深蓝学院
shenlanxueyuan.com

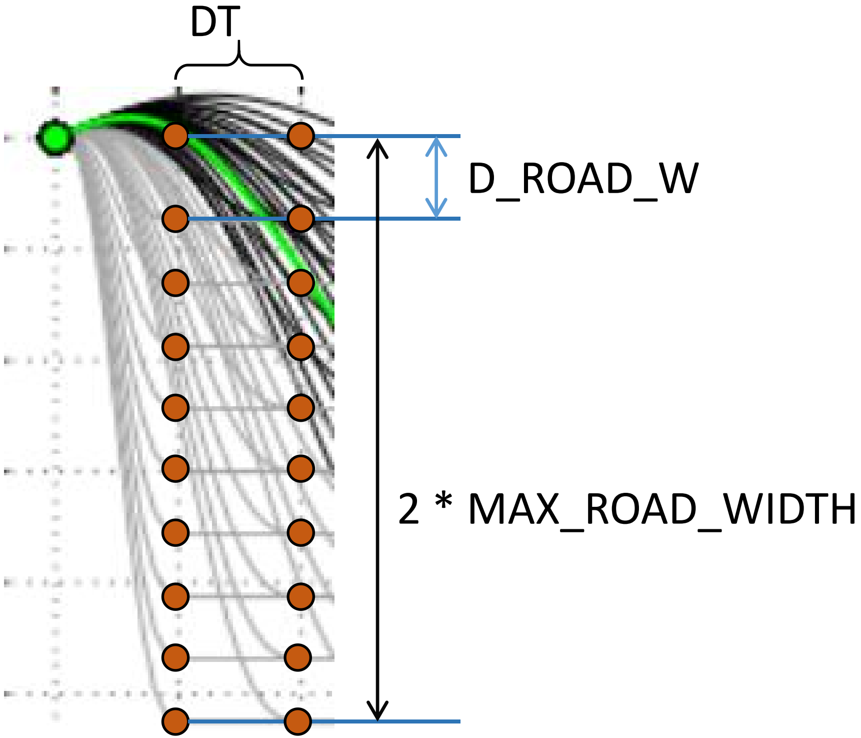
自动驾驶控制与规划
第六章思路提示

主讲人 助教-邱润其

思路提示

- Part 1 横向采样

```
for (float di = -1 * MAX_ROAD_WIDTH; di <
MAX_ROAD_WIDTH; di += D_ROAD_W) {
    for (float Ti = MINT; Ti < MAXT; Ti += DT)
    {
      FrenetPath fp;
      QuinticPolynomial lat_qp(…);
      for (float t = 0; t < Ti; t += DT) {
        fp.t.push_back(t);
        fp.d.push_back(……);
        fp.d_d.push_back(……);
        fp.d_dd.push_back(……);
        fp.d_ddd.push_back(……);
      }
    }
}
```

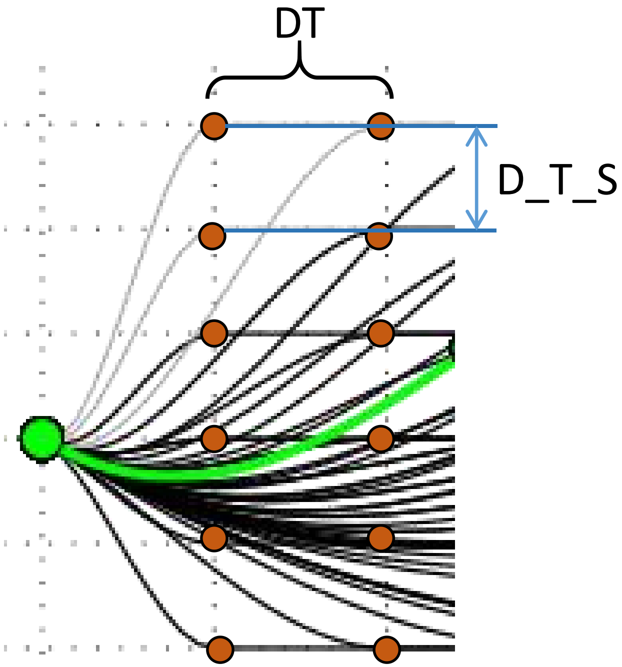


思路提示

- Part 2 纵向采样

```
for (float Ti = MINT; Ti < MAXT; Ti += DT) {
  for (float tv = TARGET_SPEED - D_T_S*N_S_SAMPLE;
       tv < TARGET_SPEED + D_T_S*N_S_SAMPLE;
       tv += D_T_S) {
    FrenetPath fp_bot = fp;
    QuarticPolynomial lon_qp(……);

    for (float t_ : fp.t) {
      fp_bot.s.push_back(……);
      fp_bot.s_d.push_back(……);
      fp_bot.s_dd.push_back(……);
      fp_bot.s_ddd.push_back(……);
    }
  }
}
```



思路提示

- Part 3 建立横纵向采样轨迹的评价函数

```
float Jp = sum_of_power(fp.d_ddd);
float Js = sum_of_power(fp_bot.s_ddd);
float ds = (TARGET_SPEED - fp_bot.s_d.back());

fp_bot.cd = KJ * Jp + KT * Ti + KD *
std::pow(fp_bot.d.back(), 2);
fp_bot.cv = KJ * Js + KT * Ti + KD * ds;
fp_bot.cf = KLAT * fp_bot.cd + KLON * fp_bot.cv;

fp_list.push_back(fp_bot);
```

$$C_d = k_j J_t(d(t)) + k_t T + k_d d_1^2,$$

$$C_v = k_j J_t(s(t)) + k_t T + k_{\dot{s}}[\dot{s}_1 - \dot{s}_d]^2$$

Conjoint function

思路提示

- Part 4 主函数部分

```
FrenetPath FOT::frenet_optimal_planning(……) {
  Vec_Path fp_list = calc_frenet_paths(……);
  calc_global_paths(……);

  Vec_Path save_paths = check_paths(……);
```

筛掉速度、曲率、加速度越界的轨迹

```
  float min_cost = std::numeric_limits<float>::max();
  FrenetPath final_path;
  for (auto path : save_paths) {
    if (min_cost >= path.cf) {
      min_cost = path.cf;
      final_path = path;
    }
  }
  return final_path;
}
```

从轨迹中挑选评价指标最优的

参考资料

- B站up主 忠厚老实的老王 — “自动驾驶决策规划算法”系列视频
- 论文 “Optimal Trajectory Generation for Dynamic Street Scenarios in a Frenet Frame”
- Github 仓库：https://github.com/fjp/frenet （用 Python 和 MATLAB 实现的 Lattice Planner)
- 关于五次多项式，可参考论文 “Local Path Planning and Motion Control for Agv in Positioning”

在线问答

Q&A

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening