        count = 0
        for line in reversed(f.readlines()):
        # Read file from bottom to get newest data
            sensor_data.append(line)
            count += 1
            if count > number_of_samples:
                break
                # Stop reading when enough samples

        return extract_relevant_sensors_data
        (sensor_data, column_of_interest)

def extract_relevant_sensors_data(data, column_of_interest):
        """
␣␣␣␣␣␣␣␣Extract␣floats␣from␣the␣text␣file
␣␣␣␣␣␣␣␣and␣discard␣data␣that␣is␣not
␣␣␣␣␣␣␣␣relevant.␣Relevant␣data␣is
␣␣␣␣␣␣␣␣given␣by␣column_of_interest,␣corresponding
␣␣␣␣␣␣␣␣to␣the␣sensor␣of␣interest.
␣␣␣␣␣␣␣␣"""

        splitting_symbol = '␣'
        # Defined by the epoc driver
        relevant_data = []

        for rows in data:
        # extract the relevant data from the right column
                temp = (rows.split(splitting_symbol)
                [column_of_interest])
                temp = temp.replace("\n", "")
                # Clean the string
                temp = float(temp)
                relevant_data.insert(0, temp)

        return relevant_data
```

A.2 Blink detection algorithm

```
import numpy                    as np
```

```
from    scipy                    import interpolate
from    scipy.interpolate        import splrep, splev, interp1d
from    scipy.signal             import argrelmin, argrelmax
import  warnings
warnings.simplefilter("ignore")

tol             = 0.05 # Zero crossing parameter
debug_mode      = False

def ndiff_extrema_zcrossing(x):
        """
␣␣␣␣␣␣␣␣Get␣the␣difference␣between␣the␣number␣of
␣␣␣␣␣␣␣␣zero␣crossings␣and␣extrema.
␣␣␣␣␣␣␣␣"""
        n_max = argrelmax(x)[0].shape[0]
        n_min = argrelmin(x)[0].shape[0]
        n_zeros = (x[:-1] * x[1:] < 0).sum()
        return abs((n_max + n_min) - n_zeros)

def sift(x):
        """
␣␣␣␣␣␣␣␣One␣sifting␣iteration.
␣␣␣␣␣␣␣␣1.␣Calculate␣the␣mean␣envelop␣by␣taking␣the
␣␣␣␣␣␣␣␣average␣of␣maxima␣envelop␣and␣minima␣envelop
␣␣␣␣␣␣␣␣2.␣Subtract␣the␣data␣set␣from␣the␣mean␣envelop
␣␣␣␣␣␣␣␣"""
        maxima = argrelmax(x)[0]
        minima = argrelmin(x)[0]
        x_upper = np.zeros((maxima.shape[0] + 2,))
        x_upper[1:-1] = maxima
        x_upper[-1] = x.shape[0] - 1
        x_lower = np.zeros((minima.shape[0] + 2,))
        x_lower[1:-1] = minima
        x_lower[-1] = x.shape[0] - 1
        tck1 = splrep(x_upper, x[x_upper.astype(int)])
        upper_envelop = splev(np.arange(x.shape[0]), tck1)
        tck2 = splrep(x_lower, x[x_lower.astype(int)])
        lower_envelop = splev(np.arange(x.shape[0]), tck2)
        mean_amplitude =
        np.abs(upper_envelop - lower_envelop) / 2
```

```
        local_mean = (upper_envelop + lower_envelop) / 2
        amplitude_error = np.abs(local_mean) / mean_amplitude
        return x - local_mean, amplitude_error.sum()

def emd(x, n_imfs):
        """
␣␣␣␣␣␣␣␣Compute␣IMFs␣by␣performing␣Empirical␣Mode
␣␣␣␣␣␣␣␣Decomposition␣n_imfs␣times
␣␣␣␣␣␣␣␣"""
        imfs = np.zeros((n_imfs + 1, x.shape[0]))
        for i in xrange(n_imfs):
                mode = x - imfs.sum(0)
                while (ndiff_extrema_zcrossing(mode) > 1):
                        mode, amplitude_error = sift(mode)
                        if amplitude_error <= tol:
                                break
                imfs[i, :] = mode
        imfs[-1, :] = x - imfs.sum(0)
        return imfs

def low_pass(time,imfs):
        """
␣␣␣␣␣␣␣␣Takes␣the␣residue␣from␣emd␣and␣passes
␣␣␣␣␣␣␣␣the␣low␣frequency␣gignals␣below␣cut␣off␣level.
␣␣␣␣␣␣␣␣"""
        filt_tim=time
        filt_input=imfs[-1,:]
        if debug_mode:
                plt.plot(filt_tim ,filt_input)
                plt.title('Residue␣after␣first␣IMF')
                plt.ylabel('Amplitude')
                plt.xlabel('Sample(n)')
                plt.show()
        filt_output=[]
        y0=0

        for i in range(len(filt_input)):
                y1=100*((0.009/(0.009+1))*
                (filt_input[i]-filt_input[i-1]+y0))
                filt_output.append(y1)
```

