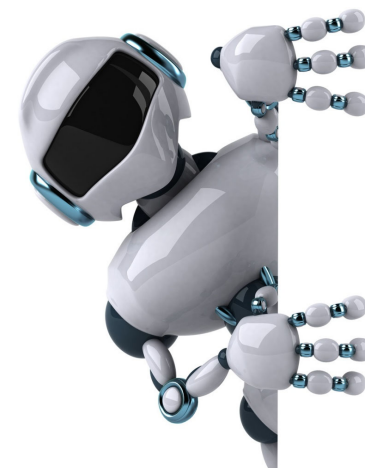


# 智能移动机器人

## SLAM

---

西安天之博特科技有限公司



# 公司简介



天之博特成立于2016年，专注于智能移动机器人模块及整机的设计、研发、生产制造，将最前沿的人工智能与机器人技术应用到产品中，推动移动机器人在行业中落地。天之博特是国内智能移动机器人教育行业的先行者，为高校新工科和双一流建设提供人工智能和机器人专业的全套的解决方案。

# 目录

01

上手实操三种建图算法

02

蒙特卡洛定位算法

03

AMCL功能包详解

# 在Tianracer上进行GMapping建图

网络配置好之后，在小车上启动Tianracer的程序

```
roslaunch tianracer_bringup tianracer_bringup.launch
```

GMapping可以在机器人板载的电脑上运行，也可以在自己电脑上运行。建议以WIFI信号为准

```
roslaunch tianracer_slam tianracer_gmapping.launch
```

RViz的可视化程序需要自己的电脑上运行

```
roslaunch tianracer_rviz view_mapping.launch
```

建图完成后需要在小车上运行保存地图的命令。注意执行命令时的路径、以及保存在哪里

```
roslaunch map_server map_saver -f name_of_map  
(or) roslaunch tianracer_slam map_save.launch
```

# 在Tianracer上进行HectorSLAM建图

网络配置好之后，在小车上启动Tianracer的程序

```
roslaunch tianracer_bringup tianracer_bringup.launch
```

HectorSLAM可以在机器人板载的电脑上运行，也可以在自己电脑上运行。

```
roslaunch tianracer_slam tianracer_hector.launch
```

RViz的可视化程序需要自己的电脑上运行

```
roslaunch tianracer_rviz view_mapping.launch
```

建图完成后需要运行保存地图的命令。注意执行命令时的路径、以及保存在哪里

```
roslaunch map_server map_saver -f name_of_map  
(or) roslaunch tianracer_slam map_save.launch
```

# 在Tianracer上进行cartographer建图

网络配置好之后，在小车上启动Tianracer的程序

```
roslaunch tianracer_bringup tianracer_bringup.launch
```

