# A cell-phone-based brain-computer interface for communication in daily life

**Yu-Te Wang, Yijun Wang and Tzyy-Ping Jung**

**Journal of Neural Engineering (2011.03)**

**Presenter : Soogil Woo**

GIST, Dept. of Information and Communication, INFONET Lab.

Gwangju Institute of Science and Technology



# Introduction

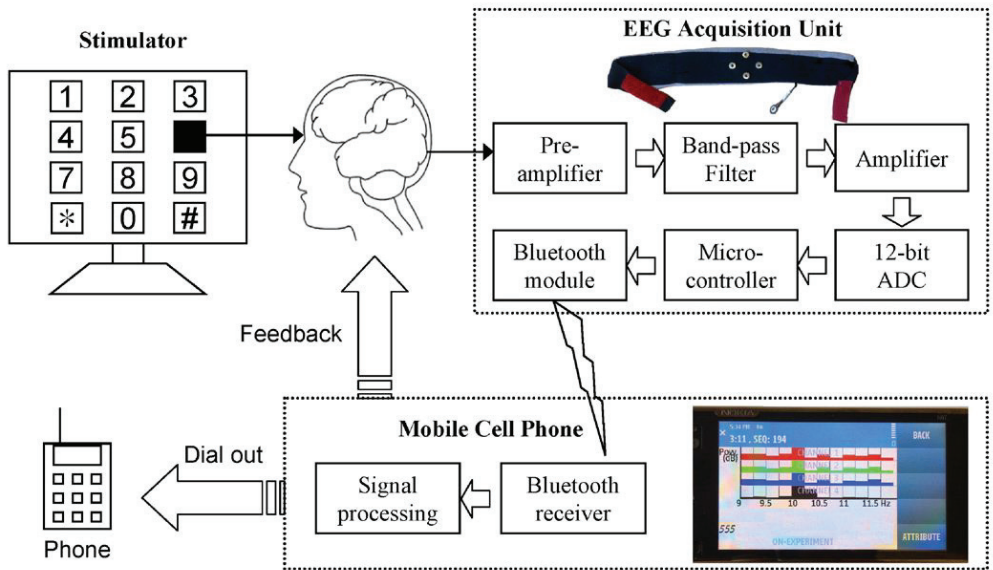
- BCI systems acquire EEG signals from the human brain and translate them into digital commands which can be recognized and processed on a computer.
- Although EEG-based BCIs have already been studied for several decades, moving a BCI system from a laboratory demonstration to real-life application still poses severe challenges to the BCI community.
- In real-life applications, BCI systems should not use bulky, expensive, wired EEG acquisition devices and signal processing platforms.
- Several studies have demonstrated the use of portable devices for BCIs.
- Recently, with advances in integrated circuit technology, cellphones combined with DSP and built-in Bluetooth function have become very popular in the consumer market.

# Introduction

- If a cell-phone-based BCI proves to be feasible, many current BCI demonstrations (gaming, text messaging) can be realized on cell phones in practice and numerous new applications might emerge.
- This system consists of a four-channel bio-signal acquisition /amplification module, a wireless transmission module and a Bluetooth-enabled cell phone.
- SSVEP is the electrical response of the brain to the flickering visual at a repetition rate higher than 6Hz.
- SSVEP-based BCI, which has recognized advantages of ease of use, little user training and high information transfer rate (ITR), was employed as a test paradigm.
- In an SSVEP BCI, the attended frequency-coded targets of the user are recognized by detecting the dominant frequency of the SSVEP.
- This study implemented and tested both single-channel FFT and multi-channel canonical correlation analysis (CCA) methods for processing SSVEPs induced by attended targets.

