

西北工业大学队伍的自主巡检方案

方案使用雷达-IMU数据融合的激光惯性里程计，加入 SC-A-LOAM 实现回环闭合和位姿图优化，改进自适应轨迹规划技术作为规划器，最终设计基于INDI的内环姿态控制+基于TinyMPC的外环位置/速度控制的控制器

实验配置

- 环境：ROS1 Noetic GAZEBO
- 激光雷达：Mid360
- IMU：设计的高频率IMU模块
- 驱动包：Livox SDK2

算法环境配置

```
https://github.com/Yuxin-Wei/TZB_ws.git# Ceres 2.1.0
sudo apt-get -y install liblapack-dev libsuitesparse-dev libcxsparse3
libgflags-dev libgoogle-glog-dev libgtest-dev
wget -O ceres-solver.zip https://github.com/ceres-solver/ceres-
solver/archive/refs/tags/2.1.0.zip
unzip -q ceres-solver.zip -d "${TRDPARTY_DIR}"
pushd "${TRDPARTY_DIR}/ceres-solver-2.1.0"
mkdir build
cd build
cmake -DBUILD_SHARED_LIBS=TRUE ..
make -j8
sudo make install

# Eigen 3.3.7
wget -O eigen3.zip
<https://gitlab.com/libeigen/eigen/-/archive/3.3.7/eigen-3.3.7.zip>
unzip -q eigen3.zip -d "${TRDPARTY_DIR}"
pushd "${TRDPARTY_DIR}/eigen-3.3.7"
mkdir build
cd build
cmake -DBUILD_SHARED_LIBS=TRUE ..
sudo make install
sudo ln -s /usr/include/eigen3/Eigen /usr/include/Eigen

# GTSAM
cd ~
git clone https://github.com/borglab/gtsam.git
mkdir build && cd build
cmake -D GTSAM_USE_SYSTEM_EIGEN=ON ..
make
sudo make install

# Livox SDK2
git clone https://github.com/Livox-SDK/Livox-SDK2.git
```

```
cd Livox-SDK2
mkdir build
cd build && cmake ..
make
sudo make install

# 下载 & 编译项目
git clone https://github.com/Yuxin-Wei/TZB_ws.git

cd TZB_ws
catkin_make
```

仿真环境配置

仿真环境基于XTdrone的 PX4 1.13 版本进行仿真环境的搭建。配置好Xtdrone环境后将model和world文件导入~/PX4_Firmware/Tools/sitl_gazebo/models与~/PX4_Firmware/Tools/sitl_gazebo/worlds

```
# 配置雷达环境
cd ~/catkin_ws/src
git clone https://github.com/fratopa/Mid360_simulation_plugin.git

cd ~/catkin_ws
source opt/ros/noetic/setup.bash
catkin build -DCMAKE_BUILD_TYPE=Release

# 在~/.bashrc 中最底部添加

source ~/catkin_ws/devel/setup.bash
source ~/PX4_Firmware/Tools/setup_gazebo.bash ~/PX4_Firmware/
~/PX4_Firmware/build/px4_sitl_default
export ROS_PACKAGE_PATH=$ROS_PACKAGE_PATH:~/PX4_Firmware
export ROS_PACKAGE_PATH=$ROS_PACKAGE_PATH:~/PX4_Firmware/Tools/sitl_gazebo
export GAZEBO_PLUGIN_PATH=~/catkin_ws/devel/lib:$GAZEBO_PLUGIN_PATH
```

运行

首先运行仿真环境，并打开qgroundcontrol地面站，然后运行自主定位、建图、轨迹规划、控制程序，开始试验

```
# 运行仿真环境 将主目录下的launch文件放到~/PX4_Firmware/launch下
roslaunch px4 bf.launch

#下载qgc程序。

https://docs.qgroundcontrol.com/master/en/qgc-user-guide/getting_started/download_and_install.html

#双击Qgc程序，打开软件，自动连接无人机
```

```
# 运行自主定位、建图、轨迹规划、控制程序

cd ~/TZB_WS
source devel/setup.bash
bash tzb.sh
bash plan.sh

#点击QGC软件中自主起飞，设置高度，确认起飞，到达后点击定点模式

#新开终端 输入并回车。同时QGC中切换为Offboard模式
rostopic pub /traj_start_trigger geometry_msgs/PoseStamped "header:
  seq: 0
  stamp:
    secs: 0
    nsecs: 0
  frame_id: ''
pose:
  position:
    x: 0.0
    y: 0.0
    z: 0.0
  orientation:
    x: 0.0
    y: 0.0
    z: 0.0
    w: 0.0"

#无人机开始自主飞行
```

注：bridge_fly_2为桥梁环境、corridor_dynamic_9为移动行人环境、floorplan1_static为走廊环境。目前程序默认为桥梁环境。如需修改为其他环境需要修改bf.launch中的第7行为对应世界文件。