深蓝学院
shenlanxueyuan.com

第八章思路提示

主讲人 夏韵凯

标题

深蓝学院
shenlanxueyuan.com

- 本章的作业比较简单，是在lanelet2库的全局规划结果的基础上，根据车辆起点与目标终点进行剪裁，利用lanelet2中的几何库就可以实现。
- 具体可以参考https://github.com/fzi-forschungszentrum-informatik/Lanelet2/blob/master/lanelet2_examples/src/01_dealing_with_lanelet_primitives/main.cpp

标题

深蓝学院
shenlanxueyuan.com

- 首先是根据起点与终点的位置，映射到起点与终点的lanelet段上的点

```
int lanelet_size = path_lanelets.size();
std::cout << "lanelet_size is " << lanelet_size << std::endl;
auto start_lanelet_pt = lanelet::geometry::project(
    path_lanelets[0].centerline2d(),
    lanelet::BasicPoint2d{start_checkpoint.pose.position.x,
                          start_checkpoint.pose.position.y});
auto end_lanelet_pt = lanelet::geometry::project(
    path_lanelets[lanelet_size - 1].centerline2d(),
    lanelet::BasicPoint2d{goal_checkpoint.pose.position.x,
                          goal_checkpoint.pose.position.y});
```

标题

深蓝学院
shenlanxueyuan.com

- 然后找到对应lanelet段上最近点的id

```
int start_index =
    clacCloestPathNodeId(path_lanelets[0].centerline2d(), start_lanelet_pt);
int end_index = clacCloestPathNodeId(
    path_lanelets[lanelet_size - 1].centerline2d(), end_lanelet_pt);
```

```
int MissionPlannerLanelet2::clacCloestPathNodeId(
    const lanelet::ConstLineString2d& center_line,
    const lanelet::BasicPoint2d& point) {
  float min_dist = 1e6;
  int cloest_node_id = -1;
  for (int i = 0; i < center_line.size(); i++) {
    auto dx = point.x() - center_line[i].x();
    auto dy = point.y() - center_line[i].y();
    auto dis = sqrt(dx * dx + dy * dy);
    if (dis < min_dist) {
      min_dist = dis;
      cloest_node_id = i;
    }
  }
  return cloest_node_id;
}
```

标题

深蓝学院 shenlanxueyuan.com

- 使用五次样条曲线进行拟合

```
using namespace shenlan;
shenlan::Vec_f wx;
shenlan::Vec_f wy;
float last_wx, last_wy;
float sum_s = 0;
```

规划结果只有一段

```
wx.push_back(start_lanelet_pt.x());
wy.push_back(start_lanelet_pt.y());
last_wx = start_lanelet_pt.x();
last_wy = start_lanelet_pt.y();

if (lanelet_size == 1) {
  auto center_line = path_lanelets[0].centerline2d();
  for (int j = start_index; j < end_index; j++) {
    auto dx = center_line[j].x() - last_wx;
    auto dy = center_line[j].y() - last_wy;
    last_wx = center_line[j].x();
    last_wy = center_line[j].y();
    auto s = std::sqrt(dx * dx + dy * dy);
    sum_s += s;
    if (sum_s > 10.0) {
      wx.push_back(center_line[j].x());
      wy.push_back(center_line[j].y());
      ROS_INFO_STREAM("waypt, x= " << wx.back() << ", y= " << wy.back());
      sum_s = 0;
    }
  }
}
```

起始段

```
else {
  //至少有两条lanelet
  std::cout << "lanelet_size is " << lanelet_size << std::endl;

  for (int i = 0; i < lanelet_size; i++) {
    auto center_line = path_lanelets[i].centerline2d();
    if (i == 0) {
      for (int j = start_index; j < cen   float last_wx ; j++) {
        auto dx = center_line[j].x() - last_wx;
        auto dy = center_line[j].y() - last_wy;
        last_wx = center_line[j].x();
        last_wy = center_line[j].y();
        auto s = std::sqrt(dx * dx + dy * dy);
        sum_s += s;
        if (sum_s > 10.0) {
          wx.push_back(center_line[j].x());
          wy.push_back(center_line[j].y());
          ROS_INFO_STREAM("waypt, x= " << wx.back()
                                       << ", y= " << wy.back());
          sum_s = 0;
        }
      }
    }
```

中间段

```
    else if (i < lanelet_size - 1) {
      for (int j = 0; j < center_line.size(); j++) {
        auto dx = center_line[j].x() - last_wx;
        auto dy = center_line[j].y() - last_wy;
        last_wx = center_line[j].x();
        last_wy = center_line[j].y();
        auto s = std::sqrt(dx * dx + dy * dy);
        sum_s += s;
        if (sum_s > 10.0) {
          wx.push_back(center_line[j].x());
          wy.push_back(center_line[j].y());
          ROS_INFO_STREAM("waypt, x= " << wx.back()
                                       << ", y= " << wy.back());
          sum_s = 0;
        }
      }
    }
```

