

目前自动驾驶应届校招算法岗位主要为以下几种：

- 自动驾驶定位与高精度地图算法工程师；
- 自动驾驶路径规划算法工程师；
- 自动驾驶决策控制工程师；
- 自动驾驶SLAM算法工程师
- ...

一、惯性导航方向

1. IMU的噪声模型是什么？

目前市面中常见的IMU一般都是六轴，即三轴加速度计和三轴陀螺仪，IMU的随机误差一般包括以下几类，各类误差项及其原理如下：

- 量化噪声。量化噪声是传感器的固有误差，在数字传感器通过AD采集把连续时间信号采集成离散信号时产生的误差。量化噪声属于高频噪声，在工程中使用低通滤波或者姿态更新消除掉；
- 零偏。加速度计/陀螺仪角速度输入为零时的输出量，描述陀螺仪的输出信号围绕其均值的起伏或波动；
- 零偏稳定性。当输入角速率为零时，陀螺仪输出量围绕其均值的离散程度，即零偏稳定性表达零偏随时间的波动趋势；
- 角度随机游走。角度随机游走属于高斯白噪声，一般在卡尔曼滤波中作为误差参数进行处理；
- 角速率随机游走。从理解上和角度随机游走一样，角速率里面并不全是白噪声，它也有马尔可夫性质的误差成分，而这个误差是由宽带角加速率白噪声累积的结果；
- 速率斜坡。速率斜坡是一种确定性误差，一般由周边环境变化引起。

2. 惯导误差模型是怎么来的？比如15维的卡尔曼滤波模型。

可参考博客文章 < <https://zhuanlan.zhihu.com/p/135230133> > 或参考英文文献 < Quaternion kinematics for the error-state Kalman filter >

3. 刚体的姿态表示方法

刚体的姿态一般可以用四种方法表示：旋转矢量、旋转矩阵、四元数和欧拉角。在IMU姿态估计中，可以使用旋转矩阵、四元数或欧拉角对角速度进行积分，并设计滤波器。在乘法扩展卡尔曼滤波器中（MEKF），是用四元数进行积分，用旋转矢量表示姿态误差及估计姿态的协方差矩阵。具体的描述内容可以参见高博的《视觉SLAM十四讲》

二、高精度地图方向

1. 常见的高精度地图格式有哪些？

目前最主流的通用格式规范分NDS和OpenDRIVE两种。此外还有日本OMP公司的格式规范。

2. 常用的高精度定位方式有哪些？

包括RTK算法、Lidar匹配、视觉匹配、SINS模块、Mapping match等等。

3. 高精度地图的全局路径规划算法有哪些？

常用的全局路径规划算法有：

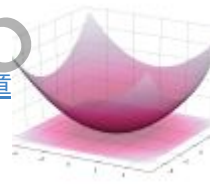
- Dijkstra
- BFS (Best-First-Search)
- A*
- hybrid A*
- D *
- RRT
- RRT*
- 蚁群算法
- Rectangular Symmetry Reduction (RSR)
- BUG
- Beam search
- Iterative Deepeningc
- Dynamic weighting
- Bidirectional search
- Dynamic A* and Lifelong Planning A *
- Jump Point Search
- Theta *

