GUJARAT TECHNOLOGICAL UNIVERSITY

Chandkheda, Ahmedabad

Affiliated

G.H. Patel College of Engineering

And Technology

An

User Defined Project Report

On

Brain Computer Interfaced Autonomous Driving

B. E. IV, Semester – VIII

(Department of Computer Engineering)

Submitted By:

Sr. No.	**Name**	**Enrollment No.**
1	Shrinand Thakkar	130110107059
2	Virali Thakkar	130110107060

Prof. Hetal Gaudani

(Faculty Guide)

Academic Year: 2016-2017

SELF-DECLARATION

We,

Shrinand Thakkar (130110107059) and **Virali Thakkar (130110107060)**, the students of **Computer** Branch, enrolled at **G.H. Patel College of Engineering & Technology** hereby certify and declare the following:

1. We have defined our project based on our personal interests and both of us will make significant efforts to solve the challenges. We will attempt the project work at my college or at any location under the direct and consistent monitoring of our internal guide. We will adopt all ethical practices to share credit amongst all the contributors based on their contributions during the project work.

2. We have not purchased the solutions developed by any 3rd party directly and the efforts are made by us under the guidance of our guide.

3. The project work is not copied from any previously done projects directly. (Same project can be done in different ways but if it has been done in same manner before then it may not be accepted)

4. We understand and accept that he above declaration if found to be untrue, it can result in punishment/cancellation of project definition to me/we including failure in the subject of project work.

Name:	**Contact Number:**
Shrinand Thakkar	9099782615
Virali Thakkar	9825311535

Date: 12 October 2016

Place: Vallabh Vidyanagar

CERTIFICATE

This is to certify that the User Defined Project entitled **"Brain Computer Interfaced Autonomous Driving"** has been carried out by Shrinand Thakkar (130110107059) and Virali Thakkar (130110107060) under my guidance and supervision for the degree of Bachelor of Engineering in Computer Engineering (Semester - VIII) at G H Patel College of Engineering & Technology, Vallabh Vidyanagar during the academic year 2016-17.

Date:

Faculty Guide:	**Head of Department:**
Prof. Hetal Gaudani	Dr. Maulika S. Patel
Computer Engineering Department,	Computer Engineering Department,
GCET	GCET

ACKNOWLEDGEMENT

We take this momentous opportunity to express our heartfelt gratitude to our highly esteemed project guide Prof. Hetal Gaudani, Department of Computer Engineering, GCET for providing us an opportunity to present our project on ‘Brain Computer Interfaced Autonomous Driving’.

We attribute heartiest thanks to all the faculty of Computer Department for their valuable advice and encouragement.

Shrinand Thakkar

Virali Thakkar

Abstract

Self-Driving Cars! No steering wheel, no manual control, no troubles. Just like a human, these autonomous cars uses sensor obtained data to understand the situations around them, and a brain that receives, processes on information gathered from ultrasound or LIDAR.

Reinforcement learning algorithms have been applied in robotics to learn how to solve tasks based on reward signals obtained during task execution. In that, the basic mechanism involves providing a reward signal, reinforcement signal to the robot according to the actions performed.

But Wait. More than just that, Imagine what the outlook would be like if the human brain itself coordinates with that brain thing of the self-driving car.

With the inclusion of that, more than just being safe and reliable, for user acceptance, such vehicles will also provide a rich user experience. Here, learning using Brain Computer Interface (BCI) overpowers tedious and error prone manual tuning of human dependent parameters. Brain Computer Interface approach allows the user to just think about the desired style of driving, and the car will.

INDEX

TABLE OF FIGURES

1 Introduction

1.1 Problem Summary

The intended course of work is to review and analyze the current collaboration of the BCI and Reinforcement Learning. Reinforcement Learning and Deep Learning algorithms have been successfully applied in robotics to increase the productivity and the efficiency of the robots or the prostheses based on the reward signals obtained. These reward signals without getting modeled by the programmer could be directly provided from the electrical activity in Brain. The intent from the user's brain triggers a complex process in which certain brain areas are activated, and hence signals are sent to the corresponding muscles via the peripheral nervous system, which in turn perform the movement necessary for the communication. There is a protocol designed by I. Iturrate, L. Montesano and J. Minguez [4] to achieve this, which is thoroughly reviewed.

1.2 Aim and Objectives of the Project

The objective is to study a developed approach which computes the rewards for the given task directly from brain activity recorded with the help of a non-invasive Brain-Computer Interface (BCI). This could in lead to robotic systems such as prostheses or bots that operate close to the human and self-adaptive in nature or can adapt themselves to new tasks.

1.3 Problem Specifications

Learning different Algorithms based on Reinforcement Learning and Deep Learning, and using the same to correctly interpret the electric signals from the brain and use the framework as a whole for decision making.

1.4 Literature Review

1.4.1 BCI (Brain Computer Interface)

A brain–computer interface (BCI), which is also called mind-machine interface (MMI), direct neural interface (DNI), or brain–machine interface (BMI), is a direct means of communication between the brain and the external device receiving the signals from the brain. A brain-computer interface (BCI) is a system where only one's brainwaves are used to control a computer or other electronic device. There are many fields where these BCIs can be used, like mapping, researching, assisting something or someone, augmenting, or restoring human cognitive or sensory-motor functions. BCIs can also be used for controlling robotic devices like wheelchair or prostheses, among other applications.

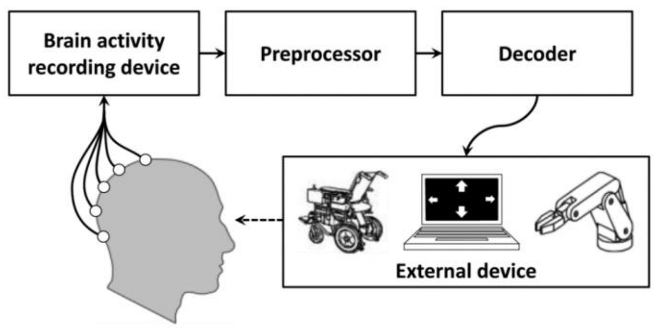


Fig. 1.4.1.1

Anupama. H. S, N. K. Cauvery, Lingaraju G. M [5], mentioned in their review study that the major goal of BCI research is to unfold a system that allows specially abled people to communicate with others and helps in interacting with the external world. They basically progressed as including components like, comparing the invasive and noninvasive methodologies to measure the brain activity, assessment of control signals (i.e. patterns of brain generated signals that can be used for interaction), development of algorithms for generating useful information from brain signals, and the development of new BCI applications. Jonathan R. Wolpaw, Niels

Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, Theresa M. Vaughan [6] also talks about the same, A BCI allows a person to interact with or control the external world without using the brain's normal output tracks of peripheral nerves and muscles. Messages and commands are expressed by electrophysiological phenomena such as evoked or spontaneous EEG features and not by muscle contractions.

Certainly, Rabie A. Ramadan etal [7] stated that, there are many challenges that the BCI faces when used in real world tasks as follows:

(1) **Low BCI signal strength**: Signal amplification is often required as it has been noticed that signals extracted from the brain has the signal strength in most of the cases low. Many of the toolkits or packages include amplifiers for the same.

(2) **Data transfer rate (bandwidth)**: the bandwidth from most of the subjects at max was 3 characters. And due to this, the BCI applications suffer from fast responses.

(3) **High error rate**: The above two mentioned in whole increases the error percentage. In addition to that, the brain signal is highly variable. Therefore, the error rate is in most cases high.

(4) **Inaccurate signal classification**: Classifying the captured signals from the brain suffer from high interference and inaccurate classification. There are many signal classification techniques proposed and used including the computational intelligence techniques.

1.4.2 MDP (Markov Decision Process)

Markov decision processes (**MDPs**) bestow a mathematical framework for decision making in situations where the possibilities are partially under the control of a decision maker and partially random. For studying wide range of optimization problems solved via reinforcement learning or dynamic programming, MDPs are useful. Robotics, economics, automated control, and manufacturing are some area of disciplines of MDPs.

More precisely, a Markov Decision Process is a discrete time stochastic control process. At each time step, the process is in some state, and the decision maker may choose any action from the available ones in state. The process reacts at the next time step by randomly moving into a new state, and giving a corresponding reward to the decision.

A Markov Decision Process (MDP) model contains:

- A set of states S
- A set of actions A
- A real valued reward function R(S), R(S,a), R(S,a, S')
- A description T of each action's effects in each state.

Markov Property: the effects of an action taken in a state depend only on that state and not on anything from the history.

MDP's can be further classified into,
Centralized and **Decentralized** Models, where the Decentralized models are one where the decision-making authority is distributed throughout a larger group.

Among the centralized ones, partially observable Markov decision process (**POMDP**), is a generalization of MDP which allows uncertainty regarding the state of a Markov Process and allows for state information acquisition.

George E. Monahan [8] surveyed models and algorithms dealing with the POMDP. Applications of whose as he mentioned were, machine maintenance, quality control, learning theory, etc.

Furthermore in the Decentralized Models,

One is a generalization of a partially observable Markov decision process (POMDP), which is called a decentralized partially observable Markov decision process (**DEC-POMDP**). In a DEC-POMDP, the process is controlled by multiple distributed agents, each with possibly different information about the state.

The other is a generalization of an MDP, called a decentralized Markov decision process (**DEC-MDP**). A DEC-MDP is a subset of DEC-POMDP with the restriction that at each time step the agents' observations together uniquely determine the state.

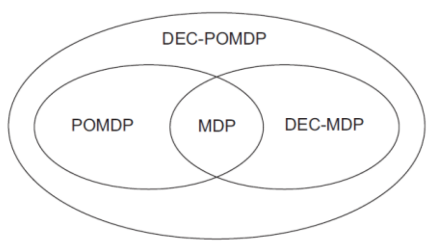


Fig. 1.4.2.1

Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein [9] considered decentralized control of Markov decision processes and gave complexity bounds on the worst-case running time for algorithms that find the most optimal solutions. Generalizations of both the fully observable case and the partially observable case that allow for decentralized control are described. Several types of networking problems can be viewed within this framework (Altman 2001) [11] including the distributed control of a power grid (Schneider et al. 1999) [10].

1.5 Plan of the Work

Study different Reinforcement and Deep Learning Algorithms

↓

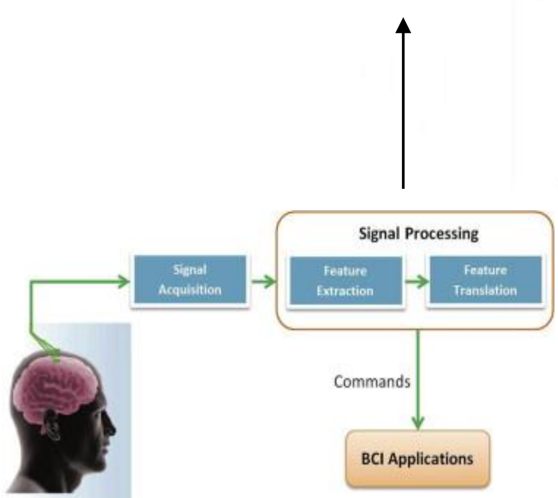
Extracting correctly the data from the electric signals of the brain

↓

Using those Algorithms on the brain signals to use them as a means of communication

↓

Using the Framework for Decision Making without any user modelling



1.6 Materials / Tools Required

GNU Octave GNU Octave is a high-level programming language software, primarily intended for numerical computations. Numerically, linear and nonlinear problems are solved by Octave along with performing other numerical experiments using MATLAB compatible languages.

Non-invasive BCI Non-invasive BCI are based on electroencephalogram (EEG) signals. Electrodes placed on the client's head helps recording the EEG. This technology is not invasive and only records the electrical activity of the brain without getting along with it.

Algorithms for control learning Algorithms like **Criterion of optimality**, **Monte Carlo methods, Value function approaches**, **Temporal difference methods**, **Direct policy search** will be used by the system.

Q-learning it is a model-free reinforcement learning technique. Specifically, to find an optimal policy in selecting an action for any given (finite) Markov decision process (MDP), *Q*-learning can be used. It works by learning an action-value function that ultimately gives the expected utility of taking a given action in a given state and following that favorable policy afterward.

Arduino, Used this board to make an **autonomous vehicle** with an assortment of **electronic sensors** (HC-SR04) and components like **Wi-Fi module** (ESP8266) and **Servo Motors**, which pulls together the knowledge it had learned and synthesized from many sources along the way.

2 Design: Analysis, Design Methodology and Implementation Strategy

- **Empathy Canvas**

Users

- Military, Ambient Intelligence
- Differently abled, as in communication, interaction.
- Scientists, for behavioral research and development.
- Manufacturing Industry, Production Control

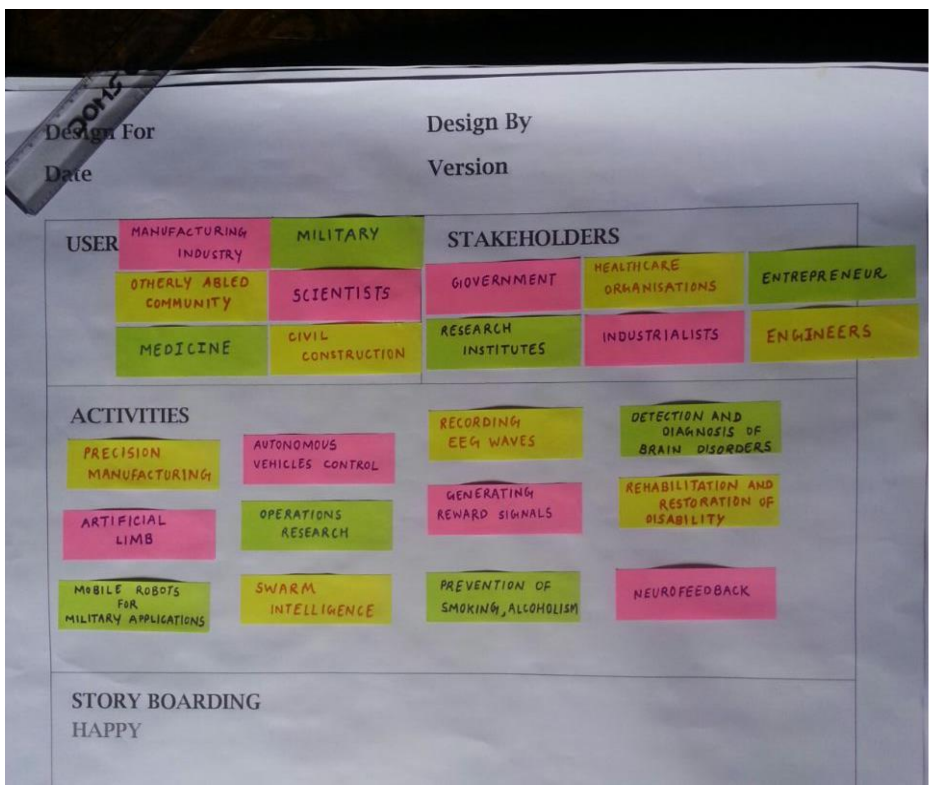


Fig. 2.1

Activities

- Autonomous Vehicle Control
- Recording EEG signals
- Generating reward signals
- Swarm Intelligence
- Rehabilitation and Restoration of Disability
- Artificial Limbs
- Mobile Robots for Military Applications

Stake Holders

- Government
- Research Institutes
- Industrialists
- Healthcare Industry
- Engineers

- <u>**Ideation Canvas**</u>

People

- Military, Ambient Intelligence
- Differently abled, as in communication, interaction.
- Scientists, for behavioral research and development.
- Manufacturing Industry, Production Control

Activities

- Autonomous Vehicle Control
- Recording EEG signals
- Generating reward signals
- Swarm Intelligence
- Rehabilitation and Restoration of Disability
- Artificial Limbs
- Mobile Robots for Military Applications

Fig. 2.2

Props

- Electroencephalogram (**EEG**), measures electrical activity in brain.
- Electrodes, a conductor to measure the electrical activity in brain.
- Reward Function, develops reward signals and use them for influencing brain activity.
- Datasets
- RL Algorithms
- Machine Learning, is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence.

Situation/Context/Location

- Defence, military applications
- Hospitals, medicinal uses
- Game development
- Robotics
- R&D labs
- Manufacturing Industries

- <u>**Product Development Canvas**</u>

Purpose

- Recording brain activity using a non-invasive brain computer interface.
- To develop robotic system such as prostheses, adaptive in nature.

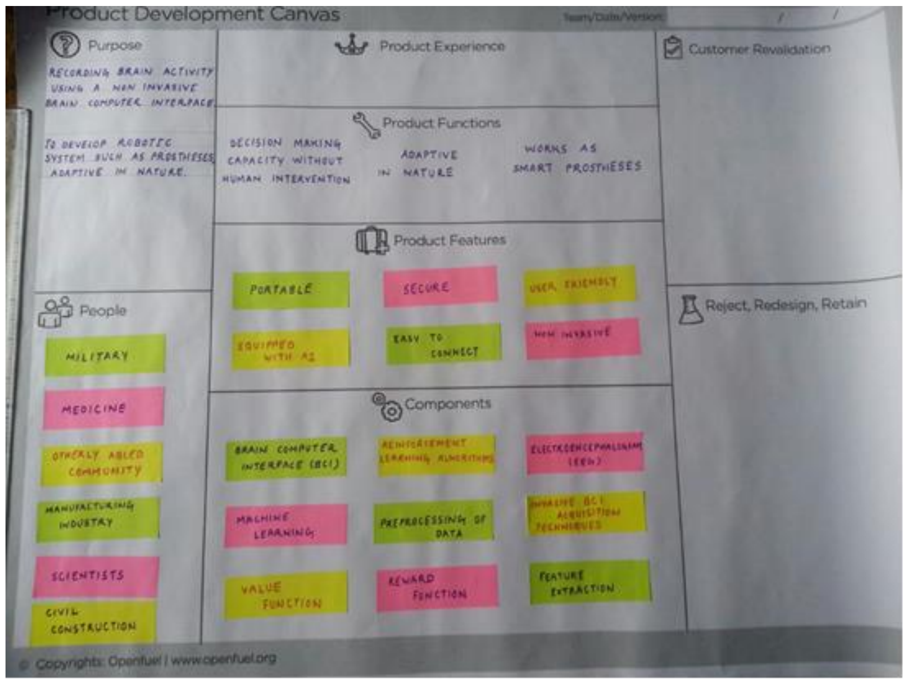


Fig. 2.3

People

- Military, Ambient Intelligence
- Differently abled, as in communication, interaction.
- Scientists, for behavioral research and development.
- Manufacturing Industry, Production Control

Product Functions

- Decision making capacity without human intervention
- Adaptive in nature
- Work as smart prostheses

Product Features

- Portable
- Secure
- Equipped with Artificial Intelligence
- User Friendly

Components

- Brain Computer Interface, is a direct communication pathway between an enhanced or wired brain and an external device.
- Electroencephalogram (**EEG**), measures electrical activity in brain.
- Machine Learning, is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence.
- Reward Function, develops reward signals and use them for influencing brain activity.

- **<u>AEIOU Summary Canvas</u>**

Environment

- Defence, military applications
- Hospitals, medicinal uses
- Outer Space
- Robotics
- R&D labs
- Manufacturing Industries

Objects

- Electroencephalogram (**EEG**), measures electrical activity in brain.
- Electrodes, a conductor to measure the electrical activity in brain.
- Reward Function, develops reward signals and use them for influencing brain activity.
- Datasets
- RL Algorithms
- Machine Learning, is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence.

Fig. 2.4

Activities

- Autonomous Vehicle Control
- Recording EEG signals
- Generating reward signals
- Swarm Intelligence
- Rehabilitation and Restoration of Disability
- Artificial Limbs
- Mobile Robots for Military Applications

Users

- Military, Ambient Intelligence
- Differently abled, as in communication, interaction.
- Scientists, for behavioral research and development.
- Manufacturing Industry, Production Control

- **<u>Business Model Canvas</u>**

A Business Model describes the rationale how an organization creates, delivers, and captures value. It is simply a business model describes how a company creates an offering, get it to customers and generates profit from the transaction.

A company's business model canvas can be classified into nine elements:

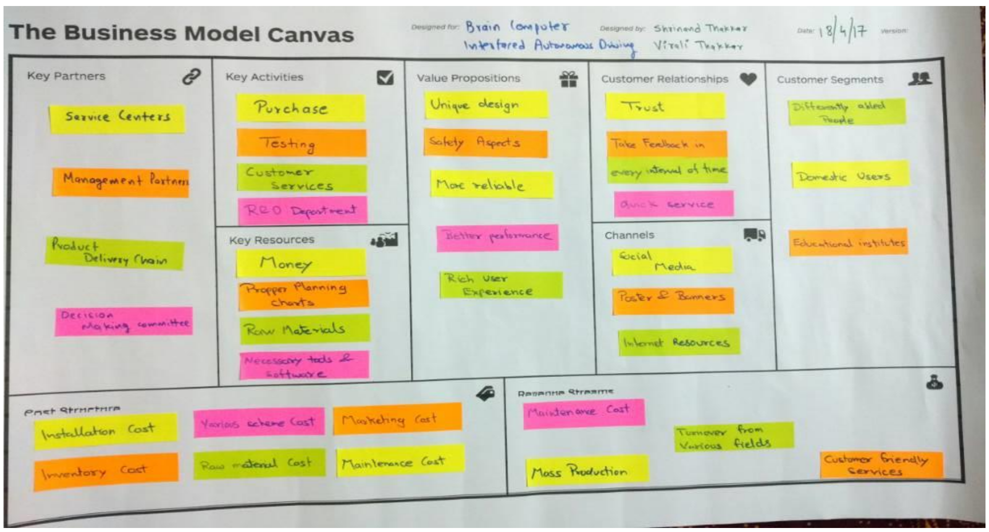


Fig. 2.5

3 Implementation

3.1 Electroencephalography (EEG)

About

It is an electrophysiological monitoring approach to document electrical activity of the brain. It is commonly noninvasive, with the electrodes positioned along the scalp, and invasive electrodes are now and again used in specific applications.

EEG signals basically measures voltage fluctuations resulting from ionic current in the neurons of the brain. EEG is most frequently used to diagnose epilepsy, causing abnormalities in EEG readings. It is also used to diagnose sleep disorders, coma, encephalopathy. EEG used to be a first-line approach of diagnosis for tumors, stroke and different focal talent disorders.

Derivatives of the EEG technique consist of evoked potentials (EP), which entails averaging the EEG activity time-locked to the presentation of a stimulus of some kind (visual, somatosensory, or auditory).

Biopac

BIOPAC is a tool which provides a wide range of functionalities for recording, displaying, and analyzing surface EEG as well as implanted EEG signals from human and animal subjects.

Number of hardware solutions are provided by the BIOPAC that allow us to record from a single channel of EEG and up to 32 channels of wireless and logged data. There are also hardware options available for full-band EEG recordings with bandwidths from DC to hundreds of Hz.

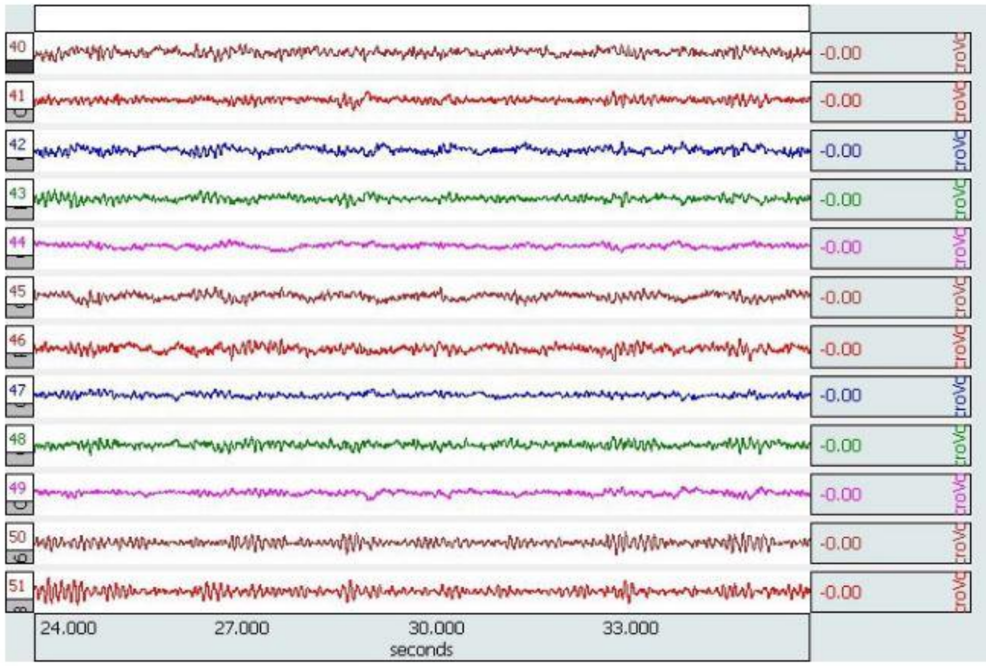


Fig. 3.1.1

Automated EEG Analysis

Compute Approximate Entropy

We can compute *Approximate Entropy* by using statistical measure that attempts to quantify the predictability of a data sequence. A pure sine wave which is perfectly predictable data series has an approximate entropy of zero. Several studies are examining approximate entropy of EEG data and its relationship to external factors such as drugs and sleep states.

Derive Alpha RMS

A standard alpha RMS waveform can be constructed by the Derive Alpha RMS script from an alpha EEG signal. It is the windowed root mean square (rms) value of the signal using a window width of 0.25 seconds.

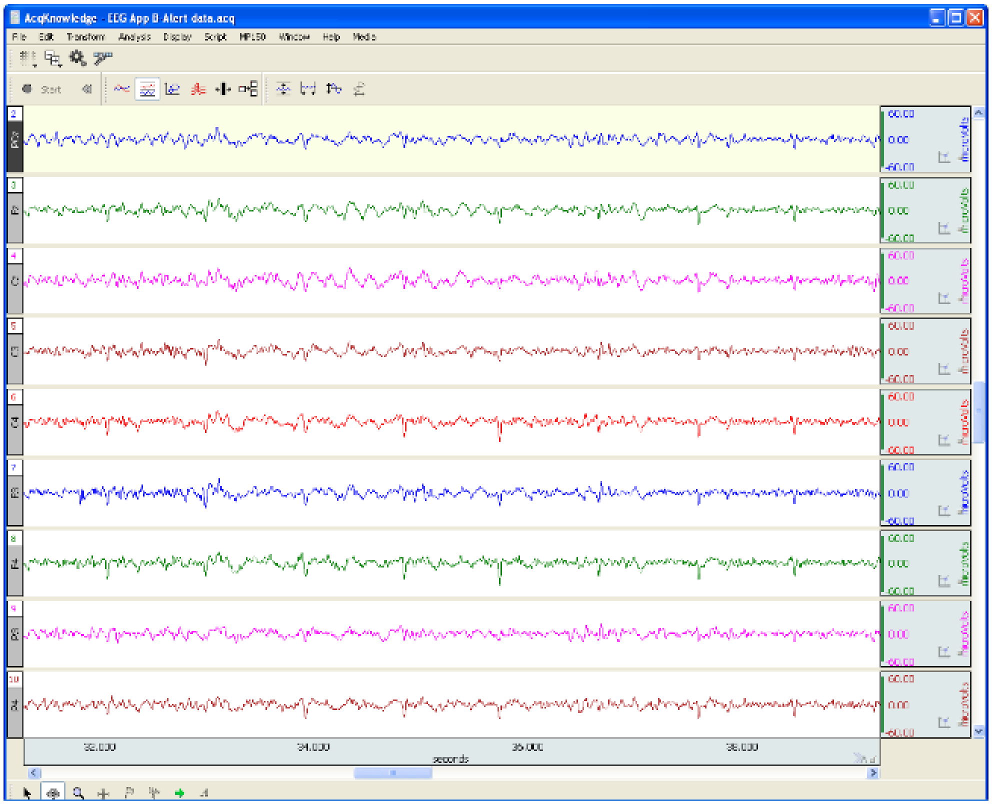


Fig. 3.1.2

Derive EEG Frequency Bands

The BIOPAC provides functionality for automatic filtering of the raw EEG signal for the appropriate band widths. The different bands available are following: Alpha, Beta, Theta, Delta, and Gamma