```
            y0=y1
        #Defn of low pass filter
        (https://en.wikipedia.org/wiki/Low-pass_filter)
        see:Discrete-time realization
        return filt_output


def EMD_lowpass_blinks(time,dat3):
        """
␣␣␣␣␣␣␣␣Takes␣the␣raw␣data␣and␣returns␣the
␣␣␣␣␣␣␣␣binary␣blinks
␣␣␣␣␣␣␣␣"""
        input_EMD_lowpass=np.asarray(dat3)
        time_samples=np.asarray(time)
        n_imfs = 2
        imfs = emd(input_EMD_lowpass, n_imfs)
        global lowpass_out
        lowpass_out=low_pass(time,imfs)
        blink_bin=blink_detect(lowpass_out)
        return blink_bin

def blink_detect(filt_output):
        """
␣␣␣␣␣␣␣␣takes␣the␣filtered␣data␣and␣gives␣bin␣output
␣␣␣␣␣␣␣␣for␣blinks
␣␣␣␣␣␣␣␣"""
        global blink_step
        blink_step=[int(filt_output[fi]>50)
        for fi in range(len(filt_output))]
        global blink_bin
        blink_bin=[int(blink_step[bl+1] > blink_step[bl])
        for bl in range(len(blink_step)-1)]
        # appending 1-step to match the length of string
        blink_bin.append(blink_bin[len(blink_bin)-2])
        return blink_bin
```

A.3 Post processing codes

```
def sleep_t(drone, sec):
```

```
        """
␣␣␣␣␣␣␣␣sleep_t␣is␣a␣sleep␣function␣that␣does␣not␣halt
␣␣␣␣␣␣␣␣any␣processes.
␣␣␣␣␣␣␣␣It␣waits␣for␣a␣time␣given␣by
␣␣␣␣␣␣␣␣sec␣in␣seconds.

␣␣␣␣␣␣␣␣It␣takes␣in␣the␣drone␣object␣and␣uses
␣␣␣␣␣␣␣␣the␣global␣variables
␣␣␣␣␣␣␣␣drone_flying␣and␣drone_flown_ahead
␣␣␣␣␣␣␣␣to␣maintain␣forward␣and
␣␣␣␣␣␣␣␣backward␣orders.
␣␣␣␣␣␣␣␣"""
        t0 = time()
        t1 = t0
        while t1 < t0 + sec:
                t1 = time()
                if drone_flying and drone_flown_ahead:
                        drone.move_forward()
                elif drone_flying:
                        drone.move_backward()


def drone_command(drone, number_of_blinks):
        """
␣␣␣␣␣␣␣␣Gives␣commands␣to␣the␣drone␣based␣on
␣␣␣␣␣␣␣␣the␣number␣of␣blinks
␣␣␣␣␣␣␣␣given␣by␣number_of_blinks

␣␣␣␣␣␣␣␣The␣drone␣input␣is␣the␣drone␣object
␣␣␣␣␣␣␣␣generated␣by␣ARDrone

␣␣␣␣␣␣␣␣Protocol␣for␣control␣of␣drone
␣␣␣␣␣␣␣␣2␣blinks␣-␣takeoff␣or␣land
␣␣␣␣␣␣␣␣1␣blink␣␣-␣forward␣or␣backward
␣␣␣␣␣␣␣␣no␣blinks-␣hover/Stay␣still
␣␣␣␣␣␣␣␣"""
        global drone_in_air
        global drone_flown_ahead
        global drone_flying
        global c
```

```
print 'Drone␣command:␣',

if number_of_blinks > 1:
    drone_flying = False
    if drone_in_air:
        drone.land()
        drone_in_air = False
        print "Landing"
    else:
        try:
            drone.takeoff()
        except Exception, e:
            print c.red,
            'Cannot␣takeoff:␣', e, c.end
            raise
        drone_in_air = True
        print "Take␣off"
elif number_of_blinks == 1 and drone_in_air:
    drone_flying = True
    if drone_flown_ahead:
        drone.move_backward()
        drone_flown_ahead = False
        print "Going␣Back"
    else:
        drone.move_forward()
        drone_flown_ahead = True
        print "Forward"
else:
    print 'No␣change'
```

A.4 Terminal interface

```
class c:
    """
␣␣␣␣␣␣␣␣Defines␣colors␣that␣can␣be␣used␣when␣printing
␣␣␣␣␣␣␣␣to␣enhance␣readability.
␣␣␣␣␣␣␣␣"""
    pink = '\033[95m'
    blue = '\033[94m'
    green = '\033[92m'
```

```
        yellow = '\033[93m'
        red = '\033[91m'
        end = '\033[0m'

        def disable(self):
                self.pink = ''
                self.blue = ''
                self.green = ''
                self.yellow = ''
                self.red = ''
                self.end = ''
```

A.5 Post processing and drone API

```
from    time                               import time
from    libardrone                         import ARDrone

from    bci_filter_functions
import ndiff_extrema_zcrossing, sift,
emd, low_pass, EMD_lowpass_blinks, blink_detect
from    bci_get_sensordata
import read_sensor_data, extract_relevant_sensors_data
from    colour          import c
import  warnings
warnings.simplefilter("ignore")

freq_sensor                     = 128
drone_in_air            = False
drone_flown_ahead       = False
drone_flying            = False

blink_duration = 2
# The time in sec to wait after a blink has been detected
max_duration_between_blinks = 2
# sec

def main():
        global c
        global blink_duration
        global max_duration_between_blinks
```

```
print c.blue, 'Initialize drone... ', c.red,
try:
        drone = ARDrone()
        sleep_t(drone, 1)
except Exception, e:
        print c.red, 'Error connecting to the drone: ',
        e, c.end
        raise
speed = 0.05
drone.speed = speed
print c.blue, 'done', c.end

print c.yellow, 'Speed of drone is at', speed * 100,
'%.', c.end

try:
        Running = True
        while Running:
                number_of_blinks = 0;

                print c.green, 'Waiting for blink', c.e
                while blink() < 1:
                        pass

                print "One Blink!"
                number_of_blinks += 1
                drone_flying = False
                # Stop drone and wait for command

                sleep_t(drone, 1.2 * blink_duration)
                print
                'Waiting for blink nr.2 (optinal)'
                t0 = time()
                t1 = t0
                while t1 < t0 +
                max_duration_between_blinks:
                        if blink():
                                print "Two Blinks!"
                                number_of_blinks += 1
```

```
                                        break
                                t1 = time()

                        print 'Registered ',
                        number_of_blinks,
                        ' Blink(s).'
                        drone_command
                        (drone, number_of_blinks)
                        sleep_t(drone, 4)

        except Exception, e:
                print c.red, "Error in main: ", e, c.end
                raise
        finally:
                print c.blue, 'Enter Final state'
                if drone_in_air:
                        print c.red, 'Emergency landing!'
                drone.land()
                print "Shutting down drone...",
                drone.halt()
                print "Done.", c.end

def blink():
        """
        Return high if a blink is detected within a scope
        given by the global variable blink_duration.
        """
        global blink_duration
        sensor_data_location = '/tmp/epoc_data'
        sensor_number = 12 # Sensor that detects the blinks
        number_of_samples = freq_sensor * blink_duration
        # sampling frequency times three seconds of data

        raw_data=read_sensor_data
        (sensor_data_location, number_of_samples, sensor_number)
        sample_index = [i for i in range(len(raw_data))]
        return (1 in EMD_lowpass_blinks(sample_index, raw_data))

def sleep_t(drone, sec):
        """
```

```
        sleep_t is a sleep function that
        does not halt any processes.
        It waits for a time given by sec in seconds.
        It takes in the drone object and uses the
        global variables
        drone_flying and drone_flown_ahead to
        maintain forward and
        backward orders.
        """
        t0 = time()
        t1 = t0
        while t1 < t0 + sec:
                t1 = time()
                if drone_flying and drone_flown_ahead:
                        drone.move_forward()
                elif drone_flying:
                        drone.move_backward()


def drone_command(drone, number_of_blinks):
        """
        Gives commands to the drone based
        on the number of blinks
        given by number_of_blinks

        The drone input is the drone object
        generated by ARDrone
        Protocol for control of drone
        2 blinks - takeoff or land
        1 blink  - forward or backward
        no blinks- hover/Stay still
        """
        global drone_in_air
        global drone_flown_ahead
        global drone_flying
        global c

        print 'Drone command: ',

        if number_of_blinks > 1:
```

```
                drone_flying = False
                if drone_in_air:
                        drone.land()
                        drone_in_air = False
                        print "Landing"
                else:
                        try:
                                drone.takeoff()
                        except Exception, e:
                                print c.red, 'Cannot takeoff: ',
                                e, c.end
                                raise
                        drone_in_air = True
                        print "Take off"
        elif number_of_blinks == 1 and drone_in_air:
                drone_flying = True
                if drone_flown_ahead:
                        drone.move_backward()
                        drone_flown_ahead = False
                        print "Going Back"
                else:
                        drone.move_forward()
                        drone_flown_ahead = True
                        print "Forward"
        else:
                print 'No change'

if __name__ == "__main__":
    main()
```

Appendix B

An extended abstract

Experience designing a BCI for flying a drone based on the use of Empirical Mode Decomposition

Sumit Kumar Ram[1,2*], Geir Kulia[1], Marta Molinas[1]

[1]Norwegian University of Science and Technology, Trondheim, Norway;[2]Indian Institute of Science Education and Research, Kolkata, India

*P.O. Box 7030, Trondheim, Norway. E-mail: skr12ms039@iiserkol.ac.in

Introduction:

Recently there have been significant improvements and increased interest in BCI research due to development of affordable EEG recording devices, e.g. Emotive EPOC, which are today widely used by researchers as well as commercial actors [1]. A major challenge that occurs when using low-end EEG devices is a low signal to noise ratio. New approaches are therefore necessary to distinguish the signal from the noise. This paper will describe the design and implementation of an application used to classify, analyze and translate raw data from the EEG into navigation commands for a drone control application program interface (API).

The EEG signals of interests were obtained from the Emotive EPOC, which can detect intentional eye blink events. An eye blink detection algorithm was developed based on the Empirical Mode Decomposition (EMD) [2] technique. The analysis involves Empirical Mode Decomposition of the raw EEG signals up to the second Intrinsic Mode Function (IMF) [2], smoothening of the residue after second IMF and selection of the intentional eye blink by setting a cut-off signal strength. The signal obtained after the EMD manipulation was processed and converted into binary signals that were sent to the drone API. This API controlled the drone using Wi-Fi.

Material:

The entire system consisted of an Emotiv EPOC headset, which has 16 electrodes and can record signals at a sampling rate of 128 Hz [1], a personal computer with the operating system Ubuntu 15.10 and a wireless Drone. The application's dependencies consist only of free software. Four subjects participated in testing and verifying the system [3, 4]

Methods:

The BCI system acquires signals coming from the F8 electrode of the Emotive EPOC headset, it analyzes them using the eye blink detection algorithm,. The system applies the algorithm each time for the latest 256 points, which can subsequently give binary outputs as 1 or 0 in real time for intentional eye blink events and for other events respectively.

The translation phase of the BCI system takes this binary inputs and gives a ***take-off command*** whenever it detects two eye blinks in a certain interval of time (in our case it was 2 seconds) and gives ***forward/backward commands*** whenever it detects one blink while it is hovering in air and again gives ***landing command*** whenever it detects two blinks in a certain interval of time.

Results:

Four subjects including two females (12, 36 years) and 2 males (20, 44 years) participated in the tests and verify the system [3] . 16 event tests were conducted. The events included takeoff and landing as well as moving back and forth. Of the 16 tests, 13 were completely successful, which indicates an accuracy of 81 %.

Discussion:

The research work was carried out by the undergraduate researchers` who took Emotive EPOC headset for the development of their first BCI system without any a-priori experience. Further improvement in this project is to introduce the detection of imagined motor movement with the EPOC headset (motor imagery). On the other hand, the proposed technique can be modified to include an eye blink artifact removal algorithm.

Significance:

The described system is the first step towards building a complete BCI system, which can use the EEG signals to control unmanned vehicles for exploration, rescue and defense purposes.

Acknowledgements:

The authors would like to acknowledge Department of Engineering Cybernetics for the financial support through the Jens Balchen Scholarship to carry out the research work.

References (8pt)

[1] Duvinage, Matthieu, et al. "Performance of the Emotiv Epoc headset for P300-based applications." Biomedical engineering online 12.1 (2013): 56.

[2] Huang, Norden E., et al. "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis." Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences. Vol. 454. No. 1971. The Royal Society, 1998.

[3] M Lystad et al.(08.10.2015,kl. 10:48 p.m.) NRK Trøndelag [online]. Available: http://www.nrk.no/trondelag/far-drone-til-a-fly-ved-a-blunke-1.12492845.

[4] Geir Kulia (Nov 13, 2015) youtube [online]. Available: https://youtu.be/HD5_ZCt3YUE

Appendix C

Media

- NRK media
  Web
  http://www.nrk.no/trondelag/far-drone-til-a-fly-ved-a-blunke-1.12492845
  TV
  https://tv.nrk.no/serie/distriktsnyheter-midtnytt/DKTL98081015/10-08-2015t=59s
  Radio
  https://radio.nrk.no/serie/her-og-naa-hovedsending/DMNH01015615/10-08-2015t=1h20m2s

- Gemini.no
  http://gemini.no/2015/09/tenk-tanken-og-fly-av-garde/

- abc nyheter
  http://www.abcnyheter.no/livet/2015/09/06/194870294/tenk-tanken-og-fly-av-garde

- TEK.no
  http://www.tek.no/artikler/her-styrer-han-dronen-med-hjernebolger/192625

- TU.no
  http://www.tu.no/automatisering/incoming/2015/11/25/forsker-pa-tankestyring

- youtube link
  http://www.tu.no/automatisering/incoming/2015/11/25/forsker-pa-tankestyring