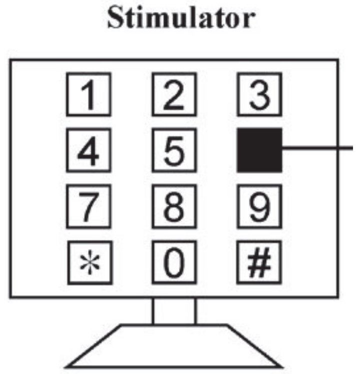


# System Architecture




- System hardware diagram
- System software design
- BCI experiment design

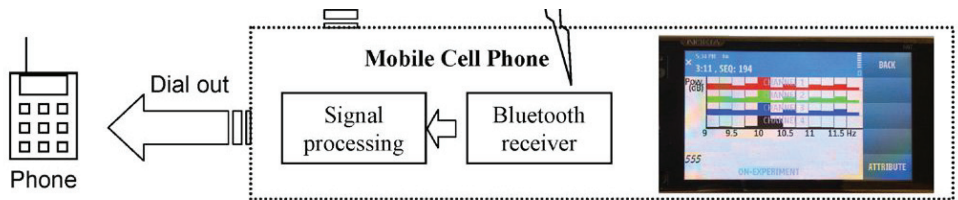
# System hardware diagram




- The visual stimulator comprises a 21 inch CRT monitor with 4*3 stimulus matrix constituting a virtual telephone keypad which includes digits 0-9, BACKSPACE and ENTER.
- The stimulus frequencies ranged from 9 to 11.75 Hz with an interval of 0.25Hz between two consecutive digits.
- The stimulus program was developed in Microsoft Visual C++ using the Microsoft DirectX 9.0 framework.



# System software design




- The signal-processing unit was realized using a Nokia N97 cell phone.
- A J2ME program developed in BorlandJBuilder2005 and Wireless Development Kit 2.2 were installed to perform online procedures including :(1) displaying EEG signals in time-domain, frequency-domain and CCA-domain on the LCD screen of the cell phone, (2) band-pass filtering, (3) estimating the dominant frequencies of the VEP using FFT or CCA, (4) delivering auditory feedback to the user, (5) dialing a phone call
- Users can choose the format of the display btw time-domain and frequency-domain.
- Under the frequency-domain display mode, the power spectral densities of 40channel EEG will be plotted on the screen and updated every second.

# System software design

- In the FFT mode, a 512 point FFT is applied to the EEG data using a 4 sec moving window advancing at 1 sec steps for each channel.
- In the CCA mode, it uses all four channels of the EEG with a 2 sec moving window advancing 1 sec steps.
- To improve the reliability, a target is detected only when the same dominant frequency is detected in two consecutive windows.
- At the k and k+1 sec, k≥4 in the FFT mode
- At the k and k+1 sec, k≥2 in the CCA mode
- The subjects were instructed to shift their gaze to the next target once they heard the auditory feedback.



# System BCI experiment design

- The experiments were conducted in a typical office room without room without any electromagnetic shielding.
- Subjects were seated in a comfortable chair at a distance of about 60cm from screen.
- Four electrodes on the EEG headband were placed 2cm apart, surrounding a midline occipital (Oz) site, all referred to a forehead midline electrode.
- The FFT- and CCA-based approaches were tested separately.
- All subjects participated in the experiments during which the cell phone used FFT to detect frequencies of SSVEPs, and four subjects were selected to do a comparison study btw using FFT and CCA for SSVEP detection.
- In the FFT mode, the channel with highest SNR, which is based on the power spectra of the EEG data, was selected for online target detection.
- The EEGs in the CCA experiments were feedback codes for an offline comparison study btw FFT and CCA.

# Results

- Tables show results of the SSVEP BCI using FFT and CCA.

| Subject | Input length | Time (s) | Accuracy (%) | ITR (bits $min^{-1}$) |
|---|---|---|---|---|
| s1 | 11 | 72 | 100 | 32.86 |
| s2 | 11 | 72 | 100 | 32.86 |
| s3 | 19 | 164 | 78.9 | 14.67 |
| s4 | 11 | 73 | 100 | 32.4 |
| s5 | 17 | 131 | 82.4 | 17.6 |
| s6 | 11 | 67 | 100 | 35.31 |
| s7 | 11 | 72 | 100 | 32.86 |
| s8 | 13 | 93 | 92.3 | 20.41 |
| s9 | 11 | 79 | 100 | 29.95 |
| s10 | 11 | 66 | 100 | 35.85 |
| Mean | 12.6 | 88.9 | 95.9 | 28.47 |

FFT-based online test results of the SSVEP BCI in ten subjects



# Results

- Tables show results of the SSVEP BCI using FFT and CCA.

| Subject | Online CCA | Online FFT | Offline FFT | Putative ITR from offline FFT | | | |
|---|---|---|---|---|---|---|---|
| | | | | Ch1 | Ch2 | Ch3 | Ch4 |
| s1 | 44.79 | 32.86 | 36.68 | **36.68** | 33.58 | 32.48 | 29.77 |
| s2 | 46.25 | 32.86 | 26.49 | **26.49** | 10.51 | 5.91 | 9.29 |
| s6 | 49.05 | 35.31 | 19.43 | **19.43** | 3.03 | 3.15 | 1.92 |
| s10 | 43.18 | 35.85 | 15.24 | 2.2 | 8.46 | **15.24** | 4.21 |
| Mean | 45.82 | 34.22 | 24.46 | 21.2 | 13.9 | 14.2 | 11.3 |

CCA-based test results (ITR) of the SSVEP BCI in four subjects. In each row, the bold value highlights the maximum ITR of single channel FFT.

