



ログイン

新規登録

トレンド

質問

公式イベント

公式コラム

募集

Organization

DX課題を「デザイン思考」で解き明かす！テクノプロ・デザイン社の魅力を知る —Qiita Zine



この記事は最終更新日から3年以上が経過しています。



@protocol1964 (Sota Watanabe)

ROS MelodicでTurtlebot3をGazeboで動かしてついでにSLAMする

ROS gmapping TurtleBot3

最終更新日 2019年05月17日 投稿日 2019年05月09日

まえがき

Gazebo上に置いたTurtlebot3でgmappingをやるための手順のメモ。

Turtlebot3とは？ → ROSが使える差動二輪ロボットプラットフォーム。

gmappingとは？ → 格子地図を生成するためのSLAMパッケージ。

gmappingは超簡単に言うと、tfでセンサの座標を見ながら/scanをsubscribeして/mapを作ってくれるすごいやつです。

Turtlebot3は自身のオドメトリから計算されるtfを更新しながら、搭載しているレーザセンサ(Laser Distance Sensor LDS-01)の値を/scanでpublishしています。

なので、Turtlebot3のgazeboシミュレータとgmappingのノードを動かせば地図が生成される。はず。

環境



40



36



- CPU: Intel core i7 3770
- GPU: Geforce RTX2070
- OS: Ubuntu 18.04 LTS
- ROS: Melodic

手順

前提

ROS Melodicのインストールが済んでいること。インストール方法は色々と情報が転がっていると思うので割愛。

GazeboとRVizも使うので、ROSインストール時に一緒に入ると思うけど一応動作を確認しておいてください。

```
$ gazebo
$ rviz
```

依存パッケージをインストール

多い

```
sudo apt-get install ros-melodic-joy ros-melodic-teleop-twist-joy ros-melodic-teleop-tw:
```

本当はgmappingもaptで入ってくれと嬉しいんだけど、残念ながらまだMelodic版はaptで入ってくれない(2019.5.9現在)ので、これは後々手動で追加します。

Turtlebot3の環境構築



40

36

```
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/src
$ catkin_init_workspace
$ cd ~/catkin_ws
```

Turtlebot3のROSパッケージをcloneします.

```
$ cd ~/catkin_ws/src
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3.git
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_gazebo_plugin.git
```

ビルド.

```
$ cd ~/catkin_ws
$ catkin_make
$ source ~/catkin_ws/devel/setup.bash
```

RVizで動作確認

環境変数でTurtlebot3のモデルを指定する必要があるので以下を実行してください.