不过目前主流的还是深度优先搜索、广度优先搜索、A*和Dijkstra。

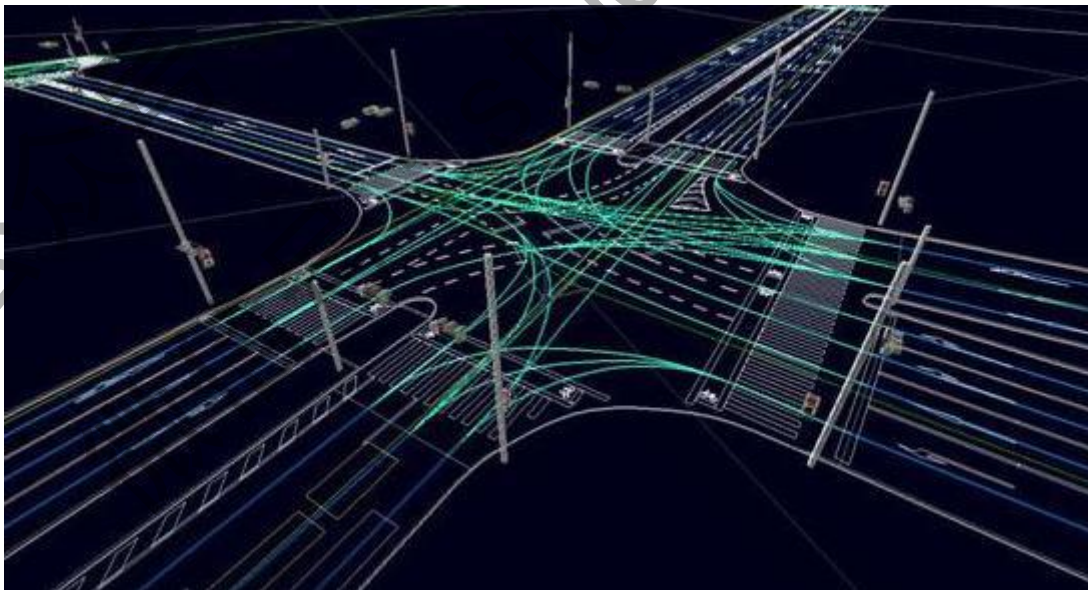
4、高精度地图曲线拟合算法算有哪些？

高精度地图曲线拟合和局部车道线规划如出一辙，一般有：多项式曲线拟合、三次样条插值曲线、贝塞尔曲线拟合、微分平坦、B样条曲线、RANSAC等等，具体可见以下文章：

[【自动驾驶算法】基于差分进化算法的车道线拟合36 赞同 · 10 评论文章](#)



[【自动驾驶-车道线拟合算法】最小二乘法拟合-线性篇43 赞同 · 6 评论文章](#)



[【自动驾驶】高精度地图车道线三次样条插值算法（一）7 赞同 · 1 评论文章](#)

5、如何使用ceres库？

Ceres是一个C++库，可以求解通用的最小二乘问题，使用方法大体上分为如下三步：

1. 定义代价函数；

2. 构建最小二乘问题，向问题中添加误差项，此时会用到第一步的代价函数；
3. 配置求解器，开始求解。

三、路径规划方向

1、路径规划的主要内容有哪些？

路径规划只要分为：**寻路 (Routing)**、**决策 (Behavioral Decision)**、**运动规划 (Motion Planning)**。寻路主要为全局路径规划，现阶段主要为传统导航地图 (SD地图) 和高精度地图结合进行路径规划；决策主要为在遇到交通灯和行人时的避让、高速路分流合流的选择、以及路口时转向判断等等；运动规划是指局部路径规划，判断前方几百米内的期望行驶路径，需结合汽车运动学、动力学、舒适性和无碰撞等要求。

2、是否了解基于人工势场的规划算法？

人工势场的规划算法主要假设车辆行驶在虚拟力场下、车辆的初始点在高势场下，要达到低势场。这种算法使得车辆在势的影响下，避开障碍物达到目标点，常见的相关算法也有很多，重点需要解决的问题就是如何避免车辆先入局部最优解。

3、是否了解RRT算法？

RRT算法主要原理是基于采样的算法，把起点作为一颗种子，向下生长枝桠。像树一样的路径以探索空间的大部分区域，快速的进行扩张，伺机找到目标点，但找到的路径质量不高，通常不是最优路径。

四、c++相关

c++方向除了手撕代码以外，还是会问一些基础知识，具体可见：

https://github.com/xiaoguangcong/cpp_pro/blob/master/C%2B%2B%E5%B8%B8%E8%A7%81%E9%9D%A2%E8%AF%95%E9%A2%98%E7%9B%AE.md

五、Linux 相关

1、linux内核主要分为哪几个系统？

进程管理系统、内存管理系统、I/O管理系统和文件管理系统

2、vi有哪两种模式？

命令模式和输入模式

3、进程的查看和调度命令

进程查看ps和top

进程调度 kill、at、batch

4、Linux属于哪种类型的操作系统

多用户、多任务、多进程

5、如何查看当前进程？如何退出？

ps; exit

6、ls命令执行什么功能？有哪些参数,有什么区别？

ls命令显示当前路径下的所有文件和文件夹

参考：-a 全部文件

-d 只显示目录

-l 详细信息

7、创建目录什么命令？创建文件？复制文件？

mkdir创建文件夹；vi, vim, cat, touch创建文件；cp复制文件

8、文件修改权限命令？

chmod 权限属性 文件或者目录

其中，ubuntu文件权限的设置主要为以下方法：

-rw----- (600) 只有所有者才有读和写的权限

-rw-r--r-- (644) 只有所有者才有读和写的权限，组群和其他人只有读的权限

-rwx----- (700) 只有所有者才有读，写，执行的权限

-rwxr-xr-x (755) 只有所有者才有读，写，执行的权限，组群和其他人只有读和执行的权限

-rwx--x--x (711) 只有所有者才有读，写，执行的权限，组群和其他人只有执行的权限

-rw-rw-rw- (666) 每个人都有读写的权限

-rwxrwxrwx (777) 每个人都有读写和执行的权限

其中：

- 代表的是类型 第2-4位代表u所有者 (user) 第5-7位代表的是g组群 (group) 第8-10位代表的o是其他人 (other) 另外，a代表所有人 (all) r表示文件可以被读 (read) w表示文件可以被写 (write) x表示文件可以被执行 (如果它是程序的话) - 表示相应的权限还没有被授予 - 其中：rwx也可以用数字来代替 r-----4 w-----2 x-----1 -----0

