



بسمه تعالی
دانشگاه صنعتی امیرکبیر دپارتمان مهندسی کامپیوتر
درس اصول علم ربات نیمسال دوم سال تحصیلی ۱۴۰۱-۱۴۰۲
تمرین سری دوم



انجام این تمرین به صورت انفرادی می باشد

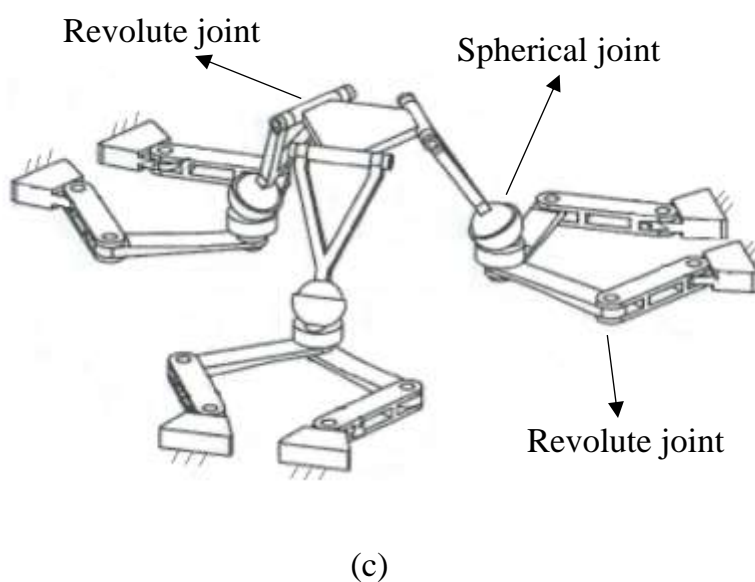
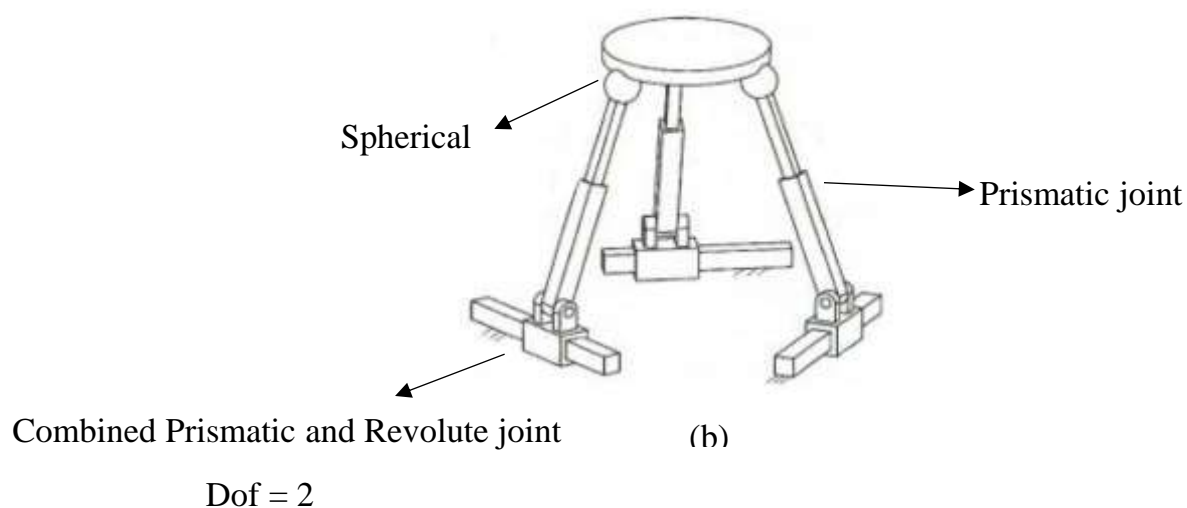
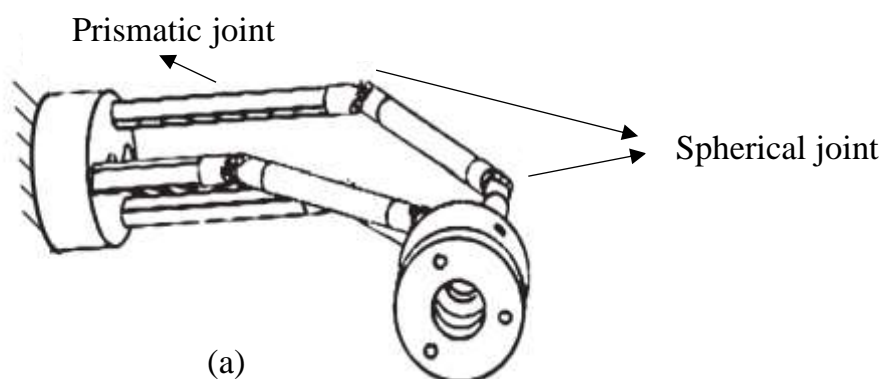
هدف از انجام این تمرین