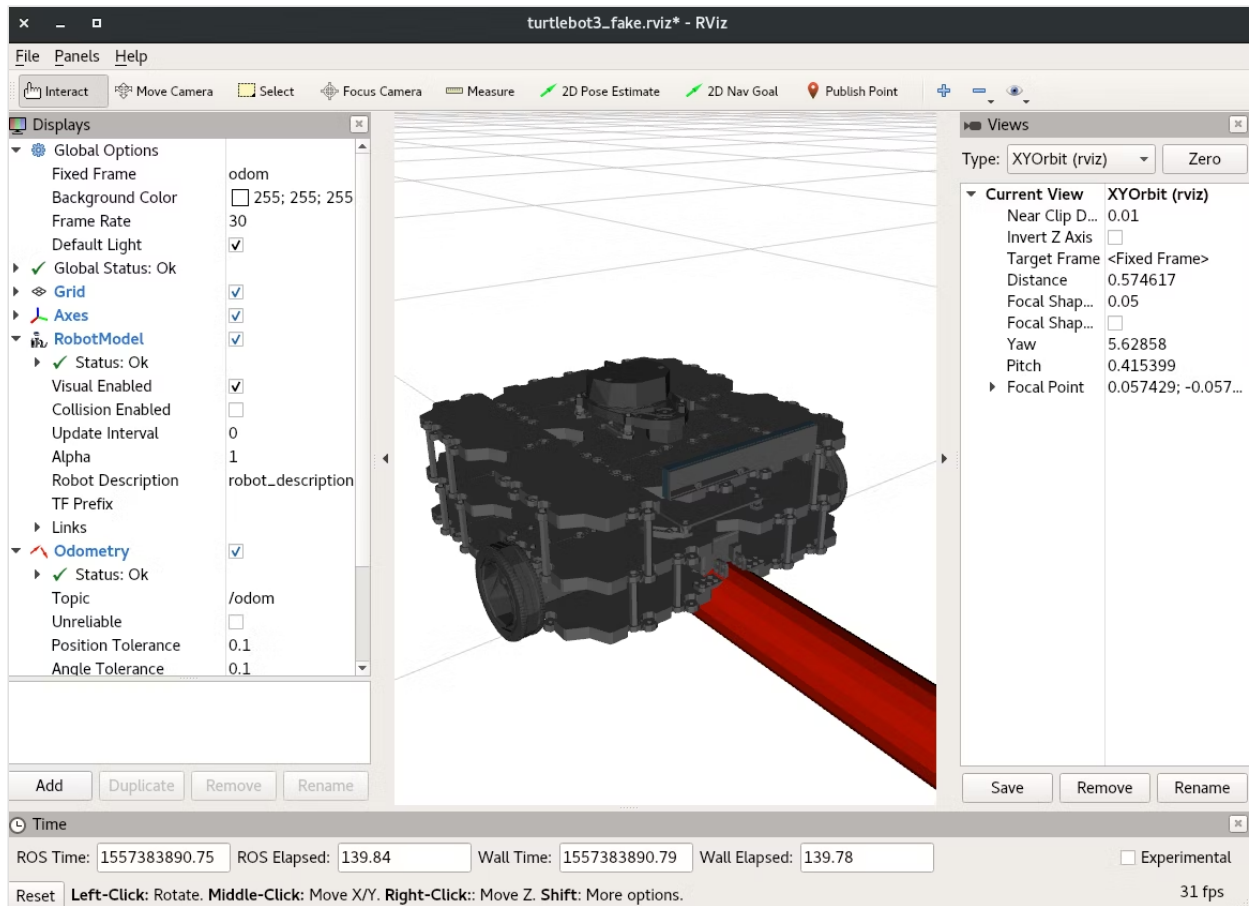
```
$ export TURTLEBOT3_MODEL=burger
```

モデルには burger の他に waffle と waffle_pi があります. 好きなやつを指定してください. bashrcとかに書きちゃってもいいと思う.

まずRVizで動作を確認します.

```
$ roslaunch turtlebot3_fake turtlebot3_fake.launch
```



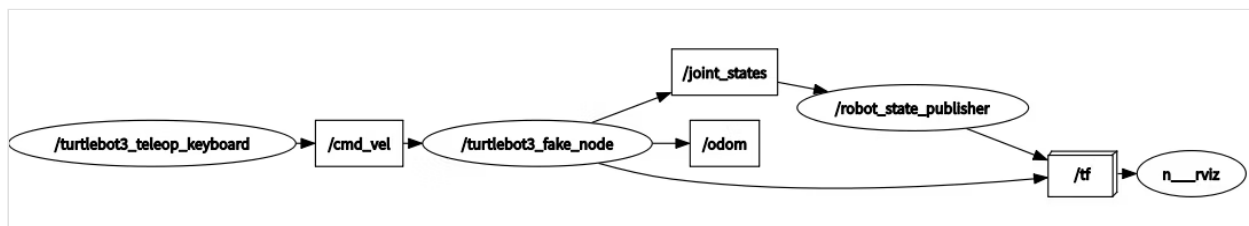


もしTurtlebot3のモデルが真っ白になってる場合は、他に起動しているroscoreを落としてもう一回RVizを立ち上げるとちゃんと表示されると思います。

/cmd_velでTwistを投げると動きます。rqtから手動で送ることもできるけど、キーボードで動かしたい場合はターミナルをもう一つ開いて、

```
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

でキーボードコントローラを実行できます。WASDXキーでRViz上のTurtlebot3が動きます。



ros graphを表示した図 キーボードコントローラから/cmd_velトピックがTurtlebot3

Gazeboで動作確認

RVizでTurtlebot3が動くことを確認したらGazeboで動かします。

RVizを閉じて以下のコマンドを実行してください。

```
$ roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

このとき、

```
terminate called after throwing an instance of 'std::runtime_error'
  what():  locale::facet::_S_create_c_locale name not valid
```

