深蓝学院
shenlanxueyuan.com

第三章作业思路提示

主讲人 高宇辉

作业内容：Stanley Method

深蓝学院 shenlanxueyuan.com

需要完成 stanley_control.cpp 中 todo 部分，分别是：

```
// /**to-do**/ 实现stanley算法
void StanleyController::ComputeControlCmd(
    const VehicleState &vehicle_state,
    const TrajectoryData &planning_published_trajectory, ControlCmd &cmd)
```

```
// /**to-do**/ 计算误差
void StanleyController::ComputeLateralErrors(const double x, const double y,
                                             const double theta, double &e_y,
                                             double &e_theta) {
```

作业内容：Stanley Method

深蓝学院 shenlanxueyuan.com

- Stanley control law

$$\delta(t) = \theta_e(t) + tan^{-1}\left(\frac{ke(t)}{v_f(t)}\right), \delta(t) \in [\delta_{min}, \delta_{max}]$$

首先是ComputerControlCmd函数，这一部分比较简单，就是根据Stanley算法的公式进行代码编写，所以需要调用接下来的误差计算函数，然后整个前轮转角控制命令分为两部分，分别是由航向误差和由横向误差引起的转角。需要注意的是：

1.计算反正切函数值时，建议使用atan2函数，其返回值为点和原点连线与x轴正方向的夹角，值域对应为-pi到+pi；

2.实际的前轮转角有一个范围，即δ（t）∈[δmin，δmax]，所以需要对其进行限幅处理。

代码实现

深蓝学院
shenlanxueyuan.com

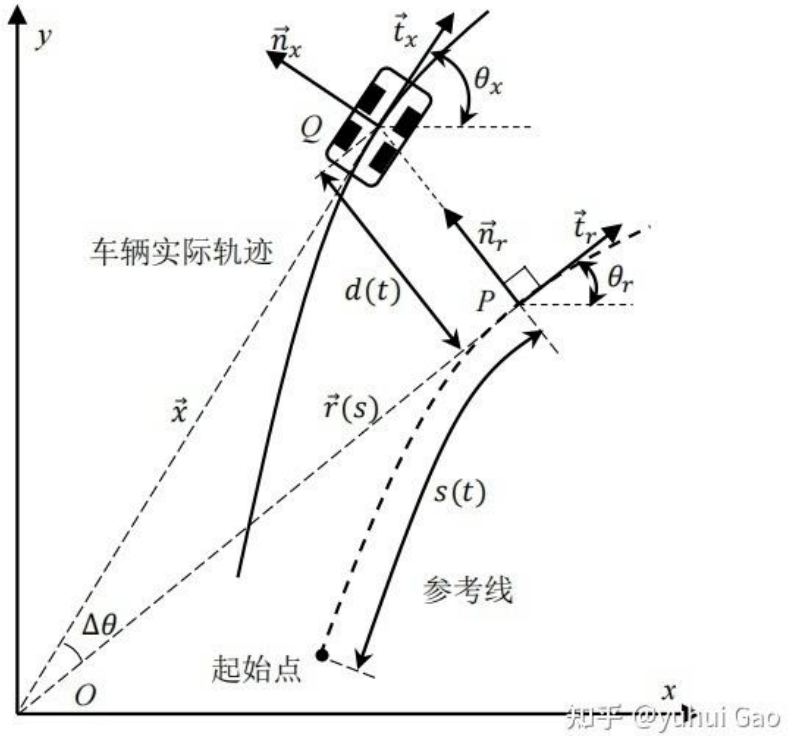
```
// /** to-do **/ 计算需要的控制命令,
// 实现对应的stanley模型,并将获得的控制命令传递给汽车
// 提示，在该函数中你需要调用计算误差
void StanleyController::ComputeControlCmd(
    const VehicleState &vehicle_state,
    const TrajectoryData &planning_published_trajectory, ControlCmd &cmd) {
  //控制的轨迹
  trajectory_points_ = planning_published_trajectory.trajectory_points;
  double e_y = 0.0;
  double e_theta = 0.0;

  cout << "vehicle_state.heading: " << vehicle_state.heading << endl;
  ComputeLateralErrors(vehicle_state.x, vehicle_state.y, vehicle_state.heading,
                       e_y, e_theta);

  cout << "e_theta: " << e_theta << endl;
  cout << "e_y: " << e_y << endl;
  double steer_angle =
      (e_theta + std::atan2(k_y_ * e_y, vehicle_state.velocity));

  // 控制对应的转角
  if (steer_angle > M_PI / 12) {
    steer_angle = M_PI / 12;
  } else if (steer_angle < -M_PI / 12) {
    steer_angle = -M_PI / 12;
  }
```

作业内容：Stanley Method



然后是ComputeLateralErrors函数，也就是分别表示航向误差e_theta和横向误差e_y。通过QueryNearestPointByPosition可以得到距离当前自车位置最近的点。

1.对于航向误差，即车身方向与参考轨迹最近点的切线方向的夹角，使用自车航向角减去参考点航向角（ e_theta=θx-θr ），并转换到-pi到+pi之间即可。

2.对于横向误差，需要进行判断，笛卡尔坐标系下自车位置和参考点之间的距离可以表示为e_y=±sqrt((xr-x)^2, (yr-y)^2,)，其中，若(yr-y)cosθr- (xr-x)sinθr＞0， e_y为负，否则e_y为正。

代码实现


```
// /** to-do **/ 计算需要的误差，包括横向误差，航向误差
void StanleyController::ComputeLateralErrors(const double x, const double y,
                                             const double theta, double &e_y,
                                             double &e_theta) {
 TrajectoryPoint target_point;
 target_point = QueryNearestPointByPosition(x, y);

 double heading = target_point.heading;
 const double dx = target_point.x - x;
 const double dy = target_point.y - y;
 e_y = sqrt(dx * dx + dy * dy);
 double fx = dy * cos(heading) - dx * sin(heading);  // 横向误差
 if (fx > 0) {
   e_y = -abs(e_y);
 } else {
   e_y = abs(e_y);
 }
 e_theta = theta - heading;  //航向偏差
 if (e_theta > M_PI) {
   e_theta = -2.0 * M_PI + e_theta;
 } else if (e_theta < -M_PI) {
   e_theta = 2.0 * M_PI + e_theta;
 }
 return;
}
```

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