Task Completed

- [x] Spawn at the origin of the environment
- [x] Teleoperate the robot and map the environment.

Task pending

Due to time constrains the following tasks were not completed.

- [ ] Scan the ArUco markers and store the robot's closest possible pose with respect to the marker when scanning the ArUco marker.
- [ ] Autonomously navigate to each ArUco marker
- [ ] Collision avoidance and control of the robot.

Ideology

Spawning of the robot happens as simply as launching the two launch files provided. `spawn_model` within `gazebo_ros` pacakge, the node responsible to launch the robot has been included within the `tortoisebotpromax_playground` itself. With the robot spawned the lidar data published on the topic `scan` by gazebo can be visualized using rviz.

Visualization of /scan topic

To generate the map, `gmapping` package was utilized which only has two requirements, `/scan` topic and transform between `/odom` and `base_link` frame.

Apart from gmapping, slam toolbox and cartographer package were also taken into consideration for mapping, but gmapping was chosen for its simplicity.

Generated map by gmapping package

Video Reference

Errors during Execution

First error encountered was the,

```
[ERROR] [1696795821.366083727]: material 'silver' is not unique.
```

After launching `tortoisebotpromax_playground.launch` before the issue was resolved an attempt was made to solved the issue by changing the `material.xacro` file within the `tortoisebotpromax_description` package.

From,

```
<material name="silver">
  <color rgba="0.700 0.700 0.700 1.000"/>
</material>
```

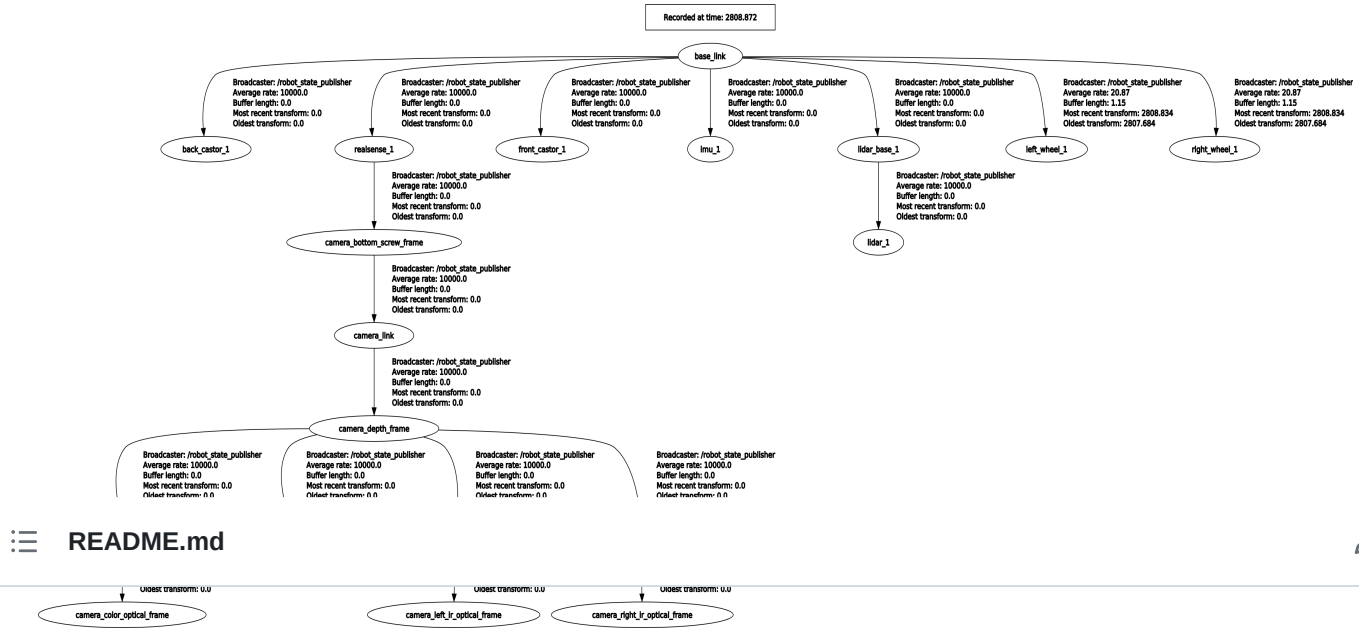
To,

```
<material name="grey">
  <color rgba="0.700 0.700 0.700 1.000"/>
```

```
</material>
```

After resolving this robot was spawned successfully within the gazebo and the `robot_description` was able to visualize in rviz.

Secondly, while generating the map of the environment, it was noticed that out of two things required for the `gmapping` to generate the map, which are `scan` topic where the lidar data is published and transform between `odom` and `base_link` frame which is used to identify the position of the robot from initial point. The transform was missing.

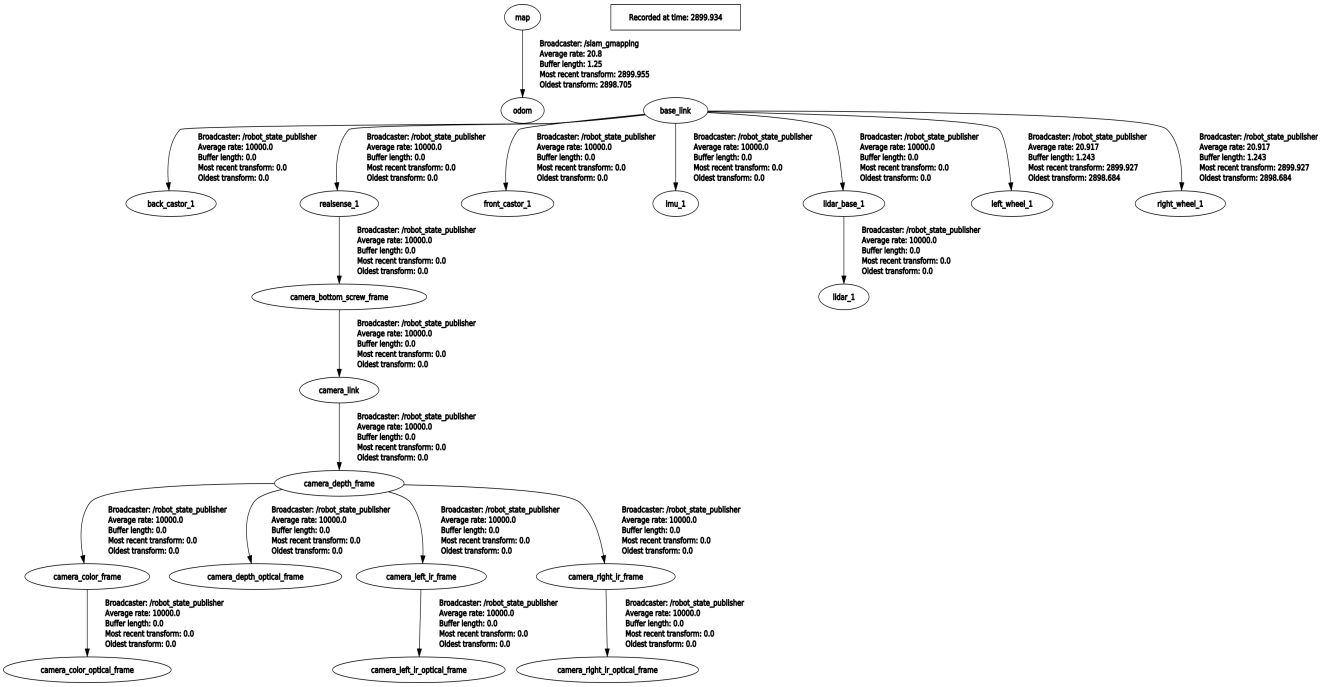


README.md

Missing /odom frame in the tf_tree

Before the error was resolved, a idea was presented to create a node which will subscribe to the topic `/odom` (which did exist) and then publish the transform so that the requirements of gmapping can be satisfied.

It was also found that if any frame is found missing by the gmapping package it initiates to publish to the frame itself.



No link between odom and base_link frame as gmapping publishes the frame which it should.

Before a node can created the issue was resolved.

Usage

Clone the repository into your src directory of ros workspace

```
cd ~/catkin_ws/src

git clone git@github.com:maker-ATOM/robonautica_mapping.git
```

Build the workspace

```
cd ~/catkin_ws
catkin_make
```

Launch gazebo using,

```
roslaunch tortoisebotpromax_gazebo tortoisebotpromax_playground.launch
```

Launch rviz to visualize the map,

```
roslaunch tortoisebotpromax_description display.launch
```

Launch gmapping to generate the map of the environment,

```
roslaunch robonautica_mapping gmapping.launch
```

Teloperate the robot using,

```
rosrun teleop_twist_keyboard teleop_twist_keyboard.py
```

As the robot is being controller using teleop node, generated map can be visualized in rviz.

To save the generated map,

```
rosrun map_server map_saver -f map
```

Note: All the above ros command should be executed in different terminal.

What next?

As the robot is teleoperated, a individual script will be running which detects the ArUco markers using `openCV`. After detection Ids will be assigned to each marker. The robot will keep on moving until the size of the maker sensed by the camera does reaches the mentioned threshold and the shape of the maker does not aligns to be square, inferring that the robot has aligned with the makers. At this position the pose of the robot will be stored in the form of waypoints.

One thing that actually concerns me is that does not this defies teh concept of autonomous navigation. Since the waypoints are detected by the robot operated in teleoperation mode and the robot is made to traversal those waypoint, there is a human intervention involved.

What I feel should actually happen is that, there should a navigation stack such as the `ROS Navigation Stack` which should be responsible to map the environment, localize the robot and navigate the environment while avoiding any obstacle within the path. The robot will receive goal positions to reached indicated using AcUro markers. Initially with no makers the robot will traverse the environment trying to visit the unvisited area of the environment until any AuRco makers is detected.