**第十九届全国大学生智能汽车竞赛**

**室外无人驾驶自行车挑战赛**

**技 术 手 册**

学 校：XXXX学院

队伍名称：XX

参赛队员：XX、XX、XX

带队教师：XX、XX

关于技术手册使用授权的说明

本人完全了解全国大学生智能汽车竞赛关于保留、使用技术手册和研究论文的规定，即：参赛技术手册内容著作权归参赛者本人，比赛组委会和赞助公司可以在相关主页上收录并公开参赛作品的设计方案、技术手册、源码以及参赛模型车的视频、图像资料，并将相关内容编纂收录在组委会出版论文集中。

参赛队员签名：

带队教师签名：

日 期： 2024.\*\*.\*\*

（本页需要签字后，替换为扫描页）

**技术手册撰写注意事项：**

了解到实验室队伍老生带新生容易断带、知识不能很好的积累沉淀转化等痛点，本次技术手册的内容更加聚焦于技术总结，知识整理。力图将技术手册打造为老队员、新队员、指导老师之间共用的知识纽带。通过几年的打磨，便于带队老师开设自动化控制、自动驾驶、人工智能、软件编程等相关课程以及开展新队员的筛选工作，便于新队员更快的传承已有的知识储备，使得实验室队伍建设构成闭环。

技术手册内容主要是将自行车作为硬件载体，以自动化控制、自动驾驶、人工智能、软件编程等专业为导向，构建一套用于学习的实验指导书。书写的内容结合车模配套的指导手册、视频资料以及比赛规则。

技术手册书写注意事项如下：

1. 提交形式为**实验指导书+源码**；

(2) 所有内容，需要以自行车作为唯一硬件载体；

(3) 提供不少于3个实验，以实验指导书+源码的形式提交；

(4) 裁判将结合实验难易程度、书写清晰度、实验个数、实验间是否有关联性等多种因素，决定本技术手册的优劣。

技术手册总分10分，评判标准如下：

(1)每个实验最高分5分，总体不少于2个实验；

(2)每个实验得分根据实验难易程度、书写清晰度、实验间是否有关联性、创新性等因素进行判定；

(3)若书写多于2个实验，逐个实验判分后，以分数最高的2个实验的成绩加和作为最终成绩。

### 目录（供参考）

实验一 单片机设计原理 2

实验二 供电系统设置 4

实验三 电机调节步骤及参数解析 6

实验四 PID算法调试 8

实验五 激光雷达数据结构分析及滤波 10

[实验六 视觉功能调试](#_bookmark0) 12

[实验七 opencv数据处理](#_bookmark0) 14

[实验八 机器人运动学平台](#_bookmark0) 16

... ...

