

1. 自动驾驶轨迹跟踪控制

1.1 任务定义

基于自然驾驶场景，搭建仿真测试场景，基于搭建的场景和给定的全局路径、车辆位置信息等进行循迹，车辆为半载工况。其中，路径点间隔 0.2m – 1.5m，每个路径点包含车辆速度、加速度、曲率、航向角等信息；场景包括：

- 1) 长直道：100m - 1000m；
- 2) 大曲率弯道：U 型弯（转弯半径 10m – 50m）、直角弯（转弯半径 8m – 20m）；
- 3) 小曲率弯道：转弯半径 80m – 300m；
- 4) 上下坡及过渡：上坡水平距离为 20m – 60m；下坡水平距离为 20m – 60m；上下坡之间的水平路面长度 20m – 50m。

注意：仿真测试场景包含 1~2 个 1) -4) 各场景。

1.2 环境部署

环境部署分为 Ubuntu20.04 系统安装、换源、安装 Nvidia 显卡驱动、VTD 安装、VTD 工程文件导入、docker 安装、ROS1 安装、输入输出接口功能包部署，[详见附件 1-环境部署.pdf](#)。

1.3 输入输出说明

算法输入输出以 ROS 话题的形式给出

1.3.1 输入信息

- (1) 定位信息：/cicv_location

消息格式：perception_msgs/PerceptionLocalization

PerceptionLocalization.msg	
Header header	
uint64 frame_unmber	
int8 fusion_level	
int8 status	#4:固定解, 5: 浮点解, 2: 伪距差分, 1: 单点解
#rotation	
float64 roll	# x,R, car body (FLU)
float64 pitch	# y,R, car body (FLU)
float64 yaw	# z,R, car body (FLU) utm (0,360) 0--east, 90--north
float32 roll_std	

```
float32 pitch_std

float32 yaw_std

float64 qw          # w
float64 qx          # x
float64 qy          # y
float64 qz          # z

float64 angular_velocity_x
float64 angular_velocity_y
float64 angular_velocity_z

#position

float64 latitude

float64 longitude

float64 altitude

float32 latitude_std
float32 longitude_std
float32 altitude_std

float64 position_x   # utm_east
float64 position_y   # utm_north
float64 position_z   # utm_up

float32 position_x_std
float32 position_y_std
float32 position_z_std

float64 velocity_x   # utm_east_speed
float64 velocity_y   # utm_north_speed
float64 velocity_z   # utm_up_speed

float32 velocity_x_std
float32 velocity_y_std
float32 velocity_z_std

float64 velocity_rx   # ins_x_speed
```

float64 velocity_ry	# ins_y_speed
float64 velocity_rz	# ins_z_speed
float32 velocity_rx_std	
float32 velocity_ry_std	
float32 velocity_rz_std	
float64 accel_x	# R, car body (RFU)
float64 accel_y	# F, car body (RFU)
float64 accel_z	# U, car body (RFU)

(2) 轨迹信息: /cicv_amr_trajectory

消息格式: perception_msgs/Trajectory

Trajectory.msg
Header header
uint64 frame_number
bool is_valid_frame
TrajectoryInfo trajectoryinfo

TrajectoryInfo.msg
int32 path_id
float32 total_path_length
float32 total_path_time
int8 decision_type
int8 light_type
string[] lane_ids
TrajectoryPoint[] trajectorypoints

TrajectoryPoint.msg
Point2D position
float32 velocity
float32 heading

float32 curvature
float32 s
float32 t
int8 point_type

上述信息 msg，详见附件 perception_msgs。

1.3.2 输出信息

(1) 控制指令话题：/control_test

消息格式：/common_msgs/ control_test.msg

control_test.msg	
std_msgs/Header	header #消息头（包含时间戳和帧 ID）
uint32 Gear	#档位指令，1-P 挡 2-R 挡 3-N 挡 4-D 挡，默认为 4-D 挡
float32 ThrottlePedal	#加速踏板开度指令，范围为 0-1
float32 BrakePedal	#制动踏板开度指令，范围为 0-1
float32 SteeringAngle	#方向盘转角控制指令，单位：度，范围为-540-540，左正右负

上述信息 msg，详见附件 2。

参赛队伍的控制算法下发控制指令的频率统一为 100hz。测评时，会记录车辆跑过的实际路径点，计算车速准确性、车辆横向误差、加速度误差、车辆航向角误差时，统一使用距离每个全局路径点最近的实际路径点。因此，参赛队伍自测时注意记录实际跑过的路径点的车速、加速度、航向角等信息。