终点段

```
    else {
      for (int j = 0; j < end_index; j++) {
        auto dx = center_line[j].x() - last_wx;
        auto dy = center_line[j].y() - last_wy;
        last_wx = center_line[j].x();
        last_wy = center_line[j].y();
        auto s = std::sqrt(dx * dx + dy * dy);
        sum_s += s;
        if (sum_s > 10.0) {
          wx.push_back(center_line[j].x());
          wy.push_back(center_line[j].y());
          ROS_INFO_STREAM("waypt, x= " << wx.back()
                                       << ", y= " << wy.back());
          sum_s = 0;
        }
      }
    }
  }
}
wx.push_back(end_lanelet_pt.x());
wy.push_back(end_lanelet_pt.y());
auto global_spline = Spline2D(wx, wy);
```

标题

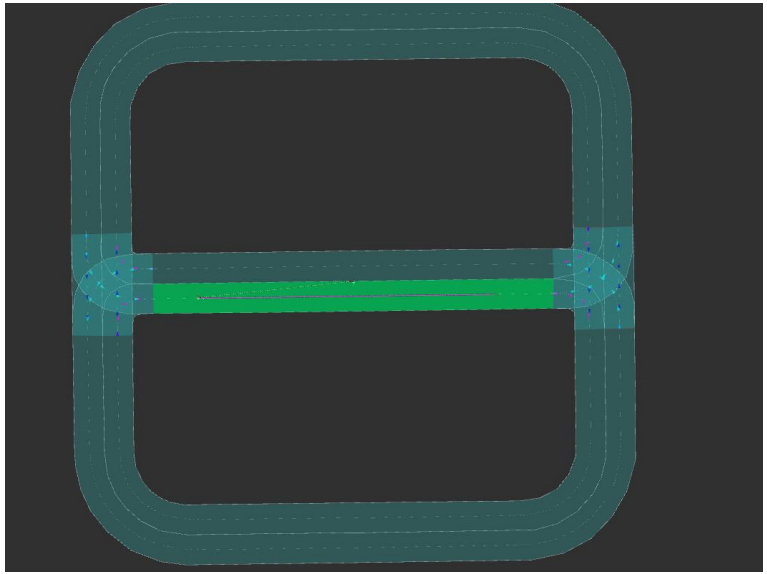
深蓝学院
shenlanxueyuan.com

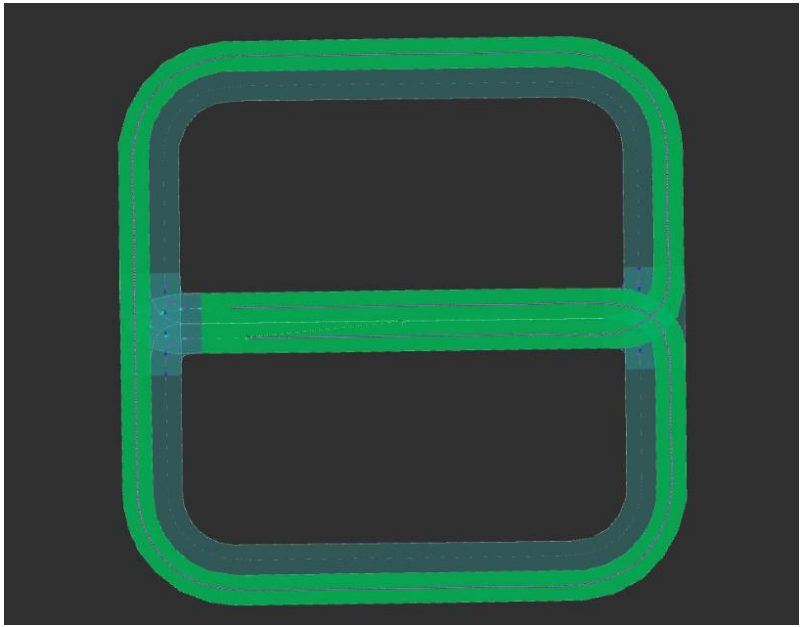
- 使用五次样条曲线进行拟合

```
global_path.header.frame_id = map_frame_;
global_path.header.stamp = ros::Time::now();
global_path.poses.clear();
std::cout << "global_spline.s.back() is " << global_spline.s.back()
          << std::endl;
for (float i = 0; i < global_spline.s.back(); i += 0.1) {
  auto point = global_spline.calc_postion(i);
  geometry_msgs::Pose pt_pose;
  pt_pose.position.x = point[0];
  pt_pose.position.y = point[1];
  pt_pose.position.z = 0.0;
  global_path.poses.push_back(pt_pose);
  // std::cout << "x is " << point[0] << std::endl;
  // std::cout << "y is " << point[1] << std::endl;
}
```

结果展示

深蓝学院
shenlanxueyuan.com





在线问答

深蓝学院
shenlanxueyuan.com

Q&A

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening