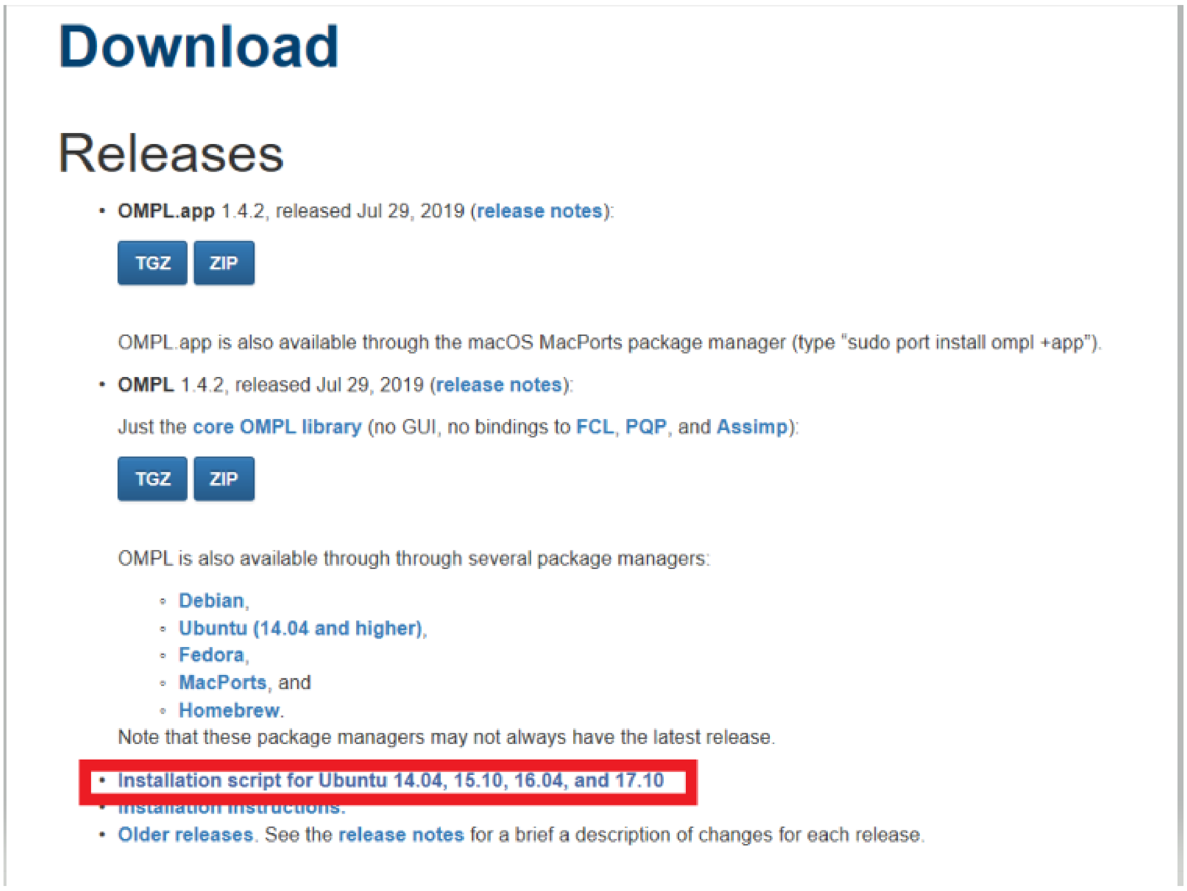
ompl 学习

1 下载ompl库

1　登录ompl官网：https://ompl.kavrakilab.org/index.html

2　进入Download页面，下载保存脚本文件install-ompl-ubuntu.sh



Download

Releases

- **OMPL.app** 1.4.2, released Jul 29, 2019 (**release notes**):

  TGZ ZIP

  OMPL.app is also available through the macOS MacPorts package manager (type "sudo port install ompl +app").

- **OMPL** 1.4.2, released Jul 29, 2019 (**release notes**):

  Just the **core OMPL library** (no GUI, no bindings to **FCL**, **PQP**, and **Assimp**):

  TGZ ZIP

  OMPL is also available through through several package managers:

  - **Debian**,
  - **Ubuntu (14.04 and higher)**,
  - **Fedora**,
  - **MacPorts**, and
  - **Homebrew**.

  Note that these package managers may not always have the latest release.

- **Installation script for Ubuntu 14.04, 15.10, 16.04, and 17.10**
- **Installation instructions.**
- **Older releases**. See the **release notes** for a brief a description of changes for each release.

3　运行脚本文件

```
sudo chmod+x install-ompl-ubuntu.sh
./install-ompl-ubuntu.sh
```

2 ROS查找依赖包ompl

1 修改`src/grid_path_searcher/CMakeLists.txt`，使用`find_package()`查找`ompl`的头文件、库路径等信息

```
find_package(Eigen3 REQUIRED)
find_package(PCL REQUIRED)
# add your code here: find_package(xxx REQUIRED)
```

2 在代码中添加使用到的`ompl`的头文件（该部分代码中已经添加）

见文件`src/grid_path_search/src/demo_node.cpp`

```
#include <ompl/config.h>
#include <ompl/base/StateSpace.h>
#include <ompl/base/Path.h>
#include <ompl/base/spaces/RealVectorBounds.h>
#include <ompl/base/spaces/RealVectorStateSpace.h>
#include <ompl/base/StateValidityChecker.h>
#include <ompl/base/OptimizationObjective.h>
#include <ompl/base/objectives/PathLengthOptimizationObjective.h>
#include <ompl/geometric/planners/rrt/RRTstar.h>
#include <ompl/geometric/SimpleSetup.h>
```

3 学习调用opml实现RRT*

要学会调用ompl实现RRT*，需要实现的功能如下：

- 把用户定义的起点、终点、地图用ompl库定义的数据结构表示
- 了解ompl调用RRT*的方法和步骤
- 把ompl库求解得到的路径转换为用户定义的数据结构

本次作业需要添加的代码集中在文件`src/grid_path_searcher.cpp/src/demo_node.cpp`中的一个函数`void pathFinding(const Vector3d start_pt, const Vector3d target_pt)`和一个类`class ValidityChecker : public ob::StateValidityChecker`。其中，`pathFinding()`交代了完整的代码流程，需要重点关注。

需要添加的代码在文件中以注释的形式标出，共有 7 处。

e.g.

```
class ValidityChecker : public ob::StateValidityChecker
{
public:
    ValidityChecker(const ob::SpaceInformationPtr& si) :
        ob::StateValidityChecker(si) {}
    // Returns whether the given state's position overlaps the circular obstacle
    bool isValid(const ob::State* state) const
    {
        // We know we're working with a RealVectorStateSpace in this
        // example, so we downcast state into the specific type.
        const ob::RealVectorStateSpace::StateType* state3D =
            state->as<ob::RealVectorStateSpace::StateType>();
        /**
        *
        *
        STEP 1: Extract the robot's (x,y,z) position from its state
        *
        *
        */
        return _RRTstar_preparatory->isObsFree(x, y, z);
    }
};
```