ROS tutorial

navigation/gmapping

ROBOT OPERATING SYSTEM LAB SESSION 7

17/04/2018

gmapping

- The gmapping package provides laser-based SLAM
- you can create a 2-D occupancy grid map (like a building floorplan) from laser and pose data collected by a mobile robot
- node —— slam_gmapping

Slam_gmapping

Subscribe Topics

tf(tf/tf_Message)

Transforms necessary to relate frames for laser, base, and odometry

Scan(sensor_msg/LaserScan)

Laser scans to create the map from

Published Topics

Map_metadata(nav_msgs/MapMetaData)

Get the map data from this topic, which is latched, and updated periodically.

map (nav_msgs/OccupancyGrid)

Get the map data from this topic, which is latched, and updated periodically

Slam_gmapping

Parameters:

~map_frame (string, default: "map") The frame attached to the map.

~odom_frame (string, default: "odom") The frame attached to the odometry system

~minimumScore (float, default: 0.0)

Minimum score for considering the outcome of the scan matching good. Can avoid jumping pose estimates in large open spaces when using laser scanners with limited range

~particles (int, default: 30) Number of particles in the filter

Transforms

Required tf

Laser scan   base_link

usually a fixed value, broadcast periodically by a robot_state_publisher, or a tf static_transform_publisher.

base_link   odom

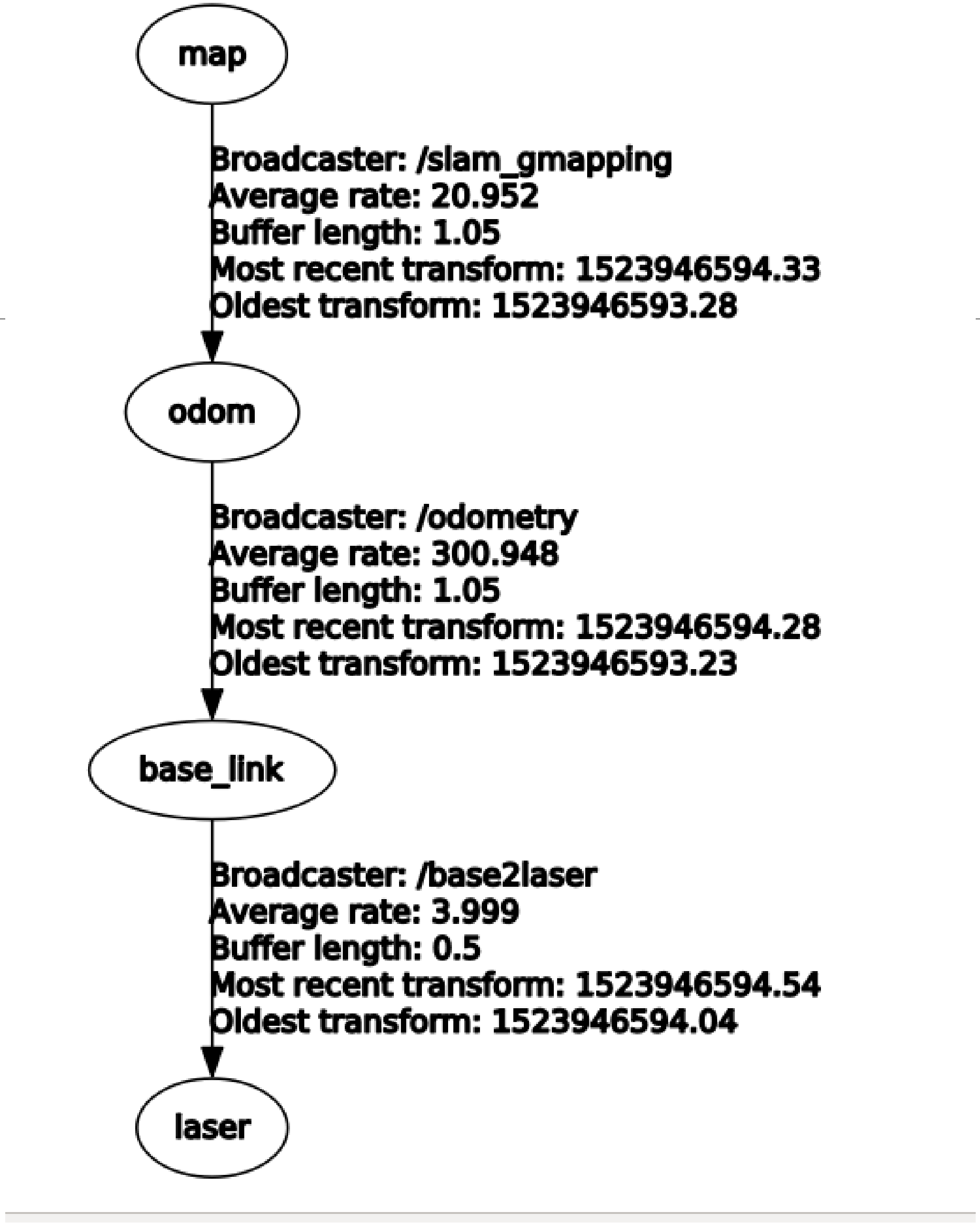
usually provided by the odometry system (e.g., the driver for the mobile base)

