

1. navigation.pynavigation

➤ TODO 邏輯 (控制與碰撞處理):

- 根據 simulator 類型提取狀態資訊 info。
- 呼叫控制器 feedback() 得到控制命令 (速度/轉向)。
- 新增倒退處理邏輯與觸發 set_controller_path = True。

➤ 設計理由:

- 增加 info 的完整欄位 (如 delta, l, dt) 以支援進階控制器 (如 LQR)。
- 藉由倒退重規劃，避免機器人碰撞後卡死。

2. planner_a_star.pyplanner_a_star

➤ TODO 邏輯:

- 實作 A* 演算法主迴圈，透過 $f = g + h$ 選擇最佳節點擴展。
- 使用固定方向鄰點 (上/下/左/右) 搜尋。
- 記錄 parent 以便重建路徑。

➤ 設計理由:

- 使用 inter 間距進行等步長搜尋，平衡搜尋密度與速度。
- 節點只在 g 更小時更新，保證路徑代價最小。

3. planner_rrt_star.pyplanner_rrt_star

➤ TODO 邏輯:

- Re-parent: 找出半徑範圍內成本最低的父節點。
- Re-wire: 嘗試讓已存在的鄰近節點改用 new_node 為父節點，若成本更低。
- 加入 goal_bias 與 path smoothing 提高品質與效率。

➤ 設計理由:

- Re-parent + Re-wire 是 RRT* 核心，讓路徑在擴展過程中優化。
- 路徑平滑處理減少 Zigzag，使後續控制更穩定。

4. controller_lqr_basic.pycontroller_lqr_basic

➤ TODO 邏輯:

- 選取 look-ahead target。
- 建立 A, B 狀態空間模型，設計 LQR 控制律。
- 使用速度調整 (dynamic_v) 根據曲率減速。
- 若接近終點，直接停車。

➤ 設計理由:

- 使用 heading error (yaw_error) 與 lateral error (ep) 作為狀態向量。
- 使用 LQR 計算最佳 delta，平衡追蹤與穩定性。
- 在彎道處降低速度，提高轉向穩定度。

5. controller_lqr_bicycle.pycontroller_lqr_bicycle

➤ TODO 邏輯：

- 類似 basic LQR，但用 v/l 建立自行車模型的 B 矩陣。
- 未設 lookahead，直接選最鄰近點控制。

➤ 設計理由：

- 自行車模型比 basic 多一階動態（控制量為 delta 而非 w ）。
- 使用狀態誤差來推估轉向角度，符合物理模型。

6. controller_pid_basic.pycontroller_pid_basic

➤ TODO 邏輯：

- 使用 PID 計算橫向誤差 ep 對應 yaw_error。
- 若接近目標，控制輸出為 0。

➤ 設計理由：

- 用 $\sin(\text{yaw_error}) \times \text{min_dist}$ 作為誤差來源，符合真實偏移。
- PID 控制方式簡單，適用於初學控制系統。

7. controller_pid_bicycle.pycontroller_pid_bicycle

➤ TODO 邏輯：

- 與 basic PID 相同，但輸出控制量為 delta（轉向角）。

➤ 設計理由：

- 自行車模型的控制角度需以 delta 表示，非 w 。
- 同樣利用 yaw_error 對轉向進行 PID 調節。

8. controller_pure_pursuit_basic.pycontroller_pure_pursuit...

➤ TODO 邏輯：

- 根據速度調整 L_d ，搜尋距離足夠的前方點作為追蹤目標。
- 計算 alpha 與角速度 next_w。
- 根據轉角限制自動減速。

➤ 設計理由：

- 加入最大角速度 MAX_W 限制，防止急轉。
- 利用弧度誤差比例減速控制速度輸出。

9. controller_pure_pursuit_bicycle.pycontroller_pure_pursuit...

➤ TODO 邏輯：

- 同上，但控制量改為 δ 。
- 使用公式 $\arctan(2L \sin(\alpha) / L_d)$ 計算轉向角。

➤ 設計理由：

- 自行車模型需要轉向角控制，使用幾何追蹤法。
- look-ahead 長度與速度有關，提升預判能力。

10. controller_stanley_bicycle.pycontroller_stanley_bicy...

➤ TODO 邏輯：

- 使用前輪位置計算 lateral error。
- 利用公式 $\delta = \arctan(-k_p * e / v) + \text{heading_error}$ 控制轉向角。

➤ 設計理由：

- Stanley 方法具良好穩定性，使用 error 項與速度比例調整偏移。
- 修正航向誤差與 lateral error 可穩定收斂至路徑。