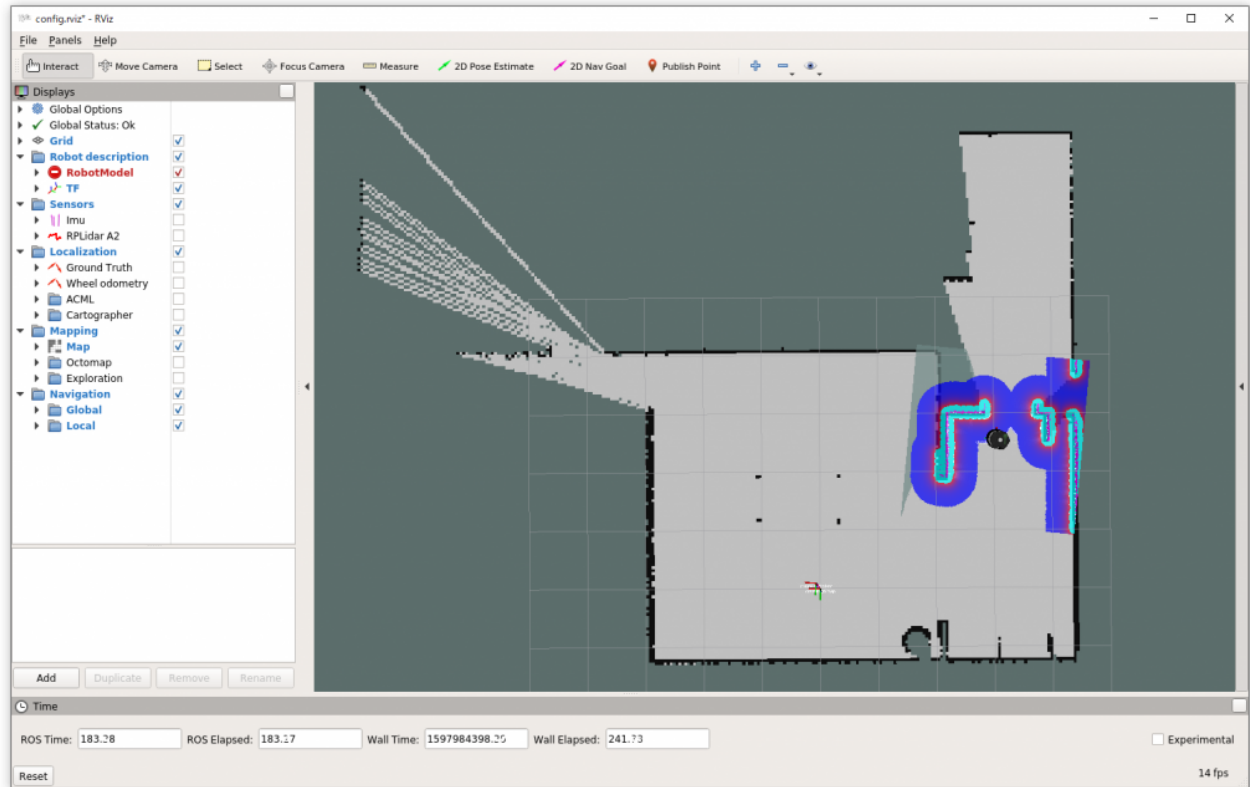


HARD2021：シミュレータで地図作成からナビゲーションまでしよう！

 demura.net/robot/hard/20061.html

2021.03.11



シミュレータGazeboを使って地図を作り、自己位置を推定し、目的地までナビゲーションする自律移動ロボットに必要な一連のタスクをさせましょう！

ここでは地図作成にgmappingパッケージ、自己位置推定にamclパッケージ、ナビゲーションにnavigationパッケージを使います。

参 考

https://github.com/RoboticaUtnFrba/create_autonomy/wiki

0. 準 備

エディタを使って~/.bashrcに以下があるか確認しなければ追加する。ワークショップ第2回目の準備をしていればあるはず。ない場合は、[\[HARD2021: シミュレータでルンバを動かそう!\]](#)を読んで作業を全部終えているか確認しよう。

```
export LASER=rplidar
export NUM_ROBOTS=1
export RVIZ=true
```

1. 地図作成

地図を作るために端末を3つ開く。

(1) GazeboとRvizの起動

1番目の端末で次のコマンドを実行すると、gazeboシミュレータとROSの視覚化ツールrvizが起動する。初めて、このコマンドを実行するときは必要な3Dモデルファイルをネットからダウンロードしてくるので少し時間がかかる。私の環境では数分かかった。

- `$ cd`
- `$ export LOCALIZATION=slam`
- `$ export RVIZ_CONFIG=navigation`
- `$ roslaunch ca_gazebo create_house.launch`
 - 最後の引数にはシミュレーション環境を指定する。create_house.launchは家のシミュレーション環境である。
~/catkin_ws/src/create_autonomy/ca_gazebo/launchにいろいろな環境がある。create_house.launchの代わりに、他のlaunchファイル（例えばcreate_restaurant.launch）などを指定すると違う環境でルンバを動かすことができる。
 - エラーが出る場合
 - エラー1：Gazeboを起動しても以下のエラーで表示され画面が真っ暗な場合
[Err] [REST.cc:205] Error in REST request
libcurl: (51) SSL: no alternative certificate subject name matches target host name 'api.ignitionfuel.org'
~/.ignition/fuel/config.yamlを次のように変更する。
 - 変更前： `url: https://api.ignitionfuel.org`
 - 変更後： `url: https://api.ignitionrobotics.org`
 - エラー2：rvizがOpenGL関係のエラーで落ちる場合は、以下の1行を.bashrcの最後に付け加える。
`export LIBGL_ALWAYS_SOFTWARE=""`

(2) 地図作成

次にロボットをキー操作し動かしながら地図を作っていこう。地図の作成にはルンバに搭載されているLIDARの情報を使う。初期の設定ではGazeboとRviz上の物体が180度ずれているので、Rvizの画面をマウスでドラッグして向きを揃えると作業がやりやすくなる。

2番目の端末を開き、キー操作でロボットを動かすとrviz上で地図が生成されるのがわかる。ロボットを動かして各部屋の地図を作ろう。注意点としてはロボットを壁などに衝突させないこと。衝突するとロボットの位置がずれて、地図も不正確な地図になってしまう。

```
$ roslaunch ca_tools keyboard_teleop.launch
```

(3) 地図の保存

地図は自動的に保存されないので、3番目の端末を開き、以下のコマンドを実行して保存する。ここで、mymapは保存する地図名なので適宜変更する。

このコマンドを実行したディレクトリに画像ファイルのmymap.pgmと設定ファイルのmymap.yamlの2つのファイルが作られる。設定ファイルには画像ファイル名、解像度、占有領域はフリースペースなどの情報がテキスト形式で保存されている。通常、あまり変更しない。画像ファイルは、ノイズなどのない綺麗な地図にするため画像ファイルはGimpなどの画像編集ソフトで編集するが多い。

```
$ rosrun map_server map_saver -f mymap
```

2. ナビゲーション

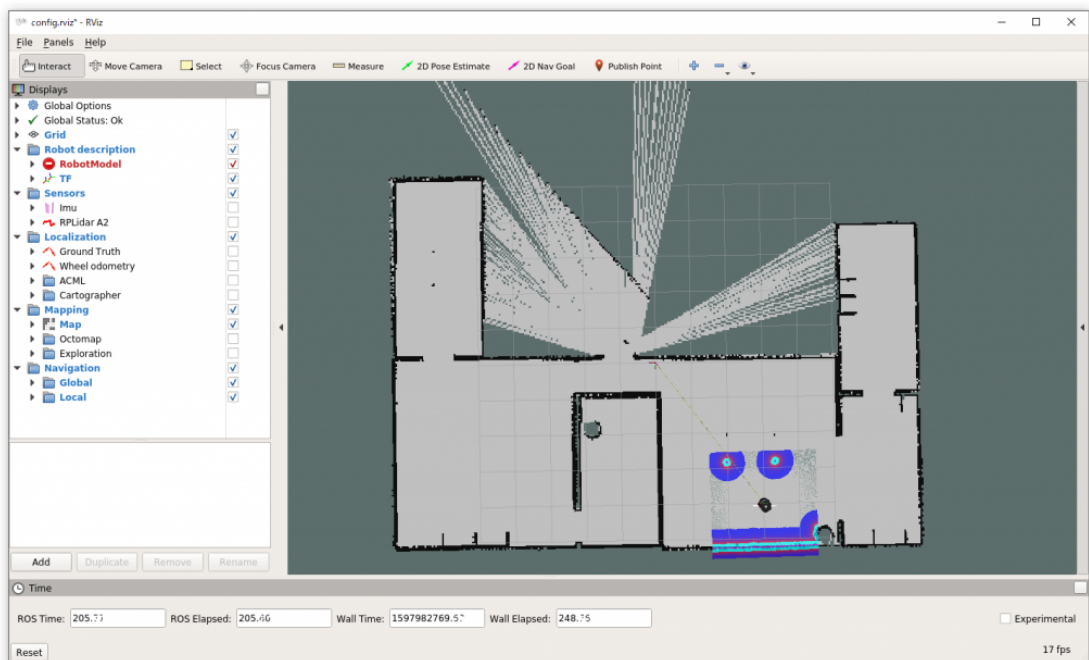
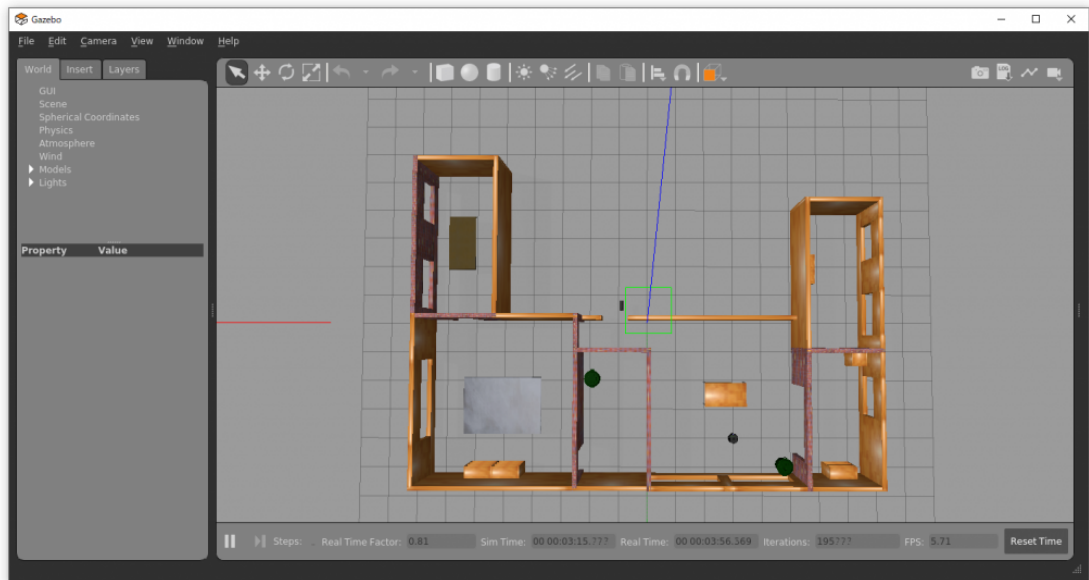
一度開いている端末を全部閉じ、実行中のノードを終了させる。新たに端末を2つ開く。本来なら上で作成した地図を使ってナビゲーションをするが、ここでは予めcreate_autonomyに用意されている地図を使う。~/catkin_ws/src/create_autonomy/ca_gazebo/launchに準備されている環境にはすべて地図がある。

(1)1番目の端末で以下のコマンドを実行してGazeboシミュレータとRvizを起動する。

- ```
$ export LOCALIZATION=amcl
```

- 最後の引数にはナビゲーションしたい環境のlaunchファイルを指定する。実行すると以下のGazeboとRvizが起動する。

```
$ roslaunch ca_gazebo create_house.launch
```



## (2) ナビゲーションの実行

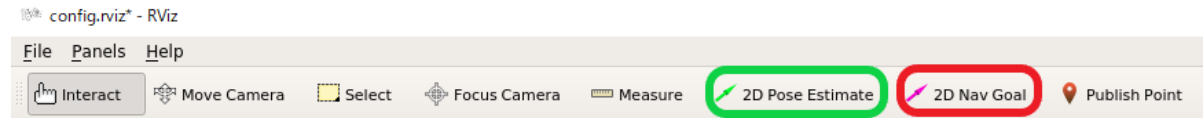
### • Launchファイルの実行

2番目の端末で以下のコマンドを実行し、このナビゲーション用のsend\_goal.launchファイルを起動する。

```
$ roslaunch ca_move_base send_goal.launch
```

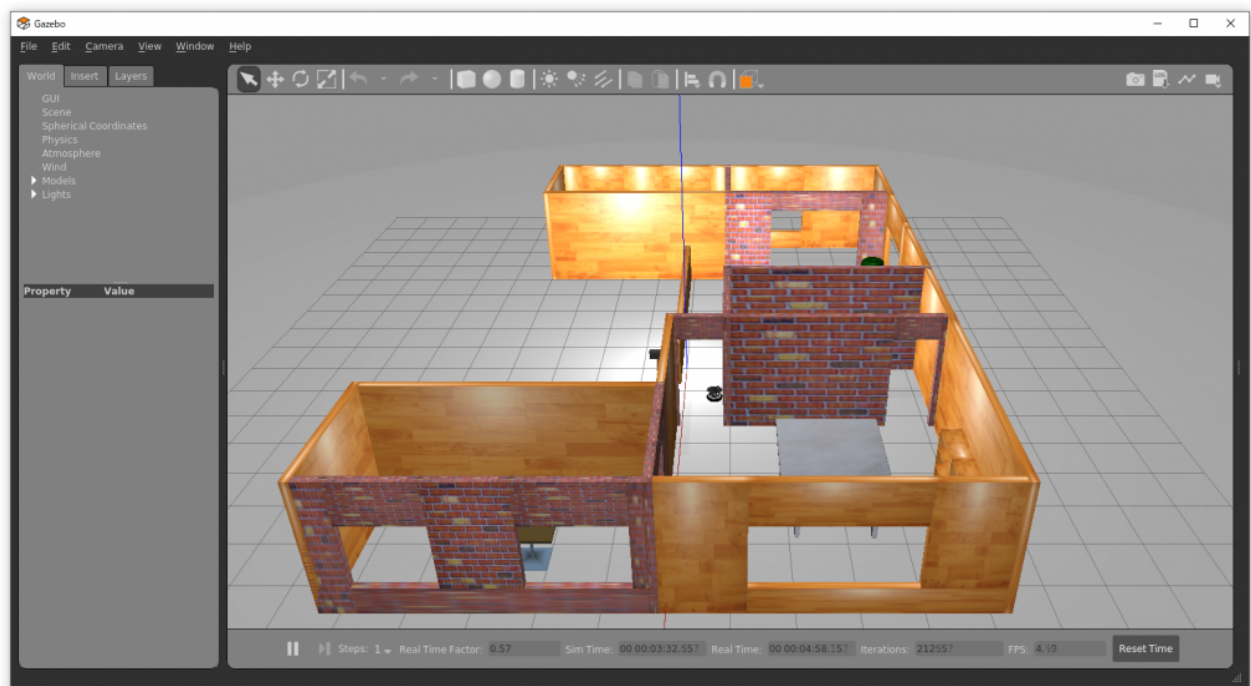
- **ロボット初期位置の確認と設定**

次に、GazeboとRviz上でのロボットの位置が同じか確認する。ずれている場合は、起動したRvizメニューバー（下図）の緑枠で囲んだ”2D Pose Estimation”をクリックして選択し、ロボットの姿勢と位置が合うようにロボットの位置と向きを設定する。ロボットの位置にマウスのカーソルを持って行き、ロボットの向きにドラッグすると設定終わり。



- **ゴール設定とナビゲーション**

最後に上図の赤枠で囲んだ”2D Nav Goal”をクリックして、初期位置を設定したのと同様に、ゴールの位置と向きを設定する。設定するとロボットがナビゲーションを開始し、途中に障害物があればそれを避けてゴールまで向かい、設定した位置と向きを向いて止まる。



### 3. ハンズオン

- **ベーシック**

create\_house.launch以外のシミュレーション環境を使って、地図生成とナビゲーションをやってみよう！

- アドバンスト
  - この例ではマッピングにslam\_gmappingパッケージを使いましたが、create\_autonomyパッケージには次に示すいろいろなマッピング手法を使うことができます。
    - cartographer
    - hector\_slam
    - octomap\_mapping
    - rtabmap\_ros
    - slam\_gmapping
  - 各手法の概要を調べて、実施に動かして違いを体験してみよう！使い方は次のリンクです。  
[Mapping and Exploration](#)

終わり