# Ego planner 代码流程

## Ego planner node.cpp

  EGOReplanFSM rebo\_replan;

  rebo\_replan.init(nh);

## Ego replan fsm .cpp

### void EGOReplanFSM::init(ros::NodeHandle &nh)  // EGOReplanFSM的初始化函数

exec\_timer\_=nh.createTimer(ros::Duration(0.01),&EGOReplanFSM::execFSMCallback, this);

每0.01秒进入第一个回调函数检查无人机的状态

当状态是**GEN\_NEW\_TRA**J

（1）进入函数**callReboundReplan**

1. 获取局部目标点。
2. 调用 **reboundReplan** 方法进行重规划，这涉及到考虑起始点、速度、加速度和目标点。
3. 根据重规划结果，创建一个B样条消息，包含了规划出的路径的控制点和节点，并将其发布。
4. 如果规划成功，使用可视化工具显示控制点。
5. 返回重规划是否成功的结果。
6. 进入函数**reboundReplan（最重要的函数）**

double time = pow(pp\_.max\_vel\_, 2) / pp\_.max\_acc\_ > dist ?

// 如果可以在不超过最大加速度的情况下达到最大速度

sqrt(dist / pp\_.max\_acc\_) :

// 如果不能在给定距离内达到最大速度，则计算加速到最高速度所需时间，以最高速度行驶一段距离，然后减速到停止所需的总时间

(dist - pow(pp\_.max\_vel\_, 2) / pp\_.max\_acc\_) / pp\_.max\_vel\_ + 2 \* pp\_.max\_vel\_ / pp\_.max\_acc\_;