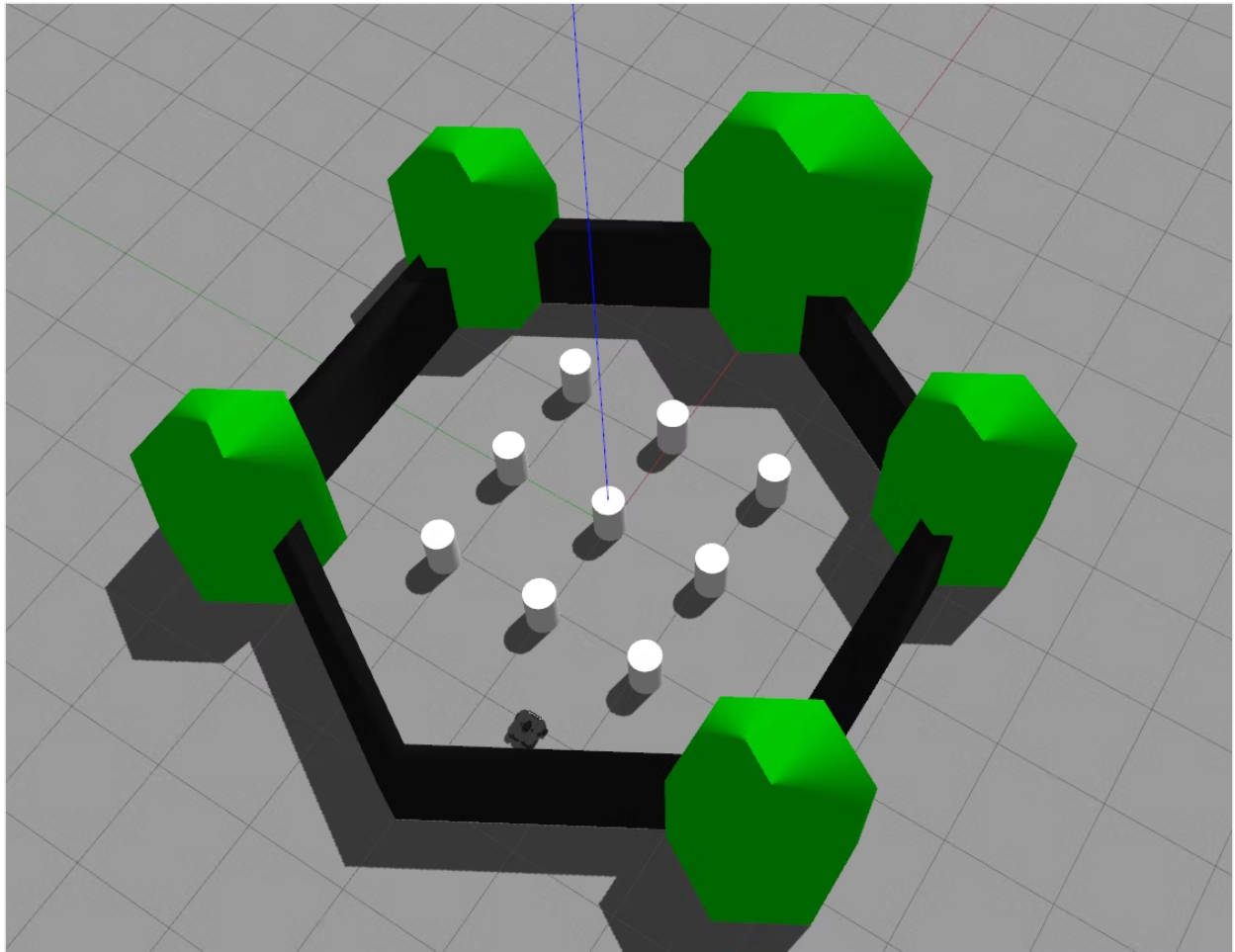
とか出てGazeboが起動しない場合は

```
$ export LC_ALL=C; unset LANGUAGE
```

