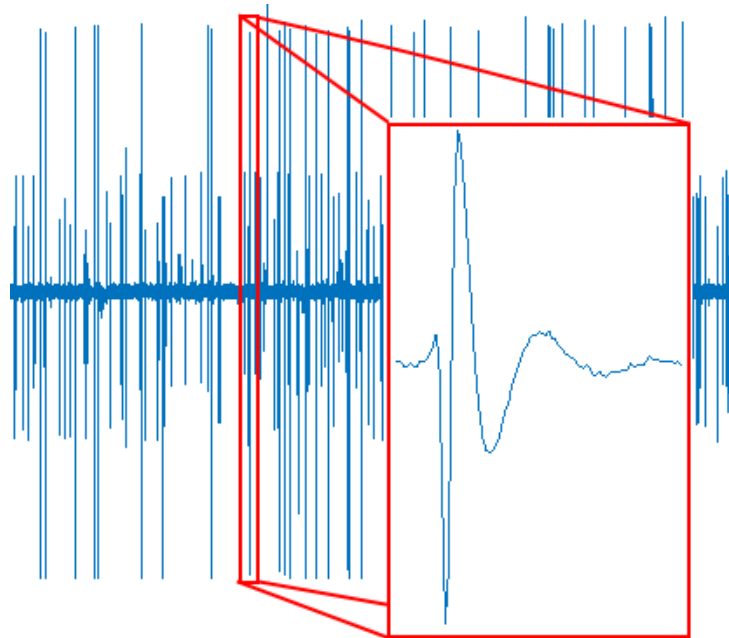




## Βιοϊατρική Τεχνολογία



### Ομάδα 9

Παπαδόπουλος Κωνσταντίνος (8677) [konserpap@auth.gr](mailto:konserpap@auth.gr)

Τοπαλίδης Ευθύμιος (8417) [eatopalid@auth.gr](mailto:eatopalid@auth.gr)

Φάββας Αντώνης (8675) [afavvask@auth.gr](mailto:afavvask@auth.gr)

## Πίνακας Περιεχομένων

|                                 |    |
|---------------------------------|----|
| Εργασία-Project 2018-2019 ..... | 4  |
| Εισαγωγή.....                   | 4  |
| Θέμα 1 <sup>ο</sup> .....       | 4  |
| Ερώτημα 1.1.....                | 4  |
| Matlab .....                    | 4  |
| Γραφήματα.....                  | 5  |
| Παρατηρήσεις.....               | 9  |
| Ερώτημα 1.2.....                | 9  |
| Matlab .....                    | 10 |
| Γραφήματα.....                  | 11 |
| Παρατηρήσεις.....               | 15 |
| Ερώτημα 1.3.....                | 16 |
| Κανόνας - Παρατηρήσεις .....    | 16 |
| Matlab .....                    | 18 |
| Γραφήματα.....                  | 19 |
| Θέμα 2 <sup>ο</sup> .....       | 20 |
| Ερώτημα 2.1.....                | 20 |
| Matlab .....                    | 20 |
| Ερώτημα 2.2.....                | 22 |
| Matlab .....                    | 22 |
| Γραφήματα.....                  | 24 |
| Παρατηρήσεις.....               | 26 |
| Ερώτημα 2.3.....                | 26 |
| Matlab .....                    | 26 |
| Παρατηρήσεις.....               | 27 |
| Ερώτημα 2.4.....                | 29 |
| Matlab .....                    | 29 |
| Γραφήματα.....                  | 32 |
| Παρατηρήσεις.....               | 34 |
| Ερώτημα 2.5.....                | 34 |
| Matlab .....                    | 34 |
| Παρατηρήσεις.....               | 37 |

|   |    |
|---|----|
| Βιβλιογραφική Εργασία 2018-2019.....  | 39 |
| Το Αντίστροφο Πρόβλημα του ΗΕΓ με Ανάλυση Ανεξαρτήτων Συνιστωσών (Inverse problem of EEG with Independent Component Analysis) ..... | 39 |
| Εισαγωγή.....   | 39 |
| Ηλεκτροεγκεφαλογράφημα (ΗΕΓ) .....  | 39 |
| Εισαγωγή.....   | 39 |
| Ιστορικά στοιχεία .....   | 40 |
| Τεχνική ανάλυση.....  | 40 |
| Εγκεφαλικοί ρυθμοί.....   | 42 |
| Σύστημα 10-20 .....   | 44 |
| Σύγκριση του ΗΕΓ με άλλες τεχνικές μέτρησης εγκεφαλικής δραστηριότητας.....   | 46 |
| Ευθύ και Αντίστροφο Πρόβλημα .....  | 49 |
| Ανάλυση Ανεξάρτητων Συνιστωσών (ICA).....   | 49 |
| Ιστορικά στοιχεία .....   | 49 |
| Ενδεικτικές Μεθοδολογίες και Περιγραφή τους.....  | 50 |
| Εφαρμογές .....   | 57 |
| Υπολογισμός της υπνηλίας των οδηγών αυτοκινήτων βασισμένη στο ΗΕΓ με τη χρήση ICA .....   | 58 |
| ICA αποσύνθεση ΗΕΓ σημάτων για επεξεργασία με fMRI σε περιστατικά επιληψίας .....   | 60 |
| Μελλοντικές εφαρμογές .....   | 62 |
| Βιβλιογραφικές Αναφορές.....  | 63 |
| Παράρτημα (Πηγαίος Κώδικας Matlab) .....  | 64 |
| Θέμα 1 <sup>ο</sup> .....   | 64 |
| Ερώτημα 1.1.....  | 64 |
| Ερώτημα 1.2.....  | 64 |
| Ερώτημα 1.3.....  | 66 |
| Θέμα 2 <sup>ο</sup> .....   | 66 |
| Ερώτημα 2.1.....  | 66 |
| Ερώτημα 2.2.....  | 67 |
| Ερώτημα 2.3.....  | 69 |
| Ερώτημα 2.4.....  | 70 |
| Ερώτημα 2.5.....  | 73 |

## Εργασία-Project 2018-2019

### Εισαγωγή

Στην εργασία αυτή θα ασχοληθούμε με τον εντοπισμό και την ανάλυση κορυφών (spikes) από εξωκυτταρικές καταγραφές του εγκεφάλου.

### Θέμα 1<sup>ο</sup>

Για την ανίχνευση κορυφών από εξωκυτταρικές καταγραφές που έχουν υποστεί ζωνοδιαβατό φιλτράρισμα 300-3000Hz, χρησιμοποιείται συνήθως η μέθοδος της κατωφλίωσης. Πιο συγκεκριμένα χρησιμοποιώντας ένα κατώφλι πλάτους, εντοπίζεται και καταγράφεται ως κορυφή η δραστηριότητα του σήματος όπου σε μια γειτονιά υπερβαίνει την τιμή ενός κατωφλίου  $T$ .

### Ερώτημα 1.1

Ζητείται να απεικονιστούν τα 10000 πρώτα δείγματα των σημάτων που βρίσκονται στα αρχεία Data\_Test\_x.

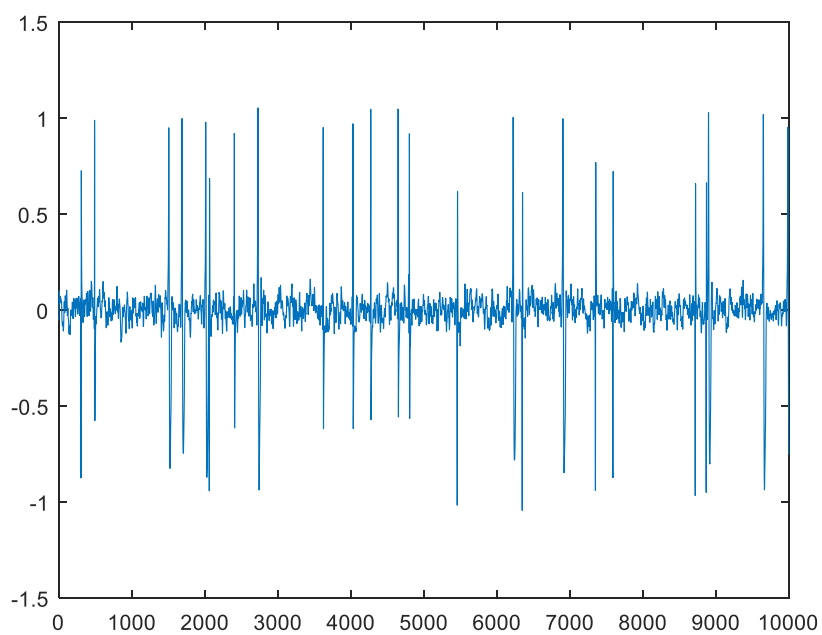
### Matlab

```
function erotimal_1(data)
    x = 1:10000;           % x coordinated of plot

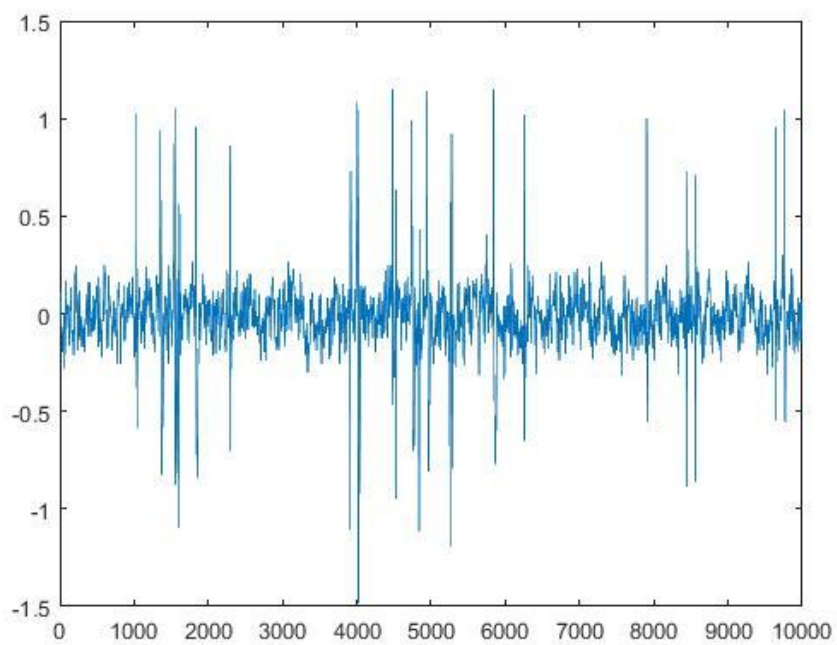
    y = data(:,1:10000);   % y coordinates of plot

    plot(x,y);
```

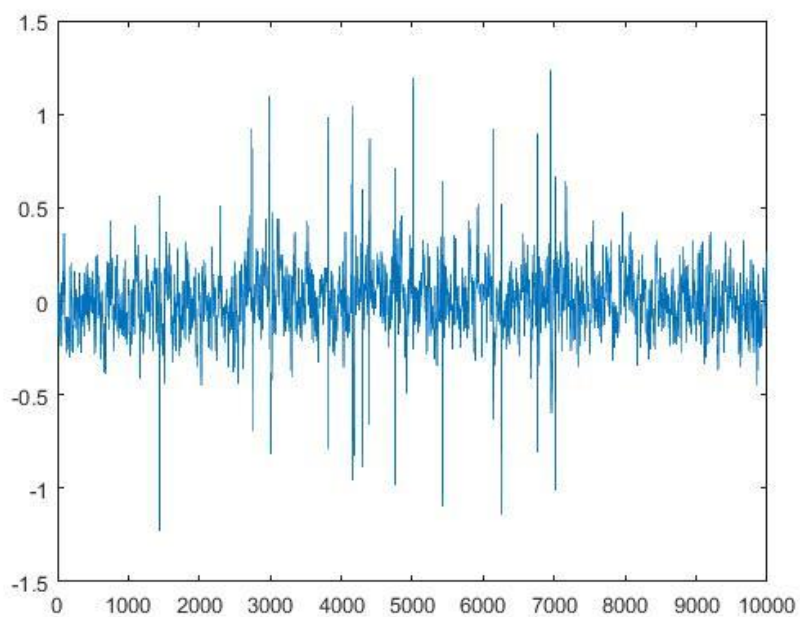
## Γραφήματα



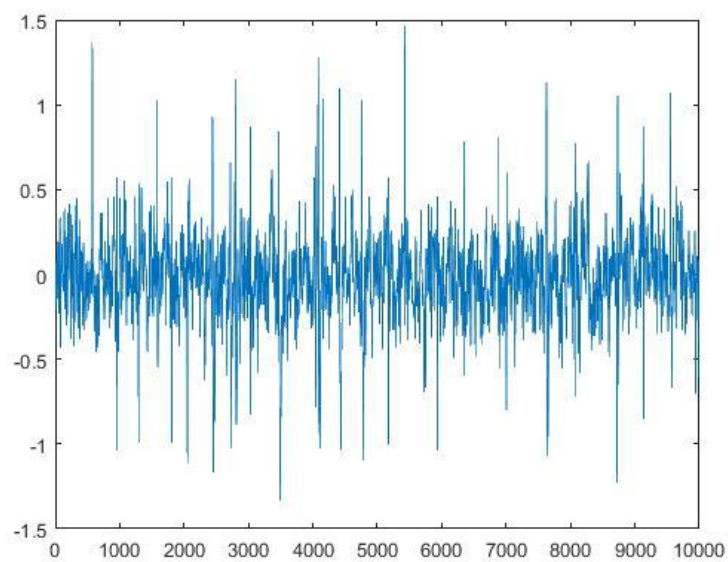
Γράφημα 1: Ερώτημα 1.1 - Data\_Test\_1



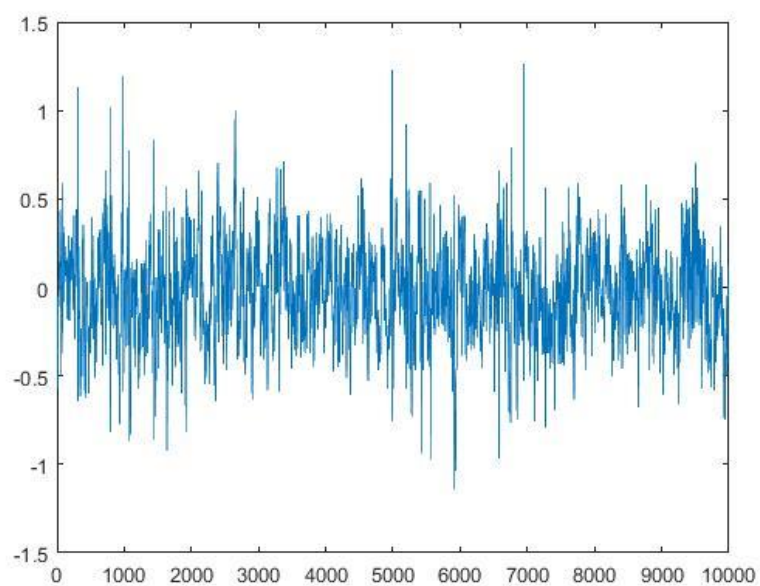
Γράφημα 2: Ερώτημα 1.1 - Data\_Test\_2



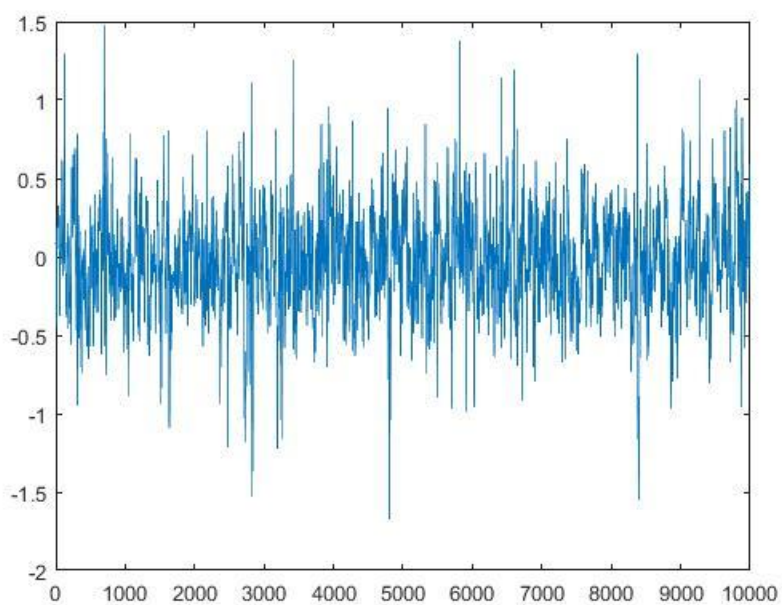
Γράφημα 3: Ερώτημα 1.1 - Data\_Test\_3



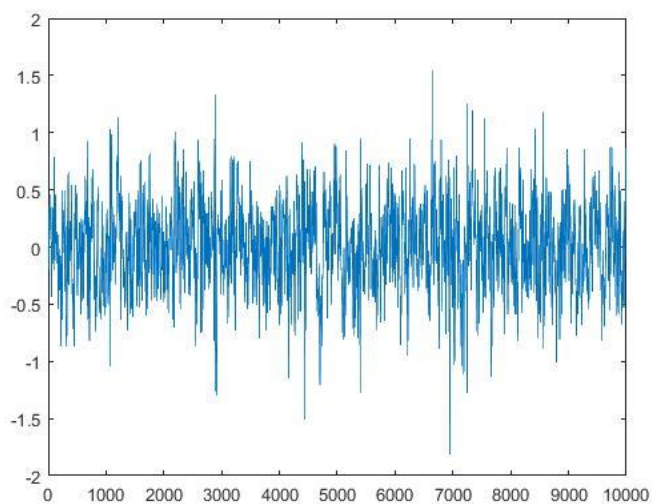
Γράφημα 4: Ερώτημα 1.1 - Data\_Test\_4



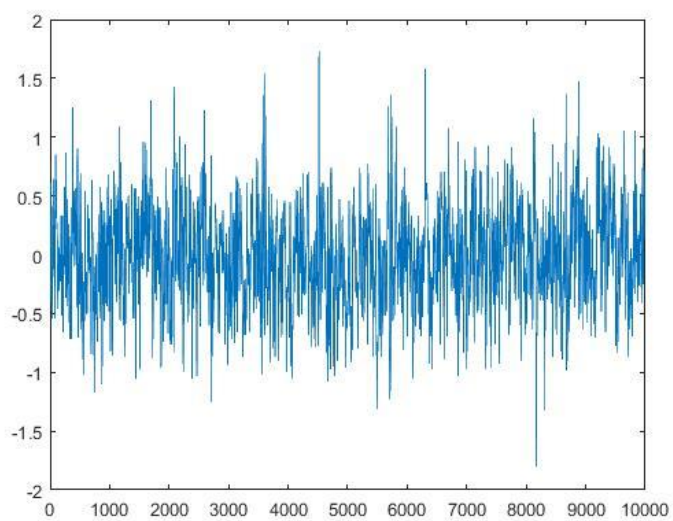
Γράφημα 5: Ερώτημα 1.1 - Data\_Test\_5



Γράφημα 6: Ερώτημα 1.1 - Data\_Test\_6



Γράφημα 7: Ερώτημα 1.1 - Data\_Test\_7



Γράφημα 8: Ερώτημα 1.1 - Data\_Test\_8



## Παρατηρήσεις

Αρχικά παρατηρούμε ότι τα γραφήματα είναι όλα φραγμένα στο διάστημα -2 έως 2. Κάποια μέρη του σήματος δημιουργούν αιχμές πολύ μεγαλύτερες από το υπόλοιπο σήμα τόσο στα θετικά, όσο και στα αρνητικά.

Πηγαίνοντας από το γράφημα 1 προς το 8 παρατηρείται επίσης μία μείωση της διαφοράς των «αιχμών» του σήματος με τα υπόλοιπα σημεία (κάτι που θα ερμηνεύσουμε αργότερα ως αύξηση της τυπικής απόκλισης του σήματος και της τυπικής απόκλισης του θορύβου).

## Ερώτημα 1.2

Μια σχέση που χρησιμοποιείται συνήθως στη βιβλιογραφία για τον ορισμό της τιμής του κατωφλίου για τον εντοπισμό κορυφών είναι η σχέση  $T = k \cdot \sigma_n$  όπου  $\sigma_n$  είναι μια εκτίμηση της τυπικής απόκλισης του θορύβου (πληροφορία άσχετη με τη δραστηριότητα των κορυφών).

