Lab3: Car Competition

Method

- **Signal**
  We have two versions of code : one incorporating EEG and eye movement signals, while the other solely relies on EEG signals, and in the demo, we used **only the EEG signal** results. However, the eye movement signal component can still be viewed in the comments within the code.

- **Preprocessing**
  Bandpass Finite Impulse Response (FIR) Filter
  We set the length of the filter to 21 and specified the cutoff frequencies of the filter in a list format as a bandpass filter, with a pass frequency range between 1.0 and 50.0.

```
FIRbp = firwin(numtaps=21, cutoff=[1.0, 50.0], fs=FS, pass_zero = 'bandpass')
```

- **Baseline design**

1. **buffer**
   Using `pull_chunk()` continuously fetches power data of EEG from the StreamInlet. It continues pulling when the number of rows in BUFFER is less than three times the sampling frequency. It utilizes the `np.vstack` function to append the new data chunk to the bottom of the BUFFER array until sufficient data is collected, that is reaching the sampling duration of 3 seconds.

```
while BUFFER.shape[0] < FS * 3:
    chunk, stamps = EEG_in.pull_chunk()
    chunk = np.array(chunk)
    if len(chunk) != 0:
        BUFFER = np.vstack([BUFFER, chunk[:, 0:ch_num]])
```

2. **baseline**
   Average the data collected over 3 seconds using the aforementioned method to establish the baseline.

```
for i in range(5):
    EEG_array = BUFFER[int((1 + FS * i * 0.5)):int((1 + FS * (i + 2) * 0.5)), 0:ch_num]
    EEG_array = EEG_array - np.mean(EEG_array, axis = 0)
```

Then accumulate the power values within the frequency range of 8Hz to 15Hz into base_alpha, then proceed similarly for base_theta (4Hz to 7Hz) and base_mu (18Hz to 22Hz) within their respective frequency ranges. Finally, normalize these values.

```
    base_alpha += bandpower_(f, Pxx, 8, 15)
    base_theta += bandpower_(f, Pxx, 4, 7)
    base_mu += bandpower_(f, Pxx, 18, 22)
```

- **Control method**

  `Frequency domain:`

  Utilize the difference between the actual received signal and the baseline for each frequency band to set thresholds, then design corresponding actions to control the car.

```
        # open/close eyes
        # Occipital alpha
        comm_rotate =  determine_stop(alpha, base_alpha)

        # attention
        # Frontal theta alpha
        comm_forward = determine_forward(alpha, theta, base_alpha, base_theta)
```

forward : focus

```
if comm_forward == 1: # focus
```

right : open eyes + distraction

```
elif (comm_forward == 2 and comm_rotate == False): # distraction
```

left : close eyes + distraction

```
elif (comm_forward == 2 and comm_rotate == True): # distraction & close eye
```

stop : other

```
else: # unknown EEG signal or noise
```

Discussion 1

(The challenges and difficulties you encountered during the process and how you solve them.)

1. EEG Signal Acquisition

   Due to unfamiliarity with the OpenVibe, which offered by TAs, at the beginning we encountered some latency issues as well as receivied blank data while attempting to receive EEG signals via OpenVibe. Hence, we had tried several settings within the OpenVibe Designer. However, the issue was resolved by opting not to use OpenVibe and directly switching to StreamInlet for signal reception.

2. Discrepancy Between Commands and Actual Actions of car.
   There could be various reasons for discrepancies such as the vehicle inexplicably stopping, potentially stemming from excessive noise, significantly higher alpha and theta waves compared to the baseline, or possible delays or high impedance issues.

   Here are the methods we had tried to figure out the problems:
   - Modified thresholds: including changes to baseline algorithms and definitions.
   - Increased the original four channels to eight channels.
   - Avoided signal input speeds that were either too fast or too slow by adjusting `sleep()`.
   - Altered actions corresponding to different commands.
   - Utilized more stable commands from the demo and excluded eye movement signals to lower the scoring criteria.

Discussion 2

(Other things you wish to discuss, such as contribution.)

1. The threshold for determining actions based on brainwave activity is hardcoded. During practical use, we often need to adjust these thresholds to fit the current situation. Given more time, exploring adaptive threshold techniques might be considered to enhance efficiency.

2. Using machine learning to classify signals: Relying solely on the magnitude differences of band power values for classification might result in insufficient accuracy. Therefore, considering the use of supervised learning to train an automated classification model in the future might lead to better control effectiveness.

3. Our Demo results was improved after removing eye movement features, but It was contrary to our expectations. We suspect that our initial reception of eye movement signals might not have been optimal, leading to the weird results.

4. Contribution
   Designing the Experiment and Method : ALL
   Coding : 彭浩宇、張詠哲
   Providing Demo EEG : 施欣妤
   Report Writing : 林曉玫