# Discussions and conclusions

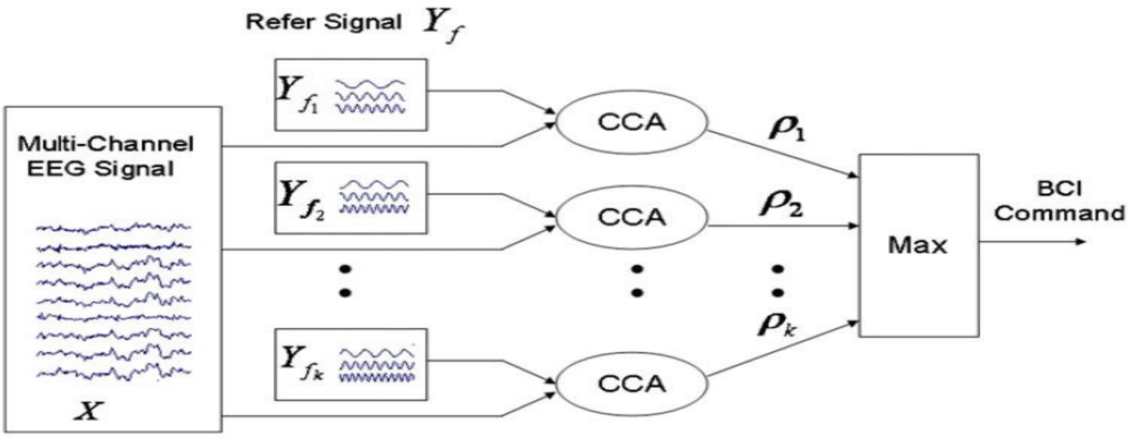
- This study designed, developed and evaluated a portable, cost effective and miniature cell-phone-based online BCI platform for communication in daily life.
- The practicality and implications of the proposed BCI platform were demonstrated through the high accuracy and ITR of an online SSVEP-based BCI.
- The decline in accuracy and ITR in offline FFT analysis could be attributed to a lack of sufficient data for FFT to obtain accurate results.
- In other words, FFT, in general, required more data than CCA to accurately estimate the dominant frequencies in SSVEPs.



Thank you!

# Appendix(CCA)

- Canonical correlation analysis (CCA) is a multivariable used when there are two sets of data, which may have some underlying correlation.
- It finds a pair of linear combinations, for two sets, such that the correlation btw the two canonical variables is maximized.
- Consider two multidimensional random variables X, Y and their linear combinations $x = X^T W_x$ and $y = Y^T W_y$ .




An illustration for usage of CCA in EEG signals analysis



# Appendix(CCA)

- CCA finds the weight vectors, $W_x$ and $W_y$, which maximize the correlation btw x and y, by solving the following problem:

$$\max_{W_x, W_y} \rho(x, y) = \frac{E[x^T y]}{\sqrt{E[x^T x]E[y^T y]}}$$
$$= \frac{E[W_x^T X Y^T W_y]}{\sqrt{E[W_x^T X X^T W_x]E[W_y^T Y Y^T W_y]}}$$

- The maximum of ρ with respect to $W_x$ and $W_y$ is the maximum canonical correlation. Projections onto $W_x$ and $W_y$, i.e. x and y, are called canonical variants.

Supervised Machine Learning: A Review of Classification Techniques
Kotsiantis S.B.

