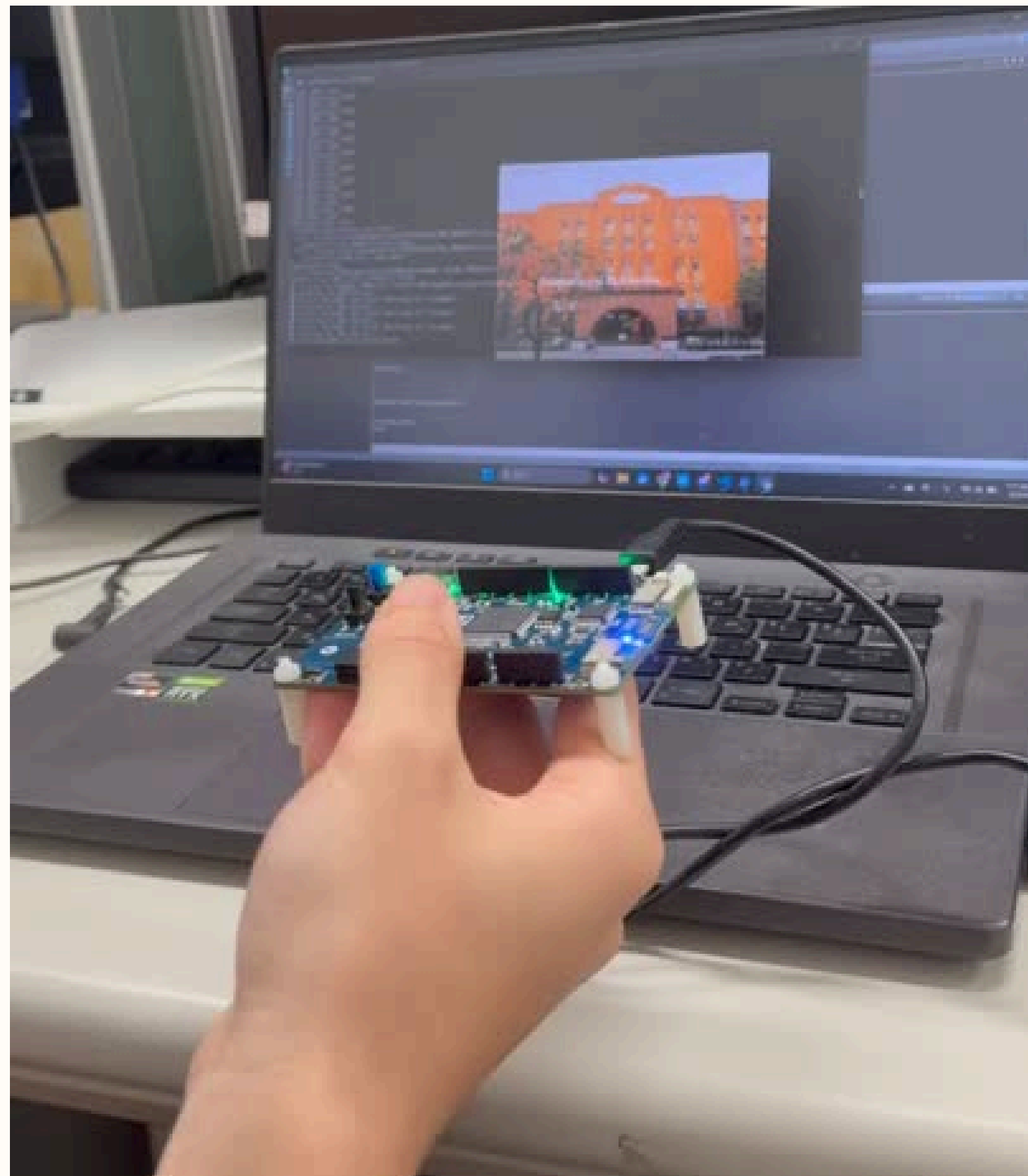


NTU run

b10901162 張逸安

b11901095 朱振瑤



問題描述：

在遊戲中使用加速度感測器控制角色上下左右移動。

當玩家上左右移動時，裝置誤判為上下移動，導致操作錯誤。

需要提升方向判定的準確性，避免誤觸。

比較偵測到加速度方向

分析結果：

左右移動有瞬間的高頻z軸訊號,震動干擾造成z軸訊號影響。
加速度訊號變化量小但方向敏感，導致微小晃動被誤判為指令。

僅看單軸最大值方向不足以應對實際操作變化。

解法一 —— DSP Running Average Filter

```
float32_t firCoeffs[NUM_TAPS] = {  
    0.2f, 0.2f, 0.2f, 0.2f, 0.2f  
};
```

✅ 最終效果：

濾波器成功過濾掉短期雜訊與非主要方向的擾動
控制邏輯穩定，操作靈敏度更符合玩家預期
誤判率明顯降低，提升整體使用體驗

```
BSP_ACCELERO_AccGetXYZ(pDataXYZ);  
  
float32_t xout, yout, zout;  
  
float32_t xin = (float32_t)pDataXYZ[0];  
float32_t yin = (float32_t)pDataXYZ[1];  
float32_t zin = (float32_t)pDataXYZ[2];  
  
arm_fir_f32(&Sx, &xin, &xout, BLOCK_SIZE);  
arm_fir_f32(&Sy, &yin, &yout, BLOCK_SIZE);  
arm_fir_f32(&Sz, &zin, &zout, BLOCK_SIZE);  
  
int16_t x = (int16_t)xout;  
int16_t y = (int16_t)yout;  
int16_t z = (int16_t)zout;  
z -= g;
```

motion logic

```
if (cooldown){
    if (HAL_GetTick() - cd_start_time > 350){
        cooldown = false;
    }else{
        continue;
    }
}
switch (x_state){
case 0: //no move
    x_max = 0;
    x_min = 0;
    if (x > x_small_threshold){
        x_max = x;
        x_state = 1;
    }
    if (x < -x_small_threshold){
        x_min = x;
        x_state = -1;
    }
    break;
case 1:
    if (x > x_max){
        x_max = x;
    }
    if (x < x_small_threshold){
        x_state = 0;
        if (x_max > x_threshold){
            printf("right: %d\n",x_max);
            char msg[20];
            sprintf(msg, "r%d",x_max);
            Send_Action(msg,strlen(msg));
            cooldown = true;
            cd_start_time = HAL_GetTick();
        }
        x_min = x;
    }
    break;
```