在自己的电脑上运行

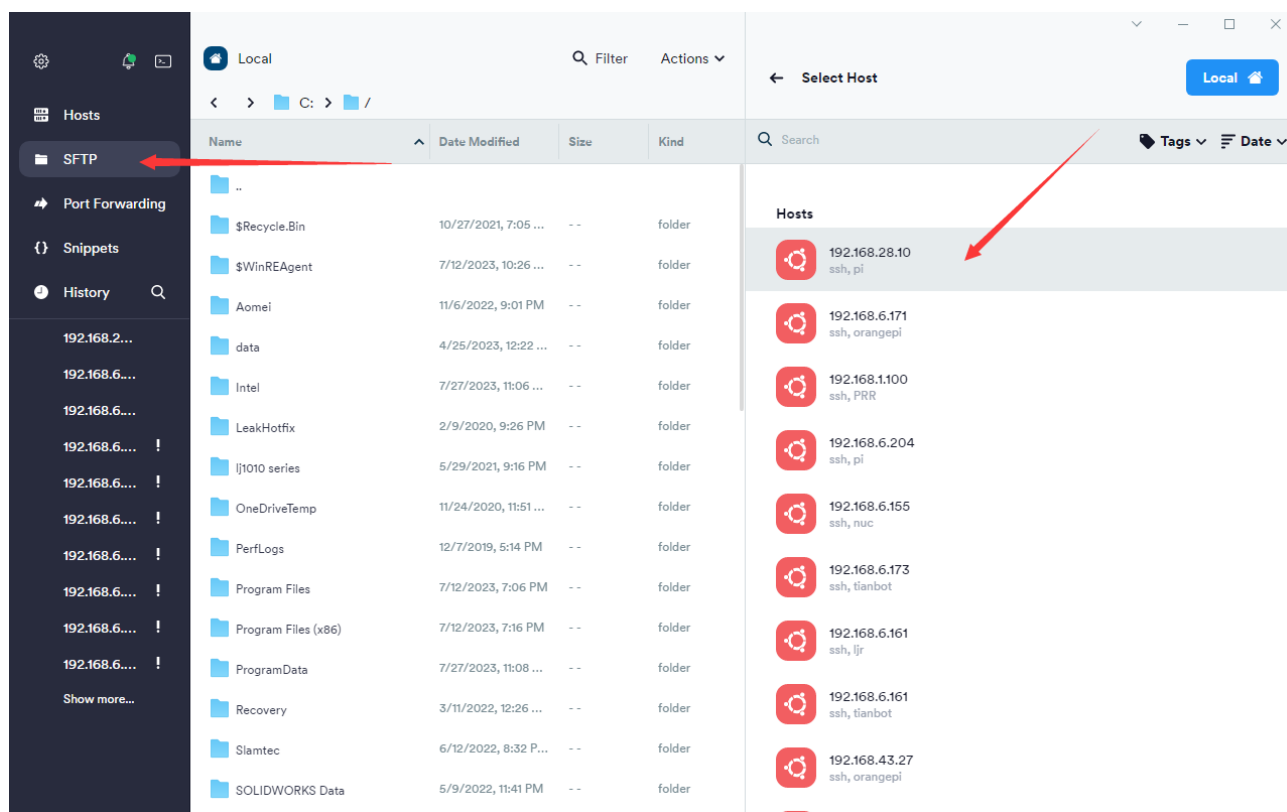
```
roslaunch tianracer_slam tianracer_cartographer.launch
```

遥控建图，然后保存地图。注意Cartographer的地图表达不同，建图命令可额外添加参数。

```
roslaunch map_server map_saver --occ 51 --free 49 -f name_of_map  
(or) roslaunch tianracer_slam map_save.launch
```

# 保存注意事项

需要保证地图保存在小车的工作空间下：../src/[tianracer](#)/tianracer\_slam/maps 目录  
打开termius，进入自己小车的SFTP服务器里面

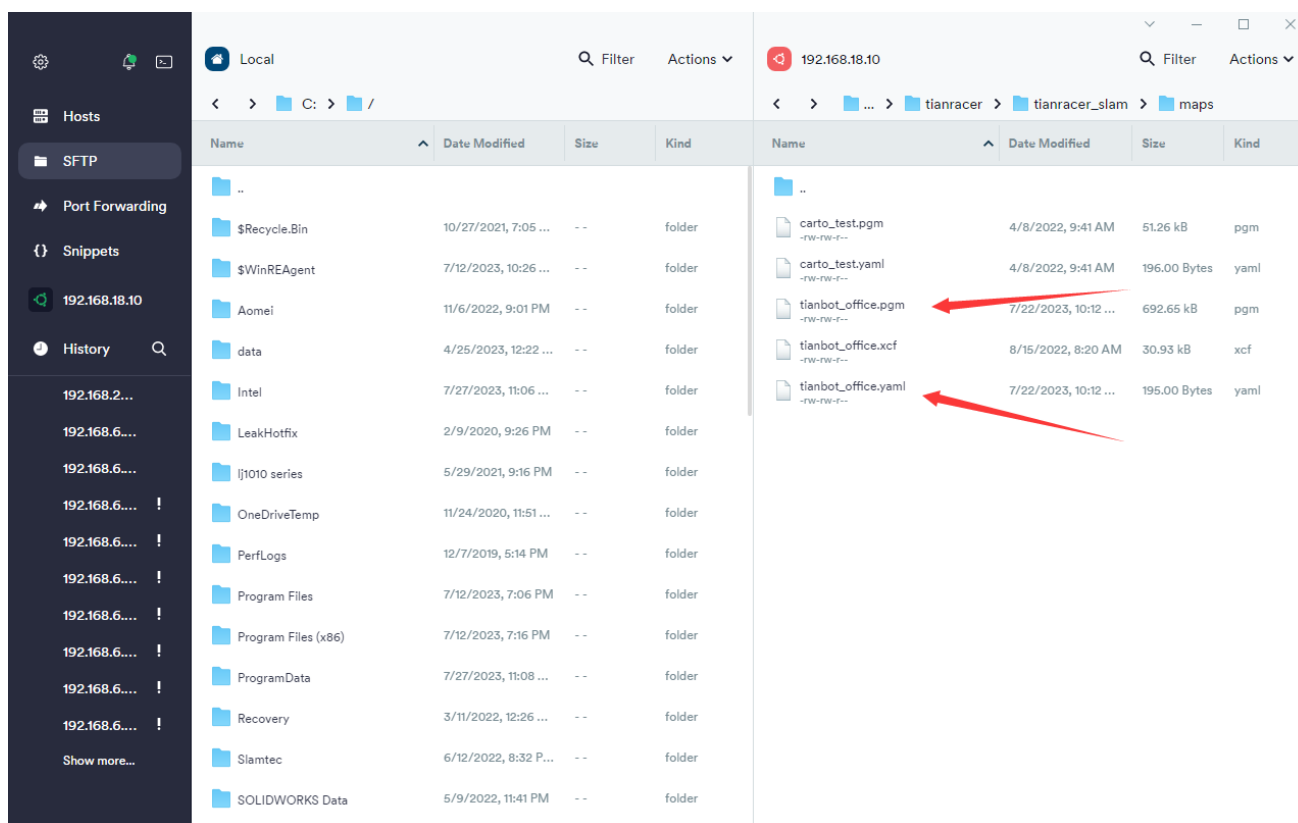


# 保存注意事项

如果使用launch文件保存，文件名称应如下：

tianbot\_office.pgm tianbot\_office.yaml

如果不一致需要自行调整。





## 蒙特卡洛定位算法

2

## AMCL定位功能包详解

3

# AMCL功能包介绍

- ROS中，利用激光扫描数据在地图中定位的功能包是amcl（Adaptive Monte Carlo Localization），其中只有一个节点名为amcl，小写是符合ROS中的功能包与节点的命名规范，大写的AMCL则符合英文缩写的文法规范。
- AMCL功能包从属于导航元功能包（Navigation Meta-Package），因为定位功能一般都是为机器人导航服务，当然AMCL功能包也可以独立使用。AMCL功能包就是基于粒子滤波的蒙特卡洛定位算法在ROS框架下的程序实现。

# AMCL功能包介绍

- 粒子滤波算法应用到实际的移动机器人定位问题时，还有一些问题需要考虑。虽然足够多的粒子能够保证机器人实现全局定位，但是多少粒子才是“足够多”呢？显然，地图越大，AMCL就需要越多的粒子以保证能够全局定位成功。如果给定机器人的初始位姿或者全局定位时粒子已经收敛到真实位姿附近，那么仅需少量的粒子即可追踪机器人的位姿。AMCL采用了KLD（Kullback-Leibler distance）采样来调整样本数量，以提高算法效率。这也是AMCL中的适应性（Adaptive）的来源。
- 除了KLD适应性采样，AMCL还应用了《概率机器人学》中所给出的 `sample_motion_model_odometry`、`beam_range_finder_model`、`Augmented_MCL`、`likelihood_field_range_finder_model`等算法，读者可以参考原著以及AMCL的源码。

# 在Tianracer上进行nav初步准备

网络配置好之后，在小车上启动Tianracer的程序

```
roslaunch tianracer_bringup tianracer_bringup.launch
```

在小车上打开teb导航launch文件

```
roslaunch tianracer_navigation tianracer_teb_nav.launch
```

在电脑上打开rviz，可视化amcl运行状态

```
roslaunch tianracer_rviz view_amcl.launch
```

# 谢谢观看



扫描二维码关注公众号

