

# 第1季

## 第12章：典型自主导航系统



主讲人：张虎

(小虎哥哥爱学习)

- 先导课
- 第1季：快速梳理知识要点与学习方法 ✓
- 第2季：详细推导数学公式与代码解析
- 第3季：代码实操以及真实机器人调试
- 答疑课

----- (永久免费 • 系列课程 • 长期更新) -----

## 本书内容安排

### 一、编程基础篇

第1章：ROS入门必备知识

第2章：C++编程范式

第3章：OpenCV图像处理

### 二、硬件基础篇

第4章：机器人传感器

第5章：机器人主机

第6章：机器人底盘

### 三、SLAM篇

第7章：SLAM中的数学基础

第8章：激光SLAM系统

第9章：视觉SLAM系统

第10章：其他SLAM系统

### 四、自主导航篇

第11章：自主导航中的数学基础

第12章：典型自主导航系统

第13章：机器人SLAM导航综合实战

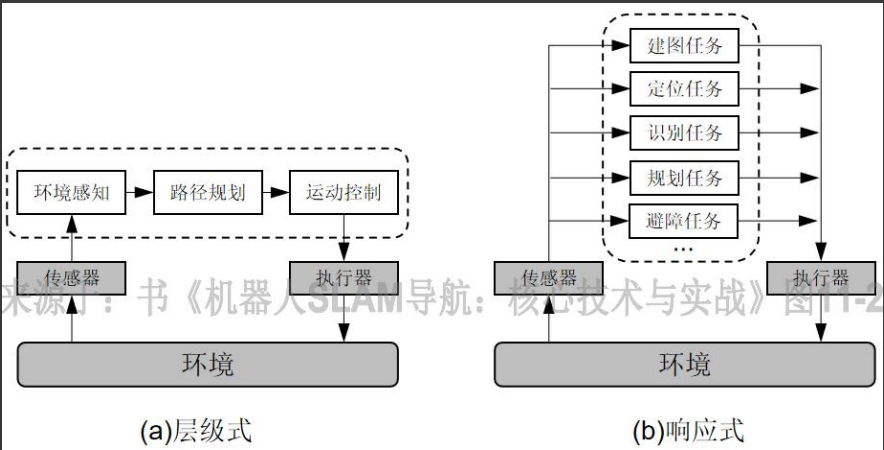
# 自主导航 vs 自主导航系统

实际问题

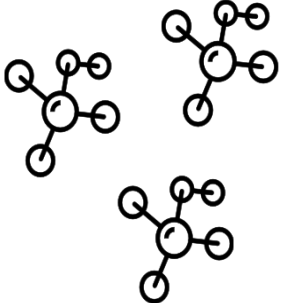
工程实现

软件项目

- 我在哪？
- 我将到何处去？
- 我该如何去？



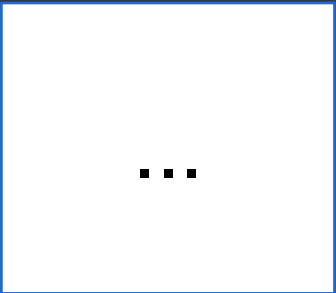
- ros-navigation
- riskrrt
- autoware
- ...



人工规则库 (IF-ELSE)



