ivrc_6t规划&控制库

环境配置

建议：使用命令行进行安装，脚本安装最后会将文件删除，容易找不到安装位置

0）安装catkin工具

sudo apt-get install python-catkin-tools

安装依赖库

1）grid_map_core

```
sudo apt-get update
sudo apt-get -y install ros-melodic-pcl-ros ros-melodic-costmap-2d ros-melodic-grid-map
```

2）ceres solver

```
bash install_ceres.sh
```

3）cppad

```
bash install_cppad.sh
```

按照龚小姐的文件安装

4）IPOPT solver

```
bash ./ipopt_install/install_ipopt.bash
```

5）benchmark

```
bash install_googlebenchmark.sh
```

6）rosparam handler

```
sudo apt-get install ros-melodic-rosparam-handler
```

7)OsqpEigen

```
bash install_osqp.sh
```

8)autoreconf

```
sudo apt-get install autoconf automake libtool
```

9)NLOPT solver

```
bash ./installers_plan/install_install_nlopt.sh
```

10)qt_build

```
sudo apt-get install ros-melodic-qt-build
```

文件原本需要qt4，先将其改成qt5检测

11)rviz_visual_tools

```
sudo apt-get install ros-melodic-rviz-visual-tools
```

12)OpenCV

OpenCV2.4 version

```
bash install_opencv2_4.sh
```

if there occurs the problem that the configuration incomplete:

CMake Error at cmake/OpenCVDetectCXXCompiler.cmake:85 (list)

or complied failed:

/usr/include/c++/7/cstdlib:75:15: fatal error: stdlib.h: 没有那个文件或目录

报错的时候选择下面的安装：

```
bash install_opencv2_4gcc6.sh
```

查看自己环境中的版本信息：

```
pkg-config --modversion opencv
```

部分代码运行环境需要OpenCV3，因此需要额外安装OpenCV3.2（至少3.2以上版本）

OpenCV3.2version

还不饿 Ubuntu18.04安装opencv3.2

安装包：

opencv3.2.0
opencv-contrib3.2.0

百度网盘下载：https://pan.baidu.com/s/1Mjq1n6Yj62iwVqLFgfULkQ
提取码：czeu

官网下载：

https://github.com/opencv/opencv/archive/3.2.0.tar.gz
https://github.com/opencv/opencv_contrib/archive/3.2.0.tar.gz

安装

```
sudo add-apt-repository "deb http://security.ubuntu.com/ubuntu xenial-security main"
sudo apt-get install cmake libgtk2.0-dev libavcodec-dev libavformat-dev libjpeg-dev libpng-dev libtiff-dev libtiff5-dev libswscale-dev libjasper-dev libcurl4-openssl-dev libtbb2 libdc1394-22-dev
```

解压之后在opencv3.2中建立build文件夹

`mkdir build && cd build`

配置文件

```
cmake -D CMAKE_BUILD_TYPE=Release -D OPENCV_GENERATE_PKGCONFIG=ON -D ENABLE_PRECOMPILED_HEADERS=OFF  CMAKE_INSTALL_PREFIX=/usr/local/opencv3.2 OPENCV_EXTRA_MODULES_PATH=/home/x/ivrc_ws/dependent_6t/dependence/opencv3.2/opencv_contrib-3.2.0/modules ..
```

```
make -j8
sudo make install
```

配置OpenCV库

```
sudo gedit  /etc/ld.so.conf.d/opencv.conf
```

添加OpenCV安装路径到.bashrc

```
export SNOPT_HOME=/usr/local/lib/
```

```
sudo ldconfig
```

13)FCL

```
bash install_libccd_fcl.sh
```

14)grid_map_sdf

```
sudo apt-get install ros-melodic-grid-map-sdf
```

15)Protobuf

```
bash install_googleProtobuf.sh
```

当catkin build时候报错“fatal error: google/protobuf/stubs/stringprintf.h: 没有那个文件或目录”

找到QpSpeed package包中CMakelist.txt，在133行添加你的安装路径如下：

```
include_directories(
    include
        /home/luyaomin/0Disk/ros_dep/protobuf/protobuf-3.13.0/src/
    ${catkin_INCLUDE_DIRS}
    ${Boost_INCLUDE_DIRS}
        ${EIGEN3_INCLUDE_DIR}
        ${PROTOBUF_INCLUDE_DIRS}
        third_lib/qpOASES/include
)
```

16)tf2_sensor_msgs

```
sudo apt-get install ros-melodic-tf2-sensor-msgs
```

17)casadi

```
bash install_casadi.sh
```

18)Vrep中坐标计算库

sudo apt-get -y install libgeographic-dev

19)跨平台检测

sudo apt-get install -qq libgtest-dev

20)vrep

添加文献权限

sudo chmod u+x vrep.sh

echo "alias vrep="$HOME /application/V-REP_PRO_EDU_V3_5_0_Linux/vrep.sh"" >> ~/.bashrc

21)tf2_sensor_msgs

```
sudo apt-get install ros-melodic-tf2-geometry-msgs
```

其他：

基本安装过程：

进入到软件安装包的主目录下；

```
mkdir build && cd build
```

```
cmake ..
```

```
make -j8
```

```
sudo make install
```

库文件：

1. controllib
2. Bench_mark
3. Casadi

给文件提高权限：

```
sudo chmod u+x install_libccd_fcl.sh
```

报错：

ivrc规划环境报错与修改

state_sampling `abs` report error

将对应文件里的 `abs` 改为 `fabs` ：

```
if ((path_length > 30 || path_length > ref_path.back().s * 3 / 4)
        && std::fabs(i - center_id) < std::fabs(nearest_id - center_id)) {
        nearest_id = i;
    }
```

stringprintf.h: 没有那个文件或目录

```
fatal error: google/protobuf/stubs/stringprintf.h: 没有那个文件或目录
 #include "google/protobuf/stubs/stringprintf.h"
```

参考：https://blog.csdn.net/NotANumber123/article/details/127367360

代码运行

规划启动文件：

地图更新：

roslaunch map_server map_server_node.launch

动态碰撞检测

roslaunch collision_detection collision_detection.launch

速度规划

roslaunch constrained_speed toyota.launch

控制启动文件

底层通讯

roslaunch ecucomm ecucomm_nodes.launch

轨迹跟踪

roslaunch path_tracking path_tracking_node.launch

速度控制

roslaunch speed_control speed_control_node.launch

图形化界面

roslaunch manual_gui manual_gui_node.launch

其他

全局路网

```
roslaunch waypoint_manager route_publisher_node.launch
```

状态采样demo

```
roslaunch state_sampling state_sampling_node.launch
```

录包内容：

```
rosbag record /global_path/traj_plan /trajectory /ecudatareport /sensor_fusion_output /imudata /gpsdata /final_traversable_area_optimized_topic /lidar_odometry_to_earth /GPSmsg /speed_plan /steer_cmd /lidar_preciseodometry_to_earth /single_traversable_area_optimized_topic /topology_global_path /search_plan_end /parking_space_control /lidar_odometry_for_mapping /gps_by_lidar_odometry /observer /collisionspeed_ref /vehiclestate /suspension /weapon /udp_recv_rawdata /udp_send_rawdata /speed_debug /insvelocity
```