Natalia Sempere 2025

Week 23

Tongue EMG XR Project

Getting Started

- I got started with the project, read the project overview, prepared a folder and **GitHub**.
- [HyperDecahedron/tongue_emg_xr_project: Project to identify tongue's position as an XR input.](#)

Anatomy, EMG theory and Related Work

- Researched the mouth, tongue and neck anatomy to familiarise with the muscles. Noted down some muscles of interest:
    - Submental triangle
    - Sternocleidomastoideus
    - Masseter: right/left?
    - Genioglossus
    - Styloglossus
        - Pulls the tongue back and up.
        - Harder to access.
    - Hyoglossus
        - Under chin, slightly lateral.
    - Suprahyoid muscles (mylohyoid, geniohyoid)
        - Under chin, near genioglossus.
    - Forehead for teeth clench
- I read [SEMGIntro.fm](#) to learn about EMG basics.

- Read the work by Zhang 2014 (TongueSee), and noted down important information for the online classification
    - 2048 Hz
    - Window of 128 samples
    - RMS of the window.
    - RMS of signed difference between signal value and mean value.
    - Majority voting

Testing OpenBCI Cyton board

- **Muscles work up to 500 Hz, but the board samples at 250 Hz. Zhang2014 employs 2048 Hz. We might have to increase the sampling rate.**
- Installed drivers and GUI
- Tested the signal in several muscles and selected those that worked better.
    - All of them: very sensitive when I move my head!
    - Forehead for teeth clench VS Masseter
        - **Forehead much better than masseter**
        - Can detect different right/left.
    - **Submental triangle left/right**
    - Sternocleidomastoideus
        - Clear signal, no difference left/right
    - Genioglossus & Styloglossus
        - **Genioglossus: same as submental**
        - Styloglossus: very light signal when going left/right
    - Simple tongue touch
        - Useless
    - Hyoglossus
        - Good if touching the last molars
- Tested board settings
    - **Set SRB2 off**
    - **Creep-: default value is the best.**

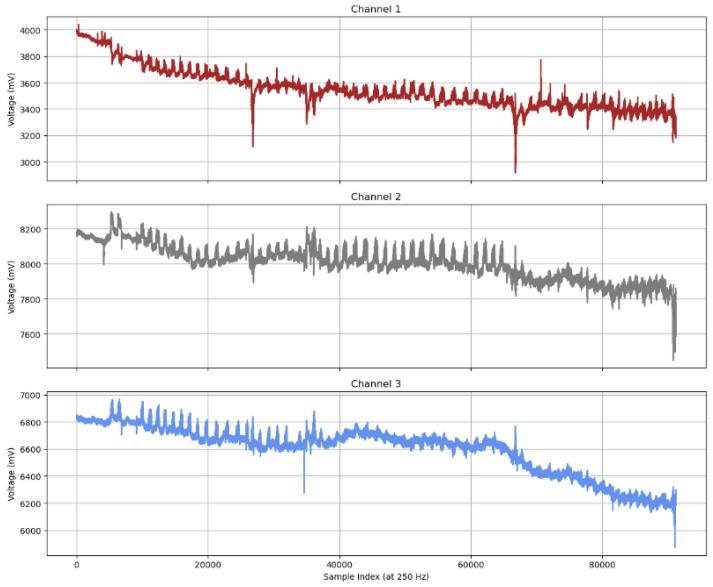
First Recording (02/06/2025)