**Informatica (2007)**

**Presenter : Evgenii Kim**

GIST, Dept. of Information and Communication

Gwangju Institute of Science and Technology



# Outline

- Introduction
- General issues of supervised learning algorithms
- Logic based algorithms
- Perceptron-based techniques
- Instance-based learning
- Support Vector Machines
- Discussion
- Conclusion

# Introduction

- Machine learning (ML) is about the construction and study of systems that can learn from data.
- There are three types:
  - Supervised
  - Unsupervised
  - Reinforcement (The training information provided to the learning system by the environment)
- Numerous ML can be set up as supervised



# General issues of supervised ML

- There are several main steps:
  1. Collecting the dataset. In case of unknowing required feature we are measuring everything available in the hope that it is right. But in most of cases it contains noise and missing feature values, therefore the pre-processing is significant.
  2. Data pre-processiong. The goal is to maintain the mining quality while minimizing the sample size.
  3. Feature subset selection (feature extraction). It is a process of identifying and removing as many irrelevant and redundant features as possible. This reduces the dimensionality of the data
  4. Classification

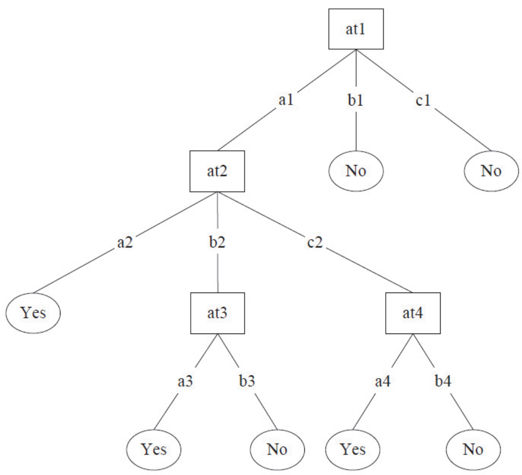
# Algorithm selection

- The choice of which specific learning algorithm we should use is a critical step. The classifier's evaluation is most often based on prediction accuracy.
- There are at least three techniques:
  - Split training set by two-thirds for training and the other third for estimate accuracy.
  - Cross-validation
  - Leave-one-out



# Decision trees

- Decision trees are trees that classify instances by sorting them based on feature values.
- Each node in the tree represents a feature in an instance to be classified
- Each branch represents a value that the node can be assume
- The feature that best divides the training data would be the root node of the tree.
- This method is quite sensitive to overfitting
- The most straightforward way of tackling overfitting is to prune the decision tree.
- One of the famous algorithm to generate a decision tree is C4.5



A decision tree

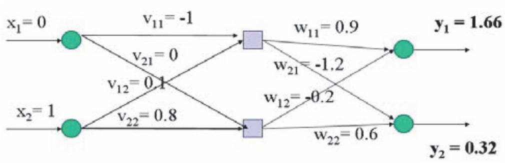
# Single layered perceptrons

- Input feature values $x_1,\ldots,x_n$
- Connection weights/predication vector (typically real numbers in the interval [-1,1]) $w_1,\ldots,w_n$
- The single layered perceptron computes the sum of weighted inputs: $\sum_i x_i w_i$ and output goes through an adjustable threshold: if the sum is above threshold, output is 1; else it is 0.
- WINNOW (1994) is based on the perceptron idea and updates its weights.
- If prediction value $y' = 0$ and actual value is 1 , then the weights are too low; so, for each feature $x_i$, $w_i = w_i \cdot \alpha$ , where $\alpha > 1$
- If $y' = 1$ and actual value is 0, then the weight is too high $w_i = w_i \cdot \beta$, where $0 < \beta < 1$



# Multilayered perceptrons

- Perceptrons can not classify non-linear problem.
- To solve this problem the multilayered perceptron has been created.
- A multi-layer neural neural network consists of large number of units joined together in a pattern of connections
- Units are usually segregated into three classes: input units; output units; and units in between known as hidden units




Feed-forward ANN

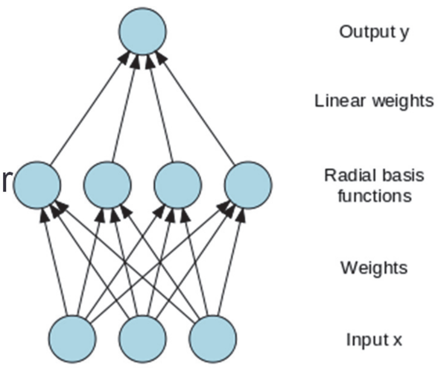
# Multilayered perceptrons

- Generally, properly determining the size of the hidden layer is a problem, because to small number can be cause lower accuracy , while excessive nodes can result of overfitting.
- The second important step is to calculate weight
- The most well-known and widely used learning algorithm to estimate the values of the weights is the Back Propagation



# Radial Basis Function (RBF) networks

- An RBF network is a three-layer feedback network, in which each hidden unit implements a radial activation function aand each output unit implements a weighted sum of hidden units outputs
- Training procedure:
  - Determining of hidden layer
  - Connections between hidden layer and output are determined by LMS




- The problem of selecting the appropriate number of basis functions remains a critical issue for RBF.

