

# 人本計算實驗

柯立偉 教授

陽明交通大學電機工程學系

# Real-time Brain-controlled Drone

國立陽明交通大學  
神經工程實驗室



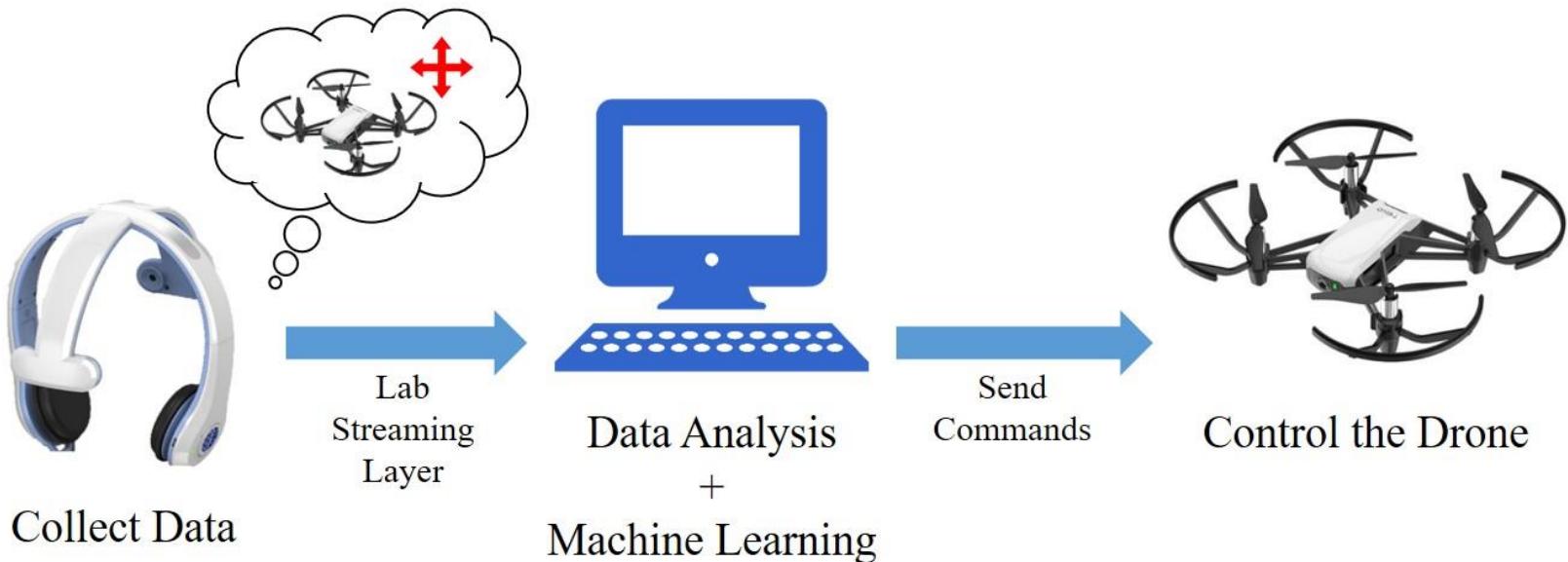
## Device



8-channel wearable EEG headset  
NYCU BCI



DJI-Ryze Tech  
Tello drone



包含4種模式：

1. 訓練階段
2. 測試階段

[Brain-Controlled Drone]  
1. Train stage  
2. Test stage

Select mode:

7項控制指令：

1. 上升 / 下降
2. 降落
3. 後退
4. 前進
5. 左轉 / 右轉

Commands	Movements
Up / Down	Blink fast / slow
Land	Close eyes
Backward	Grit teeth
Forward	Attention level rise
Left / Right	Arms movement

# 實驗器材

- Windows 系統電腦（需安裝好Cygnus）

- NYCU BCI 腦波帽套組

\* 腦波帽

\* dongle接收器

\* 海綿電極

\* Micro USB線

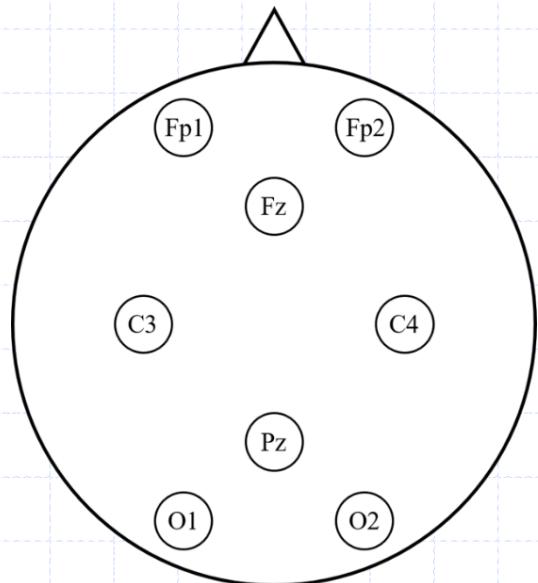
- Tello 無人機

- 腦控無人機程式：

- EEG\_data
- exp\_img
- function
- main.py
- requirements.txt
- tunable\_params.json

# 腦波訊號處理及分析

## 本課程腦波帽channel對應編號



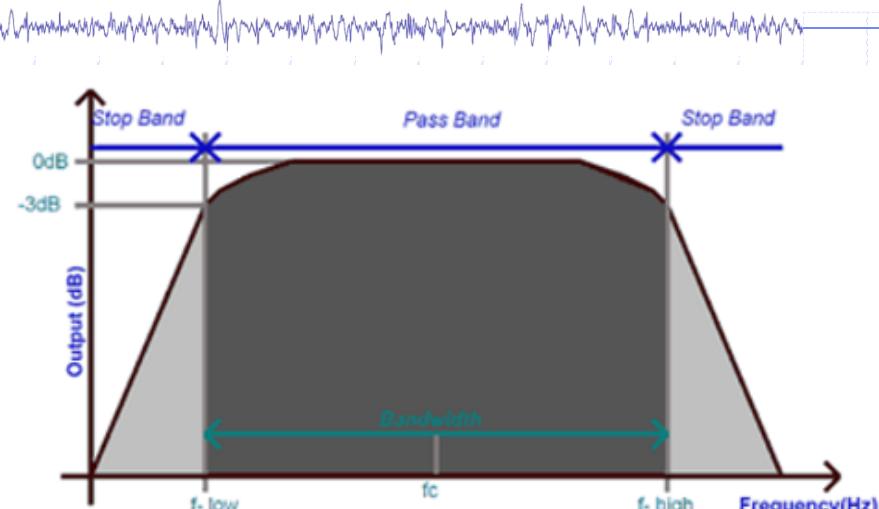
Channel編號	名稱	位置
0	Fp1	左前額葉
1	Fp2	右前額葉
2	Fz	額葉中央
3	C3	左腦運動皮質區
4	C4	右腦運動皮質區
5	Pz	頂葉中央
6	O1	左腦枕葉視覺區
7	O2	右腦枕葉視覺區

