详细说明

# 算法说明

## 1.各个包的说明

1.在fastlio\_ws包中，mid360雷达跑fastlio算法，进行去噪和去除自身点云，将/livox/lidar话题（其消息类型为：livox\_laser\_simulation/CustomMsg）输出为/cloud\_registered话题（其消息类型为：sensor\_msgs/PointCloud2）给规划部分建图。

FAST-LIO是一个激光雷达惯性里程计算法，他可以根据激光雷达和IMU的测量数据，实现地图的构建并输出里程计数据。而里程计数据，就是机器人相对出发点的位置姿态及速度数据。在本算法中，使用gazebo真值进行定位，用输出的点云话题给规划部分建图。

2.在catkin\_ctr包中，使用ego-planner算法进行避障规划，px4ctrl进行控制。

Planner包中包括：

Bspline\_opt包是B样条优化包，包括L-Bfgs求解器和均匀B样条和各个代价函数和梯度。

Path\_searching是前端搜索包，包括动态A\*搜索。

Plan\_env包是构建栅格地图的包。

Plan\_manage是最主要的包，包括主节点的创建和无人机各个状态切换的流程和算法的流程。

Traj\_utils包括曲线、点的可视化和初始一条不管碰撞的曲线。

Uav\_simulator包是各种仿真包。

Px4ctrl是仿真中用到的控制器。

整个避障规划算法的流程图如下：

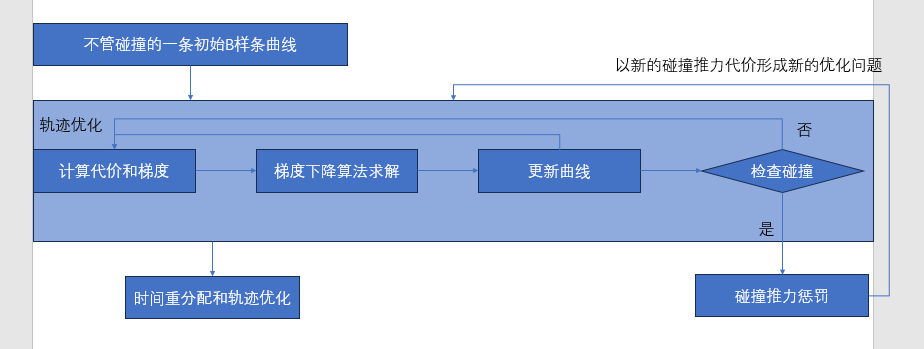


图1：避障规划算法的流程图

## 2.launch文件的说明：

1. roslaunch px4 test\_mid360.launch：

启动gazebo仿真环境，加载无人机和mid360的模型

2. roslaunch fast\_lio mapping\_mid360\_norviz.launch：启动fastlio算法，进行去噪和去除自身点云，将/livox/lidar话题（其消息类型为：livox\_laser\_simulation/CustomMsg）输出为/cloud\_registered话题（其消息类型为：sensor\_msgs/PointCloud2），是全局坐标系下的点云，不是Body坐标系下的点云。其对应的config文件需要修改雷达和imu的话题，

1. roslaunch ego\_planner single\_run\_in\_gazebo\_lidar.launch：

启动ego-planner算法进行避障规划。

1. roslaunch px4ctrl run\_node.launch：

启动px4ctrl控制器进行无人机的控制

1. rosrun rqt\_reconfigure rqt\_reconfigure：

启动rqt工具箱，mode\_bool使得无人机进入offboard模式，cmd\_bool使得无人机触发

1. roslaunch ego\_planner rviz.launch：

启动rviz可视化

## 3.节点和话题

ego\_planner需要输入深度图+相机位姿或是点云，这里以点云为例进行仿真。

1. 话题

/planning/bspline：该话题用于发布来自路径规划器的B样条曲线路径信息。

/traj\_start\_trigger：该话题为px4ctrl触发ego节点的话题。

/planning/pos\_cmd：该话题用于发布机器人的位置控制指令，通过订阅该话题，机器人可以按照指定的位置移动到目标点。

/cloud\_registered：该话题发布经过imu去畸变后的降采样后的点云，已转换到全局坐标系

/livox/lidar：该话题为雷达的点云话题，livox\_ros\_driver/CustomMsg格式。

/mavros/imu/data\_raw：该话题发布无人机的原始imu数据。

/mavros/local\_posion/odom：该话题为里程计话题。

1. 节点

/drone\_0\_ego\_planner\_node：在gazebo环境的无人机负载的雷达中获取点云话题，将接收到的无人机当前位置周围的障碍物深度信息发给drone\_0\_ego\_planner\_node。

/drone\_0\_odom\_visualization：仿真中基于GPS和气压计的定位，Gazebo提供了无人机位姿姿态的真实值，作用类似于实物系统中的动作捕捉系统。位姿真值主要有两个作用，一是对于集群编队等任务，提供了可靠准确的全局定位。从gazebo真值中获得里程计信息odom，将该话题发给drone\_0\_odom\_visualization对无人机进行可视化。

/drone\_0\_traj\_server：drone\_0\_ego\_planner\_node根据障碍物信息和无人机状态使用ego-planner规划处均匀B样条轨迹发给drone\_0\_traj\_server。

/px4\_ctrl：drone\_0\_traj\_server根据接收到的轨迹将控制指令发送给控制器px4ctrl。

其通过发布/traj\_start\_trigger话题给drone\_0\_ego\_planner\_node进行指点飞行。

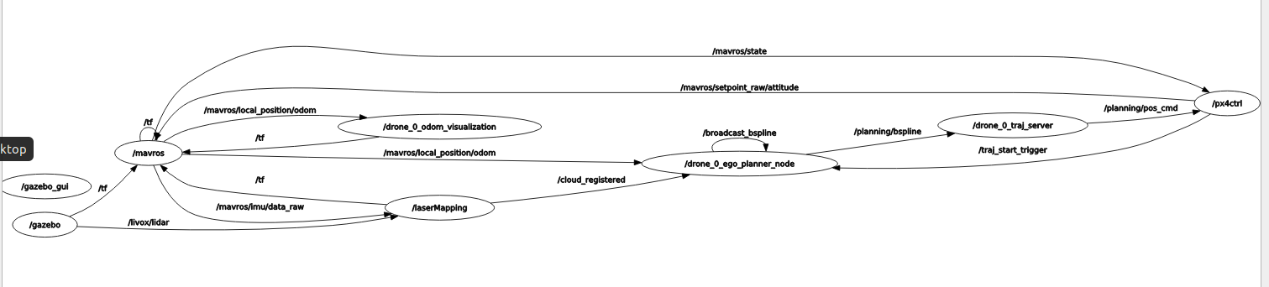


图2：基于mid360仿真的单机避障算法的节点图

# 操作步骤

1. roslaunch px4 test\_mid360.launch
2. 打开qgc
3. cd fastlio\_ws/

source ./devel/setup.bash

roslaunch fast\_lio mapping\_mid360\_norviz.launch

1. cd catkin\_ctr/

source ./devel/setup.bash

cd scripts/

sh ego\_gazebo\_lidar.sh

1. 用qgc控制无人机起飞
2. 针对rqt工具箱，点击mode\_bool，再点击cmd\_bool
3. 打印输出的路径点：

cd catkin\_ctr/

source ./devel/setup.bash

rostopic echo planning/pos\_cmd