

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 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.

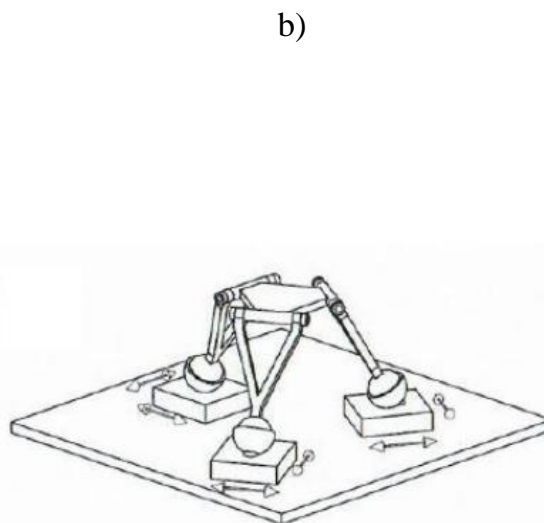
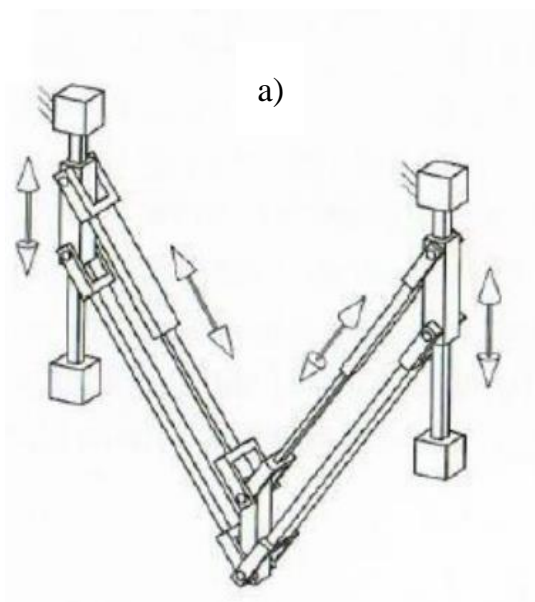
۱.

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. همچنین به نکات زیر توجه کنید و به سوالات پاسخ کامل دهید.) (۲۰ نمره)

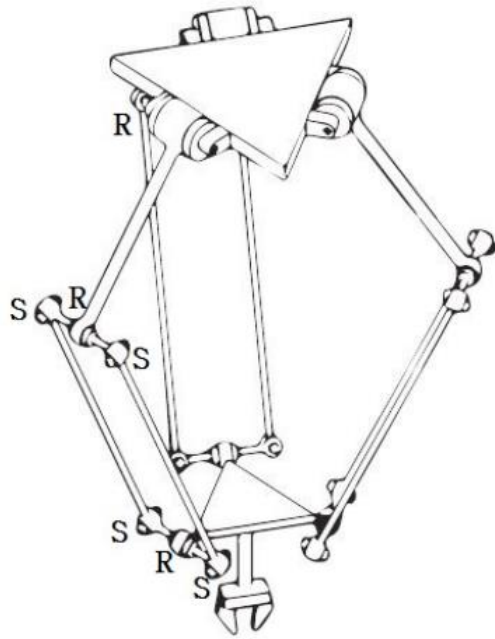
• شکل b یک مکانیزم 3-PPSR را نشان می دهد که در آن جعبه ها فقط می توانند در جهت فلش ها بلغزند. آنها نمی توانند بچرخند.

• ربات دلتا در شکل c از دو پلتفرم تشکیل شده است - پلتفرم پایینی متحرک است و قسمت بالایی ثابت است. این دو پلتفرم، توسط سه پایه یکسان به هم متصل شده اند: هر پایه یک زنجیره سریال RR است که به یک حلقه بسته متصل است. پیوند متوازی الاضلاع (هر یک از انواع اتصالات در شکل نشان داده شده است). این ربات دلتا ۳ درجه آزادی دارد. برای یافتن درجات آزادی این ربات از فرمول Grubler استفاده کنید. آیا پاسخ شما ۳ است؟ اگر نه، توضیح دهید که چرا فرمول گروبلر شکست می خورد.

