# Seminar Hands-on BCI implementation

## Session 1: Introduction to the data structure and variables

This seminar is based on data from an experiment in which epilepsy patients with subdurally implanted electrode arrays (ECoG) performed different hand gestures to control an avatar through a virtual reality. The aim of the original project was to detect the gestures from the recorded brain signals. For further information, please read 'Experimental paradigm and data structure.pdf'.
For this seminar we took one of these gestures (wiggling with the index finger) and divided it in two antagonistic parts (flexion and extension).

## Test glove data set

To make you more familiar with the data you get from a sensor glove we will first look at a test glove data set, that was recorded in order to gain information about the time course of each sensor while performing a particular gesture.

```
Load test data (consisting of the variable gloveResamp)
load gloveResamp_Dav_2011_9_7_14_28_21.mat

now plot the test glove data
plotRawGesture(gloveResamp)
```

Upper subplot: data from the sensors of the glove (For us the blue trace is the sensor of interest, as it indicates index finger wiggling)
Lower subplot: Accelerometer values (Cartesian coordinates)

### TASK 1 (1 pt):

Try to find the interval in which the subject did the finger wiggling gesture. There are four other gestures (See 'Experimental paradigm and data structure.pdf'). Can you discriminate them?
Write down the gestures you can identify and their approximate time.

```
Close the figure and delete the variable gloveResamp, as you will not need it
anymore.
clear gloveResamp
```

# Analyzing the glove data of our subject

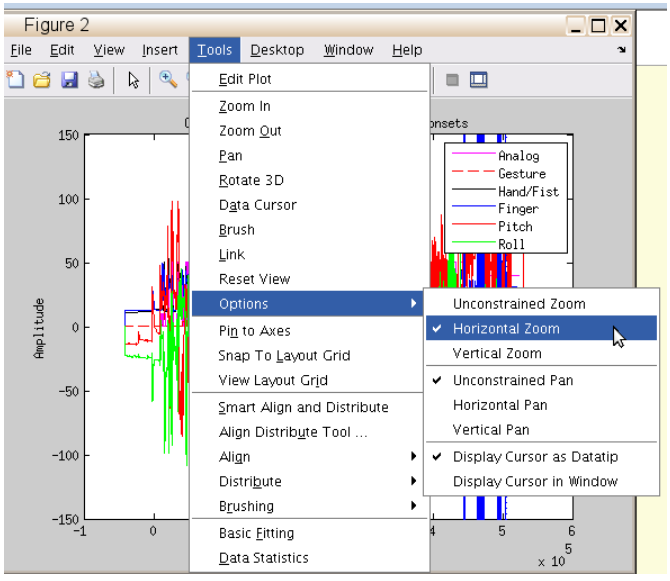
Now continue with the data from our subject:
First load the following data files to your workspace:

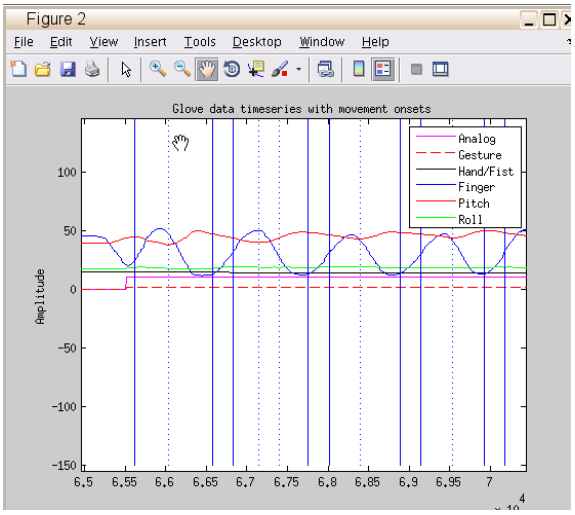
```
load gloveResamp.mat  % Motion data from the sensor glove

load ecogAnalog.mat % contains data from an analog channel to synchronize
brain and glove data (it is important to synchronize when using the glove data
to find the gesture onsets, that are later used to analyze the brain data)

load epoch.mat % provides onsets and labels of gestures (hand labeled)
```

The gesture onsets and labels contained in the variable epoch were found by hand labeling.
In Task 2 you will try this hand labeling yourself.
BEFORE you start with the task, plot the hand labeled data saved in epoch, so you know how it will look like when finished (We will plot the glove data time series with onsets of flexion and extension of the index finger.)

```
plotGloveData(gloveResamp,ecogAnalog,epoch)
```

As you can see it is hard to recognize anything in this figure. It is best to zoom in using a **horizontal zoom** (to change to this option go to Tools -> Options -> Horizontal Zoom). Then click on the zoom-in symbol (magnifying glass) and choose a time window in your figure. If you want to scroll to other time windows, click on the hand symbol (scrolling).