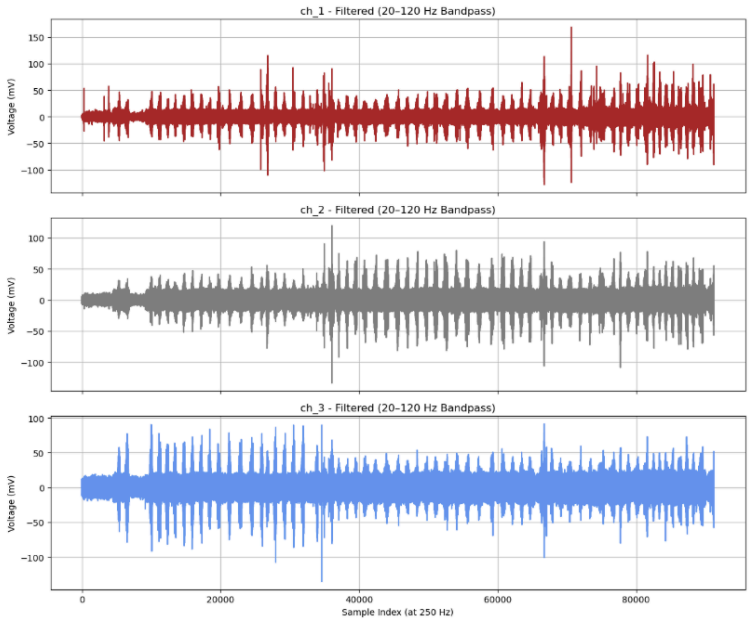
- Created a key logger file (IDE Visual Studio) that runs via an Anaconda Prompt.
- 3 classes (left, right, front)
- First I press the class on the keyboard, then I move the tongue.

- Channels:
    - 3x channels in submental triangle.
    - Channel 1: Submental center
    - Channel 2: Submental right
    - Channel 3: Submental left

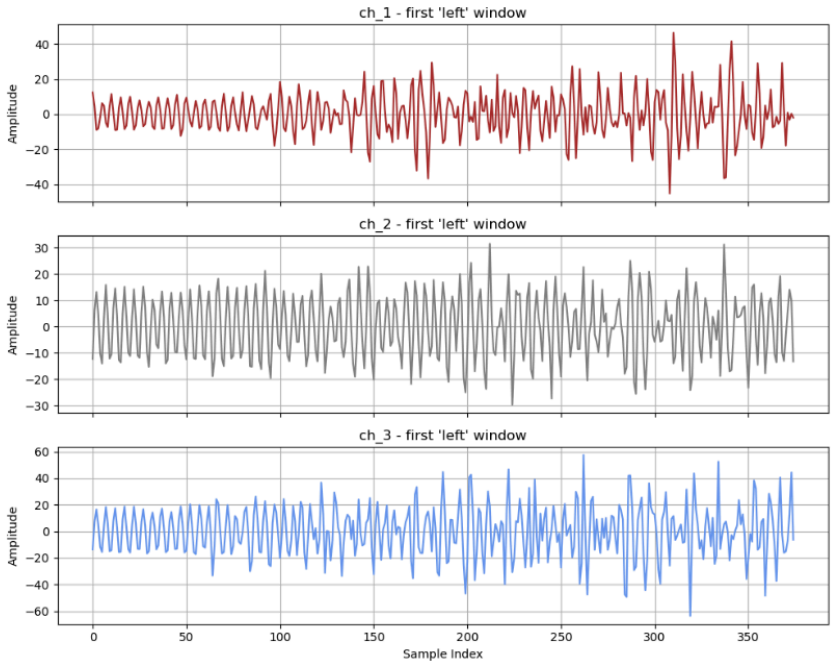
First Data Analysis

- **Filtering**
    - Bandpass filter with 20-100 Hz. Can't raise the upper bound because of Nyquist.





- **Windows**
  - I used the timestamps from key_logger to detect the windows.
  - Window size: 1.5s



- **Feature extraction**
  - Added RMS, ZC and WL.
- **Classification**
  - SVM
    - Best results with linear kernel
    - **Accuracy: 0.94 (left, right, front)**
  - Random Forest Classifier
    - Changing the number of classificators doesn't affect the result
    - Accuracy: 0.89
- **Fine tuning**
  - I created a list of variables to check for the next recording.
    - Recording
      - Hz
      - Creep-
      - Position of the sensors
    - Filtering
      - Highcut
      - Type of filtering
    - Windowing
      - Size of the window in seconds
    - Feature extraction
      - RMS, ZC, WL
      - Anything else?
    - Classification

