通訊專題-20250610

蘇天樂
黃英杰
徐子珩

1.目標：

* 自動鎖定並追蹤一個相位會隨時間漂移的展頻信號。

2.核心策略 (狀態機)：

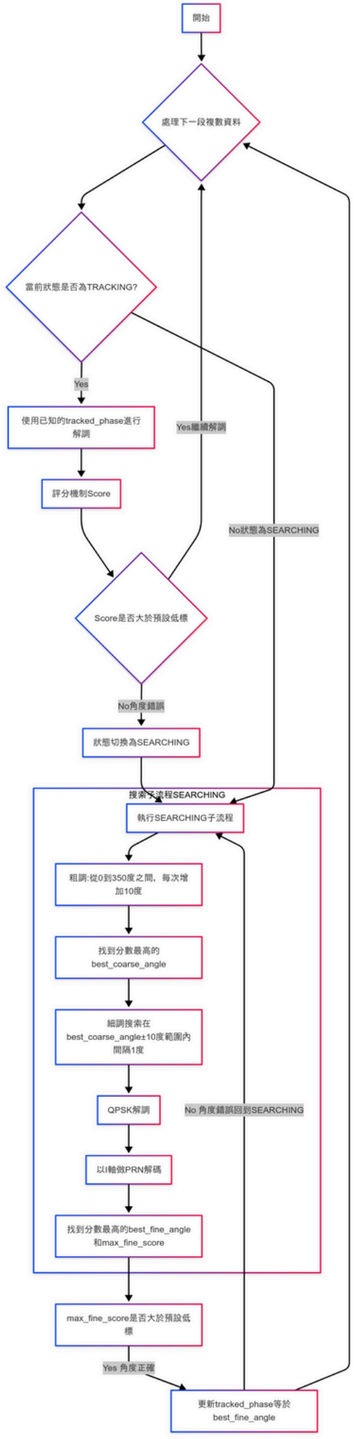
* SEARCHING 狀態 (搜索模式)： 當信號相位未知時，啟動一個「粗調+細調」的二階段搜索，來精準定位當前的相位角度。
* TRACKING 狀態 (追蹤模式)： 成功找到角度後，切換至此模式。後續資料直接用已知的角度快速處理，運算成本極低。