# 腦波訊號前處理

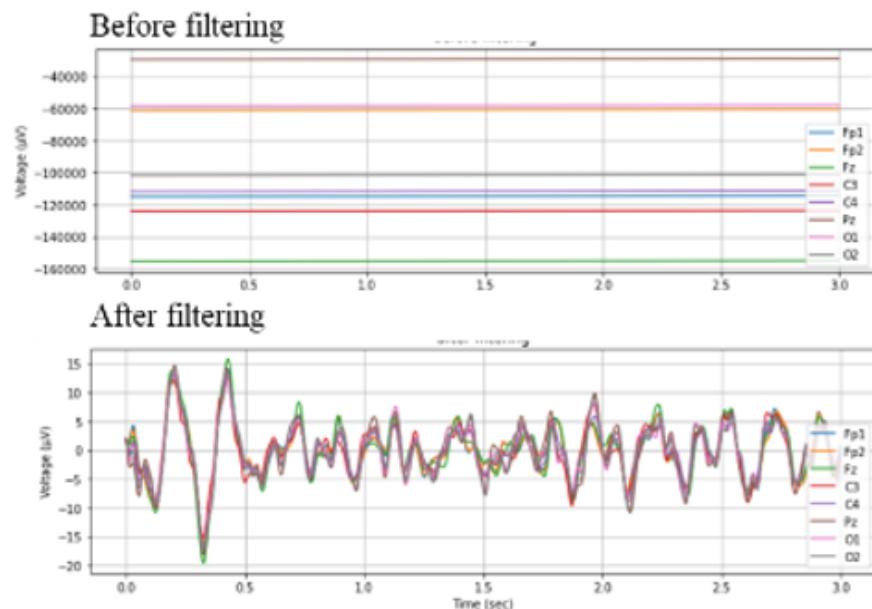
錄製腦波時可能會有其他來源的雜訊（EMG、電源的基線飄離等），會干擾頻譜與波型分析，可利用濾波器先行去除

腦波分析通常使用帶通濾波器（Bandpass filter）保留頻率一般設在肌電訊號（1.0 Hz）與交流電訊號（50 Hz或60 Hz）或是高頻雜訊（30 Hz以上）之間

本實驗保留頻段設為1 Hz ~ 30 Hz



$$f\text{-low} = 1 \text{ Hz}; f\text{-high} = 30 \text{ Hz}$$



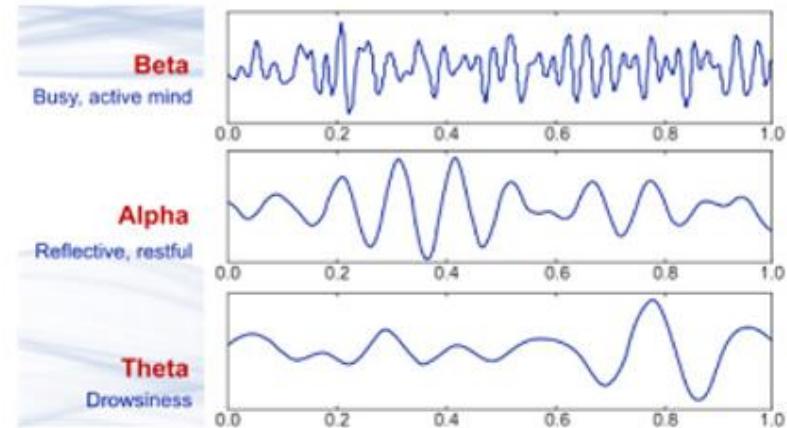
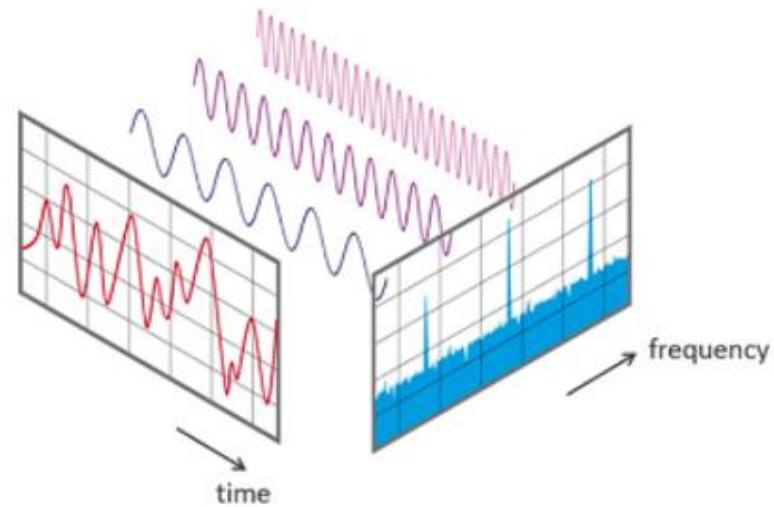
# 頻譜分析

## 傅立葉轉換 (Fourier transform) :

複雜的週期函數（如**腦波**）可以分解為一系列不同強弱的正弦函數 (sine) 以及餘弦函數 (cosine)

藉由傅立葉轉換，我們可以把複雜的腦波訊號分離到不同的頻段，使其可以分析

常用腦波	頻段
Delta	0.1 – 3 Hz
Theta	4 – 7 Hz
Alpha	8 – 15 Hz
Beta	15 – 30 Hz
Gamma	30 – 100 Hz



```
# sos IIR filter coeff
def SOS_IIR_coeff( fre_cutoff = [0.5, 50], # 截止頻率 (0.5~50 Hz)
                    fs = 1000, # 採樣頻率
                    pass_type = 'bandpass', # 帶通模式
                    ftype = 'butter', # 應用 Butterworth Filter
                    filter_order = 10): # IIR_filter 階數

    sos = iirfilter( filter_order, # IIR_filter 階數
                      Wn = fre_cutoff, # 截止頻率 (0.5~50 Hz)
                      btype = pass_type, # 帶通模式
                      analog = False, # False代表離散訊號模式
                      ftype = ftype, # 應用 Filter 類型 (Butterworth)
                      output = 'sos', # 使用 Second-Order Section Filtering
                      fs = fs)

    return sos

# SOS IIR Filter
def SOS_IIR_filter(filter_coeff, EEG):
    # input 二維 EEG 訊號 (ch,t)
    return sosfiltfilt(filter_coeff, EEG, axis=1)
```

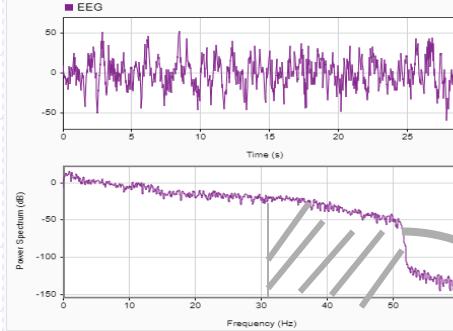
```
def get_bandpower_db(PS,X_epoch):
    if X_epoch.ndim == 2:
        X_epoch = X_epoch[np.newaxis,:,:] # 新增軸在第一軸
    f, Pxx = signal.welch( X_epoch ,
                           fs = PS.fs ,
                           window = 'hamming',
                           nperseg = PS.fs//2,
                           noverlap = PS.fs//4,
                           nfft = PS.nfft,
                           axis = 2
                         )
    bandpower = {}
    bands = {'theta': (4, 8), 'alpha': (8, 13), 'beta': (13, 30)}
    for band_name, (f_low, f_high) in bands.items():
        idx, = np.where((f >= f_low) & (f <= f_high))
        bp = np.trapz(Pxx[:, :, idx], x=f[idx], axis=2) # shape: (trial, ch)
        bandpower[band_name] = 20 * np.log10(np.clip(np.abs(bp), 1e-20, 1e100)) # 轉成 dB
    return bandpower
```

## SOS\_IIR\_filter(filter\_coeff, EEG)

- ❖ 利用濾波器過濾訊號，保留頻率一般設在肌電訊號 (1.0Hz) 與交流電訊號 (50Hz或60Hz) 或是高頻雜訊 (30Hz以上) 之間

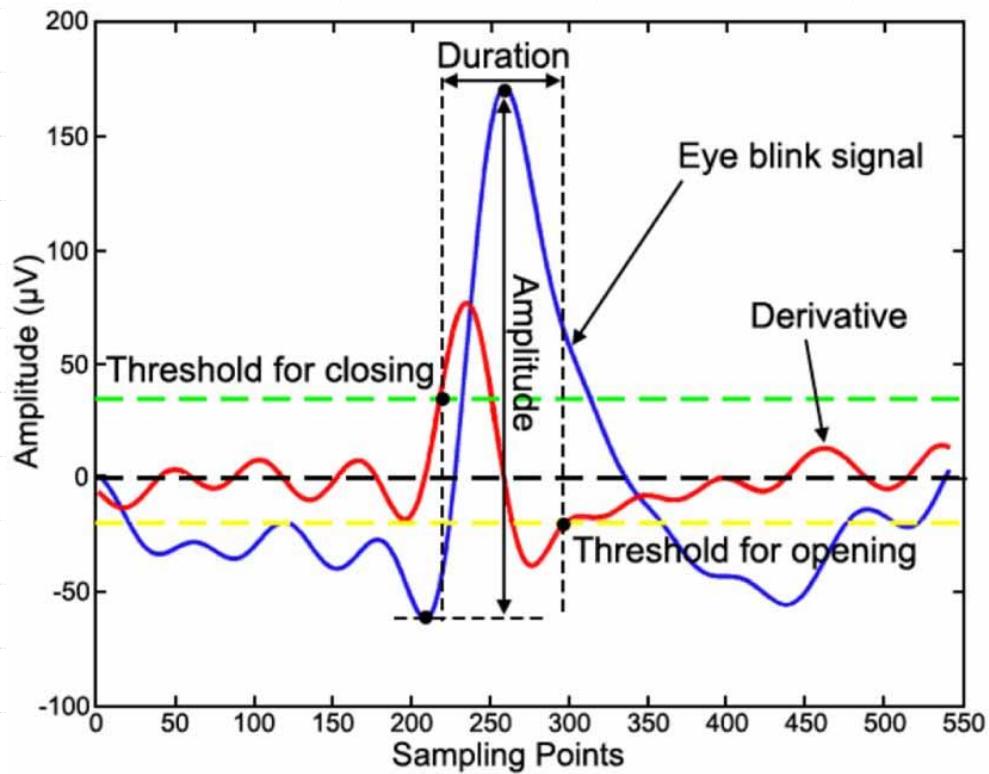
## get\_bandpower\_db(PS,X\_epoch)

- ❖ 從頻段下限至頻段上限的積分，即為該段頻段之power



該段程式在algorithm.py

眼瞼的肌肉在活動時，會在前額葉（frontal）的EEG電極（Fp1, Fp2）產生一個明顯的脈衝[1]



[1]

Ko, L. W., Komarov, O., Lai, W. K., Liang, W. G., & Jung, T. P. (2020). Eyeblink recognition improves fatigue prediction from single-channel forehead EEG in a realistic sustained attention task. *Journal of neural engineering*, 17(3), 036015.

```
# 眨眼 控制上下
def determine_updown(EEG, PS):
    up_threshold = PS.TP["up_threshold"]
    down_threshold = PS.TP["down_threshold"]
    peaks_Fp1, _ = signal.find_peaks(EEG[:, 0], prominence=150, width=50)
    peaks_Fp2, _ = signal.find_peaks(EEG[:, 1], prominence=150, width=50)
    thres_ud = np.min([len(peaks_Fp1), len(peaks_Fp2)])
    thres_ud_ct = C_text(f'{thres_ud}', 'lc')
    if thres_ud >= up_threshold:
        print(f'      updown : Eye Blink:{thres_ud_ct}, [{down_threshold}~{up_threshold}]:UP')
        return 1
    elif thres_ud >= down_threshold:
        print(f'      updown : Eye Blink:{thres_ud_ct}, [{down_threshold}~{up_threshold}]:Down')
        return -1
    print(f'      updown : Eye Blink:{thres_ud_ct}, [{down_threshold}~{up_threshold}]:None')
    return 0
```

find\_peaks(EEG, [檢測條件])

❖ 計算眨眼訊號在Fp1, Fp2 產生的脈衝數，來判斷眼動的次數

- 眨5次以上 → 無人機上升
- 眨3~4次 → 無人機下降

變數名稱	描述
prominence	脈衝變化幅度底限，在此幅度以下的脈衝不計算（單位：uV）
width	個別脈衝間的時間間隔底限，在此間隔以內的脈衝視為同一個（單位：1/1000秒）
peaks	包含所有脈衝時間點的list，若沒有找到脈衝則長度為0