Χρησιμοποιώντας τη σχέση  $T = k \cdot \sigma_n$  για διάφορες τιμές της παραμέτρου  $k$  ζητείται να καταμετρήσουμε με τη μέθοδο της κατωφλίωσης τις κορυφές που υπάρχουν σε κάθε ένα από τα σήματα στα αρχεία Data\_Test\_χ. και να κάνουμε το γράφημα των #κορυφών vs.  $k$  και για τα οκτώ σήματα (ένα γράφημα για κάθε αρχείο).

## Matlab

Υπολογίζουμε τον αριθμό των spikes:

```
function num = spikenum(data,k)
    s = median(abs(data)) / 0.6745; % calculation of standard deviation of
noise
    T = k*s;                        % T: the threshold
    len = length(data);
    num = 0;                        % initializing spikes to zero
    i = 1;                          % i variable used to parse through the
data
    flag = 1;                       % flag used to spot the correct # of
spikes

    while i <= len
        if data(i) > T && flag ==1
            num = num + 1;
            flag = 0;
        elseif data(i) < T
            flag = 1;
        end
        i = i + 1;
    end
end
```

Για τη σχεδίαση των γραφημάτων:

```
k = 0:0.01:10;    % x coordinated of plot

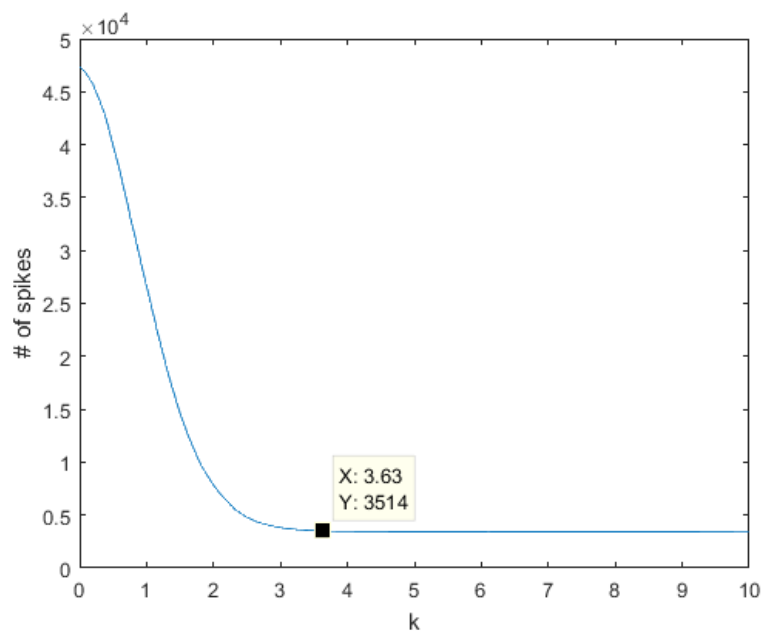
y = zeros(1,1000); % preallocating memory, the code runs much faster
because there is   % no need to repeatedly reallocate memory for the growing
data structure

z = 1;            % initial value of counter

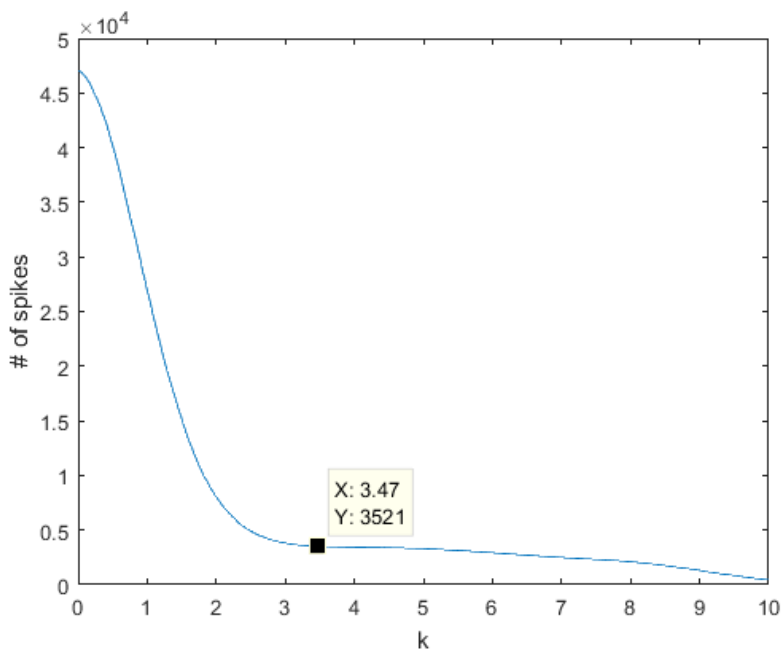
for i = k
    y(z) = spikenum(data8,i);
    z = z + 1;
end

plot(k,y);
xlabel('k')
ylabel('# of spikes')
```

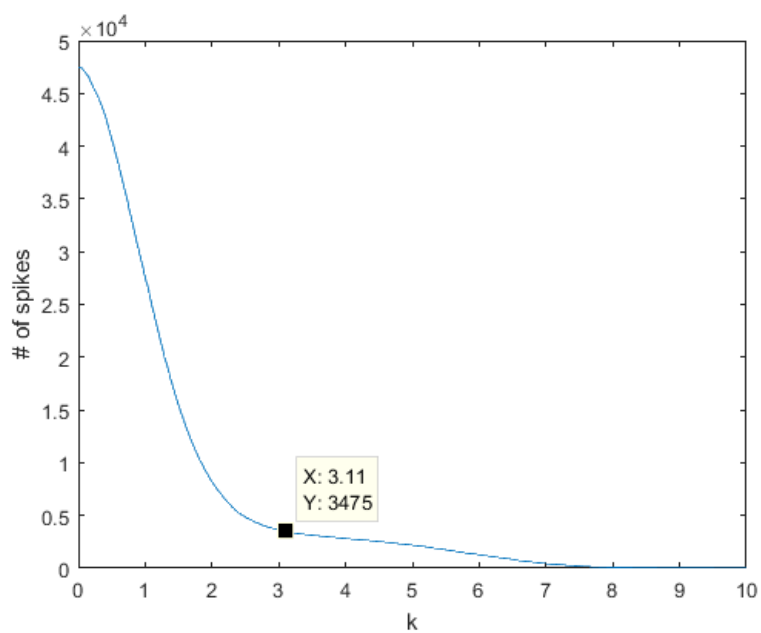
## Γραφήματα



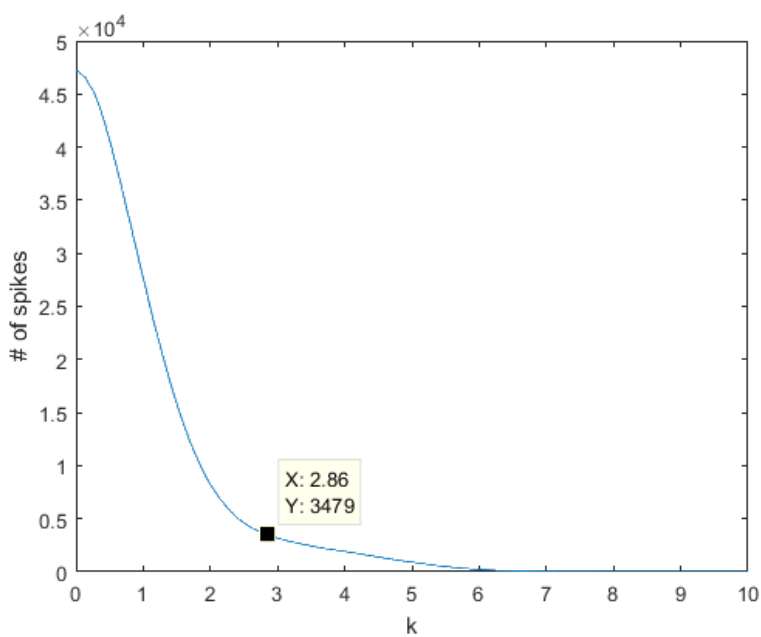
Γράφημα 9: Ερώτημα 1.2 - Data\_Test\_1



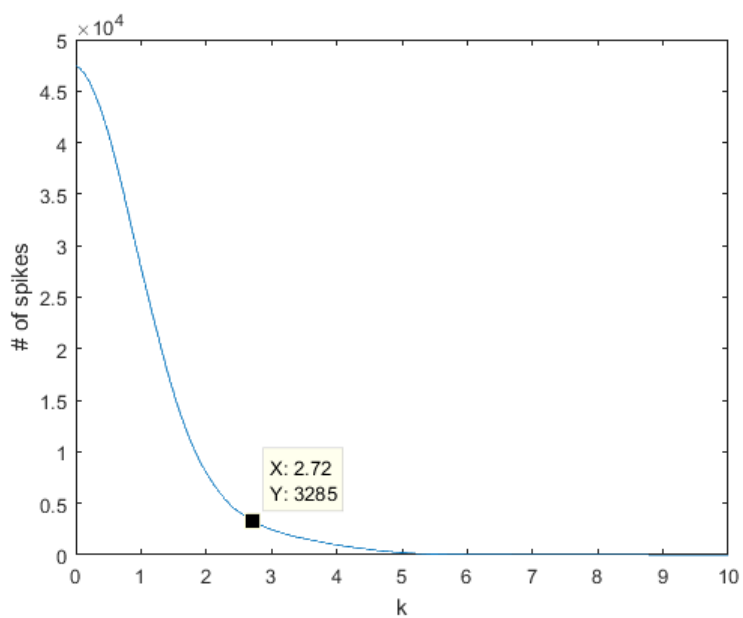
Γράφημα 10: Ερώτημα 1.2 - Data\_Test\_2



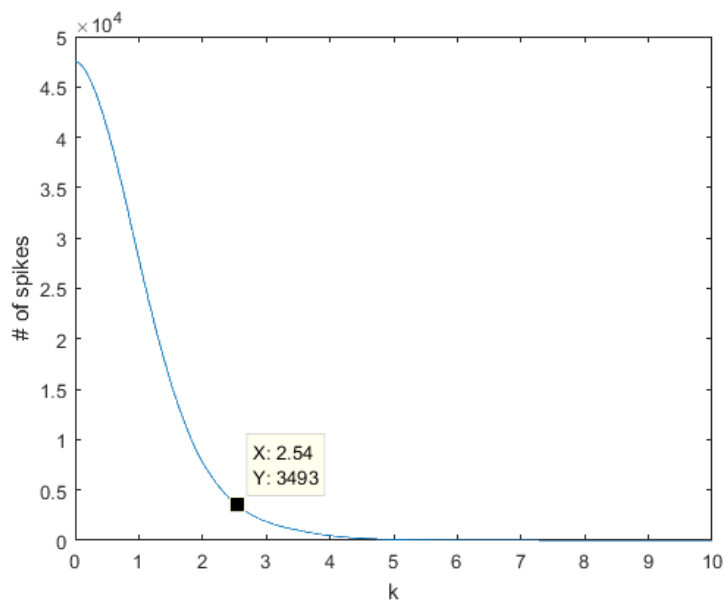
Γράφημα 11: Ερώτημα 1.2 - Data\_Test\_3



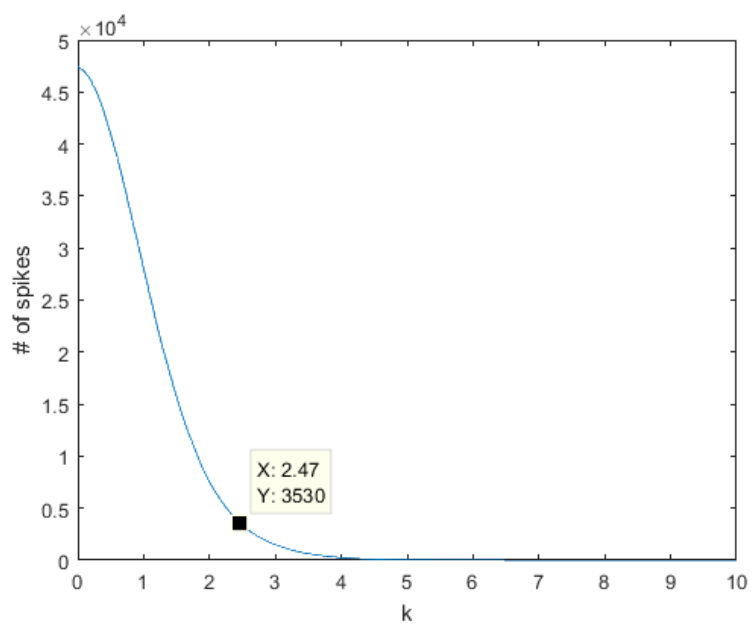
Γράφημα 12: Ερώτημα 1.2 - Data\_Test\_4



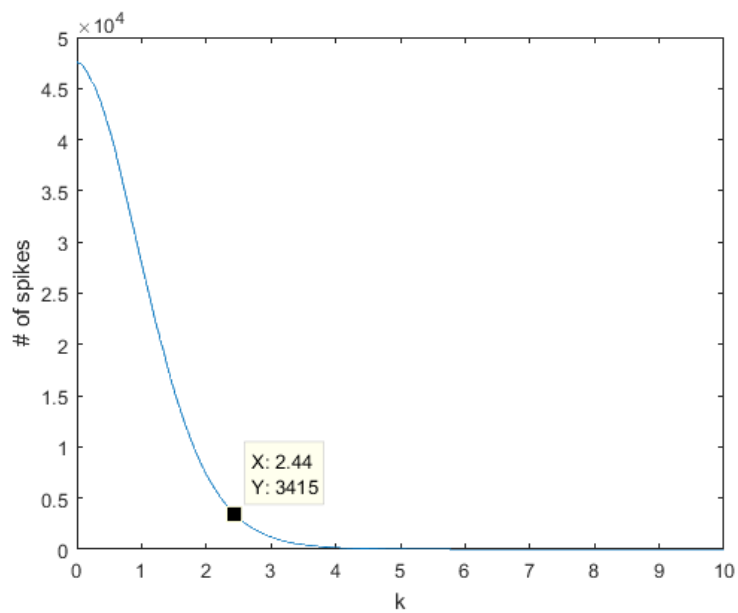
Γράφημα 13: Ερώτημα 1.2 - Data\_Test\_5



Γράφημα 14: Ερώτημα 1.2 - Data\_Test\_6



Γράφημα 15: Ερώτημα 1.2 - Data\_Test\_7



Γράφημα 16: Ερώτημα 1.2 - Data\_Test\_8

Παρατηρήσεις

Παρατηρήθηκε ότι από τα γραφήματα 1 προς 8 η τυπική απόκλιση του σήματος αυξάνεται, ενώ αντίθετα το  $k$  μειώνεται. Υπάρχει γενικά και μια αύξηση του σφάλματος του αριθμού των spikes που εντοπίζονται (έχοντας πάρει ως βήμα 0.01 στο  $k$  και προσπαθώντας να βρεθεί το  $k$  που προσεγγίζει καλύτερα το σωστό αριθμό spikes).

| $k$          | # Spikes Detected | # True Spikes | Error                 | Standard Deviation |
|--------------|-------------------|---------------|-----------------------|--------------------|
| $k_1 = 3.63$ | 3514              | 3514          | 0                     | 0.1509             |
| $k_2 = 3.47$ | 3521              | 3522          | $0.28 \times 10^{-3}$ | 0.1751             |
| $k_3 = 3.11$ | 3475              | 3477          | $0.58 \times 10^{-3}$ | 0.2062             |
| $k_4 = 2.86$ | 3479              | 3474          | $1.44 \times 10^{-3}$ | 0.2446             |
| $k_5 = 2.72$ | 3285              | 3298          | $3.94 \times 10^{-3}$ | 0.2851             |
| $k_6 = 2.54$ | 3493              | 3475          | $5.18 \times 10^{-3}$ | 0.3315             |
| $k_7 = 2.47$ | 3530              | 3534          | $1.13 \times 10^{-3}$ | 0.3779             |
| $k_8 = 2.44$ | 3415              | 3386          | $8.56 \times 10^{-3}$ | 0.4238             |

### Ερώτημα 1.3

Ζητείται να ορίσουμε έναν εμπειρικό κανόνα για την παράμετρο  $k$  ώστε ο αριθμός των κορυφών που υπολογίσαμε για κάθε σήμα εξωκυτταρικής καταγραφής να είναι κοντά στον πραγματικό αριθμό (μεταβλητή spikeNum).

#### Κανόνας - Παρατηρήσεις

Αρχικά παρατηρήθηκε η συσχέτιση μεταξύ  $k$  και τυπικής απόκλισης (std\_dev), κάτι που επιβεβαιώθηκε και στο Matlab, βρίσκοντας το συντελεστή αυτοσυσχέτισης να είναι ίσος με -0.9441.

Στη συνέχεια δημιουργήθηκε το διάγραμμα τυπικής απόκλισης –  $k$  και η παραγόμενη καμπύλη προσεγγίστηκε με ένα πολυώνυμο 4<sup>ου</sup> βαθμού (με αρκετά καλή προσέγγιση).

Πρέπει να λάβουμε υπόψη την πιθανότητα υπερπροσαρμογής (overfitting) του μοντέλου μας. Ένα υπερπροσαρμοσμένο μοντέλο είναι ένα στατιστικό μοντέλο που περιέχει περισσότερες παραμέτρους από αυτές που μπορούν να δικαιολογηθούν από τα δεδομένα. Ένα σύνολο δεδομένων εκπαίδευσης, το οποίο αποτελείται από πεπερασμένο πλήθος στιγμιότυπων μπορεί σε περιορισμένο βαθμό να αναπαραστήσει το σύνολο των δεδομένων. Συνεπώς, είναι πιθανό τα υποδείγματα που παράγει το σύνολο εκπαίδευσης να μην αναπαριστούν πλήρως τα πραγματικά υποδείγματα του συνόλου των δεδομένων, οδηγώντας σε



υπερπροσαρμογή. Οι συνέπειες της υπερπροσαρμογής μπορεί να μειωθούν αν αυξηθεί το μέγεθος εκπαίδευσης.

Η υπερπροσαρμογή είναι πιθανή στην περίπτωση μας λόγω του ενδεχομένως μικρού σετ δεδομένων εκπαίδευσης (Data\_Test\_1 έως Data\_Test\_8)

Έχοντας βρει το πολυώνυμο που προσεγγίζει την καμπύλη ο κανόνας που ορίζουμε είναι:

**Κανόνας:** Η παράμετρος  $k$  είναι συνάρτηση της τυπικής απόκλισης  $s$  και θα βρίσκεται από τη σχέση  $k(s) = -122.8*s^4 + 108.1*s^3 - 10.256*s^2 - 13.068*s + 5.5507$

\* Διαισθητικά θα μπορούσαμε να πούμε ότι η συσχέτιση που υπάρχει μεταξύ του  $k$  και της τυπικής απόκλισης  $s$  (και συγκεκριμένα όσο αυξάνεται το ένα μέγεθος, μειώνεται το άλλο μέγεθος και αντιστρόφως) εξηγείται με το ότι χρειαζόμαστε χαμηλότερο *threshold* (άρα και  $k$ ,  $T = k * \sigma_n$ ) όταν το  $s$  είναι μεγαλύτερο, για να «πιάσουμε» τα *spikes* που μπορεί να είναι είτε πολύ ψηλά είτε πολύ χαμηλά.

Στον παρακάτω πίνακα βλέπουμε πόσο ακριβής είναι ο κανόνας μας στο training set (για κάθε Data\_Test\_x) που δόθηκε.

| Proposed $k$   | # Spikes Detected | # True Spikes | Error (%) |
|----------------|-------------------|---------------|-----------|
| $k_1 = 3.6530$ | 3508              | 3514          | 0.17      |
| $k_2 = 3.4129$ | 3540              | 3522          | 0.51      |
| $k_3 = 3.1458$ | 3430              | 3477          | 1.35      |
| $k_4 = 2.8831$ | 3431              | 3474          | 1.24      |
| $k_5 = 2.6851$ | 3416              | 3298          | 3.58      |
| $k_6 = 2.5466$ | 3464              | 3475          | 0.32      |
| $k_7 = 2.4771$ | 3490              | 3534          | 1.25      |
| $k_8 = 2.4374$ | 3430              | 3386          | 1.30      |

## Matlab

Συσχέτιση μεταξύ k και τυπικής απόκλισης (std\_dev):

```
corrcoef(k,std_dev) = -0.9441
```

Εύρεση συντελεστών πολυώνυμο 4<sup>ου</sup> βαθμού για την προσέγγιση της καμπύλης:

```
P = polyfit(std_dev,k,4)
```

Συνάρτηση που περιγράφει τον κανόνα που ορίστηκε:

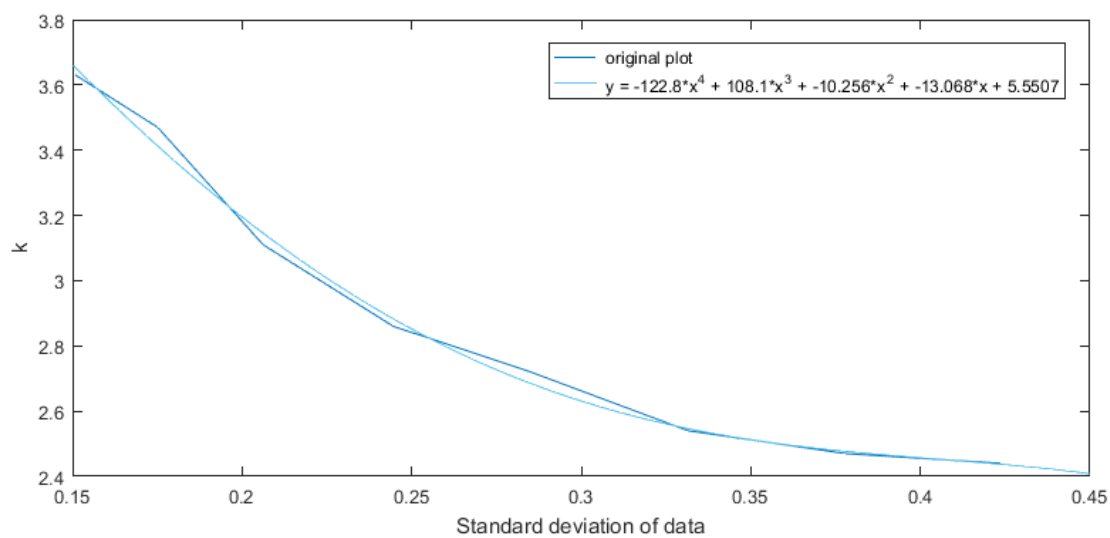
```
function y = pol(x)
```

```
%Coefficients:
```

```
p1 = -122.8;  
p2 = 108.1;  
p3 = -10.256;  
p4 = -13.068;  
p5 = 5.5507;
```

```
y = p1*x^4 + p2*x^3 + p3*x^2 + p4*x + p5;
```

## Γραφήματα



Στο παραπάνω γράφημα φαίνεται το πολυώνυμο 4<sup>ου</sup> βαθμού που προσεγγίζει την καμπύλη της τυπικής απόκλισης –  $k$ .

## Θέμα 2<sup>ο</sup>

Τα αρχεία Data\_Eval\_E\_x όπου  $x = 1, \dots, 4$  περιέχουν διαφορετικά σήματα εξωκυτταρικής καταγραφής. Το σήμα της καταγραφής υπάρχει στο διάνυσμα με το όνομα 'data'.

### Ερώτημα 2.1

Ζητείται να εντοπίσουμε τον αριθμό των κορυφών σε κάθε ένα από τα σήματα στα αρχεία Data\_Eval\_E\_x χρησιμοποιώντας τον εμπειρικό κανόνα του ερωτήματος 1.3 και να καταχωρίσουμε τις χρονικές στιγμές που τις εντοπίσαμε σε ένα διάνυσμα με όνομα spikeTimesEst (ένα διάνυσμα για κάθε αρχείο).

#### Matlab

Η παρακάτω συνάρτηση αποθηκεύει τις χρονικές στιγμές που εντοπίζουμε τις κορυφές στο σήμα:

```
function spikeTimeEst = spikedet(data,k)
    s = median(abs(data)) / 0.6745; % calculation of standard deviation
    T = k*s;                        % T: the threshold
    len = length(data);
    spikeTimeEst = zeros(1,len);    % preallocating memory
    num = 0;                        % initializing spikes to zero
    i = 1;                          % i variable used to parse through the
data
    flag = 1;                       % flag used to spot the correct # of
spikes

    while i <= len
        if data(i) > T && flag ==1
            num = num + 1;
            spikeTimeEst(num) = i;
            flag = 0;
        elseif data(i) < T
            flag = 1;
        end
        i = i + 1;
    end
    spikeTimeEst = spikeTimeEst(1:num); % removes the rest of non used array
```

Στη συνέχεια αποθηκεύουμε στα αντίστοιχα διανύσματα τις χρονικές αυτές στιγμές:

```
spikeTimeEst1 = spikedet(data1,pol(std(data1)));
```

```
spikeTimeEst2 = spikedet(data2,pol(std(data2)));
```

```
spikeTimeEst3 = spikedet(data3,pol(std(data3)));
```

```
spikeTimeEst4 = spikedet(data4,pol(std(data4)));
```

Μπορούμε να βρούμε τον αριθμό κορυφών μέσω των μηκών των προηγούμενων διανυσμάτων:

```
num_spikes1 = length(spikeTimeEst1);
```

```
num_spikes2 = length(spikeTimeEst2);
```

```
num_spikes3 = length(spikeTimeEst3);
```

```
num_spikes4 = length(spikeTimeEst4);
```

|                      | Data1       | Data2       | Data3       | Data4       |
|----------------------|-------------|-------------|-------------|-------------|
| # of spikes detected | <b>3371</b> | <b>3542</b> | <b>3576</b> | <b>3946</b> |

## Ερώτημα 2.2

Ζητείται να απομονώσουμε τις κυματομορφές των κορυφών που εντοπίσαμε σε κάθε αρχείο και να τις στοιχίσουμε με βάση το πρώτο ακρότατό τους σε έναν πίνακα *SpikeEst* (έναν πίνακα για κάθε αρχείο). Επιπλέον, θα απεικονίσουμε όλες τις κυματομορφές των κορυφών που εντοπίσαμε σε ένα γράφημα (ανά αρχείο δεδομένων).

### Matlab

Απομονώνουμε αρχικά τις κυματομορφές των κορυφών και τις στοιχίζουμε με βάση το πρώτο ακρότατο:

```
function SpikeEst = isolate(data, spikeTimesEst)
    len = length(spikeTimesEst); % length of spikeTimeEst
    i = 1; % initializing counter
    SpikeEst = zeros(len, 64); % preallocating 2d matrix
    hold on;

    while i <= len
        temp = data(spikeTimesEst(i)-32:spikeTimesEst(i)+31); % finding min
        and max indexes in data for the % 64 samples
        window
        maximum = max(temp);
        max_i = find(temp == maximum); % index (in
        the temp array) of max
        minimum = min(temp);
        min_i = find(temp == minimum); % index (in
        the temp array) of min
        first = min(max_i,min_i); % index (in
        the temp array) of the first extremum
        first_val = temp(first); % value of
        the first extremum
        index = find(data == first_val); % index (in
        the data array) of the first extremum

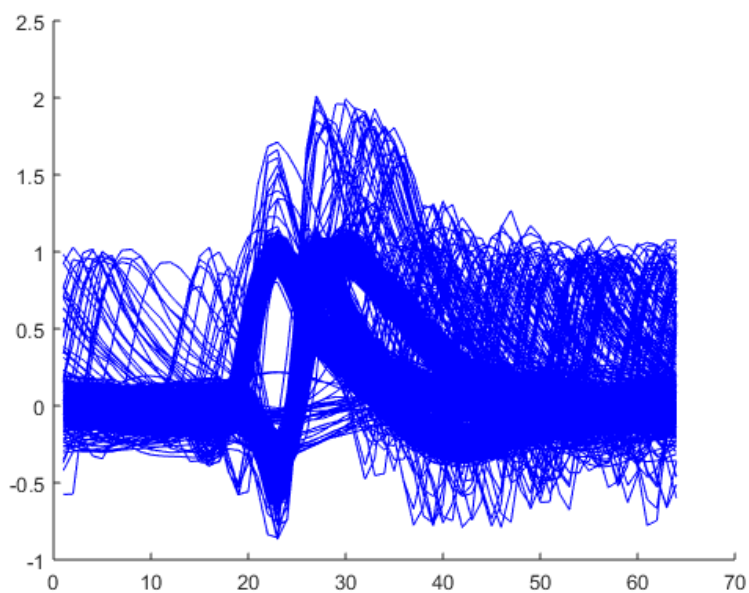
        SpikeEst(i,:) = data(index - 22:index + 41); % chose 64
        sample window % waveforms
        are aligned with index (the first extremum)
        i = i + 1;
    end
```

Για τη σχεδίαση των γραφημάτων:

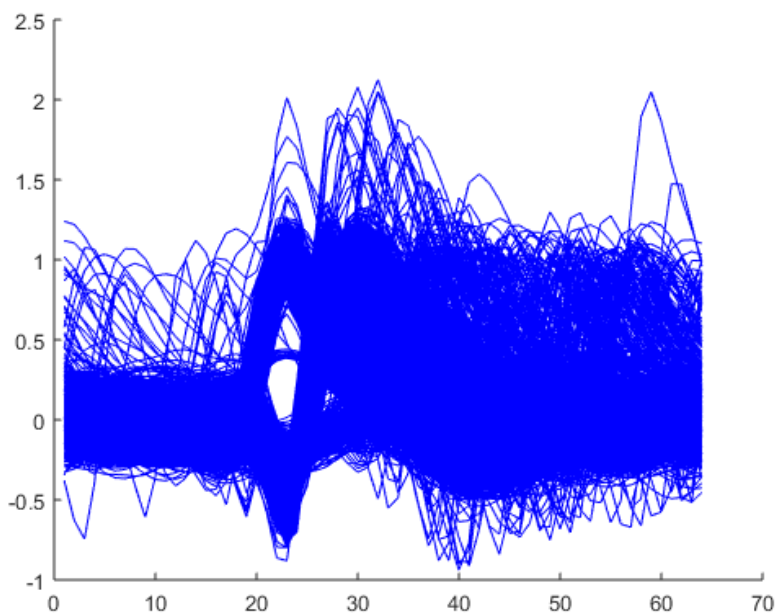
```
x = 1:1:64;
SpikeEst = isolate(data1,spikeTimesEst1);
len = length(spikeTimesEst1);    % length of spikeTimeEst

for i = 1:len
    plot(x,SpikeEst(i,:), 'color', 'blue');
    i = i + 1;
end
```

## Γραφήματα

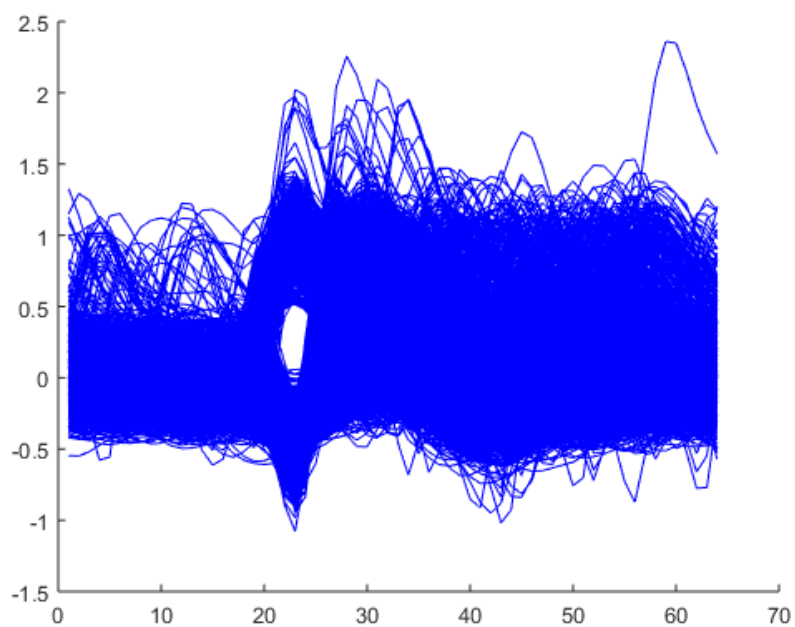


Γράφημα 17: Ερώτημα 2.2 - Data\_Eval\_E\_1

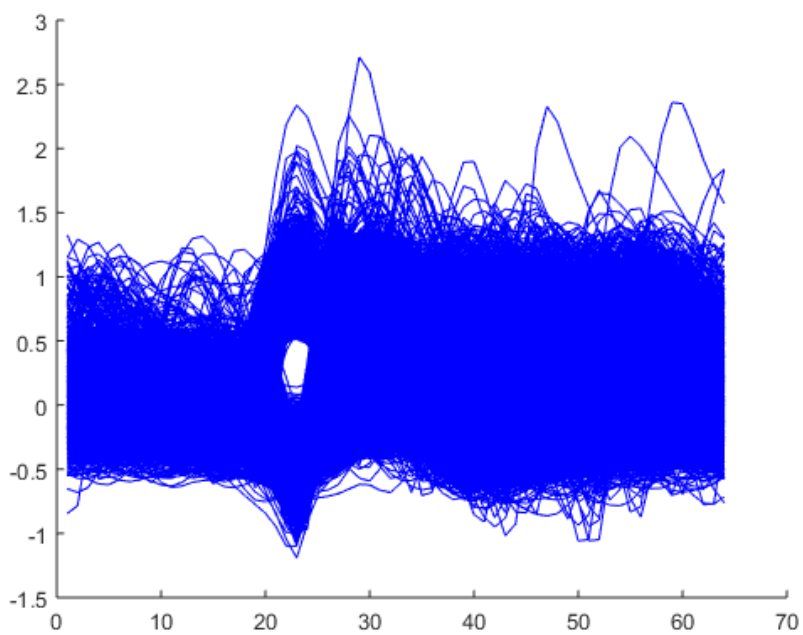


Γράφημα 18: Ερώτημα 2.2 - Data\_Eval\_E\_2





Γράφημα 19: Ερώτημα 2.2 - Data\_Eval\_E\_3



Γράφημα 20: Ερώτημα 2.2 - Data\_Eval\_E\_4

## Παρατηρήσεις

Προχωρώντας από το αρχείο δεδομένων 1 προς το 4 οι κυματομορφές γίνονται όλο και πιο δυσδιάκριτες λόγω του θορύβου. Στο πρώτο γράφημα, που ο θόρυβος είναι χαμηλός, μπορούμε να διακρίνουμε τρεις διακριτές κυματομορφές, λογικά μία από κάθε νευρώνα.

Τα spikes που φαίνονται να μην είναι στοιχισμένα και είναι αριστερά και δεξιά από το σημείο στοίχισης (πρώτο ακρότατο) είναι τα overlapping spikes. Αυτά είναι λίγα σε αριθμό σε σχέση με το συνολικό αριθμό spikes.

## Ερώτημα 2.3

Με βάση τη μεταβλητή spikeTimes πρέπει να υπολογίσουμε πόσες κορυφές εντοπίσαμε σωστά και πόσες από τις κορυφές που εντοπίσαμε δεν είναι στην πραγματικότητα κορυφές αλλά δραστηριότητα που δεν σχετίζεται με τη δραστηριότητα νευρώνων (θόρυβος).

## Matlab

```
spikeTimesEst = spikeTimesEst1;
spikeTimes = spikeTimes1;

len = length(spikeTimesEst); % length of spikeTimeEst
len2 = length(spikeTimes); % length of spikeTimes
displacement = 40; % how much shifting of the two spikes in time
we consider to be the same spike
correct = 0; % number of correct spikes detected
false = 0; % number of wrong spikes detected
y = 1;
init = 1; % where to start checking from in spikeTimes
index_pos_real = zeros(1,len2); % the index position of the spikeTimes
array, where a spike was correctly found
index_pos_est = zeros(1,len); % the index position of the spikeTimesEst
array, where a spike was correctly found
```

```

recorded_shift = zeros(1,len);      % what was the displacement/shift for the
different samples
flag = 1;
for i = 1:len
    while y <= len2
        if abs(spikeTimesEst(i) - spikeTimes(y)) < displacement &&
index_pos_real(y) ~= 1
            correct = correct + 1;
            index_pos_real(y) = 1;      % denotes which particular spike was
found (from spikeTimes)
            index_pos_est(i) = 1;      % denotes if particular spike was
correctly found (from spikeTimesEst)
            recorded_shift(correct) = spikeTimesEst(i) - spikeTimes(y);
            break;
        else
            y = y + 1;
        end
    end
    y = 1;
    i = i + 1;
end

false = len - correct;

fprintf('The number of correctly detected spikes is: ');
correct
fprintf('\n');
fprintf('The number of falsely detected spikes is: ');
length(find(index_pos_est == 0))      % or just false
fprintf('\n');

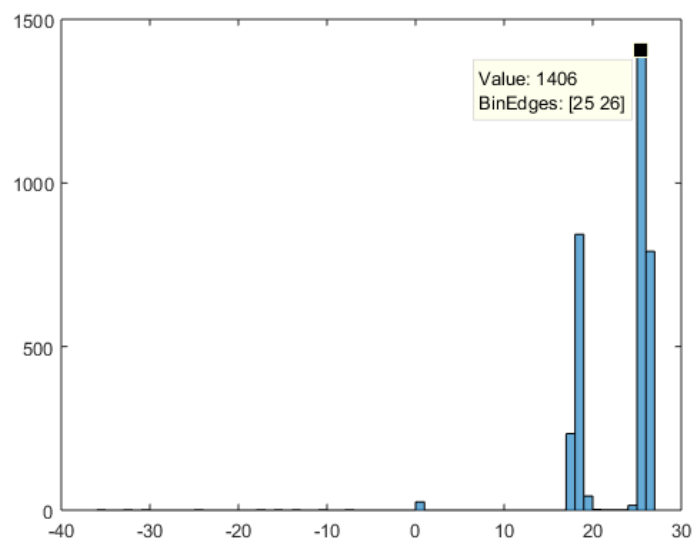
```

## Παρατηρήσεις

Στον παρακάτω πίνακα βλέπουμε για κάθε αρχείο δεδομένων τον αριθμό των spikes που εντοπίσαμε με βάση τον κανόνα (**# Spikes Detected**), τον αριθμό από αυτά που πραγματικά αντιστοιχούν σε spikes (**# Spikes Detected Correctly**) και τον αριθμό των spikes που υπάρχουν πραγματικά με βάση τη μεταβλητή spikeTimes (**# True Spikes**).

| Data  | # Spikes Detected | # Spikes Detected Correctly | # Spikes Detected Falsely | # True Spikes |
|-------|-------------------|-----------------------------|---------------------------|---------------|
| Data1 | 3371              | 3346                        | 25                        | 3410          |
| Data2 | 3542              | 3478                        | 64                        | 3520          |
| Data3 | 3576              | 3370                        | 206                       | 3411          |
| Data4 | 3946              | 3396                        | 550                       | 3526          |

Οι χρονικές στιγμές που καταχωρήθηκαν στο spikeTimesEst του ερωτήματος 2.1 αναφέρονται στη χρονική στιγμή υπέρβασης του threshold, για αυτό και διαφέρουν ελάχιστα από αυτές που αναγράφονται στο spikeTimes παρόλο που αναφέρονται στις ίδιες κορυφές. Ο κώδικάς μας προσαρμόστηκε κατάλληλα για να γίνει αντιστοίχιση μεταξύ των πραγματικών κορυφών και των κορυφών που εντοπίσαμε εμείς. Παραθέτουμε ενδεικτικά για το Data1 τη μετατόπιση αυτή στο ιστόγραμμα που ακολουθεί (Matlab: `histogram(recorded_shift)`).



## Ερώτημα 2.4

Για κάθε μία από τις κυματομορφές που εντοπίσαμε και καταχωρίσαμε στον πίνακα `SpikesEst` για κάθε αρχείο `Data_Eval_E_x`, ζητείται να υπολογίσουμε δύο χαρακτηριστικά της. Στη συνέχεια θα απεικονίσουμε σαν σημεία στο δισδιάστατο χώρο τα ζεύγη των χαρακτηριστικών που υπολογίσαμε για κάθε κυματομορφή χρησιμοποιώντας διαφορετικό χρώμα για κάθε νευρώνα (ένα γράφημα για κάθε αρχείο).

Τα χαρακτηριστικά που επιλέγουμε είναι η τυπική απόκλιση του σήματος και η κλίση της κυματομορφής (κλίση μεταξύ των δύο ακροτάτων).