هدف این تمرین مرور برخی از مفاهیم مطرح شده در کلاس درس و همچنین یادگیری کار با ربات در محیط های شبیه سازی gazebo و RViz، به حرکت درآوردن ربات با استفاده از کیبورد و کد و همچنین توانایی کار با سنسور Lidar است. برای انجام بخش پیاده سازی این تمرین، بهتر است در ابتدا ویدیو های ۳،۴ و ۵ را مشاهده نمایید.

بخش های تئوری:

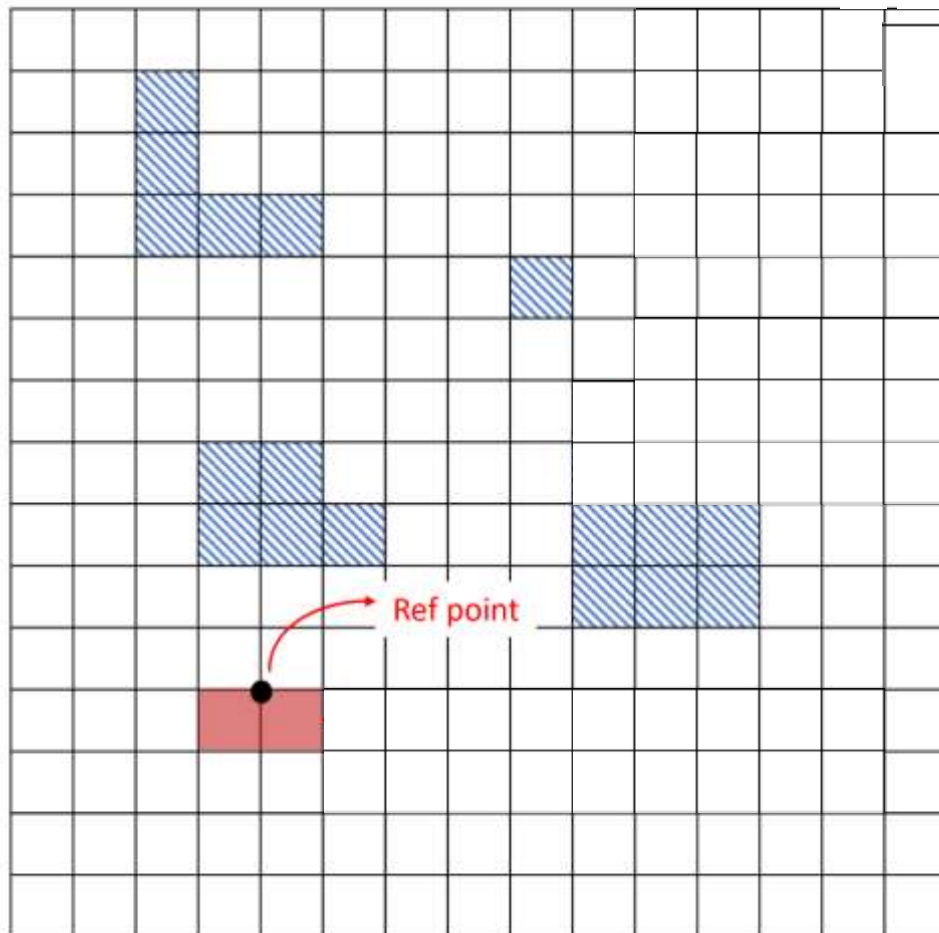
۱.

الف) درجه آزادی شکل های زیر را حساب کنید. (از فرمول Grubler استفاده کنید و هر یک از اجزای فرمول را به صورت کامل بنویسید)



ب) بیشینه درجه آزادی اشیائی که روی یک صفحه حرکت میکنند چقدر است؟ توضیح دهید.

۲. برای ربات مشخص شده در تصویر زیر، C-space را مشخص کنید. (فرض کنید که ربات تنها امکان دوران ۹۰ درجه را دارد)



۳. تفاوت ربات های holonomic و non-holonomic را شرح دهید.

بخش های شبیه سازی:

❖ گام اول :

این گام در محیط **empty map** انجام می شود.

در این گام، با سرویس ها آشنا شده و ربات را در Gazebo به حرکت در می آوریم. در آغاز، به یک custom service به نام **GetNextDestination** نیاز داریم. این سرویس می بایست position کنونی ربات را دریافت کرده و مقصد بعدی را برگرداند. یک روش می تواند به شکل زیر باشد:

```
float64 current_x
float64 current_y
---
float64 next_x
float64 next_y
```

در این گام به دو نود نیاز داریم:

۱. **Mission Node**: این نود می بایست پاسخگوی سرویس **GetNextDestination** باشد، به گونه ای که برای هر درخواست، به صورت رندوم دو عدد در بازه ی ۱۰- تا ۱۰+ به عنوان X و Y مقصد بعدی بسازد. توجه کنید این نقطه باید با محل کنونی ربات دست کم ۵ متر فاصله داشته باشد.

۲. **Control Node**: این نود وظیفه ی کنترل ربات را بر عهده دارد. فرض کنید که ربات ما دو state دارد:

۱. حرکت به سمت جلو با سرعت خطی ثابت

۲. دوران در حالت ایستاده با سرعت زاویه های دلخواه

در آغاز کار، ربات باید درخواستی به سرویس **GetNextDestination** بفرستد. پس از دریافت مقصد بعدی به اندازه ی مناسب دوران کرده تا هدف دقیقاً روبروی ربات قرار گیرد، سپس مستقیم به سمت هدف حرکت کند. پس از رسیدن به مقصد ایستاده و دوباره درخواستی به **GetNextDestination** بفرستد و این فرایند را تا دست کم ۵ بار ادامه دهد.

برنامه شما در این بخش باید قابلیت گرفتن سرعت خطی را از ورودی سمت کاربر داشته باشد. میتوانید از تگ **param** در **launch file** خود استفاده کنید. نقطه شروع ربات را بر روی مختصات (۰, ۰) قرار دهید. سپس به ازای سرعت خطی های ۲، ۴ و ۸ متر بر ثانیه اجرا بگیرید.

موارد زیر را در گزارش خود قرار دهید:

۱. ویدیوی کارکرد ربات برای هر سرعت خطی.
۲. نمای مسیر ربات در شبیه ساز Rviz به ازای هر سرعت خطی.
۳. خطای انحراف از مقصد به ازای هر سرعت خطی. برای این کار می توانید پس از ایست ربات فاصله اش را تامقصد کنونی به دست آورده و درنهایت میانگین فواصل را گزارش دهید.

❖ گام دوم:

این گام در محیط obstacle-detect.world که در فایل تمرین قرار داده شده انجام می شود.

هدف از این بخش آشنایی با سنسور LiDAR و استفاده از تاپیک LaserScan است. این گام نیازمند دو نود است:

