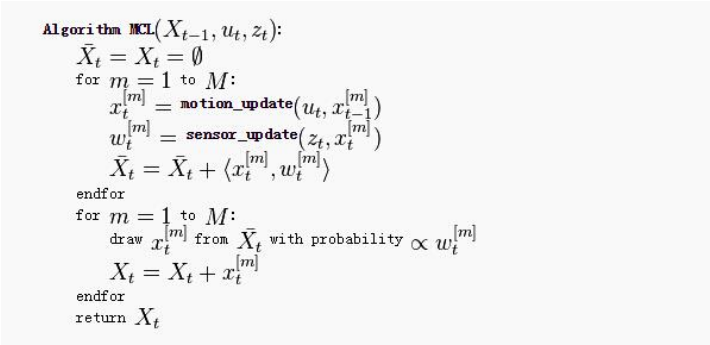
第十八课 Amcl 定位算法学习

1. 关于 Amcl

Amcl 的英文全称是 adaptive Monte Carlo localization，其实就是蒙特卡洛定位方法的一种升级版，使用自适应的 KLD 方法来更新粒子， 而 mcl（蒙特卡洛定位）算法使用的是粒子滤波的方法来进行定位的。而粒子滤波很粗浅的说就是一开始在地图空间很均匀的撒一把粒子，然后通过获取机器人的 motion 来移动粒子，比如机器人向前移动了一米，所有的粒子也就向前移动一米，不管现在这个粒子的位置对不对。使用每个粒子所处位置模拟一个传感器信息跟观察到的传感器信息（一般是激光）作对比，从而赋给每个粒子一个概率。之后根据生成的概率来重新生成粒子，概率越高的生成的概率越大。这样的迭代之后，所有的粒子会慢慢地收敛到一起，机器人的确切位置也就被推算出来了。 Amcl 算法步骤如下：



2. ROS 中的 Amcl

1>. 订阅的主题

scan (sensor_msgs/LaserScan)

激光雷达发布的/scan 话题。

tf (tf/tfMessage)

各个坐标系的 tf 转换 。

initialpose (geometry_msgs/PoseWithCovarianceStamped)

初始位置给定，若没有初始位置时，map->base_footprint= (0)。

map (nav_msgs/OccupancyGrid)

发布的地图话题，在实验过程中我们会用 map_server 加载地图到 ROS 空间。

2>. 发布的主题

amcl_pose (geometry_msgs/PoseWithCovarianceStamped)

发布 amcl 定位后机器人在地图中的位姿 (map->base_footprint)。

particlecloud (geometry_msgs/PoseArray)

发布 amcl 的粒子位姿。

tf (tf/tfMessage)

发布 map->base_footprint 之间的 tf 转换。

3. Amcl 参数说明

```
<param name="min_particles" value="500"/>   //允许的粒子数量的最小值，默认 100
<param name="max_particles" value="5000"/> //允许的例子数量的最大值，默认 5000
```

<param name="kld_err" value="0.05"/> //真实分布和估计分布之间的最大误差，默认 0.01

<param name="kld_z" value="0.99"/> //上标准分位数（1-p），其中 p 是估计分布上误差小于 kld_err 的概率，默认 0.99。

<param name="update_min_a" value="0.5"/> //执行滤波更新前旋转的角度，默认 pi/6 rad

<param name="resample_interval" value="1"/> //在重采样前需要的滤波更新的次数,默认 2

<param name="transform_tolerance" value="0.1"/> //tf 变换发布推迟的时间

<param name="recovery_alpha_slow" value="0.0"/> //慢速的平均权重滤波的指数衰减频率，用作决定什么时候通过增加随机位姿来 recover，默认 0（disable），可能 0.001 是一个不错的值

<param name="recovery_alpha_fast" value="0.0"/> //快速的平均权重滤波的指数衰减频率，用作决定什么时候通过增加随机位姿来 recover，默认 0（disable），可能 0.1 是个不错的值

