RPLidar For ROS Based SLAM

赵永华

Email: yhzhao_hit@163.com

上海思岚科技有限公司/SLAMTEC

1、Introduce RPLidar

- RPLidar A1/A2 Performance and ToolKit

2、RPLidar Driver Package for ROS： rplidar_ros

- Introduce Package
- How to using it on Robot Base

3、 RPLidar for SLAM

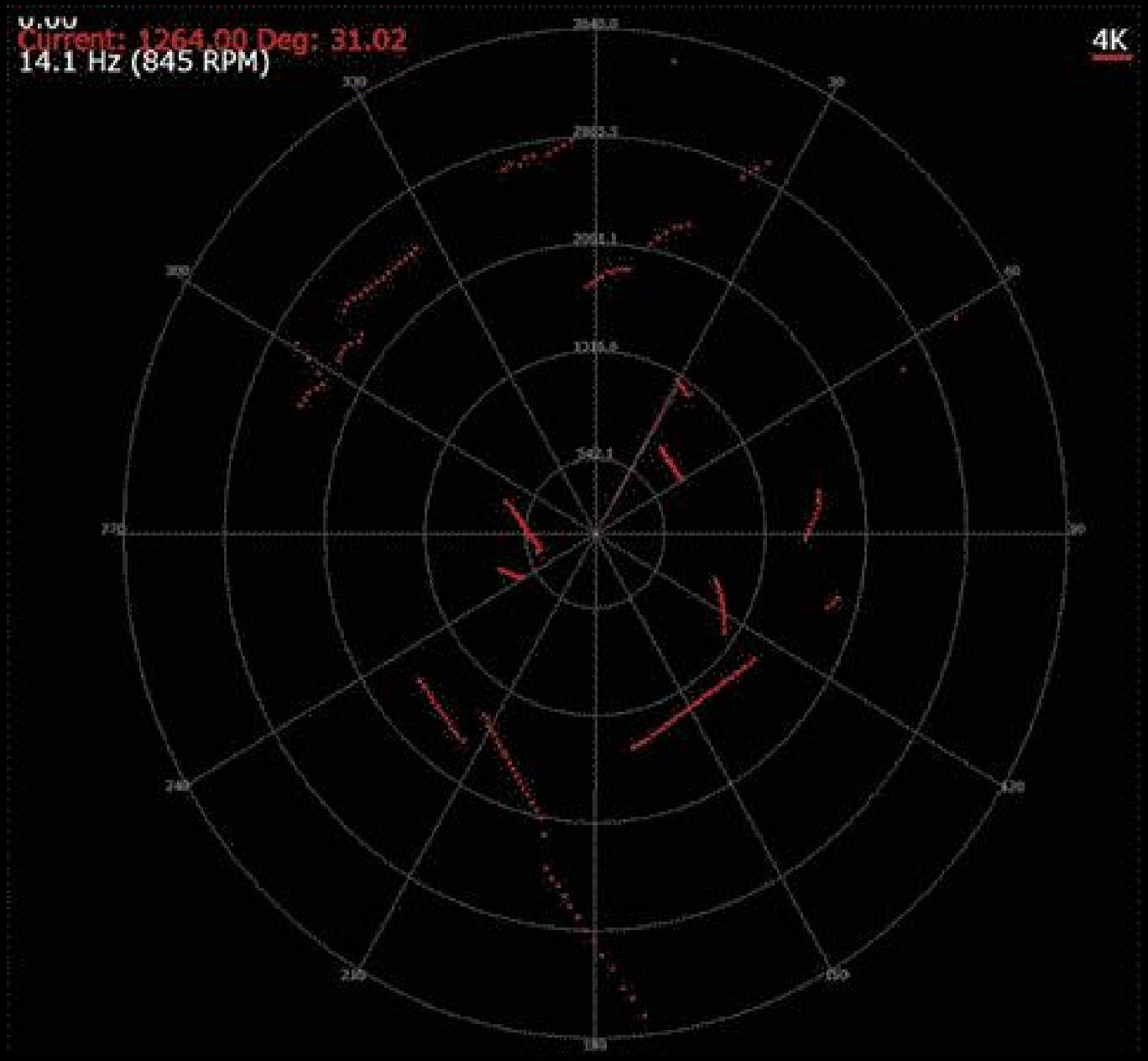
- RPLidar running SLAM: Gmapping/Hector/slam_Karto /cartographer
- Laser SLAM (2D) - ROS open Sources

4、SLAMWARE Solution for Mobile Robot

1、Introduce RPLidar

RPLIDAR A1

RPLIDAR A2


Current: 1264.00 Deg: 31.02
14.1 Hz (845 RPM)
4K

RPLidar A1

- 仅针对型号 A1M8

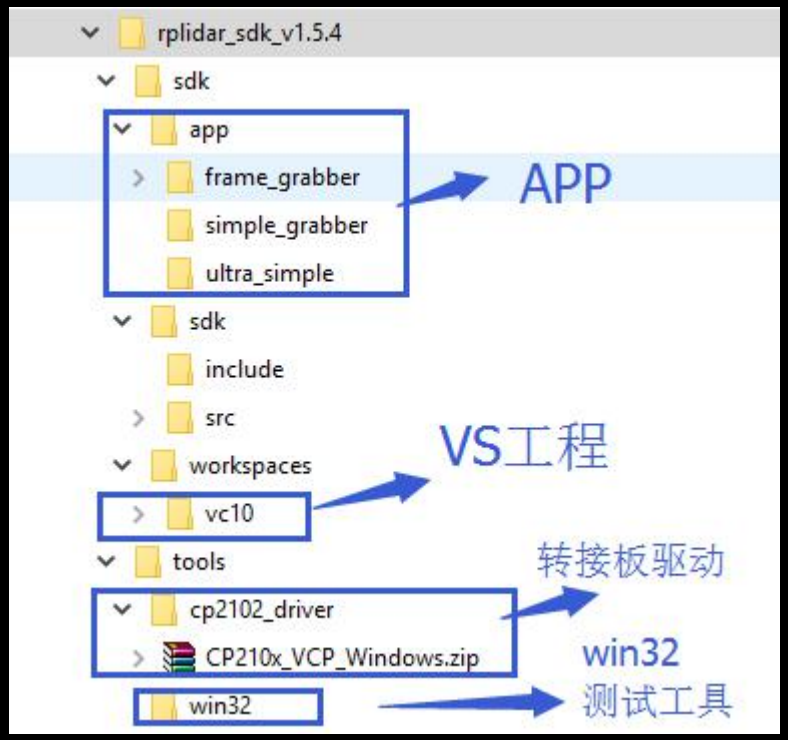
项目	单位	最小值	典型值	最大值	备注
测距范围	米(m)	待定	0.15 - 6	待定	基于白色高反光物体测得
扫描角度	度(Deg)	不适用	0-360	不适用	
测距分辨率	毫米 (mm)	不适用	<0.5	不适用	测量物体在 1.5 米以内
			<实际距离的 1%		全部量程范围内*
角度分辨率	度(Deg)	不适用	≤1	不适用	5.5hz 扫描时
单次测距时间	毫秒(ms)	不适用	0.5	不适用	
测量频率	赫兹(Hz)	不适用	≥2000	2010	
扫描频率	赫兹(Hz)	1	5.5	10	扫描 360 度的频率。典型值为一次扫描恰好 360 个采样点的情况

RPLidar A2

仅针对型号 A2M7/A2M8

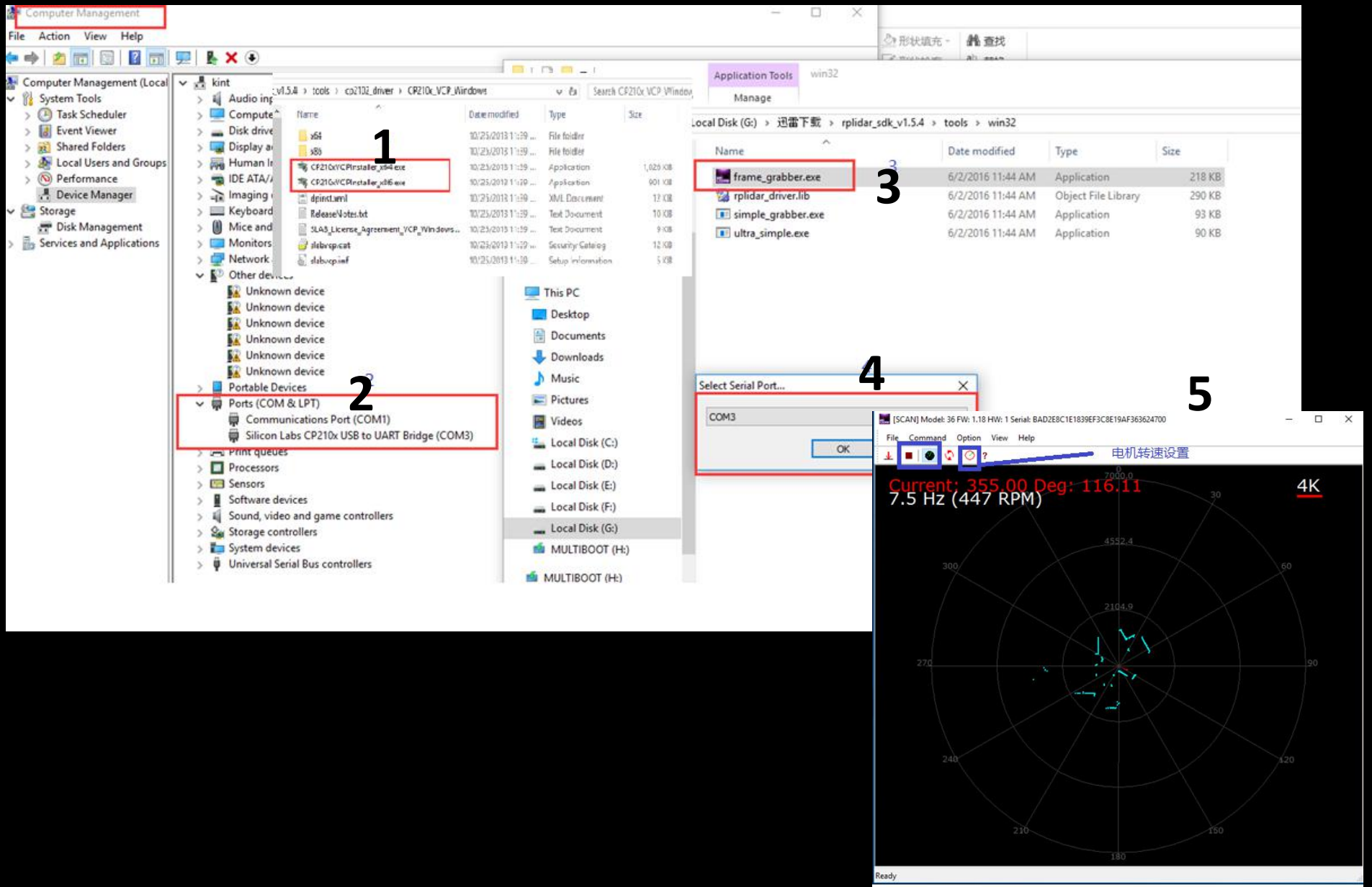
项目	单位	最小值	典型值	最大值	备注
测距范围	米(m)	0.15	-	8	基于白色 70%反射率物体
扫描角度	度 (Deg)	-	0-360	-	-
测距分辨率	毫米 (mm)	-	<0.5	-	测量物体在 1.5 米以内
			<实际距离的 1%*		全部量程范围内*
角度分辨率	度 (Deg)	0.45	0.9*	1.35	10hz 扫描时
单次测距时间	毫 秒 (ms)	-	0.25	-	-
测量频率	赫 兹 (Hz)	2000	4000	4100	-
扫描频率	赫 兹 (Hz)	5	10	15	扫描一周的频率。典型值为一次扫描恰好 400 个采样点的情况

RPLidar SDK ToolKit



RPLIDAR SDK V1.5.7

RPLidar Test Tool (win32)



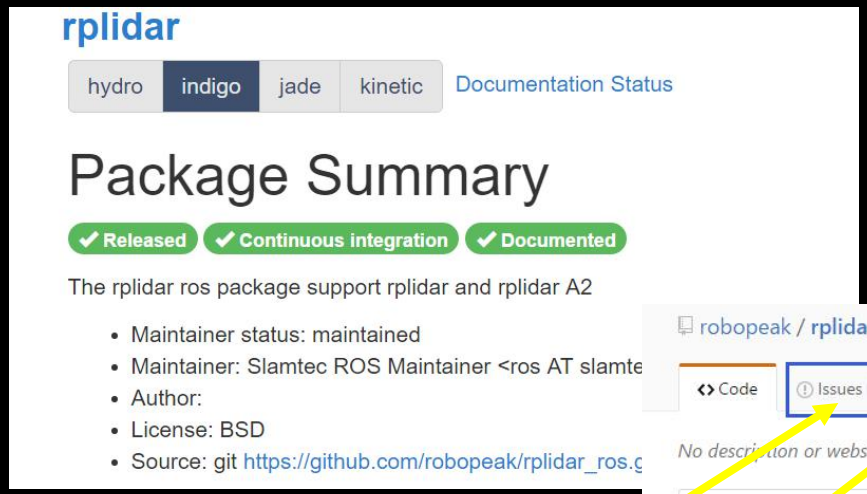
2、 RPLidar drive package for ROS： rplidar_ros

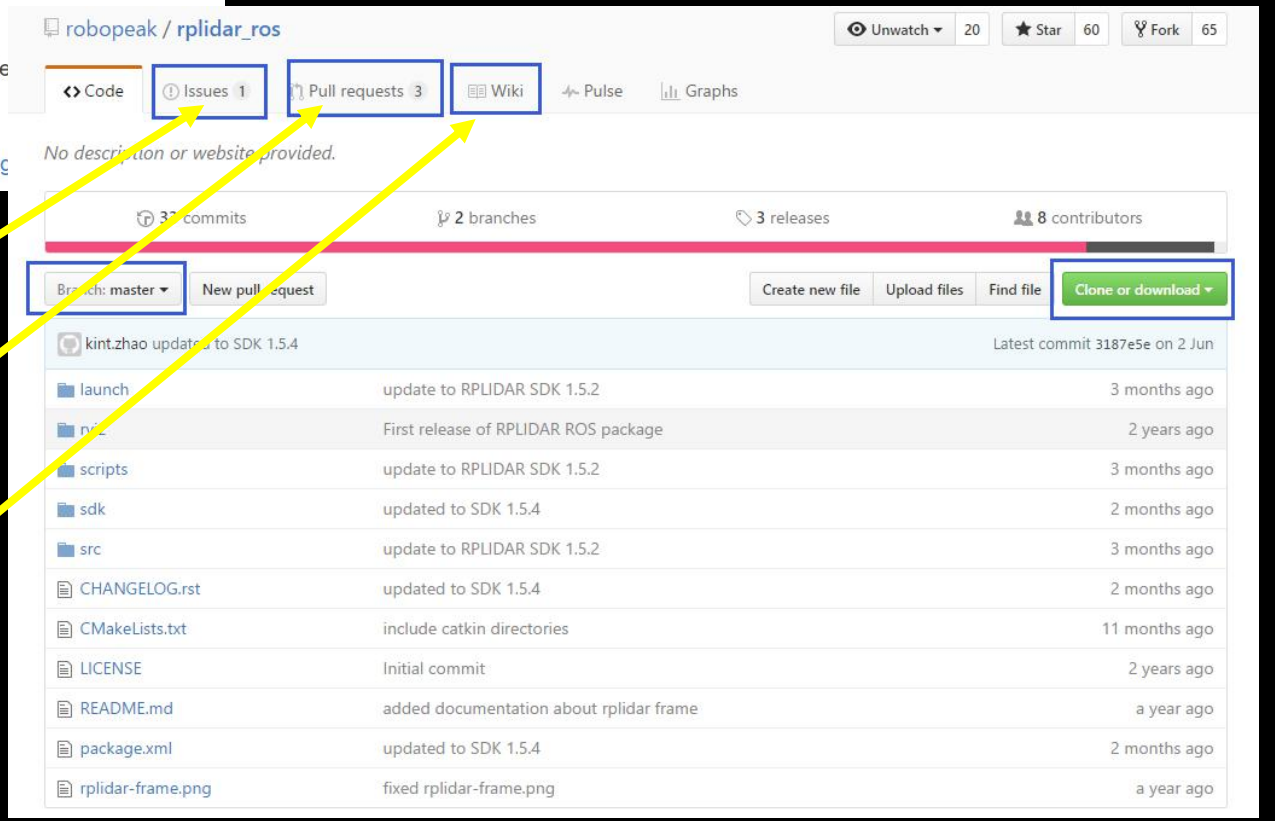
rplidar_ros ROS wiki

ros-jade-rplidar-ros

ros-xxx-rplidar-ros

rplidar_ros GitHub





Issue(closed)

Pull Request

Wiki (tutorial)

rplidar_ros

Topic:

scan (sensor_msgs/LaserScan)

Services:

stop_motor (std_srvs/Empty)

start_motor (std_srvs/Empty)

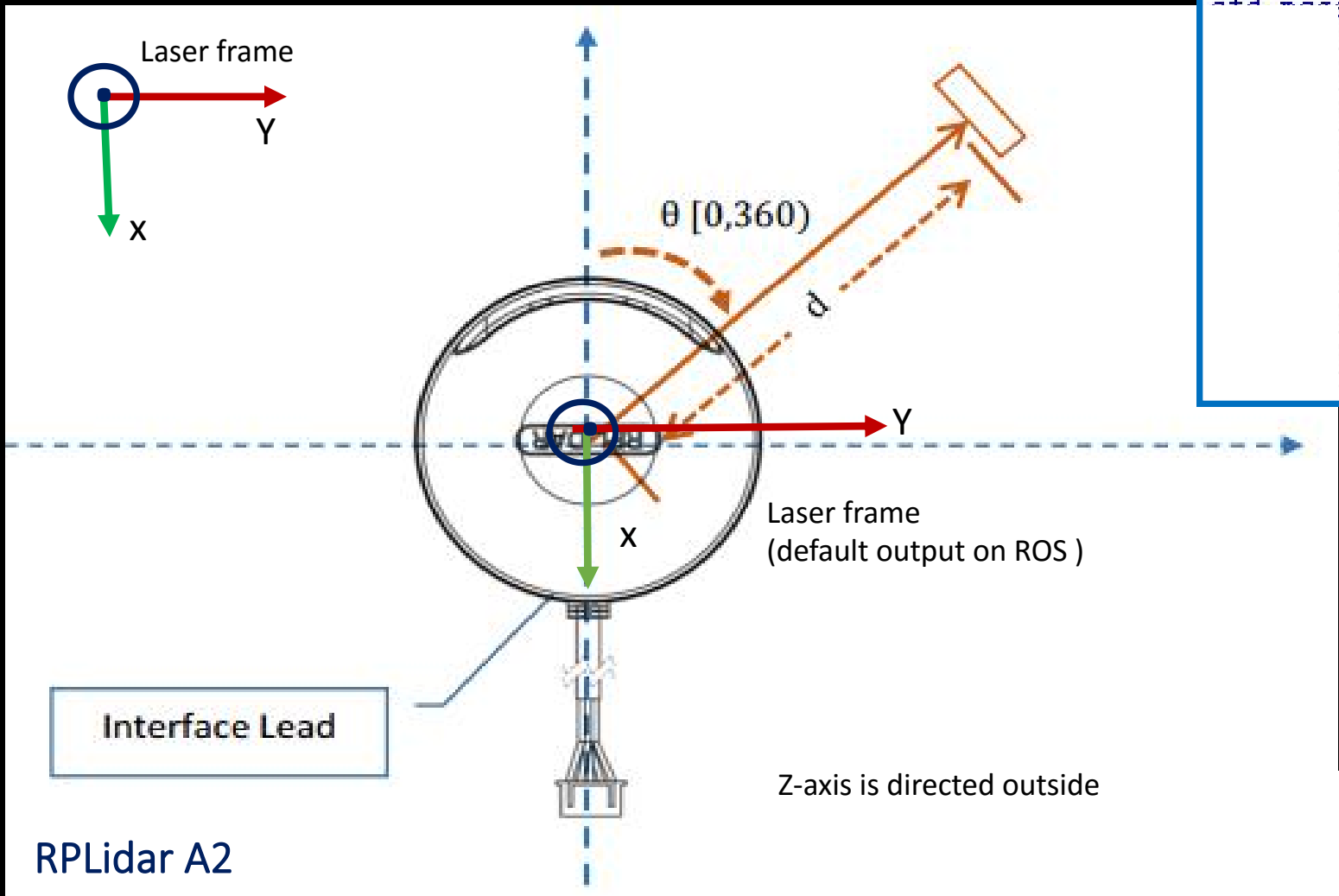
rplidar.launch

```
<launch>
  <node name="rplidarNode"          pkg="rplidar_ros"  type="rplidarNode" output="screen">
  <param name="serial_port"         type="string" value="/dev/ttyUSB0"/>
  <param name="serial_baudrate"     type="int"    value="115200"/>
  <param name="frame_id"            type="string" value="laser"/>
  <param name="inverted"            type="bool"   value="false"/>
  <param name="angle_compensate"    type="bool"   value="true"/>
  </node>
</launch>
```

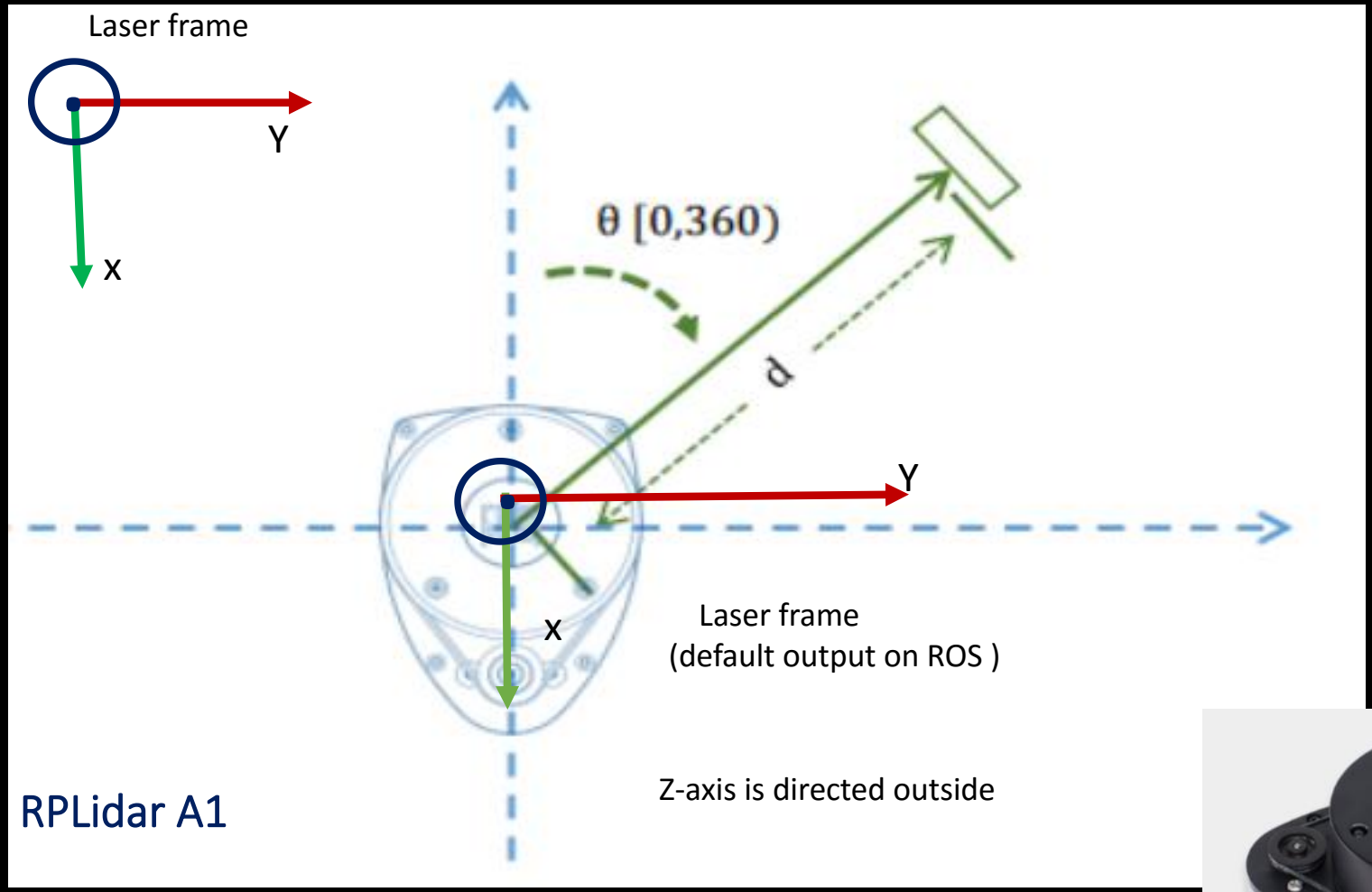
How to install RPLidar： TF

sensor_msgs/LaserScan.msg

```
std_msgs/Header header
        angle_min
        angle_max
        angle_increment
        time_increment
        scan_time
        range_min
        range_max
       [] ranges
       [] intensities
```

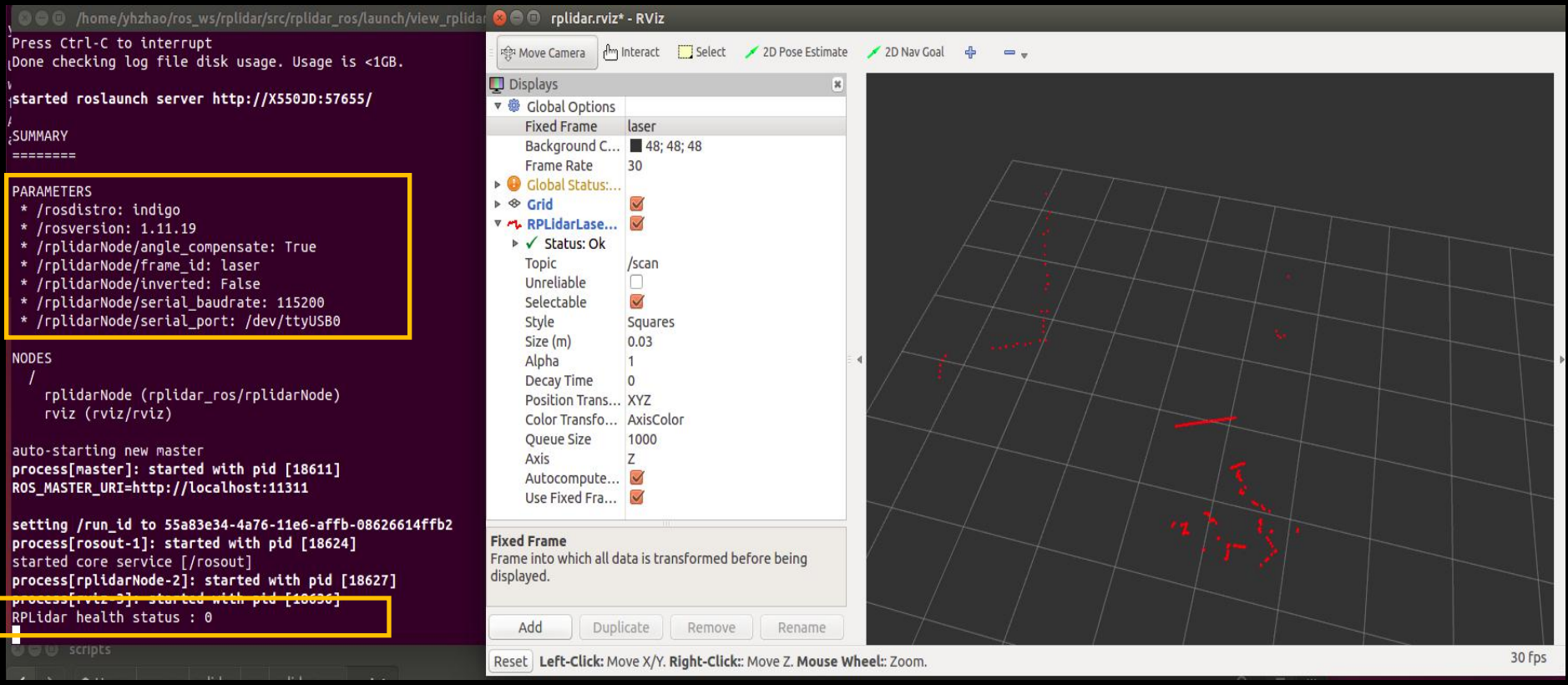


How to install RPLidar： TF



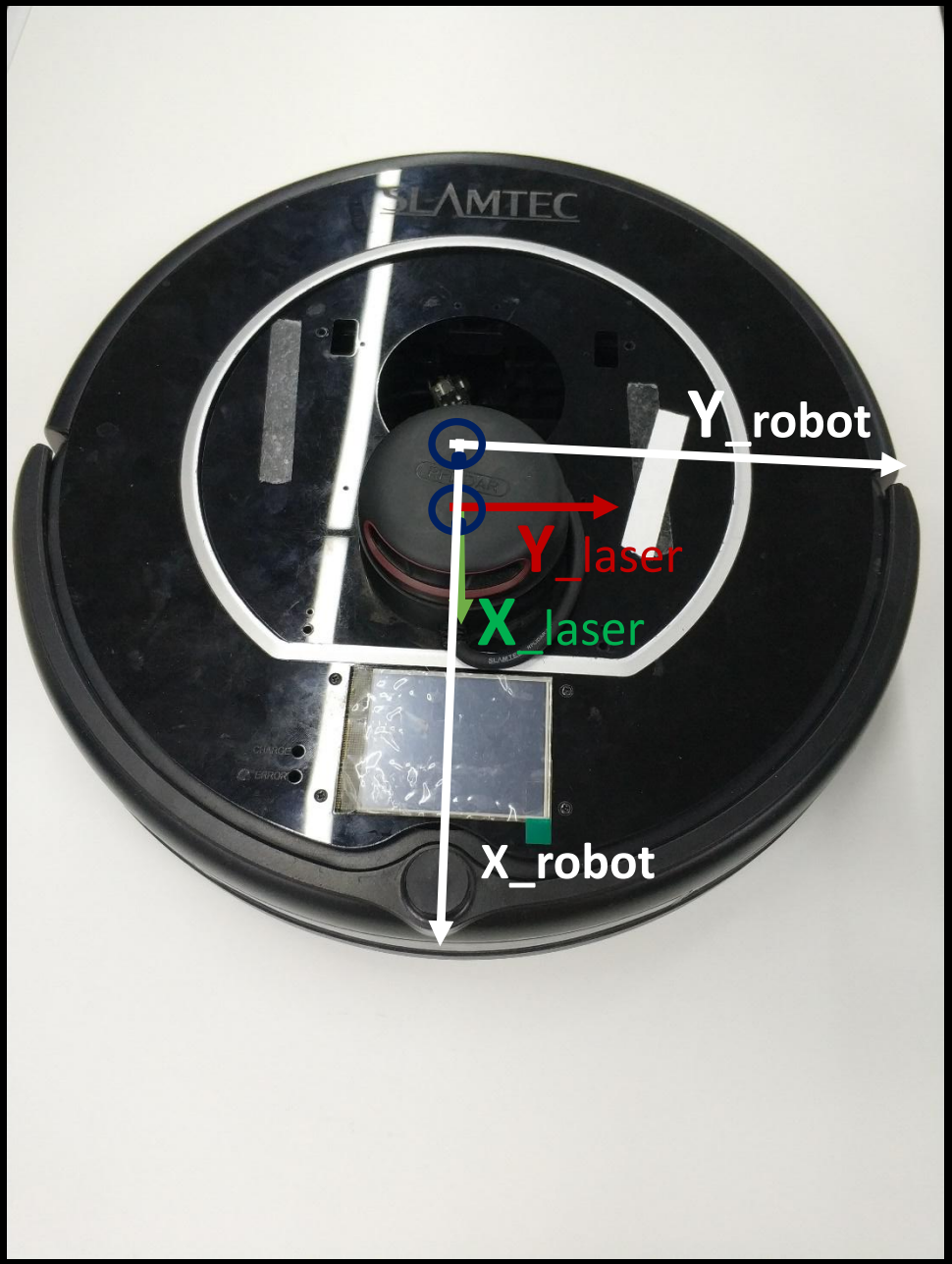
view_rplidar.launch

roslaunch rplidar_ros view_rplidar.launch



rplidar_ros start status: Health status: 0

Mobile robot platform



SDP

rplidr A2

RPLidar install on base：TF

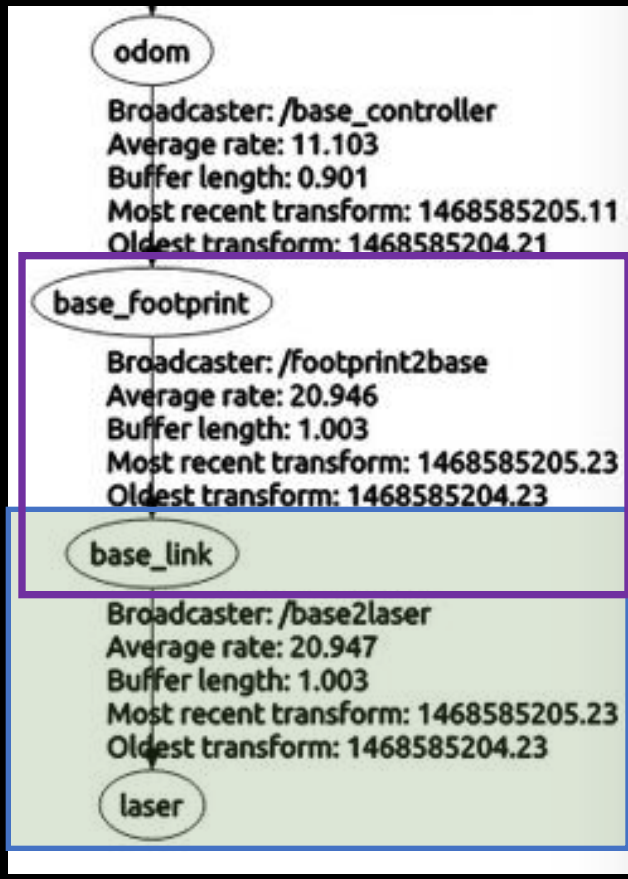
TF: RPLidar with Base

1.Setup model:URDF

URDF(user-defined reference frame **)**

2.TF publish

(static_transform_publisher)



