BCI Control Missions

柯立偉 教授

國立陽明交通大學電機工程學系

BCI drone

國立陽明交通大學
神經工程實驗室

包含2種模式：

1. 訓練模型
2. 實時測試

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

Select mode:
```

7項控制指令：

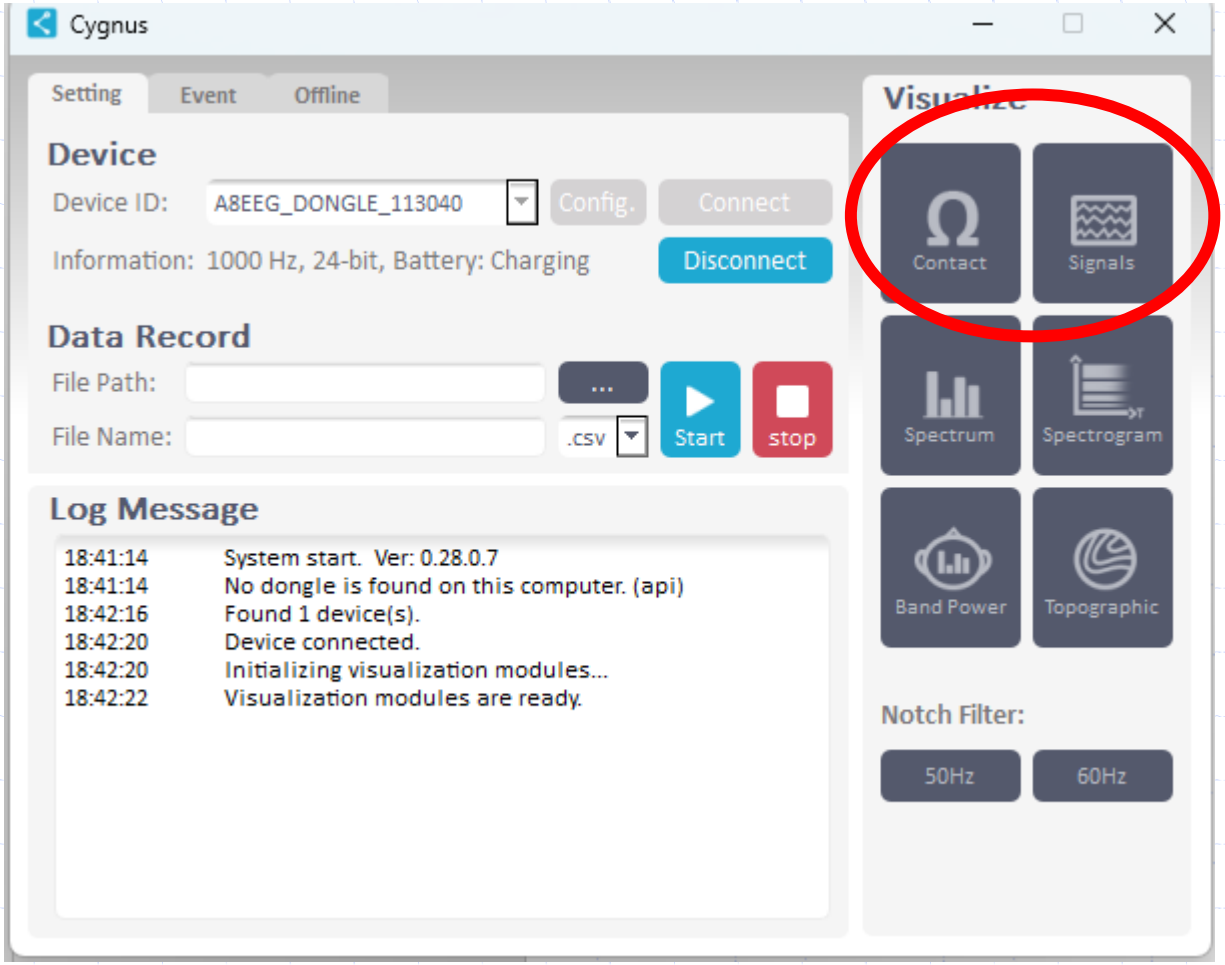
1. 上升 / 下降
2. **模式切換**
3. 左轉 / 右轉
4. 降落
5. 前進

Commands	Movements
Up / Down	Blink fast / slow
Change mode	Grit teeth
Left / Right	Blink fast / slow
Land	Close eyes
Forward	Attention level rise

無人機程式調整及說明

- 腦波判斷 Threshold
- 無人機控制指令

檢查 Cygnus



Contact 和 Signals
記得關掉 !!!!

檢查 Train model 的 Level

國立陽明交通大學
神經工程實驗室