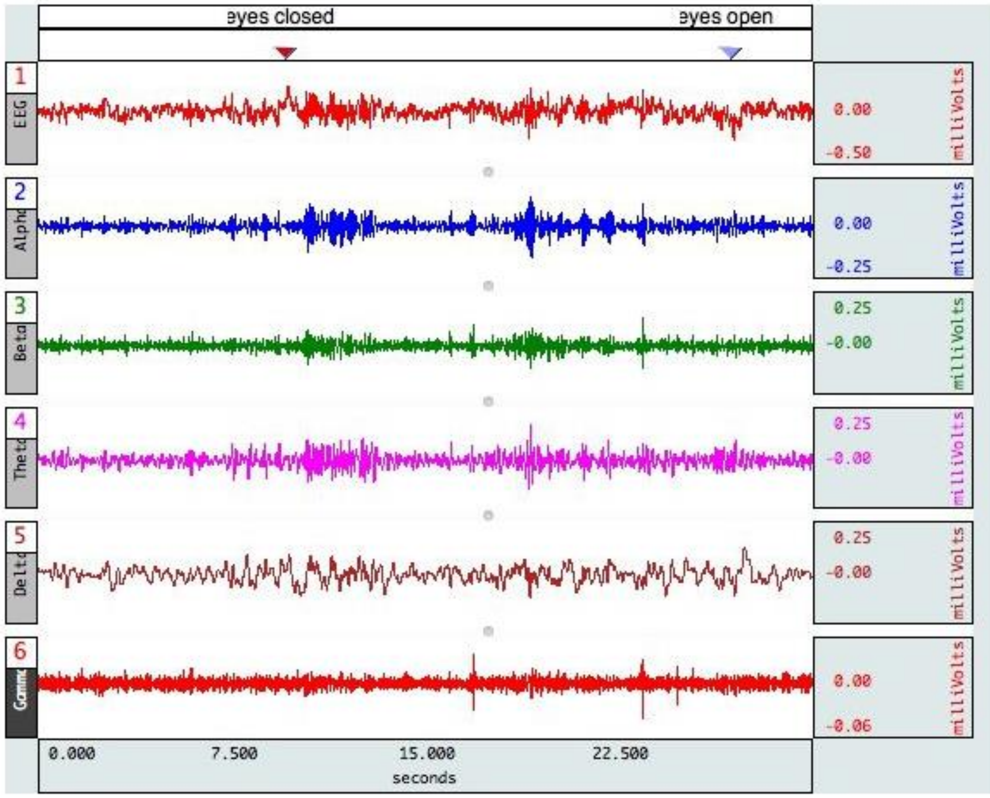


Fig. 3.1.3

EEG Frequency Analysis

The EEG signals are divided into fixed-width time epochs by the EEG Frequency Analysis script. For each individual time epoch, the power spectrum of that epoch is estimated using a Welch periodogram estimation method by the Power Spectral Density function of Acq*Knowledge*. The following measures are extracted for each epoch from this PSD: Mean Power, Mean Frequency, Median Frequency, Peak Frequency and Spectral Edge.

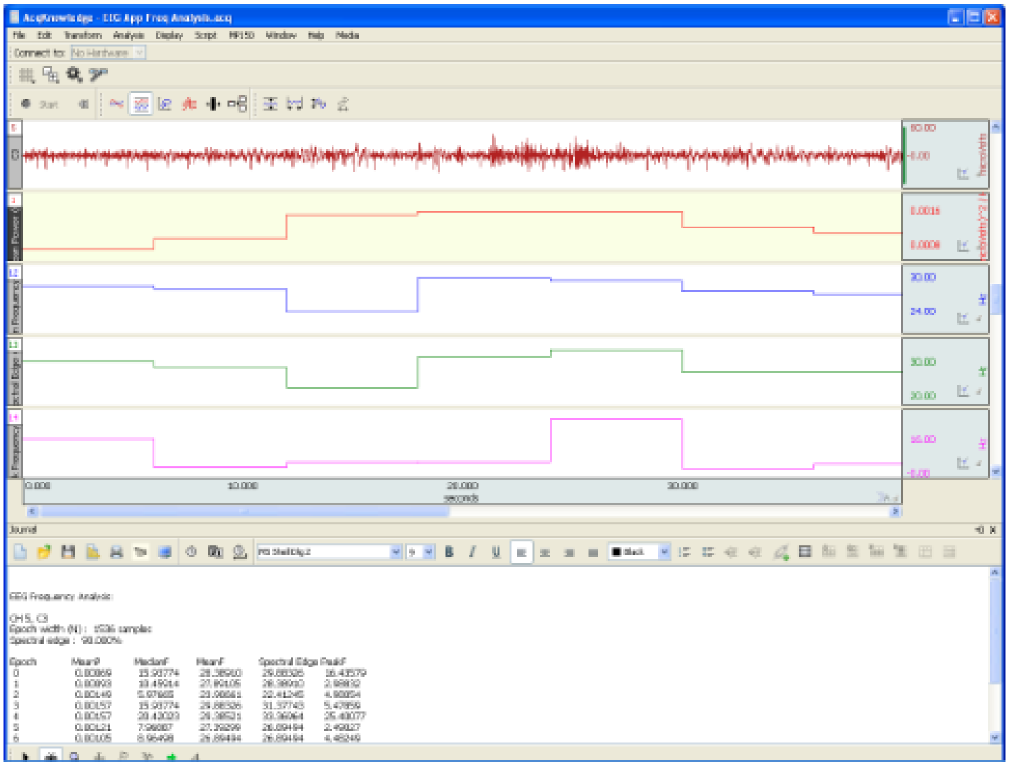


Fig. 3.1.4

EEG remove EOG noise

An epoch driven, fully automated, analysis of the EEG signal is provided by the EEG frequency analysis. Using EOG removal utility it automatically removes eye blink artifacts from the EEG signal. The technique which analysis uses to separate the EOG signal from the EEG waveform is an independent component analysis technique.