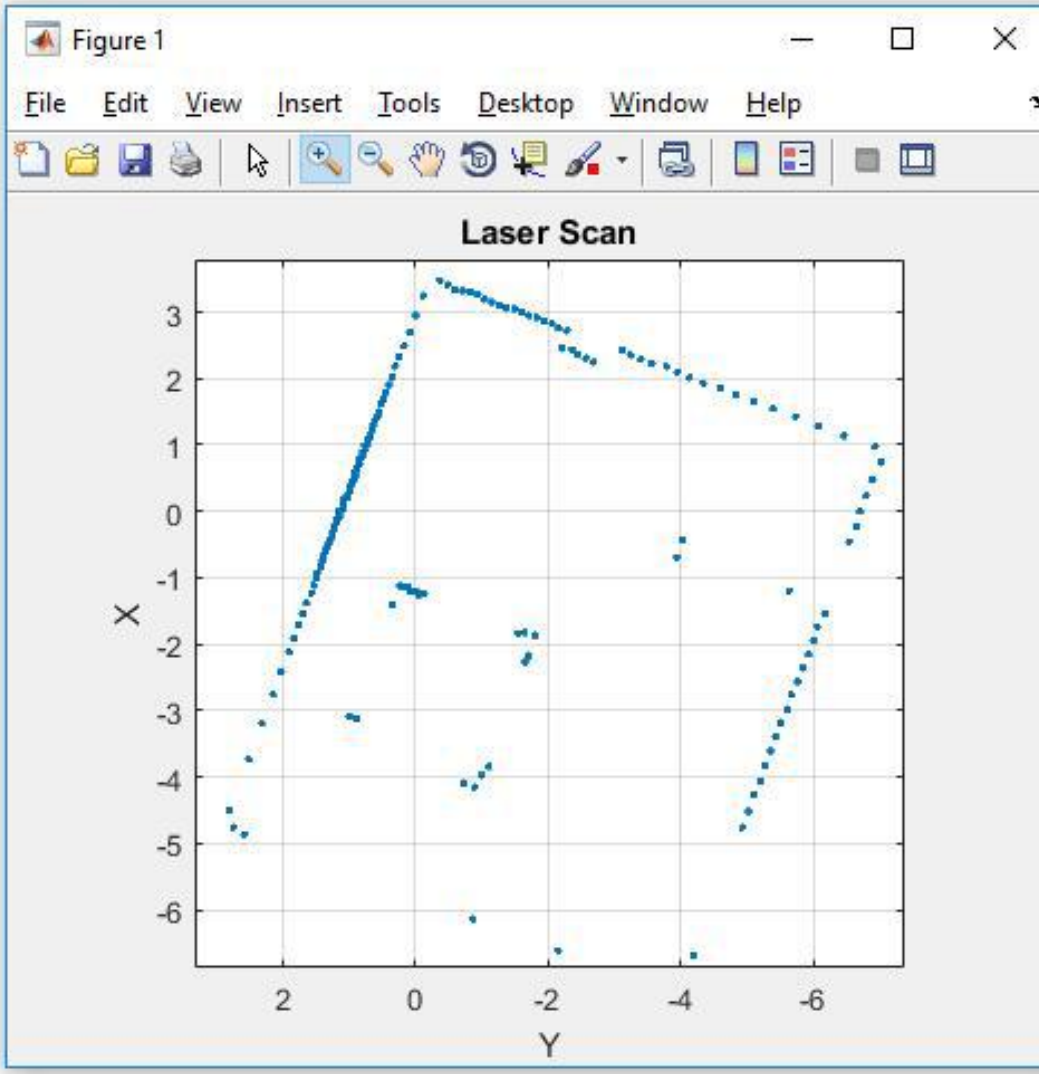

```
<node name="base2laser" pkg="tf"
type="static_transform_publisher"
args="0.07 0 0 0 0 0 1 /base_link /laser 50"/>
```

x y z qx qy qz qw frame_id child_frame_id period(milliseconds)

Launch_file 链接: http://pan.baidu.com/s/1hswfHKK 密码: r2d2

rplidar_ros on MATLAB

```
>> plot(msg)
>> rostopic list
/clicked_point
/cmd_vel
/initialpose
/move_base_simple/goal
/rosout
/rosout_agg
/slamwareNode/global_plan_path
/slamwareNode/map
/slamwareNode/map_metadata
/slamwareNode/map_updates
/slamwareNode/odom
/slamwareNode/scan
/tf
/tf_static
>> sub = rossubscriber('/slamwareNode/scan');
>> msg = receive(sub);
>> plot(msg);
>>
>>
>>
>>
>>
>>
>>
```

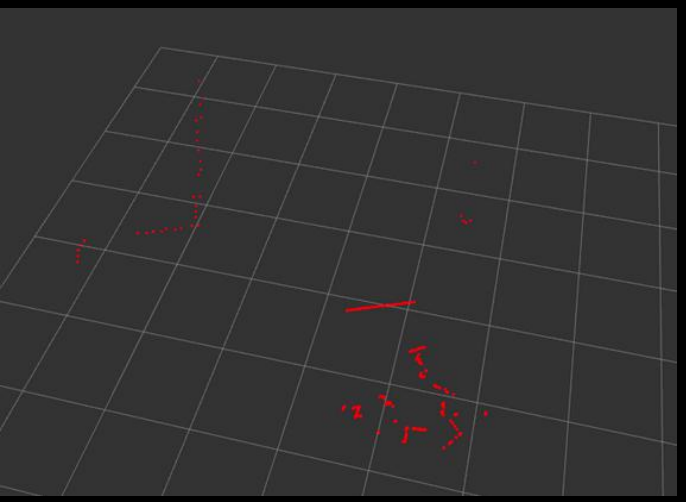


http://blog.csdn.net/zyh821351004/article/details/51124658

RPLidar For SLAM/Navigation

RVIZ : Different Display

3、RPLidar For SLAM/Navigation



Robot

rplidar_ros

odom

Navigation

robot

amcl + move_base

SLAM

robot

gmapping

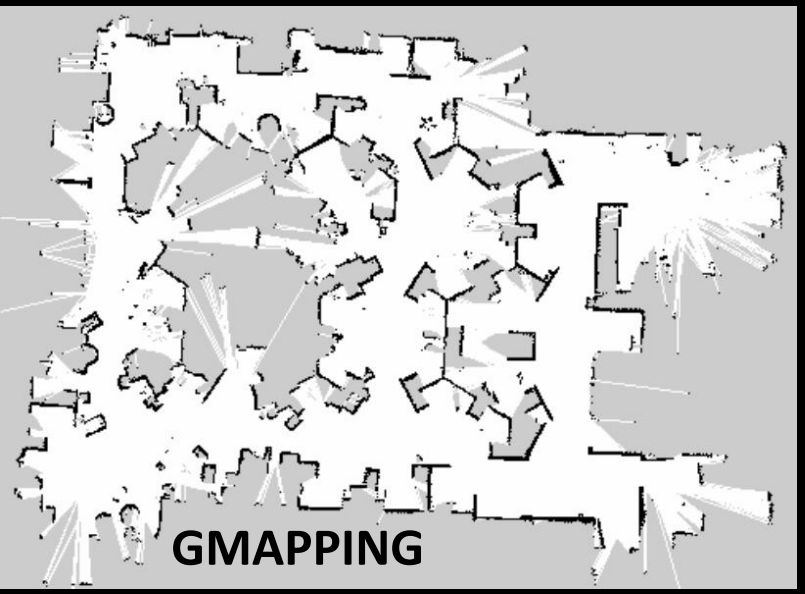
Explore

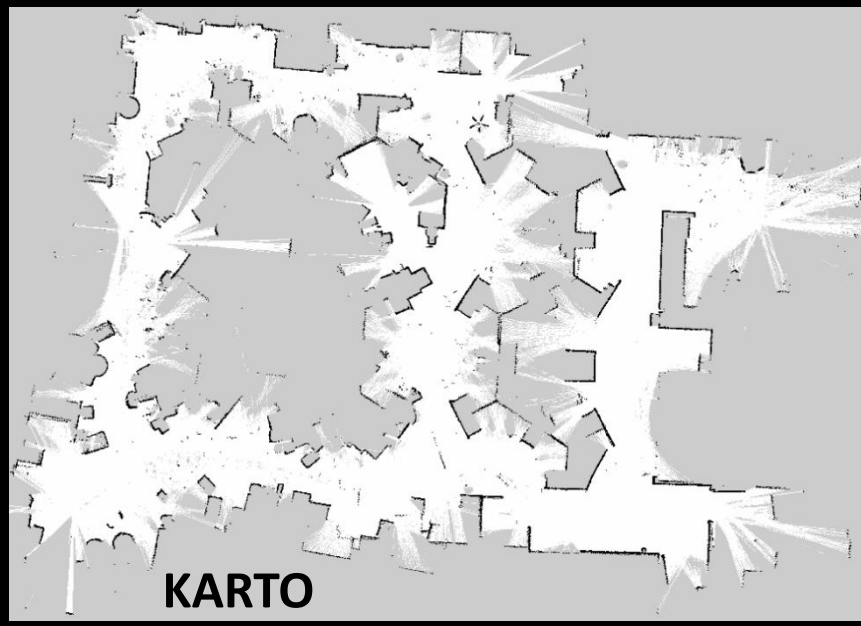
Robot + gmapping + move_base
goal(map analysis)

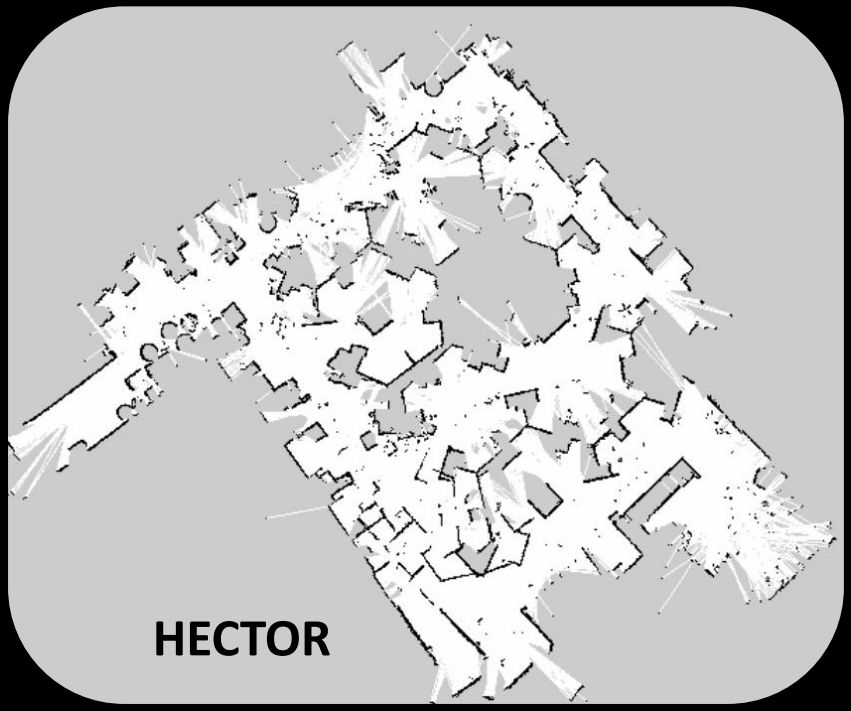
Laser SLAM (2D) - ROS open Sources

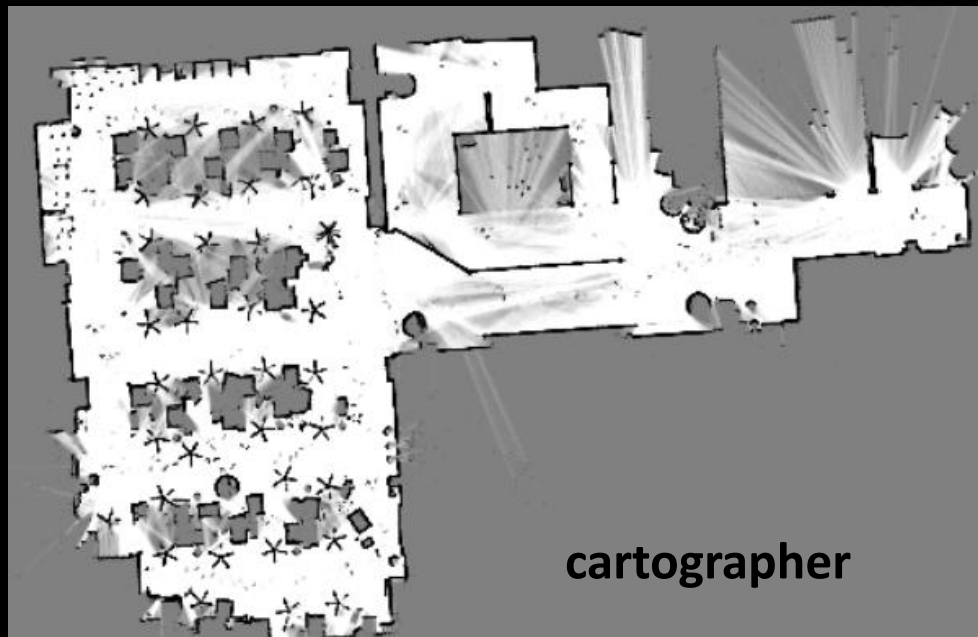
Gmapping	ros-perception/slam_**gmapping** ros-perception/openslam_**gmapping**
Hector	tu-darmstadt-ros-pkg **hector_slam**
Karto	ros-perception **slam_karto** ros-perception **open_karto** skasperski **navigation_2d**
Cartographer	googlecartographer **cartographer** googlecartographer **cartographer_ros**

maps







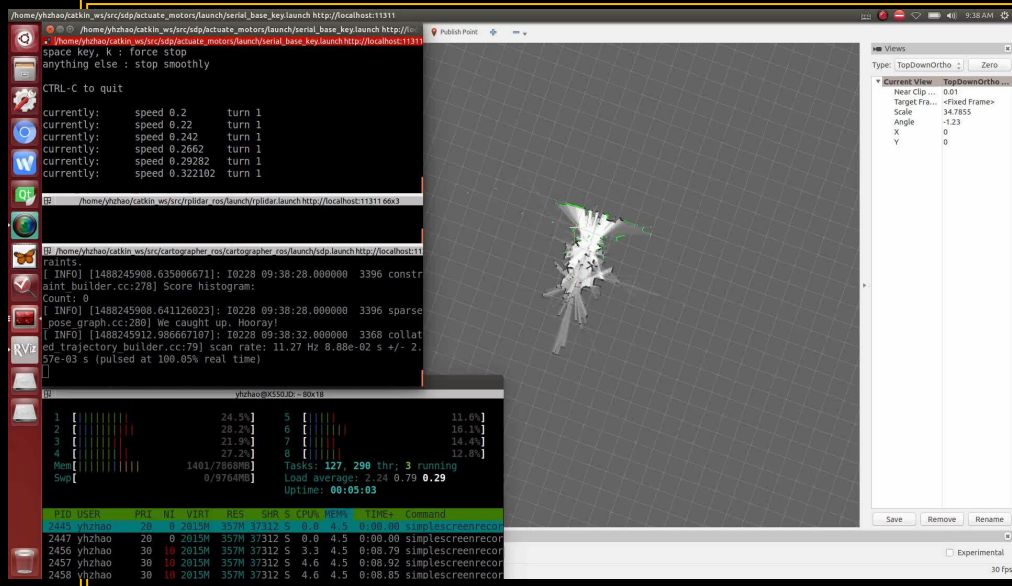


Running SLAM(videos) : Gmapping Hector karto cartographer

Gmapping

Hector

Karto



cartographer

链接: https://pan.baidu.com/s/1boNxWFt 密码: iuwh

Laser SLAM - ROS open Sources

- Framework method：
  - Filter：

    KF structure： ekf/ukf　　feature

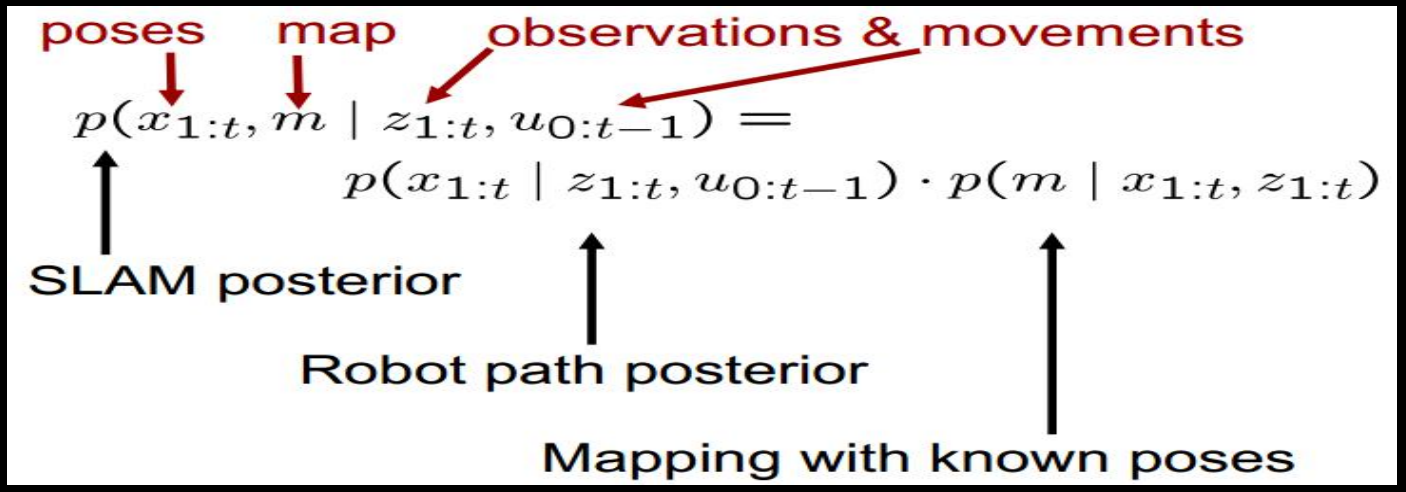
    Particle Filter: rbpf　　***gmapping***

    ScanMatcher:　　***hector***

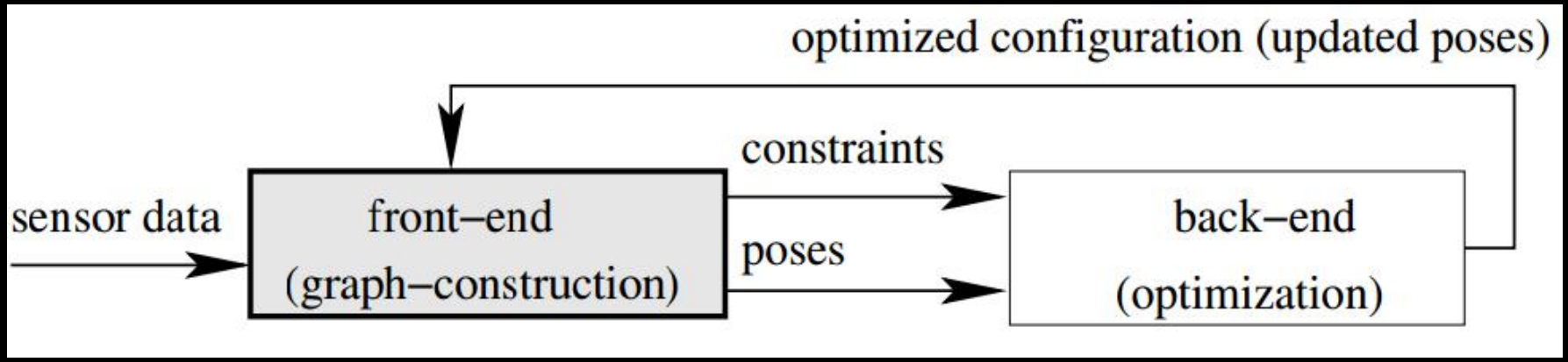
  - Graph optimizer: ***karto*** ***cartographer***
- Scan Matcher： correct trajectory
- Loop Closure： global accumulator error
- Gridmap: increase map / rebuild map / submap

Framework method

Rao-Blackwellized Particle Filters(RBPF)



Graph optimizer



Scan Matcher

Non-linear optimization

Scan matcher(scan-map)

$$x_t^* = \operatorname*{argmax}_{x_t} p(z_t \mid x_t, m_{t-1}) \cdot p(x_t \mid x_{t-1}^*, u_{t-1})$$

1 Simple Gradient Descent / ICP Gmapping

2 Guass-Newton (multi-resolution map) Hector

3 Real Time Correlative Scan Matcher karto / cartographer

Multi-resolution window

4 Fast Correlative Scan Matcher cartographer

Resolution + Branch And Bound → Loop detector

Scan matcher (evaluation)

Total score: $s(x, z, m) = \sum_i s(x, z^i, m)$

Cell score: $s(x, z^i, m) = e^{d^2/\sigma}$

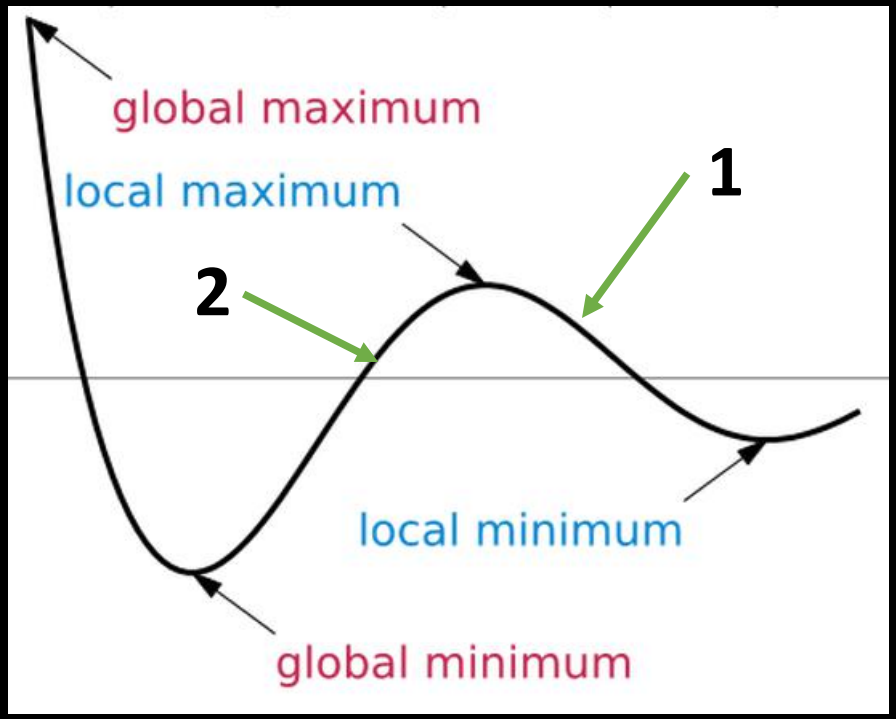
laser endPoints → map frame $\hat{z}^i = x \oplus z^i$

mapCell (accumulator) **cell.mean** $(x, y)^T$

$$d^2 = (\hat{z}^i - (x, y)^T))^T \cdot (\hat{z}^i - (x, y)^T))$$

Score(m,p,r) / likelihoodAndScore(w,m,p,r)

Decrease global error

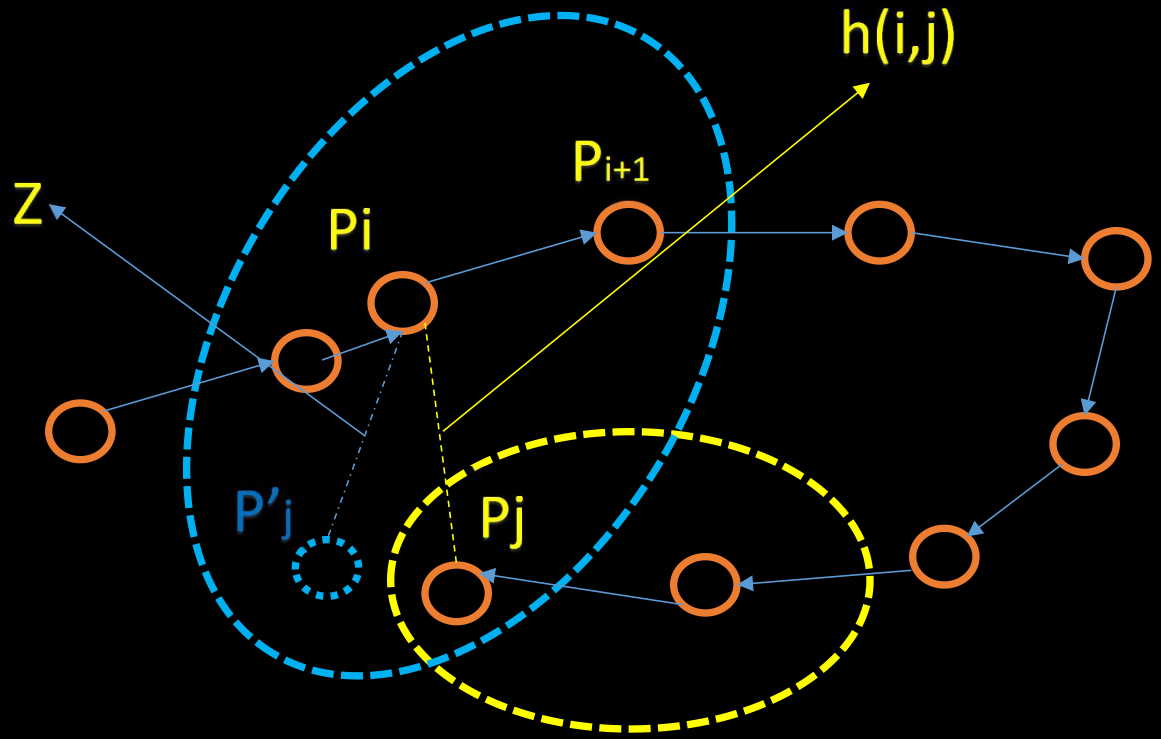


PF(gmapping)

- Sample
  - motion model
- Weight update
  - likelihood score
- Adaptive resampling
  - effective sample size

Decrease global error

Create Graph and optimizer (karto)



Robot pose:

$$c_i = \left[t_i^T, \theta_i \right]^T = \left[x_i, y_i, \theta_i \right]^T$$

Offset of c_i and c_j :
（ Constraint ）

$$h\left(c_i, c_j\right) = \begin{bmatrix} R_i^T \left(t_j - t_i \right) \\ \theta_j - \theta_i \end{bmatrix}$$

$$e_{ij} \equiv z_{ij} - h\left(c_i, c_j\right)$$

$$F\left(c, e\right) = \sum_{i,j \in E} e_{ij}^T \Lambda_{ij} e_{ij}$$

Sparse Pose Adjustment / g2o / ceres

eliminate global error

Occupancy Grid Map update

1. Insert scan data to create gridmap: **based frequancy**

$$p(m_t^{x,y}) = \frac{b_t^{x,y}}{v_t^{x,y}}$$

$$m_t^{x,y} = (b_t^{x,y}, v_t^{x,y}) = \begin{cases} (b_t^{x,y}+1, v_t^{x,y}+1) & \text{if occupied} \\ (b_t^{x,y}, v_t^{x,y}+1) & \text{if free} \end{cases}$$

2. keep increased building / rebuilding / submap

3. **nav_msgs/OccupancyGrid.msg**

std_msgs/Header	header
nav_msgs/MapMetaData	info
int8[]	data

range [0,100]. Unknown is -1

nav_msgs/MapMetaData.msg

time	map_load_time
float32	resolution
uint32	width
uint32	height
geometry_msgs/Pose	origin

Configure Launch file for RPLidar A2 to SLAM

Gmapping + Hector + Slam_karto

Configure

Topic: /scan

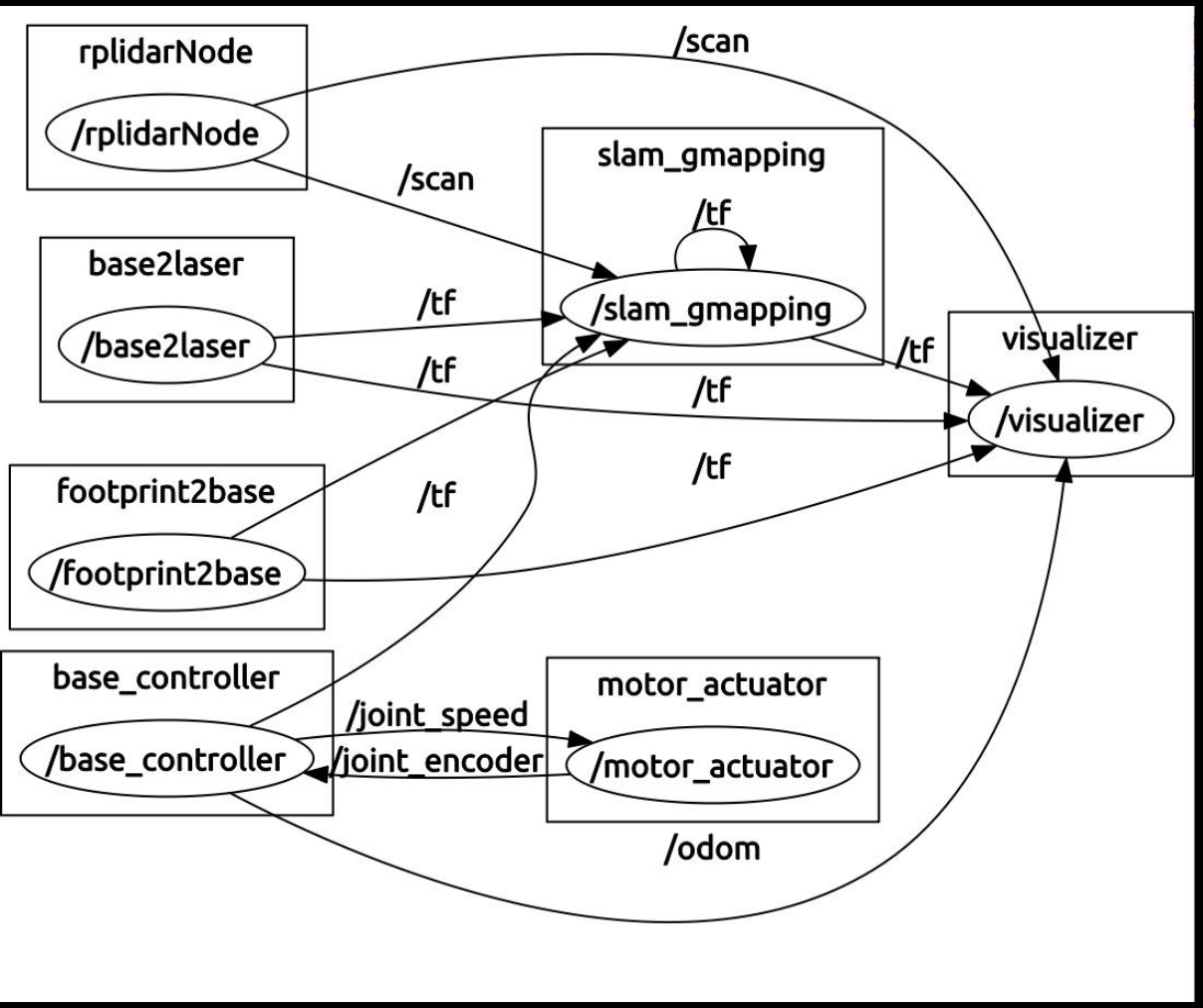
TF （ frame_ID 、 transform）: base->laser

Configure param: max_range(8.0), min_range(0.15)

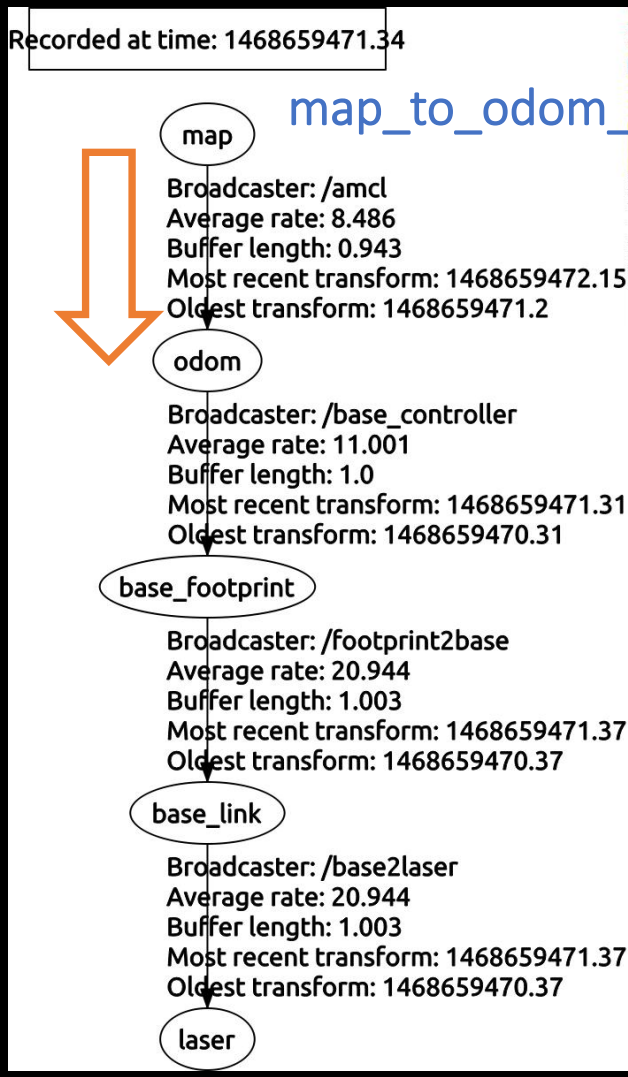
Check

```
rosrun tf view_frames
rqt_graph
rostopic  list
rqt
```

rqt_graph


rplidarNode
/rplidarNode
/scan
/scan
slam_gmapping
/tf
/slam_gmapping
base2laser
/base2laser
/tf
/tf
/tf
visualizer
/visualizer
/tf
/tf
footprint2base
/footprint2base
base_controller
/base_controller
/joint_speed
/joint_encoder
motor_actuator
/motor_actuator
/odom

TF


Recorded at time: 1468659471.34
map_to_odom_
map
Broadcaster: /amcl
Average rate: 8.486
Buffer length: 0.943
Most recent transform: 1468659472.15
Oldest transform: 1468659471.2
odom
Broadcaster: /base_controller
Average rate: 11.001
Buffer length: 1.0
Most recent transform: 1468659471.31
Oldest transform: 1468659470.31
base_footprint
Broadcaster: /footprint2base
Average rate: 20.944
Buffer length: 1.003
Most recent transform: 1468659471.37
Oldest transform: 1468659470.37
base_link
Broadcaster: /base2laser
Average rate: 20.944
Buffer length: 1.003
Most recent transform: 1468659471.37
Oldest transform: 1468659470.37
laser

```
<launch>
    <param name="use_sim_time" value="false"/>
    <node pkg="gmapping" type="slam_gmapping"
name="slam_gmapping" output="screen">
      <!--remap from="scan" to="base_scan"/-->
      <param name="map_update_interval" value="5.0"/>
      <param name="maxUrange" value="6.0"/>
      <param name="sigma" value="0.05"/>
      <param name="kernelSize" value="1"/>
      <param name="lstep" value="0.05"/>
      <param name="astep" value="0.05"/>
      <param name="iterations" value="5"/>
      <param name="lsigma" value="0.075"/>
      <param name="ogain" value="3.0"/>
      <param name="lskip" value="0"/>
      <param name="minimumScore" value="50"/>
      <param name="srr" value="0.1"/>
      <param name="srt" value="0.2"/>
      <param name="str" value="0.1"/>
      <param name="stt" value="0.2"/>
      <param name="linearUpdate" value="1.0"/>
      <param name="angularUpdate" value="0.5"/>
      <param name="temporalUpdate" value="3.0"/>
      <param name="resampleThreshold" value="0.5"/>
      <param name="particles" value="30"/>
      <param name="xmin" value="-5.0"/>
      <param name="ymin" value="-5.0"/>
      <param name="xmax" value="5.0"/>
      <param name="ymax" value="5.0"/>
      <param name="delta" value="0.05"/>
      <param name="llsamplerange" value="0.01"/>
      <param name="llsamplestep" value="0.01"/>
      <param name="lasamplerange" value="0.005"/>
      <param name="lasamplestep" value="0.005"/>
    </node>
    <node name="visualizer" pkg="rviz" type="rviz" args="-d
$(find sdp_navigation)/rviz/navigation.rviz"/>
</launch>
```

Gmapping launch

Hector_mapping launch

```
<!-- Author:  Kint Zhao @SLAMTEC      Jan.29.2016     -->
<launch>
  <arg name="tf_map_scanmatch_transform_frame_name"
default="odom laser"/>
  <arg name="base_frame" default="base_link"/>
  <arg name="odom_frame" default="odom"/>
  <arg name="pub_map_odom_transform" default="true"/>
  <arg name="scan_subscriber_queue_size" default="5"/>
  <arg name="scan_topic" default="scan"/>
  <arg name="map_size" default="800"/>

  <node pkg="hector_mapping" type="hector_mapping"
name="hector_mapping" output="screen">

    <!-- Frame names -->
    <param name="map_frame" value="map" />
    <param name="base_frame" value="$(arg base_frame)" />
    <param name="odom_frame" value="$(arg odom_frame)" />

    <!-- Tf use -->
    <param name="use_tf_scan_transformation" value="true"/>
    <param name="use_tf_pose_start_estimate" value="false"/>
    <param name="pub_map_odom_transform" value="$(arg
pub_map_odom_transform)"/>

    <!-- Map size / start point -->
    <param name="map_resolution" value="0.050"/>
    <param name="map_size" value="$(arg map_size)"/>
    <param name="map_start_x" value="0.5"/>
    <param name="map_start_y" value="0.5" />
    <param name="map_multi_res_levels" value="2" />
```

```
    <!-- Map update parameters -->
    <param name="update_factor_free" value="0.4"/>
    <param name="update_factor_occupied" value="0.9" />
    <param name="map_update_distance_thresh" value="0.4"/>
    <param name="map_update_angle_thresh" value="0.06" />
    <param name="laser_z_min_value" value = "-1.0" />
    <param name="laser_z_max_value" value = "1.0" />

    <param name="laser_max_dist" value = "5.8" />
    <param name="laser_min_dist" value = "0.15" />

    <!-- Advertising config -->
    <param name="advertise_map_service" value="true"/>

    <param name="scan_subscriber_queue_size" value="$(arg
scan_subscriber_queue_size)"/>
    <param name="scan_topic" value="$(arg scan_topic)"/>

    <!-- Debug parameters -->
    <!--
      <param name="output_timing" value="false"/>
      <param name="pub_drawings" value="true"/>
      <param name="pub_debug_output" value="true"/>
    -->
    <param name="tf_map_scanmatch_transform_frame_name"
value="$(arg tf_map_scanmatch_transform_frame_name)" />
  </node>

<node name="visualizer" pkg="rviz" type="rviz" args="-d $(find
sdp_navigation)/rviz/navigation.rviz"/>
</launch>
```