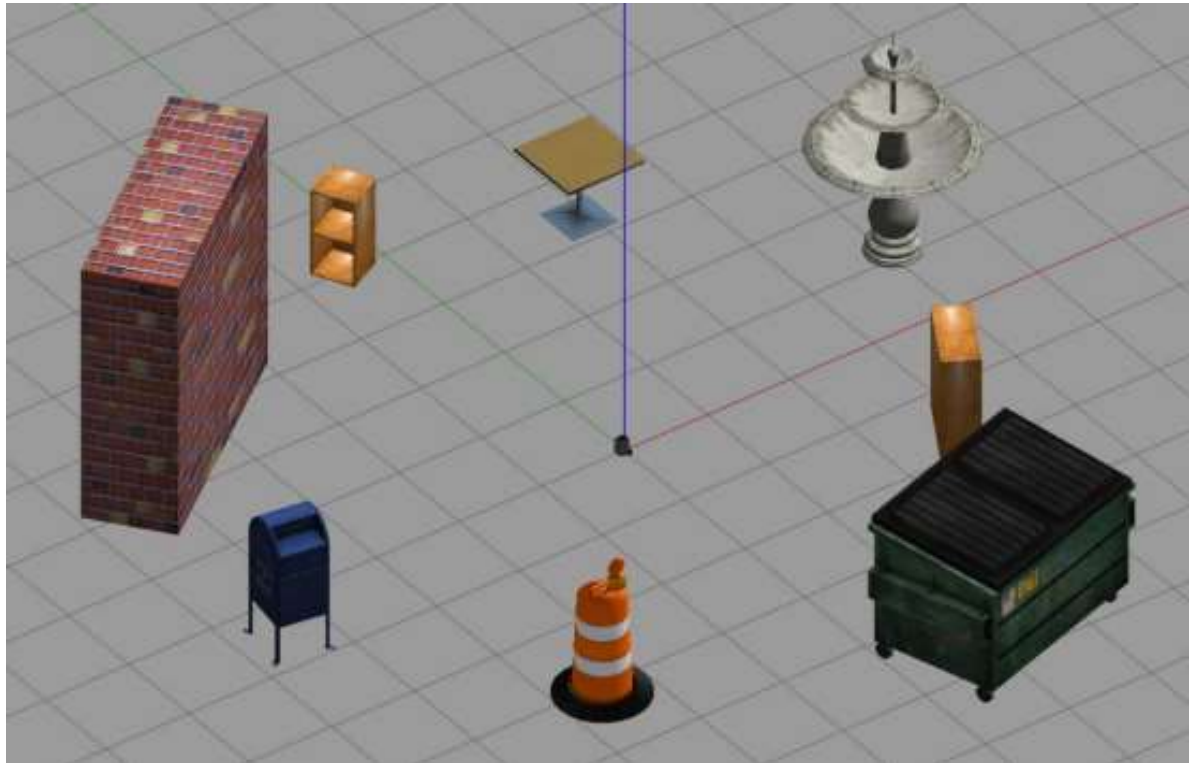
۱. Sensor Node : این نود باید به فاصله ی ربات را تا موانع با استفاده از سنسور LiDAR و تاپیک LaserScan بخواند و همواره مشخصات نزدیک ترین مانع را شامل فاصله و زاویه بر روی تاپیک ClosestObstacle بفرستد. این تاپیک نیز به یک custom message با دو متغیر مانند شکل زیر نیاز دارد .

```
float64 distnace  
float64 direction
```

با استفاده از این سنسور می خواهیم زاویه چرخش مناسب ربات را برای فاصله گرفتن از نزدیکترین مانع محاسبه کنیم. به عنوان مثال اگر کمترین مقدار فاصله متعلق به ایندکس ۳۰ام آرایه ی ranges خوانده شده از LiDAR باشد، ربات باید ۱۵۰ درجه به سمت راست خود بچرخد.

۲. Control Node : این نود حرکت ربات را کنترل می کند. مانند گام پیشین، در اینجا نیز حرکت رو به جلو و چرخش را داریم. ربات باید همواره رو به جلو حرکت کند تا فاصله اش از مانعی کمتر از ۲ متر شود. سپس در جهت مناسب به اندازه ای بچرخد تا مانع دقیقا پشت سرش قرار گیرد. سپس به حرکت مستقیم خود ادامه دهد. برای این کار نود کنترل می بایست از روی تاپیک ClosestObstacle تولید شده در نود سنسور subscribe نماید و از مقدار دو متغیر بالا به عنوان فاصله تا نزدیک ترین مانع و زاویه ی مانع استفاده نماید.

