

问题记录

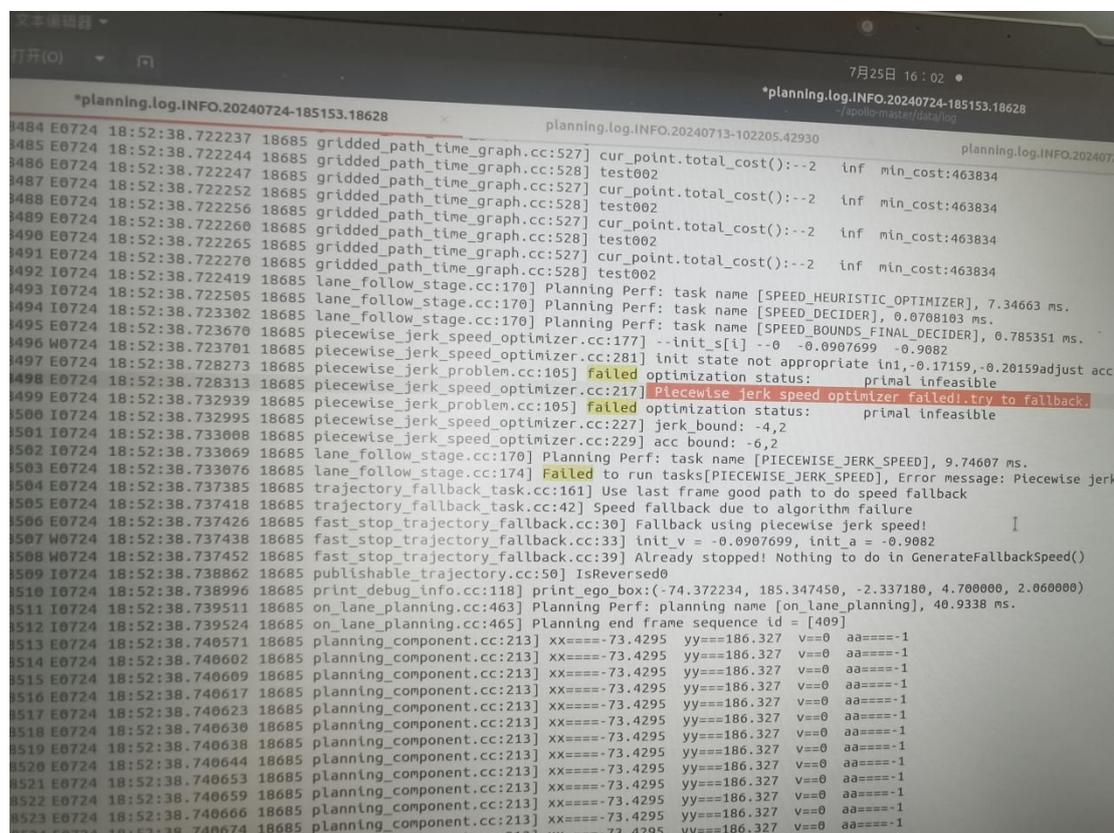
1.代码版本：

Apollo9.0
Carla 0.9.15
Bridge 0.9.14

2.存在问题：

在 Apollo 与 Carla 联合仿真时，如果 bridge 中发送车辆状态消息的频率太快（周期 $< 0.1s$ ），就会出现走走停停的现象，即车子给了油门车速刚起来一点就会刹车。设定频率减小后，异常现象会有缓解。推测是因为设定周期大时（或 carla 反馈数据延迟） $vehicle_vel_time > 0.1$ ，此时会跳出图 2 中的逻辑（因此时 $prev_trajectory.start_point().relative_time = 0.1s$ ）；

planning 的 log 文件报错如下：



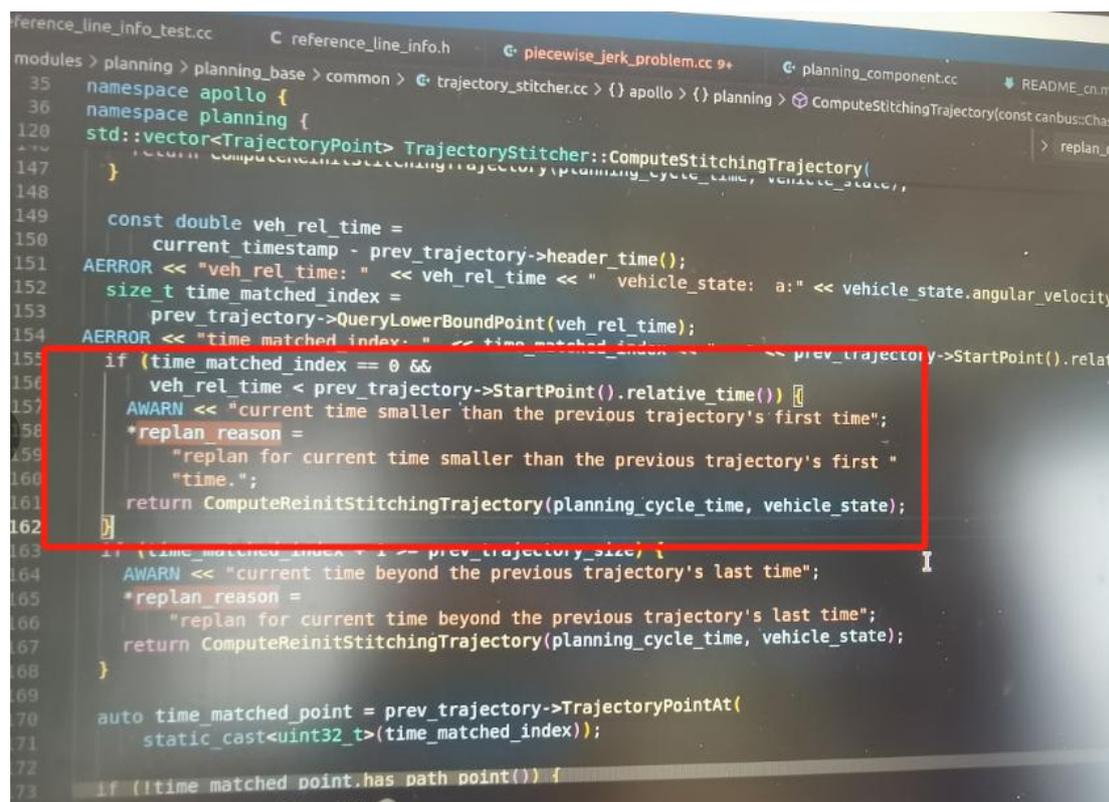
```
*planning.log.INFO.20240724-185153.18628
484 E0724 18:52:38.722237 18685 gridded_path_time_graph.cc:527] cur_point.total_cost():--2 inf min_cost:463834
485 E0724 18:52:38.722244 18685 gridded_path_time_graph.cc:528] test002
486 E0724 18:52:38.722247 18685 gridded_path_time_graph.cc:527] cur_point.total_cost():--2 inf min_cost:463834
487 E0724 18:52:38.722252 18685 gridded_path_time_graph.cc:528] test002
488 E0724 18:52:38.722256 18685 gridded_path_time_graph.cc:527] cur_point.total_cost():--2 inf min_cost:463834
489 E0724 18:52:38.722260 18685 gridded_path_time_graph.cc:528] test002
490 E0724 18:52:38.722265 18685 gridded_path_time_graph.cc:527] cur_point.total_cost():--2 inf min_cost:463834
491 E0724 18:52:38.722270 18685 gridded_path_time_graph.cc:528] test002
492 I0724 18:52:38.722419 18685 lane_follow_stage.cc:170] Planning Perf: task name [SPEED_HEURISTIC_OPTIMIZER], 7.34663 ms.
493 I0724 18:52:38.722505 18685 lane_follow_stage.cc:170] Planning Perf: task name [SPEED_DECIDER], 0.0708103 ms.
494 I0724 18:52:38.723302 18685 lane_follow_stage.cc:170] Planning Perf: task name [SPEED_BOUNDS_FINAL_DECIDER], 0.785351 ms.
495 E0724 18:52:38.723670 18685 piecewise_jerk_speed_optimizer.cc:177] --init_s[i] --0 -0.0907699 -0.9082
496 W0724 18:52:38.723701 18685 piecewise_jerk_problem.cc:105] failed optimization status: primal infeasible
497 E0724 18:52:38.728273 18685 piecewise_jerk_speed_optimizer.cc:217] Ptecewise jerk speed optimizer failed, try to fallback.
498 E0724 18:52:38.728313 18685 piecewise_jerk_problem.cc:105] failed optimization status: primal infeasible
499 E0724 18:52:38.732939 18685 piecewise_jerk_speed_optimizer.cc:227] jerk bound: -4,2
500 I0724 18:52:38.733008 18685 piecewise_jerk_speed_optimizer.cc:229] acc bound: -6,2
501 E0724 18:52:38.733069 18685 lane_follow_stage.cc:170] Planning Perf: task name [PIECEWISE_JERK_SPEED], 9.74607 ms.
502 E0724 18:52:38.733076 18685 lane_follow_stage.cc:174] Failed to run tasks[PIECEWISE_JERK_SPEED], Error message: Piecewise jerk
503 E0724 18:52:38.737385 18685 trajectory_fallback_task.cc:161] Use last frame good path to do speed fallback
504 E0724 18:52:38.737418 18685 trajectory_fallback_task.cc:42] Speed fallback due to algorithm failure
505 E0724 18:52:38.737426 18685 fast_stop_trajectory_fallback.cc:30] Fallback using piecewise jerk speed!
506 W0724 18:52:38.737438 18685 fast_stop_trajectory_fallback.cc:33] init_v = -0.0907699, init_a = -0.9082
507 W0724 18:52:38.737452 18685 fast_stop_trajectory_fallback.cc:39] Already stopped! Nothing to do in GenerateFallbackSpeed()
508 I0724 18:52:38.738862 18685 publishable_trajectory.cc:50] IsReversed0
509 I0724 18:52:38.738996 18685 print_debug_info.cc:118] print_ego_box:(-74.372234, 185.347450, -2.337180, 4.700000, 2.060000)
510 I0724 18:52:38.739511 18685 on_lane_planning.cc:463] Planning Perf: planning name [on_lane_planning], 40.9338 ms.
511 I0724 18:52:38.739524 18685 on_lane_planning.cc:465] Planning end frame sequence id = [409]
512 E0724 18:52:38.740571 18685 planning_component.cc:213] xx====-73.4295 yy==186.327 v==0 aa====-1
513 E0724 18:52:38.740602 18685 planning_component.cc:213] xx====-73.4295 yy==186.327 v==0 aa====-1
514 E0724 18:52:38.740609 18685 planning_component.cc:213] xx====-73.4295 yy==186.327 v==0 aa====-1
515 E0724 18:52:38.740623 18685 planning_component.cc:213] xx====-73.4295 yy==186.327 v==0 aa====-1
516 E0724 18:52:38.740617 18685 planning_component.cc:213] xx====-73.4295 yy==186.327 v==0 aa====-1
517 E0724 18:52:38.740623 18685 planning_component.cc:213] xx====-73.4295 yy==186.327 v==0 aa====-1
518 E0724 18:52:38.740630 18685 planning_component.cc:213] xx====-73.4295 yy==186.327 v==0 aa====-1
519 E0724 18:52:38.740638 18685 planning_component.cc:213] xx====-73.4295 yy==186.327 v==0 aa====-1
520 E0724 18:52:38.740644 18685 planning_component.cc:213] xx====-73.4295 yy==186.327 v==0 aa====-1
521 E0724 18:52:38.740653 18685 planning_component.cc:213] xx====-73.4295 yy==186.327 v==0 aa====-1
522 E0724 18:52:38.740659 18685 planning_component.cc:213] xx====-73.4295 yy==186.327 v==0 aa====-1
523 E0724 18:52:38.740666 18685 planning_component.cc:213] xx====-73.4295 yy==186.327 v==0 aa====-1
524 E0724 18:52:38.740674 18685 planning_component.cc:213] xx====-73.4295 yy==186.327 v==0 aa====-1
```