<param name="gui_publish_rate" value="10.0"/> //扫描和路径发布到可视化软件的最大频率，设置参数为-1.0 意为失能此功能，默认-1.0

<param name="save_pose_rate" value="0.5"/> //存储上一次估计的位姿和协方差到参数服务器的最大速率。被保存的位姿将会用在连续的运动上来初始化滤波器。-1.0 失能。

<param name="use_map_topic" value="false"/> //当设置为 true 时，AMCL 将会订阅 map 话题，而不是调用服务返回地图。也就是说，当设置为 true 时，有另外一个节点实时的发布 map 话题，也就是机器人在实时的进行地图构建，并供给 amcl 话题使用；当设置为 false 时，通过 map server，也就是调用已经构建完成的地图。在 navigation 1.4.2 中新加入的参数。

<param name="first_map_only" value="false"/> //当设置为 true 时，AMCL 将仅仅使用订阅的第一个地图，而不是每次接收到新的时更新为一个新的地图。

<param name="laser_min_range" value="-1.0"/> //被考虑的最小扫描范围；参数设置为-1.0 时，将会使用激光上报的最小扫描范围

<param name="laser_max_range" value="-1.0"/> //被考虑的最大扫描范围；参数设置为-1.0 时，将会使用激光上报的最大扫描范围

<param name="laser_max_beams" value="30"/> //更新滤波器时，每次扫描中多少个等间距的光束被使用（减小计算量，测距扫描中相邻波束往往不是独立的可以减小噪声影响，太小也会造成信息量少定位不准）

HiBot 上的启动参数配置如下：

```
<node pkg="amcl" type="amcl" name="amcl" clear_params="true">
  <param name="use_map_topic" value="$(arg use_map_topic)"/>
  <!-- Publish scans from best pose at a max of 10 Hz -->
  <param name="odom_model_type" value="diff"/>
  <param name="odom_alpha5" value="0.1"/>
  <param name="gui_publish_rate" value="10.0"/>
  <param name="laser_max_beams" value="60"/>
  <param name="laser_max_range" value="12.0"/>
  <param name="min_particles" value="500"/>
  <param name="max_particles" value="2000"/>
  <param name="kld_err" value="0.05"/>
  <param name="kld_z" value="0.99"/>
  <param name="odom_alpha1" value="0.2"/>
  <param name="odom_alpha2" value="0.2"/>
```

```
  <param name="odom_alpha3" value="0.2"/>
  <param name="odom_alpha4" value="0.2"/>
  <param name="laser_z_hit" value="0.5"/>
  <param name="laser_z_short" value="0.05"/>
  <param name="laser_z_max" value="0.05"/>
  <param name="laser_z_rand" value="0.5"/>
  <param name="laser_sigma_hit" value="0.2"/>
  <param name="laser_lambda_short" value="0.1"/>
  <param name="laser_model_type" value="likelihood_field"/>
```

```
  <!-- <param name="laser_model_type" value="beam"/> -->
  <param name="laser_likelihood_max_dist" value="2.0"/>
  <param name="update_min_d" value="0.25"/>
  <param name="update_min_a" value="0.2"/>
  <param name="odom_frame_id" value="odom_combined"/>
  <param name="resample_interval" value="1"/>
  <!-- Increase tolerance because the computer can get quite busy -->
  <param name="transform_tolerance" value="1.0"/>
  <param name="recovery_alpha_slow" value="0.0"/>
  <param name="recovery_alpha_fast" value="0.0"/>
  <remap from="scan" to="$(arg scan_topic)"/>
</node>
```

4. HiBot 测试 amcl 功能包

1>. 将机器人放到上节课程新建地图的起点，打开机器人电源开关，等待大概 1 分钟后。Ubuntu 主机连接树莓派开放的 wifi。然后打开新的终端 ssh 连接到树莓派，运行 HiBot 启动节点：

```
huanyu@ubuntu:~$ ssh huike@192.168.12.1
```

```
huike@huike-desktop:~$ roslaunch huanyu_robot_start Huanyu_robot_start.launch
```

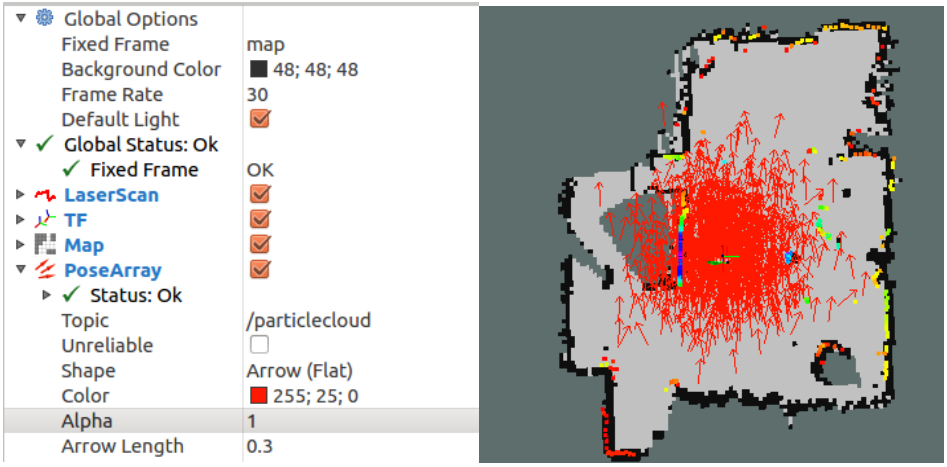
```
[ INFO] [1455208732.734492355]: [ZHOUXUEWEI] Serial Port opened
[ INFO] [1455208732.745544095]: output frame: odom_combined
[ INFO] [1455208732.746011741]: base frame: base_footprint
[ INFO] [1455208734.751010952]: Initializing Odom sensor
[ INFO] [1455208734.751690729]: Initializing Imu sensor
[ INFO] [1455208734.800748993]: Odom sensor activated
[ INFO] [1455208734.801159557]: Imu sensor activated
[ INFO] [1455208734.820802883]: Kalman filter initialized with odom measurement
```

2>. 运行 amcl 节点：运行之前确保 map_server 加载的是我们新创建的环境地图。

```
<arg name="map_file" default="$(find huanyu_robot_start)/map/map.yaml"/>
<node name="map_server_for_test" pkg="map_server" type="map_server" args="$(arg map_file)">
</node>
```

```
huike@huike-desktop:~$ roslaunch huanyu_robot_start amcl_test.launch
```

3>. 在 Ubuntu 主机上打开新的终端运行 rviz，rviz 消息和话题选择如下：就可以看到机器人初始定位状态下的粒子分布，会发现粒子（红色箭头）的分布比较散乱。



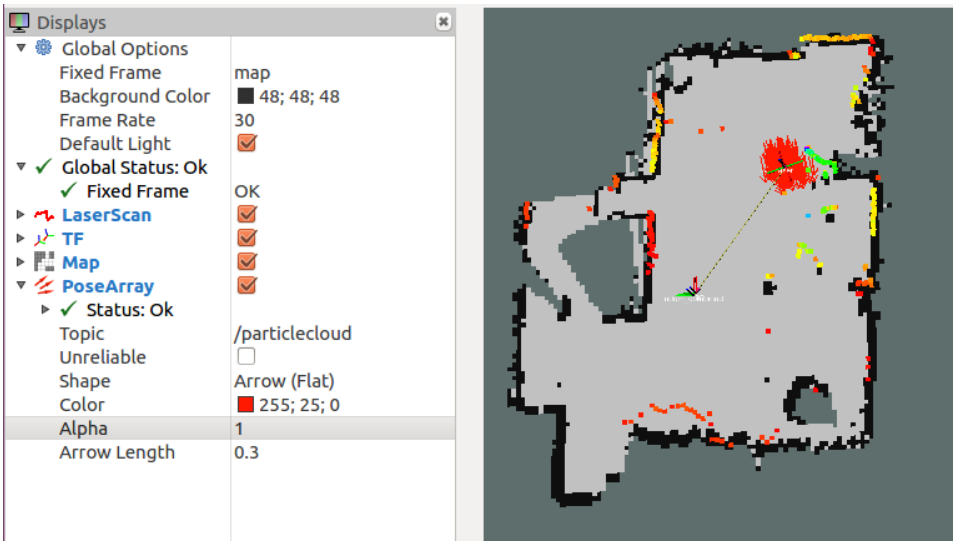
4>. 现在我们试着移动机器人，然后查看粒子的分布状态。打开新的终端 ssh 连接到树莓派，然后运行以下 launch：

```
huanyu@ubuntu:~$ ssh huike@192.168.12.1
```

```
huike@huike-desktop:~$ roslaunch turtlebot_teleop keyboard_teleop.launch
```

```
Moving around:
   u    i    o
   j    k    l
   m    ,    .
```

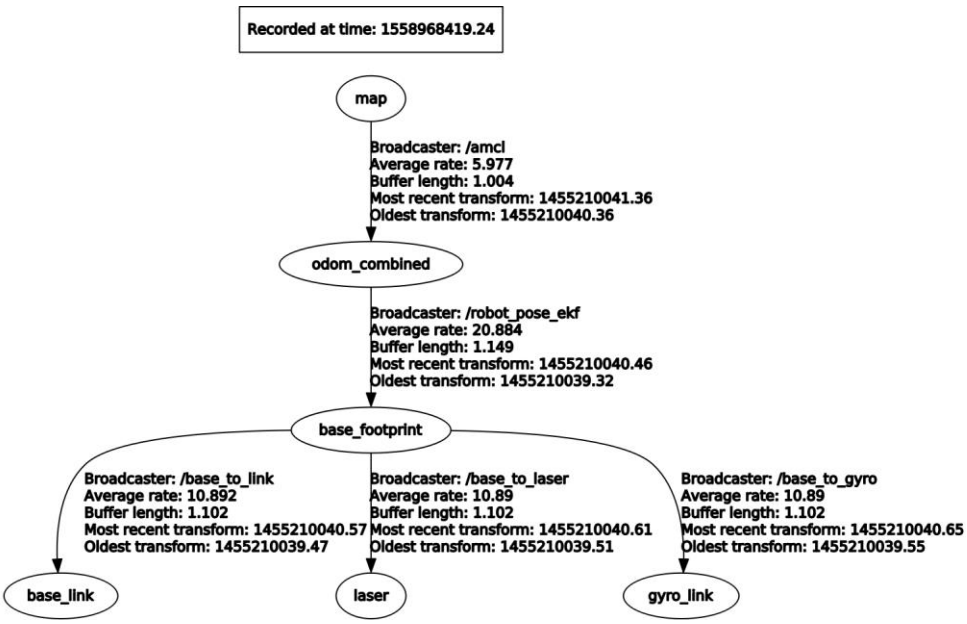
将鼠标光标放置在上面窗口内，按下键盘让机器人向前移动或者原地旋转。此刻你会发现 amcl 的粒子逐渐在收敛。



5>. 在 Ubuntu 主机上打开新的终端，运行 tf 树查看命令：

```
huanyu@ubuntu:~$ rosrun rqt_tf_tree rqt_tf_tree
```

等待以下会出现 tf 的树形结构，如下图所示。在图中我们可以发现，map->odom_combined 之间的 tf 转换是由 amcl 提供的，



4. 总结

本节课程我们学习了 amcl 算法实现原理，同时详细配置了 ROS 下 amcl 的参数。然后在 HiBot 上测试了 amcl 的定位原理和粒子收敛的过程。