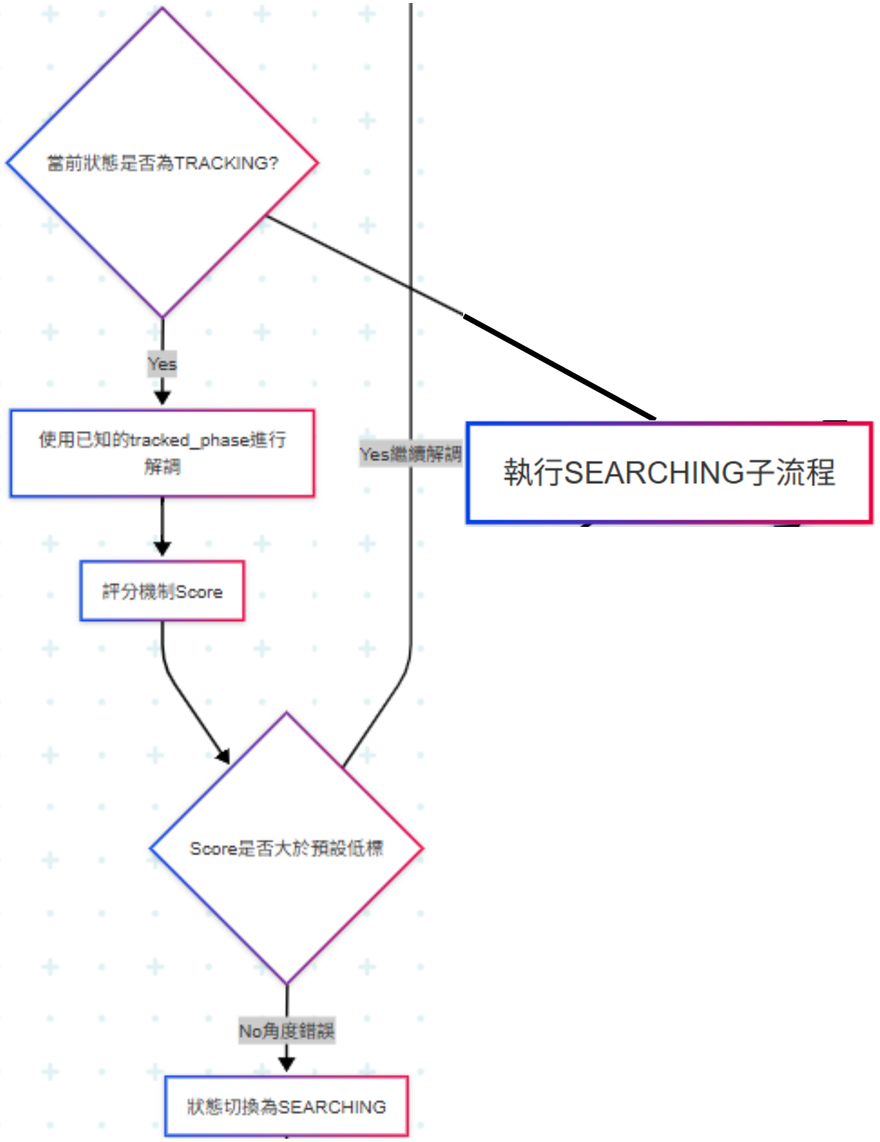
3.適應機制 (失鎖偵測)：

* 在TRACKING模式下，程式會持續監控信號品質（相關分數）。
* 若分數突然下降（代表真實相位已改變），程式會立刻判定「失鎖」，並自動切換回SEARCHING狀態去尋找新的相位。