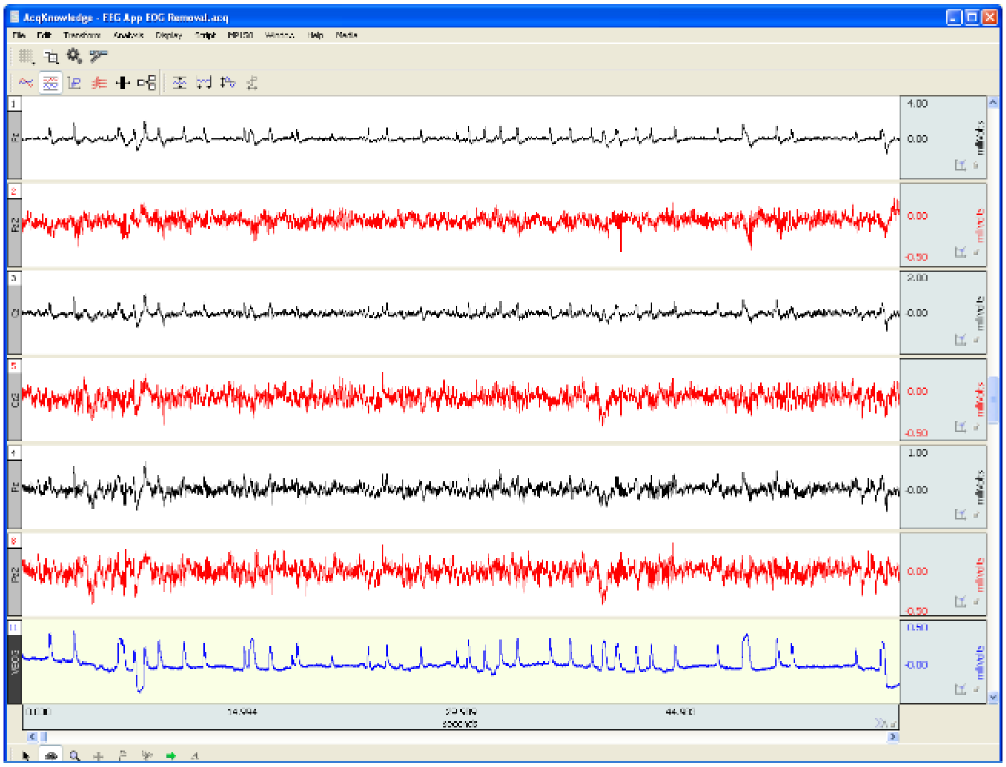


Fig. 3.1.5

3.2 SVM (Support Vector Machines)

The SVM is yet another type of ***supervised*** machine learning algorithm which can generate optimal hyperplane for linearly separable patterns. We are given training data $\{x_1 \ldots x_n\}$ that are vectors in some space $X \subseteq \mathbb{R}^d$. We are also given their corresponding output values $\{y_1 - y_n\}$ where y_i can be either 1 or -1. In their simplest form, SVMs are hyperplanes that separate the training data by a margin with a maximum distance between the nearest data and the plane. All vectors lying on one side of the hyperplane have their corresponding y values as -1, or negative, and all vectors lying on the other side have their corresponding y values as 1, or positive. The training data values that lie closest to the hyperplane are called **support vectors**.

The cost function for support vector machines,

$$J(\theta) = C \sum_{i=1}^{m} y^{(i)} cost_1\left(\theta^T x^{(i)}\right) + \left(1 - y^{(i)}\right) cost_0\left(\theta^T x^{(i)}\right) + \frac{1}{2}\sum_{j=1}^{m} \theta_j^2$$

$$cost_0(z) = max(0, k(1 + z))$$

$$cost_1(z) = max(0, k(1 - z))$$

Here, when we wish to regularize more (which basically is, reducing the overfitting) or we *decrease* the value of C, and when we wish to regularize less (which basically is, reducing the underfitting), we *increase* the value of C.

The hypothesis of the Support Vector Machine outputs either 1 or 0, as

$$h_\theta(x) = \begin{cases} 1 & if\ \theta^T x \geq 0 \\ 0 & otherwise \end{cases}$$

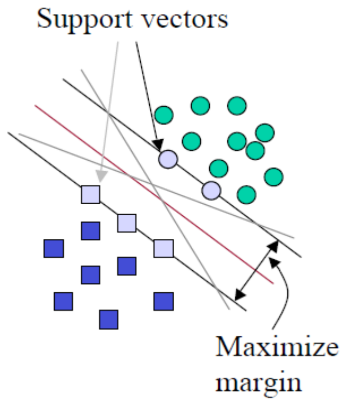


Fig 3.2.1

SVM can also extend to patterns that are not linearly separable by transformations of original data to map into new space, the **Kernel function**. It performs a transformation to a **new feature space** where the data point has *N* features, one for each support vector. The value of the i^{th} feature is equal to the value of the kernel between the i^{th} support vector and the data point being classified. In this space, the original (possibly non-linear) SVM classifier is just like any other linear one. After the transformation, the original features of the data point are irrelevant. It is represented only in terms of its **dot products with support vectors** (which are basically special data points chosen by the SVM optimization algorithm). For Example,

Given x, compute new feature depending on proximity to landmarks $l^{(1)}, l^{(2)}, l^{(3)}$.

$$f_i = similarity(x, l^{(i)}) = exp(-\frac{||x - l^{(i)}||^2}{2\sigma^2})$$

This "similarity" function is called a **Gaussian Kernel**. It is a specific example of a kernel.

Computing this gives us a "feature vector," $f_{(i)}$ of all our features.

Thus given training example $x_{(i)}$:

$$x^{(i)} \rightarrow \begin{bmatrix} f_1^{(i)} = similarity(x^{(i)}, l^{(1)}) \\ f_2^{(i)} = similarity\left(x^{(i)}, l^{(2)}\right) \\ \cdot \\ \cdot \\ \cdot \\ f_m^{(i)} = similarity(x^{(i)}, l^{(m)}) \end{bmatrix}$$

Now to get the parameters θ we can use the SVM minimization algorithm which will minimize the cost function but with $f^{(i)}$substituted in for $x^{(i)}$.

