



Search or ask your question

ASK YOUR QUESTION

ROS Eloquent Elusor tshirts are available until Dec 12th. [Order yours now!](#)

# 1 Why gmapping package requires the base\_link -> odom tf?

gmapping

in the <http://wiki.ros.org/gmapping>

## 4.1.5 Required tf Transforms

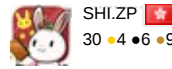
1 the frame attached to incoming scans → base\_link usually a fixed value, broadcast periodically by a robot\_state\_publisher, or a tf static\_transform\_publisher.

2 base\_link → odom usually provided by the odometry system (e.g., the driver for the mobile base)

About 2, should be odom -> base\_link rather than base\_link -> odom, am I correct?

[add a comment](#)

asked Feb 24 '16

SHI.ZP  
30 ●4 ●6 ●9

1 Answer

Sort by » oldest newest most voted

3

/odom frame is a pseudo-global reference frame in which a dead reckoning position and orientation estimate is represented. This estimate is known as odometry. Essentially, the robot measures its linear and angular velocity using wheel encoders and/or inertial sensors, and integrates those values to obtain a position and orientation.

Typically, some node in the ROS system is responsible for subscribing to the raw encoder and inertial data and publishing the relationship between the base frame of the robot, usually called 'base\_link', and the 'odom' frame using a tf::TransformBroadcaster class. Using the notation used in the wiki that you cite, the frame the arrow points to is the parent frame of the transform.

answered Mar 5 '16

robustify  
561 ●2 ●9 ●11

## Question Tools

Follow

2 followers

[subscribe to rss feed](#)

## Stats

Asked:	Feb 24 '16
Seen:	1,121 times
Last updated:	Mar 04 '16

## Related questions

[how to fix the map drifting in rviz ?](#)[How is possible to modify manually occupancy grid cells on a map?](#)[error:can't load nodelet openni\\_launch "navigation" \[closed\]](#)[what map.info.origin.orientation is for?](#)[Problem with navigation and move\\_base stacks](#)[Is robot\\_pose\\_ekf acting correctly?](#)[LaserScan Message vs MultiEchoLaserScan Message \[closed\]](#)[Gmapping map has mixed colors](#)[Changing map\\_saver map size and resolution?](#)[Laser scan to probabilistic map based on occupancy grid](#)

Because encoder and inertial data is noisy and biased, and there is some uncertainty in the physical system, position and orientation estimates based solely on dead reckoning will quickly drift and diverge from the true position. This is where SLAM algorithms such as gmapping come in. The odometry estimate is a key input to gmapping, because at the core of the gmapping algorithm is a particle filter.

The particle filter simultaneously maintains a number of hypothetical map states and corresponding robot poses, with each combination of map and pose being called a *particle*. During each update of the map, gmapping updates each particle's pose estimate with a random sampling around the odometry estimate's velocity values. This propagates the position and heading of the particle according to the odometry, with the random sampling accounting for the possibility of the odometry being inaccurate.

Based on the propagated position and heading of the particles, the current LIDAR scan is matched with each of the separate map hypotheses in each particle. Each particle is then assigned a score based on how well the scan aligns with the map hypothesis. Since the particles that have higher scores are more likely to be good estimates of the true pose, they have a much higher chance of being selected to continue in the next update iteration.

Finally, once all the particles are updated, the one that yielded the highest score is output. The occupancy grid that represents the map features of that particle is published on the '/map' topic, and the corresponding pose is the algorithm's best determination of the current pose. However, instead of just publishing out the pose relative to '/map' frame, gmapping publishes a transform from '/odom' (child frame) to '/map' (parent frame), such that the odometry pose estimate aligns with the estimated pose output from gmapping.

I might be slightly misrepresenting the particular implementation of gmapping, and perhaps others could point out what I'm missing here, but this is the general idea behind particle filter SLAM algorithms.

#### **TL; DR**

So, to answer your question... it should definitely be base\_link (child) --> odom (parent), and that transform should be generated by some node that integrates encoders and/or inertial sensors to get a pose estimate.

gmapping needs this to guide the random sampling of the robot's pose when it updates the internal particle filter. The output from gmapping will be both the map itself, as well as a transform from odom (child) to map (parent) that serves to correct drift in the

odometry estimate to yield an accurate localization solution.

[link](#)

[add a comment](#)

---

## Your Answer

**Please start posting anonymously** - your entry will be published after you log in or create a new account.

[Add Answer](#)

ROS Answers is licensed under Creative Commons Attribution 3.0 Content on this site is licensed under a [Creative Commons Attribution Share Alike 3.0](#) license.



[about](#) | [faq](#) | [help](#) | [privacy policy](#) | [terms of service](#)  
Powered by Askbot version 0.10.2