在终端输入：chmod o+w <http://xxx.xxx> 表示给其他人授予写<http://xxx.xxx>这个文件的权限 chmod go-rw <http://xxx.xxx> 表示删除<http://xxx.xxx>中组群和其他人的读和写的权限 行动：+ 表示添加权限 - 表示删除权限 = 表示使之成为唯一的权限

9、删除文件用哪个命令？如果需要连目录及目录下文件一块删除呢？删除空文件夹用什么命令？

rm; 加-r递归删除; rmdir

10、使用什么命令查看网络连通性？

ifconfig

六、ROS相关

1、ROS中roslaunch与roslaunch的区别

官网中：

roslaunch allows you to run an executable in an arbitrary package without having to cd (or roscd) there first.

roslaunch is a tool for easily launching multiple ROS nodes locally and remotely via SSH, as well

as setting parameters on the Parameter Server. It includes options to automatically respawn processes that have already died. roslaunch takes in one or more XML configuration files (with the .launch extension) that specify the parameters to set and nodes to launch, as well as the machines that they should be run on.

翻译成中文就是：

roswun 允许您在任意软件包中运行可执行文件，而无需先在其中进行cd（或roscd）。

roslaunch 是一种工具，可通过SSH在本地和远程轻松启动多个ROS节点，以及在参数服务器上设置参数。它包括自动重生已经死掉的进程的选项。roslaunch接收一个或多个XML配置文件（带有.launch扩展名），这些文件指定要设置的参数和要启动的节点以及应在其上运行的计算机。

可以看出：**roswun只能运行一个node，roslaunch可以同时运行多个nodes.**

roslaunch可以用来启动定义在launch文件中的多个节点，通常的命名方案是以.launch作为启动文件的后缀，启动文件是xml文件，一般把启动文件存储在取名为launch的目录中，用法

```
$ roslaunch [package] [filename.launch]
```

2、ROS常用命令有哪些？

```
#最常用
roscore                                #启动rosmaster
roswun pkg_name node_name              #启动ros节点
roslaunch pkg_name launch_files_name  #启动launch文件
catkin_make                             #编译工作空间
rospack profile                         #刷新功能包路径

#环境变量
echo $ROS_PACKAGE_PATH                 #打印ros环境变量
export | grep ROS                      #确认环境变量已经设置正确
source devel/setup.bash                #刷新环境变量
echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc  #刷新环境变量，永久有效
source ~/.bashrc                      #生效上一句

#功能包
catkin_create_pkg test_package std_msgs roscpp rospy  #创建名字为
test_package的功能包，添加std_msgs roscpp rospy依赖
rospack list                             #查看软件包列表
rospack find package-name               #定位软件包
roscd package-name                     #切换到指定功能包目录

#话题
rostopic list                           #输出当前运行的topic列表
rostopic info topic-name                #查看话题信息
rostopic echo topic-name                #输出话题数据
rostopic hz topic-name                  #每秒发布的消息数量
rostopic bw topic-name                  #每秒发布信息所占的字节量

#工具
rviz                                    #启动rviz
rqt_graph                               #可视化节点关系
rqt_plot                                 #可视化话题数据
roswun rqt_tf_tree rqt_tf_tree          #查看tf树

#数据记录与播放
roswag record -a                        #录制所有topic到bag文件
roswag play 包名字                     #回放bag包数据
```

```
rosv bag filter 1.bag 2.bag "topic == '/scan'" #将1.bag文件中scan话题过滤到2.bag中
```

3、ROS中常用的开发工具

- launch文件，可以同时启动多个ros节点，可自动启动rosmaster。
- TF坐标变换，用来描述机器人系统中繁杂的坐标系。
- QT工具箱，用来可视化各个ros节点之间的订阅或者请求关系或者可视化各种话题的数据。常用命令，rqt_graph、rqt_plot，而且还可以动态修改部分参数。
- rviz可视化平台，机器人可视化平台，几乎是最常用的工具。使用rviz命令启动
- Gazebo物理仿真环境，可对机器人进行三维物理仿真。需要仿真的同学常用。

4、如何新建一个工作空间？

```
cd ~/catkin_test_ws/src #进入存放功能包的地方
catkin_create_pkg test std_msgs roscpp rospy #创建名字为test的功能包，添加std_msgs roscpp rospy依赖
cd ~/catkin_test_ws
catkin_make #编译
source devel/setup.bash #导入功能包刷新环境变量，才能找到功能包
rospack profile
```