# Instance-based learning

- Instance-based learning is lazy- learning algorithm, as they delay the induction or generalization process until classification is performed.
- It is required less time during training (compare with decision tree, neural, Bayes) but more time during making decision.
- One the instance-based algorithm is nearest neighbour
- The absolute position of the instances within this space is not as significant as the relative distance between instances.
- This distance is determined by using a distance metric



# kNN

- The power kNN has been demonstrated in a number of real domains, but it also has some problem:
  - It is required large storage
  - It is sensitive to the choice of the similarity function that is used to compare instances
  - There is not principle way to choose k

| |
|---|
| Minkowsky: $D(x,y)=\left(\sum_{i=1}^{m}\lvert x_i-y_i\rvert^r\right)^{1/r}$ |
| Manhattan: $D(x,y)=\sum_{i=1}^{m}\lvert x_i-y_i\rvert$ |
| Chebychev: $D(x,y)=\max_{i=1}^{m}\lvert x_i-y_i\rvert$ |
| Euclidean: $D(x,y)=\left(\sum_{i=1}^{m}\lvert x_i-y_i\rvert^2\right)^{1/2}$ |
| Camberra: $D(x,y)=\sum_{i=1}^{m}\frac{\lvert x_i-y_i\rvert}{\lvert x_i+y_i\rvert}$ |
| Kendall's Rank Correlation: $D(x,y)=1-\frac{2}{m(m-1)}\sum_{i=j}^{m}\sum_{j=1}^{i-1}sign(x_i-x_j)sign(y_i-y_j)$ |

Approaches to define the distance between instances (x and y)

# Support Vector Machines

- SVMs revolve around the notation of a “margin”- either side of hyperplane that separates two data classes.
- Maximizing the margin and thereby creating the largest possible distance between the separating hyperplane and the instances on either side of it has been proven to reduce an upper bound on the expected generalization error.
- If the training data is linearly separable, then a pair (*w,b*) exists such that

$$\mathbf{w}^T\mathbf{x}_i + b \geq 1, \text{ for all } \mathbf{x}_i \in P$$

$$\mathbf{w}^T\mathbf{x}_i + b \leq 1, \text{ for all } \mathbf{x}_i \in N$$

- With the decision rule given by $f_{\mathbf{w},b}(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^T\mathbf{x} + b)$
- Kernel trickes



# Discussion

| | Decision Trees | Neural Networks | Naïve Bayes | kNN | SVM | Rule-learners |
|---|---|---|---|---|---|---|
| Accuracy in general | ** | *** | * | ** | **** | ** |
| Speed of learning with respect to number of attributes and the number of instances | *** | * | **** | **** | * | ** |
| Speed of classification | **** | **** | **** | * | **** | **** |
| Tolerance to missing values | *** | * | **** | * | ** | ** |
| Tolerance to irrelevant attributes | *** | * | ** | ** | **** | ** |
| Tolerance to redundant attributes | ** | ** | * | ** | *** | ** |
| Tolerance to highly interdependent attributes (e.g. parity problems) | ** | *** | * | * | *** | ** |
| Dealing with discrete/binary/continuous attributes | **** | ***(not discrete) | ***(not continuous) | ***(not directly discrete) | **(not discrete) | ***(not directly continuous) |
| Tolerance to noise | ** | ** | *** | * | ** | * |
| Dealing with danger of overfitting | ** | * | *** | *** | ** | ** |
| Attempts for incremental learning | ** | *** | **** | **** | ** | * |
| Explanation ability/transparency of knowledge/classifications | **** | * | **** | ** | * | **** |
| Model parameter handling | *** | * | **** | *** | * | *** |

Comparing learning algorithms (**** stars represent the best and * star the worst performance)

# Conclusion

- The SVM has been shown the best accuracy, but it also has weak point.
- By combination of different ML algorithms the efficiency can be improved.