總結流程：

整個程式就是一個在「搜索」和「追蹤」兩種模式間智慧切換的循環，目的是以最低的運算成本，持續與一個動態變化的信號保持同步。

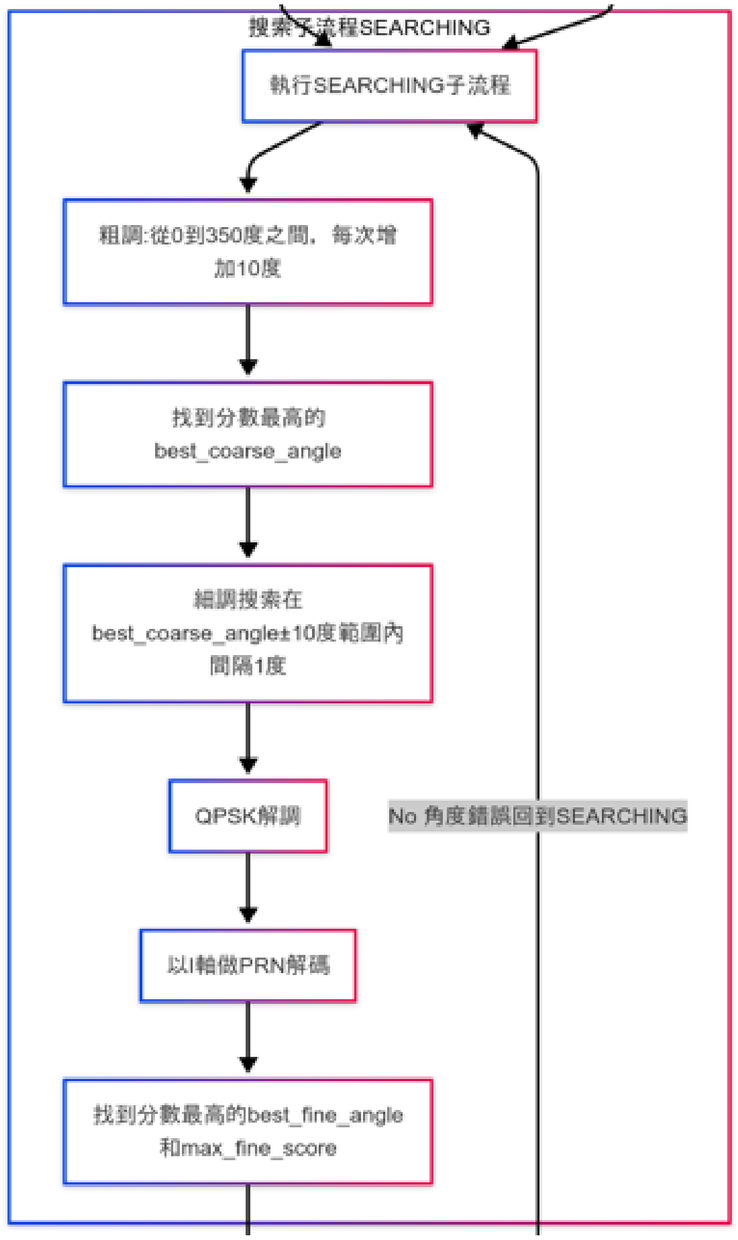

開始
處理下一段複數資料
當前狀態是否為TRACKING?
Yes
使用已知的tracked_phase進行解調
Yes繼續解調
評分機制Score
No狀態為SEARCHING
Score是否大於預設低標
No角度錯誤
狀態切換為SEARCHING
搜索子流程SEARCHING
執行SEARCHING子流程
粗調:從0到350度之間，每次增加10度
找到分數最高的best_coarse_angle
細調搜索在best_coarse_angle±10度範圍內間隔1度
QPSK解調
No 角度錯誤回到SEARCHING
以I軸做PRN解碼
找到分數最高的best_fine_angle和max_fine_score
max_fine_score是否大於預設低標
Yes 角度正確
更新tracked_phase等於best_fine_angle




```
        if (currentState == State::TRACKING) {
            // --- 追蹤模式 ---
            double tracked_angle_rad = (double)tracked_phase_deg * PI / 180.0;
            double current_score = calculate_correlation(current_block, pn_code,
tracked_angle_rad);

            if (current_score <= S_LOCK_LOSS_THRESHOLD) {
                // 事件：鎖定遺失 (峰值偏移)
                if (tracking_start_block != -1) {
                    cout << ">> Tracking state maintained for blocks: " <<
tracking_start_block << " to " << i - 1 << endl;
                }
                cout << ">> Phase Lost! Detected at sample " << i * N_BLOCK << " (Block
" << i << "). Re-searching..." << endl;
                cout << "--------------------------------------------------" << endl;

                currentState = State::SEARCHING;
                tracking_start_block = -1;
            }
        }
```




```
if (currentState == State::SEARCHING) {
    // --- 搜索模式 ---
    // 1. 粗調: 找峰值大概在哪
    double max_coarse_score = -1.0;
    int best_coarse_angle_deg = -1;
    for (int angle_deg = 0; angle_deg < 360; angle_deg += 10) {
        double score = calculate_correlation(current_block, pn_code, (double)angle_deg * PI / 180.0);
        if (score > max_coarse_score) {
            max_coarse_score = score;
            best_coarse_angle_deg = angle_deg;
        }
    }

    // 2. 細調: 精準定位峰值中心
    double max_fine_score = -1.0;
    int best_fine_angle_deg = -1;
    for (int angle_deg = best_coarse_angle_deg - 10; angle_deg <= best_coarse_angle_deg + 10; ++angle_deg) {
        int normalized_angle = (angle_deg + 360) % 360;
        double score = calculate_correlation(current_block, pn_code, (double)normalized_angle * PI / 180.0);
        if (score > max_fine_score) {
            max_fine_score = score;
            best_fine_angle_deg = normalized_angle;
        }
    }
```


```
            // 3. 判斷是否成功找到峰值
            if (max_fine_score > S_ACQUIRE_THRESHOLD) {
                // 事件：成功找到新峰值
                cout << ">> Phase Lock Successful!" << endl;
                cout << "   - Found at block: " << i << endl;
                cout << "   - Estimated phase angle (Peak): " << best_fine_angle_deg << " degrees" << endl;

                // 計算並顯示校正後的複數資料
                vector<complex<double>> corrected_block(N_BLOCK);
                double best_angle_rad = (double)best_fine_angle_deg * PI / 180.0;
                complex<double> final_rotation = exp(complex<double>(0, -best_angle_rad));
                for(int j=0; j<N_BLOCK; ++j){
                    corrected_block[j] = current_block[j] * final_rotation;
                }
                print_complex_vector("   - Data after phase correction (first 5 samples): ", corrected_block, 5);
                cout << "-------------------------------------------------------" << endl;

                // 更新狀態，準備開始追蹤
                currentState = State::TRACKING;
                tracked_phase_deg = best_fine_angle_deg;
                tracking_start_block = i + 1;
            }
        }
    }
```

```
================== SIMULATION START ==================
Original complex data (first 5 samples): (0.87, -0.49) (-0.87, 0.49) (0.87, -0.49) (0.87, -0.49) (0.87, -0.49) ...
------------------------------------------------------
>> Phase Lock Successful!
   - Found at block: 0
   - Estimated phase angle (Peak): 150 degrees
   - Data after phase correction (first 5 samples): (-1.00, -0.01) (1.00, 0.01) (-1.00, -0.01) (-1.00, -0.01) (-1.00, -0.01) ...
------------------------------------------------------
>> Tracking state maintained for blocks: 1 to 8
>> Phase Lost! Detected at sample 2304 (Block 9). Re-searching...
------------------------------------------------------
>> Phase Lock Successful!
   - Found at block: 9
   - Estimated phase angle (Peak): 74 degrees
   - Data after phase correction (first 5 samples): (-1.00, -0.01) (1.00, 0.01) (-1.00, -0.01) (-1.00, -0.01) (-1.00, -0.01) ...
------------------------------------------------------
>> Tracking state maintained for blocks: 10 to 17
>> Phase Lost! Detected at sample 4608 (Block 18). Re-searching...
------------------------------------------------------
>> Phase Lock Successful!
   - Found at block: 18
   - Estimated phase angle (Peak): 125 degrees
   - Data after phase correction (first 5 samples): (-1.00, 0.00) (1.00, -0.00) (-1.00, 0.00) (-1.00, 0.00) (-1.00, 0.00) ...
------------------------------------------------------
>> Tracking state maintained for blocks: 19 to 26
>> Phase Lost! Detected at sample 6912 (Block 27). Re-searching...
------------------------------------------------------
>> Phase Lock Successful!
   - Found at block: 27
   - Estimated phase angle (Peak): 47 degrees
   - Data after phase correction (first 5 samples): (1.00, 0.00) (-1.00, -0.00) (1.00, 0.00) (1.00, 0.00) (1.00, 0.00) ...
------------------------------------------------------
>> Tracking state maintained for blocks: 28 to 38
================== SIMULATION END ==================
PS C:\Users\user\Desktop\新增資料夾 (4)>
```