

深蓝学院激光 SLAM 第三期第二次作业

一·本次作业练习目标

巩固课程所学知识，通过代码实践理解基于直接线性法和基于模型的标定方法，加深对最小二乘问题求解的数学理解。

二·作业计分原则

满分为 12 分，10 分为优秀，8 分为良好，6 分为及格。

三·作业提交说明

需提供完整的 PDF 报告及代码。公式推导题可以手写照相粘进 PDF，也可以直接写进 PDF。

四·作业题目说明

1. 补充直接线性方法的里程计标定模块代码；（6 分）
2. 补充基于模型方法的里程计标定模块代码；（2 分）
3. 通过互联网总结学习线性方程组 $Ax=b$ 的求解方法，回答以下问题：（2 分）
 - （1）对于该类问题，你都知道哪几种求解方法？
 - （2）各方法的优缺点有哪些？分别在什么条件下较常被使用？

4. 简答题，开放性答案：设计里程计与激光雷达外参标定方法。（2 分）

我们一般把传感器内自身要调节的参数称为内参，比如前面作业中里程计模型的两轮间距与两个轮子的半径。把传感器之间的信息称为外参，比如里程计与激光雷达之间的时间延迟，位姿变换等。请你选用直接线性方法或基于模型的方法，设计一套激光雷达与里程计外参的标定方法，并回答以下问题：

- （1）你设计的方法是否存在某些假设？基于这些假设下的标定观测值和预测值分别是什么？
- （2）如何构建你的最小二乘方程组求解该外参？

五·作业提示与学习材料

第一题代码说明：

1. 本次的作业为用直接线性方法来对机器人的里程计进行校正。
2. 给出的文件中包含有本次作业使用的 bag 数据，路径为 odom_ws/bag/odom.bag。
3. 本次的作业中，需要实现三个函数，分别为：
 1. Main.cpp，第 358 行中的 cal_delta_distance() 函数，该函数的功能为给定两个里程计位姿，计算这两个位姿之间的位姿差。
 2. Odom_Calib.cpp，第 28 行 Add_Data() 函数，该函数的功能为构建超定方程组 $Ax=b$ ，具体参考 PPT。
 3. Odom_Calib.cpp，第 48 行 Solve() 函数，该函数的功能为对 2 中构建的超定方程组进行求

解。

本次程序的运行过程为：

Step1: 实现上述的三个函数，在 odom_ws 下用 catkin_make 进行编译。

提示：如果编译过程中提示缺少 csm 库，可以用命令 `sudo apt-get install ros-kinetic-csm` 进行安装。如果你用的 ROS 版本为其它版本，则将 CMakeLists.txt 中所有 kinetic 内容改为你的 ROS 版本。其它库安装方式也类似，比如缺少 nav_core，则用 `sudo apt-get install ros-kinetic-nav-core` 安装。

Step2: 在 odom_ws 下，进行 source，具体命令为：source devel/setup.bash

Step3: 运行 launch 文件：roslaunch calib_odom odomCalib.launch。执行本条指令的时候，必须保证没有任何 ros 节点在运行，roscore 也要关闭。

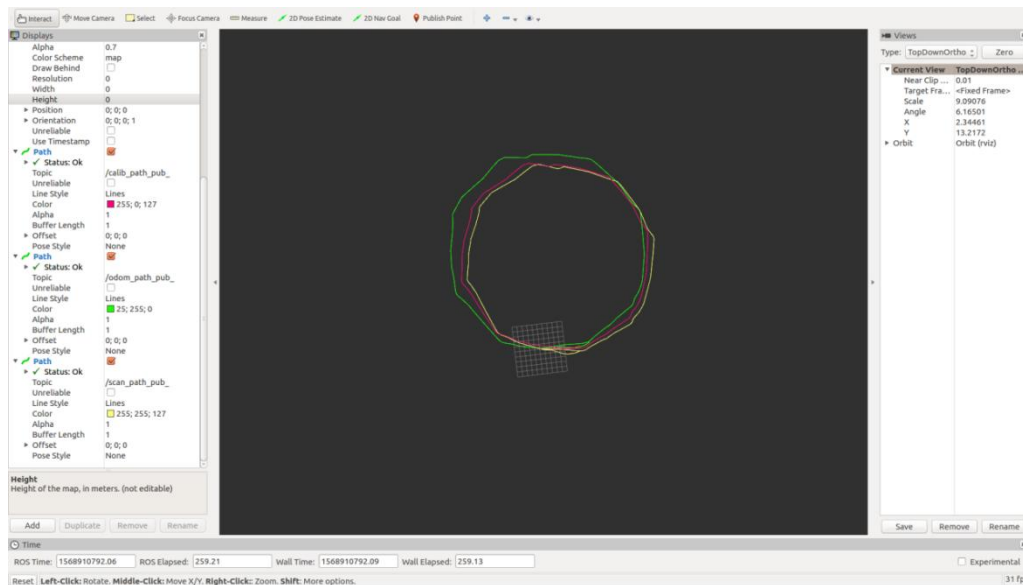
Step4: 在 3 正常的情况下，运行 rviz，fix_frame 选择为 odom。在 Add 选项卡中增加三条 Path 消息。一条订阅的 topic 为：odom_path_pub_；一条订阅的 topic 为：scan_path_pub_；最后一条为：calib_path_pub_。分别选择不同的颜色。