**(以上目录仅供参考，具体实验名称及实验内容由学生自主设定)**

实验案例书写具体要求如下：

实验案例需言简意赅，便于后续师弟师妹消化传承；

实验案例撰写可自行创新发挥，新颖的案例更加容易获得高分；

每个实验需要包含实验目的、实验内容、实验器材、实验原理、实验步骤这几部分；

实验目的：参考模板，提炼若干条

实验内容：针对实验目的，开展的具体操作总结

实验器材：罗列实验中用到的器材设备、传感器、工具、测量仪器等

实验原理：描述实验中硬件实现原理、物理相关基础、算法实现原理等

实验步骤：实验开展过程中具体的运行指令、操作方式、测试方式、代码讲解等

### **实验五 激光雷达数据结构分析及滤波(参考样例)**

**实验目的：**

1、熟悉激光雷达数据结构；

2、了解激光雷达数据的处理方法；

3、熟练掌握laser\_filters功能包的使用；

**实验内容：**

1、获取并分析激光雷达数据；

2、对激光雷达数据进行滤波处理。

**实验仪器：**

ROS智能车、激光雷达、电压表

**实验原理：**

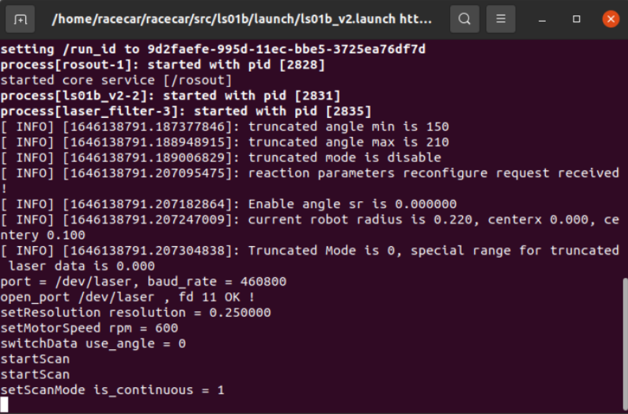
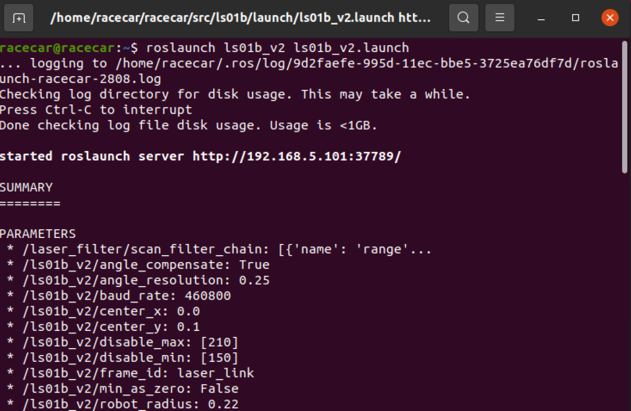
利用laser\_filters功能包，对雷达数据进行处理，得到需要的滤波后的雷达数据。