1.4 完美传感器参数

序号	名称	水平视场角/°	垂直视场角/°	探测范围/米
1	sensor1	360	60	200
2	sensor2	75	42	150

1.5 车辆参数

参数名称	参数
车长	5990(mm)
车宽	2065(mm)
车高	2820(mm)
整备质量	3990(kg)
轴距	3800(mm)

最大设计时速	60(km/h)
轮胎规格	215/75R17.5
转向传动比	29
驱动方式	前驱

1.6 评价指标

本任务以平均车速误差、最大车速误差、最大加速度误差、平均横向误差、最大横向误差、平均航向角误差、最大航向角误差、终点纵向误差、终点横向误差、终点航向角误差等 5 类 10 个指标进行评价。

假设全局路径点集合为 $P=\{p_1, p_2, p_3, \dots, p_n\}$, 参赛算法跑完全局路径后得到其实际行驶路径点 $Q=\{q_1, q_2, q_3, \dots, q_n\}$, q_n 是距离全局路径点 p_n 最近的实际路径点。

(1) 车速准确性

1) 平均车速误差:

$$average_speed_error = average \sum_1^n fabs(p_i.v - q_i.v)$$

2) 最大车速误差:

$$max_speed_error = \max \{ fabs(p_1.v - q_1.v), \dots, fabs(p_n.v - q_n.v) \}$$

(2) 加速度准确性 (最大加速度误差)

$$max_accel_error = \max \{ fabs(p_1.a - q_1.a), \dots, fabs(p_n.a - q_n.a) \}$$

(3) 车辆横向误差

1) 平均横向误差:

$$average_lateral_error = average \sum_1^n fabs(p_i.t - q_i.t)$$

2) 最大横向误差:

$$max_lateral_error = \max \{ fabs(p_1.t - q_1.t), \dots, fabs(p_n.t - q_n.t) \}$$

(4) 车辆航向角误差

1) 平均航向角误差:

$$average_heading_error = average \sum_1^n fabs(p_i.heading - q_i.heading)$$

2) 最大航向角误差:

$$max_speed_error = \max \{ fabs(p_1.heading - q_1.heading), \dots, fabs(p_n.heading - q_n.heading) \}$$

(5) 车辆终点停车精确性

设跑完路径全程后，本车最终停车位置为 $P(x_{veh}, y_{veh}, heading_{veh})$ ，终点坐标为 $Q(x_{ter}, y_{ter}, heading_{ter})$ ，将终点坐标转换到车体坐标下（车辆前方为 x 方向，车辆左侧方向为 y 方向，z 方向指向天空），得到 $Q_1(x_{ter_1}, y_{ter_1}, heading_{ter_1})$ ，则终点横、纵向误差，航向角误差计算公式如下：

1) 终点纵向误差：

$$terminal_longitudinal_error = fabs(x_{ter_1})$$

2) 终点横向误差：

$$terminal_lateral_error = fabs(y_{ter_1})$$

3) 终点航向角误差：

$$terminal_heading_error = fabs(heading_{veh} - heading_{ter})$$

1.7 评分规则：

(1) 总分为 100 分。评价指标共 10 条，每一条指标满分 100 分。其中，平均车速误差、最大加速度误差、平均横向误差所占权重各为 12%，终点横向误差、终点纵向误差、终点航向角误差所占权重各为 8%，其他各指标权重为 10%，总分计算方式如下：

$$\text{总分} = \sum (\text{各指标得分} \times \text{权重})$$

为了保证参赛作品质量，总分大于 50.0 分的参赛队伍才有机会获得奖金。

(2) 平均车速误差指标：误差在 0~0.3m/s 范围内，得分 85~100；误差在 0.3~0.6m/s 范围内，得分 60~85；误差超过 0.6m/s，得 0 分。具体得分根据误差在对应得分区间线性插值得到。平均车速误差得分=长直道路段平均车速误差得分×30%+其他路段平均车速误差得分×70%。

(3) 最大车速误差指标：误差在 0~0.4m/s 范围内，得分 80~100；误差在 0.4~0.7m/s 范围内，得分 60~80；误差超过 0.7m/s，得 0 分。具体得分根据误差在对应得分区间线性插值得到。

