DWA 代码

整体结构

你提供的 `dwa\_planner.py` 是一个非常完整、工程化程度高的\*\*动态窗口法（DWA, Dynamic Window Approach）路径规划器\*\*，集成了：

- 🧠 核心算法：DWA 路径规划与避障。

- 🖼️ 可视化界面：基于 Tkinter 实时仿真轨迹与障碍物交互。

- ⚙️ 参数调节器：GUI 修改和保存配置。

- 📁 配置管理器：基于 YAML 文件读写参数配置。

# 核心类 DWAPlanner

### 初始化阶段

构造函数 `\_\_init\_\_()`：

- 如果提供 `config\_manager`，优先读取配置文件 `dwa\_config.yaml`。

- 否则使用内建默认配置（最大速度、加速度、权重参数等）。

- 参数通过字典统一管理 `self.base\_config`。

# dwa\_planning() 方法

这是动态窗口法的核心实现：

输入参数：

- `current\_pos`: `[x, y, theta]` 当前位姿

- `current\_vel`: 当前线速度

- `current\_omega`: 当前角速度

- `goal`: `[x, y]` 目标点

- `obstacles`: 障碍物（支持静态和动态格式）

- `current\_state`: 当前行为状态，如 `"NORMAL\_NAVIGATION"`、`"OVERTAKE\_MANEUVER\_RIGHT"` 等

执行步骤：

1. 根据行为状态调整参数（避障权重、安全距离等）。

2. 根据当前速度和加速度计算动态窗口 `dw`。

3. 对速度对 `(v, omega)` 进行采样，预测轨迹并打分：

- 轨迹末端角度与目标方向差异（heading score）

- 距离目标距离（distance score）

- 当前速度（velocity score）

- 与障碍物距离（obstacle cost）

4. 综合评分找出最佳速度。

5. 返回最优 `(v, omega, trajectory)`。

# 轨迹预测 predict\_trajectory()

- 输入：当前位置、角度、速度、角速度、预测时间。

- 使用匀速圆周运动学模型预测轨迹。

- 小时间步下使用离散模型模拟未来路径。

# 障碍物代价 calc\_obstacle\_cost()

- 遍历预测轨迹每个点，检查是否与障碍物发生碰撞。

- 若碰撞（与障碍物距离 < 半径）返回 `inf`。

- 否则计算最小距离，根据当前安全距离生成 0~1 的代价值。

- 支持不同障碍物类型（vessel, buoy, debris）动态调整安全半径。

# SimulationVisualizer 模拟器

- 使用 Tkinter 实时绘图：机器人、障碍物、目标点、轨迹。

- 每个时间步更新机器人状态和障碍物状态。

- 调用 DWAPlanner 获取轨迹并可视化展示。

# 参数调节器 ParamTuner

- 使用 Tkinter 提供参数输入框界面。

- 用户可修改 max\_speed、加速度、各类权重等参数。

- 支持保存配置到 YAML 文件中。

# 配置管理器 ConfigManager

- 负责从 `config/dwa\_config.yaml` 中加载或创建默认参数。

- 支持保存修改后的配置文件。

# 运行方式

- 运行 `python3 dwa\_planner.py`：

- 启动仿真模拟窗口（可视化轨迹与障碍物）。

- 启动参数调节器 GUI。

- 可进行不同行为状态实验或路径规划策略对比。