برای این گام نقطه‌ی شروع ربات را $(0, 0)$ قرار داده و yaw angle آن را 0.75 بگذارید.



obstacle-detect.world – Second step word map

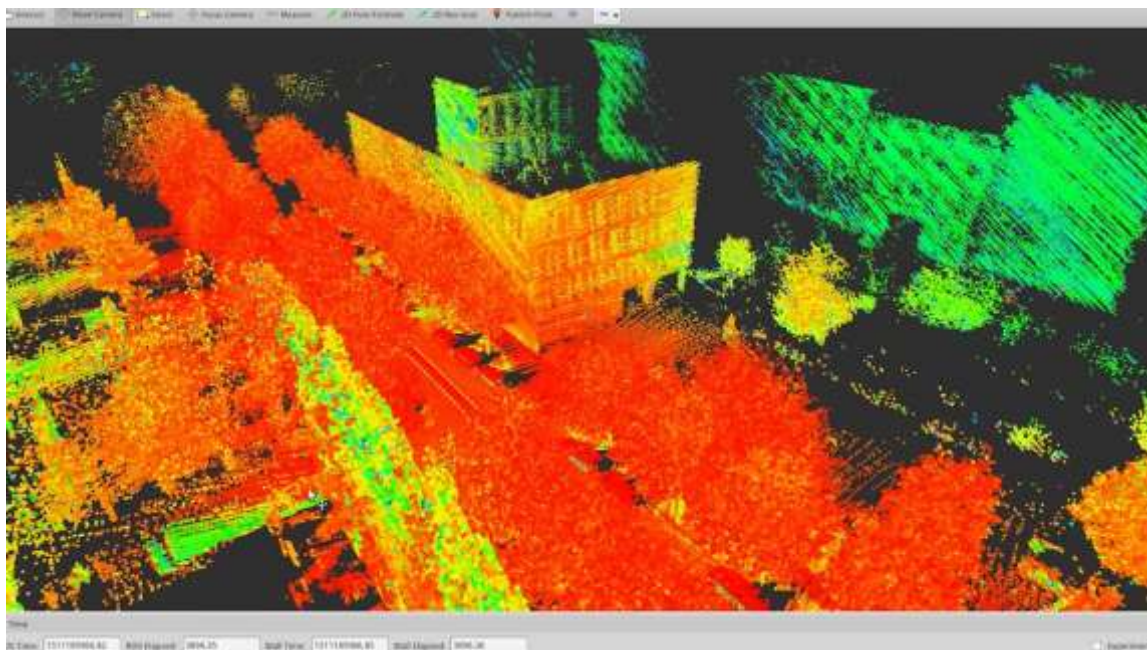
موارد زیر را در گزارش خود قرار دهید:

۱. ویدیوی کارکرد ربات

۲. نمای مسیر ربات در شبیه ساز Rviz

❖ گام سوم:

هدف از این گام آشنایی با Pointcloud است. Pointcloud داده ای است از نقاط در فضای سه بعدی که نمایشی از محیط پیرامون به دست می دهد. یکی از راه های visualise کردن بهتر داده های LiDAR استفاده از این روش می باشد.



بخش اول:

پکیجی به نام pcd_view بسازید. یک launch file در آن گذاشته و launch file های turtlebot3_stage_4.launch و turtlebot3_gazebo_rviz.launch را به روش زیر در فایل خود include نمایید.