- SVM
    - Kernel
- Random Forest Tree

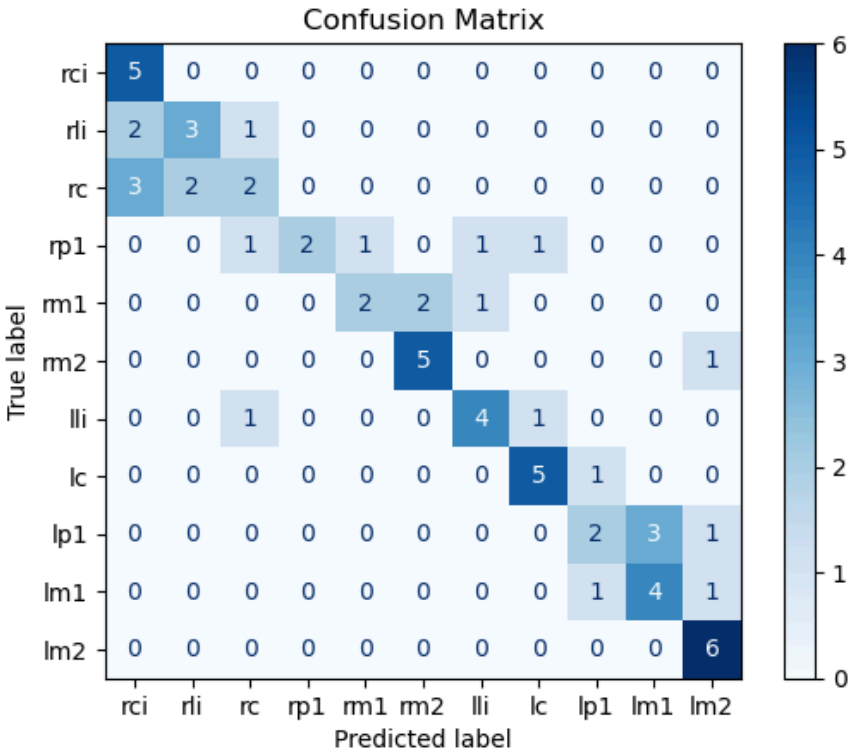
More recordings and their data analysis

- **Higher sample rate?** Not directly possible, we would need an SD card and the process is not straightforward. To not waste time, I will skip this.
- **Creep- and low limit:** not much change, will use default values.
- **Second Recording**
    - This recording was a repetition of the first one but with four classes instead of three (left, right, front, none).
    - Data Analysis:

Offline accuracy	Classifier	Window size	Observations
0.58	SVM	1.5	
0.96	RF	1.5	
0.62	SVM	1	
0.75	RF	1	
0.62	SVM	2	
0.92	RF	2	
0.96	**RF**	**1.5**	**Z-Score normalization.**
0.88	SVM	1.5	Z-Score, rectify, smooth
0.79	SVM	1	Z-Score, rectify, smooth
0.92	RF	1	Z-Score, rectify, smooth
0.88	SVM	2	Z-Score, rectify, smooth
0.92	RF	2	Z-Score, rectify, smooth
			Tested both of them with more features, but got less accuracy. tested features are: 'RMS', 'ZC', 'WL', 'MAV', 'SSC', 'WAMP', 'IEMG'

- **Third recording**
    - To test tooth-by-tooth precision
    - Updated the key_logger to include a class for each tooth.

- I recorded tooth by tooth, 20 times each, split in two recordings of the left and the right side.
- The accuracy was awful…
  - … but I saw that **the tooth that was more difficult to detect was the second premolar in both left and right sides…**
  - **… so I repeated the analysis with both sides together and without the second premolar.**
  - **The final offline accuracy was 0.62, which is low, but the confusion matrix is very diagonal, so it always detects the correct tooth +- 1 tooth of error.**