人工神经网络端到端



## 内容概要

12.1 ros-navigation导航系统

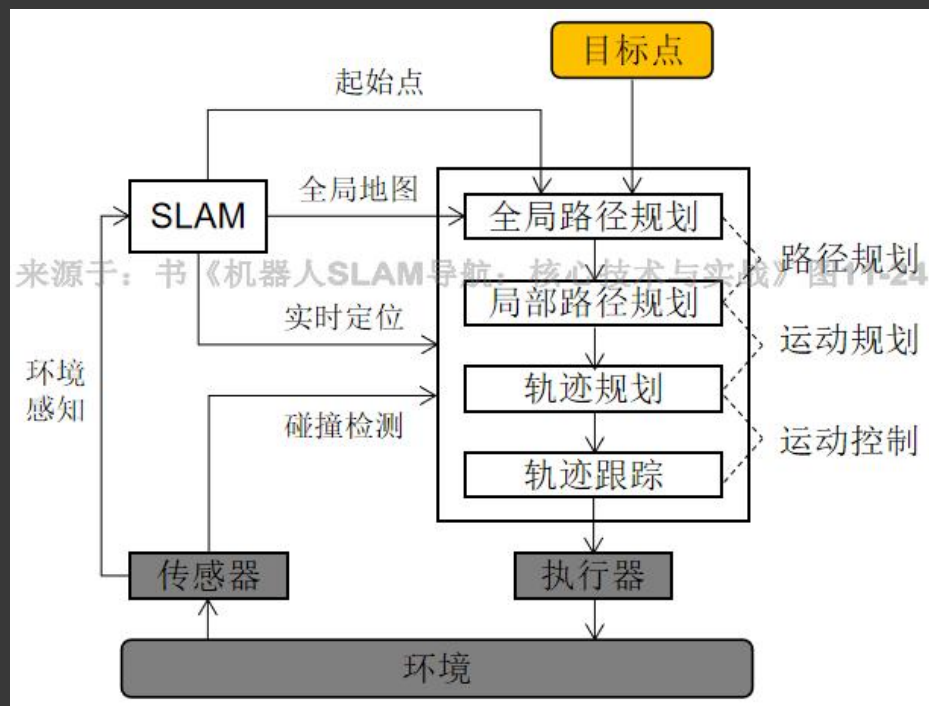
12.2 riskrrt导航系统

12.3 autoware导航系统

12.4 导航系统面临的一些挑战

## 12.1 ros-navigation导航系统

- ros-navigation原理分析
- ros-navigation源码解读
- ros-navigation安装与运行
- 路径规划改进
- 环境探索



自主导航的常见技术架构图

### 输入：

- 导航目标
- 定位信息
- 地图信息
- 障碍信息

### 输出：

- 控制量

## 12.1 ros-navigation导航系统

### ■ ros-navigation原理分析

- ros-navigation源码解读
- ros-navigation安装与运行
- 路径规划改进
- 环境探索

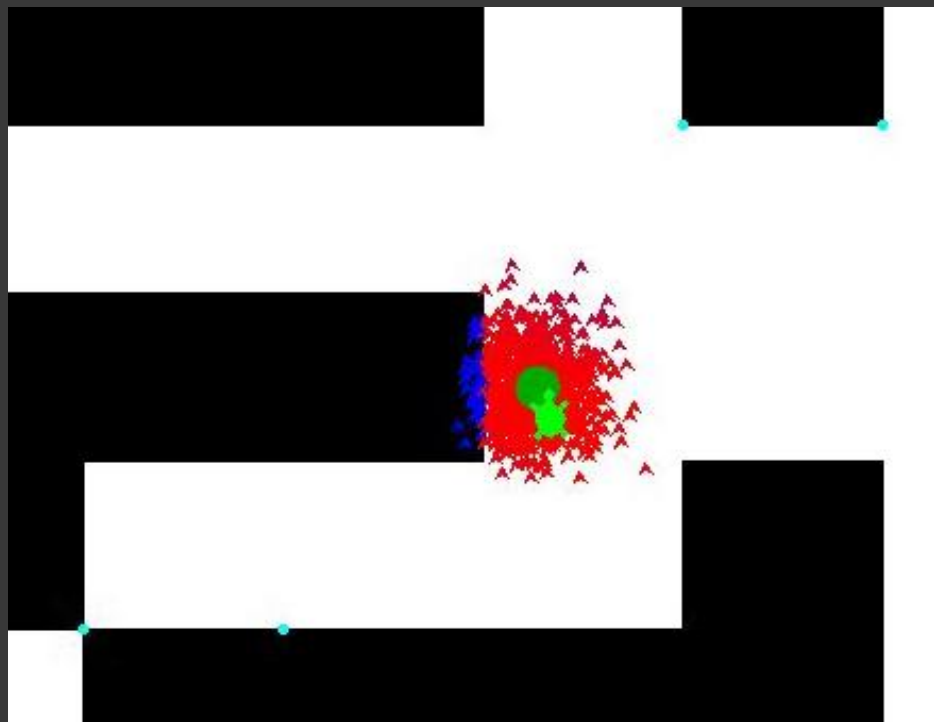
AMCL定位

Costmap代价地图

ros-navigation系统框架

在ros-navigation中提供了一种比SLAM定位更轻量级的方案，

即AMCL (Adaptive Monte Carlo Localization, 自适应蒙特卡洛定位)，具体实现功能包有amcl和amcl3d。



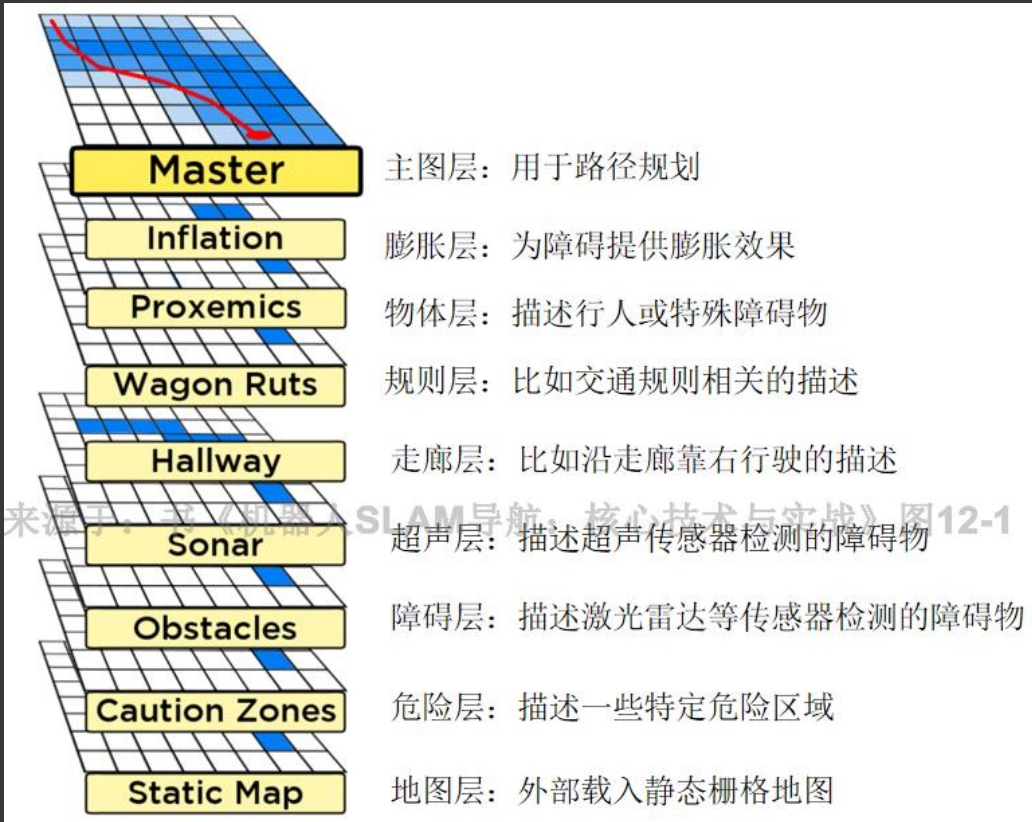
AMCL定位过程粒子滤波器的粒子分布情况

# 12.1 ros-navigation导航系统

- ros-navigation原理分析
- ros-navigation源码解读
- ros-navigation安装与运行
- 路径规划改进
- 环境探索



导航控制策略的首要任务是避障，那么对障碍物的度量就成了关键问题。由于SLAM直接提供的地图格式种类繁多（比如特征地图、点云地图、几何地图、栅格地图、拓扑地图等），这些地图度量障碍物的能力参差不齐。为了解决各种复杂障碍物的度量问题，ros-navigation采用代价地图（Costmap）对障碍物进行统一度量。





## 12.1 ros-navigation导航系统

- ros-navigation原理分析
- ros-navigation源码解读
- ros-navigation安装与运行
- 路径规划改进
- 环境探索

## AMCL定位

## Costmap代价地图

## ros-navigation系统框架

ros-navigaion其实是一个功能包集，里面包含了大量的ROS功能包

以及各种算法的具体实现节点。

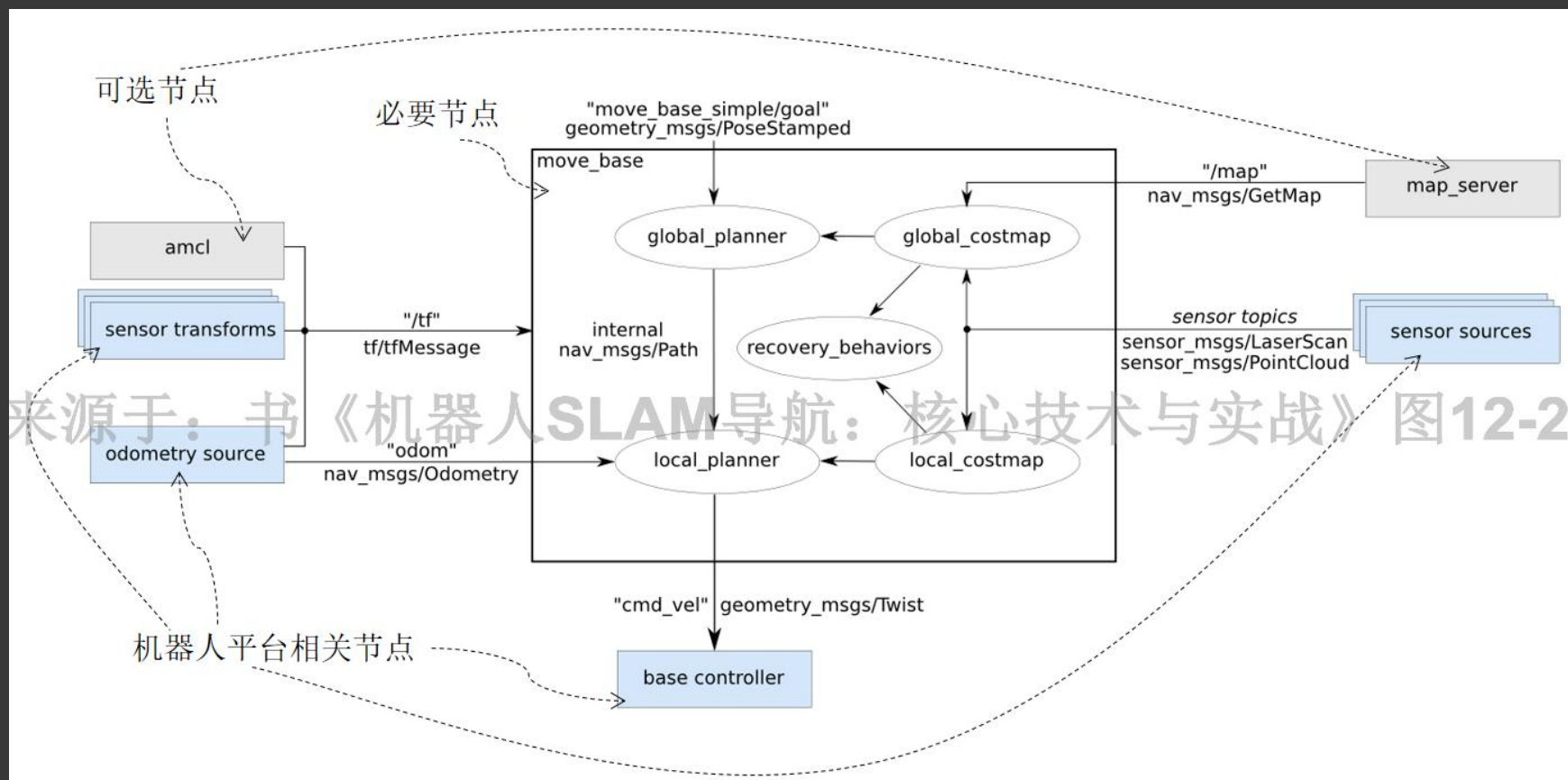


图12-2

7/27

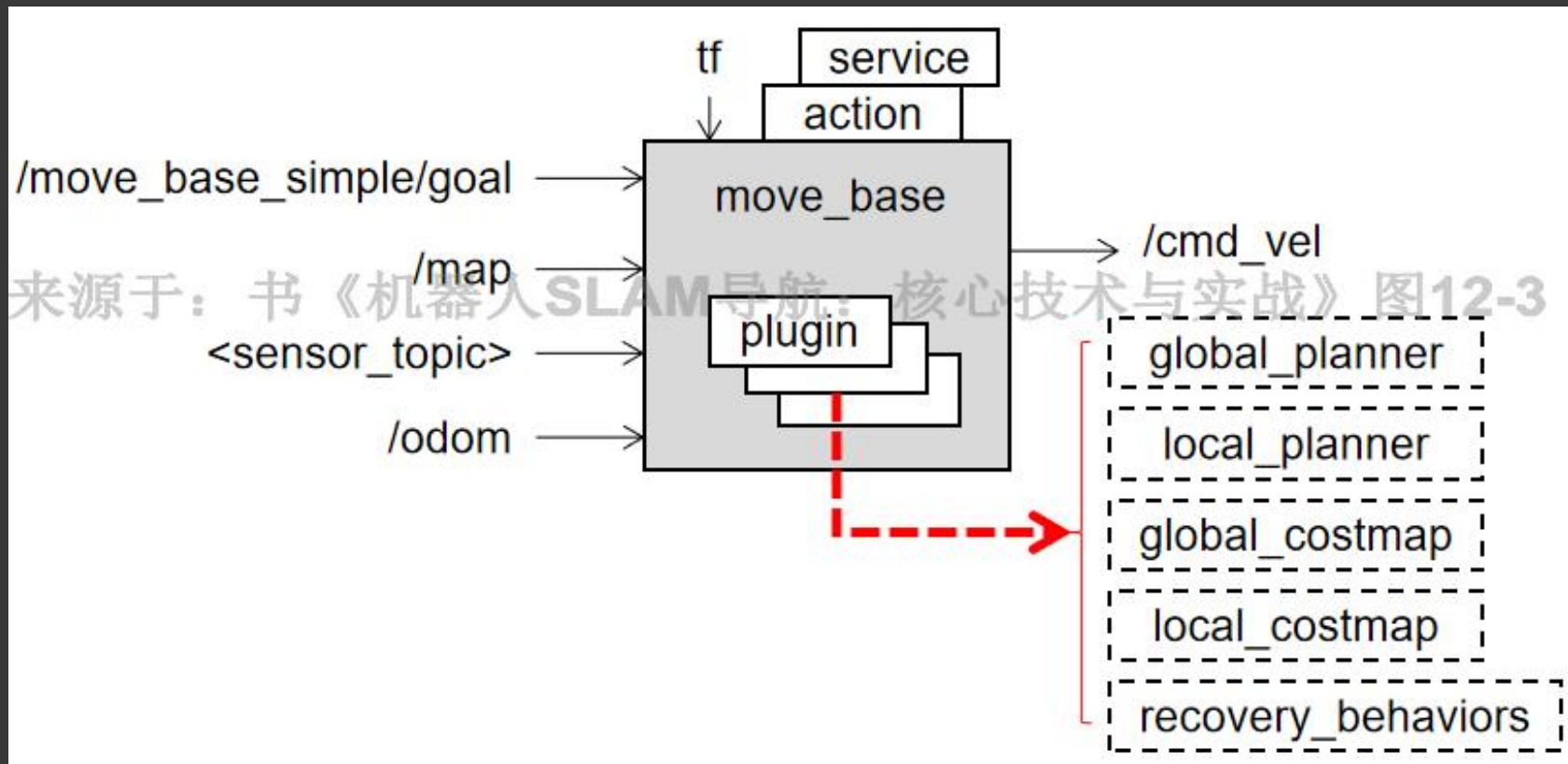
## ros-navigation系统框架

课件下载: [www.xiihoo.com](http://www.xiihoo.com)



## 12.1 ros-navigation导航系统

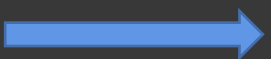
- ros-navigation原理分析
- **ros-navigation源码解读** →
- ros-navigation安装与运行
- 路径规划改进
- 环境探索



ros-navigation代码框架

# 12.1 ros-navigation导航系统

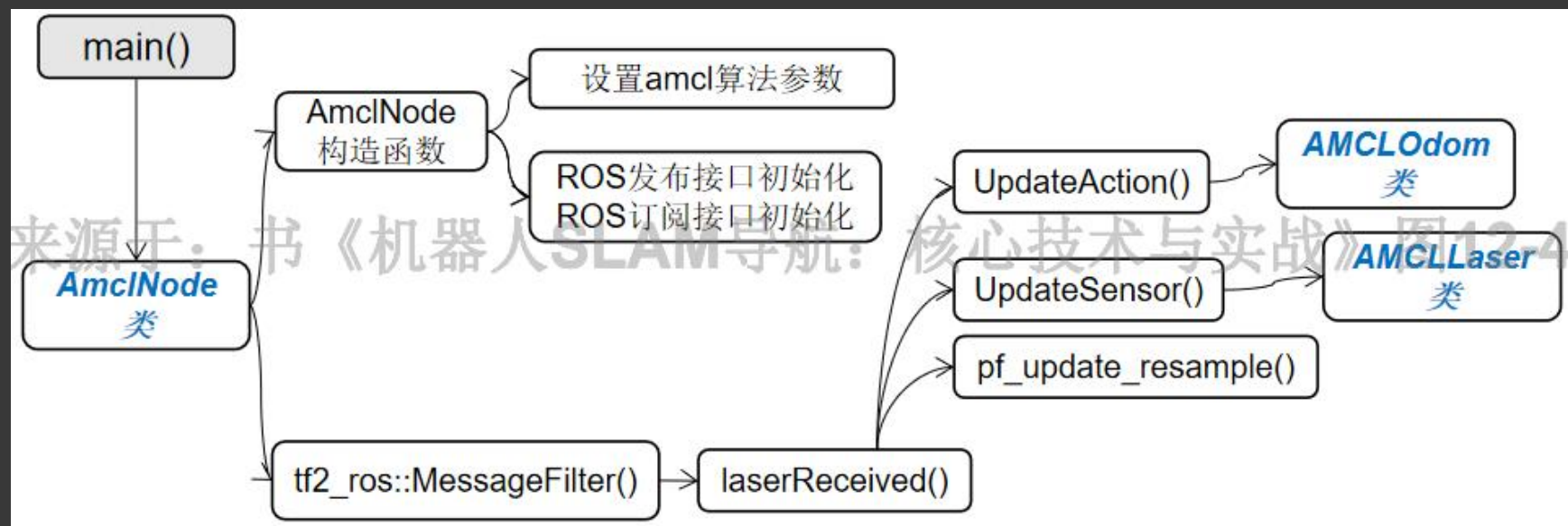
- ros-navigation原理分析
- ros-navigation源码解读
- ros-navigation安装与运行
- 路径规划改进
- 环境探索



功能包	类型	说明
amcl	可选功能包	提供全局定位
map_server		加载静态地图文件
move_base	必要功能包	导航框架的虚拟壳体
nav_core	插件接口组件	专门为 BaseGlobalPlanner, BaseLocalPlanner, RecoveryBehavior 提供统一的插件接口
navfn	全局路径规划插件	基于 Dijkstra 的全局路径规划
global_planner		在 navfn 基础上做了改进
carrot_planner		处理目标点更灵活的全局路径规划
base_local_planner	局部路径规划插件	基于动态窗口轨迹试探的局部路径规划
dwa_local_planner		在 base_local_planner 基础上做了改进
costmap_2d	代价地图插件	实现二维代价地图
rotate_recovery	恢复策略插件	原地旋转 360° 来清除空间障碍物
move_slow_and_clear		缓慢移动来清除障碍物
clear_costmap_recovery		强制清除一定半径范围内的障碍物
voxel_grid	其他	实现三维体素栅格
fake_localization		用里程计推演提供虚假的全局定位
move_base_msgs		定义 move_base 通信用到的消息类型

## 12.1 ros-navigation导航系统

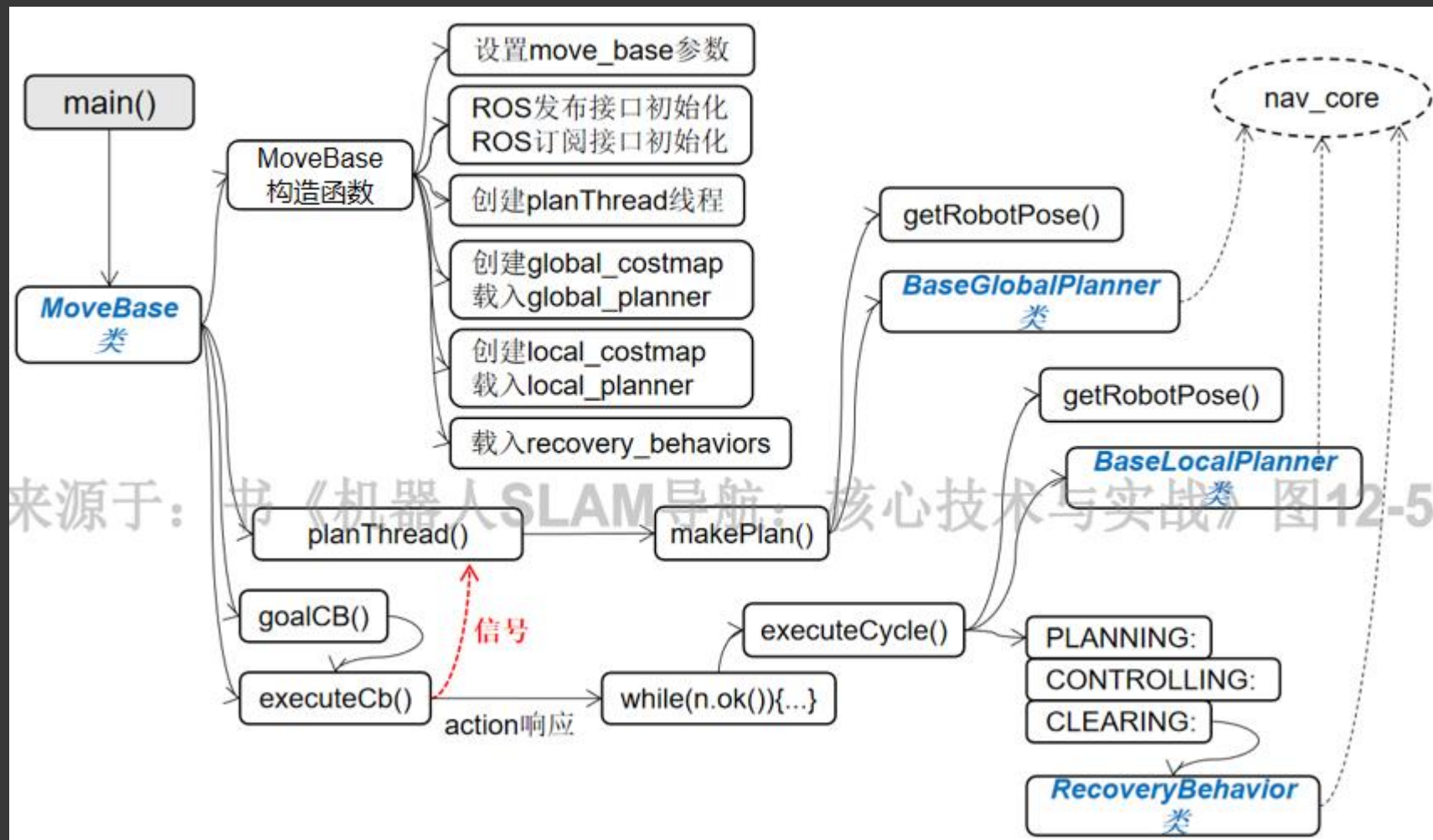
- ros-navigation原理分析
- **ros-navigation源码解读**
- ros-navigation安装与运行
- 路径规划改进
- 环境探索



amcl程序调用流程

## 12.1 ros-navigation导航系统

- ros-navigation原理分析
- ros-navigation源码解读
- ros-navigation安装与运行
- 路径规划改进
- 环境探索



move\_base程序调用流程

## 12.1 ros-navigation导航系统

- ros-navigation原理分析
- ros-navigation源码解读
- ros-navigation安装与运行
- 路径规划改进
- 环境探索



代价地图插件调用流程

## 12.1 ros-navigation导航系统

- ros-navigation原理分析
- ros-navigation源码解读
- **ros-navigation安装与运行**
- 路径规划改进
- 环境探索