Configure param of slam_karto

```
<launch>
  <node pkg="slam_karto" type="slam_karto" name="slam_karto"
output="screen">
    <remap from="scan" to="scan"/>
    <param name="odom_frame" value="odom"/>
    <param name="map_update_interval" value="25"/>
    <param name="resolution" value="0.025"/>
    <rosparam command="load" file="$(find
sdp_navigation)/param/karto_mapper_params.yaml" />
  </node>

<node name="visualizer" pkg="rviz" type="rviz" args="-d $(find
sdp_navigation)/rviz/navigation.rviz"/>
</launch>
```

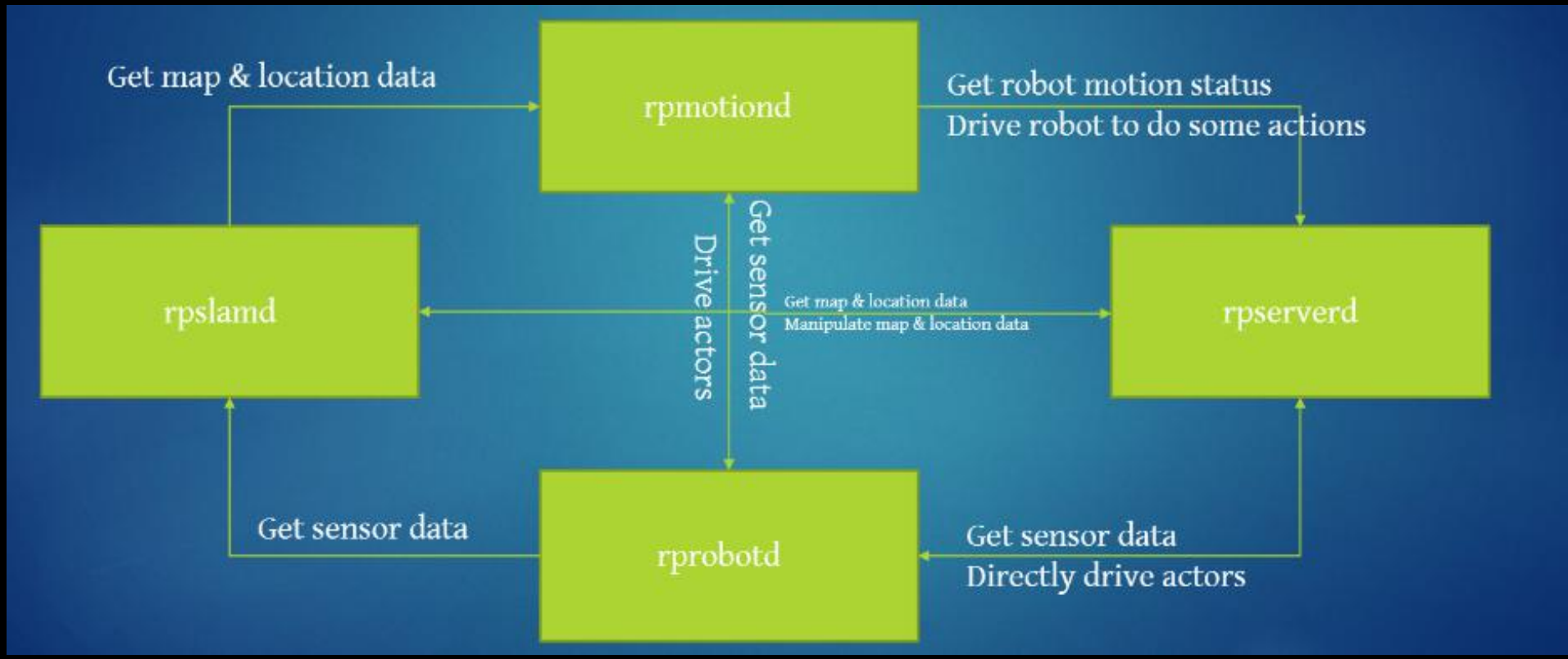
```
use_scan_matching: true
use_scan_barycenter: true
minimum_travel_distance: 0.3
minimum_travel_heading: 0.4  # 0.2
scan_buffer_size: 67
scan_buffer_maximum_scan_distance: 20.0
link_match_minimum_response_fine: 0.6
link_scan_maximum_distance: 4
correlation_search_space_dimension: 2
correlation_search_space_resolution: 0.05
correlation_search_space_smear_deviation: 0.03

do_loop_closing: true
loop_match_minimum_chain_size: 5
loop_match_maximum_variance_coarse: 0.4
loop_match_minimum_response_coarse: 0.4
loop_match_minimum_response_fine: 0.6
loop_search_space_dimension: 10  # 2.8
loop_search_space_resolution: 0.1
loop_search_space_smear_deviation: 0.03
loop_search_maximum_distance: 4.0

distance_variance_penalty: 0.3
angle_variance_penalty: 0.35
fine_search_angle_offset: 0.00349
coarse_search_angle_offset: 0.349
coarse_angle_resolution: 0.0349
minimum_angle_penalty: 0.9
minimum_distance_penalty: 0.5
use_response_expansion: false
```

Configure param of cartographer

```
<launch>
  <!--param name="/use_sim_time" value="true" /-->

  <node name="cartographer_node" pkg="cartographer_ros"
      type="cartographer_node" args="
          -configuration_directory $(find cartographer_ros)/configuration_files
          -configuration_basename sdp.lua"
      output="screen">
    <!--remap from="scan" to="horizontal_laser_2d" /-->
  </node>

  <node name="rviz" pkg="rviz" type="rviz" required="true"
      args="-d $(find cartographer_ros)/configuration_files/demo_2d.rviz" />

</launch>
```

```
include "map_builder.lua"

options = {
  map_builder = MAP_BUILDER,
  map_frame = "map",
  tracking_frame = "laser",
  published_frame = "laser",
  odom_frame = "odom",
  provide_odom_frame = true,
  use_odometry = false,
  use_laser_scan = true,
  use_multi_echo_laser_scan = false,
  num_point_clouds = 0,
  lookup_transform_timeout_sec = 0.2,
  submap_publish_period_sec = 0.3,
  pose_publish_period_sec = 5e-3,
}

MAP_BUILDER.use_trajectory_builder_2d = true

TRAJECTORY_BUILDER_2D.laser_min_range = 0.15
TRAJECTORY_BUILDER_2D.laser_max_range = 8.
TRAJECTORY_BUILDER_2D.laser_missing_echo_ray_length = 1.
TRAJECTORY_BUILDER_2D.use_imu_data = false
TRAJECTORY_BUILDER_2D.use_online_correlative_scan_matching = true

SPARSE_POSE_GRAPH.optimization_problem.huber_scale = 1e2

return options
```

4. SLAMWARE solution for localization and navigation



4. SLAMWARE solution for localization and navigation

Topic:

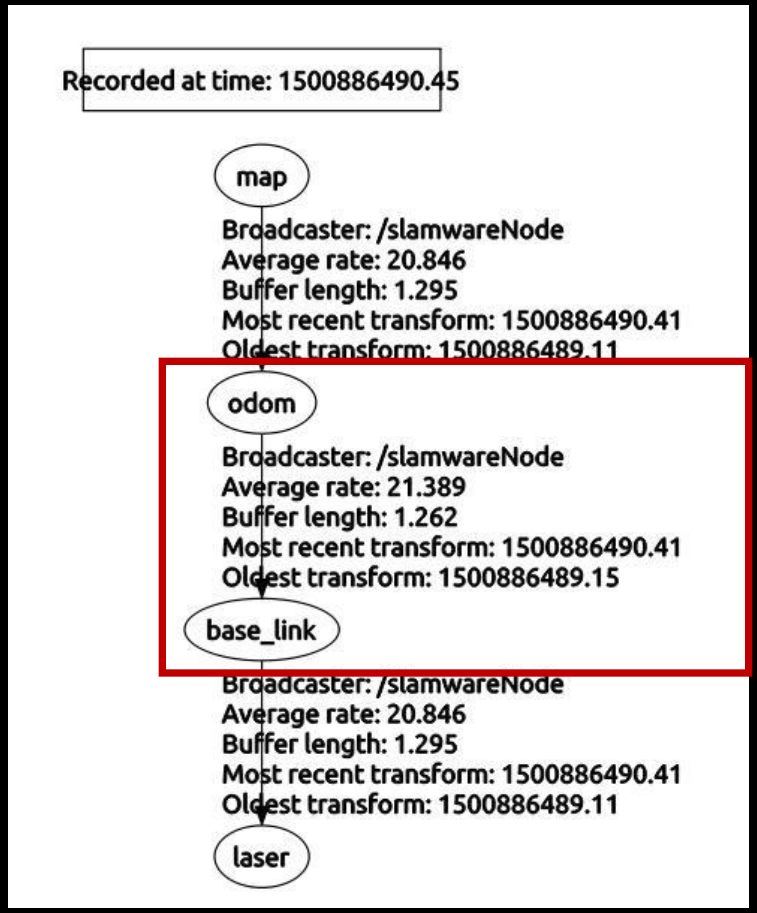
Odom(50hz)

Scan(10hz)

Map(2hz)

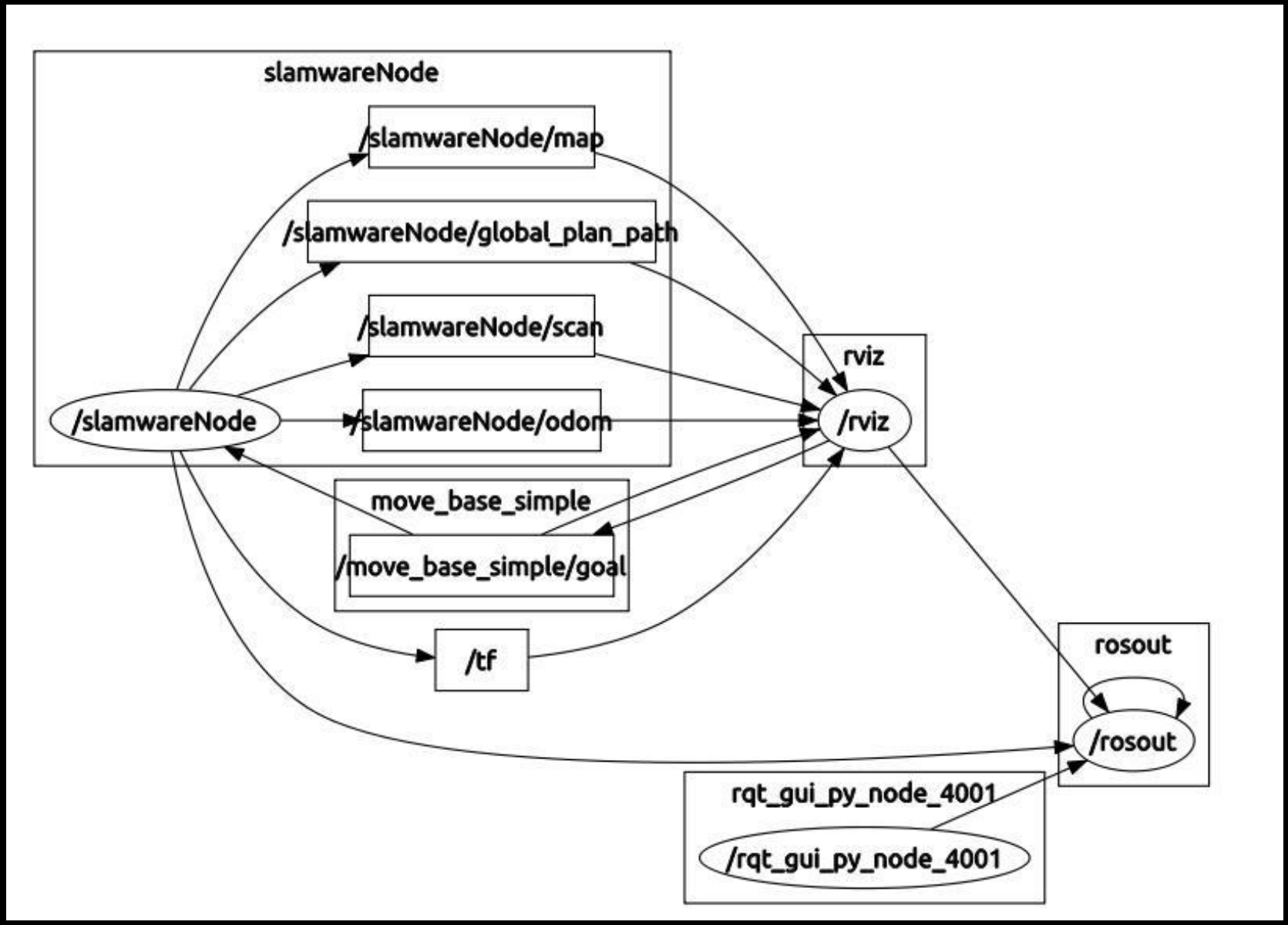
/cmd_vel

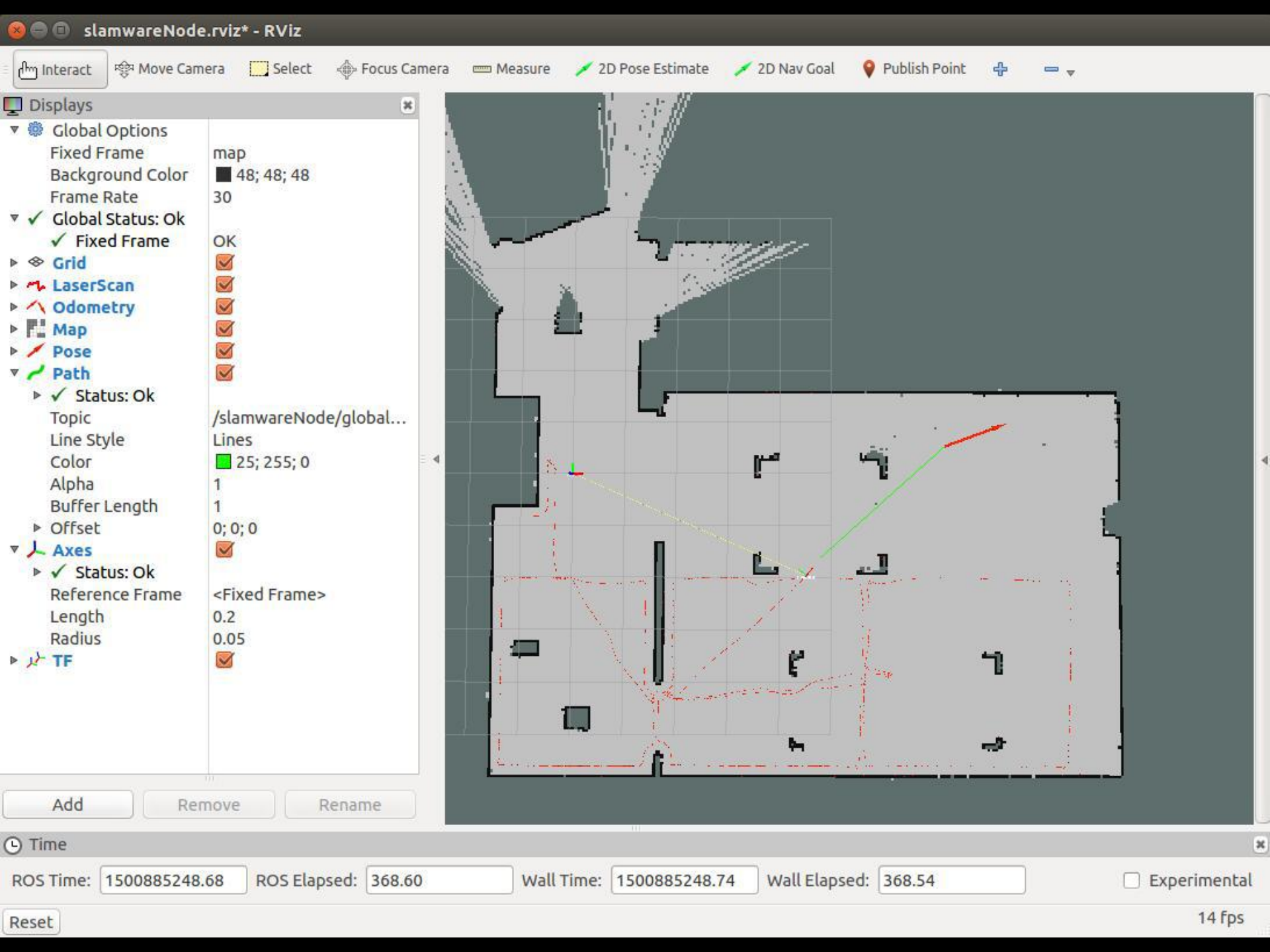
/move_base_simple/goal



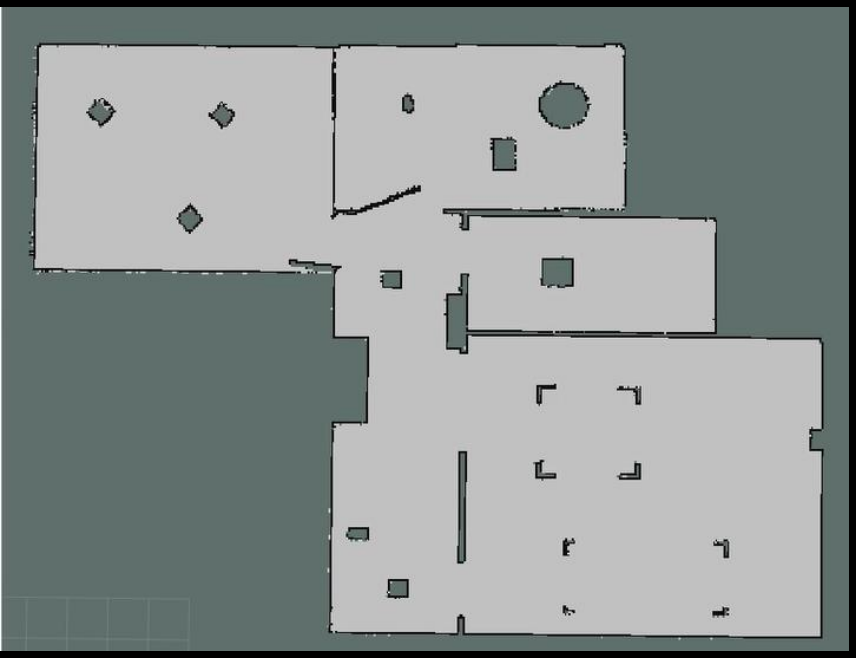
slamwareAPP Test for ROS Github: ***https://github.com/kintzhao/slamwareAPP***

4. SLAMWARE solution for localization and navigation



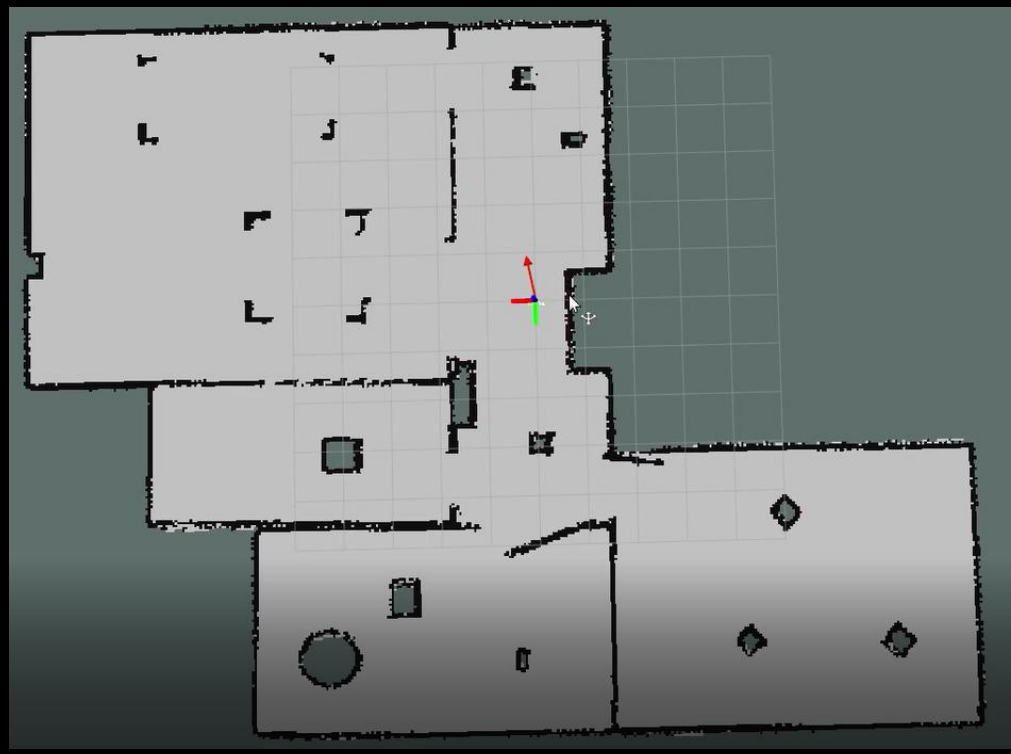

slamwareNode.rviz* - RViz
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame
map
Background Color
48; 48; 48
Frame Rate
30
Global Status: Ok
Fixed Frame
OK
Grid
LaserScan
Odometry
Map
Pose
Path
Status: Ok
Topic
/slamwareNode/global...
Line Style
Lines
Color
25; 255; 0
Alpha
1
Buffer Length
1
Offset
0; 0; 0
Axes
Status: Ok
Reference Frame
<Fixed Frame>
Length
0.2
Radius
0.05
TF
Add
Remove
Rename
Time
ROS Time:
1500885248.68
ROS Elapsed:
368.60
Wall Time:
1500885248.74
Wall Elapsed:
368.54
Experimental
Reset
14 fps

4. SLAMWARE solution for localization and navigation



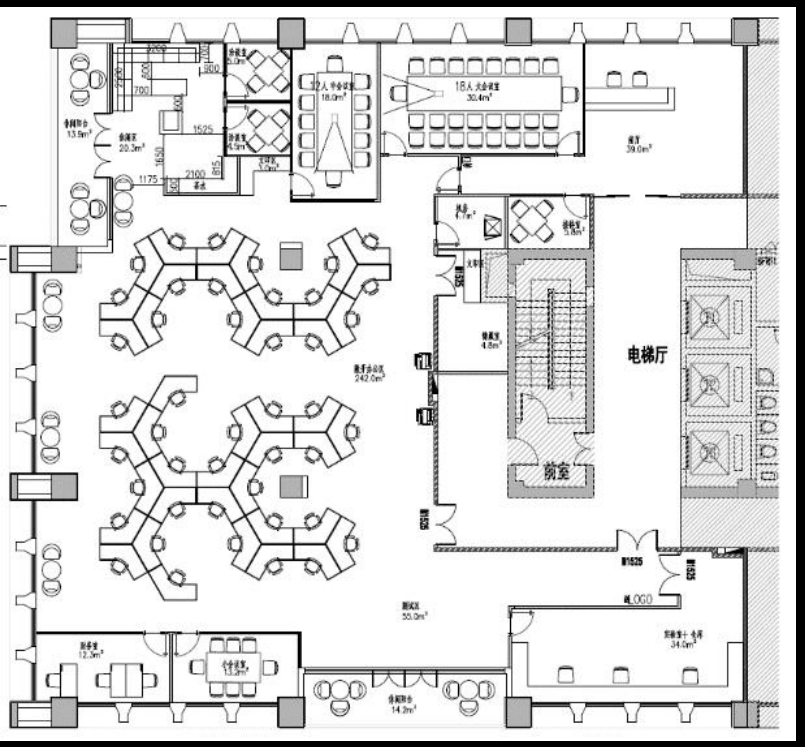
slamware

Laser: angle_increment

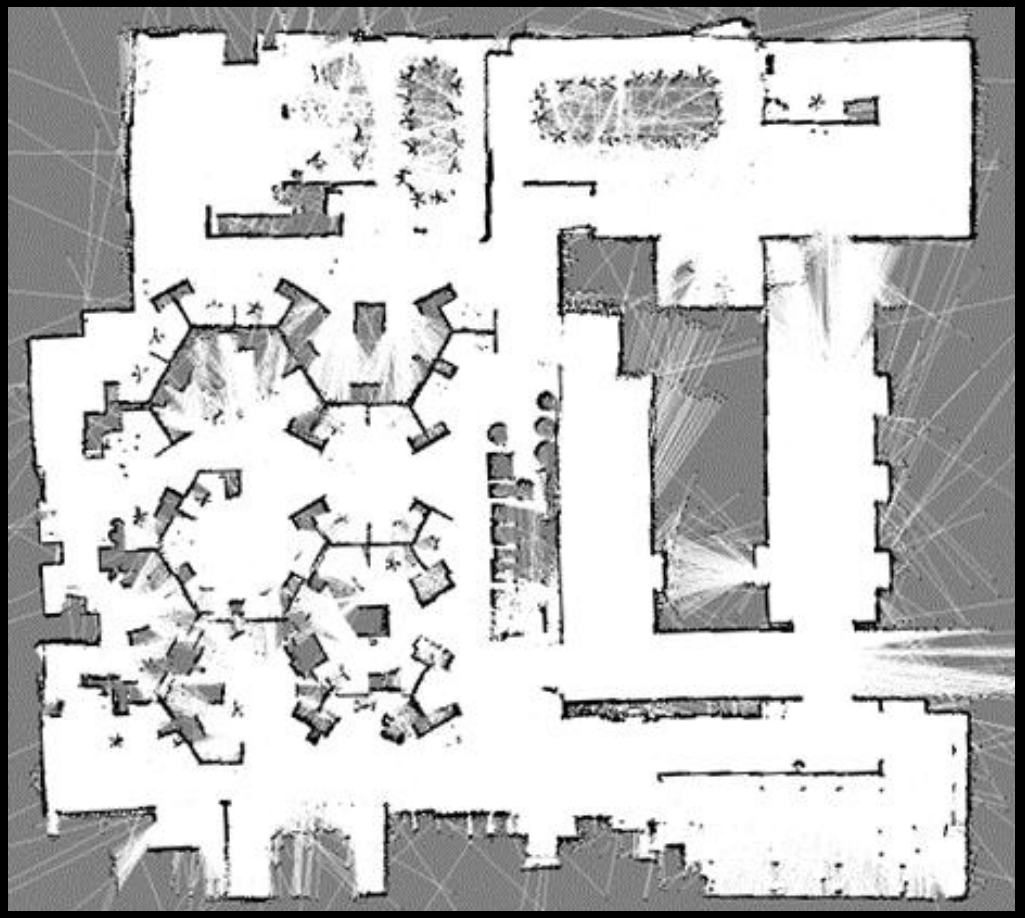


gmapping

SLAMWARE solution for localization and navigation

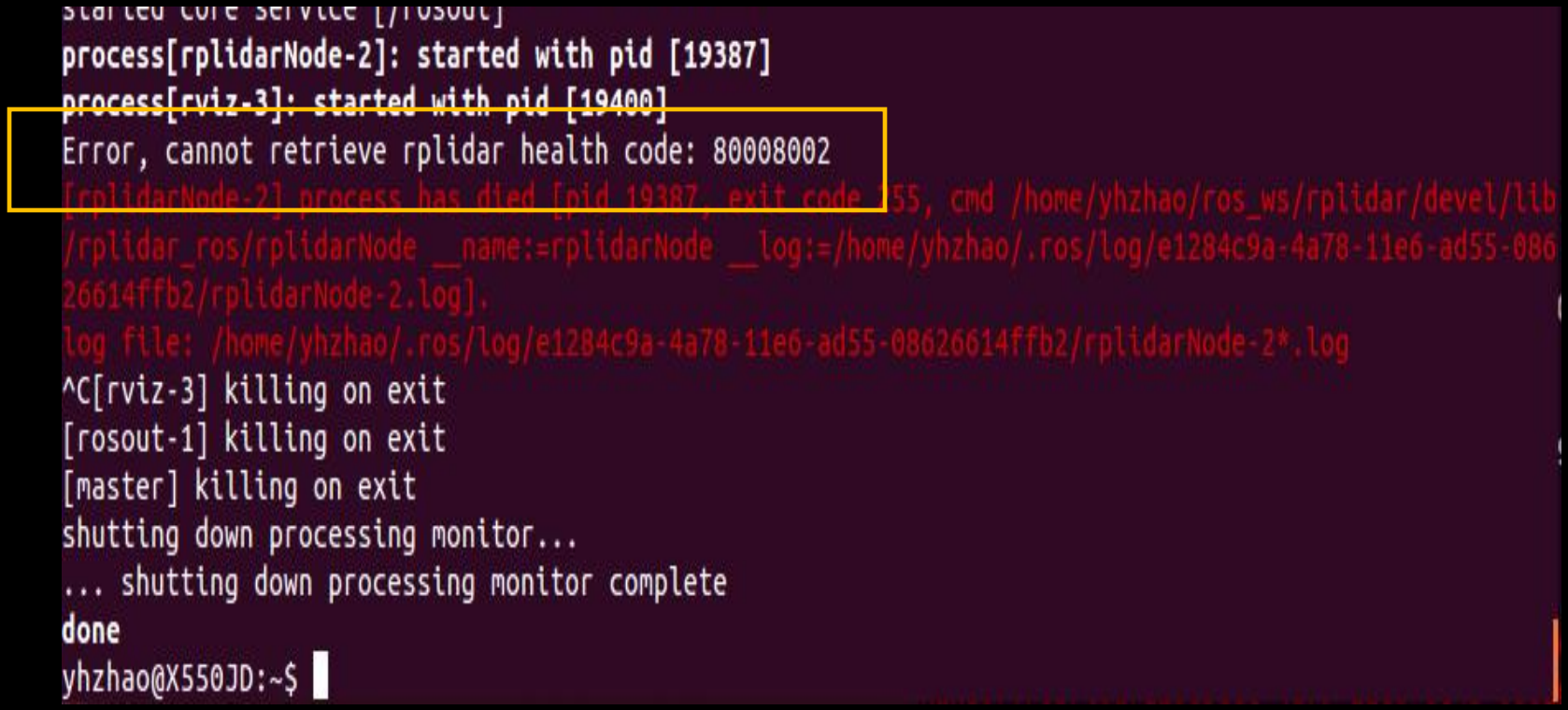


Design



SLAMWARE BUILDE

ISSUE 1： Power



Check your power: current

ISSUE2： Authority

```
started core service [/rosout]
process[rplidarNode-2]: started with pid [16292]
process[rviz-3]: started with pid [16300]
Error, cannot bind to the specified serial port /dev/ttyUSB0.
[rplidarNode-2] process has died [pid 16292, exit code 255, cmd /home/yhzhao/ros
_ws/rplidar/devel/lib/rplidar_ros/rplidarNode __name:=rplidarNode __log:=/home/y
hzhao/.ros/log/89cb440c-4a73-11e6-91e2-08626614ffb2/rplidarNode-2.log].
log file: /home/yhzhao/.ros/log/89cb440c-4a73-11e6-91e2-08626614ffb2/rplidarNode
-2*.log
^C[rviz-3] killing on exit
[rosout-1] killing on exit
[master] killing on exit
shutting down processing monitor...
... shutting down processing monitor complete
done
yhzhao@X550JD:~$ roslaunch rplidar_ros view_rplidar.launch
```

Check USB Authority

Sudo chmode

```
yhzhao@X550JD: ~
yhzhao@X550JD:~$ ls -l /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0  7月 15 16:31 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1  7月 15 16:31 /dev/ttyUSB1
```

```
yhzhao@X550JD: ~
yhzhao@X550JD:~$ ls -l /dev/ttyUSB*
crwxrwxrwx 1 root dialout 188, 1  7月 15 18:51 /dev/ttyUSB1
crwxrwxrwx 1 root dialout 188, 2  7月 15 18:51 /dev/ttyUSB2
yhzhao@X550JD:~$
```

USB REMAP: udev

```
sudo cp `rospack find rplidar_ros`/scripts/rplidar.rules  /etc/udev/rules.d
```

```
# set the udev rule , make the device_port be fixed by rplidar
#
KERNEL=="ttyUSB*", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60", MODE:="0777", SYMLINK+="rplidar"
```

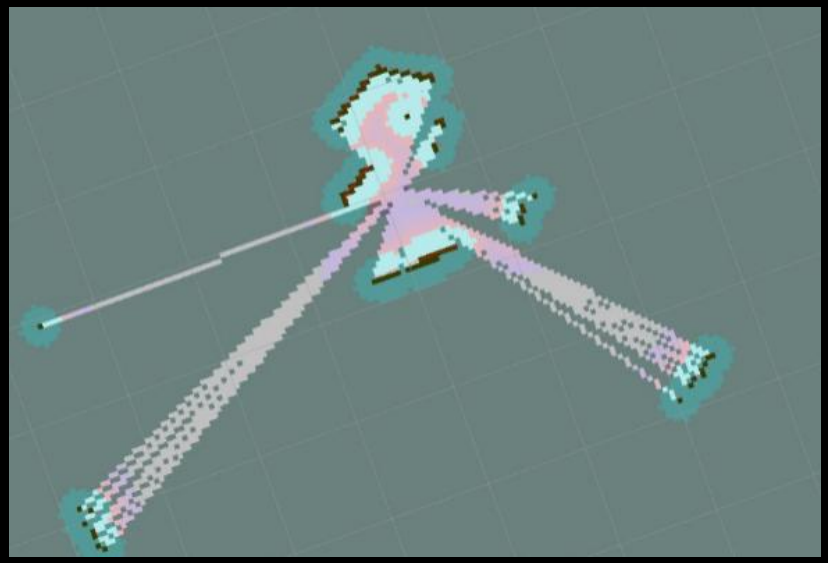
lsusb

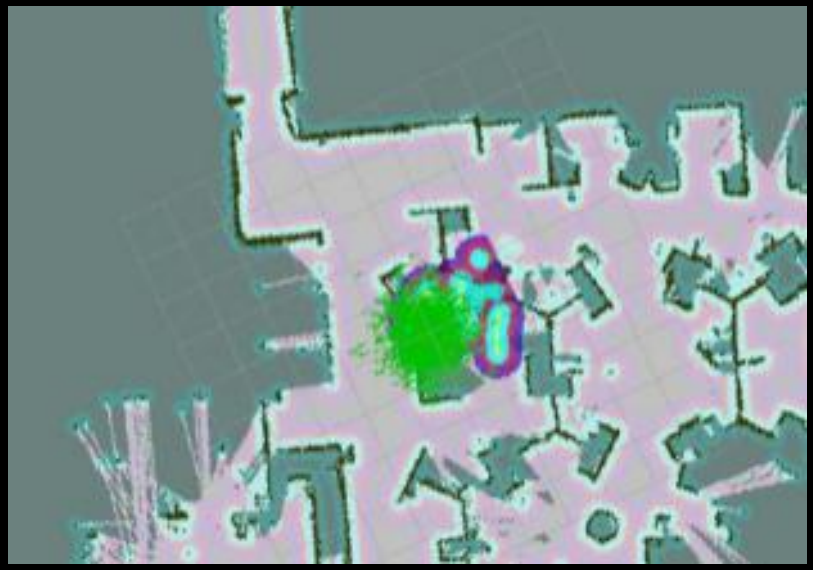
udevadm info --attribute-walk --path=/sys/bus/usb-serial/devices/ttyUSB0

```
    SUBSYSTEMS=="usb"
    DRIVERS=="cp210x"
    ATTRS{bInterfaceClass}=="ff"
    ATTRS{bInterfaceSubClass}=="00"
    ATTRS{bInterfaceProtocol}=="00"
    ATTRS{bNumEndpoints}=="02"
    ATTRS{supports_autosuspend}=="1"
    ATTRS{bAlternateSetting}==" 0"
    ATTRS{bInterfaceNumber}=="00"
    ATTRS{interface}=="CP2102 USB to UART Bridge Controller"
KERNEL=="ttyUSB*", KERNELS=="1-2.1", MODE:="0777",SYMLINK+="slamtec_base_"
KERNEL=="ttyUSB*", KERNELS=="1-2.2", MODE:="0777",SYMLINK+="slamtec_laser_"
```

```
yhzhao@X550JD:~/ros_ws/rplidar/src/scripts$ ls -l /dev|grep ttyUSB
lrwxrwxrwx  1 root root           7  7月 15 19:18 rplidar -> ttyUSB0
lrwxrwxrwx  1 root root           7  7月 15 19:18 slamtec_base_ -> ttyUSB0
lrwxrwxrwx  1 root root           7  7月 15 19:18 slamtec_laser_ -> ttyUSB1
crwxrwxrwx  1 root dialout 188,   0  7月 15 19:18 ttyUSB0
crwxrwxrwx  1 root dialout 188,   1  7月 15 19:18 ttyUSB1
```

RPLidar A2 running Navigation: Notice





p_navigation › src › sdp_navigation › param

Name	Date modified	Type	Size
costmap_common_params.yaml	7/17/2016 11:56 AM	YAML File	2 KB
dummy.yaml	7/16/2016 3:26 PM	YAML File	1 KB
dwa_local_planner_params.yaml	7/16/2016 3:26 PM	YAML File	3 KB
global_costmap_params.yaml	7/16/2016 3:26 PM	YAML File	1 KB
global_planner_params.yaml	7/16/2016 3:26 PM	YAML File	2 KB
karto_mapper_params.yaml	7/14/2016 4:11 PM	YAML File	2 KB

```
topic: scan
marking: true
clearing: true
min_obstacle_height: 0.0 #0.25
max_obstacle_height: 2.0 #0.35
```

Local_costmap all zero :

Map type: voxel (3D)

Min_obstacle_height

Max_obstacle_height

Refrence

- https://github.com/ros-perception/slam_karto
- http://wiki.ros.org/move_base
- http://wiki.ros.org/navigation?distro=kinetic
- http://wiki.ros.org/rplidar
- https://github.com/robopeak/rplidar_ros
- http://www.slamtec.com/
- http://wiki.ros.org/hector_slam
- https://github.com/googlecartographer/cartographer_ros/tree/master/cartographer_ros
- http://blog.csdn.net/zyh821351004/article/details/51945143
- http://blog.csdn.net/zyh821351004/article/category/2737261 （ROS ）
- http://blog.csdn.net/heyijia0327/article/category/2768679 （用ROS开发自己的机器人）
- http://blog.csdn.net/csshell2002/article/category/5801947 （ movebase导航和地图数据的使用）
- https://mp.weixin.qq.com/s/LdbFp-Zvkr02-_25ILb16g（**cartographer**）

Konolige K, Grisetti G, Kümmerle R, et al. Efficient sparse pose adjustment for 2D mapping[C]//Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. IEEE, 2010: 22-29.

Santos J M, Portugal D, Rocha R P. An evaluation of 2D SLAM techniques available in robot operating system[C]//2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2013: 1-6.

Grisetti G, Stachniss C, Burgard W. Improved techniques for grid mapping with rao-blackwellized particle filters[J]. IEEE transactions on Robotics, 2007, 23(1): 34-46.

Thrun S. Probabilistic robotics[J]. Communications of the ACM, 2002, 45(3): 52-57.

Kohlbrecher S, Von Stryk O, Meyer J, et al. A flexible and scalable slam system with full 3d motion estimation[C]//2011 IEEE International Symposium on Safety, Security, and Rescue Robotics. IEEE, 2011: 155-160.

Kretzschmar H, Stachniss C. Information-theoretic compression of pose graphs for laser-based SLAM[J]. The International Journal of Robotics Research, 2012, 31(11): 1219-1230.

My CSDN： http://blog.csdn.net/zyh821351004

2017思岚体验营

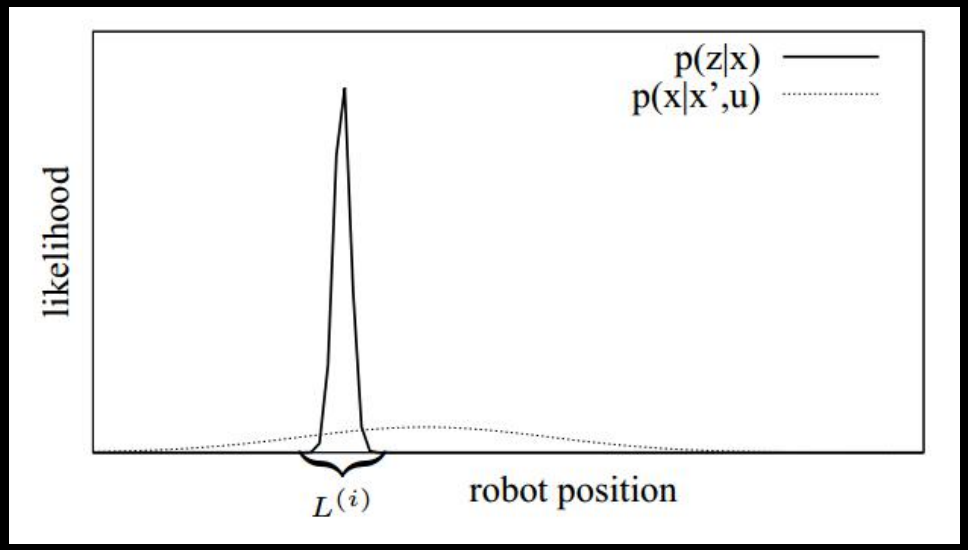
WELCOME JOIN IN SLAMTEC

Email: *jobs@Slamtec.com*

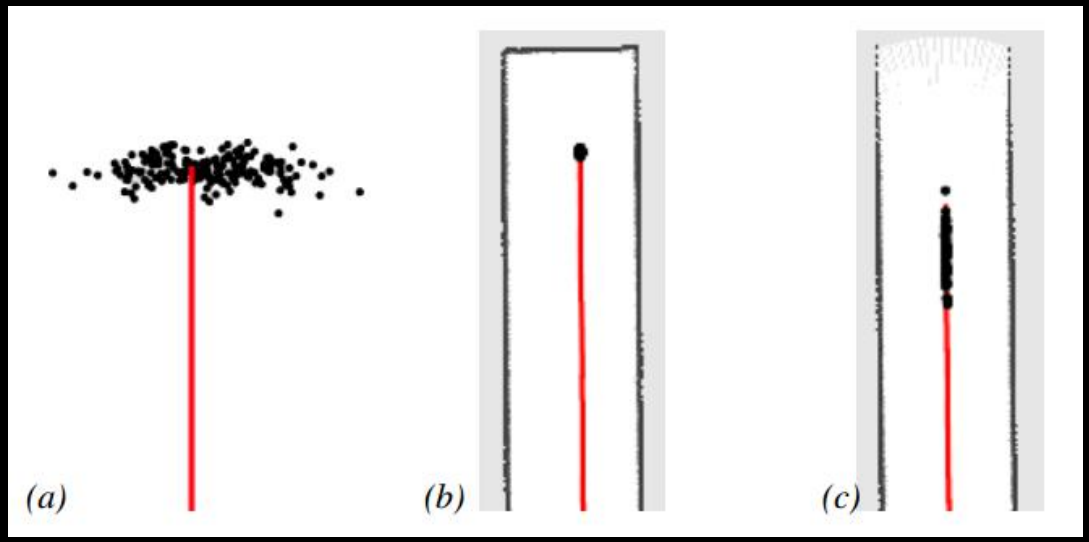
Gmapping

MCL : Particle Filters(PF)

Improve Odom



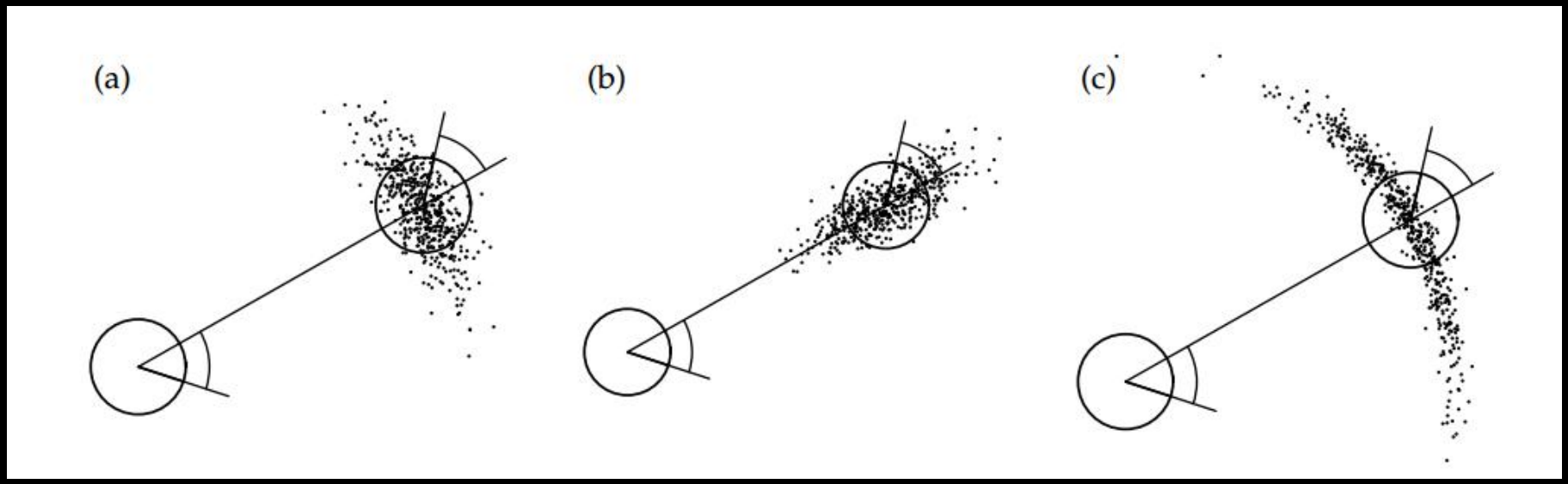
Observed during mapping



Gmapping

MCL / Particle Filters(PF)

Sampling : Motion model(odom)



The number of particles → **small**

Scan Matcher (SGD/ICP)

$$x_t^* = \underset{x_t}{\operatorname{argmax}}\, p(z_t \mid x_t, m_{t-1}) \cdot p(x_t \mid x_{t-1}^*, u_{t-1})$$

- $bestPose = x_{\text{init}}$
- $bestScore = s(bestPose, z, m)$
- $searchStep = initialSearchStep$
- $iterations = 0$
- while $(!iterations < maxIterations)$
  - $maxMoveScore = bestScore$
  - $bestMovePose = bestPose$
  - for $move$ in $(Backward, Forward, Left, Right, RotateLeft, RotatrRight)$
    - $testPose = computePose(bestPose, move)$
    - $score = s(testPose, z, m)$
    - if $(maxMoveScore < score)$
      - $maxMoveScore = score$
      - $bestMovePose = testPose$
  - if $(bestScore < maxMoveScore)$
    - $bestScore < maxMoveScore$
    - $bestPose = bestMovePose$
  - else
    - $searchStep = searchStep/2$
    - $iterations++$

Simple Gradient Descent

Iteration number

Local minimal value

Score Function

updateTreeWeights / Resample

updateTreeWeights ：Normalize (max + distance)

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} \left(\tilde{w}^{(i)}\right)^2},$$

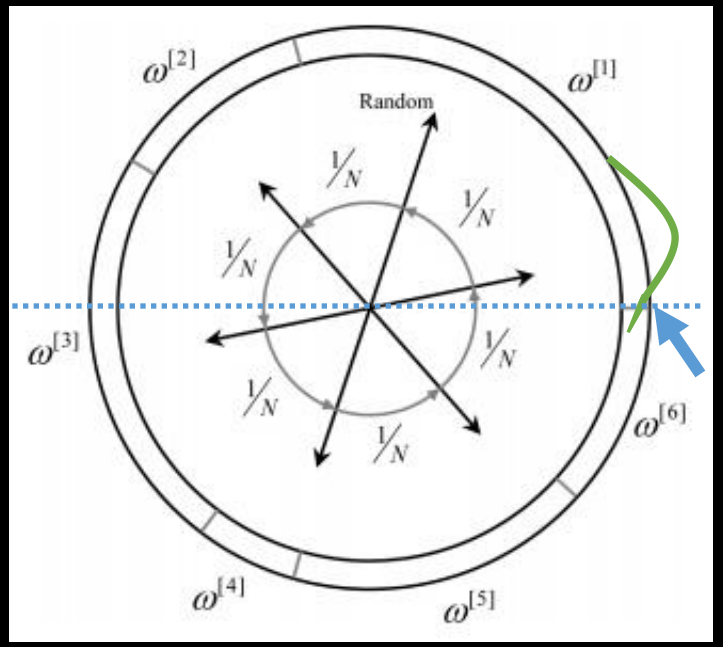
Neff （effective number of particles）

Neff < threshold : Particle deletion

Resample :

Roulette wheel resampling

Stochastic universal



karto_slam(SPA) / nav_2d_karto（TBB+G2O）

- Optimizer： SPA(Sparse Pose Adjustment)

Objective function:

$$F(c,e)=\sum_{i,j\in E} e_{ij}^{T}\Lambda_{ij}e_{ij}$$

Continuable Levenberg-Marquardt:

$$\left(H+\lambda\cdot H_{diag}\right)\Delta c=J^{T}\Lambda e$$

where

$$\mathbf{H}\equiv\mathbf{J}^{\top}\mathbf{\Lambda}\mathbf{J}$$

$$J=\partial e/\partial c$$

$$\Lambda=\mathrm{diag}\left(\Lambda_{\mathrm{ij}},ij\in E\right)$$

$$c=c+\Delta c$$

```
sudo apt-get install  libcsparse3.1.2 libcxsparse3.1.2
libsuitesparse-dev
```

Hector SLAM

Only scan

Multi-Resolution Map

ScanMatcher (Guass-Newton)

IncreasedMap

Features

Demand

Laser frequent

Move slowly

Map memory constant

ScanMatcher (Gaussian-Newton)

State (t-1) $\xi = (p_x, p_y, \psi)^{\mathrm{T}}$ $\Delta\xi$ $\xi + \Delta\xi$ Estimate (t)

Optimization Function:

$$\xi^* = \underset{\xi}{\operatorname{argmin}} \sum_{i=1}^{n} [1 - M(\mathbf{S}_i(\xi))]^2$$

Measure error:

$$\sum_{i=1}^{n} [1 - M(\mathbf{S}_i(\xi + \Delta\xi))]^2 \to 0\,.$$

Active map_cell: (t)

$$\mathbf{S}_i(\xi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & cos(\psi) \end{pmatrix} \begin{pmatrix} s_{i,x} \\ s_{i,y} \end{pmatrix} + \begin{pmatrix} p_x \\ p_y \end{pmatrix}$$

$$\frac{\partial S_i(\xi)}{\partial \xi} = \begin{pmatrix} 1 & 0 & -\sin(\psi)s_{i,x} - \cos(\psi)s_{i,y} \\ 0 & 1 & \cos(\psi)s_{i,x} - \sin(\psi)s_{i,y} \end{pmatrix}$$

ScanMatcher (Gaussian-Newton)

$$\sum_{i=1}^{n} [1 - M(\mathbf{S}_i(\boldsymbol{\xi} + \Delta\boldsymbol{\xi}))]^2 \to 0 .$$

Taylor Expansion:

$$\sum_{i=1}^{n} \left[1 - M(\mathbf{S}_i(\boldsymbol{\xi})) - \nabla M(\mathbf{S}_i(\boldsymbol{\xi})) \frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \Delta\boldsymbol{\xi}) \right]^2 \to 0 .$$

Partial Derivative :

$$2 \sum_{i=1}^{n} \left[\nabla M(\mathbf{S}_i(\boldsymbol{\xi})) \frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \right]^T \left[1 - M(\mathbf{S}_i(\boldsymbol{\xi})) - \nabla M(\mathbf{S}_i(\boldsymbol{\xi})) \frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \Delta\boldsymbol{\xi})) \right] = 0$$

Estimate : (t)

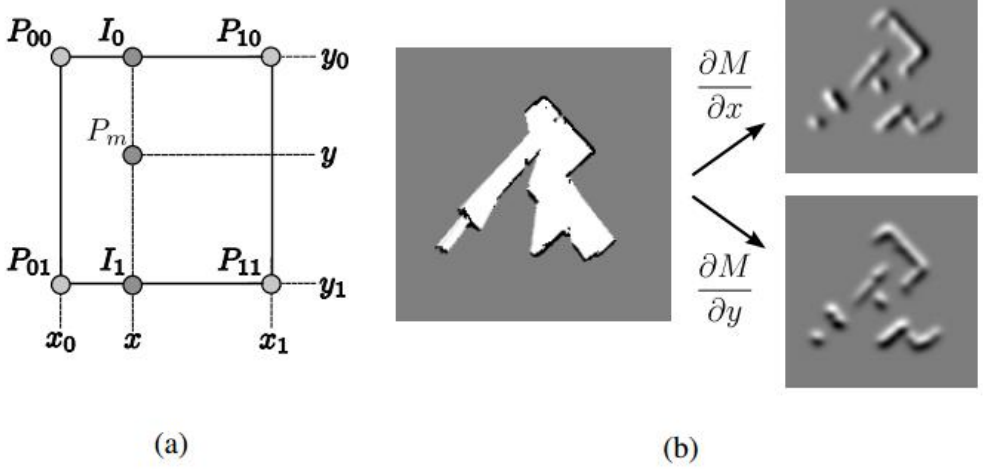
$$\Delta\boldsymbol{\xi} = \mathbf{H}^{-1} \sum_{i=1}^{n} \left[\nabla M(\mathbf{S}_i(\boldsymbol{\xi})) \frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \right]^T [1 - M(\mathbf{S}_i(\boldsymbol{\xi}))]$$

$$\mathbf{H} = \sum_{i=1}^{n} \left[\nabla M(\mathbf{S}_i(\boldsymbol{\xi})) \frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \right]^T \left[\nabla M(\mathbf{S}_i(\boldsymbol{\xi})) \frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \right]$$

Map Access

Bilinear filtering

$$M(P_m) \approx \frac{y - y_0}{y_1 - y_0}\left(\frac{x - x_0}{x_1 - x_0}M(P_{11}) + \frac{x_1 - x}{x_1 - x_0}M(P_{01})\right) + \frac{y_1 - y}{y_1 - y_0}\left(\frac{x - x_0}{x_1 - x_0}M(P_{10}) + \frac{x_1 - x}{x_1 - x_0}M(P_{00})\right)$$



(a) (b)

Fig. 2. (a) Bilinear filtering of the occupancy grid map. Point P_m is the point whose value shall be interpolated. (b) Occupancy grid map and spatial derivatives.