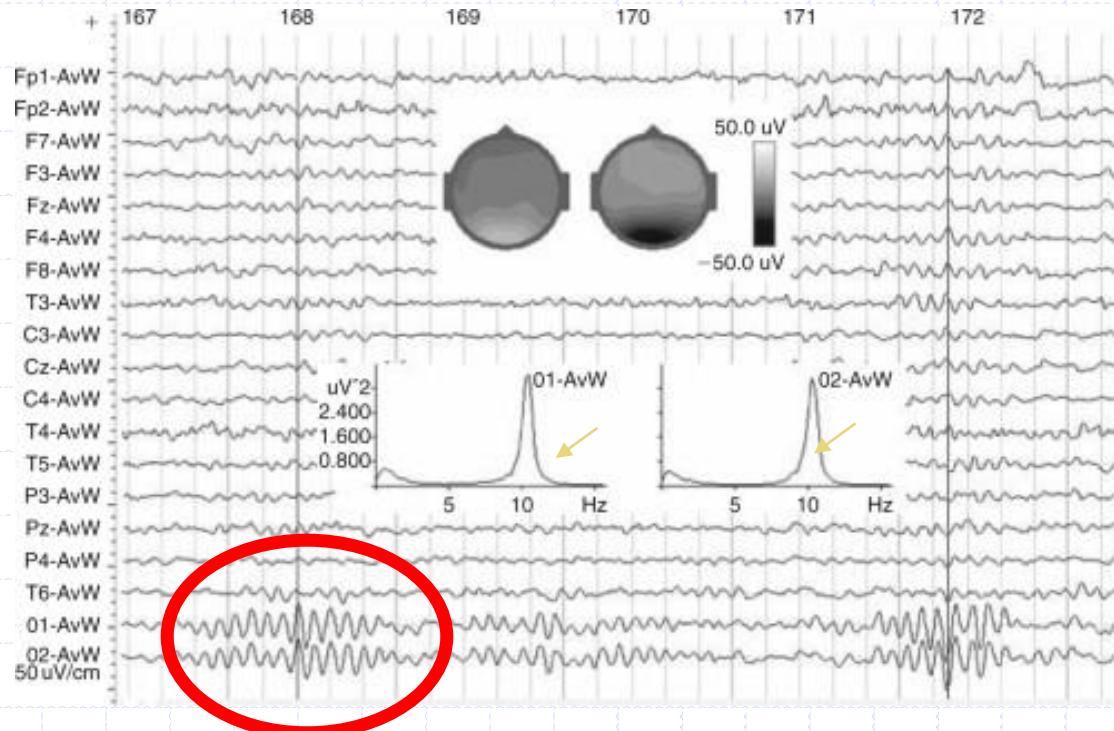
該段程式在test\_phase.py

# 閉眼訊號

國立陽明交通大學  
神經工程實驗室



在閉眼時，枕葉視覺區進入低刺激狀態，可以在O1以及O2觀察到與閒置有關的alpha波段腦波的活化[2]



[2]  
*Quantitative EEG, Event-Related Potentials and Neurotherapy.* Ch 2, p29-58

```
# 閉眼 控制降落
def determine_land(EEG, PS):
    # 計算這 3秒腦波的 bandpower -> (1,ch)_dict{theta,alpha,beta}
    curr_bp = get_bandpower_db(PS, EEG)
    # 去掉第 trial 軸, 只保留 ch 軸
    curr_bp_m = {label: data[0,:] for label, data in curr_bp.items()}
    # 計算 diff_bandpower
    diff_bandpower = calc_close_eyes_from_alpha(curr_bp_m, PS.close_eyes_baseline, show_msg=False)
    diff_bandpower_ct = C_text(f'{diff_bandpower:.2f}', 'lc')
    if diff_bandpower > PS.close_eyes_th*PS.TP["land_params"]:
        print(f'    land : close eyes level:{diff_bandpower_ct}, [{PS.close_eyes_th:.2f} * {PS.TP["land_params"]:.2f}]:True')
        return True
    print(f'    land : close eyes level:{diff_bandpower_ct}, [{PS.close_eyes_th:.2f} * {PS.TP["land_params"]:.2f}]:False')
    return False
```

- 比較O1以及O2在當下alpha波相較於baseline的上升倍數，超過land\_params則降落
- 閉眼 → O1, O2 Alpha上升 → 降落

變數名稱	描述
Land_params	Alpha power的上升倍數門檻，經 train_pase 計算得出

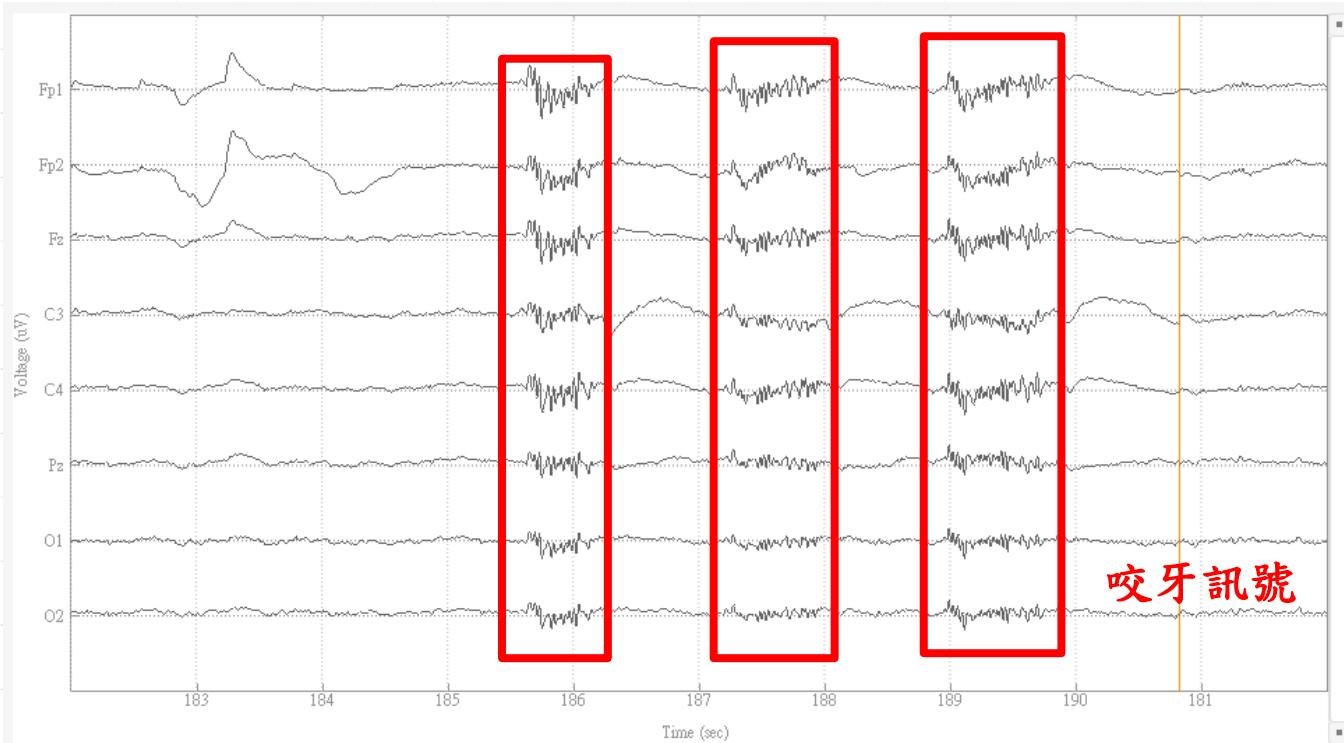
該段程式在test\_phase.py

# 咬牙訊號

國立陽明交通大學  
神經工程實驗室



咬牙會牽動面部肌肉產生肌電訊號（EMG），此時可以從EEG訊號觀察到EMG造成的震顫。因EMG的訊號較為高頻，透過比較beta power差異可判斷是否咬牙



```
# 咬牙 控制後退
def determine_backward(EEG, PS):
    # 計算這 3秒腦波的 bandpower -> (1,ch)_dict{theta,alpha,beta}
    curr_bp = get_bandpower_db(PS, EEG)
    # 取最高頻的 beta 其 FP1、FP2 通道 做分析
    Grit_teeth_beta = np.mean(curr_bp['beta'][0,:2])
    # base 用專注力的平均值，省的在抓一筆
    base_beta = np.mean(PS.attention_baseline['beta'][0,:2])
    #
    Grit_teeth_beta_ct = C_text(f'{Grit_teeth_beta:.2f}', 'lc')
    if Grit_teeth_beta > base_beta*PS.TP["backward_params"]:
        print(f'    backward : Grit teeth beta power: {Grit_teeth_beta_ct}, [{base_beta:.2f} * {PS.TP["backward_params"]:.2f}]:True')
        return True
    print(f'    backward : Grit teeth beta power: {Grit_teeth_beta_ct}, [{base_beta:.2f} * {PS.TP["backward_params"]:.2f}]:False')
    return False
```

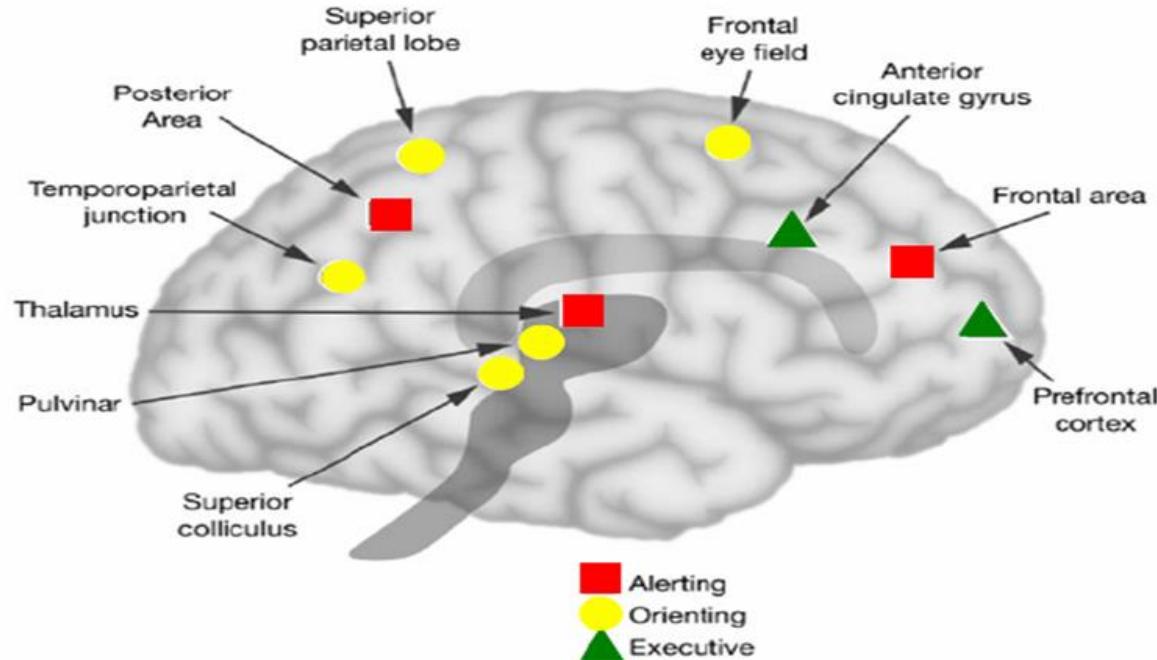
- 將Fp1, Fp2的beta power與baseline時相減，超過backward\_params則後退
- 咬牙 → beta 上升 → 後退

變數名稱	描述
backward_params	Beta power的門檻，經 train_pase 計算得出

該段程式在test\_phase.py

# Attention Networks

- ◆ Alerting ■: It involves the ability to maintain the **alert state**.  
frontal and parietal regions and brainstem. (Fan et al. J. Neurosci, 2007)
- ◆ Orienting ■: It involves the **selection of information** from inputs.  
right temporal-parietal junction(R-TPJ) and right inferior frontal gyrus (R-IFG). (K. Konrad et al. NeuroImage ,2005)
- ◆ Executive ■: It involves in **conflict processing**  
anterior cingulate cortex(ACC) and lateral prefrontal cortex(preFC).  
(Fan et al. J. Neurosci, 2007)



# 專注力

人在閉眼精神專注的時候，與閒置有關的alpha power，以及與疲倦有關的theta power均會隨著attention network活化而下降[3]，這些變化在與意識有關的額葉（Fz、F3、F4）最明顯[4]

[3]

Doesburg, S. M., Bedo, N., & Ward, L. M. (2016). Top-down alpha oscillatory network interactions during visuospatial attention orienting. *Neuroimage*, 132, 512-519.

[4]

Chang, Y., He, C., Tsai, B. Y., & Ko, L. W. (2021). Multi-Parameter Physiological State Monitoring in Target Detection Under Real-World Settings. *Frontiers in human neuroscience*, 793.

```
# 專注 控制前進
def determine_forward(EEG, PS):
    # 計算這 3秒腦波的 bandpower -> (1,ch)_dict{theta,alpha,beta}
    curr_bp = get_bandpower_db(PS, EEG)
    # 去掉第 trial 軸, 只保留 ch 軸
    curr_bp_m = {label: data[0,:] for label, data in curr_bp.items()}
    # 計算 diff_bandpower
    attentionlevel = calc_attention_from_theta_alpha(curr_bp_m, PS.attention_baseline, show_msg=False)
    attentionlevel_ct = C_text(f'{attentionlevel}', 'lc')
    if attentionlevel > PS.attention_th*PS.TP["forward_params"]:
        print(f'    forward : Attention level:{attentionlevel_ct}, [{PS.attention_th:.2f} * {PS.TP["forward_params"]:.2f}]:True')
        return True
    print(f'    forward : Attention level:{attentionlevel_ct}, [{PS.attention_th:.2f} * {PS.TP["forward_params"]:.2f}]:False')
    return False
```

- 將Fp1, Fp2, Fz的theta, alpha power 加總，並與baseline時的相減，即為當下的專注程度
- 專注 → Frontal Theta, Alpha大幅下降 → 前進

變數名稱	描述
forward_params	theta power的下降數值門檻，經train_pase 計算得出

該段程式在test\_phase.py

# 機器學習

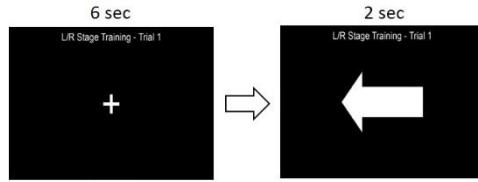
# 建置左右轉指令分類模型

國立陽明交通大學  
神經工程實驗室



## EEG data acquisition

L/R Stage (30 trials, 10 trials per label)

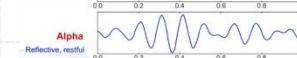
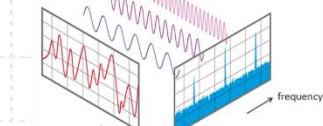
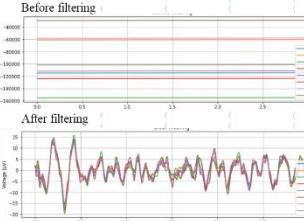
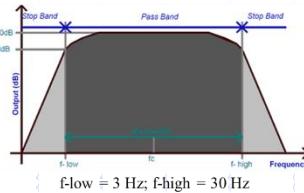


## Feature Extraction

Raw Data

Bandpass Filter

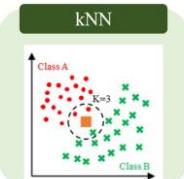
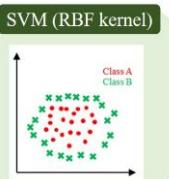
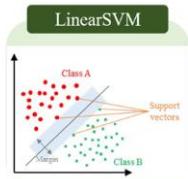
FFT



Theta: 4 - 7 Hz  
Alpha: 7.5 - 12.5 Hz  
Beta: 18 - 24 Hz

## Machine Learning

## Feature Selection



All Features



Feature Selection



Final Features

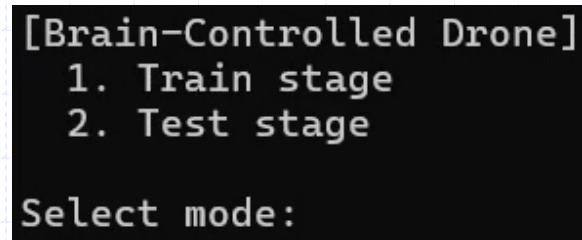


# BCI DRONE 系統介紹

# 操作說明

1. 執行“BCI\_drone\_2025(main).py”

看到此畫面→



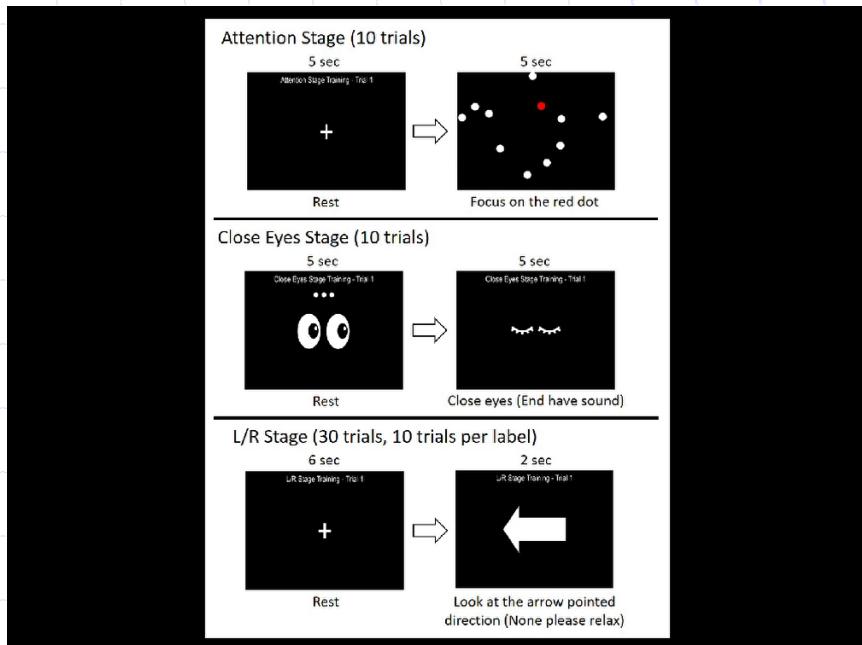
2. 第一次使用先選1（Train stage），根據指示完成腦波訊號錄製
3. 選擇2（Test stage）進行實時腦控無人機測試
4. 若模型效果不佳，可選擇1（Train stage），繼續收集新腦波訊號，優化左右轉判斷模型(同一人使用同樣的名字!)