图 1

通过查找原因，发现联合仿真时会进入如下逻辑，然后偶发上述 bug 的现象；单独 apollo pnc 模块（不带 carla）不会进入该逻辑。

报警函数地址：

/modules/planning/planning_base/common/trajectory_stitcher.cc



```
149     const double veh_rel_time =
150         current_timestamp - prev_trajectory->header_time();
151     AERROR << "veh_rel_time: " << veh_rel_time << " vehicle_state: a:" << vehicle_state.angular_velocity;
152     size_t time_matched_index =
153         prev_trajectory->QueryLowerBoundPoint(veh_rel_time);
154     AERROR << "time matched index: " << time_matched_index << " prev_trajectory->StartPoint().relative_time: " << prev_trajectory->StartPoint().relative_time();
155     if (time_matched_index == 0 &&
156         veh_rel_time < prev_trajectory->StartPoint().relative_time()) {
157         AWARN << "current time smaller than the previous trajectory's first time";
158         *replan_reason =
159             "replan for current time smaller than the previous trajectory's first "
160             "time.";
161         return ComputeReinitStitchingTrajectory(planning_cycle_time, vehicle_state);
162     }
163     if (time_matched_index + 1 == prev_trajectory->size()) {
164         AWARN << "current time beyond the previous trajectory's last time";
165         *replan_reason =
166             "replan for current time beyond the previous trajectory's last time";
167         return ComputeReinitStitchingTrajectory(planning_cycle_time, vehicle_state);
168     }
169     auto time_matched_point = prev_trajectory->TrajectoryPointAt(
170         static_cast<uint32_t>(time_matched_index));
171     if (!time_matched_point.has_path_point()) {
```

图 2