結合：

變化量大小判斷 ($x_{\max} > \text{threshold}$)

狀態轉移過濾 (x_{state})

時間冷卻保護 (cooldown)

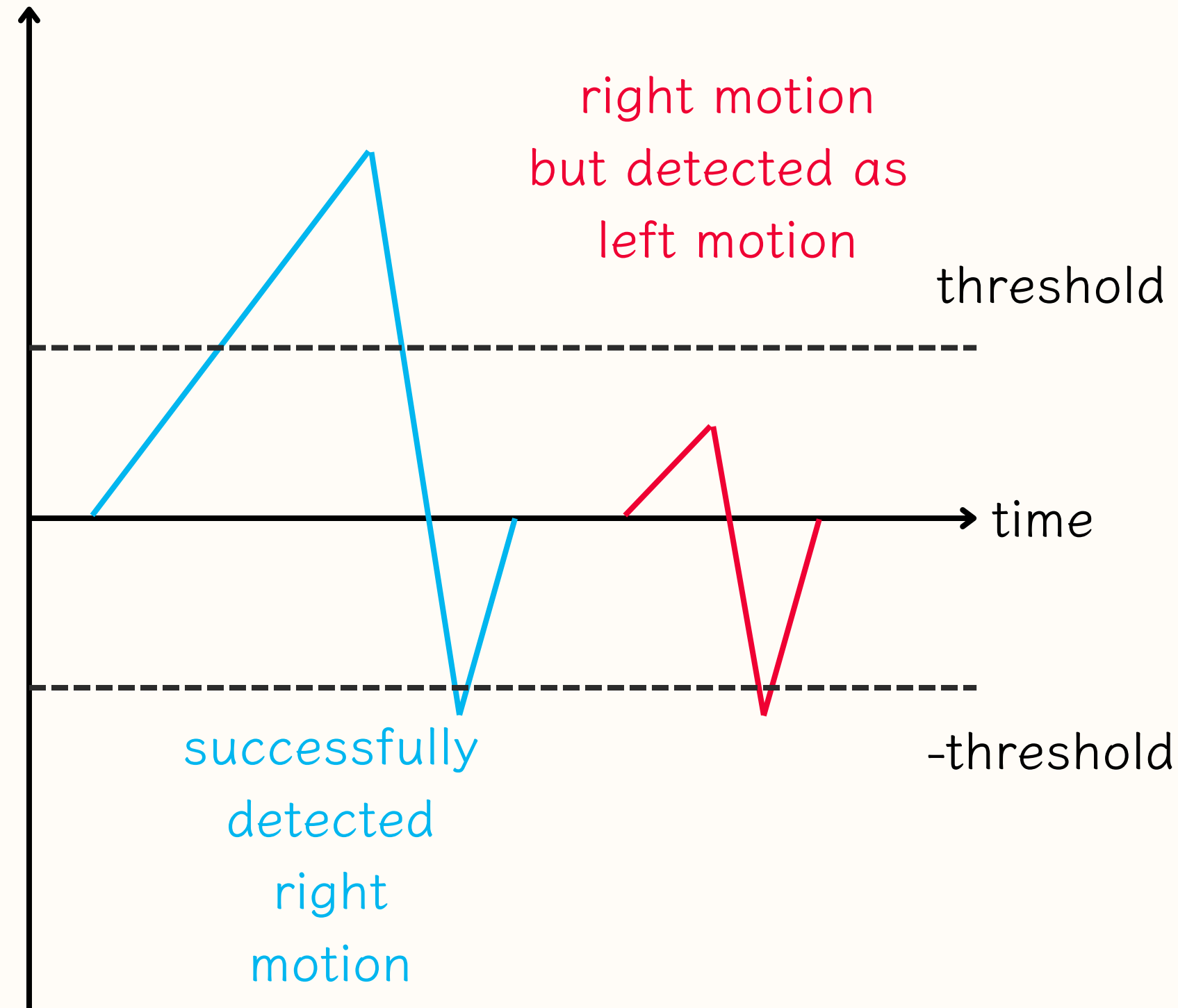
有效降低加速度震動導致的誤判與連續觸發

可擴展至 y 軸與 z 軸，只需加入對應狀態與判斷邏輯
wireless communication

問題描述：
有時玩家左右移動時
偵測的移動方向與實際相反

分析結果：
移動方向的加速度計讀值未超過threshold
而減速時的值有超過threshold

sampled x-axis acceleration data



嘗試過的解法：

把波形拆state(以):

- 0.靜止:未達small threshold
 - 1.開始移動:達small threshold
 - 2.準備減速:數值降到small threshold以內
 - 3.減速中:達 $-\text{small threshold}$
 - $> -\text{small threshold}$ 則回到state 0
- 需導入running average
使得加速度增加或減少的過程

為何失敗：

未使用BLE時沒問題

但BLE通訊時會導致latency

使一段時間無法讀加速度計的值
state的切換就不順利

實際解法：

減速的過程通常很急

因此直接用running average
把減速的signal濾掉即可

sampled x-axis acceleration data