Provided tf

map → odom

the current estimate of the robot's pose within the map frame

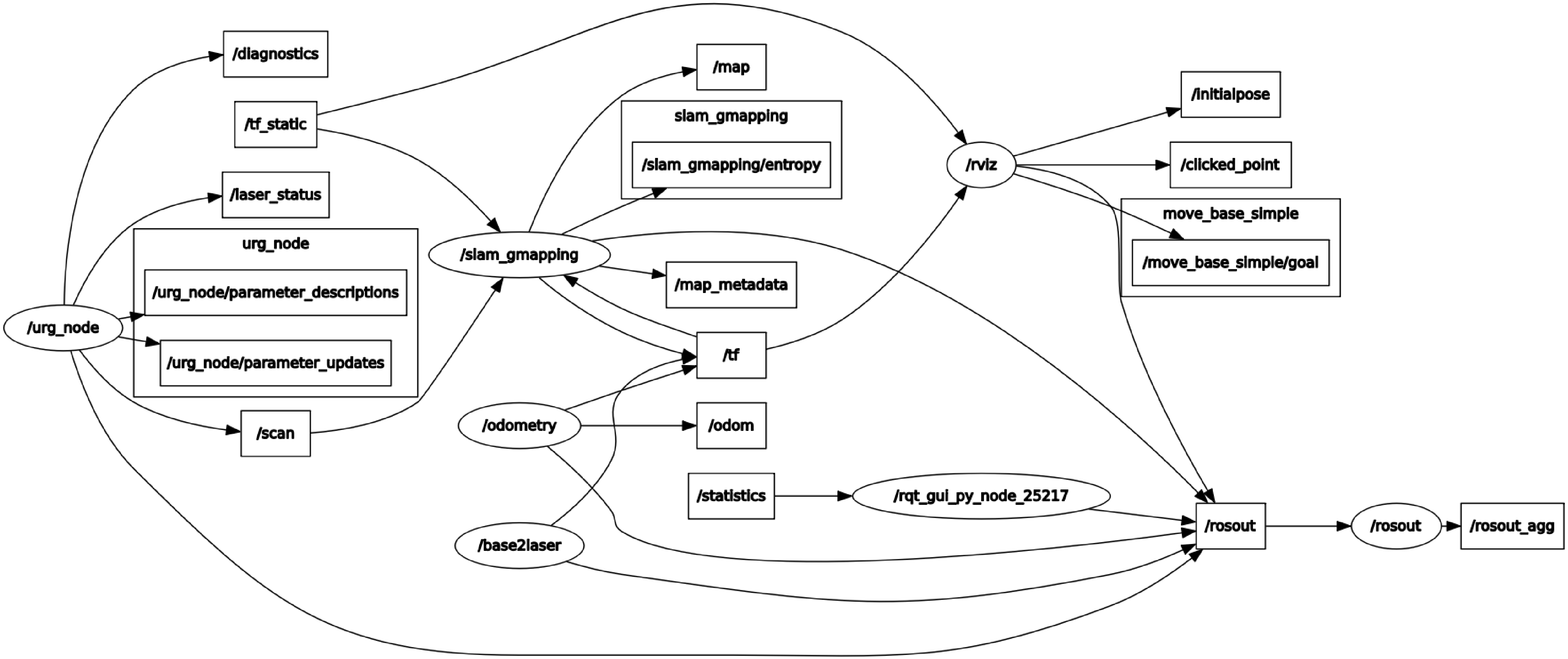
transform



Broadcaster: /slam_gmapping
Average rate: 20.952
Buffer length: 1.05
Most recent transform: 1523946594.33
Oldest transform: 1523946593.28