**实验步骤：**

**<1> 启动雷达，并获取雷达数据(需要先编译我的功能包，并加入环境变量)**

$ roslaunch ls0lb\_v2 ls01b\_v2.launch

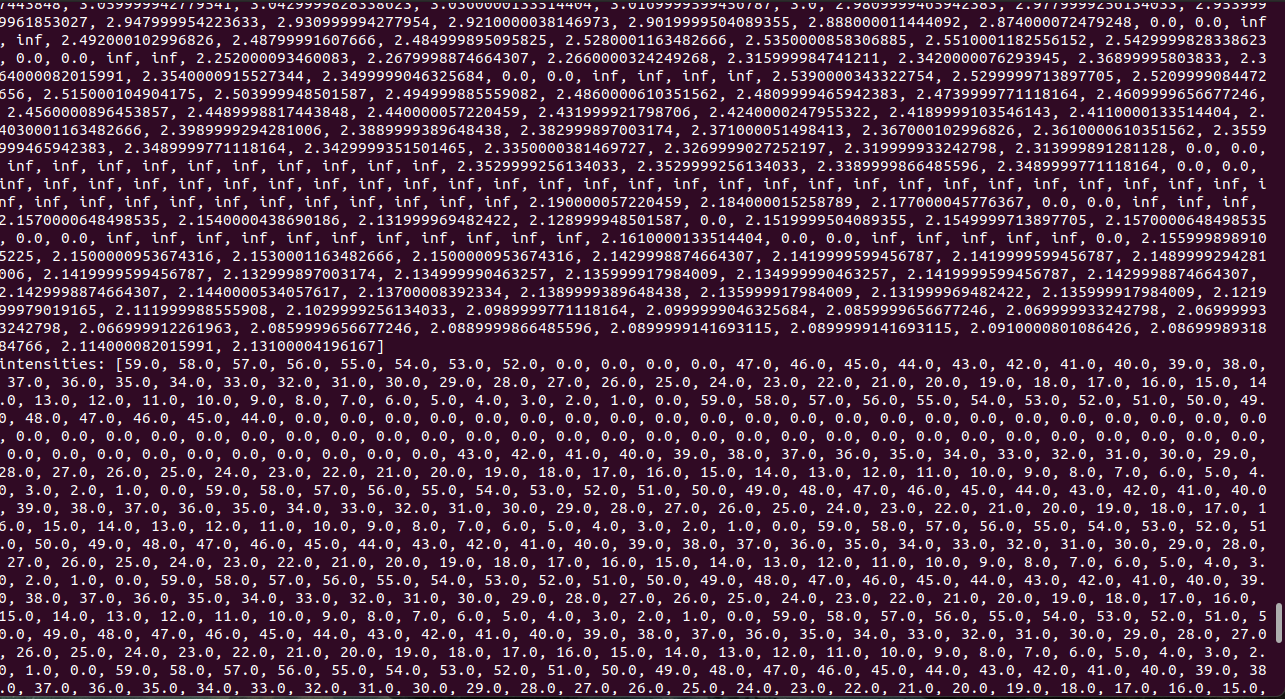
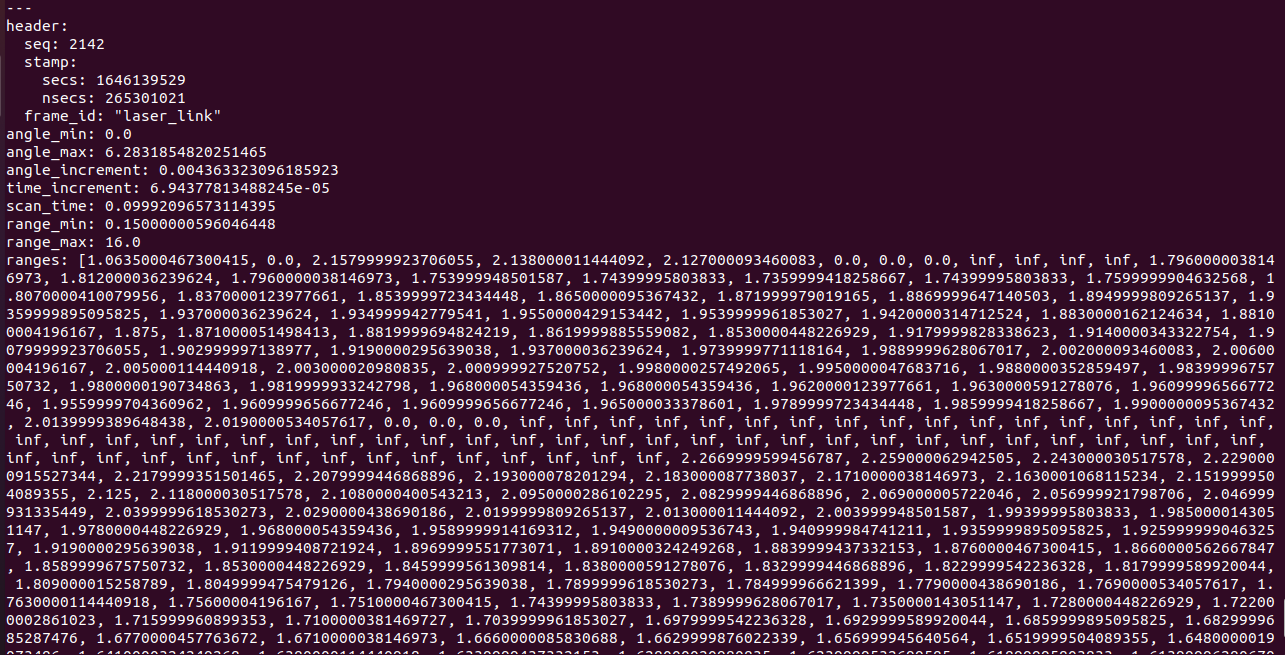
启动后的效果如下图所示，这个时候就启动了雷达节点和滤波节点，具体为什么会启动，已经如何配置launch文件里那些参数，后面会讲到。首先我们先来看一下雷达的原始数据，分析一下雷达数据结构。



**<2> 打印雷达数据消息，并分析雷达数据(新开一个终端输入下面命令)**

$ rostopic echo /scan

运行后会如下图所示，不断打印雷达数据。



下面我们来看一下雷达数据结构：

激光雷达是避障与SLAM常用的装备，ROS中也对激光雷达作出了相应的支持，各个厂家也有不少提供了自己的ROS库供开发者使用，使得无论是初学者还是大神们都能轻松愉快的使用激光雷达为自己的机器人导航。