P = occ/(occ+free)

$$\frac{\partial M}{\partial x}(P_m) \approx \frac{y - y_0}{y_1 - y_0}(M(P_{11}) - M(P_{01})) + \frac{y_1 - y}{y_1 - y_0}(M(P_{10}) - M(P_{00}))$$

$$\frac{\partial M}{\partial y}(P_m) \approx \frac{x - x_0}{x_1 - x_0}(M(P_{11}) - M(P_{10})) + \frac{x_1 - x}{x_1 - x_0}(M(P_{01}) - M(P_{00}))$$

Add Links

KeyScan: 机器人运动一定的距离或角度

addScans: 将chain laser生成map(occupy), **scanToMap**的方式调整odom预测的pose.

Link to previsous scan

Link to Running scans

RunningScan chain：一定数量且距当前一定距离内的激光数据链。滑动窗口式抛 距离最远的scan。

Link to other near chains

NearChain: 以当前节点开始广度优先的方式从graph中遍历相邻的一定距离范围内所有节点，依据当前id从sensorManager中分别递增与递减寻找一定范围内的chain，生成nearLinkScans.

Loop Closure：

1) 依据当前的Vertex, 从Graph中找到与之相邻的所有vertex(一定距离范围内).

2) 采取广度优先搜索的方式，将相邻（next）与相连（adjacentVertices）添加进 nearLinkedScans.

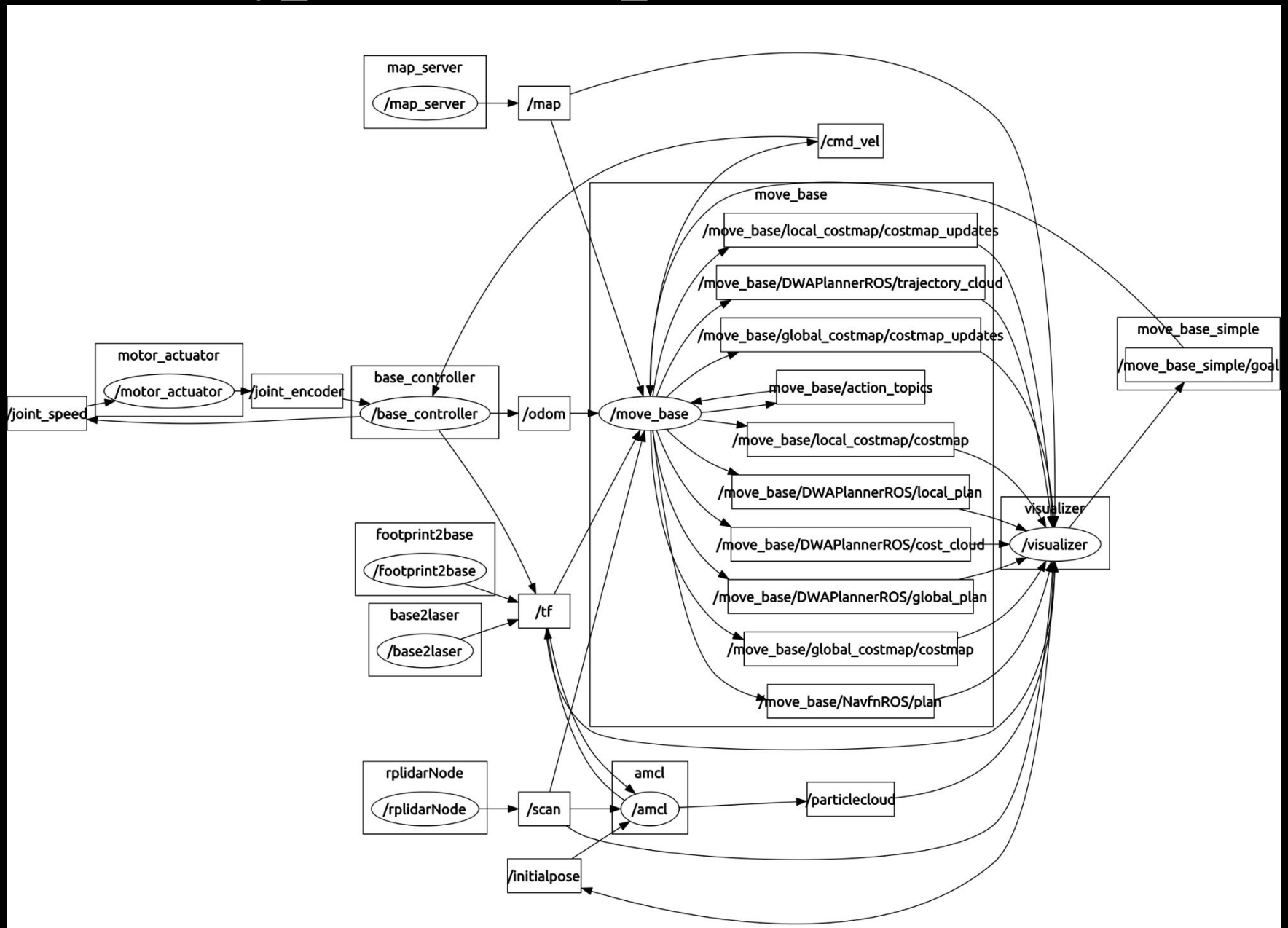
3) 从sensorManager中取从前到后，依据id序号挑选与当前在一定距离范围内，且不在 nearLinkedScans中的candidateScans, 当数量达到一定size，返回。

4）loopScanMatcher进行scanTomap的匹配，当匹配response 和covariance达到一定要求 认为闭环检测到。得到调整的correct pose.

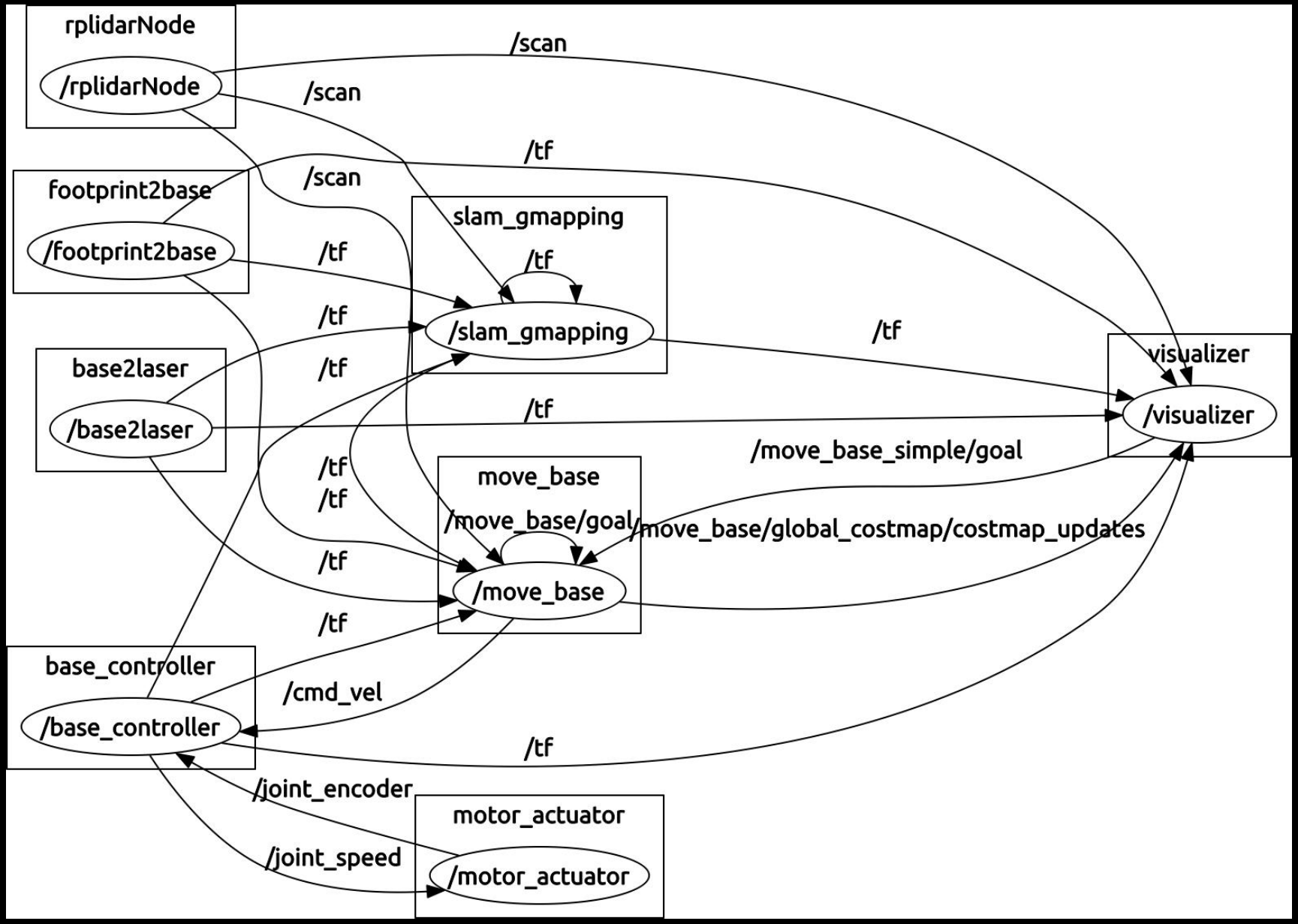
5）Add link to loop： 调整边（全局闭环）

6) 触发correctPose: spa优化

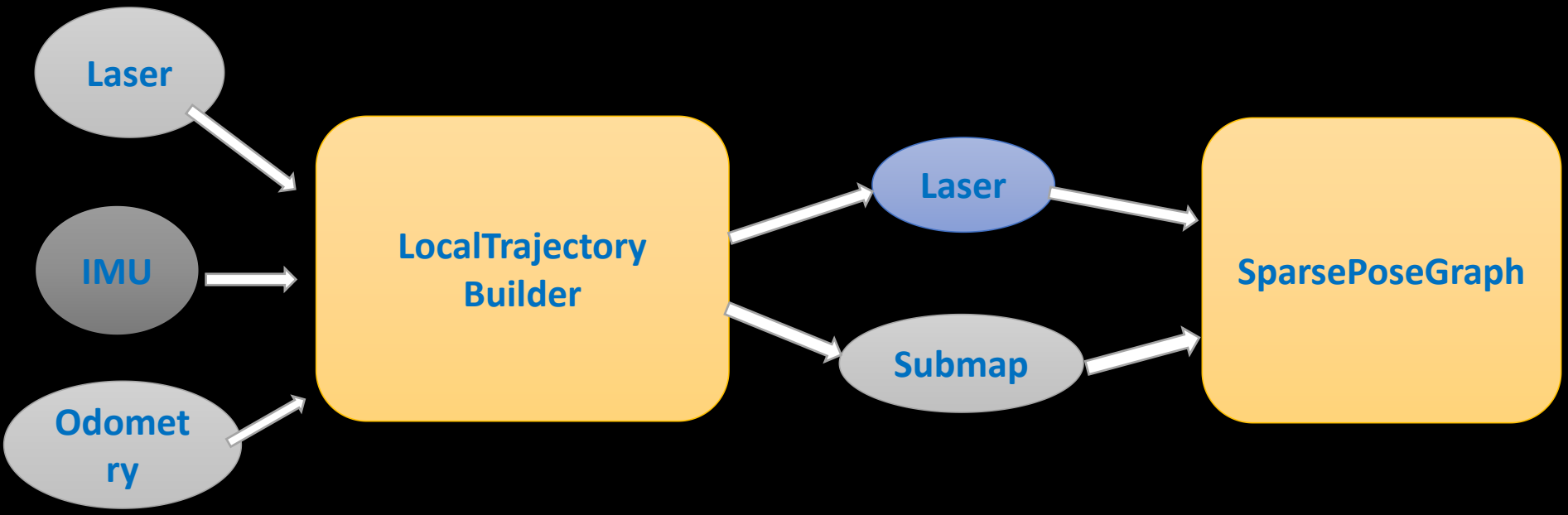
AMCL + map_server + move_base + ROBOT(base + sensor)



SLAM + move_base + ROBOT(base + sensor)



CartoGrapher 流程



LocalTrajectoryBuilder模块：使用Odometry数据和上一帧机器人位姿得到机器人在当前时刻的初始位姿，利用Laser Scan数据和Submap，使用RealTimeCorrelativeScanMatch方法找到最优的dx,dy,dtheta.

利用找到的最优位姿进行进一步的优化，优化目标函数如下所示:

$$\underset{\xi}{\operatorname{argmin}} \sum_{k=1}^{K} \left(1 - M_{\text{smooth}}(T_\xi h_k)\right)^2$$

Platform：

pc(i7-4710hq 8G) ubuntu 14.04 indigo + sdp(serial) + rplidar A2

e105 : 16.9*25.2 = 425 m^2

内存：500M scan_index 3065 constraint 3394 submap 31 （一圈）

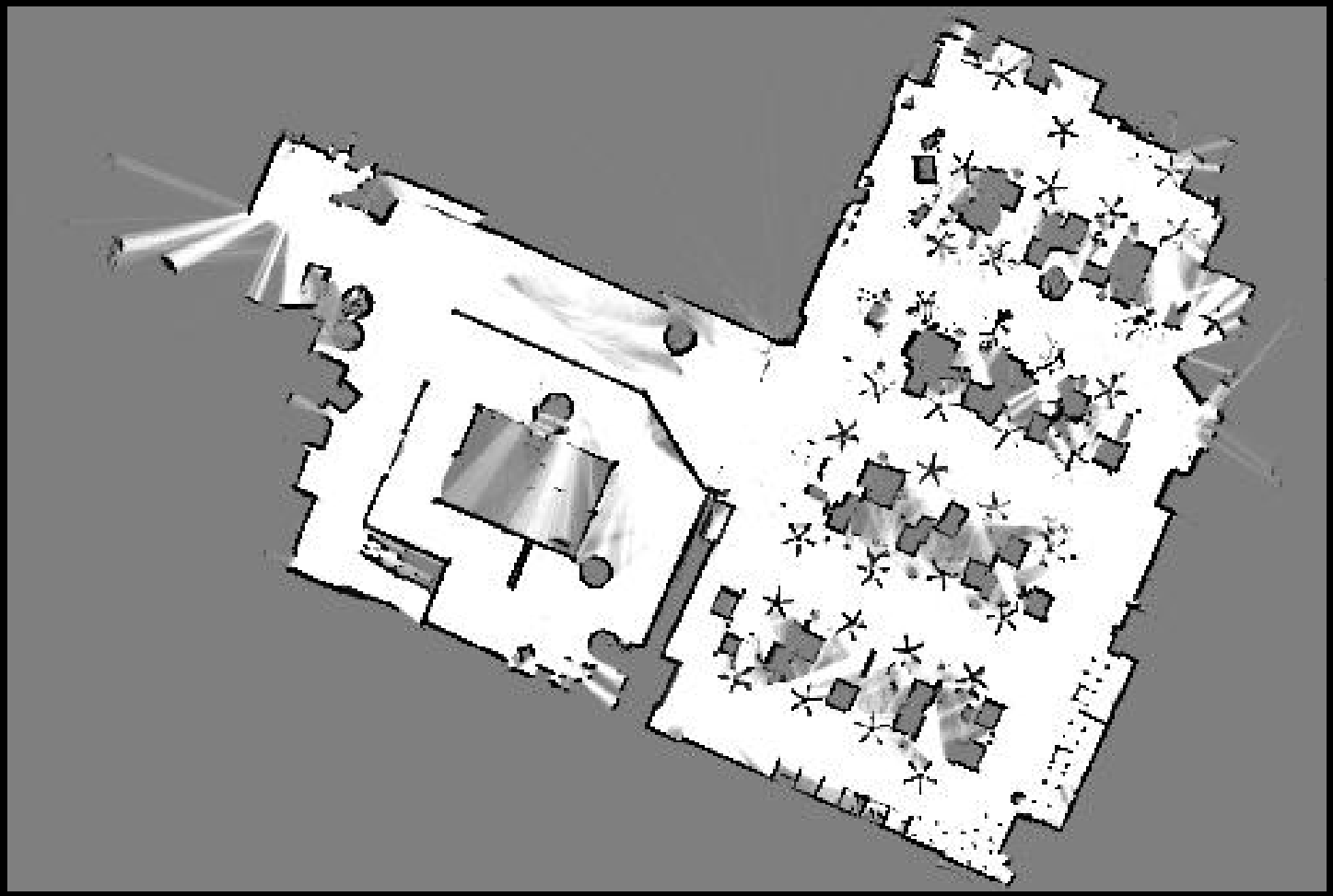
Platform：

工控机（celeronJ900 1.9Ghz×4， 3.8G）ubuntu16.04 Kinect [cartographer 算法]

raspberry pi2 ubuntu 14.04 inidgo [sdp(serial) + rplidar A2]

e105 : 16.9*25.2 = 425 m^2

内存：内存 2.78G



rosbag : 面积： 166.3× 271.4 = 4万 m^2

c 内存： 2.7G 28552 constraint ; 356 submap 32283 scan_index

pc(i7-4710hq 8G) ubuntu 14.04 indigo