```
====================== model Training ======================
已移除0筆資料
已移除0筆資料
attention baseline       : 60.17
attention bandpower      : 56.91
attention level          : -3.25
close eyes baseline      : 26.80
close eyes bandpower     : 33.86
close eyes level         : 7.06
累計資料筆數:5
```

正常值約為負值

正常值約為 5 ~ 10

上升/下降 判斷

國立陽明交通大學
神經工程實驗室

```python
# 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
```

該段程式在test_phase.py

❖ 計算眨眼訊號在Fp1, Fp2 產生的脈衝數，來判斷眼動的次數

執行結果

```
====================== Trial No.24 ======================

  state :
    updown : Eye Blink:3,[3~5]:Down
    land : close eyes level:9.47,[13.73*1.00]:False
    switch mode : Grit teeth beta power: 26.59,[18.41*2.00]:False
    forward : Attention level:19.29,[-3.70*1.00]:False
  output : down
down 30
```

- 上升下降模式下:
- 眨5次以上 → 無人機上升
- 眨3~4次 → 無人機下降

模式切換判斷

國立陽明交通大學
神經工程實驗室

該段程式在Test.py

```python
# 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'][: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和平均腦波電位與baseline時相減，超過threshold則切換模式
- 咬牙 → beta上升 → 切換模式

執行結果

```
====================== Trial No.5 ======================

state :
  updown : Eye Blink:1,[3~5]:None
  land : close eyes level:20.13,[13.73*1.00]:True
  switch mode : Grit teeth beta power: 53.22,[18.41*2.00]:True
  forward : Attention level:35.14,[-3.70*1.00]:False
output : switch mode
```

左右轉執行結果

```
======================== Trial No.6 ========================

  state :
    updown : Eye Blink:8 [3~5]:Left
    land : close eyes level:10.09,[13.73*1.00]:False
    switch mode : Grit teeth beta power: 28.72,[18.41*2.00]:False
    forward : Attention level:58.24,[-3.70*1.00]:False
  output : left
ccw 45
```

- 計算眨眼訊號在Fp1, Fp2 產生的脈衝數，來判斷眼動的次數

- 左右轉模式下:
- 眨5次以上 → 無人機左轉
- 眨3~4次 → 無人機右轉

降落判斷

國立陽明交通大學
神經工程實驗室

```python
# 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
```

該段程式在test_phase.py

- 比較O1以及O2在當下alpha波相較於baseline的上升倍數，超過模型計算出的threshold則降落
- 閉眼 → O1, O2 Alpha上升 → 降落

```
====================== Trial No.6 ======================
  state :
    updown : Eye Blink:0,[3~5]:None
    land : close eyes level:21.59,[13.73*1.00]:True
    switch mode : Grit teeth beta power: 25.83,[18.41*2.00]:False
    forward : Attention level:8.41,[-3.70*1.00]:False
  output : land
land
```

執行結果

前進判斷


```python
# 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
```

該段程式在test_phase.py

- 將Fp1, Fp2, Fz的theta power加總，並與baseline時的總合相減，低於模型計算出的threshold則前進

- 專注 → Frontal Theta大幅下降 → 前進

執行結果

```
====================== Trial No.2 ======================

  state :
    updown : Eye Blink:0,[3~5]:None
    land : close eyes level:-1.20,[13.73*1.00]:False
    switch mode : Grit teeth beta power: 22.89,[18.41*2.00]:False
    forward : Attention level:-15.88,[-3.70*1.00]:True
  output : forward
forward 100
```

無人機控制指令參數調整

國立陽明交通大學
神經工程實驗室


```
'up_threshold": 5,
'down_threshold": 3,
"sw_params": 2,
'land_params": 0.75,
'forward_params": 1,
"command": {
  "up": "up 30",
  "down": "down 30",
  "left": "ccw 45",
  "right": "cw 45",
  "forward": "forward 60",
  "land": "land",
  "no action": "cw 0"
```

上下左右眨眼次數

切換模式(咬牙)參數倍率

降落前進參數倍率

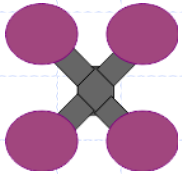
無人機上升/下降高度

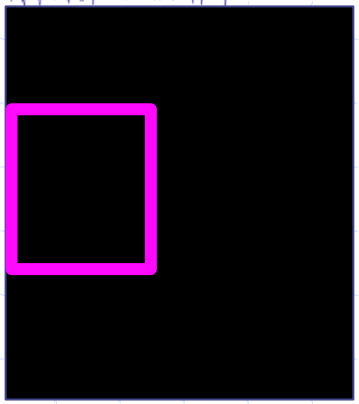
無人機左轉/右轉角度

無人機前進距離

Mission 1 直線飛行

國立陽明交通大學
神經工程實驗室





讓無人機直線飛行，並在黑色墊子上降落(不限挑戰次數)

