fNIRS Retinopy

Matlab Case Study for Signals and Systems (Draft)

Functional Near-Infrared Spectroscopy is an imaging method which measures levels of hemoglobin in various parts of the brain to detect areas of increased blood flow. Similar to other methods of brain imaging, fNIRS can be used to detect correlations between external stimuli and increased brain activity.

By providing a known visual stimulus and then measuring regions of the brain with increased blood flow, we can assemble a kind of “map” of which spatial parts of the brain correspond to which spatial parts of a subject’s field of vision.

[Visual aid: a colored map of the brain that we’ve produced of one such possible mapping]

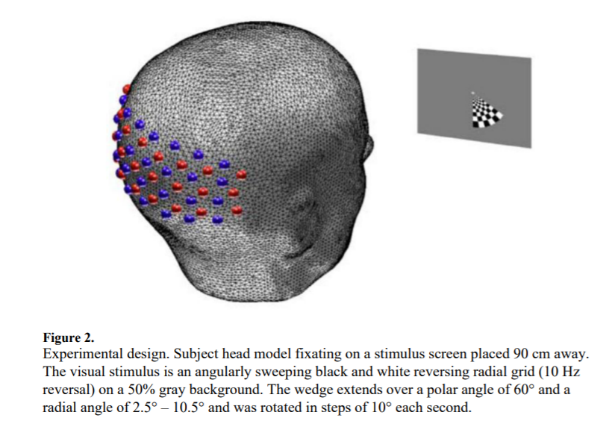
For this assignment, we will use data from an experiment by (author names here), in which subjects looked at a slowly rotating wedge as fNIRS measurements were taken. By examining the Fourier transform of the data from each sensor pair and determining the phase angle at the stimulus frequency, we hope to find a pattern between regions of the brain and the stimulus image at a particular moment in time.

In this case study, you will:

* Familiarize yourself with a MATLAB data structure containing the experiment results
* Use data manipulation techniques to examine the data in a variety of ways to determine which parts to use.
* Design a filter to remove noise from the data
* Use the Fourier transform to isolate important frequencies in the data
* Create a visualization of the data to map the brain’s response to a stimulus.

When you are finished, you will have a greater proficiency with

# Experimental Setup

****In this experiment, a subject views a checkerboard wedge that slowly rotates on the screen, completing one revolution every 36 seconds (about 0.028 Hz). Meanwhile, infrared sensors measure brain oxygenation to see which spatial regions of the brain are active as the wedge moves. Data is collected for 9 revolutions – a little over five minutes.

Each measurement is associated with two devices, a source and a detector. The source sends out different wavelengths of infrared light and the detector measures how each one is attenuated as it travels through the brain. Included in the MATLAB script is code to generate a map of each source and receiver and its physical position on the outside of the subject’s head.

A close up of a map

Description automatically generated

Each detector receives a signal from each source and measures the attenuation. With 24 sources and 28 detectors, that makes for 672 total source-detector pairs. (In the data structure provided to you, each source-detector pair is counted twice, so there are 1344 measurements listed. Part of this assignment will be learning to interpret this data structure.)

672 device pairs taking 2 datapoints 15 times per second for 5 minutes is a lot of measurements! (Around 6.5 million, in fact). But not all are equally important. Source-detector pairs that are farther away travel through more of the brain and can therefore detect more differences in oxygenation, but they will also be greatly attenuated in the process. Pairs that are closer together have less risk of being attenuated below the noise level of the sensors, but don’t travel through very much of the brain. Part of your job for the case study will be to judge which sets of source-detector pairs will yield the most valuable information about brain oxygenation levels and which should be ignored.

# Data Structure

The results of the experiment are stored in a data structure, stored in the Matlab workspace file, ‘NeuroDOT\_Data\_Sample\_CCW1.mat’. The following variables are included in the workspace:

* data, a 1344x4857 matrix that contains the raw measurements for each source-detector pair. Each row corresponds to one source-detector pair. The columns represent different samples taken at different times. The array pairs can be used to filter important
* info, a structure containing several sub-structures, each with useful information:
  + optodes, a structure containing the positions, in both 2d and 3d space, of each source and each detector
  + pairs, a structure containing information for each row of data. It records which source-detector pair created the data, how far apart in both 2d and 3d space they are, and the number of nearest neighbors.
  + paradigm, <thing here>
  + system, a structure containing the sample frequency at which the measurements were made
* flags, a structure containing additional parameters you will not need to make use of

# Referencing the Data Structure

When MATLAB data is located within a structure, you can reference it using the format “structurename.structurevalue.” For instance, if you want to retrieve the 3-D spatial coordinates of all the detectors used in the experiment, you can use the reference: info.optodes.dpos3

Conditional indexing will be very helpful for this case study. For more information on how conditional indexing works, see [this document](https://www.mathworks.com/help/matlab/matlab_prog/find-array-elements-that-meet-a-condition.html). For an example, consider the line below:

data(info.pairs.r3d < 30,:)

This command returns a subset of the data that only includes measurements from source-detector pairs that are at a distance of less than 30.

# Case Study

Open the *fNIRSCaseStudy.m* script in MATLAB and read through it, then run it. Examine the plots produced.

* The Data Traces plot overlays the signal from every sensor pair on a log scale. What do you notice about signals with a smaller amplitude? What about signals with a larger amplitude? Do you see any signals with a low frequency oscillation in them? Where do those signals appear on the data trace?
* The LFO plot shows the average amplitude of the signal from each sensor pair. The x-axis represents the distance each signal penetrates before reaching the detector. The most useful sensor data will be from sensors that have passed through enough of the brain to measure hemoglobin content, but not so much that the desired signal has been attenuated enough to be crowded out by noise. Based on the LFO plot, what range of distance do you think will be most useful for this purpose? Record your observations in your writeup.
* Use conditional indexing to take a subset of the data based on your work in the previous task.
* The Log-Ratio Signals shows each signal pair over time, with intensity mapped to color. You should see some periodic behavior on this plot. What frequency is it? What other behavior do you observe?
* Consider the experimental setup. How might we use a highpass or lowpass filter to clean up this data? Examine the highpass() and lowpass() functions included in the case study and use them to generate a new version of the Log-Ratio Signals plot that attenuates noise without attenuating the stimulus frequency of .028 Hz.
* Once you’ve chosen a reasonable subset of the detector data and filtered it to your liking, plot the magnitudes of the resulting Fourier transform. To prevent overplotting, you may want to only plot a few representative signals from the data or decrease the line width. Make some observations: where are there spikes or high amplitude components? Are some signals noisier than others? Can you spot the stimulus frequency in the data?
* You should have a