ROS中的sensor\_msgs/LaserScan提供了最基础的针对一线激光雷达消息格式：

# 一线激光雷达的一次扫描

Header header # timestamp是扫描中第一束激光的获取时间

# 角度环绕Z轴（Z轴为激光雷达正上方），逆时钟方向增加，0度角指向X轴

float32 angle\_min # 扫描的起始角度 [rad]

float32 angle\_max # 扫描的结束角度 [rad]

float32 angle\_increment # 每次测量间的角度差 [rad]

float32 time\_increment # 每次测量间的时间差 [seconds]

# 这个数值会用来生成点云（而不是scan\_time）

float32 scan\_time # 每次扫描的时间差 [seconds]

# 这个时间差是每次激光发射的时间差，上面那个是每次接收（测量）的时间差

float32 range\_min # 距离最小值 [m]

float32 range\_max # 距离最大值 [m]

float32[] ranges # 距离数据 [m]

float32[] intensities # 强度数据 [单位视设备标定]

#如果设备不产生强度数据，则该数据为空

对应前面的雷达数据，我们可以将sensor\_msgs/LaserScan消息和智能车雷达数据一一对应起来，我重点讲的是range数据，因为后面实验三会用到。

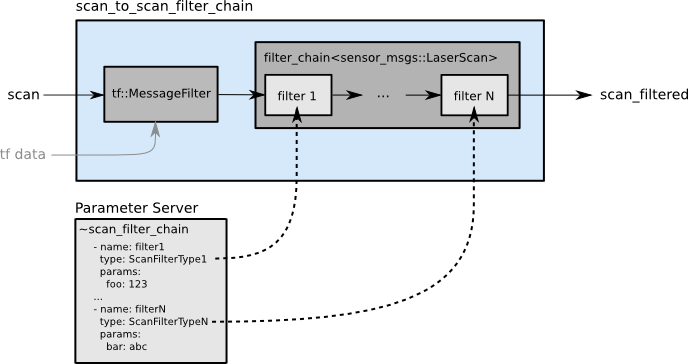
range是数组，数组的长度越长，说明激光雷达的精度越高，咱们的智能车的range数组长度是1140，也就是一周每一度都有4份距离信息。

range数组是如何和角度对应的呢？range[0]对应的角度就是激光雷达的零点方向（一般为激光雷达的正前方），后面数据rangep[1]、range[2]……依次为沿激光雷达逆时针方向对应角度的距离数据。

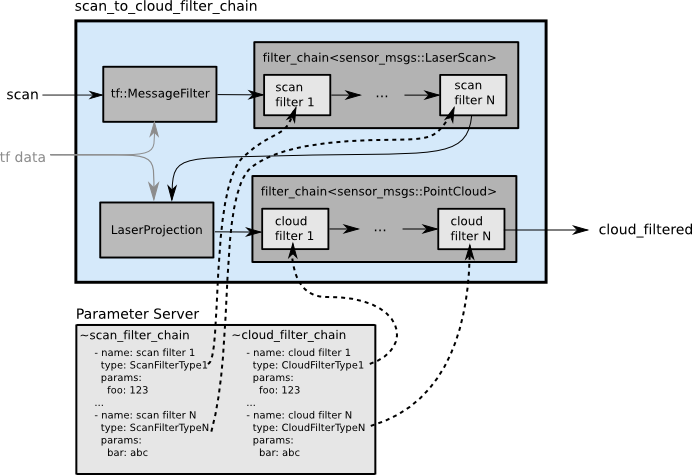
**<3> 使用laser\_filters对雷达数据进行滤波**

$ [laser\_filters的wiki页](http://wiki.ros.org/laser_filters)

http://wiki.ros.org/laser\_filters

在yaml文件里面定义了很多laser\_filters消息的参数后，通过启动laser\_filters节点完成激光雷达消息的过滤。其中scan\_to\_scan\_filter\_chain提供了从初始扫描到过滤后扫描的流程：

而scan\_to\_cloud\_filter\_chain提供了从scan消息生成过滤后的点云的流程：



这两类的示例yaml和launch都可以在上面提供的wiki页找到，很有参考价值的（基本上拷下来调调参数就能用了）设置参数放在yaml里，消息转换和参数加载放在launch里。

**<4> laser\_filters常用参数类型：**

**1.InterpolationFilter**

将扫描中出现的异常数据通过使用两侧的正常数据进行补齐，效果如下：

.....`..... => InterpolationFilter => ...........