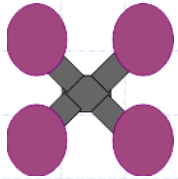
- 必須在3個指令以上完成
- 完成可得分數2分
- 降落指令必須由使用者發出，否則0分

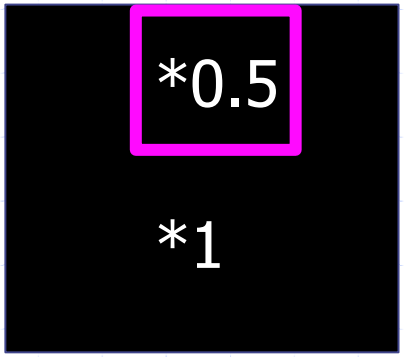
Bonus: 競速比賽（一組限挑戰3次）

- 飛行時間最短的三組有獎
- 降落在粉色框時間乘0.5倍，黑色墊子外挑戰失敗

Mission 1 直線飛行

國立陽明交通大學
神經工程實驗室

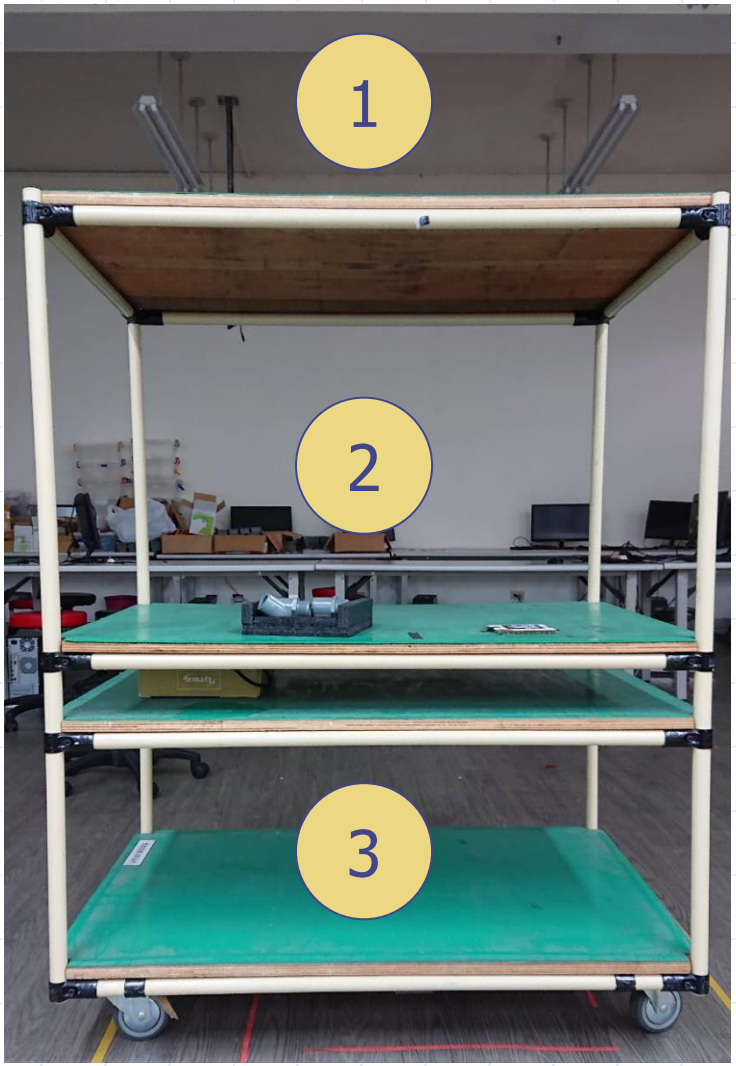




失敗

Mission 2 障礙飛行

1. 直線飛行穿過1, 2, 3任選一個通道通過
2. 轉身180度後，由其餘通道返回
3. 與 Mission 1不得為同一名駕駛
4. 穿過通道得一分，轉身得一分，成功穿回通道得一分
5. 落地即換下組挑戰



Mission 2 障礙飛行

國立陽明交通大學
神經工程實驗室