### Matlab

Για τον υπολογισμό της τυπικής απόκλισης:

```
%SpikeEst = isolate(data1, spikeTimesEst1);
cols = length(SpikeEst(1,:));
rows = length(SpikeEst(:,1));
SpikeStd = zeros(rows,1);

for i = 1:rows
    SpikeStd(i) = std(SpikeEst(i,:));           % find the standard deviation of
    spike array
end
```

Για τον υπολογισμό της κλίσης:

```
%SpikeEst = isolate(data1, spikeTimesEst1);
cols = length(SpikeEst(1,:));
rows = length(SpikeEst(:,1));
SpikeSlope = zeros(rows,1);

for i = 1:rows
    maximum = max(SpikeEst(i,:));
    max_i = find(SpikeEst(i,:) == maximum);      % index (in the
    temp array) of max
    minimum = min(SpikeEst(i,:));
```

```

min_i = find(SpikeEst(i,:) == minimum);

SpikeSlope(i) = (maximum - minimum) / (max_i - min_i);    % find the slope
between the two extremums
end

```

Για τη σχεδίαση του γραφήματος:

```

function scat_plot(Char1, Char2, spikeClass, index_pos_est, index_pos_real)
    %erotima2_3;                                     % run the script with the
corresponding spikeTimesEst, spikeTimes                                     % to get the index_pos_est,
index_pos_real variables
    hold on

    Char1Match = transpose(Char1) .* index_pos_est;      % to only get the
correctly found spikes
    Char2Match = transpose(Char2) .* index_pos_est;
    spikeClassMatch = spikeClass .* index_pos_real;

    empty1 = find(Char1Match == 0);                      % finding zeros to
get rid of the spikes that are false
    empty2 = find(spikeClassMatch == 0);

    Char1Match(empty1) = [];                             % getting rid of the
spikes that are false
    Char2Match(empty1) = [];

    spikeClassMatch(empty2) = [];

    indexes1 = find(spikeClassMatch == 1);               % finding at which
indexes there are spikes of class 1, class 2, class 3
    indexes2 = find(spikeClassMatch == 2);
    indexes3 = find(spikeClassMatch == 3);

    k = 1;

    for i = indexes1                                     % saving the spikes
of only one neuron that are correct according to spikeTimes
        char1_c1(k) = Char1Match(i);
        char2_c1(k) = Char2Match(i);
        k = k + 1;
    end

    scatter(char1_c1, char2_c1, '*', 'blue')

    %-----
    k = 1;

    for i = indexes2
        char1_c2(k) = Char1Match(i);
        char2_c2(k) = Char2Match(i);
        k = k + 1;
    end

```

```
end

scatter(char1_c2, char2_c2, '*', 'red')

%-----
k = 1;

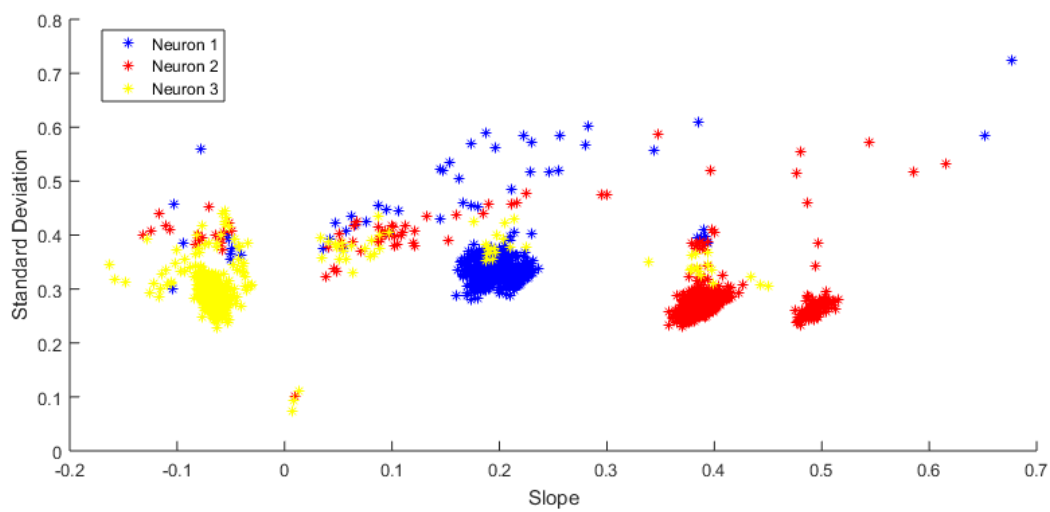
for i = indexes3
    char1_c3(k) = Char1Match(i);
    char2_c3(k) = Char2Match(i);
    k = k + 1;
end

scatter(char1_c3, char2_c3, '*', 'yellow')
```

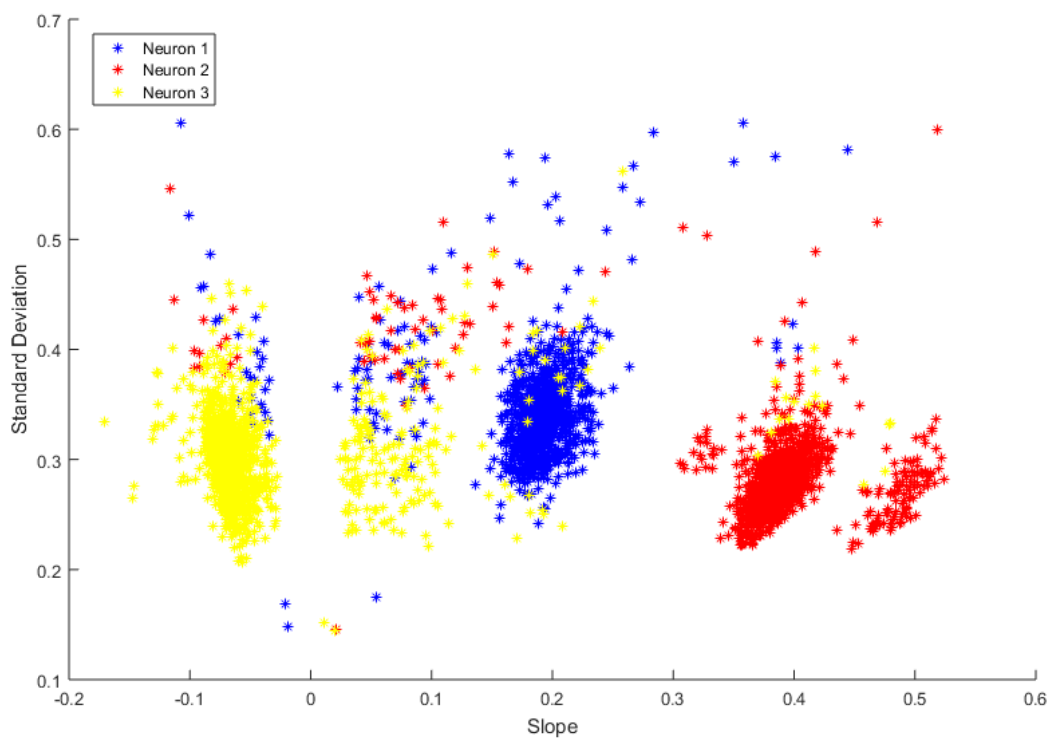
Τρέχουμε στη Matlab:

```
scat_plot(SpikeStd1,SpikeSlope1,spikeClass1, index_pos_est1, index_pos_real1);
scat_plot(SpikeStd2,SpikeSlope2,spikeClass2, index_pos_est2, index_pos_real2);
scat_plot(SpikeStd3,SpikeSlope3,spikeClass3, index_pos_est3, index_pos_real3);
scat_plot(SpikeStd4,SpikeSlope4,spikeClass4, index_pos_est4, index_pos_real4);
```

## Γραφήματα

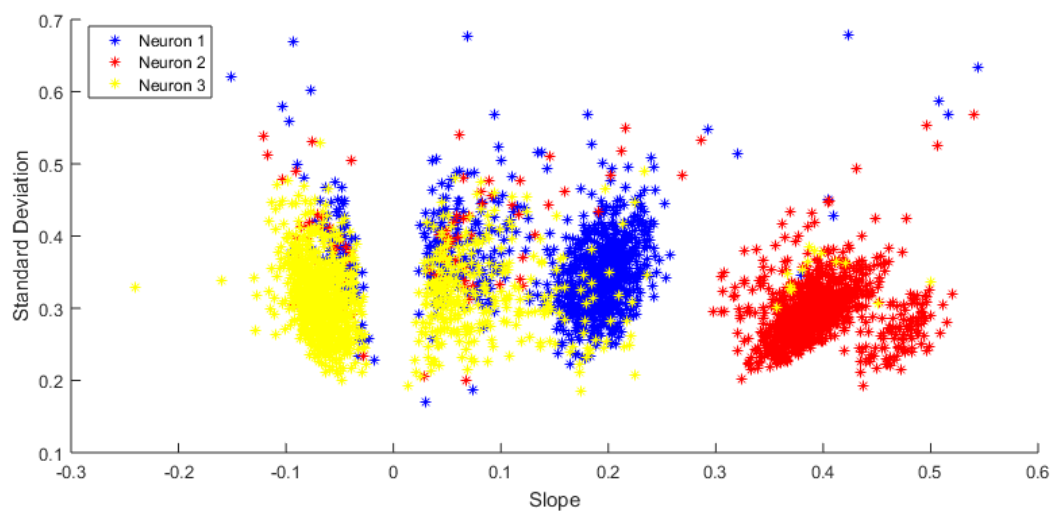


Γράφημα 21: Ερώτημα 2.4 - Data\_Eval\_E\_1

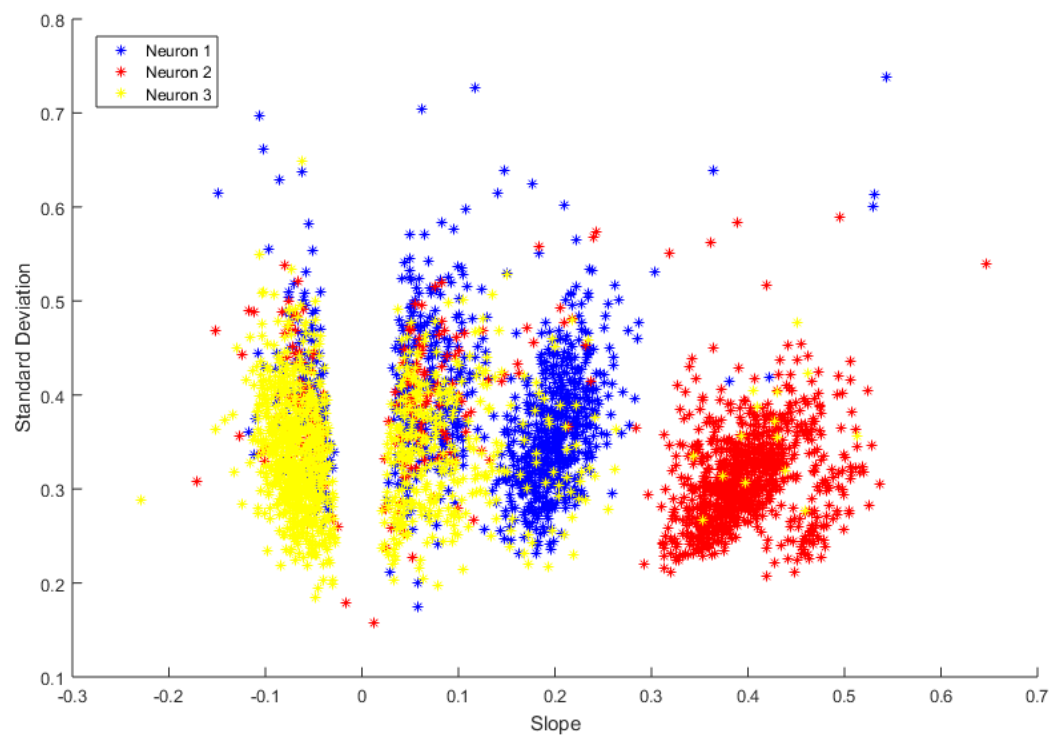


Γράφημα 22: Ερώτημα 2.4 - Data\_Eval\_E\_2





Γράφημα 23: Ερώτημα 2.4 - Data\_Eval\_E\_3



Γράφημα 24: Ερώτημα 2.4 - Data\_Eval\_E\_4

## Παρατηρήσεις

Παρατηρούμε ότι καθώς προχωράμε από το Data 1 προς το Data 4, που αυξάνεται ο θόρυβος, τα συσσωματώματα (clusters) που δημιουργούνται είναι λιγότερο συμπαγή και αρχίζουν να αλληλεπικαλύπτονται. Συνεπώς γίνεται πιο δύσκολο και μπορούμε να αποφανθούμε με μικρότερη ακρίβεια για το αν μία κορυφή με τα συγκεκριμένα χαρακτηριστικά ανήκει ή όχι σε ένα συγκεκριμένο νευρώνα.

## Ερώτημα 2.5

Ζητείται να δημιουργήσουμε τον πίνακα Data και το διάνυσμα group για κάθε ένα από τα αρχεία Data\_Eval\_E\_x. Στη συνέχεια καλούμαστε να ελέγξουμε το ποσοστό σωστής ταξινόμησης (αναγνώρισης) των τριών νευρώνων.

## Matlab

Πέραν από την κλίση και την τυπική απόκλιση, επιλέχθηκαν ως χαρακτηριστικά ο αριθμός διαβάσεων από το μηδέν, ο υπολογισμός της ενέργειας, αν το πρώτο ακρότατο είναι μέγιστο ή ελάχιστο και η απόσταση (πλάτος) μεταξύ του μεγίστου και του ελαχίστου.

### Αριθμός διαβάσεων από το μηδέν:

```
%SpikeEst = isolate(data1, spikeTimesEst1);
cols = length(SpikeEst(1,:));
rows = length(SpikeEst(:,1));
Zeros = zeros(rows,1);

for i = 1:rows
    Zeros(i) = spikenum(SpikeEst(i,:),0);           % find how many times each
    spike has crossed zero
end
```

### Υπολογισμός της ενέργειας (με Fourier):

```
%SpikeEst = isolate(data1, spikeTimesEst1);
cols = length(SpikeEst(1,:));
rows = length(SpikeEst(:,1));
SpikeEnergy = zeros(rows,1);

for i = 1:rows
    F = fft(SpikeEst(i,:));
    pow = F.*conj(F);
    SpikeEnergy(i) = sum(pow) / cols;           % find how much energy the spike has
    with DFT
end
```

### Έλεγχος αν το πρώτο ακρότατο είναι μέγιστο ή ελάχιστο:

```
%SpikeEst = isolate(data1, spikeTimesEst1);
cols = length(SpikeEst(1,:));
rows = length(SpikeEst(:,1));
SpikeMaxInd = zeros(rows,1);

for i = 1:rows
    maximum = max(SpikeEst(i,:));
    max_i = find(SpikeEst(i,:) == maximum);           % index (in the
    temp array) of max
    minimum = min(SpikeEst(i,:));
    min_i = find(SpikeEst(i,:) == minimum);

    if max_i < min_i
        max_first = 1;                               % finds where
    else
        max_first = 0;
    end

    SpikeMaxInd(i) = max_first;
end
```

## Απόσταση (πλάτος) μεταξύ του μεγίστου και του ελαχίστου:

```
%SpikeEst = isolate(data1, spikeTimesEst1);
cols = length(SpikeEst(1,:));
rows = length(SpikeEst(:,1));
SpikeWidth = zeros(rows,1);

for i = 1:rows

    maximum = max(SpikeEst(i,:));
    max_i = find(SpikeEst(i,:) == maximum);           % index (in the
temp array) of max
    minimum = min(SpikeEst(i,:));
    min_i = find(SpikeEst(i,:) == minimum);

    SpikeWidth(i) = abs(max_i - min_i);
end
```

## Για να δώσουμε την κατάλληλη δομή στο Data (ανάλογα με τα χαρακτηριστικά που εξετάζουμε) και στο group:

```
Char1Match = transpose(SpikeSlope) .* index_pos_est;      % to only get the
correctly found spikes
Char2Match = transpose(SpikeStd) .* index_pos_est;
Char3Match = transpose(Zeros) .* index_pos_est;
Char4Match = transpose(SpikeEnergy) .* index_pos_est;
Char5Match = transpose(SpikeMaxInd) .* index_pos_est;
Char6Match = transpose(SpikeWidth) .* index_pos_est;
spikeClassMatch = spikeClass1 .* index_pos_real;

empty1 = find(Char1Match == 0);                            % findind zeros to
get rid of the spikes that are false
empty2 = find(spikeClassMatch == 0);

Char1Match(empty1) = [];                                   % getting rid of the
spikes that are false
Char2Match(empty1) = [];
Char3Match(empty1) = [];
Char4Match(empty1) = [];
Char5Match(empty1) = [];
Char6Match(empty1) = [];

spikeClassMatch(empty2) = [];

group = transpose(spikeClassMatch);

T1 = transpose(Char1Match);
T2 = transpose(Char2Match);
T3 = transpose(Char3Match);
T4 = transpose(Char4Match);
```

```
T5 = transpose(Char5Match);  
T6 = transpose(Char6Match);  
Data = [T1 T2];
```

## Παρατηρήσεις

Σύνολο των εξεταζόμενων χαρακτηριστικών:

1. T1 = κλίση της κυματομορφής
2. T2 = τυπική απόκλιση
3. T3 = αριθμός διαβάσεων από το μηδέν
4. T4 = ενέργεια του σήματος
5. T5 = έλεγχος αν το πρώτο ακρότατο είναι ελάχιστο ή μέγιστο
6. T6 = απόσταση μεταξύ των δύο ακροτάτων

Παρατίθενται τα αποτελέσματα σωστής ταξινόμησης για τα 4 αρχεία με χαρακτηριστικά (στο Data) την κλίση της κυματομορφής (T1) και την τυπική απόκλιση (T2).

|   | Data1   | Data2   | Data3   | Data4   |
|---|---------|---------|---------|---------|
| Correct Classification Percentage (CCP) | 93.9243 | 90.5172 | 84.3719 | 72.1295 |

Αναλόγως τον συνδυασμό των χαρακτηριστικών ο γραμμικός αλγόριθμος μηχανικής μάθησης βγάζει καλύτερα ή χειρότερα αποτελέσματα. Η αύξηση των χαρακτηριστικών, ειδικά αν αυτά δεν είναι «ποιοτικά» (να δίνουν καλό ποσοστό ταξινόμησης και μόνα τους) δεν συνεπάγεται αναγκαστικά αύξηση στα ποσοστά σωστής αναγνώρισης των νευρώνων.

Να σημειωθεί ότι ταξινομητής με ποσοστό σωστής ταξινόμησης της τάξης του 50% δεν θεωρείται καλός, είναι σχεδόν τυχαία αναγνώριση.

Στη συνέχεια δίνουμε ενδεικτικά κάποιους συνδυασμούς χαρακτηριστικών που δοκιμάστηκαν για την αναγνώριση των νευρώνων.

| CCP of<br>Characteristic<br>Combinations | Data1   | Data2   | Data3   | Data4   |
|--|---------|---------|---------|---------|
| T1-T2                                    | 93.9243 | 90.5172 | 84.3719 | 72.1295 |
| T1-T3                                    | 94.6215 | 90.3257 | 85.1632 | 70.6575 |
| T2-T3                                    | 63.8446 | 56.3218 | 47.2799 | 46.4181 |
| T1-T2-T3                                 | 93.9243 | 91.6667 | 85.8556 | 72.7184 |
| T1-T2-T3-T4                              | 94.8207 | 92.1456 | 86.7458 | 72.0314 |
| T1-T2-T3-T4-T5                           | 94.3227 | 89.2720 | 85.5589 | 76.0550 |
| T1-T2-T3-T4-T5-<br>T6                    | 95.2191 | 91.7625 | 84.6686 | 75.9568 |

## Βιβλιογραφική Εργασία 2018-2019

### Το Αντίστροφο Πρόβλημα του ΗΕΓ με Ανάλυση Ανεξαρτήτων Συνιστωσών (Inverse problem of EEG with Independent Component Analysis)

#### Εισαγωγή

Κύριος στόχος αυτής της βιβλιογραφικής εργασίας είναι η λεπτομερής περιγραφή και τεχνική ανάλυση του αλγόριθμου Ανάλυσης Ανεξάρτητων Συνιστωσών (ICA) που χρησιμοποιείται στην επίλυση του αντίστροφου προβλήματος ΗΕΓ, ώστε ο αναγνώστης να κατανοήσει σε βάθος την αρχή λειτουργίας αλλά και τη σημασία του αλγορίθμου σε διάφορες σύγχρονες εφαρμογές. Ωστόσο προκειμένου να γίνει αυτό πρέπει πρώτα να κατανοήσουμε τις βασικές αρχές του ηλεκτροεγκεφαλογράφηματος καθώς τα αποτελέσματα αυτού αποτελούν τις μεταβλητές εισόδου του αλγορίθμου.