# 腦波訓練收集 (Train phase)

國立陽明交通大學  
神經工程實驗室



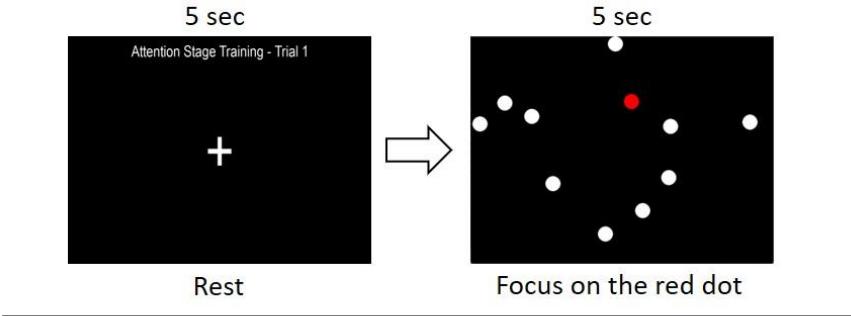
看到說明圖後按enter進行EEG錄製



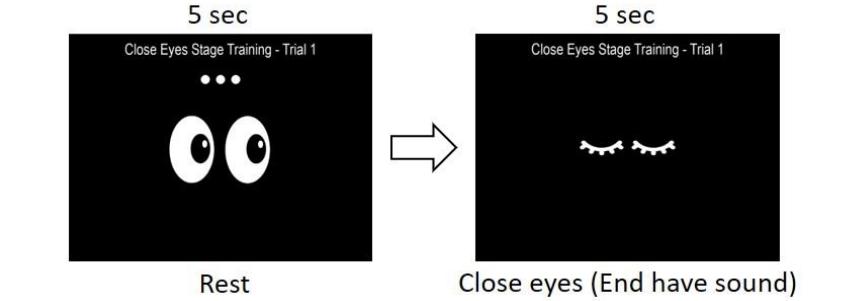
產生檔案：

- name\_data.npz
- name\_model1.joblib
- name\_model2.joblib

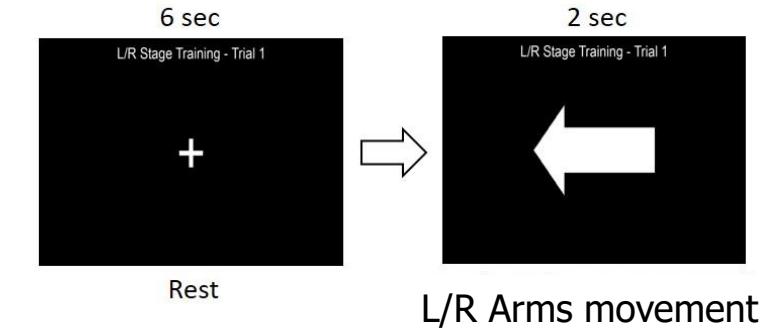
Attention Stage (10 trials)



Close Eyes Stage (10 trials)



L/R Stage (30 trials, 10 trials per label)



# 訓練模型 (Train model)

國立陽明交通大學  
神經工程實驗室



```
Please enter your name: t0 → model name
Please enter the number of training trials: 3
Attention Stage |██████████| 100.0% (3/3)
Close Eyes Stage |██████████| 100.0% (3/3)
L/R Stage |██████████| 100.0% (9/9)

===== model Training =====
已移除 0 筆資料
已移除 0 筆資料
已移除 0 筆資料
attention baseline      : -87.24
attention bandpower     : -87.38
attention level          : 5.0 → level
close eyes baseline     : -42.77
close eyes bandpower    : -42.40
close eyes level         : -0.37
判斷是否動作:
  CSP Accuracy: 60.30 % → accuracy
判斷左右:
  CSP Accuracy: 50.05 %
累計資料筆數: 48
```

