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：
4. Main.cpp，第340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。
5. Odom\_Calib.cpp，第23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。
6. Odom\_Calib.cpp，第44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：

1. 实现上述的三个函数，并且进行编译。
2. 在odom\_ws下，进行source：source devel/setup.bash
3. 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。
4. 在3正常的情况下，运行rviz，fix\_frame选择为odom\_frame。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。
5. 进入到odom\_ws/bag目录下，运行指令：rosbag play –clock odom.bag。
6. 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。
7. 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 “{}”。
8. 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区别来判断此次矫正的效果。