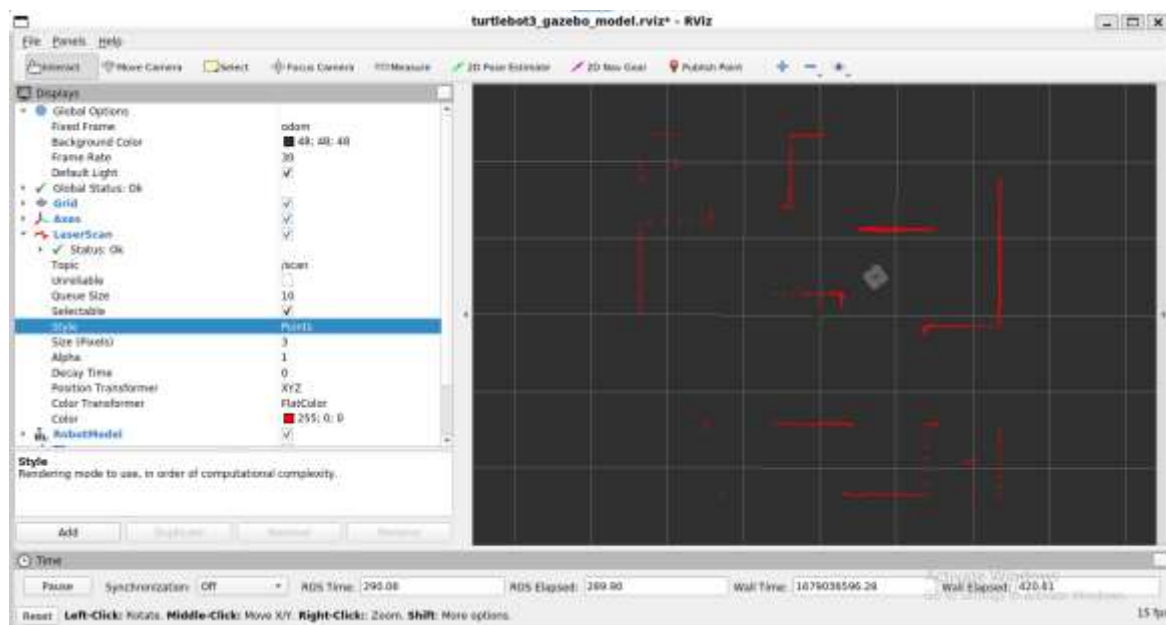
```
<include file="$(find turtlebot3_gazebo)/launch/turtlebot3_stage_4.launch"/>
```

پس از catkin_make و source کردن workspace خود، مدل ربات را مشخص کنید:

```
export TURTLEBOT3_MODEL=waffle
```

سپس با دستور roslaunch پکیج خود را اجرا کنید. با این دستور Gazebo و RViz اجرا می شوند. در یک ترمینال دیگر turtlebot3_teleop_key.launch را از پکیج turtlebot3_teleop ران کنید و ربات را با کیبورد کنترل نمایید.

آنچه در RViz دیده می شود داده های لحظه ای LiDAR به دست آمده از ربات هستند (برای نمای بهتر می توانید از منوی سمت چپ، Style مربوط به LaserScan را به Points تغییر دهید).



موارد زیر را در گزارش خود قرار دهید:

۱. تصویری از کارکرد ربات

۲. دلیل لرزش نقاط داده در شبیه ساز چیست؟ دلیل خود را در گزارش تمرین بیاورید.

بخش دوم:

در این بخش از داده ی LaserScan به PointCloud می رسیم.

۱. برای این کار به سرویس laser_scan_assembler از پکیج laser_assembler نیاز داریم. این سرویس داده های مربوط به LaserScan را جمع آوری کرده و مجموع آنها را در فرمت یک پیام PointCloud2 از پکیج sensor_msgs میگرداند. دستور اجرای نود این سرویس را به انتهای launch file خود اضافه کنید.

دقت کنید که این نود دو پارامتر دریافت می کند:

۱. max_scans : که تعداد بیشینه اسکن ها را در یک پیام PointCloud2 مشخص می کند. مقدار آنرا

"۴۰۰" بگذارید.

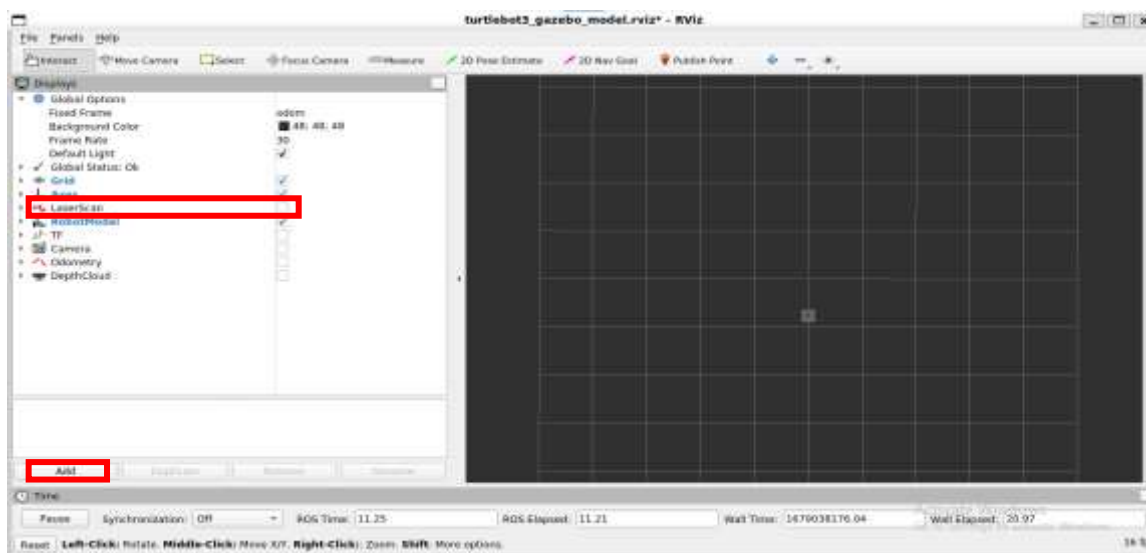
۲. fixed_frame : که فریم publish کردن پیام ها مشخص می کند. برای کاربرد ما این مقدار را "odom" ست کنید.