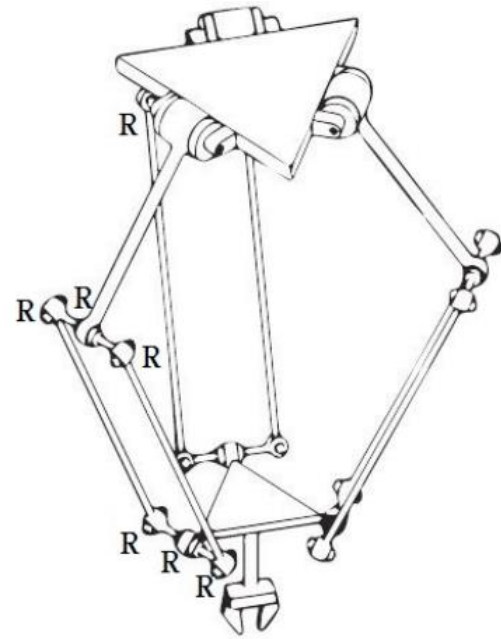
• شکل d نسخه دیگری از ربات دلتا را نشان می دهد که در آن مفاصل S اکنون با اتصالات R جایگزین شده اند. برای یافتن درجات آزادی این ربات از فرمول Grubler استفاده کنید. آیا این ربات می تواند حرکت کند؟ پاسخ خود را شامل تمام فرضیات بیان شده توضیح دهید.



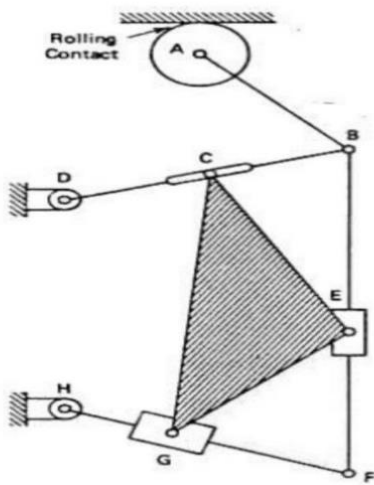
c)



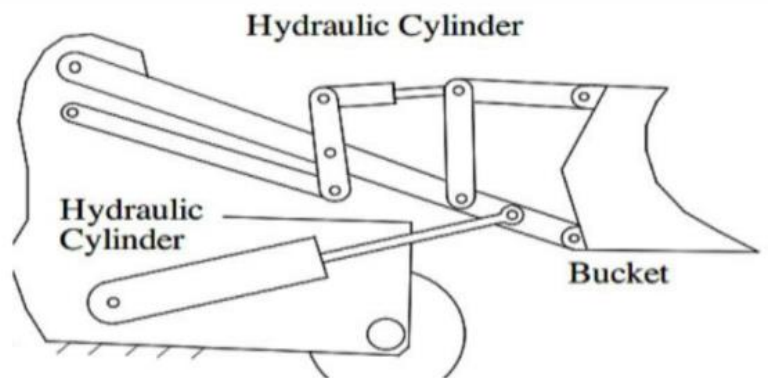
d)