(4) 最大加速度误差指标：误差在 0~0.2m/s² 范围内，得分 80~100；误差在 0.2~0.5m/s² 范围内，得分 60~80；误差超过 0.5m/s²，得 0 分。具体得分根据误差在对应得分区间线性插值得到。

(5) 平均横向误差指标：误差在 0~10cm 范围内，得分 90~100；误差在 10~30cm 范围内，得分 70~90；误差在 30~50cm 范围内，得分 60~70；误差超过 50cm，得 0 分。具体得分根据误差在对应得分区间线性插值得到。平均横向误差得分=长直道路段平均横向误差得分×30%+其他路段平均横向误差得分×70%。

(6) 最大横向误差指标：误差在 0~15cm 范围内，得分 90~100；误差在 15~35cm 范围内，得分

70~90; 误差在 35~55cm 范围内, 得分 60~70; 误差超过 55cm, 得 0 分。具体得分根据误差在对应得分区间线性插值得到。

(7) 平均航向角误差指标: 误差在 0~10°范围内, 得分 90~100; 误差在 10°~20°范围内, 得分 75~90; 误差在 20°~30°范围内, 得分 60~75; 误差超过 30°, 得 0 分。具体得分根据误差在对应得分区间线性插值得到。平均航向角误差得分=长直道路段平均航向角误差得分×30%+其他路段平均航向角误差得分×70%。

(8) 最大航向角误差指标: 误差在 0~15°范围内, 得分 90~100; 误差在 15°~25°范围内, 得分 75~90; 误差在 25°~35°范围内, 得分 60~75; 误差超过 35°, 得 0 分。具体得分根据误差在对应得分区间线性插值得到。

(9) 终点纵向误差指标: 误差在 0~5cm 范围内, 得分 90~100; 误差在 5~20cm 范围内, 得分 75~90; 误差在 20~50cm 范围内, 得分 60~75; 误差超过 50cm, 得 0 分。具体得分根据误差在对应得分区间线性插值得到。

(10) 终点横向误差指标: 误差在 0~5cm 范围内, 得分 90~100; 误差在 5~20cm 范围内, 得分 75~90; 误差在 20~50cm 范围内, 得分 60~75; 误差超过 50cm, 得 0 分。具体得分根据误差在对应得分区间线性插值得到。

(11) 终点航向角误差指标: 误差在 0~10°范围内, 得分 90~100; 误差在 10°~20°范围内, 得分 80~90; 误差在 20°~30°范围内, 得分 60~80; 误差超过 30°, 得 0 分。具体得分根据误差在对应得分区间线性插值得到。

1.8 比赛规则及作品要求

1.8.1 比赛规则

- (1) 每个团队在作品提交期间可提交 1 次;
- (2) 比赛期间会提供样例数据;
- (3) 决赛拟邀请预赛前 20 名参加;
- (4) 作品必须保证原创性, 不违反任何中华人民共和国有关法律法规, 不侵犯任何第三方知识产权或其他权利, 一经发现或经权利人提出并查证, 组织方将取消其参与资格和成绩并进行严肃处理;
- (5) 参赛选手需要配合组织方对作品的有效性 with 真实性进行验证, 同时自行检查提交作品的正确性, 确认无误后再进行提交, 组织方不负责对比赛作品进行更改和调整。

1.8.2 作品要求

- (1) 提供 Docker 镜像及 Dockerfile 文件: Docker 镜像内需包含所需环境、算法源代码等, 并以“任

务编号_队伍名称.tar”进行命名，如 task1_zhangsan.tar;

(2) 提供详细说明文档：包含任务算法整体描述、解决思路、架构设计、运行指令说明等，文件格式为 pdf;

(3) 其他相关支撑材料，如：辅助展示设计、方案材料、演示 demo 视频等（不强制）;

(4) 以上所有材料以“任务序号_队伍名称_版本号.tar”进行命名，如 task1_zhangsan_v1.tar;

(5) 作品原创：作品必须保证原创性，不违反任何中华人民共和国有关法律法规，不侵犯任何第三方知识产权或其他权利，一经发现或经权利人提出并查证，组织方将取消参与资格和成绩并进行严肃处理;

(6) 作品复现及验证：参赛选手需要配合组织方对作品的有效性与真实性进行验证，同时自行检查提交作品的正确性，确认无误后再进行提交，组织方不负责对比赛作品进行更改和调整。