Broadcaster: /odometry
Average rate: 300.948
Buffer length: 1.05
Most recent transform: 1523946594.28
Oldest transform: 1523946593.23

Broadcaster: /base2laser
Average rate: 3.999
Buffer length: 0.5
Most recent transform: 1523946594.54
Oldest transform: 1523946594.04


/diagnostics
/tf_static
/laser_status
urg_node
/urg_node/parameter_descriptions
/urg_node/parameter_updates
/urg_node
/scan
/map
slam_gmapping
/slam_gmapping/entropy
/slam_gmapping
/map_metadata
/tf
/odometry
/odom
/statistics
/rqt_gui_py_node_25217
/base2laser
/rviz
/initialpose
/clicked_point
move_base_simple
/move_base_simple/goal
/rosout
/rosout
/rosout_agg

Hokuyo kinetic gmapping

Step 1 hokuyo driver hokuyo_node / urg_node publish /scan

Step 2 transform publish /tf /tf_static

Laser scan base_link

<node pkg="tf" type="static_transform_publisher" name="base2laser" args="0.28 0 0.16 0 0 0 base_link laser 500"/>

base_link odometry

geometry_msgs::TransformStamped odom_trans;

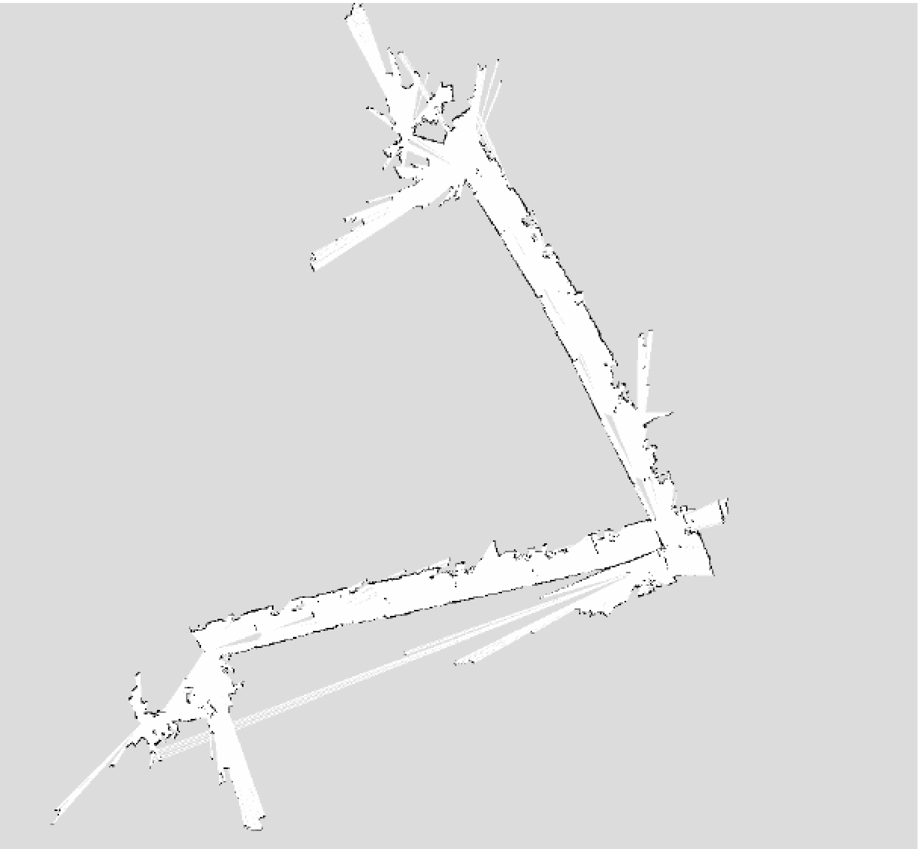
Step3 odometry publish /odom

nav_msgs::Odometry odom

Step4 slam_gmapping publish /map

rosrun gmapping slam_gmapping

demostration



```
rosrun map_server map_saver -f <map_name>
```

homework

bagfile as a reference

turtle_slam0417.bag

Generate your map.pgm by your turtlebot and hokuyo.bag

Submit by a group of four

rosrun map_server map_saver -f <map_name>   save the map