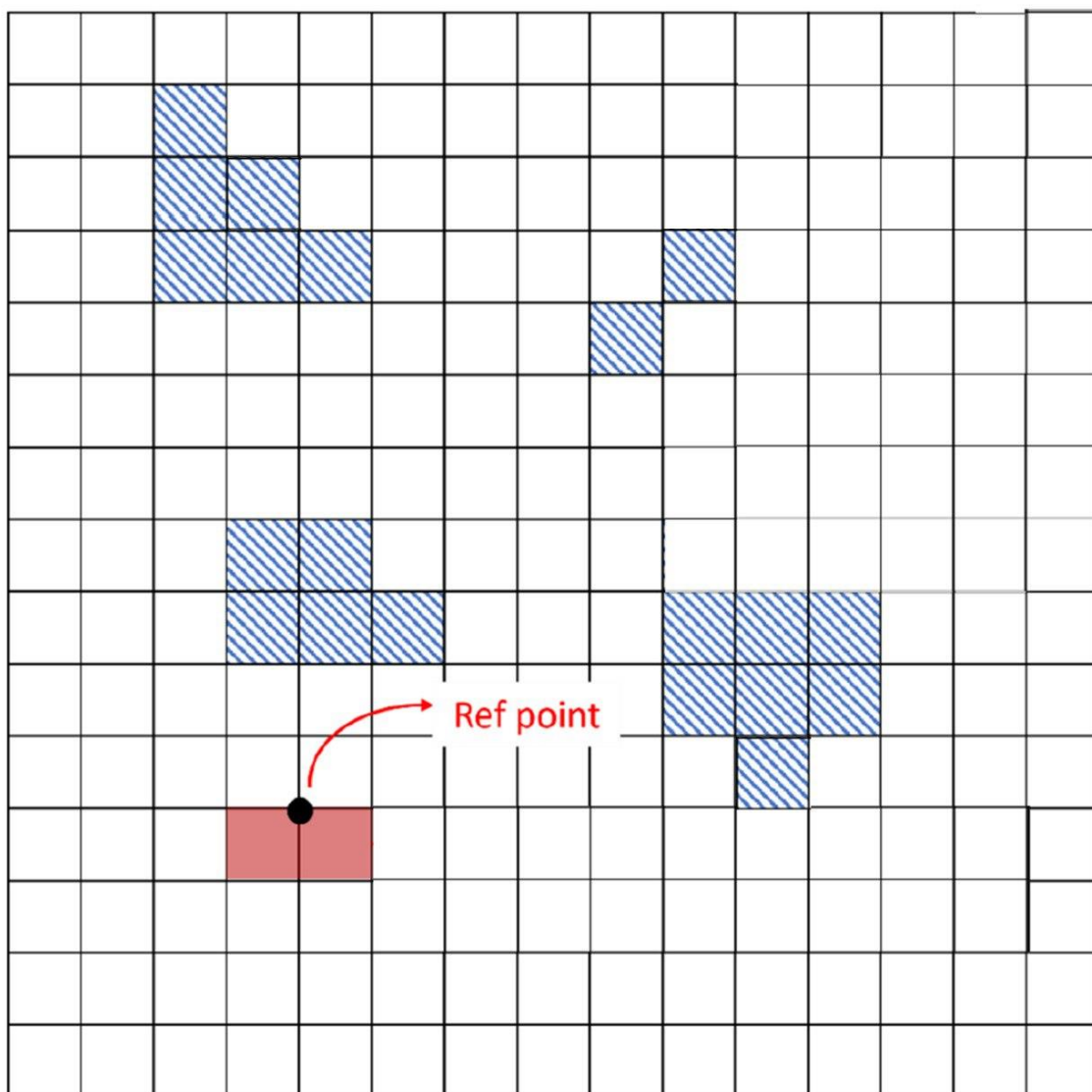
e)



f)



۲. برای ربات مشخص شده در تصویر زیر، C-space را مشخص کنید. (فرض کنید که ربات تنها امکان دوران ۹۰ درجه را دارد) (۵ نمره)



بخش های شبیه سازی:

❖ گام اول (۶۰ نمره) :

در این گام، ابتدا با سرویس ها آشنا می شویم. در آغاز، به یک custom service به نام GetNextDestination نیاز داریم. این سرویس می بایست مقصد بعدی را از یک فایل text خوانده و برگرداند. یک روش می تواند به شکل زیر باشد:

```
float64 next_x  
float64 next_y
```

از آنجایی که نقشه خالی نیست و مقصدهای انتخابی، موقعیت های موانع درون نقشه هستند، در ادامه باید با استفاده از سنسور LiDAR و تاپیک LaserScan قبل از رسیدن به مانع، توقف کنیم.

در این گام به سه نود نیاز داریم:

۱. **Mission Node**: این نود می بایست پاسخگوی سرویس GetNextDestination باشد، به

گونه ای که برای هر درخواست، مقصد بعدی را از یک فایل text بخواند.

۲. **Sensor Node**: این نود باید فاصله ربات تا موانع را با استفاده از سنسور LiDAR و تاپیک

LaserScan بخواند و همواره مشخصات نزدیک ترین مانع را شامل فاصله بر روی تاپیک ClosestObstacle بفرستد. این تاپیک نیز به یک custom message با متغیر distance نیاز دارد.

۳. **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 ای 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. ربات باید همواره رو به جلو حرکت کند تا فاصله اش از مانعی کمتر از ۲ متر شود. سپس ایستاده و دوباره درخواستی به

GetNextDestination بفرستد و این فرایند را تا رسیدن به مقصد نهایی ادامه دهد. برای این کار نود کنترل می بایست از روی تاپیک ClosestObstacle تولید شده در نود سنسور subscribe نماید و از مقدار متغیر distance به عنوان فاصله تا نزدیک ترین مانع استفاده نماید.

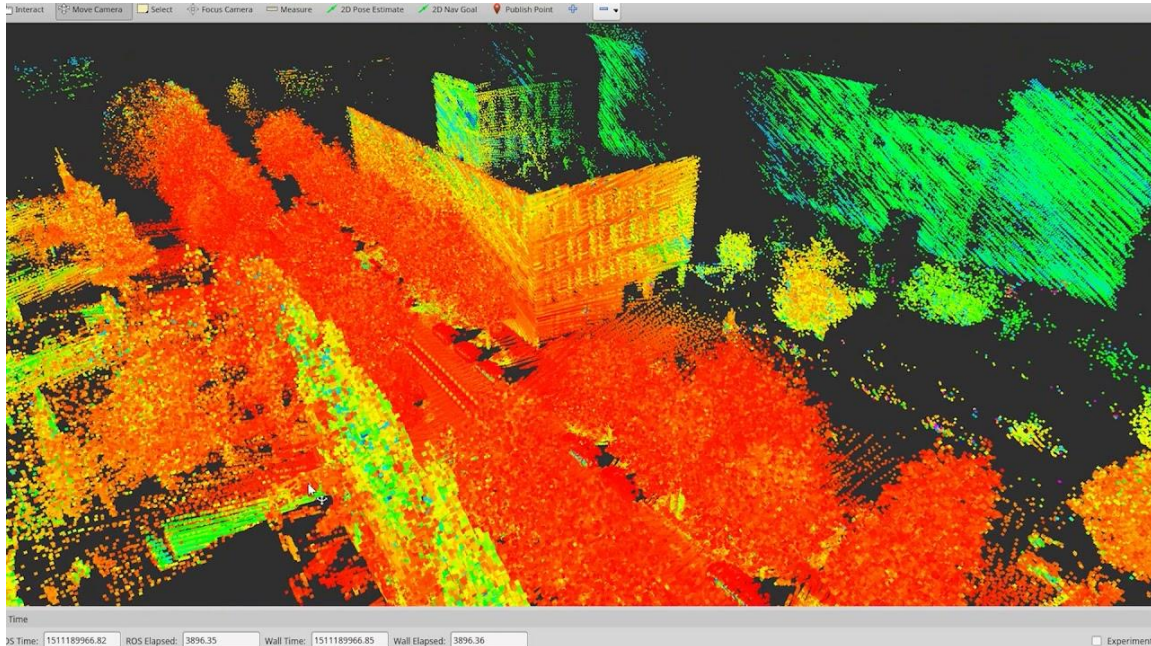
برنامه شما در این بخش باید قابلیت گرفتن سرعت خطی را از ورودی سمت کاربر داشته باشد. میتوانید از تگ param در launch file خود استفاده کنید. نقطه شروع ربات را بر روی مختصات (۰, ۰) قرار دهید. سپس به ازای سرعت خطی های ۱، ۲ و ۴ متر بر ثانیه اجرا بگیرید.

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 (۱۵ نمره):

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.

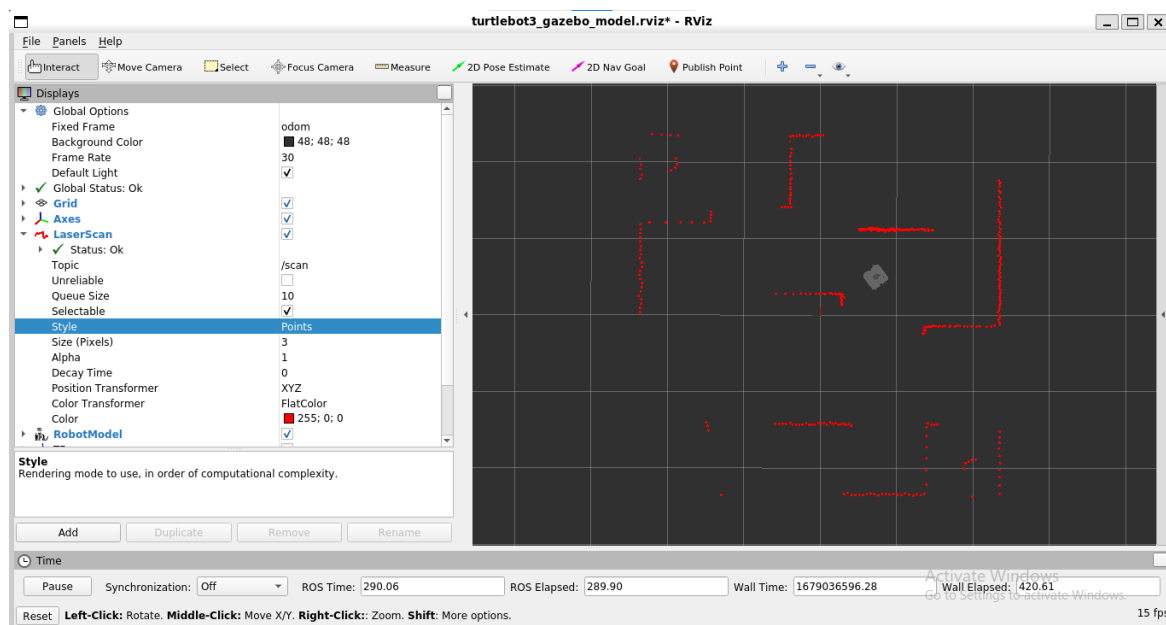
```
<include file="$(find turtlebot3_gazebo)/launch/turtlebot3_stage_4.launch"/>
```

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.