参数设置示例：

scan\_filter\_chain:

- name: interpolation

type: InterpolationFilter

**2.LaserScanIntensityFilter**

强度值过滤器，将超出范围的扫描结果中的距离结果设为最大距离+1，从而将结果标记为异常。这思路有点迷，参数有三个：

lower\_threshold (double) 最小阈值

upper\_threshold (double) 最大阈值

disp\_histogram (int) 是否输出柱状图(°Д°)居然有这功能，牛逼

参数设置示例：

scan\_filter\_chain:

- name: intensity

type: LaserScanIntensityFilter

params:

lower\_threshold: 8000

upper\_threshold: 100000

disp\_histogram: 0

1. **LaserScanRangeFilter：**

距离值过滤器，会把超过设定范围阈值的数据改为NaN或根据设定进行更改。另外可以读取scan消息中的范围阈值信息。

lower\_threshold (double) 最小阈值

upper\_threshold (double) 最大阈值

use\_message\_range\_limits (bool) 是否使用scan消息中的阈值信息，默认为false

lower\_replacement\_value (double) 将小于最小阈值的数据改为该值，默认NaN

upper\_replacement\_value (double) 将大于最大阈值的数据改为该值，默认NaN

参数设置示例：

scan\_filter\_chain:

- name: range

type: LaserScanRangeFilter

params:

use\_message\_range\_limits: false

lower\_threshold: 0.3

upper\_threshold: .inf

lower\_replacement\_value: -.inf

upper\_replacement\_value: .inf

1. **LaserScanAngularBoundsFilter**

将设定的角度外的扫描数据删除。

lower\_angle (double) 最小角度（rad）

upper\_angle (double) 最大角度（rad）

参数设置示例：

scan\_filter\_chain:

- name: angle

type: LaserScanAngularBoundsFilter

params:

lower\_angle: -1.57

upper\_angle: 1.57

1. **其他参数**

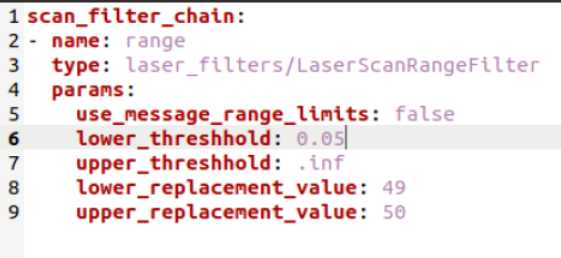
LaserScanBoxFilter 无视一个区块内的数据（常用于无视机器人本体对激光雷达数据的干扰）

LaserScanAngularBoundsFilterInPlace 不会删除目标角度扇区外的数据，但会把对应扫描的距离值设为最大距离阈值+1

ScanShadowsFilter 针对物体边沿的扫描和识别

LaserArrayFilter 使用数学过滤器对距离和强度进行过滤

**<5> 该方案采用的滤波参数详解：**



本方案采用的laser\_filters滤波参数如上图所示。最小阈值设置为了0.05，最大阈值设置为了.inf。也就是小于等于0.05的距离会变成49，大于等于.inf的会变成50。

下面我们再来看一下launch文件是如何编写的。



如图所示，雷达滤波功能包订阅的话题名称是/scan话题，对/scan话题数据滤波处理后发布的话题是/scan\_filtered，launch文件的第15行将话题名/scan\_filtered重映射为/scan\_1。

输入如下命令，查看滤波前的雷达数据/scan和滤波后的雷达数据/scan\_q

$ roslaunch ls0lb\_v2 ls01b\_v2.launch (终端1)

$ rostopic echo /scan (终端2)

$ rostopic echo /scan\_1 (终端3)

\_

这个时候对比观察可以看出，/scan\_1的雷达数据是按照我们设置滤波后的数据。

为什么我们要把小于等于0.05的数据变成49，把大于等于.inf的数据变成50呢？下面的实验三会详细讲到，其目的就是为了方便后面利用雷达数据定位到锥桶位置。