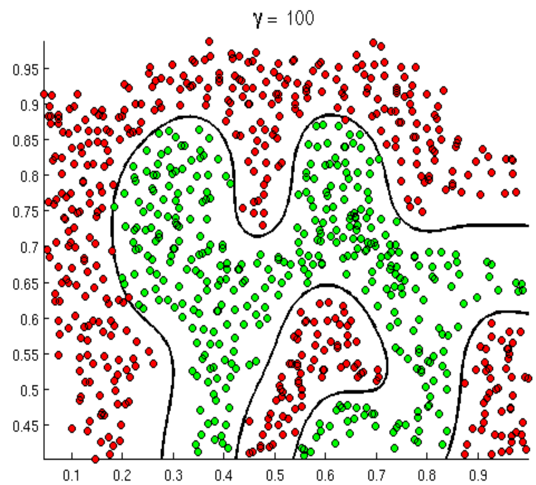


Fig. 3.2.2

3.3 Arduino Powered Car

It will work as an autonomous vehicle with an assortment of **electronic sensors** (HC-SR04) and components like **Wi-Fi module** (ESP8266) and **Servo Motors**, which pulls together the knowledge it had learned and synthesized from many sources along the way.

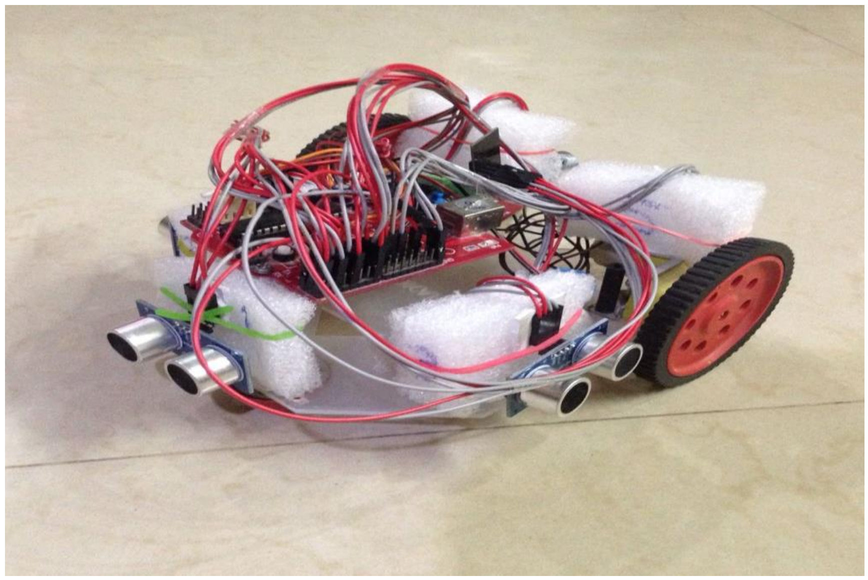


Fig. 3.2.3

3.4 Q-learning Algorithm

Q-learning is basically a model-free reinforcement learning technique. It works by learning an action-value function that ultimately gives the expected utility of taking a given action in a given state and following that favourable policy afterward.

We first trained the car in an virtual environment for every possible permutation of obstacles which could hinder the car in its way to reach the destination using Q-learning. Trained data is uploaded on a local server which is easily accessible by the Wi-Fi module embedded on the Arduino based car.

4 Conclusion and Future Work

Conclusion

We reviewed several papers regarding the reinforcement learning, used to improve the robustness of the robots and BCI. Learned about the preprocessing of the EEG (Electroencephalogram) signals which are to be provided as input to the system. Then extracting features from those signals and SVM (Support Vector Machine) to get better intuition of the intended methodologies.

Future Work

We look forward to develop algorithms which can extract feature based on the synaptic excitations of dendrites of many pyramidal neurons in the cerebral cortex and train them to make decision based on those excitations. This Framework then could be used in many field such as a mechanism to control prostheses for the specially abled client, accessing the connected devices by just a thought like, flying drones, etc.

5 References

1 Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra and Martin Riedmiller, "*Playing Atari with Deep Reinforcement Learning*", 2013

2 Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Adrià Puigdomènech Badia, Mehdi Mirza, Tim Harley and Timothy P. Lillicrap, "*Asynchronous Methods for Deep Reinforcement Learning*", 2016

3 Mehdi Bennis, Samir M. Perlaza, Pol Blasco, Zhu Han, and H. Vincent Poor, " *Self-Organization in Small Cell Networks: A Reinforcement Learning Approach*", 2013

4 I. Iturrate, L. Montesano and J. Minguez, "*Robot Reinforcement Learning using EEG-based reward signals*", 2010

5 Anupama.H.S, N.K.Cauvery, Lingaraju.G.M; "Brain Computer Interface and Its Types - A Study "International Journal of Advances in Engineering & Technology, May 2012. ©IJAET ISSN: 2231-1963

6 Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, Theresa M. Vaughan; Invited review "Brain–computer interfaces for communication and control"

7 Rabie A. Ramadan, S. Refat, Marwa A. Elshahed and Rasha A. Ali; "Basics of Brain Computer Interface"

8 George E. Monahan; "A survey of partially observable Markov Decision Processes: Theory, Models, and Algorithms "

9 Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein; "The Complexity of Decentralized Control of Markov Decision Processes"

10 Schneider, J., W.-K. Wong, A. Moore, M. Riedmiller. Distributed value functions. *Proc. Sixteenth Internat. Conf. on Machine Learning*.

11 Altman, E. 2001. Applications of Markov decision processes in telecommunications: A survey. E. Feinberg, A. Shwartz, ed. *Markov Decision Processes: Models, Methods, Directions, and Open Problems*. Kluwer, New York.