#### Ηλεκτροεγκεφαλογράφημα (ΗΕΓ)

##### Εισαγωγή

Το ΗΕΓ είναι μια από τις βασικές μεθόδους καταγραφής της ηλεκτρικής δραστηριότητας του εγκεφάλου. Είναι μη επεμβατική μέθοδος καθώς τα ηλεκτρόδια τοποθετούνται στην εξωτερική επιφάνεια της κεφαλής χωρίς την ύπαρξη κάποιας επέμβασης στο εσωτερικό. Τα ηλεκτρόδια αυτά είναι υπεύθυνα για την καταγραφή των ηλεκτρικών φαινομένων που προέρχονται από ρεύματα ιόντων που αναπτύσσονται από τους νευρώνες του εγκεφάλου. Το ΗΕΓ χρησιμοποιείται ευρέως στη ιατρική για την διάγνωση πολύ σοβαρών

εγκεφαλικών παθήσεων (π.χ. επιληψία, κρανιοεγκεφαλικές κακώσεις, νοητική καθυστέρηση, ψυχικές διαταραχές, όγκοι κ.α.). Τέλος αξιοσημείωτο είναι το γεγονός πως το ΗΕΓ φαίνεται να έχει ευρύ πεδίο εφαρμογών καθώς χρησιμοποιείται σε πολλούς κλάδους της ιατρικής, περισσότερα όμως για αυτό το θέμα θα δούμε στην ενότητα των εφαρμογών.

### *Ιστορικά στοιχεία*

Η ηλεκτρική δραστηριότητα του εγκεφάλου παρατηρήθηκε για πρώτη φορά το 1875 από τον Βρετανό Richard Caton. Η παρατήρηση του Caton επαληθεύτηκε λίγα χρόνια αργότερα το 1890 από τον Πολωνό Adolf Beck ο οποίος παρατήρησε ηλεκτρικές ταλαντώσεις σε διάφορα εγκεφαλικά ηλεκτρικά σήματα. Ωστόσο ο όρος Εγκεφαλογράφημα που χρησιμοποιούμε ακόμη και σήμερα εφευρέθηκε για πρώτη φορά το 1924-1925 από τον Γερμανό ψυχίατρο Hans Berger. Επιπρόσθετα αξιοσημείωτο είναι το γεγονός πως ο Berger είναι υπεύθυνος για την ανακάλυψη των 2 πιο σημαντικών εγκεφαλικών ρυθμών (άλφα και βήτα) καθώς επίσης και για την εφεύρεση του πρώτου ηλεκτροεγκεφαλογράφου. Έπειτα υπήρξαν πάρα πολλές προσπάθειες συσχέτισης του ΗΕΓ με διάφορες παθήσεις του νευρικού συστήματος. Πιο αναλυτικά το 1936 ο Walter προσπάθησε να αποδείξει ότι η ύπαρξη εγκεφαλικών σημάτων χαμηλής συχνότητας σχετίζεται με την ύπαρξη όγκων στον εγκεφαλο. Την ίδια χρονιά οι Gibbs, Davis, Fisher και Lowenback χρησιμοποίησαν το ΗΕΓ για την περιγραφή και την κατανόηση των επιληπτικών κρίσεων. Το 1936 γενικά αποτελεί ορόσημο έτος καθώς τότε ιδρύθηκε και το πρώτο εργαστήριο ΗΕΓ στο γενικό Νοσοκομείο της Μασαχουσέτης στην Αμερική. Αργότερα το 1944 ο D.Hill χρησιμοποίησε το ΗΕΓ με σκοπό την διάγνωση ψυχικών διαταραχών.

### *Τεχνική ανάλυση*

Η καταγραφή των ηλεκτρικών σημάτων του εγκεφάλου γίνεται με ηλεκτρόδια που τοποθετούνται σε αποστάσεις πολύ μεγαλύτερων διαστάσεων (τάξεων μεγέθους) σε σχέση με τις αποστάσεις των νευρώνων που παράγουν τα σήματα. Άρα



συμπεραίνουμε πως τα ηλεκτρόδια καταγράφουν συνισταμένες ηλεκτρικές δραστηριότητες ενός τεράστιου πλήθους νευρώνων που σχηματίζουν ηλεκτρικά δίπολα. Επίσης το πλάτος των σημάτων που ανιχνεύει ο ΗΕΓ βρίσκεται στο διάστημα από 25  $\mu\text{V}$  μέχρι το μέγιστο πλάτος των 100 mV. Για αυτό και χρειάζονται περίπου 10000 νευρώνες προκειμένου να παραχθεί το ελάχιστο ηλεκτρικό δυναμικό που μπορεί να ανιχνευτεί από τον ΗΕΓ. Επιπρόσθετα πρέπει να αποσαφηνιστούν ποια είναι τα βασικά χαρακτηριστικά που καθορίζουν ποιι νευρώνες αποτελούν ηλεκτρικά δίπολα και συνεπώς συνεισφέρουν στο τελικό συνιστάμενο ηλεκτρικό σήμα που καταγράφεται. Πιο αναλυτικά τα χαρακτηριστικά είναι:

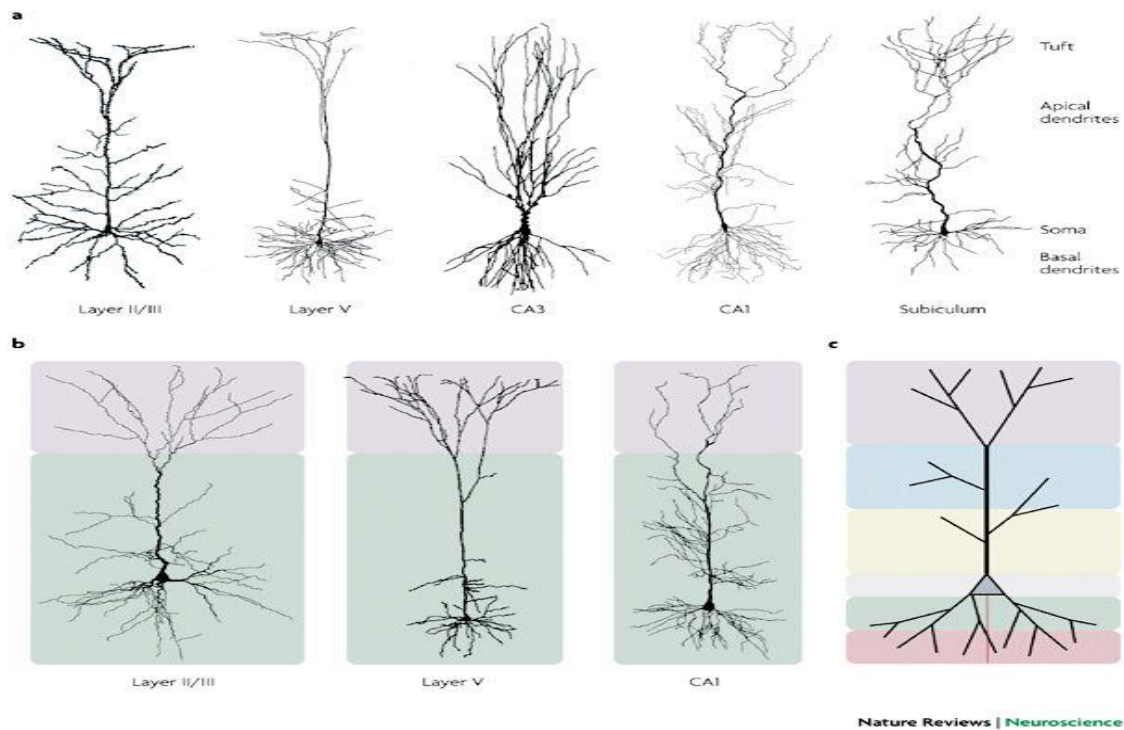
A) Να βρίσκονται κοντά στο εξωτερικό της κεφαλής και να έχουν μεγάλους δενδρίτες.

Όστε το σήμα που δημιουργούν να μην αλλοιώνεται πολύ κατά τη μεταφορά του στην εξωτερική επιφάνεια της κεφαλής

B) Να είναι ομοιόμορφα διατεταγμένα και κάθετα προσανατολισμένα στον φλοιό του εγκεφάλου.

Με αποτέλεσμα τα ηλεκτρικά πεδία που δημιουργούν να μην αλληλοαναιρούνται.

Τέτοια κύτταρα κατά το πλείστον είναι τα πυραμοειδή κύτταρα του φλοιού και συγκεκριμένα εκείνα του γαγγλιονικού στρώματος. Συμπερασματικά το ΗΕΓ είναι ένα άθροισμα πεδιακών δυνάμεων, ενός μεγάλου αριθμού διπόλων, που δημιουργούνται από τα ρεύματα που διαρρέουν τα πυραμοειδή κύτταρα.



Φωτογραφία απεικόνισης πυραμοειδών κυττάρων.

### Εγκεφαλικοί ρυθμοί

Πρέπει να τονίσουμε το γεγονός πως οι νευρώνες παράγουν σήματα με μία περιοδικότητα. Η συχνότητα αυτής της περιοδικής δραστηριότητας εμπεριέχει μεγάλο πλήθος μεταβλητών και ποικίλει ανάλογα με την διεργασία που εκτελεί ο εγκέφαλος αλλά και την συνολική κατάσταση στην οποία βρίσκεται το άτομο (π.χ. εγρήγορσης, ύπνου, κούρασης κ.α.). Έτσι το ΗΕΓ σαν σήμα στην ουσία είναι η υπέρθεση πολλών απλούστερων κυματομορφών που ονομάζονται εγκεφαλικοί ρυθμοί. Κατά τη διάρκεια μιας δραστηριότητας είναι δυνατόν να εμφανίζονται και οι 4 βασικοί ρυθμοί, ωστόσο αυτό δεν συμβαίνει συχνά, καθώς συνήθως έχουμε την εμφάνιση μόνο των 2 βασικών ρυθμών (άλφα και βήτα).

Στη συνέχεια παραθέτουμε τα ιδιαίτερα χαρακτηριστικά κάθε ρυθμού καθώς επίσης και την περιοχή συχνοτήτων που εμφανίζονται:

#### 1. Άλφα Ρυθμός (Berger - 1929)

Ονομάστηκε έτσι καθώς ήταν τα πρώτα κύματα που παρατήρησε ο Berger. Τα κύματα αυτού του ρυθμού έχουν εύρος συχνοτήτων από 8-12 Hz, πλάτος από 20 έως και 60  $\mu V$  και ημιτονοειδή μορφή με συχνότητα 1-3 δευτερολέπτων. Παρατηρείται στις οπίσθιες περιοχές του εγκεφάλου και χαρακτηρίζεται από συμμετρία ως προς τα 2 ημισφαίρια. Εμφανίζεται κατά κόρον όταν το άτομο χαλαρώνει με κλειστά μάτια ενώ διακόπτεται κατά το άνοιγμα των οφθαλμών. Ζωτικής σημασίας είναι να τονίσουμε ότι διαφορά 1 Hz στην συχνότητα ανάμεσα στα 2 ημισφαίρια θεωρείται ανησυχητική. Επίσης υποκατηγορία του Άλφα θεωρείται ο ρυθμός Μι που έχει εύρος συχνότητας 7-11 Hz και εμφανίζεται σε καταστάσεις ανυπομονησίας. Αν όμως δεν εμφανίζεται και στα 2 ημισφαίρια τότε υπάρχει κίνδυνος ύπαρξης εγκεφαλικής δυσλειτουργίας.

## 2. Βήτα Ρυθμός (Berger - 1931)

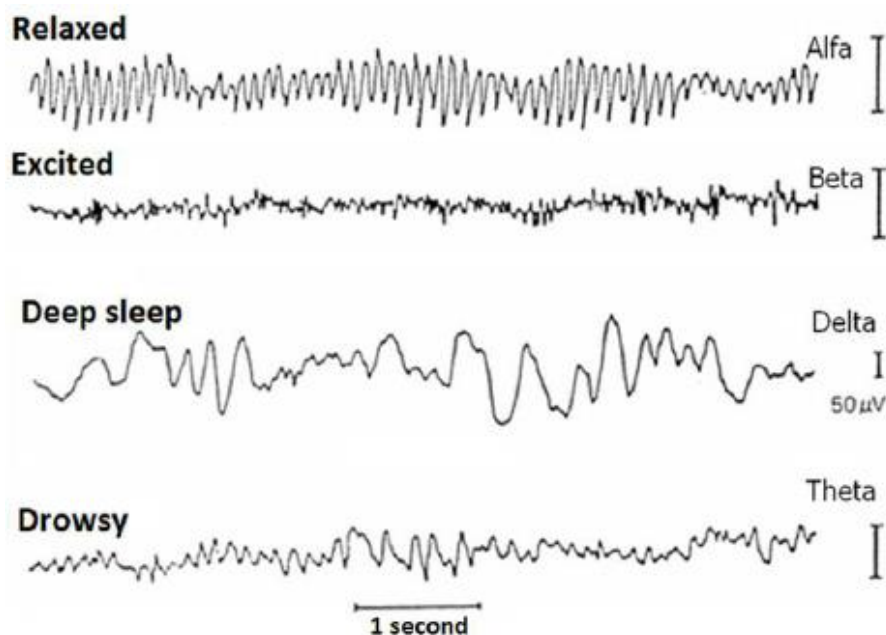
Ο ρυθμός αυτός έχει εύρος συχνότητας από (12-30 Hz) με πιο συνηθισμένες αυτές στο διάστημα των 20-25 Hz και πλάτος που φθάνει μέχρι και τα 20  $\mu V$ . Τοπογραφικά εμφανίζεται συμμετρικά και στα 2 ημισφαίρια του εγκεφάλου. Επίσης καταγράφεται συνήθως σε ενήλικα άτομα που βρίσκονται σε καταστάσεις εγρήγορσης, άγχους ακόμα και βαθιάς συγκέντρωσης. Τέλος πρέπει να τονιστεί πως παντελής έλλειψη αυτού του ρυθμού αποτελεί ανησυχητικό φαινόμενο ακόμα και σε άτομα μικρής ηλικίας.

## 3. Θήτα Ρυθμός (Walter και Doney - 1944)

Ο ρυθμός αυτός έχει εύρος συχνότητας από 4-8 Hz και πλάτος σήματος από 50 έως 150  $\mu V$ . Τοπογραφικά εμφανίζεται στον οπίσθιο θάλαμο του εγκεφάλου για αυτό άλλωστε και έλλειψη αυτού του ρυθμού συνήθως συνεπάγεται βλάβη στο συγκεκριμένο σημείο του εγκεφάλου. Εμφανίζεται κυρίως στα πρώτα χρόνια της ζωής του ατόμου και μειώνεται σταδιακά με την πάροδο των ετών. Ωστόσο μπορεί να εμφανιστεί και σε ενήλικες όταν αυτοί βρίσκονται σε κατάσταση υπνηλίας ή υπερχαλάρωσης. Υποκατηγορίες αυτού ρυθμού αποτελούν ο ρυθμός Κάπα που σχετίζεται με την εκπόνηση μαθηματικών πράξεων από το άτομο και ο παθητικός Θήτα που είναι ιδιαίτερα ανησυχητικός καθώς σχετίζεται άμεσα με σοβαρές εγκεφαλικές παθήσεις όπως για παράδειγμα επιληψίες, όγκους και μηνιγγίτιδες.

#### 4. Δέλτα Ρυθμός(Walter - 1936)

Ο ρυθμός αυτός έχει εύρος συχνότητας μικρότερο των 4 Hz και έχει αξιοσημείωτο πλάτος που μπορεί να φθάσει μερικά εκατοντάδες  $\mu V$ . Είναι φυσιολογικός όταν εμφανίζεται σε ενήλικες και παιδιά που βρίσκονται στο στάδιο του Βαθύ Ύπνου (REM). Συνήθως όμως σε άλλες περιπτώσεις η εμφάνιση του προμηνύει σοβαρές εγκεφαλικές βλάβες. Τέλος σημαντικότερη υποκατηγορία του είναι ο ρυθμός Πι που εμφανίζεται για πρώτη φορά στην εφηβική ηλικία με συχνότητα 3 Hz και πλάτος 200-300 $\mu V$ .



Σχηματική απεικόνιση των 4 βασικών ρυθμών του εγκεφάλου.

#### Σύστημα 10-20

Όπως σε κάθε τεχνική μέτρησης έτσι και στο ΗΕΓ θέλουμε τα όργανα καταγραφής να τοποθετούνται με τον ίδιο τρόπο σε κάθε μέτρηση. Για αυτό από νωρίς εμφανίστηκε η ανάγκη του ακριβή προσδιορισμού της θέσης των ηλεκτροδίων

στην επιφάνεια της κεφαλής. Το πιο σύνηθες σύστημα είναι το 10-20 το οποίο περιγράφει τις διαφορετικές θέσεις 21 ηλεκτροδίων. Οι αριθμοί 10 -20 αναφέρονται στην ποσοστιαία απόσταση από το ριζορρίνιο (Nasion) μέχρι το ινίο (Inion). Επίσης οι ζυγοί αριθμοί αντιστοιχούν στο δεξί ημισφαίριο ενώ οι μονοί στο αριστερό. Ακόμα το γράμμα z (zero) συμβολίζει το 0 για αυτό και όσο ηλεκτρόδια το εμπεριέχουν στο όνομα τους συνήθως λαμβάνονται ως σημεία αναφοράς. Τέλος τα ονόματα των ηλεκτροδίων στην παρακάτω εικόνα αντιστοιχίζονται ως:

FP: Pre-Frontal

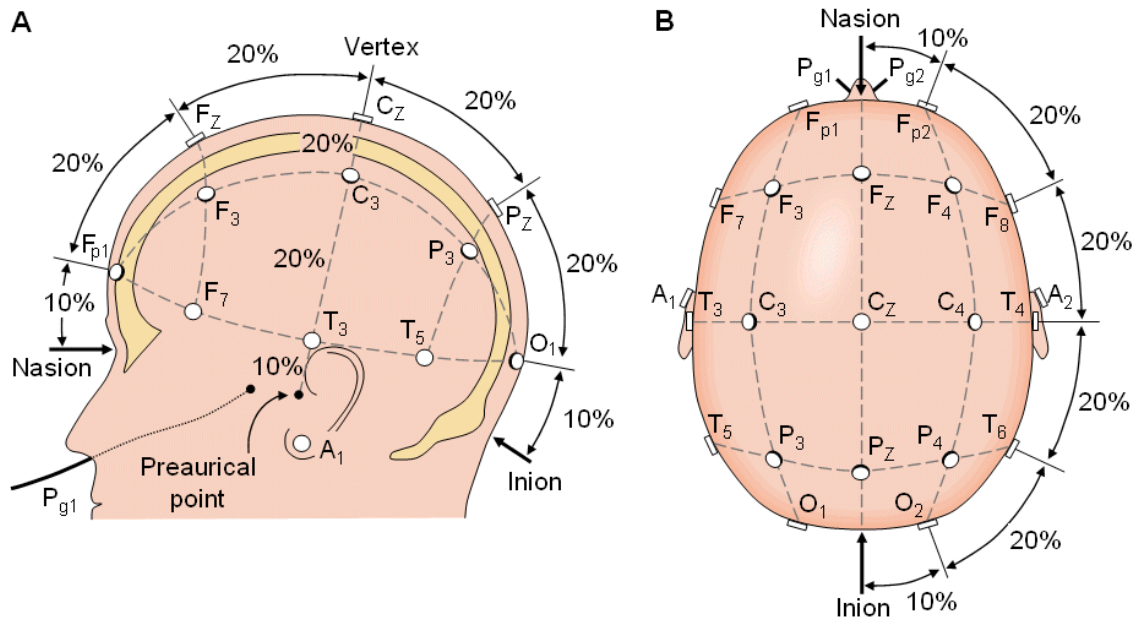
F: Frontal

T: Temporal

P: Parietal

O: Occipital

C: Central



Απεικόνιση του συστήματος 10-20.