二次更新：

七、手撕算法题：

- 1、树的前序、后序、深度、广度遍历
- 2、手写一个迪杰斯特拉算法并分析复杂度并优化；
- 3、两个双向循环链表 A B，删除两个链表中相同的节点；
- 4、最长公共子序列长度；
- 5、手写反转链表；
- 6、最长连续子串；

八、自动驾驶代码分析题：

- 1、泊车模块计算涉及到部分内联函数定义：

```
inline float calcProjLen(const Point_T *pa, const Point_T *pb, const Point_T
*uv) {
    float lx = pb->x - pa->x;
    float ly = pb->y - pa->y;
    float dp = lx * uv->x + ly * uv->y;
    return fabsf(dp);
}
```

请问使用inline函数的注意事项，优点跟缺点分别是什么：

2、泊车模块计算涉及结构体的定义，在32位操作系统里面，以下的结构体定义哪个更好，为什么？

第一种：

```
typedef struct _Udata_Info_T1{
    int num;
    short peak_2nd;
    char peak;
}Udata_Info_T1;
```

第二种：

```
typedef struct _Udata_Info_T2{
    char peak;
    int num;
    short peak_2nd;
}Udata_Info_T2;
```

3、已知识别的图像里面，单位像素代表实际的长度2cm，角点定位的标准差为 $\sigma = 3$ pixel，某个被识别的车位长度为280像素，宽度为120像素，假设长度和宽度的测量是完全独立的，求该车位实际长度、宽度、面积，以及三者的标准差。

九、SLAM相关

1、提取矩阵左上角3x3块提取出来，然后赋值为 $I_{3 \times 3}$ 。编程实现：

```
#include<iostream>
#include<Eigen/Core>
#include<Eigen/Geometry>

#define MATRIX_SIZE 30
using namespace std;

int main(int argc, char **argv)
{
    Eigen::Matrix<double, MATRIX_SIZE, MATRIX_SIZE> matrix_NN =
    Eigen::MatrixX<double>::Random(MATRIX_SIZE, MATRIX_SIZE);
    Eigen::Matrix<double, 3, 3> matrix_3d1 = Eigen::MatrixX<double>::Random(3, 3);
    Eigen::Matrix3d matrix_3d = Eigen::Matrix3d::Random();
    /*方法1: 循环遍历矩阵的三行三列 */
    for(int i = 0; i < 3; i++){
        for(int j = 0; j < 3; j++){
            matrix_3d(i, j) = matrix_NN(i, j);
            cout<<matrix_NN(i, j)<<" ";
        }
        cout<<endl;
    }
    matrix_3d = Eigen::Matrix3d::Identity();

    /*方法2: 用.block函数 */
    /*
    matrix_3d = matrix_NN.block(0, 0, 3, 3);
    matrix_3d = Eigen::Matrix3d::Identity();
    */
    return 0;
}
```

2、线性方程 $Ax = b$ 的解法。

详见：

[灰色的石头 - 博客园](#)

3、手写曲线拟合的任一算法（最小二乘法、三次样条曲线、Ransac）

4、是否了解Pnp算法与ICP算法，简述其原理；

5、是否了解常见的光流算法；

6、常见的SLAM主流框架，包括激光和视觉；

SLAM即实时定位建图，按照使用的传感器分为激光SLAM（LOAM、V-LOAM、cartographer、gmapping）与视觉SLAM，其中视觉SLAM又可分为单目SLAM（MonoSLAM、PTAM、DTAM、LSD-SLAM、ORB-SLAM（单目为主）、SVO）、双目SLAM（LIBVISO2、S-PTAM等）、RGBD SLAM（KinectFusion、ElasticFusion、Kintinuous、RGBD SLAM2、RTAB SLAM）；按照前端方法分为特征点法（稀疏法）、光流法、稀疏直接法、半稠密法、稠密法；按照后端方法分为基于滤波（EKF，UKF，PF原理简介）与基于图优化的方法。

7、LSLAM和VSLAM各自的优缺点

8、八叉树地图的优劣势

9、视觉SLAM常用的相机包括，单目，双目，RGB-D相机，请分别说说它们的优缺点

10、是否使用过李代数，李群与李代数的关系？

1、旋转矩阵自身是带有约束的，即是正交矩阵且行列式为1，他们作为优化变量时，会引入额外的约束，优化变的困难，通过李群李代数的转换关系，把位姿估计变成无约束的优化问题。

2、李群与李代数是指数与指数的关系，李群到李代数是指数变换，李代数到李群是指数变换。李群是矩阵的集合，李代数是向量的集合，矩阵求导时对加法不封闭，而映射为李代数之后对向量来说，加法是封闭的。

11、SLAM中的KF, EKF, UKF, PF原理简介

12、如何标定IMU与相机之间的外参数？