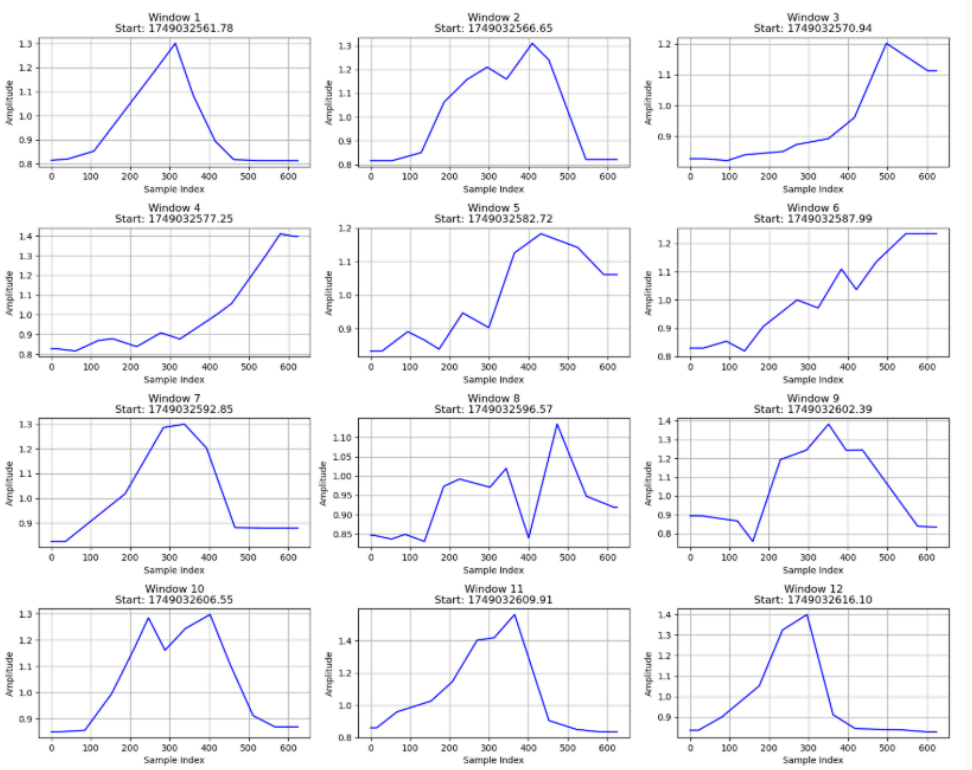


TOOTH CLASSES					
key	id	description	key	id	description
1	lci	left central incisor	a	rci	right central incisor
2	lli	left lateral incisor	s	rli	right lateral incisor
3	lc	left canine	d	rc	right canine
4	lp1	left premolar 1	f	rp1	right premolar 1
5	lp2	left premolar 2	g	rp2	right premolar 2

6	lm1	left molar 1	h	rm1	right molar 1
7	lm2	left molar 2	j	rm2	right molar 2

RIGHT + LEFT SIDES			
Accuracy	Classifier	Window size	Observations
0.52	SVM	1	all features with wamp threshold 1.1
0.51	RF	1	
			The worst tooth to detect were lp2 and rp2
0.62	**SVM**	**1**	**removing rp2 and lp2**
0.58	RF	1	

- **Fourth recording**
  - Recorded tongue pressure from light to hard and logged the timestamp with key_logger.
  - Logged in class:
    - l ascending
    - r light press
    - f hard press
  - I tried to get a smooth envelop for the ascending class without using a classifier
  - This is the best I could get for the ascending press, but will keep investigating



Natalia Sempere 2025

Real time & Unity

- **Implemented the code** to execute the filtering, windowing, feature extraction and classification **in real time.**
    - Decided to use a window size of 125 samples, but I will update it in the future.
- Implemented **UDP receiver in Unity**
    - Moves XR origin depending on the class it receives.
- Tested with the board, but without the sensors.
    - I used the classifier that I had previously trained offline in the second recording.

Next Week

- ☐ Prepare presentation Kick off
- ☐ 4 classes classifier
    - ☐ Try real-time classification of 4 classes with sensors + in Unity.
    - ☐ Online training? Get 4-5 samples to fine tune before starting the testing, is it necessary?
    - ☐ Make windows overlap?
    - ☐ Implement code to save the predicted class and timestamp, while doing a keylogger, then evaluate online accuracy.
- ☐ 6 classes classifier
    - ☐ Record more data with 6 classes. Maybe add more channels.
    - ☐ Train classifier and expand to left, right, front, back, up, down.
    - ☐ Try real-time accuracy
    - ☐ If the accuracy is bad:
        - ☐ Reconsider positioning of senors.
        - ☐ Try to increase sampling rate.
    - ☐ Validate classifier generalizability (train/test split, cross-validation)
- ☐ Think how to use those classes for navigation in XR.
- ☐ Investigate regression for analog/force estimation (e.g., pressure via RMS amplitude)