## Σύγκριση του ΗΕΓ με άλλες τεχνικές μέτρησης εγκεφαλικής δραστηριότητας

Υπάρχουν πολλές διαφορετικές τεχνικές με τις οποίες μπορούμε να συλλέξουμε πληροφορίες σχετικά με την δομή και την δραστηριότητα των εγκεφαλικών κυττάρων. Οι πιο σημαντικές από αυτές είναι:

### 1) Μαγνητική Τομογραφία (Function Magnetic Resonance Imaging, fMRI)

Το fMRI μετρά την εγκεφαλική δραστηριότητα ανιχνεύοντας δυναμικές αλλαγές στη ροή του αίματος, που προκαλούνται από την ενεργοποίηση των νευρώνων. Η λειτουργία του fMRI βασίζεται στην υπόθεση ότι οι νευρώνες χρειάζονται περισσότερο οξυγόνο όταν είναι ενεργοποιημένοι. Κατά τη διαδικασία της εξέτασης το άτομο πρέπει να είναι οριζόντια ξαπλωμένο και εντελώς ακίνητο καθώς ένα υπεραγώγιμος (superconducting) μαγνήτης περιστρέφεται με μεγάλη συχνότητα γύρω από το σώμα του. Το fMRI στην ουσία καταγράφει τις αλλαγές στο μαγνητικό πεδίο μεταξύ πλούσιου και φτωχού σε οξυγόνο αίματος και δείχνει την σχετική δραστηριότητα διαφορετικών εγκεφαλικών περιοχών. Στη συνέχεια αναφέρονται επιγραμματικά τα πιο βασικά πλεονεκτήματα και μειονεκτήματα του fMRI.

#### Πλεονεκτήματα:

A) Εξαιρετική χωρική ανάλυση της τάξεως του χιλιοστού.

Το fMRI με τις υψηλής ευκρίνειας φωτογραφίες που προσφέρει μας δίνει τη δυνατότητα να προσδιορίζουμε επακριβώς σε ποια περιοχή του εγκεφάλου έχουμε ενεργή δραστηριότητα των νευρώνων.

#### Μειονεκτήματα:

A) Μέτρια χρονική ανάλυση της τάξεως των δευτερολέπτων.

Επειδή οι αιμοδυναμικές αλλαγές γίνονται με πολύ αργό ρυθμό.

B) Εξαιρετικά ακριβός εξοπλισμός.

Με αρχικό κόστος περίπου 1.5 εκατομμύρια ευρώ. Επίσης μεγάλο συνήθως είναι και το κόστος συντήρησης.

Γ) Στατικός εξοπλισμός.

Λόγω του μεγάλου μαγνήτη η μεταφορά του εξοπλισμού είναι εξαιρετικά δύσκολη.

Δ) Ευαίσθητος στις κινήσεις του εξεταζόμενου.

Αν ο εξεταζόμενος κατά τη διάρκεια του FMRI δεν είναι απόλυτα ακίνητος υπάρχει η πιθανότητα ύπαρξης σφαλμάτων στα τελικά αποτελέσματα.

Ε) Απαιτείται ειδική άδεια και ιδιαίτερα καταρτισμένο προσωπικό.

## 2) Μαγνητοεγκεφαλόγραμμα (ΜΕΓ)

Το ΜΕΓ καταγράφει τα μαγνητικά πεδία που δημιουργούνται από την δραστηριότητα των νευρώνων. Έχει παρόμοια τεχνικά χαρακτηριστικά με αυτά του FMRI που αναφέρονται παραπάνω. Πιο αναλυτικά τα βασικά του πλεονεκτήματα και μειονεκτήματα είναι:

Πλεονεκτήματα:

Α) Χωρική ανάλυση της τάξεως εκατοστών.

Βασικό πλεονέκτημα του ΜΕΓ έναντι του ΗΕΓ όσον αφορά τη χωρική ανάλυση είναι το γεγονός πως το ΜΕΓ έχει τη δυνατότητα καταγραφής εγκεφαλικής δραστηριότητας νευρώνων τοποθετημένων πιο βαθιά μέσα στον εγκέφαλο σε αντίθεση με το ΗΕΓ όπου μελετώνται κυρίως νευρώνες που βρίσκονται κοντά στο φλοιό. Αυτό συμβαίνει διότι η ανατομία του κρανίου δεν επηρεάζει τόσο το μαγνητικό σήμα.

Β) Εξαιρετική χρονική ανάλυση της τάξεως των millisecond.

Μειονεκτήματα:

Α) Εξαιρετικά ακριβός εξοπλισμός.

Όπως και το FMRI έτσι και το ΜΕΓ έχει περίπου αρχικό κόστος στο διάστημα των 1.5 με 2 εκατομμύρια ευρώ.

Β) Στατικός εξοπλισμός

Σε αυτή την περίπτωση ο εξεταζόμενος εισέρχεται σε ένα κλειστό και προστατευμένο «θάλαμο» σε σχήμα κυλίνδρου, προκειμένου να μην υπάρχουν

παρεμβολές από εξωτερικά μαγνητικά πεδία κατά την συλλογή των απαραίτητων δεδομένων.

Γ) Ευαίσθητο στις κινήσεις του εξεταζόμενου.

Δ) Απαιτείται ιδιαίτερα καταρτισμένο προσωπικό.

### 3) Ηλεκτροεγκεφαλογράφημα (ΗΕΓ)

Έχοντας αναλύσει την βασική αρχή λειτουργίας του ΗΕΓ παραπάνω (2.1.2) θα προχωρήσουμε απευθείας στην παράθεση των πιο σημαντικών του χαρακτηριστικών:

Πλεονεκτήματα:

A) Εξαιρετική χρονική ανάλυση της τάξεως των millisecond.

Παρουσιάζει χρονική ανάλυση παρόμοια με αυτή του ΜΕΓ.

B) Χαμηλό Κόστος

Σε αντίθεση με τις 2 προηγούμενες τεχνικές το ΗΕΓ είναι αξιοσημείωτα πιο φθινό. Αναλυτικότερα το εύρος κόστους ενός ηλεκτροεγκεφαλογράφου κυμαίνεται μεταξύ 10.000-100.000 ευρώ.

Γ) Μη στατικός εξοπλισμός

Σε αντίθεση με το ΜΕΓ και το FMRI, ο ηλεκτροεγκεφαλογράφος είναι πολύ εύκολο να μεταφερθεί λόγω του μικρού όγκου του.

Δ) Δεν απαιτεί ιδιαίτερα εκπαιδευμένο προσωπικό.

Μειονεκτήματα:

A) Μέτρια χωρική ανάλυση της τάξεως των εκατοστών.



### *Ευθύ και Αντίστροφο Πρόβλημα*

Όπως γίνεται εύκολα αντιληπτό από τα προηγούμενα το ευθύ πρόβλημα του ΗΕΓ είναι στην ουσία ο υπολογισμός των τάσεων που καταγράφονται από τα ηλεκτρόδια στην εξωτερική επιφάνεια της κεφαλής και οφείλονται στην δραστηριότητα των νευρώνων στο εσωτερικό του εγκεφάλου. Για να υπάρξουν μετρήσιμα τέτοια δυναμικά βασική προϋπόθεση είναι η ύπαρξη ενεργών πυραμοειδών κυττάρων κοντά στο φλοιό της κεφαλής που λειτουργούν ως ηλεκτρικά δίπολα. Ωστόσο ένα βασικό μειονέκτημα του ΗΕΓ είναι η χαμηλή χωρική ανάλυση η οποία οφείλεται κυρίως πρώτον, στον μικρό αριθμό ηλεκτροδίων σε σχέση με τον αριθμό των ηλεκτρικών πηγών στον εγκέφαλο και κατά δεύτερον στην ανατομία και δομή του εγκεφάλου. Η βασική αντιμετώπιση αυτού του μειονεκτήματος είναι η λύση του αντίστροφου προβλήματος κατά την οποία γίνεται εκτίμηση του ηλεκτρομαγνητικού πεδίου από την επεξεργασία των αποτελεσμάτων του ΗΕΓ. Στην ουσία το αντίστροφο πρόβλημα ανάγεται στον προσδιορισμό της θέσης των πηγών στο εσωτερικό του εγκεφάλου. Υπάρχουν διάφοροι αλγόριθμοι για την επίλυση του αντίστροφου προβλήματος, ωστόσο εμείς θα εστιάσουμε στην ανάλυση του αλγόριθμου της Ανάλυσης Ανεξάρτητων Συνιστωσών (ICA).

### **Ανάλυση Ανεξάρτητων Συνιστωσών (ICA)**

Όπως αναφέραμε παραπάνω ο αλγόριθμος ανεξάρτητων συνιστωσών (ICA) έχει ως βασικό χαρακτηριστικό πως οι πηγές των σημάτων είναι στατιστικά ανεξάρτητες μεταξύ τους και αποτελεί μια ειδική υποπερίπτωση του προβλήματος τυφλού διαχωρισμού πηγής (Blind source separation).

### *Ιστορικά στοιχεία*

Το πρόβλημα του τυφλού διαχωρισμού πηγής αναλύθηκε με βάση τα στατιστικά του χαρακτηριστικά για πρώτη φορά το 1986 από τους Herault και Jutten, οι οποίοι ήταν και οι πρώτοι που χρησιμοποίησαν τον όρο «Independent Component

Analysis» σε διάφορα επιστημονικά άρθρα που δημοσίευσαν από το 1991 και μετά. Αργότερα κατά το διάστημα 1989-1994 οι Lisker, Nadal και Parga απέδειξαν πως οι αποδοτικότερες τεχνικές χαρτογράφησης είναι αυτές που οδηγούν σε συναρτήσεις παραγωγικής πιθανότητας (factorised source probability density). Την επόμενη χρονιά οι Bell, Senjowski, Roth και Braum μεμονωμένα ανέπτυξαν στοχαστικούς αλγορίθμους με στόχο την εύρεση τεχνικών αποδοτικότερης χαρτογράφησης. Επιπρόσθετα παραγωγικά (Generative) μοντέλα και προσεγγίσεις μέγιστης πιθανότητας (Maximum Likelihood) σχετικά με τον ICA αναπτύχθηκαν από τους Gaeta και Lacoume το 1990 και από τον Pham το 1992. Δύο χρόνια αργότερα (1994) ο Karhuhen αποπειράθηκε για πρώτη φορά να χρησιμοποιήσει τον ICA για προβλήματα μη γραμμικής χαρτογράφησης. Έπειτα το 1997 ο Girolami μαζί με τον Fyfe απέδειξαν την σύνδεση μεταξύ του Projection-Pursuit και των μη-Γκαουσιανών πηγών και χρησιμοποίησαν για πρώτη φορά τη λογική της Ανάλυσης ανεξάρτητων συνιστωσών σε προβλήματα εξόρυξης δεδομένων (Data-mining) αλλά και σε προβλήματα ομαδοποίησης δεδομένων (Data-classification). Στη συνέχεια το 1999 αναπτύχθηκαν από τους Lee, Everson και Roberts τεχνικές που να επιτρέπουν το διαχωρισμό σε Υπό-Γκαουσιανές και Ύπερ-Γκαουσιανές πηγές σημάτων. Τέλος από το 1999 και έπειτα πολλές προσπάθειες γίνονται με σκοπό η χρήση του αλγορίθμου να επεκταθεί σε προβλήματα μη-στατικών πηγών. Γενικά στα μέσα της δεκαετίας του 90' υπήρξε μεγάλο ενδιαφέρον αλλά και πρόοδος γύρω από τα θέματα της Ανάλυσης ανεξάρτητων συνιστωσών και του τυφλού διαχωρισμού πηγής.

## Ενδεικτικές Μεθοδολογίες και Περιγραφή τους

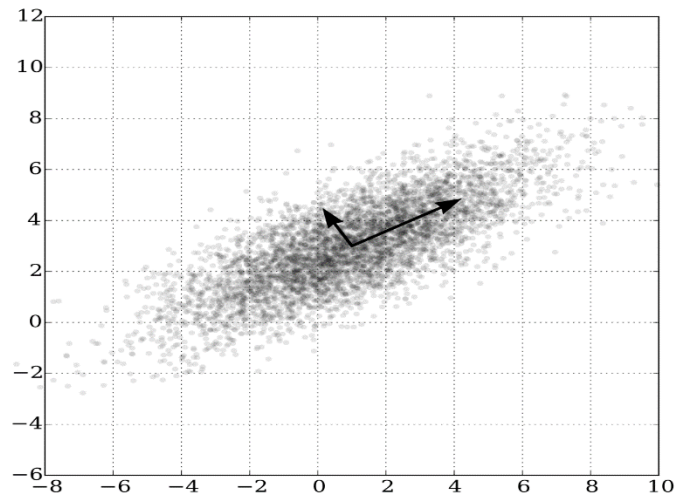
Σε αυτό το σημείο θα αναπτύξουμε ενδεικτικές μεθοδολογίες και περιγραφή τους για το αντίστροφο πρόβλημα ηλεκτροεγκεφαλογραφήματος (ΗΕΓ) μέσω Independent Component Analysis (ICA). Η αρχή της διαδικασίας βρίσκεται στην καταγραφή και την απεικόνιση μέσω του ΗΕΓ της ηλεκτρικής δραστηριότητας των νευρικών κυττάρων του φλοιού του εγκεφάλου (cerebral cortex). Έπειτα προσπαθούμε να εφαρμόσουμε μία χωρο-χρονική ανάλυση στα σήματά μας,

μέσω του κατάλληλου μοντέλου, ούτως ώστε να χαρτογραφήσουμε τον εγκέφαλο συνδέοντας την κάθε περιοχή του με μία συγκεκριμένη λειτουργία. Μέσω των μεθοδολογιών που θα αναπτυχθούν παρακάτω, αυτός ο τρόπος απεικόνισης της εγκεφαλικής δραστηριότητας επανήλθε ξανά στο προσκήνιο γιατί πέρα από την καλή χρονική ανάλυση που είχε, τώρα απέκτησε και αξιόπιστη χωρική ανάλυση. Αυτό αποτελεί μεγάλο πλεονέκτημα για το διαγνωστικό έλεγχο που εκτελούν οι γιατροί αφού δε χρειάζεται να καταφεύγουν σε άλλες κοστοβόρες μεθόδους όπως η λειτουργική μαγνητική τομογραφία (fMRI).

Τα δεδομένα που απεικονίζονται στον ΗΕΓ είναι μία μίξη από τη συνεισφορά πολλών διαφορετικών περιοχών του εγκεφάλου. Πρώτα εφαρμόζουμε σε αυτά την μέθοδο Principle Component Analysis (PCA) όπου καταφέρνουμε να διαχωρίσουμε το σήμα που πραγματικά μας ενδιαφέρει από τον θόρυβο, επιτυγχάνοντας ταυτόχρονα να μειώσουμε τις διαστάσεις του σήματος που μας ενδιαφέρει κάτι που θα μας διευκολύνει στην περαιτέρω επεξεργασία του. Στον υποχώρο του σήματός μας θα εφαρμόσουμε ICA ώστε να διαχωρίσουμε τον εγκέφαλο σε ανεξάρτητες ενεργές περιοχές που η κάθε μια σύμφωνα με τη μοντελοποίηση που θα αναλυθεί εκτενέστερα παρακάτω θα αντιστοιχεί σε ένα δίπολο. Το δίπολο αυτό αποτελεί για εμάς την πηγή ρεύματος και ταυτοχρόνως δυναμικού του οποίου τη θέση θέλουμε να προσδιορίσουμε σύμφωνα με τη λύση του αντίστροφου προβλήματος του ΗΕΓ. Σε κάθε καταγραφή συλλέγουμε δεδομένα από περίπου  $10^6$  νευρώνες που πυροδοτούν ταυτόχρονα και βρίσκονται διάσπαρτοι στον εγκεφαλικό φλοιό. Οι νευρώνες είναι τοποθετημένοι κατά στήλες και κάθετα προσανατολισμένοι στον φλοιό, όπου ο καθένας συνεισφέρει ένα ποσοστό έντασης ηλεκτρικού πεδίου σε ένα από τα 32 ηλεκτρόδια που χρησιμοποιήθηκαν για την περιγραφή της τεχνικής. Αντιλαμβανόμαστε πως η θέση, η αγωγιμότητα του εγκεφάλου αλλά και η ένταση του ηλεκτρικού πεδίου που παράγει ο νευρώνας είναι χαρακτηριστικά τα οποία πρέπει να λάβουμε σοβαρά υπόψη στην πεδιακή ανάλυση που χτίζουμε καθώς αυτή θα μας βοηθήσει να χαρτογραφήσουμε τον εγκέφαλο για διάφορα ερεθίσματα που προκαλούν την ενεργοποίησή του. Ιδιαίτερη προσοχή πρέπει να επιδείξουμε και για τον θόρυβο που υπεισέρχεται στις μετρήσεις μας. Αυτός μπορεί να προέρχεται από την κίνηση των ματιών, από το άνοιγμα και κλείσιμο των βλεφαρίδων, από μυϊκό θόρυβο ακόμα και από εξωγενείς παράγοντες όπως π.χ. λάμπες φθορισμού που επιδρούν στην καταγραφή των βιοσημάτων.

Στην ανάλυση των μεθοδολογιών που αναπτύσσουμε παρακάτω για τη λύση του αντίστροφου προβλήματος προχωράμε σε κάποιες παραδοχές. Πρώτον, θεωρούμε πως η κάθε ενεργός περιοχή μοντελοποιείται με βάση ένα και μόνο δίπολο. Σε αυτή την παραδοχή καταλήγουμε γιατί με τη μοντελοποίηση από πολλά δίπολα είμαστε επιρρεπείς σε θόρυβο, ο οποίος θόρυβος μας οδηγεί σε μη επιθυμητό clustering των νευρώνων σε ενεργές περιοχές. Δεύτερον, θεωρούμε πως οι πηγές των σημάτων δηλαδή τα δίπολα βρίσκονται σε σταθερές θέσεις και κατευθύνσεις δηλαδή μελετάμε στάσιμα σήματα για κάποιο συγκεκριμένο χρονικό διάστημα (χωρική και χρονική στασιμότητα). Αυτή η παραδοχή μας απαλλάσσει από τον κόπο να εξετάζουμε τυχόν αλληλοκαλύψεις και αλληλοακυρώσεις των σημάτων και διευκολυνόμαστε στο να καταλήξουμε στην ανεξαρτησία των συνιστωσών.

Όπως προείπαμε αρχικά εφαρμόζουμε στα δεδομένα που συλλέγουμε την τεχνική Principle Component Analysis. Τα δεδομένα που καταγράφουμε από τον ΗΕΓ είναι ουσιαστικά χρονικές ακολουθίες στα οποία προσδίδουμε κάποια χαρακτηριστικά. Χρησιμοποιούμε πράξεις της γραμμικής άλγεβρας ώστε με γραμμικό συνδυασμό των παλαιών χαρακτηριστικών να παράξουμε νέα χαρακτηριστικά που περιέχουν όλη τη χρήσιμη πληροφορία από τα παλαιά και είναι μικρότερα σε αριθμό. Τα νέα αυτά χαρακτηριστικά είναι ορθογωνικά (κύριες συνιστώσες), δηλαδή ασυσχέτιστα στον αρχικό χώρο διαστάσεων με αποτέλεσμα τα δεδομένα μας να αποκτούν τη μέγιστη διακύμανση καθώς απεικονίζονται ως προς αυτά τα χαρακτηριστικά. Η μείωση των διαστάσεων εξαλείφει τα μη σχετικά γνωρίσματα, δηλαδή εξαλείφουμε τον θόρυβο από τα δεδομένα μας και επιτρέπει την εύκολη οπτικοποίηση αυτών καθώς σε μεγάλες διαστάσεις μετρικές αξιολόγησης, όπως η απόσταση, δεν έχουν νόημα. Στη φωτογραφία βλέπουμε την απεικόνιση των δεδομένων ως προς τον νέο χώρο των καινούριων γνωρισμάτων.



Για παράδειγμα αν έχουμε δύο ενθόρυβα σήματα νευρώνων διαπιστώνουμε πως δεν είναι εύκολο, στο πεδίο του χρόνου, να διακρίνουμε το καθαρό σήμα από αυτό που δε μας ενδιαφέρει. Εφαρμόζοντας όμως μετασχηματισμό Fourier στα δεδομένα και συγκεκριμένα κατασκευάζοντας το φάσμα ισχύος αυτών, πλέον στον νέο χώρο απεικόνισης, μπορούμε εύκολα να τα διακρίνουμε.

