Localpathplan 节点

1.功能：

功能根据/task_cmd 消息所传输的路径进行路径加密，并生成全局路径和局部路径。

2.发布消息

- localpath_pub = nh_local->advertise<nav_msgs::Path>("localpath", 10);（10m 之内的路径信息）
- globalpath_pub = nh_local->advertise<nav_msgs::Path>("globalpath", 10);（加密后各点信息）
- taskpath_pub = nh_local->advertise<nav_msgs::Path>("taskpath", 10);（原始路径的各点信息）
- shotpose_pub = nh_local->advertise<geometry_msgs::PoseStamped>("target_pose", 10);（原始路径最后一个点信息）
- pathmarkers_pub = nh_local->advertise<visualization_msgs::MarkerArray>("path_markers", 10);（发布路径速度与方向标识）
- remainpath_pub = nh_local->advertise<std_msgs::Float64>("remainpath", 10);（剩余的路径长度）
- passedpath_pub = nh_local->advertise<std_msgs::Float64>("passedpath", 10);（经过的路径长度）

3.订阅消息

- task_sub = nh->subscribe<mqtt_comm::task>("/task_cmd", 10, &TLocalPathPlan::TaskCallback, this);

所传递的 msg 是否存在 task 字符，有的话消息传给 cur_task，根据消息进行 GetGlobalPathFromTask(*msg); 和 GlobalPathPlan() 处理，最后得到一个路径 global_poses（插值并速度规划后的）；

- carstate_sub = nh->subscribe<data_comm::car_state>("/can_comm/car_state", 10, &TLocalPathPlan::CarStateCallback, this);

直接回传消息给 cur_carstate

- rviz_goal_sub = nh->subscribe<geometry_msgs::PoseStamped>("/move_base_simple/goal", 10, &TLocalPathPlan::RvizGoalCallback, this);（如果运行容易出问题）

根据从 rviz 中（ 2D Nav Goal 操作）的接收到的位置消息，生成一个移动任务并进行全局路径规划。根据目标点进行 GetGlobalPathFromTask(task); 和 GlobalPathPlan()

4.主函数

发布 taskpath_pub.publish(task_path); 两个回调函数处理完的数据

发布 globalpath_pub.publish(global_path);

进行局部路径规划 LocalPathPlan，并发布 localpath_pub.publish(local_path);

发布路径速度与方向标识 PubPathMarkers();

3.其他函数说明

- GetGlobalPathFromTask 函数含义： void TLocalPathPlan::GetGlobalPathFromTask(mqtt_comm::task task)

将传递进来的路径最后一个点的位置和姿态信息传递给 shotpose 并发布 shotpose_pub.publish(shotpose);（target_pose）

遍历每个点，将每个点的位置和姿态信息传递给 p_map task_path.poses.push_back(p_map);（task_path）

- GlobalPathPlan();函数含义： void TLocalPathPlan::GlobalPathPlan()

遍历任务路径中的所有路径点（最后一个点不需要遍历）。获取相邻的两个路径点的坐标信息。检查是否需要在下一个点停止 CheckStopAtNextPoin(task_path, i, true)。进行路径加密处理 LineInterpolation(p1, p2, ds, last_path_precision)，生成路径片段 path。设置路径片段的姿态信息。将路径片段添加到 plan_path 中。判断是否有折点需要停止，如果是，则进行速度规划 SpeedPlan(plan_path, 0.1, safe_dis)并将路径添加到全局路径 global_path 中。如果全局路径非空，将最后一个路径点的高度设为 0。

- CheckStopAtNextPoint 函数含义： bool TLocalPathPlan::CheckStopAtNextPoint(nav_msgs::Path path, int id, bool action_stop)

主要目的是通过比较路径的几何特性和方向变化，判断是否需要在下一个

点停止。动作标志位、运动模式变化、车头朝向变化、路径方向变化

- LineInterpolation 函数含义：nav_msgs::Path TLocalPathPlan::LineInterpolation(geometry_msgs::Point p1, geometry_msgs::Point p2, float ds, float last_ds)

函数的目的，通过在两个相邻的路径点之间进行插值，生成一段平滑（按设定的 ds 等分）路径，同时根据是否要停车进行最后两个点之间 last_ds 间距更细密的差值， 返回 path

- SpeedPlan 函数含义： void TLocalPathPlan::SpeedPlan(nav_msgs::Path &path, float min_vel, float safe_dis)

用于对给定的路径 path 进行速度规划。函数根据路径点之间的距离和目标速度，通过加速和减速约束，规划出路径上每个点的速度（加减速而非阶跃）

- LocalPathPlan 函数含义 void TLocalPathPlan::LocalPathPlan()

调用 UpdateLaneType，并找最短路径，根据当前点发布已经走过的路径 passedpath_pub.publish(data_msg); 和没有走过的路径 remainpath_pub.publish(data_msg)，遍历全局路径 global_path，从最近点开始构建本地路径 local_path，直到超过 10 米或满足某个停止条件 CheckStopAtNextPoint 为止。

UpdateLaneType()函数含义：void TLocalPathPlan::UpdateLaneType()

更新车道类型（lane_type），并在车辆当前任务路径上找到最近的点，如果找到新的车道类型，就更新参数。

- PubPathMarkers()函数含义：

在 rviz 中显示路径点的朝向等 pathmarkers_pub.publish(markerarray);

- GenFinalPathByPose：nav_msgs::Path TLocalPathPlan::GenFinalPathByPose(geometry_msgs::PoseStamped pose)

暂时无用