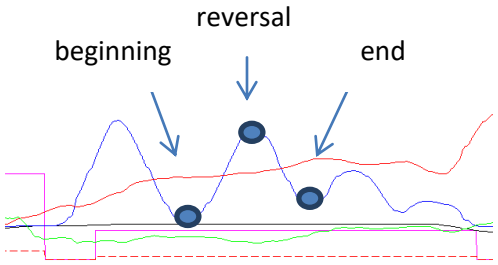




The vertical lines represent the onsets of the epochs (normal blue lines = flexion epochs; dashed blue lines = extension epochs) The epochs will be 0.25 seconds long. If the time of flexion or extension is longer than 0.5 seconds than two epochs will be fitted within one gesture.
For better understanding of how the onsets are calculated, have a look at the getEpochs function.
Now close this figure again and try finding your own epochs.

## TASK 2:

(Remember the following steps are just to give you an idea of the procedure used to cut and label the data. Afterwards we will continue using the onsets and labels already saved in epoch.)

In order to cut and label the data the function getEpochs is used.
This function first asks you for graphical input, that you will provide by mouse clicks on the figure that will appear. (To do this getEpochs uses the function ginput; for more information type help ginput).
To make own epochs select the beginning, the reversal point and the end of one gesture cycle (flexion + extension) with the mouse (crosshairs will appear to make this easier).




Mark always three points otherwise data are rejected.
From this input the function then calculates the flexion and extension onsets seen above.
After finishing one gesture, you will be asked if you want to continue with a next gesture, type either 'y'(es) to continue or 'n'(o) to end process.
IMPORTANT is to optimize your figure size (horizontal zoom like described above) before starting with the first gesture.
Try it out for at least five gestures.

```
ownEpochs = getEpochs(gloveResamp,ecogAnalog,0.25); % the third input (here 0.25) is the desired epoch length in seconds

after finishing you can plot your new epochs using 'plotOwnLabels'
plotOwnLabels(gloveResamp,ecogAnalog,ownEpochs)
```

## TASK 2.1 (1 pt):

Find the onsets of the 5 gestures you defined. Copy them in here:

## TASK 3 (Discussion):

Check if only the index finger moved in the periods you labeled.
Think about the appropriateness of the label if more than one finger moved.

```
close the figure
close all
```

## Anatomy

Now have a short look at the anatomy of our subject
Raw ECoG data were recorded from a 16x16 electrode grid with a center-to-center spacing of 0.4 cm. To reduce the amount of data, only 100 of the 256 electrodes available, will be used in this course.

Plot anatomical image of brain and electrodes

### TASK 4 (1 pt):

Figure out how to read in and plot the anatomical data in Matlab.

```
Anatomy = 'GP33_anatomy_100electrodes.png';
```

close the figure

## ECoG data

The data is saved in a structure called ecog.
Most of the functions we use in this seminar come from our ecog toolbox and are based on this kind of data structure.

clear your workspace and then load this ecog structure

```
clear all;
load ecogStruct.mat
```

now have a look at the current state of the ecog structure

```
ecog
```

### TASK 5:

Try to understand what all the fields in the ecog structure contain.

## Preprocessing of ECoG data

The first preprocessing step is baseline correction. Here the mean across the samples within one channel is subtracted from each sample in the respective channel setting the average of each channel to zero.

### TASK 6 (Discussion):

What is the goal of the baseline correction here? Can you think of other ways of achieving this goal?

```
First plot some example data without baseline correction.
figure
subplot(2,1,1) % plotting example data before baseline correction
plot(ecog.data(1,1:1000)) % just a random channel and time points for
demonstration
title('Before baseline correction')
```

### TASK 7 (2 pt):

Implement two ways of doing the baseline correction.
First, use for-loops over channels and samples.
Second, create a vector of channel means and use the repmat function for subtracting the means from all samples simultaneously.

```
plotting example data after baseline correction
subplot(2,1,2)
plot(ecog.data(1,1:1000)) % same channel and time points as above, but now
with baseline correction
title('After baseline correction')
```

```
save the ecog structure in ecogStruct1.mat, in order to save the new baseline
corrected data for later use.
save ecogStruct1.mat ecog
```