Αφού εφαρμόσουμε την PCA στα δεδομένα μας ακολουθεί η τεχνική ICA. Η ICA βασίζεται κυρίως στο εντοπισμό και απεικόνιση εκείνων των σημάτων που δείχνουν στατιστική ανεξαρτησία. Αρχικά συλλέγουμε τα δεδομένα μας και τα αποθηκεύουμε σε έναν πίνακα έστω  $\phi(t_k)$ . Οι στήλες αυτού του πίνακα αναπαριστούν τα δείγματα των σημάτων και οι γραμμές τα κανάλια παρατήρησης, ουσιαστικά τα ηλεκτρόδια. Έπειτα μέσω της PCA ανάλυσης μειώνουμε τη διάσταση των δεδομένων μας και τα διαχωρίζουμε από το θόρυβο. Αυτό γίνεται ως εξής:

Βρίσκουμε τις προβολές του διανύσματος  $\phi(t_k)$  σε κάθε διάσταση υπολογίζοντας τον πίνακα συμμεταβλητότητας :

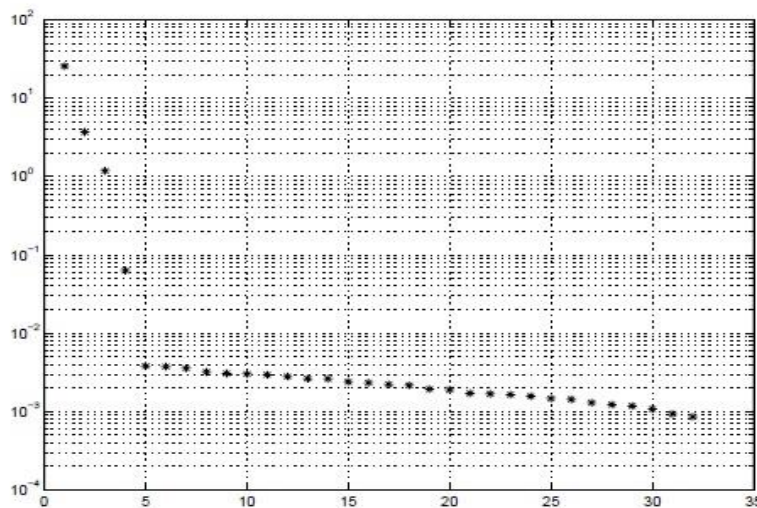
$$R = E\{ \phi(t_k) * \phi(t_k)^T \}$$

Στον πίνακα R εφαρμόζουμε Singular Value Decomposition(SVD)

$$\text{Έχουμε : } R = U_s * \Lambda * U_s^T + U_n * \Lambda * U_n^T$$

Όπου το αριστερό τμήμα αποτελεί τις ιδιάζουσες τιμές  $\Lambda$  (singular values) και τα διανύσματα αυτών  $U_s$  (singular vectors) του επιθυμητού σήματος και το δεξί τις ιδιάζουσες τιμές  $\Lambda$  (singular values) και τα διανύσματα αυτών  $U_n$  (singular vectors) του θορύβου. Αυτός ο διαχωρισμός έγινε με βάση ένα κατώφλι που ορίσαμε για τα singular values του πίνακα  $R$ . Τα singular values που βρίσκονται πάνω από αυτό το κατώφλι συνθέτουν το σήμα ενδιαφέροντος και τα singular values που βρίσκονται κάτω από το κατώφλι συνθέτουν τον θόρυβο. Πλέον μπορούμε να προβάλουμε απευθείας τα καταγραφόμενα σήματα από τα 32 κανάλια στον χώρο του επιθυμητού σήματος μέσω της σχέσης:

$$\Phi_s(t_k) = (\Lambda_s^{-1})^{1/2} * U_s^T * \phi(t_k)$$



Στον παραπάνω πίνακα παρατηρούμε πως οι τέσσερις πρώτες τιμές των singular values του πίνακα  $R$  συνεισφέρουν στον υποχώρο του σήματος μας, με τις υπόλοιπες να προσδιορίζουν τον χώρο του θορύβου.

Έπειτα προσπαθούμε να προσδώσουμε χαρακτήρα στατιστικής ανεξαρτησίας στα καταγραφόμενα σήματα, βήμα σημαντικό ώστε να αντιστοιχήσουμε τελικώς κάθε ένα από αυτά με μία ενεργή περιοχή – ένα δίπολο – και την θέση αυτού (source localization). Ουσιαστικά ψάχνουμε για έναν πίνακα  $W$  ο οποίος εφαρμοζόμενος στα καταγραφόμενα επιθυμητά σήματα  $\Phi_s(t_k)$  να καταφέρει να μας δώσει τις ανεξάρτητες πηγές από τις οποίες παράγονται  $s(t_k)$ . Δηλαδή :

$$s(t_k) = W * \Phi_s(t_k)$$

που εισάγει ο αλγόριθμος infomax ICA. Σύμφωνα με αυτόν λοιπόν:

$$W_{k+1} = W_k + \mu_k * [I + 2 * g(y_k) * y_k^T] * W_k$$

Όπου σε κάθε βήμα ο πίνακας  $W_k$  προσπαθεί να μειώσει την κοινή πληροφορία που μπορεί να έχουν οι έξοδοι  $y_k = W_k * \Phi_s(t_k)$  μέσω μίας μη-γραμμικής συνάρτησης  $g$  (τελεστής μετασχηματισμού) τέτοιας ώστε  $u = g(W * \Phi_s(t_k))$ . Το  $u$  αναπαριστά την ανεξάρτητη μεταβλητή. Δηλαδή μετά από κάποιον αριθμό επαναλήψεων – αναλόγως σε κάθε βήμα αυξάνεται ή ελαττώνεται και ο ρυθμός εκμάθησης  $\mu_k$  – προσπαθούμε να πετύχουμε για τη συνάρτηση πυκνότητας πιθανότητας  $F_u$  του πολυμεταβλητού συστήματος (32 μεταβλητές – κανάλια) να δίνεται από το γινόμενο των συναρτήσεων πυκνότητας πιθανότητας κάθε μεταβλητής χωριστά  $F_{ui}$ :

$$F_u = \prod F_{ui}(u_i)$$

Η συνάρτηση  $g$  θα μπορούσε να είναι η  $g(y_k) = \tanh(y_k)$ .

Μπορούμε έπειτα, αφού υπολογίσουμε τον πίνακα  $W$ , να προβάλουμε τις ανεξάρτητες πηγές  $s_i(t_k)$  στον χώρο των σημάτων μας (ανεξάρτητων πλέον)  $\Phi_s^i(t_k)$  (είναι το σήμα που καταγράφουμε από την  $i$ -οστη πηγή).

$$\Phi_s^i(t_k) = U_s * \Lambda^{1/2} * W^{(-1)} * s_i(t_k)$$

Το τελευταίο βήμα που απαιτείται για την ολοκλήρωση του αντίστροφου προβλήματος είναι η εύρεση της θέσης των πηγών  $s_i(t_k)$ , όπου πλέον αφού πετύχαμε στατιστική ανεξαρτησία μπορούμε να μελετήσουμε την κάθε πηγή ξεχωριστά και να την αντιστοιχήσουμε σε ένα και μόνο δίπολο. Αρχικά θεωρούμε γνωστή τη γεωμετρία και την αγωγιμότητα του εγκεφάλου  $\sigma$  χαρακτηριστικά που υπεισέρχονται στην επίλυση των εξισώσεων.

$$\nabla \cdot (\sigma \nabla \Phi) = \sum I_s(r), \text{ Poisson Equation in head volume } \Omega$$

$$\text{και οριακές συνθήκες: } \sigma(\nabla \Phi) \cdot n = 0, \text{ με } \Phi_j = \phi_j \text{ on } \Gamma_\Omega$$

Όπου τώρα σε αντίθεση με το ευθύ πρόβλημα τα δυναμικά που παράγονται από τα ρεύματα των διπόλων είναι γνωστά και ίσα με τα καταγραφόμενα δυναμικά  $\Phi_j = \phi_j$  και ψάχνουμε τις πηγές των ρευμάτων  $I_s(r)$ .

Κατασκευάζουμε τη συνάρτηση κόστους :

$$\| \varphi - \hat{\varphi} \| = \sum_k \sum_{j=1}^{32} (\varphi_j(t_k) - \hat{\varphi}_j(t_k))^2$$

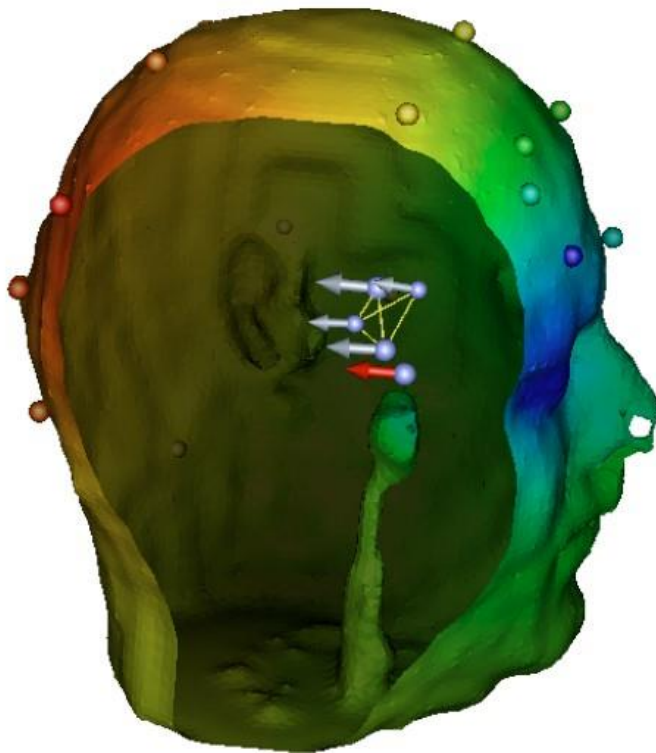
Όπου  $\varphi_j(t_k)$  είναι το δυναμικό που καταγράφουμε στο ηλεκτρόδιο j την χρονική στιγμή  $t_k$  και  $\hat{\varphi}(t_k)$  είναι το δυναμικό που προκύπτει από τη δραστηριότητα μιας συγκεκριμένης πηγής κατά την επίλυση του forward προβλήματος. Επιθυμούμε εκείνες τις λύσεις (πηγές-δίπολα) του forward προβλήματος που να ελαχιστοποιούν το παραπάνω άθροισμα κατά μήκος και των 32 καναλιών καταγραφής. Σε κάθε χρονική στιγμή προσθέτουμε ένα δίπολο στο μοντέλο μας παρατηρώντας ταυτόχρονα τις εξής παραμέτρους : α) τις συντεταγμένες της θέσης του (x,y,z) , β) τον προσανατολισμό του (θ, φ) και γ) την ισχύ του P(t). Φτάνουμε σε έναν βέλτιστο αριθμό διπόλων και κατάλληλων τιμών παραμέτρων μέσα από μία επαναληπτική διαδικασία ούτως ώστε να ελαχιστοποιείται η συνάρτηση κόστους που κατασκευάσαμε.

Τέλος προσπαθούμε να εντοπίσουμε με όσο το δυνατόν μεγαλύτερη ακρίβεια την θέση των διπόλων εφαρμόζοντας έναν αλγόριθμο αναζήτησης βασιζόμενοι στη συνάρτηση κόστους που κατασκευάσαμε παραπάνω. Σε κάθε ανεξάρτητη συνιστώσα που έχουμε εντοπίσει αντιστοιχίζουμε ένα δίπολο και μία ενεργή περιοχή στον εγκέφαλο. Ο αλγόριθμος που θα χρησιμοποιήσουμε για την αναζήτηση της θέσης είναι ο downhill simplex ή Nelder – Mead method. Αυτός ο αλγόριθμος κάνει αναζήτηση βελτιστοποιώντας σε κάθε βήμα τις τιμές της συνάρτησης κόστους η οποία μπορεί να γραφεί σε πιο κομψή μορφή ως:

$$R(x,y,z) = \| \varphi - p_x \hat{\varphi}_x - p_y \hat{\varphi}_y - p_z \hat{\varphi}_z \|$$

Όπου αναλύσαμε την ισχύ του διπόλου σε τρεις συνιστώσες. Αυτή η εξίσωση εξαρτάται μόνο από τις συντεταγμένες. Αυτό που κάνει ο αλγόριθμος simplex είναι να προσεγγίζει εκείνες τις συντεταγμένες που θα ελαχιστοποιήσουν την παραπάνω συνάρτηση. Επειδή η αναζήτησή μας γίνεται στις τρεις διαστάσεις ο αλγόριθμός μας θα έχει τη μορφή τετραέδρου που ορίζεται από τέσσερα διανύσματα όπως φαίνεται και στην παρακάτω εικόνα:





Παρατηρούμε πως το τετράεδρο προσεγγίζει το κόκκινο διάνυσμα που αντιπροσωπεύει την θέση του διπόλου από το ευθύ πρόβλημα.

## Εφαρμογές

Γενικά το Ηλεκτροεγκεφαλογράφημα (EEG ή ΗΕΓ) χρησιμοποιείται σε διάφορους επιστημονικούς τομείς όπως στη νευροεπιστήμη, στη γνωσιακή επιστήμη (cognitive science), στη νευρογλωσσολογία καθώς και σε άλλους παρεμφερείς τομείς.

Φυσικά οι κυριότερες εφαρμογές του ΗΕΓ είναι οι ιατρικές χρήσεις του. Ενδεικτικά αναφέρουμε τη χρήση του για τη μελέτη της λιποθυμίας (“The Role of EEG in Patients With Syncope”, Dantas, 2012), της ημικρανίας (“Migraine detection through spontaneous EEG analysis”, R. Bellotti, 2007), των προβλημάτων κατάποσης (“Neural and cortical analysis of swallowing and detection of motor

imagery of swallow for dysphagia rehabilitation”, H. Yang, 2016), αλλά και γενικότερα της εγκεφαλοπάθειας (“EEG Patterns and Imaging Correlations in Encephalopathy”, Kaplan, 2011).

Στην παρούσα εργασία θα επικεντρωθούμε σε δύο συγκεκριμένες εφαρμογές του αντιστρόφου προβλήματος του ΗΕΓ (με τη χρήση του ICA), αυτήν της επιληψίας, αλλά και μια πιο ιδιαίτερη εφαρμογή για τον υπολογισμό της υπνηλίας στους οδηγούς αυτοκινήτων.

### *Υπολογισμός της υπνηλίας των οδηγών αυτοκινήτων βασισμένη στο ΗΕΓ με τη χρήση ICA*

Η επιλογή της παρουσίασης της συγκεκριμένης εφαρμογής έγκειται στη διαφοροποίηση της σε σχέση με τις υπόλοιπες ιατρικές εφαρμογές και στην πιο μηχανική της σκοπιά, κάτι που αποδεικνύεται και από το ότι το άρθρο από το οποίο αντλήσαμε τις πηγές μας εκδόθηκε στο IEEE και οι συγγραφείς του είναι ηλεκτρολόγοι μηχανικοί. Με αυτόν τον τρόπο μας δίνεται η ευκαιρία να μελετήσουμε το θέμα της εργασίας μας περισσότερο από την πλευρά του μηχανικού.

Σύμφωνα με έρευνα του 2005 από το National Sleep Foundation, το 60% των ενήλικων οδηγών (περίπου 168 εκατομμύρια άνθρωποι) δηλώνουν ότι έχουν οδηγήσει ένα όχημα ενώ αισθάνονται υπνηλία και ότι πάνω από το ένα τρίτο (37% ή 103 εκατομμύρια άνθρωποι) έχουν κοιμηθεί και πάνω στο τιμόνι. Ως αποτέλεσμα των παραπάνω έχουν εκτιμηθεί 1.550 θάνατοι, 71.000 τραυματισμοί και 12.5 δισεκατομμύρια δολάρια σε υλικές ζημιές.

Αυτό μπορεί να εξηγηθεί από τα αποτελέσματα που έχει η υπνηλία ή η κούραση κατά την οδήγηση. Συγκεκριμένα προκαλεί αύξηση στο χρόνο αντίδρασης και στην επιθετική οδηγική συμπεριφορά, μείωση στην απόδοση και στην εγρήγορση, καθώς και προβλήματα με την επεξεργασία των εξωτερικών ερεθισμάτων και τη μνήμη.

Τα παραπάνω φανερώνουν την άμεση ανάγκη για τη δημιουργία ενός συστήματος αναγνώρισης της υπνηλίας και γενικότερα του αισθήματος της κούρασης των οδηγών. Ένα τέτοιο σύστημα φυσικά θα πρέπει να είναι αυστηρά πραγματικού χρόνου (hard real time system), διότι και η οδήγηση γίνεται σε πραγματικό χρόνο, συνεπώς το ίδιο θα πρέπει να ισχύει και για την ασφάλεια που θα παρέχεται για αυτήν.

Μία από τις μεθόδους που προτείνεται είναι η αξιολόγηση των επιπέδων υπνηλίας μετρώντας το οδηγικά λάθη και ταυτόχρονα τις μεταβολές στο ΗΕΓ φάσμα με στόχο τη δημιουργία ενός μοντέλου. Ένα τέτοιο σύστημα ΗΕΓ παρακολούθησης «ταλαιπωρείται» από διάφορα EEG artifacts, λόγω του θορύβου που μπορεί να προκαλείται από την κίνηση των ματιών, των μυών, των ηλεκτρονικών, ειδικά σε ένα πραγματικό περιβάλλον οδήγησης. Στο σύστημα που μελετάμε χρησιμοποιείται η τεχνική ICA για την αφαίρεση των artifacts και την εξαγωγή των γνωρισμάτων του ΗΕΓ.

Από προηγούμενες μελέτες γνωρίζουμε ότι ο ύπνος μπορεί να χωρισθεί σε 5 στάδια:

- Στάδιο 1: 5% συνολικού κύκλου, 2-7 Hz, έχουμε αργή κίνηση των οφθαλμών
- Στάδιο 2: κύματα K, χωρίς κίνηση οφθαλμών
- Στάδιο 3: κύματα δ ρυθμού
- Στάδιο 4: παρόμοιο με το στάδιο 3
- Στάδιο REM: rapid eye movement (γρήγορη κίνηση οφθαλμών), μίξη ρυθμών

Τα χαρακτηριστικά των σταδίων αυτών μπορούν να χρησιμοποιηθούν για την εξαγωγή γνωρισμάτων για τον εντοπισμό υπνηλίας στους οδηγούς.

Για τη διεξαγωγή του πειράματος που μελετάμε χρησιμοποιήθηκε ένα περιβάλλον εικονικής πραγματικότητας. Τα υποκείμενα ζητήθηκαν να οδηγήσουν στο κέντρο μίας εικονικής λωρίδας και ως λάθος οδήγησης ορίστηκε μία σχέση με την απόκλιση από το κέντρο της λωρίδας που μετρήθηκε με τη χρήση ενός VR προγράμματος. Για το ΗΕΓ χρησιμοποιήθηκαν 2 κανάλια, 10-20 σύστημα, ειδικό gel για την αύξηση της αγωγιμότητας των ηλεκτροδίων και 16-bit κβαντιστής. Για

την προεπεξεργασία των ΗΕΓ δεδομένων, τα σήματα πέρασαν από ένα χαμηλοπερατό φίλτρο. Χρησιμοποιήθηκε επιπλέον ανάλυση εικόνας και βίντεο τη στιγμή της απόκλισης από το κέντρο της λωρίδας σε συνδυασμό με δεδομένα του ΗΕΓ για την επιβεβαίωση των συμπερασμάτων. Συλλέχθηκαν 33 σήματα καναλιού ΗΕΓ και οι αποκλίσεις οδήγησης από 45λεπτες συνεδρίες.

Η μέθοδος ICA χρησιμοποιήθηκε για την αφαίρεση διάφορων artifacts όπως θόρυβο των ματιών και της καρδιάς. Στη συνέχεια, υπολογίσθηκε η κανονικοποιημένη, κινούμενου μέσου, λογαριθμική υπο-μπάντα του φάσματος ισχύος των 33 συνιστωσών του ICA. Οι συντελεστές συσχέτισης μεταξύ του σφάλματος οδήγησης των ατόμων και του φάσματος ισχύος όλων των συνιστωσών ICA σε κάθε μία ζώνη συχνοτήτων χρησιμοποιήθηκαν για να δημιουργηθεί το φάσμα συσχέτισης. Οι 2 συνιστώσες του ICA με τους υψηλότερους συντελεστές συσχέτισης επιλέχθηκαν ως χαρακτηριστικά εισόδου του μοντέλου γραμμικής παλινδρόμησης για την εκτίμηση της απόδοσης οδήγησης των υποκειμένων.

Τελικά, παρατηρήθηκε μία στενή σχέση μεταξύ των διακυμάνσεων στην οδήγηση και στο λογαριθμικό φάσμα ICA/EEG. Η σχέση αυτή φαίνεται να είναι σταθερή στα άτομα σε διαφορετικές συνεδρίες, αλλά διαφέρει από άτομο σε άτομο. Η μέση ακρίβεια κατά τη διάρκεια μίας συνεδρίας και από μία συνεδρία σε μία άλλη για πέντε υποκείμενα ήταν 86,2% και 88,2% αντίστοιχα, ποσοστά που φαίνονται πολλά υποσχόμενα.

