**CODE COMPREHENSION AND USER MANUAL – ASTRAL PILOT**

**User Manual for EEG-Driven Drone Control with Neuphony Headset:**

**Requirements:**

1. Neuphony EEG headset
2. Tello drone
3. Python environment installed with required libraries
4. Internet connection

**Setup:**

1. **Connect the Neuphony Headset:**
   * Wear the Neuphony EEG headset securely on your head.
   * Ensure a comfortable fit to optimize signal capture.
2. **Connect the Tello Drone:**
   * Power on the Tello drone.
   * Connect your computer to the Tello drone's Wi-Fi network.
3. **Install Python Dependencies:**
   * Install the necessary Python libraries using the following commands:

bashCopy code

pip install pylsl djitellopy numpy

1. **Clone the Repository:**
   * Clone the EEG-Drone-Control repository from [repository URL].
2. **Run the Main Script:**
   * Navigate to the project directory and run the main script:

bashCopy code

python main.py

**Operation:**

1. **Launch the System:**
   * The system will start looking for an EEG stream from the Neuphony headset.
2. **Predict Drone Commands:**
   * The system continuously captures EEG data and predicts drone control commands using machine learning.
   * Predicted commands include takeoff ('T'), move forward ('R'), move backward ('L'), and land ('B').
3. **Drone Control:**
   * The drone will respond to your brain signals, executing the predicted commands.
4. **Adjust Parameters (Optional):**
   * Explore the code to customize parameters, such as EEG stream resolution and prediction thresholds.

**Caution:**

* Ensure a safe flying environment for the drone.
* Be prepared to take manual control if necessary.
* Always follow local regulations for drone usage.

**Troubleshooting:**

* If the system encounters issues, check the console for error messages.
* Ensure the Neuphony headset is properly connected and functioning.

This Python code is designed for Brain-Computer Interface (BCI) applications. It involves capturing EEG signals using a Neuphony headset, processing the data through a machine learning model, and then using the predictions to control a tello drone based on specific commands.

**main script:**

1. **Import Libraries:**
   * **pylsl**: for handling EEG data streams.
   * **numpy**: for numerical operations.
   * **djitellopy**: for controlling the Tello drone.
   * **ml\_for\_bci**: a custom module for running machine learning code for BCI.
2. **Initialization:**
   * Commented out lines related to drone connection and battery status.
3. **Action Dictionary:**
   * **actionDict** function maps predicted labels to drone control actions (takeoff, move forward, move backward, land).
4. **Validation Function:**
   * **validate\_action** function checks if the drone is flying or landed and takes appropriate actions based on predictions.
5. **EEG Stream Setup:**
   * Resolves and creates an inlet for capturing EEG data using LSL (Lab Streaming Layer).
6. **Data Processing Loop:**
   * Captures EEG data samples into a buffer.
   * When the buffer reaches a certain size, it triggers the machine learning model (**ml\_for\_bci.run\_cca**) for predictions.
   * The predicted label is converted to a drone action using the action dictionary.

**ml\_for\_bci script:**

1. **Import Libraries:**
   * Several libraries for signal processing, mathematical operations, and data visualization.
2. **Constants:**
   * Constants for CCA (Canonical Correlation Analysis), sampling frequency, and candidate frequencies.
3. **Reference Signal Generation:**
   * Functions for generating reference signals at different frequencies and harmonics.
4. **CCA Calculation:**
   * **find\_maximum\_canonical\_correlations** calculates the maximum canonical correlation between two multidimensional signals.
5. **Reference Data Generation:**
   * Generates reference data for each candidate frequency.
6. **CCA Classification:**
   * **classify\_cca** function compares the input EEG signal with reference signals using CCA and returns the result with the highest correlation.
7. **Run CCA Function:**
   * **run\_cca** function takes EEG data, performs CCA classification, and returns the predicted label.

**Summary:**

The **main** script continuously captures EEG data, processes it through the **ml\_for\_bci** module using CCA, and controls the drone based on the predictions. The machine learning model classifies the EEG signals into predefined commands (e.g., 'T' for takeoff, 'R' for move forward, etc.), enabling the user to control the drone using their brain signals.