۲. همچنین ما به یک نود برای فرستادن درخواست به سرویس بالا، دریافت و publish کردن پیام PointCloud2 نیاز داریم. برای نوشتن این نود از AssembleScan2 از پکیج laser_assembler استفاده نمایید.

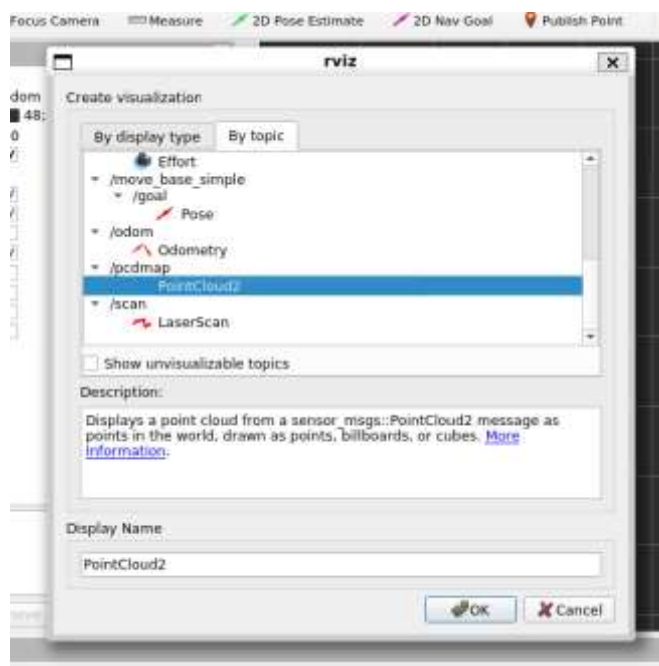
برای اضافه کردن سرویس به launch file و نوشتن نود خود می توانید از این لینک و ویدیو ی آن بهره ببرید:

<https://www.theconstructsim.com/merge-laser-scans-single-pointcloud/>

در RViz از منوی سمت چپ با Uncheck کردن LaserScan، آن را Disable نمایید. سپس گزینه ی Add را انتخاب کنید.

