

New in C Turtle

kinetic

melodic

noetic

Show EOL distros: ☐

Documentation Status

Package Links

- **Code API** (http://docs.ros.org/noetic/api/slam_karto/html)
- **FAQ** (http://answers.ros.org/questions/scope:all/sort:activity-desc/tags:slam_karto/page:1/)
- **Changelog** (http://docs.ros.org/en/noetic/changelogs/slam_karto/changelog.html)
- **Change List** (/slam_karto/ChangeList)
- **Reviews** (/slam_karto/Reviews)

Dependencies (10)**Jenkins jobs (10)**

Package Summary

✓ Released ✓ Continuous Integration ✓ Documented

This package pulls in the Karto mapping library, and provides a ROS wrapper for using it.

- Maintainer status: maintained
- Maintainer: Michael Ferguson <mfergs7 AT gmail DOT com>, Luc Bettaieb <luc.bettaieb AT gmail DOT com>, Russell Toris <rtoris AT fetchrobotics DOT com>
- Author: Brian Gerkey
- License: LGPL
- Source: git https://github.com/ros-perception/slam_karto.git (https://github.com/ros-perception/slam_karto) (branch: melodic-devel)

1. Documentation

Part of the documentation is available in the previously maintained karto package (/karto).

2. Nodes

2.1 slam_karto

The `slam_karto` node takes in `sensor_msgs/LaserScan` (http://docs.ros.org/en/api/sensor_msgs/html/msg/LaserScan.html) messages and builds a map (`nav_msgs/OccupancyGrid` (http://docs.ros.org/en/api/nav_msgs/html/msg/OccupancyGrid.html)). The map can be retrieved via a ROS topic (/Topics) or service (/Services).

2.1.1 Subscribed Topics

`tf` (`tf/tfMessage` (<http://docs.ros.org/en/api/tf/html/msg/tfMessage.html>))

Transforms necessary to relate frames for laser, base, and odometry (see below)

`scan` (`sensor_msgs/LaserScan` (http://docs.ros.org/en/api/sensor_msgs/html/msg/LaserScan.html))
Laser scans to create the map from

2.1.2 Published Topics

`map_metadata` (`nav_msgs/MapMetaData` (http://docs.ros.org/en/api/nav_msgs/html/msg/MapMetaData.html))
Get the metadata of the map data (resolution, width, height, ...)

`map` (`nav_msgs/OccupancyGrid` (http://docs.ros.org/en/api/nav_msgs/html/msg/OccupancyGrid.html))
Get the map data from this topic, which is latched, and updated periodically

`visualization_marker_array` (`visualization_msgs/MarkerArray` (http://docs.ros.org/en/api/visualization_msgs/html/msg/MarkerArray.html))
Get the pose graph from this topic, which is updated periodically

2.1.3 Services

`dynamic_map` (`nav_msgs/GetMap` (http://docs.ros.org/en/api/nav_msgs/html/srv/GetMap.html))
Call this service to get the map data

2.1.4 Parameters

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

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

`~base_frame` (string, default: "base_link")
The frame attached to the mobile base.

`~throttle_scans` (int, default: 1)
Process 1 out of every this many scans (set it to a higher number to skip more scans)

`~map_update_interval` (float, default: 5.0)
How long (in seconds) between updates to the map. Lowering this number updates the occupancy grid more often, at the expense of greater computational load.

`~resolution` (float, default: 0.05)
Resolution of the map (in metres per occupancy grid block)

`~delta` (float, default: 0.05)
Resolution of the map (in metres per occupancy grid block). Same as resolution. Defined for compatibility with the parameter names of gmapping (/gmapping).

`~transform_publish_period` (float, default: 0.05)
How long (in seconds) between transform publications. To disable broadcasting transforms, set to 0.

`~use_scan_matching` (bool, default: true)
When set to true, the mapper will use a scan matching algorithm. In most real-world situations this should be set to true so that the mapper algorithm can correct for noise and errors in odometry and scan data. In some simulator environments where the simulated scan and odometry data are very accurate, the scan matching algorithm can produce worse results. In those cases set this to false to improve results.

`~use_scan_barycenter (bool, default: true)`

Use the barycenter of scan endpoints to define distances between scans.

`~minimum_travel_distance (double, default: 0.2)`

Sets the minimum travel between scans. If a new scan's position is more than `minimumTravelDistance` from the previous scan, the mapper will use the data from the new scan. Otherwise, it will discard the new scan if it also does not meet the minimum change in heading requirement. For performance reasons, generally it is a good idea to only process scans if the robot has moved a reasonable amount.

`~minimum_travel_heading (double, default: deg2rad(10))`

Sets the minimum heading change between scans. If a new scan's heading is more than `minimum_travel_heading` from the previous scan, the mapper will use the data from the new scan. Otherwise, it will discard the new scan if it also does not meet the minimum travel distance requirement. For performance reasons, generally it is a good idea to only process scans if the robot has moved a reasonable amount.

`~scan_buffer_size (int, default: 70)`

Sets the length of the scan chain stored for scan matching. `scan_buffer_size` should be set to approximately `scan_buffer_maximum_scan_distance / minimum_travel_distance`. The idea is to get an area approximately 20 meters long for scan matching. For example, if we add scans every `minimum_travel_distance == 0.3` meters, then `scan_buffer_size` should be $20 / 0.3 = 67$.

`~scan_buffer_maximum_scan_distance (double, default: 20.)`

Sets the maximum distance between the first and last scans in the scan chain stored for matching.

`~link_match_minimum_response_fine (double, default: 0.8)`

Scans are linked only if the correlation response value is greater than this value.

`~link_scan_maximum_distance (double, default: 10.)`

Sets the maximum distance between linked scans. Scans that are farther apart will not be linked regardless of the correlation response value.

`~loop_search_maximum_distance (double, default: 4.)`

Scans less than this distance from the current position will be considered for a match in loop closure.

`~do_loop_closing (bool, default: true)`

Enable/disable loop closure.

`~loop_match_minimum_chain_size (int, default: 10)`

When the loop closure detection finds a candidate it must be part of a large set of linked scans. If the chain of scans is less than this value, we do not attempt to close the loop.

`~loop_match_maximum_variance_coarse (double, default: sqrt(0.4))`

The co-variance values for a possible loop closure have to be less than this value to consider a viable solution. This applies to the coarse search.

`~loop_match_minimum_response_coarse (double, default: 0.8)`

If response is larger than this, then initiate loop closure search at the coarse resolution.

`~loop_match_minimum_response_fine (double, default: 0.8)`

If response is larger than this, then initiate loop closure search at the fine resolution.

`~correlation_search_space_dimension` (double, default: 0.3)

Sets the size of the search grid used by the matcher. The search grid will have the size `correlation_search_space_dimension` x `correlation_search_space_dimension`.

`~correlation_search_space_resolution` (double, default: 0.01)

Sets the resolution (size of a grid cell) of the correlation grid.

`~correlation_search_space_smear_deviation` (double, default: 0.03)

The point readings are smeared by this value in X and Y to create a smoother response.

`~loop_search_space_dimension` (double, default: 8.0)

The size of the search grid used by the matcher.

`~loop_search_space_resolution` (double, default: 0.05)

The resolution (size of a grid cell) of the correlation grid.

`~loop_search_space_smear_deviation` (double, default: 0.03)

The point readings are smeared by this value in X and Y to create a smoother response.

`~distance_variance_penalty` (double, default: $\sqrt{0.3}$)

Variance of penalty for deviating from odometry when scan-matching. The penalty is a multiplier (less than 1.0) is a function of the delta of the scan position being tested and the odometric pose.

`~angle_variance_penalty` (double, default: $\sqrt{\deg2rad(20)}$)

See `distance_variance_penalty`.

`~fine_search_angle_offset` (double, default: $\deg2rad(0.2)$)

The range of angles to search during a fine search.

`~coarse_search_angle_offset` (double, default: $\deg2rad(20.0)$)

The range of angles to search during a coarse search.

`~coarse_angle_resolution` (double, default: $\deg2rad(2.0)$)

Resolution of angles to search during a coarse search.

`~minimum_angle_penalty` (double, default: 0.9)

Minimum value of the angle penalty multiplier so scores do not become too small.

`~minimum_distance_penalty` (double, default: 0.5)

Minimum value of the distance penalty multiplier so scores do not become too small.

`~use_response_expansion` (bool, default: false)

Whether to increase the search space if no good matches are initially found

2.1.5 Required tf Transforms

`<the frame_id of the incoming scans>` → `base_link`

usually a fixed value, broadcast periodically by a `robot_state_publisher` (/robot_state_publisher), or a `tf static_transform_publisher` (/tf#static_transform_publisher).

`base_link` → `odom`


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

2.1.6 Provided tf Transforms

`map` → `odom`

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

3. Report a Bug

Use github to  report a bug or propose an enhancement (https://github.com/ros-perception/slam_karto/issues).

Except where otherwise noted, the ROS wiki is licensed under the Creative Commons Attribution 3.0 (<http://creativecommons.org/licenses/by/3.0/>)

Brought to you by:  Open Source Robotics Foundation

(<http://www.osrfoundation.org>)