とかやると動くと思います。



Gazeboが起動するとカメラ 📷 のようなワールドにTurtlebot3が配置されています。



先ほどRViz内のTurtlebot3を動かしたときと同様に、キーボードコントローラを用いて動かすことができます。

gmappingの環境構築

gmappingを用いたSLAMをするための準備をします。まず起動しているGazeboとキーボードコントローラを終了してください。

gmapping関連のパッケージをcloneします。

```
$ cd ~/catkin_ws/src
$ git clone https://github.com/ros-perception/slam_gmapping.git
$ git clone https://github.com/ros-perception/openslam_gmapping.git
$ git clone https://github.com/ros-planning/navigation_msgs.git
```



```
$ git clone https://github.com/ros-planning/navigation.git
$ git clone https://github.com/ros/geometry2.git
```

再度ビルド.

```
$ cd ~/catkin_ws
$ catkin_make
$ source ~/catkin_ws/devel/setup.bash
```

gmappingのビルドにそれなりに時間がかかるのでお茶 🍵 でも飲んで待ちましょう.

gmappingの動作確認

ビルドが終了したら先程と同様にGazeboとteleopコントローラを立ち上げます.

```
$ roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

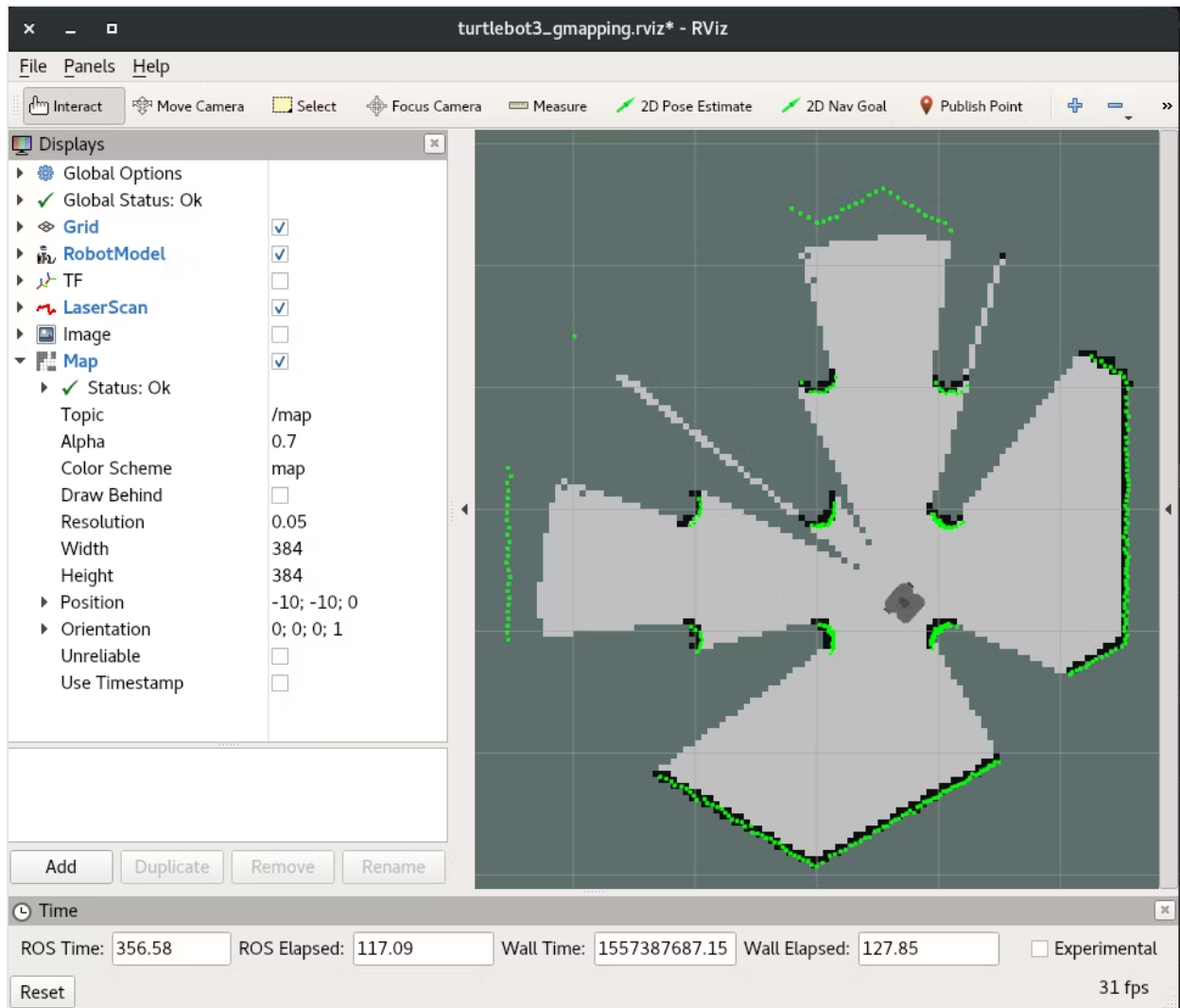
```
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