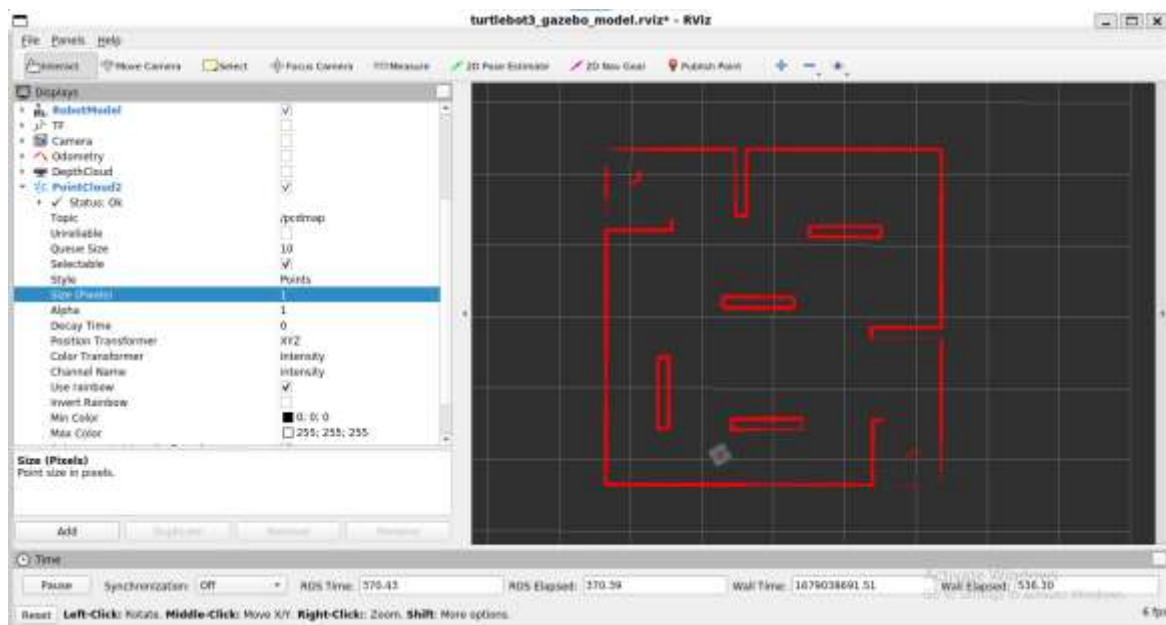


در منوی باز شده، در بخش By Topic بر روی PointCloud2 کلیک نمایید.



پس از این کار RViz داده های دریافتی از تاپیک مربوط به PointCloud2 را نمایش می دهد (دوباره برای نمای بهتر می توانید از منوی سمت چپ، Style مربوط به PointCloud2 را به Points تغییر دهید).

اکنون در یک ترمینال دیگر turtlebot3_teleop_key.launch را از پکیج turtlebot3_teleop ران کنید و ربات را با کیبورد کنترل نمایید. با حرکت ربات می توانید نقشه ای از محیط بسازید و کامل نمایید.



موارد زیر را در گزارش خود قرار دهید:

۱. ویدیوی کارکرد ربات

۲. شکل نهایی

❖ نحوه تحویل

فایل zip نهایی شما باید شامل موارد زیر با نام گذاری دقیقاً مشابه باشد. جواب سوال های تئوری را به طور جدا در فایل pdf با نام theory و فایل گزارش کار بخش عملی را در قالب pdf در فایلی با نام report قرار دهید. کل فولدر پکیج های مربوط به این تمرین را درون فولدري به نام packages قرار دهید. مجموعاً ۵ فیلم برای این تمرین باید تهیه شود. ۵ ویدیو را به طور جدا با نام گذاری نوشته شده در پایین درون فولدر video قرار دهید.

HW2_StudentNumber.zip

|__ theory.pdf

|__ report.pdf

|__ packages Folder

|__ video Folder

|__ first step 2 m/s

|__ first step 4 m/s

|__ first step 8 m/s

|__ second step

|__ third step

نکات تکمیلی در باب تحویل تمرین

۱. مدت زمان هر فیلم باید ۱ دقیقه باشد. در مجموع ۵ دقیقه فیلم برای این تمرین. حجم فایل تحویلی نهایتاً باید ۱۰۰ مگابایت باشد. برای کاهش حجم ویدیو های ضبط شده می‌توانید از نرم افزار handbrake بهره ببرید.

۲. فرمت فایل زیپ نیز به صورت HW2_StudentNumber باشد.

۳. افراد می‌بایست تمرین را به صورت انفرادی انجام دهند.

۴. دستیاران آموزشی ملزم به اجرا کردن کدهای شما نیستند. بنابراین هرگونه نتیجه و یا تحلیلی که در شرح سوال از شما خواسته شده است را به طور واضح و کامل در گزارش بیاورید. در صورت عدم رعایت این مورد، بدیهی است که از نمره تمرین کسر می‌شود.

۵. تحویل کد به همراه تمرین لازم است و در صورت تحویل ندادن کد و اکتفا به گزارش، نمره‌ی آن بخش به طور کامل کسر می‌شود.

۶. تاریخ تحویل تمرین ۱۴۰۱/۱۲/۱۹، ۱۱:۵۹ شب، می‌باشد و سیاست‌های تاخیر مطابق با موارد ذکر شده در شیوه‌نامه لحاظ خواهد شد.