

Routing Planning

1. 作业依赖安装

Ubuntu 20 的同学运行脚本

```
~/routing_ws/src$ bash install_ros-noetic.sh
```

Ubuntu18 的同学运行脚本

```
~/routing_ws/src$ bash install_ros-melodic.sh
```

2. 程序启动

```
~/routing_ws$ source devel/setup.bash
```

```
~/routing_ws$ roslaunch src/run_map_simulator.launch
```

Rviz 操作见视频“routing 操作.mp4”。

3. 项目设计思路

主体思路

这一 Project 主要的思路是，首先通过通用数据格式地图，获取高精地图(HD-Map)信息，高精地图具有非常多的信息，但对于规划模块来说，更多关注的是车道和交通信息，因此我们用的更多的是 vector map 的数据结构信息。

第一步，加载地图，我们将通过 OSM(lanelet)地图消息的格式转换成规划内部用的数据结构格式的地图，通过 topic 发送，并在 rviz 进行可视化。

第二步，当我们拿到 pnc 地图后，根据车道和交通信息，进行 routing 搜索，得到起点和终点连接的最短 routing。

第三步，将我们拿到的 routing path 发送，给下游模块进行使用，并进行可视化。

我们可以借鉴的东西

现有的成熟框架很多，但于涉及地图和路由导航设计的模块代码大多很复杂，高精地图一直是自动驾驶的重中之重。对于这次设计，主要参考了 lanelet 地图结构设计，还有 Autoware 对 OSM 地图的 lanelet_extension 扩展应用。

在关注重点上，学员可以更多关注 mission planner 的实现顺序，了解我们拿到 vector 地图后，planner 模块是如何对它进行操作，并得到初始的全局路径的。其他的包更多的是对于地图这种需要大量复杂几何操作的一个数学功能支持。

课程设计流程图

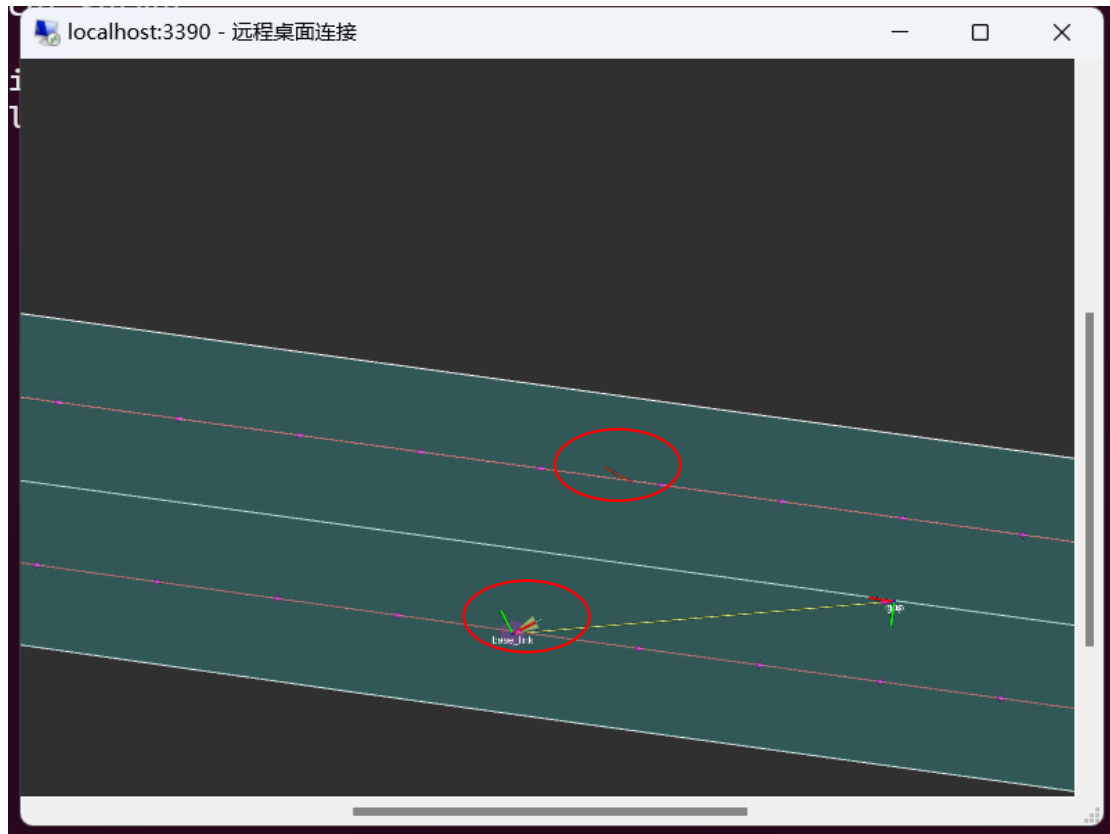


4. 作业

mission_planner package 中的 mission_planner_lanelet2.cpp

```
257
258     // TODO(shenlan):
259     // 起点和终点的剪裁
260     global_path.header.frame_id = map_frame_;
261     global_path.header.stamp = ros::Time::now();
262     global_path.poses.clear();
263     for (const auto& path_lanelet : path_lanelets) {
264         const auto& line = path_lanelet.centerline2d();
265         for (const auto& pt : line) {
266             geometry_msgs::Pose pt_pose;
267             pt_pose.position.x = pt.x();
268             pt_pose.position.y = pt.y();
269             pt_pose.position.z = 0.0;
270             global_path.poses.push_back(pt_pose);
271         }
272     }
273
```

目前项目中的全局路径是规划所有的 lanelet 段中的离散点的集合，我们需要转换成从起点到终点的一系列路径点给下游规控模块使用。Lanelet 车道内的离散点均匀位于车道线上，同学们可以利用几何知识对起点与终点进行剪裁（即全局路径切割），得到从起点到终点的一系列全局路径点。



5. 提升作业（选做）

目前离散的全局规划路径已经得出，为了全局规划和 SVL 联合在线仿真，同学们可以替换程序中的通过 rviz 打点获取的车辆起点（改为获取/odom 得到），然后使用先前课程代码中的样条曲线对全局规划轨迹进行平滑，使用先前课程中的轨迹跟踪算法进行轨迹跟踪。