Step5: 进入到 odom_ws/bag 目录下，运行指令：rosbag play --clock odom.bag。

Step6: 如果一切正常，则能看到运行矫正程序的终端会打印数据，并且 rviz 中可以看到两条路径。当打印的数据到达一个的数量之后，则可以开始矫正。

Step7: 矫正的命令为，在 calib_flag 的 topic 下发布一个数据：rostopic pub /calib_flag std_msgs/Empty "{}"。

Step8: 程序矫正完毕会输出对应的矫正矩阵，并且会在 rviz 中显示出第三条路径，即 calib_path。可以观察里程计路径 odom_path 和矫正路径 calib_path 的区别来判断此次矫正的效果。



```
Data Cnt:2500
Data Cnt:2501
Data Cnt:2502
Data Cnt:2503
Data Cnt:2504
Data Cnt:2505
Data Cnt:2506
Data Cnt:2507
Data Cnt:2508
Data Cnt:2509
Data Cnt:2510
Data Cnt:2511
Data Cnt:2512
Data Cnt:2513
Data Cnt:2514
Data Cnt:2515
correct_matrix:
  0.936074    2.08139   -0.0483652
 -0.0128072   6.91896    0.149832
  0.0014373  23.0442     0.39099
calibration over!!!!
```

参考结果：绿色为 odom，黄色为激光，红色为矫正后的轨迹，可以看到校正后的轨迹跟激光的轨迹接近了很多。

说明：以上结果为数据包开始到 179s 的 2500 个数据计算出来的，采集不同量数据得到的结果会有所不同，也并非采集时间越长结果越好，标定结果数据仅供参考，效果直接看轨迹图。

第二题代码说明：

参考 PPT，已知两轮角速度与激光雷达的匹配值，填写 TODO 部分的代码，计算轮间距与两轮半径。

如果对激光雷达和里程计内外参数标定有更多的兴趣，可以阅读论文 Simultaneous calibration of odometry and sensor parameters for mobile robots. 与该论文的开源代码实现

<https://github.com/AndreaCensi/calibration>.

对于线性方程 $Ax = b$ 的求解，Eigen 库[1]提供了多种求解方法：

Decomposition	Method	Requirements on the matrix	Speed (small-to-medium)	Speed (large)	Accuracy
PartialPivLU	partialPivLu()	Invertible	++	++	+
FullPivLU	fullPivLu()	None	-	--	+++
HouseholderQR	householderQr()	None	++	++	+
ColPivHouseholderQR	colPivHouseholderQr()	None	+	-	+++
FullPivHouseholderQR	fullPivHouseholderQr()	None	-	--	+++
CompleteOrthogonalDecomposition	completeOrthogonalDecomposition()	None	+	-	+++
LLT	llt()	Positive definite	+++	+++	+
LDLT	ldlt()	Positive or negative semidefinite	+++	+	++
BDCSVD	bdcSvd()	None	-	-	+++
JacobiSVD	jacobiSvd()	None	-	---	+++

具体可参见

http://eigen.tuxfamily.org/dox/group__TopicLinearAlgebraDecompositions.html

<https://www.cnblogs.com/wangxiaoyong/p/8977343.html>

[1] http://eigen.tuxfamily.org/index.php?title=Main_Page