环境：Ubuntu18.04+ROS melodic

### ①安装依赖：

```
#安装ros-navigation及其关联功能包，依赖也会随之安装上
sudo apt install ros-melodic-navigation*
#卸载ros-navigation但保留其依赖
sudo apt remove ros-melodic-navigation
```

### ②编译源码：

```
#切换到工作空间目录
cd ~/catkin_ws/src/
#下载ros-navigation源码
git clone https://github.com/ros-planning/navigation.git
cd navigation
#查看代码版本是否为molodic，如果不是请使用git checkout命令切换到对应版本
git branch
#编译
cd ~/catkin_ws/
catkin_make
```



# 12.1 ros-navigation导航系统

- ros-navigation原理分析
- ros-navigation源码解读
- ros-navigation安装与运行
- 路径规划改进
- 环境探索

在实际机器人运行:

① 机器人平台相关节点的配置与启动:

```
#启动底盘
roslaunch xiihoo_bringup minimal.launch

#启动激光雷达
roslaunch ydlidar my_x4.launch

#启动底盘urdf描述
roslaunch xiihoo_description xiihoo_description.launch
```

minimal.launch启动文件:



my\_x4.launch启动文件:



xiihoo\_description.launch启动文件:



\*.urdf配置文件:



② 地图供应节点的配置与启动:

```
#载入地图
roslaunch map_server map_pub.launch
```

map\_pub.launch启动文件:



③ 全局定位节点的配置与启动:

```
#启动全局定位
roslaunch amcl amcl.launch
```

map\_pub.launch启动文件:

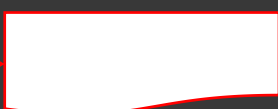


④ 导航核心节点的配置与启动:

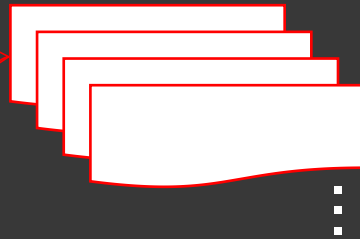
```
#启动move_base自主导航
roslaunch xiihoo_nav move_base.launch

#启动工作台电脑的rviz
rviz
```

move\_base.launch启动文件:



参数配置文件:

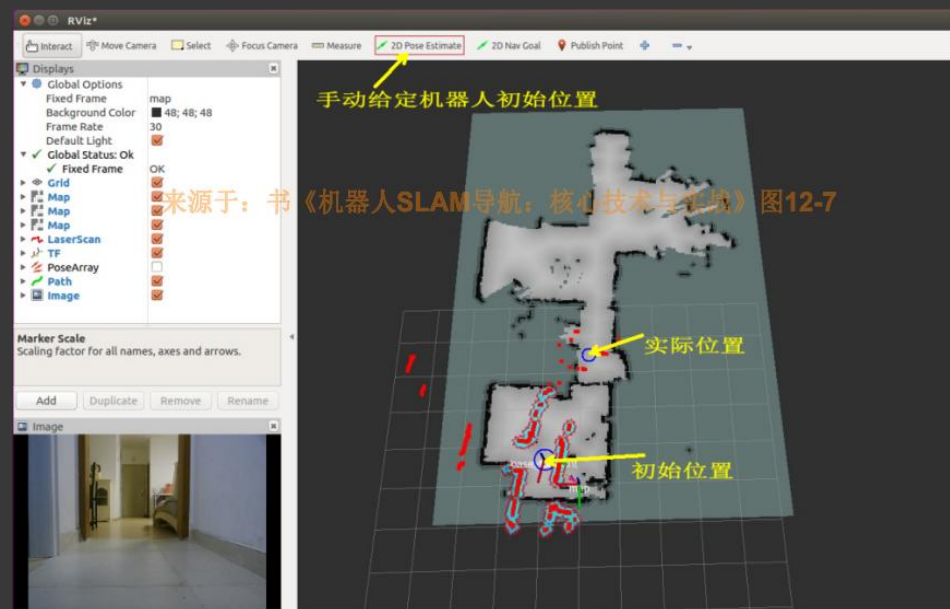


- move\_base\_params.yaml
- costmap\_common\_params.yaml
- global\_costmap\_params.yaml
- local\_costmap\_params.yaml
- navfn\_planner\_params.yaml
- base\_local\_planner\_params.yaml

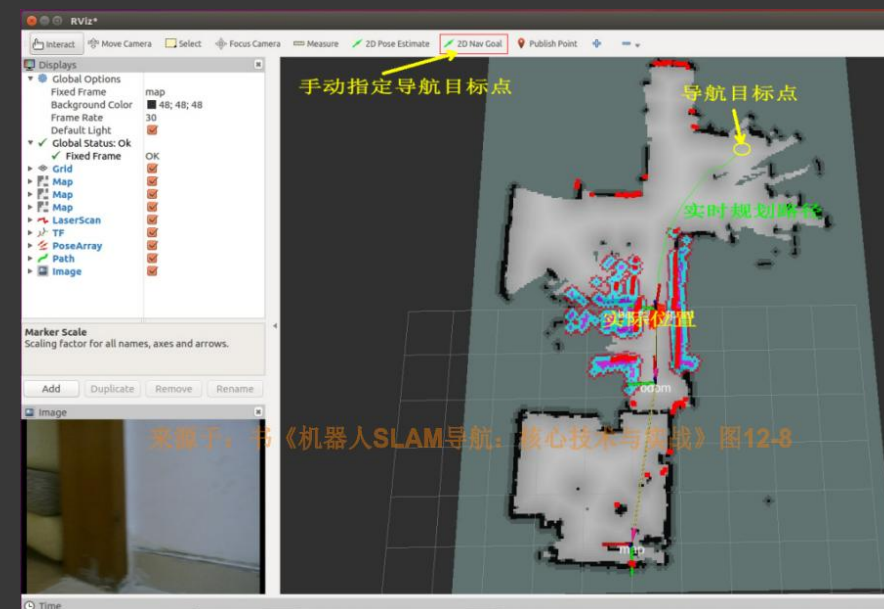


## 12.1 ros-navigation导航系统

- ros-navigation原理分析
- ros-navigation源码解读
- ros-navigation安装与运行
- 路径规划改进
- 环境探索



在rviz中手动给定机器人初始位置



在rviz中手动指定导航目标点

## 12.1 ros-navigation导航系统

- ros-navigation原理分析
- ros-navigation源码解读
- ros-navigation安装与运行
- 路径规划改进
- 环境探索

ros-navigation中虽然集成了一些不同的路径规划插件供用户在不同场景下选择使用，其中可选的全局路径规划插件包括navfn、global\_planner和carrot\_planner，可选的局部路径规划插件包括base\_local\_planner和dwa\_local\_planner。不过这些插件还远不能满足研究和工程应用时的多样化需求，此时用户可以选择其他第三方路径规划插件加载到move\_base中使用，或者用户根据nav\_core的接口规范自己开发所需的路径规划插件。