# 模型優化 (Model optimization)

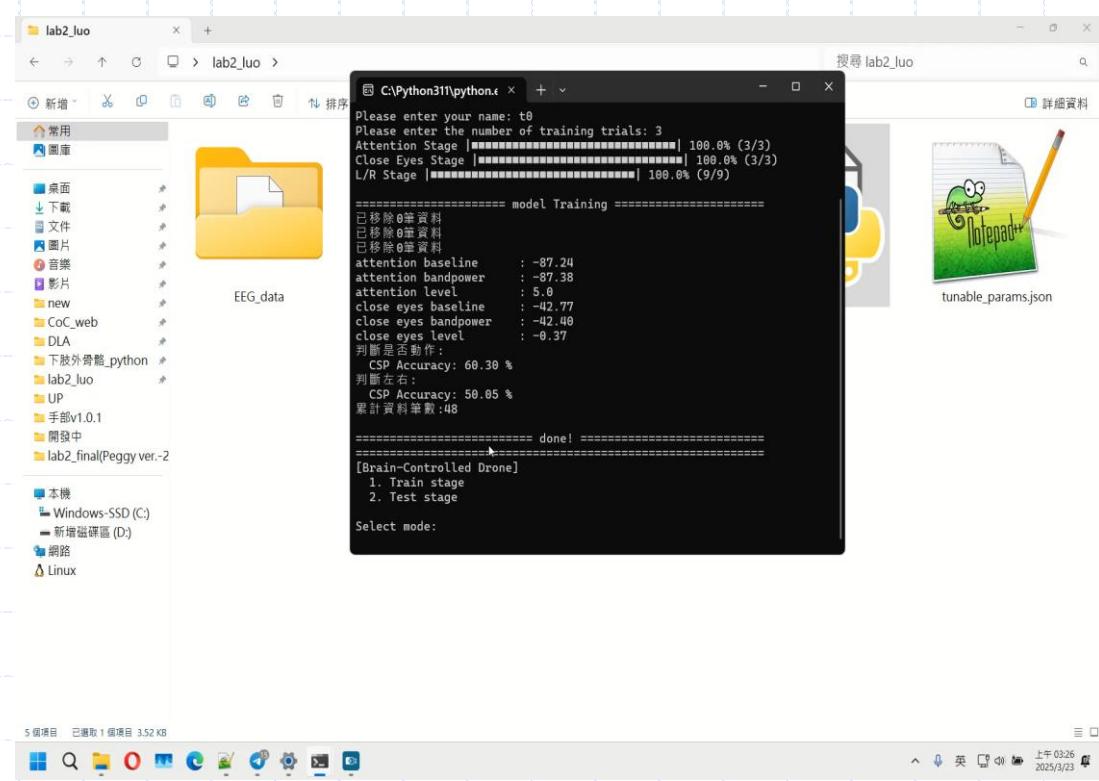
國立陽明交通大學  
神經工程實驗室



此模式為優化左右轉判斷模型，輸入想要測試的model  
名稱跟trial數（3的倍數，推薦15）

前5秒先放鬆以錄製baseline，後會聽到3種隨機提示音，  
根據提示音移動眼睛或放鬆

輸入新model名稱重新訓練



# 實時測試(Real-time test)

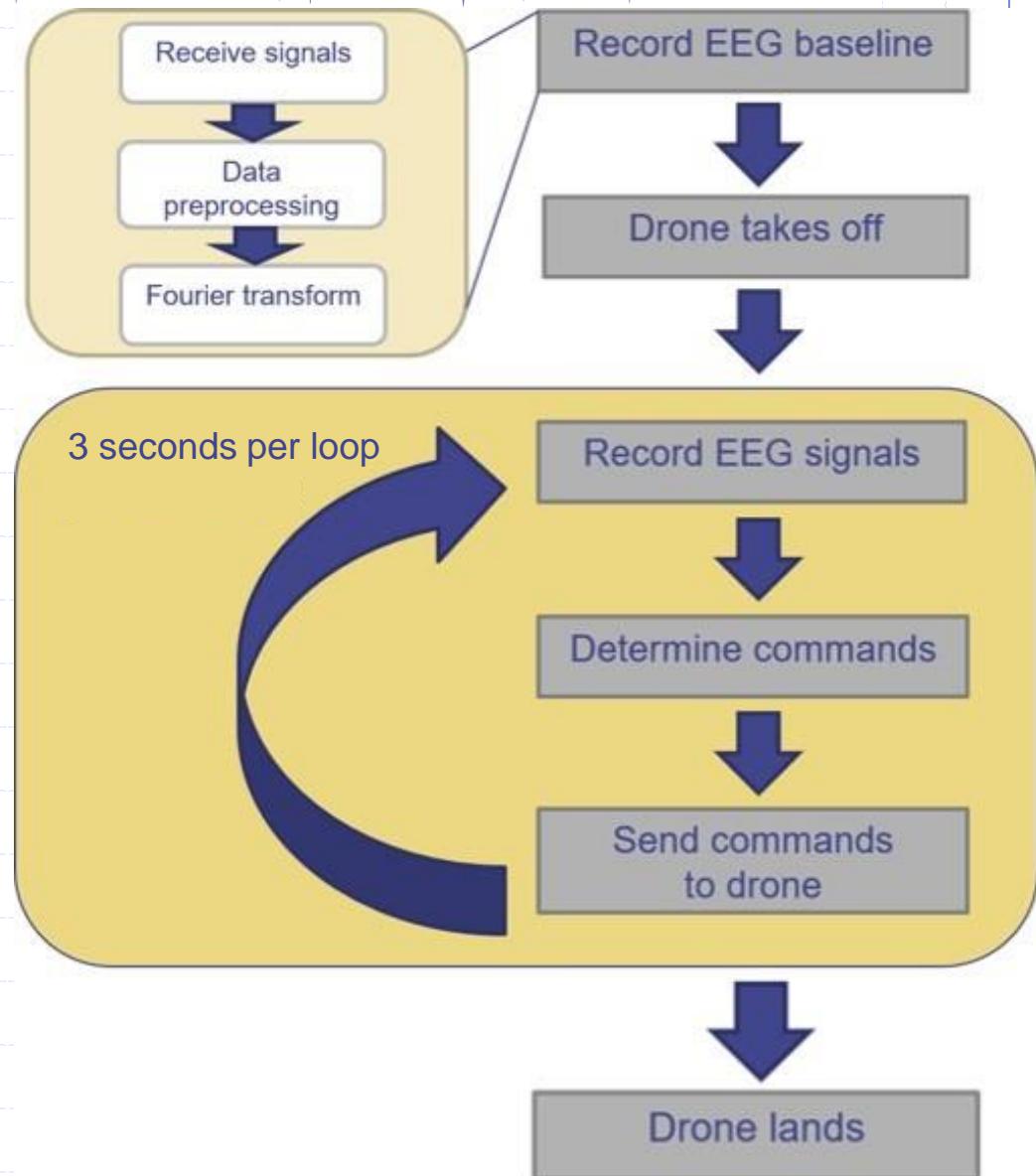
輸入要使用的model名稱及無人機移動次數

前10秒放鬆，作baseline

根據相應動作操控無人機

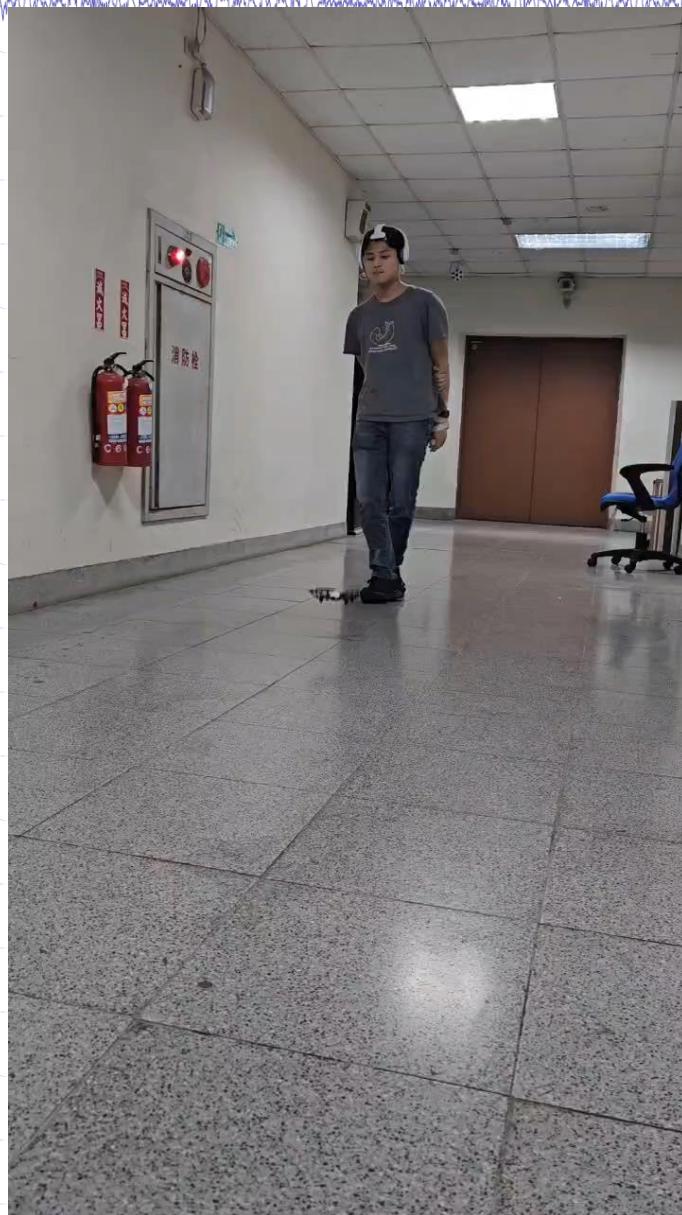
註：降落後無法再起飛，需重新執行此模式

Commands	Movements
Up / Down	Blink fast / slow
Land	Close eyes
Backward	Grit teeth
Forward	Attention level rise
Left / Right	Eyes movement



# 無人機控制實例

國立陽明交通大學  
神經工程實驗室



# 無人機控制實例

國立陽明交通大學  
神經工程實驗室



# 課堂作業

國立陽明交通大學  
神經工程實驗室



- 總分5分
- 成功執行BCI\_drone程式的Mode 1~4 (1分)
- 在Mode 4 (Real-time test) 讓無人機正確移動(3分)：
  - Up / Down 上升 / 下降
  - Land 降落
  - Backward 後退
  - Forward 前進
  - Left / Right 左轉 / 右轉
  - 每完成1個全組加1分，至多3分
- 每組超過一半人數完成Mode 2訓練(1分)
  - 附上執行的結果、截圖
  - (如24頁)

Commands	Movements
Up / Down	Blink fast / slow
Land	Close eyes
Backward	Grit teeth
Forward	Attention level rise
Left / Right	Arms movement