次にgmappingのノードを立ち上げます. Turtlebot3のパッケージにlaunchファイルが用意されているので, そちらを用います.

```
$ roslaunch turtlebot3_slam turtlebot3_slam.launch slam_methods:=gmapping
```

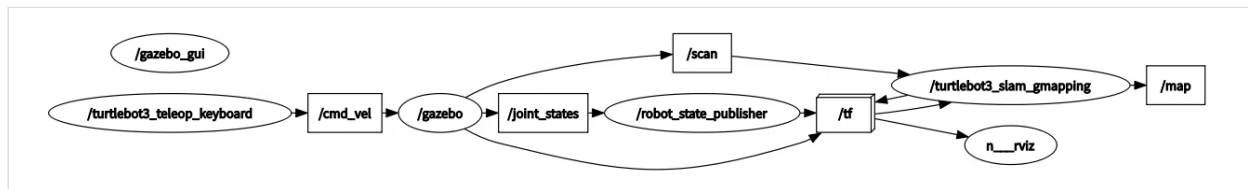
RVizが立ち上がり, Turtlebot3本体とセンサの取得データ, および生成されたmapデータが表示されていればOKです.





コントローラで動かして空間内を移動させていくと、地図が逐次更新されていくのがわかると思います。

rqt_graphで表示したノード間の関係です。



gazebo(Turtlebot3)がtfを更新しながら/scanをPublishし、gmappingノードがそれをSubscribeして/mapを出力しているのがわかります。

- [ROBOTIS e-Manual | TurtleBot3](#)
- [BRILLIANTSERVICE TECHNICAL BLOG | ROSで始めるロボティクス \(12\)](#)
- [ROS Wiki | gmapping](#)
- [ROSのSLAMを使った地図作成とナビゲーション](#)
- [技術メモ集 | ROS melodic\(Ubuntu18.04\)でmap_server, gmapping, navigationパッケージを導入する](#)



新規登録して、もっと便利にQiitaを使ってみよう

1. あなたにマッチした記事をお届けします
2. 便利な情報をあとで効率的に読み返せます

[ログインすると使える機能について](#)

新規登録

ログイン



40

36

関連記事

Recommended by



RaspberryPi3とZumoとROSで半永久自走式充放電ロボを作成したい_008日目_ SLAM...

by PINTO



ROSとGazeboによる自律移動シミュレーション ～モデル作成～

by Crafty_as_a_Fox



ラズパイ4/BusterにROSのmelodicを入れる

by Ninagawa_Izumi



40

36



ROS1/ROS2とRaspberryPi4でRealSenseD435を動かしてみた

by takarakasai



「入院を断られた!？」キーワードは【身元保証】

PR 一般社団法人 終活協議会



9割がフリーランス！最先端プロダクトを手がける精鋭プロフェッショナル集団「GNUS」

PR 株式会社GNUS

コメント

この記事にコメントはありません。

あなたもコメントしてみませんか :)

新規登録

すでにアカウントを持っている方は[ログイン](#)

How developers code is here.

© 2011-2023 Qiita Inc.

ガイドとヘルプ

About

コンテンツ

リリースノート

SNS

Qiita（キータ）公式



40

36

プライバシーポリシー	公式コラム	Qiita 人気の投稿
ガイドライン	募集	Qiita（キータ）公式
デザインガイドライン	アドベントカレンダー	
ご意見	Qiita 表彰プログラム	
ヘルプ	API	
広告掲載		
Qiita 関連サービス	運営	
Qiita Team	運営会社	
Qiita Jobs	採用情報	
Qiita Zine	Qiita Blog	
Qiita 公式ショップ		