- 基于图搜索的全局路径规划SBPL\_Lattice\_Planner

1. SBPL\_Lattice\_Planner与SBPL的安装

2. 在move\_base中加载SBPL\_Lattice\_Planner插件

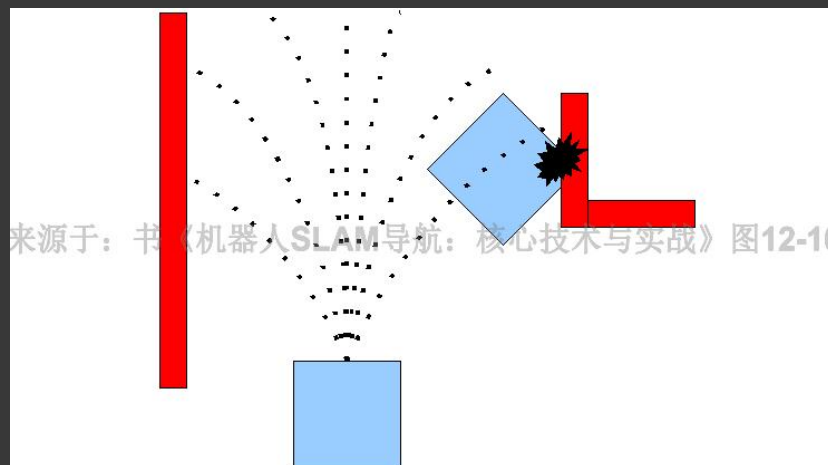
- 基于采样的全局路径规划srl\_global\_planner

- 基于弹性带的局部路径规划teb\_local\_planner

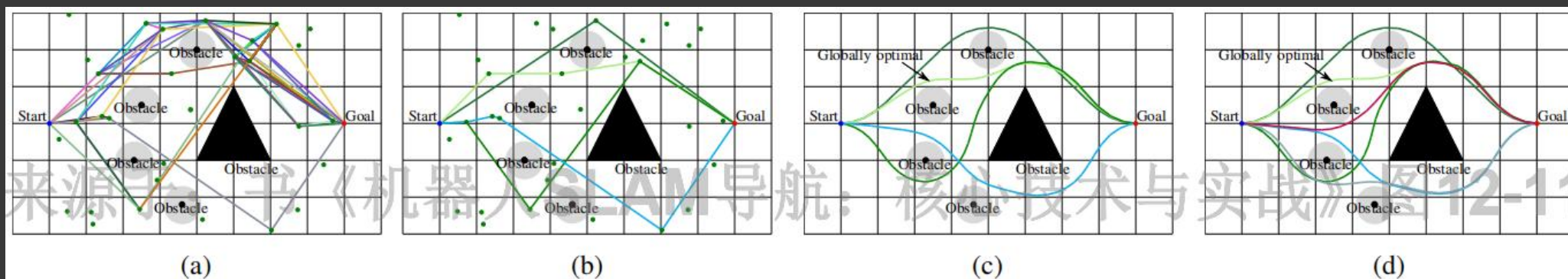
## 12.1 ros-navigation导航系统

- ros-navigation原理分析
- ros-navigation源码解读
- ros-navigation安装与运行
- 路径规划改进
- 环境探索

动态窗口理论的避障思路：



弹性带理论的避障思路：



## 12.1 ros-navigation导航系统

- ros-navigation原理分析
- ros-navigation源码解读
  - 自主导航通常是在已知环境地图的条件下进行环境交互的行为，也就是从环境的一个地点移动到另一个地点。
- ros-navigation安装与运行
- 路径规划改进
  - SLAM则通常是在人为操控的条件下进行环境交互的行为，也就是为未知环境构建地图模型。
- 环境探索
  - 自主导航和SLAM都是在受限条件下进行环境交互，自主导航的受限条件是环境地图必须已知，SLAM的受限条件是构建未知环境地图的过程需要人为操控。
  - 如果将自主导航与SLAM结合起来，就可以让机器人与环境交互的过程真正自主化，即所谓的环境探索。



## 12.1 ros-navigation导航系统

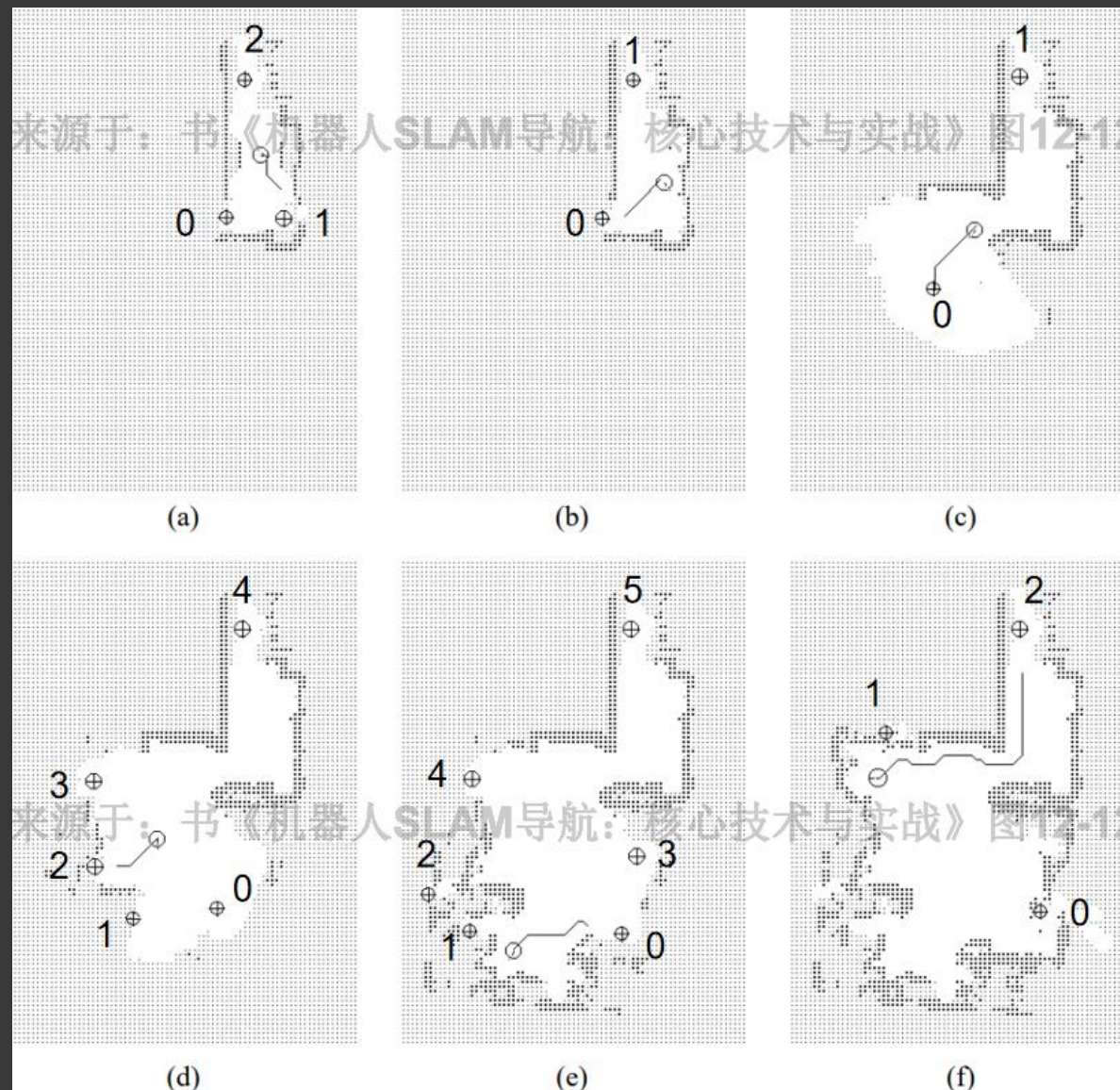
- ros-navigation原理分析
- ros-navigation源码解读
- ros-navigation安装与运行

