# Sensorimotor rhythm (SMR) based BCIs

## Exercise 5.1: ERDS time-frequency maps

The aim of this exercise is to compute Event-Related-Synchronization-and-Desynchronization (ERDS) time-frequency maps for Laplacian re-referenced channels at positions C3, Cz and C4. From the computed maps we then identify most distinctive frequency bands (features), which will be used to simulate SMR BCI control.

Provided raw data:

- `sGes`: EEG signal matrix (time x channel); Laplacian data; 3 channels; positions C3, Cz and C4; 50 trials per class.
- `hGes`: header (struct); contains: `SampleRate` (Hz), `Classlabel` (1 - left hand or 2 - feet), `TRIG` (index in the data where a cue/class started)

ERDS maps:

- Calculate ERDS maps for class left hand motor imagery (MI) and feet MI separately.
- Calculate a map per channel (Laplacian C3, Cz, C4).
- Use the function `calcErdsMap` (provided for this exercise to calculate the maps).
- Settings for `calcErdsMap`: `t=[-3, 0, 5]; f_borders=[4, 30]; t_ref = [-2.5, -0.5];`
- `calcErdsMap(sGes,hGes,t,f_borders,'ref',t_ref,'sig','boot', 'alpha',0.01);`
- Use the function `plotErdsMap` (provided for this exercise to plot the maps).

**Task:**

- Compute the ERDS time-frequency maps as outlined above
- Which frequency band(s) are most different between the classes? Which band(s) would you chose? Give reasons for your decision!

## Exercise 2: BCI simulation

The aim of this exercise is to compute a SMR-BCI simulation.

**Part 1:** Train an LDA classifier using one logarithmic band power feature extracted from one Laplacian re-referenced EEG channel (data: `sGes`; header: `hGes`).

- Band pass the EEG from the selected – Exercise 5.1 – Laplacian channel to limit the frequency range to your selected range. Use the Matlab function `butter` to define a Butterworth filter (filter order: 4). Then use `filter` to apply the filter on the data.
- For each trial, extract a 2-second segment of data (1s to 3s after cue), square each sample and sum up the squared values in the segment. Subsequently, calculate the log of the sum.
- Build 2 vectors: `X_train` (trials x features) containing the feature and `Y_train` (trials x class labels) containing the class labels.
- Train an LDA classifier with `X_train` and `Y_train`.

**Part 2:** Classify unseen data with the trained LDA (data: `sGes_test`; header: `hGes_test`).

- Repeat the feature extraction as described in part 1 with the unseen test data.
- Classify the data.

**Part 3:** Repeat Part 1 and Part 2 by selecting two logarithmic band power features.

**Task:**

- Which accuracies do you achieve in Part 1, 2 and 3?

**Pre-submission via TeachCenter at the end of the unit; final submission 02.06.2019**