### *ICA αποσύνθεση ΗΕΓ σημάτων για επεξεργασία με fMRI σε περιστατικά επιληψίας*

Η επιληψία είναι μια πάθηση του εγκεφάλου που χαρακτηρίζεται από κρίσεις κινητικών, αισθητικών, ψυχικών και άλλων εκδηλώσεων. Οι επιληπτικές κρίσεις είναι επεισόδια τα οποία μπορεί να κυμαίνονται από μικρές και σχεδόν μη εντοπίσιμες περιόδους, μέχρι μεγάλες περιόδους έντονου τρέμουλου. Αυτά τα επεισόδια μπορεί να προκαλέσουν διάφορους τραυματισμούς στους πάσχοντες από επιληψία και να θέσουν τη ζωή τους σε κίνδυνο. Οι περισσότερες αιτίες της

επιληψίας είναι άγνωστες και η καλύτερη κατανόηση και μελέτη της ασθένειας αποτελεί μία άκρως ενδιαφέρουσα πρόκληση.

Στη συγκεκριμένη υλοποίηση αναλύεται μια διαφορετική προσέγγιση που επεξεργάζεται ταυτόχρονα ΗΕΓ και fMRI (functional Magnetic Resonance Imaging) δεδομένα (με καταγραφή του ΗΕΓ μέσα στο MR). Η μέθοδος βασίζεται στην αποσύνθεση των ΗΕΓ σημάτων χρησιμοποιώντας ICA για να οριστεί ένα μοντέλο για την ανίχνευση των περιοχών που απεικονίζονται στο fMRI και σχετίζονται με interictal περιόδους. Ο όρος interictal αναφέρεται στην περίοδο μεταξύ των επεισοδίων επιληψίας. Η interictal περίοδος χρησιμοποιείται συχνά για τη διάγνωση της επιληψίας μιας και ένα ΗΕΓ του ασθενή εμφανίζει interictal spikes και άλλες ανωμαλίες.

Γενικά έχει παρατηρηθεί καλή συσχέτιση μεταξύ του σήματος fMRI και του LFP μέσα στον εγκέφαλο, τα οποία συνδέονται με τα δυναμικά EEG που παρατηρούνται στο τριχωτό της κεφαλής.

Σε αυτή τη μελέτη, το ICA αξιολογήθηκε ως μια μέθοδος για τον προσδιορισμό της ηλεκτρικής δραστηριότητας των spikes, των ριπών κορυφών και των αργών μεταβολών κυμάτων με διαφορετικά μήκη και διάρκειες. Το ICA χρησιμοποιείται ήδη για την αφαίρεση artifacts λόγω θορύβου από την κίνηση των ματιών και των μυών, αλλά λόγω του μεγάλου πλάτους της interictal δραστηριότητας μπορεί να χρησιμεύσει και για τον εντοπισμό interictal φαινομένων.

Η μελέτη πραγματοποιήθηκε χρησιμοποιώντας ένα σύστημα ΗΕΓ 32 καναλιών συμβατό με MRI στα 2048Hz και έναν 1.5 T MRI scanner. Επιλέχθηκαν 15 ασθενείς που είχαν πολύ υψηλούς δείκτες interictal spike δραστηριότητας σε παλαιότερες συνεδρίες και μόνο 10 από αυτούς επέδειξαν αρκετή interictal δραστηριότητα κατά τη διάρκεια της μελέτης για να είναι ικανό το fMRI να εφαρμοσθεί. Τα δεδομένα του ΗΕΓ επεξεργάσθηκαν με τη χρήση ενός Matlab toolbox, του EEGLAB και περάστηκαν από ζωνοδιαβατό φίλτρο. Η ανάλυση των συνιστωσών του ICA πραγματοποιήθηκε με τη συνάρτηση infomax της Matlab, του EEG toolbox.

Εξετάστηκαν δύο μοντέλα:

α. Ένα μοντέλο βασισμένο στα spikes και σε ριπές (bursts) με σταθερό πλάτος και σταθερό μήκος.

β. Ένα μοντέλο βασισμένο σε μία ή δύο συνιστώσες του ICA που συνεισφέρουν περισσότερο στο να εξηγούν τη δραστηριότητα των κρίσεων.

Το επόμενο βήμα μετά τη συλλογή των αποτελεσμάτων και την επεξεργασία τους είναι η δημιουργία ενός μοντέλου για την fMRI επεξεργασία, βασισμένο στη δραστηριότητα της συνιστώσας που βρέθηκε να εξηγεί καλύτερα τις κρίσεις επιληψίας.

Η εφαρμογή αυτή παρατηρήθηκε ότι έχει υψηλή ευαισθησία στον εντοπισμό περιοχών του εγκεφάλου που σχετίζονται με interictal επιληπτική δραστηριότητα. Το γεγονός ότι η προσέγγιση αυτή είναι βασισμένη στα δεδομένα (data driven) την καθιστά πιο ανθεκτική σε συστηματικά σφάλματα που προέρχονται από την ταξινόμηση ή κάποιο σταθερό μοντέλο για τον εντοπισμό των κρίσεων. Το πρόβλημα με μία προσέγγιση που είναι αμιγώς βασισμένη στα δεδομένα είναι η έλλειψη αιτιότητας στο υπόβαθρο, δηλαδή η μη ύπαρξη ενός ουσιαστικού μοντέλου (λ.χ. υπερ-εκπαίδευση ή overtraining). Συνεπώς, η παραπάνω μέθοδος επωφελείται από το συνδυασμό των δύο ειδών σημάτων. Αν και βασίζεται στα δεδομένα από την πλευρά του ΗΕΓ, σχετίζεται και με μία ανάλυση βασισμένη σε κάποιο μοντέλο λόγω των fMRI δεδομένων.

### *Μελλοντικές εφαρμογές*

Στην εποχή που έννοιες όπως το IoT (Internet of Things) και οι ενσωματωμένες συσκευές βρίσκονται σε πρώτο πλάνο είναι ενδιαφέρον να δούμε πως αυτό επηρεάζει το ΗΕΓ γενικότερα. Σήμερα λοιπόν, πραγματοποιείται έρευνα για να γίνουν οι συσκευές ΗΕΓ μικρότερες και φορητές ("Wearable EEG"). Μεγάλη σημασία προς αυτήν την πορεία έχει η δημιουργία συσκευών χαμηλής ενέργειας που μεταδίδουν όχι όλα τα δεδομένα που συλλέγουν, αλλά μόνο εκείνα που κρίνονται διαγνωστικά χρήσιμα. Με αυτόν τον τρόπο ο ασθενής θα μπορεί να φοράει τη συσκευή ενδεχομένως για μέρες και μέσω ενός συστήματος επίβλεψης να γίνουν βήματα προς τη καλύτερη διάγνωση και αντιμετώπιση των σχετικών νόσων ("A 950 nW Analog-Based Data Reduction Chip for Wearable EEG Systems in Epilepsy", Saam Iranmanesh, 2017).

## Βιβλιογραφικές Αναφορές

- [1] **Leonid Zhukov, David Weinstein, Chris Johnson**: Independent Component Analysis For EEG Source Localization In Realistic Head Models. *IEEE Engineering in Medicine and Biology Magazine*, 2000
- [2] **Julie Onton, Marissa Westerfield, Jeanne Townsend, Scott Makeig**: Imaging human EEG dynamics using independent component analysis. *Neuroscience & Biobehavioral Reviews Elsevier*, 2006
- [3] **Tzyy-Ping Jung, Scott Makeig, Te-Won Lee, Martin J. McKeown, Glen Brown, Anthony J. Bell, Terrence J. Sejnowski**: INDEPENDENT COMPONENT ANALYSIS OF BIOMEDICAL SIGNALS. *Proceedings of the 2nd International Workshop on Independent Component Analysis and Blind Signal Separation*, 633-44, 2000
- [4] **Νικόλαος Τερζόπουλος** : Υλοποίηση λύσης αντίστροφου προβλήματος ηλεκτροεγκεφαλογραφήματος σε πραγματικό χρόνο. *Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης Ιούνιος 2018*
- [5] **Stephen Roberts, Richard Everson**: Independent Component Analysis Principles and Practice. *Cambridge University* 2001
- [6] **Γεώργιος Δ. Σεργιάδης**: Βιοϊατρική Τεχνολογία. *University Studio Press - Θεσσαλονίκη* 2009
- [7] **Chin-Teng Lin, Tzyy-Ping Jung**: EEG-based drowsiness estimation for safety driving using independent component analysis. *Circuits and Systems I: Regular Papers, IEEE Transactions on* · January 2006
- [8] **Jose P. Marques, Jose Rebola, Patricia Figueiredo, Alda Pinto, Francisco Sales, and Miguel Castelo-Branco**: ICA Decomposition of EEG signal for fMRI Processing in Epilepsy. *Human Brain Mapping* 30:2986–2996 (2009)

## Παράρτημα (Πηγαίος Κώδικας Matlab)

Παρατίθεται ο πηγαίος κώδικας Matlab που αποτελείται από συναρτήσεις ή scripts. Η λειτουργία του εξηγείται στα σχόλια του κώδικα ή αναλυτικότερα στο πρώτο κομμάτι της εργασία πιο πάνω.

### Θέμα 1<sup>ο</sup>

#### Ερώτημα 1.1

erotima1\_1.m:

```
function erotima1_1(data)
    x = 1:10000;           % x coordinated of plot

    y = data(:,1:10000);   % y coordinates of plot

    plot(x,y);
```

#### Ερώτημα 1.2

spikenum.m:

```
function num = spikenum(data,k)
    s = median(abs(data)) / 0.6745; % calculatin of
standard deviation of noise
    T = k*s;                        % T: the threashold
    len = length(data);
    num = 0;                        % initializing
spikes to zero
    i = 1;                          % i variable used
to parse through the data
    flag = 1;                       % flag used to spot
the correct # of spikes
```



```
while i <= len
    if data(i) > T && flag ==1
        num = num + 1;
        flag = 0;
    elseif data(i) < T
        flag = 1;
    end
    i = i + 1;
end
```

erotima1\_2.m:

```
k = 0:0.01:10;      % x coordinated of plot

y = zeros(1,1000);  % preallocating memory, the code
                    % runs much faster because there is
                    % no need to repeatedly reallocate
                    % memory for the growing data structure

z = 1;              % initial value of counter

for i = k
    y(z) = spikenum(data8,i);
    z = z + 1;
end

plot(k,y);
xlabel('k')
ylabel('# of spikes')
```

## Ερώτημα 1.3

pol.m:

```
function y = pol(x)

%Coefficients:
    p1 = -122.8;
    p2 = 108.1;
    p3 = -10.256;
    p4 = -13.068;
    p5 = 5.5507;

y = p1*x^4 + p2*x^3 + p3*x^2 + p4*x + p5;
```

## Θέμα 2°

### Ερώτημα 2.1

spikedet.m:

```
function spikeTimeEst = spikedet(data,k)
    s = median(abs(data)) / 0.6745; % calculation of
standard deviation
    T = k*s; % T: the threshold
    len = length(data);
    spikeTimeEst = zeros(1,len); % preallocating
memory
    num = 0; % initializing
spikes to zero
    i = 1; % i variable used
to parse through the data
    flag = 1; % flag used to spot
the correct # of spikes
```

```

while i <= len
    if data(i) > T && flag ==1
        num = num + 1;
        spikeTimeEst(num) = i;
        flag = 0;
    elseif data(i) < T
        flag = 1;
    end
    i = i + 1;
end
spikeTimeEst = spikeTimeEst(1:num); % removes the
rest of non used array

```

## Ερώτημα 2.2

isolate.m:

```

function SpikeEst = isolate(data, spikeTimesEst)
    len = length(spikeTimesEst); % length of
spikeTimeEst
    i = 1; % initializing
counter
    SpikeEst = zeros(len, 64); % preallocating 2d
matrix
    hold on;

    while i <= len
        temp = data(spikeTimesEst(i)-
32:spikeTimesEst(i)+31); % finding min and max indexes
in data for the

% 64 samples window
        maximum = max(temp);
        max_i = find(temp == maximum);
% index (in the temp array) of max
        minimum = min(temp);
    end
end

```

```
        min_i = find(temp == minimum);  
% index (in the temp array) of min  
        first = min(max_i,min_i);  
% index (in the temp array) of the first extremum  
        first_val = temp(first);  
% value of the first extremum  
        index = find(data == first_val);  
% index (in the data array) of the first extremum  
  
        SpikeEst(i,:) = data(index - 22:index + 41);  
% chose 64 sample window  
  
% waveforms are alligned with index (the first  
extremum)  
        i = i + 1;  
    end
```

erotima2\_2.m:

```
x = 1:1:64;  
SpikeEst = isolate(data1,spikeTimesEst1);  
len = length(spikeTimesEst1); % length of  
spikeTimeEst  
  
for i = 1:len  
    plot(x,SpikeEst(i,:), 'color', 'blue');  
    i = i + 1;  
end
```

## Ερώτημα 2.3

erotima2\_3.m:

```

spikeTimesEst = spikeTimesEst1;
spikeTimes = spikeTimes1;

len = length(spikeTimesEst);    % length of spikeTimeEst
len2 = length(spikeTimes);      % length of spikeTimes
displacement = 40;               % how much shifting of
the two spikes in time we consider to be the same spike
correct = 0;                     % number of correct
spikes detected
false = 0;                       % number of wrong spikes
detected
y = 1;
init = 1;                         % where to start
checking from in spikeTimes
index_pos_real = zeros(1,len2);  % the index
position of the spikeTimes array, where a spike was
correctly found
index_pos_est = zeros(1,len);    % the index
position of the spikeTimesEst array, where a spike was
correctly found
recorded_shift = zeros(1,len);   % what was the
displacement/shift for the different samples
flag = 1;
for i = 1:len
    while y <= len2
        if abs(spikeTimesEst(i) - spikeTimes(y)) <
displacement && index_pos_real(y) ~= 1
            correct = correct + 1;
            index_pos_real(y) = 1;        % denotes which
particular spike was found (from spikeTimes)
            index_pos_est(i) = 1;        % denotes if
particular spike was correctly found (from
spikeTimesEst)
            recorded_shift(correct) = spikeTimesEst(i)
- spikeTimes(y);
            break;

```

```
        else
            y = y + 1;
        end
    end
    y = 1;
    i = i + 1;
end

false = len - correct;

fprintf('The number of correctly detected spikes is:
');
correct
fprintf('\n');
fprintf('The number of falsely detected spikes is: ');
length(find(index_pos_est == 0))           % or just false
fprintf('\n');
```

## Ερώτημα 2.4

std\_calc.m:

```
%SpikeEst = isolate(data1, spikeTimesEst1);
cols = length(SpikeEst(1,:));
rows = length(SpikeEst(:,1));
SpikeStd = zeros(rows,1);

for i = 1:rows
    SpikeStd(i) = std(SpikeEst(i,:));           % find the
standard deviation of spike array
end
```

slope\_calc.m:

```
%SpikeEst = isolate(data1, spikeTimesEst1);
cols = length(SpikeEst(1,:));
rows = length(SpikeEst(:,1));
SpikeSlope = zeros(rows,1);

for i = 1:rows
    maximum = max(SpikeEst(i,:));
    max_i = find(SpikeEst(i,:) == maximum);
% index (in the temp array) of max
    minimum = min(SpikeEst(i,:));
    min_i = find(SpikeEst(i,:) == minimum);

    SpikeSlope(i) = (maximum - minimum) / (max_i -
min_i); % find the slope between the two extremums
end
```

scat\_plot.m:

```
function scat_plot(Char1, Char2, spikeClass,
index_pos_est, index_pos_real)
    %erotima2_3; % run the
script with the corresponding spikeTimesEst, spikeTimes
% to get
the index_pos_est, index_pos_real variables
hold on

    Char1Match = transpose(Char1) .* index_pos_est;
% to only get the correctly found spikes
    Char2Match = transpose(Char2) .* index_pos_est;
    spikeClassMatch = spikeClass .* index_pos_real;

    empty1 = find(Char1Match == 0);
% finding zeros to get rid of the spikes that are false
    empty2 = find(spikeClassMatch == 0);

    Char1Match(empty1) = [];
% getting rid of the spikes that are false
    Char2Match(empty1) = [];
```

```

spikeClassMatch(empty2) = [];

indexes1 = find(spikeClassMatch == 1);
% finding at which indexes there are spikes of class 1,
class 2, class 3
indexes2 = find(spikeClassMatch == 2);
indexes3 = find(spikeClassMatch == 3);

k = 1;

for i = indexes1
% saving the spikes of only one neuron that are correct
according to spikeTimes
    char1_c1(k) = Char1Match(i);
    char2_c1(k) = Char2Match(i);
    k = k + 1;
end

scatter(char1_c1, char2_c1, '*', 'blue')

%-----
k = 1;

for i = indexes2
    char1_c2(k) = Char1Match(i);
    char2_c2(k) = Char2Match(i);
    k = k + 1;
end

scatter(char1_c2, char2_c2, '*', 'red')

%-----
k = 1;

for i = indexes3
    char1_c3(k) = Char1Match(i);
    char2_c3(k) = Char2Match(i);
    k = k + 1;
end

```



```
scatter(char1_c3, char2_c3, '*', 'yellow')
```

## Ερώτημα 2.5

zero\_crossings.m:

```
%SpikeEst = isolate(data1, spikeTimesEst1);  
cols = length(SpikeEst(1,:));  
rows = length(SpikeEst(:,1));  
Zeros = zeros(rows,1);  
  
for i = 1:rows  
    Zeros(i) = spikenum(SpikeEst(i,:),0);           % find  
    how many times each spike has crossed zero  
end
```

energy\_fft\_calc.m:

```
%SpikeEst = isolate(data1, spikeTimesEst1);  
cols = length(SpikeEst(1,:));  
rows = length(SpikeEst(:,1));  
SpikeEnergy = zeros(rows,1);  
  
for i = 1:rows  
    F = fft(SpikeEst(i,:));  
    pow = F.*conj(F);  
    SpikeEnergy(i) = sum(pow) / cols;           % find how much  
    energy the spike has with DFT  
end
```

max\_ind\_calc.m:

```
%SpikeEst = isolate(data1, spikeTimesEst1);
cols = length(SpikeEst(1,:));
rows = length(SpikeEst(:,1));
SpikeMaxInd = zeros(rows,1);

for i = 1:rows
    maximum = max(SpikeEst(i,:));
    max_i = find(SpikeEst(i,:) == maximum);
% index (in the temp array) of max
    minimum = min(SpikeEst(i,:));
    min_i = find(SpikeEst(i,:) == minimum);

    if max_i < min_i
        max_first = 1;
% finds where max is first
    else
        max_first = 0;
    end

    SpikeMaxInd(i) = max_first;
end
```

width\_calc.m:

```
%SpikeEst = isolate(data1, spikeTimesEst1);
cols = length(SpikeEst(1,:));
rows = length(SpikeEst(:,1));
SpikeWidth = zeros(rows,1);

for i = 1:rows

    maximum = max(SpikeEst(i,:));
    max_i = find(SpikeEst(i,:) == maximum);
% index (in the temp array) of max
    minimum = min(SpikeEst(i,:));
    min_i = find(SpikeEst(i,:) == minimum);
```

```
SpikeWidth(i) = abs(max_i - min_i);  
end
```

getting\_ready.m:

```
Char1Match = transpose(SpikeSlope) .*  
index_pos_est; % to only get the correctly found  
spikes  
Char2Match = transpose(SpikeStd) .* index_pos_est;  
Char3Match = transpose(Zeros) .* index_pos_est;  
Char4Match = transpose(SpikeEnergy) .*  
index_pos_est;  
Char5Match = transpose(SpikeMaxInd) .*  
index_pos_est;  
Char6Match = transpose(SpikeWidth) .*  
index_pos_est;  
spikeClassMatch = spikeClass4 .* index_pos_real;  
  
empty1 = find(Char1Match == 0);  
% findind zeros to get rid of the spikes that are false  
empty2 = find(spikeClassMatch == 0);  
  
Char1Match(empty1) = [];  
% getting rid of the spikes that are false  
Char2Match(empty1) = [];  
Char3Match(empty1) = [];  
Char4Match(empty1) = [];  
Char5Match(empty1) = [];  
Char6Match(empty1) = [];  
  
spikeClassMatch(empty2) = [];  
  
group = transpose(spikeClassMatch);  
  
T1 = transpose(Char1Match);  
T2 = transpose(Char2Match);  
T3 = transpose(Char3Match);
```

```
T4 = transpose(Char4Match);  
T5 = transpose(Char5Match);  
T6 = transpose(Char6Match);  
Data = [T1 T2 T3 T4 T5 T6];
```