- 路径规划改进

- 环境探索

frontier\_exploration

rrt\_exploration



frontier\_exploration的探索策略

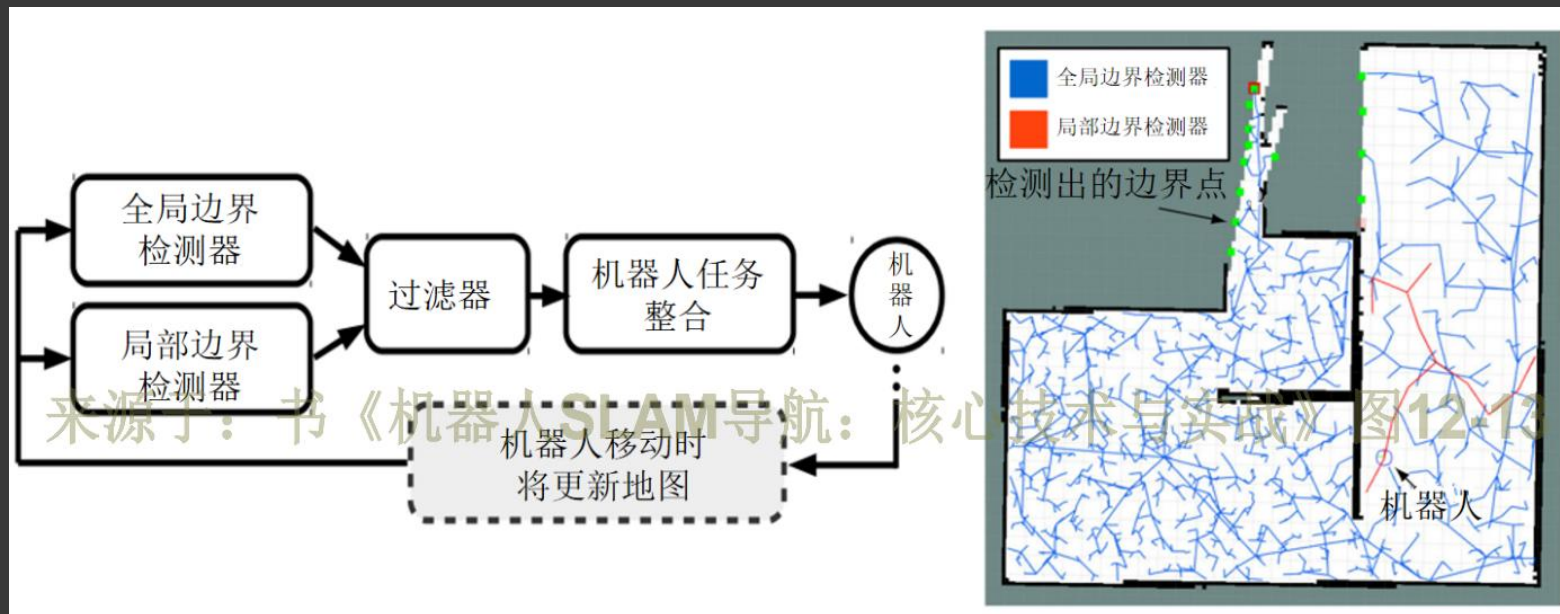
## 12.1 ros-navigation导航系统

- ros-navigation原理分析
- ros-navigation源码解读
- ros-navigation安装与运行
- 路径规划改进

- 环境探索

frontier\_exploration

rrt\_exploration



rrt\_exploration的探索策略

## 内容概要

12.1 ros-navigation导航系统

12.2 riskrrt导航系统

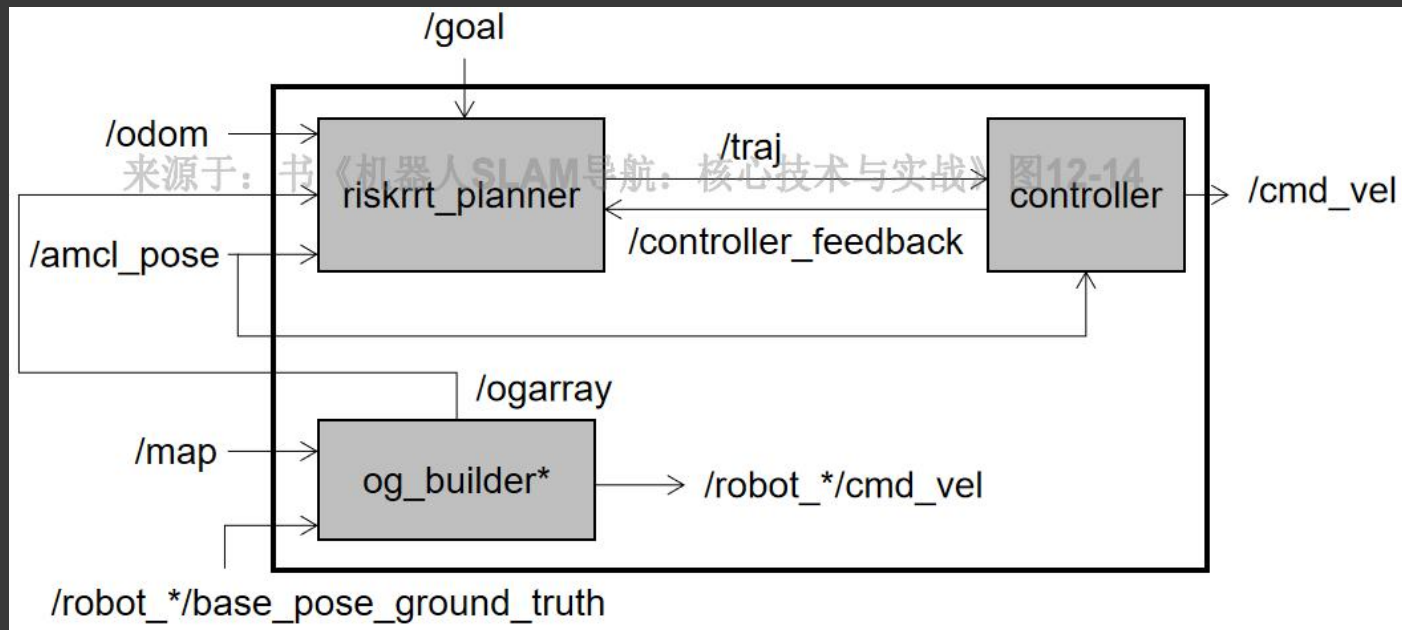
12.3 autoware导航系统

12.4 导航系统面临的一些挑战



## 12.2 riskrrt导航系统

- 对于大多数ROS学习者，最先接触到的导航系统基本都是ros-navigation。不过机器人领域的导航系统实现并不只有ros-navigation，这里要介绍的riskrrt就是另外一种导航系统实现。
- riskrrt的系统框架基本与ros-navigation一样，也就是地图供应、全局定位和路径规划这些主要模块。
- riskrrt的特别之处体现在对动态障碍物的处理更加严格，在走廊、机场、路口等人流量大的场景中对机器人与动态障碍物发生碰撞的风险把控更加严格，这种对动态障碍物碰撞的风险把控通过基于风险的RRT全局路径规划实现。



## 内容概要

12.1 ros-navigation导航系统

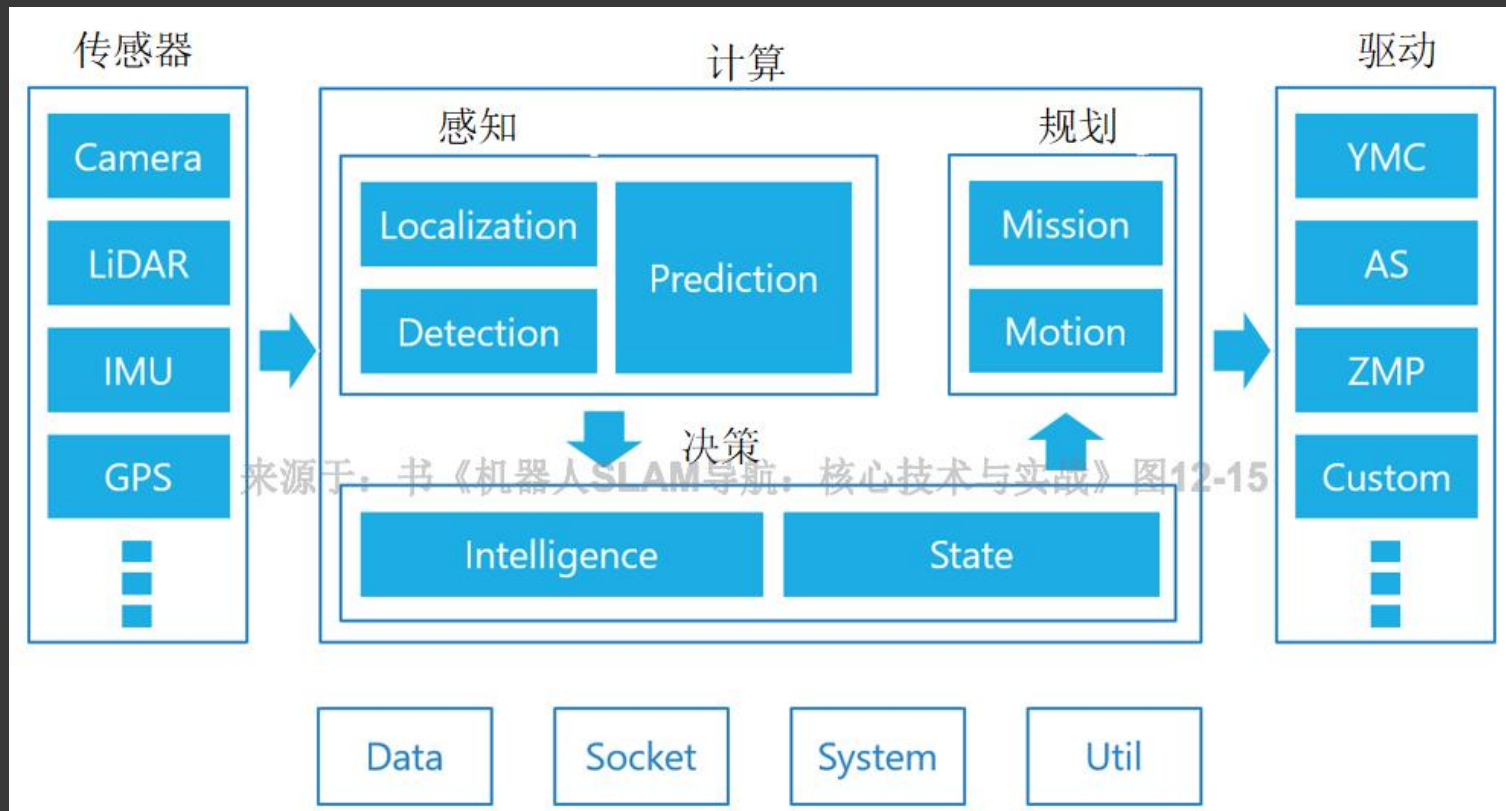
12.2 riskrrt导航系统

12.3 autoware导航系统

12.4 导航系统面临的一些挑战

## 12.3 autoware导航系统

- ros-navigation和riskrrt导航系统主要都是用于机器人的低速导航，并且大多基于2D地图。
- 而autoware导航系统主要用于无人驾驶汽车的高速导航，并且基于3D地图。
- 除了所导航速度高一点和采用3D地图外，autoware的原理几乎与ros-navigation一样。



## 内容概要

12.1 ros-navigation导航系统

12.2 riskrrt导航系统

12.3 autoware导航系统

12.4 导航系统面临的一些挑战

## 12.4 导航系统面临的一些挑战

- 立体障碍物是机器人导航中很头疼的一个问题，因为2D激光雷达只能扫描某个平面内的障碍物，对于高于或低于扫描平面的障碍物是无法探测的。比如一个很矮的扫地机器人能从桌子底下穿过去，而在扫地机器人上安装一个较高的货架后就会导致碰撞。很自然会想到在机器人不同高度不同方向都装上传感器，这样就能避免因个别传感器探测盲区而发生碰撞了。但将机器人的全身都覆盖上传感器显然不现实，并且传感器本身也有探测距离和视角的盲区问题。
- 玻璃类型的透明障碍物、镜面反射障碍物、场景多径效应、强光烟雾干扰等因素，都会导致激光雷达或相机的探测失灵。在这种情况下，如何保证机器人的安全也是个棘手的问题。另外一些特殊的情况，比如机器人在下斜坡时地面如何从障碍描述中分离也是个问题。
- 还有就是路径规划和轨迹跟踪都强依赖于全局定位，当机器人全局定位丢失后，路径规划和轨迹跟踪将直接崩溃，那么导航系统的稳定性怎么保证也是个大问题。

- 例程源码下载： [https://github.com/xiihoo/Books\\_Robot\\_SLAM\\_Navigation](https://github.com/xiihoo/Books_Robot_SLAM_Navigation)
- 课件PPT下载： [www.xiihoo.com](http://www.xiihoo.com)

敬请关注,长期更新...

下集预告