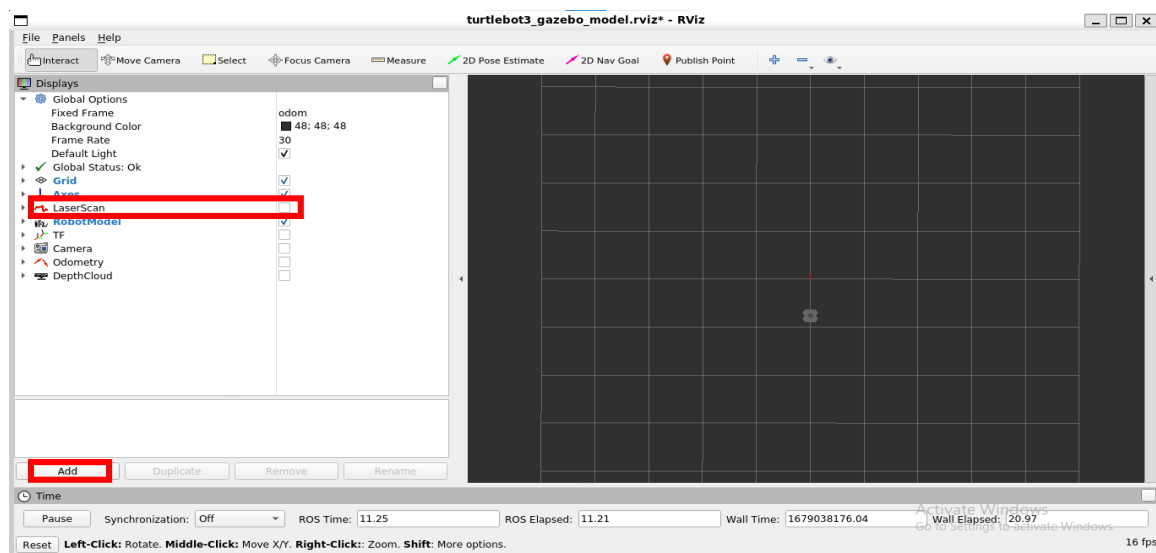
۲. fixed_frame : که فریم publish کردن پیام ها مشخص می کند. برای کاربرد ما این مقدار را "odom" ست کنید.

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

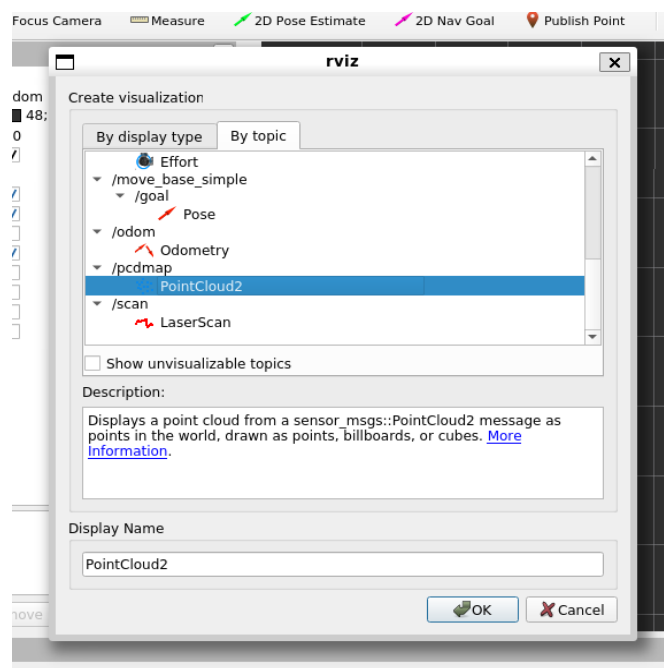
برای اضافه کردن سرویس به launch file و نوشتن نود خود می توانید از این لینک و ویدیو ی آن بهره ببرید:

<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.

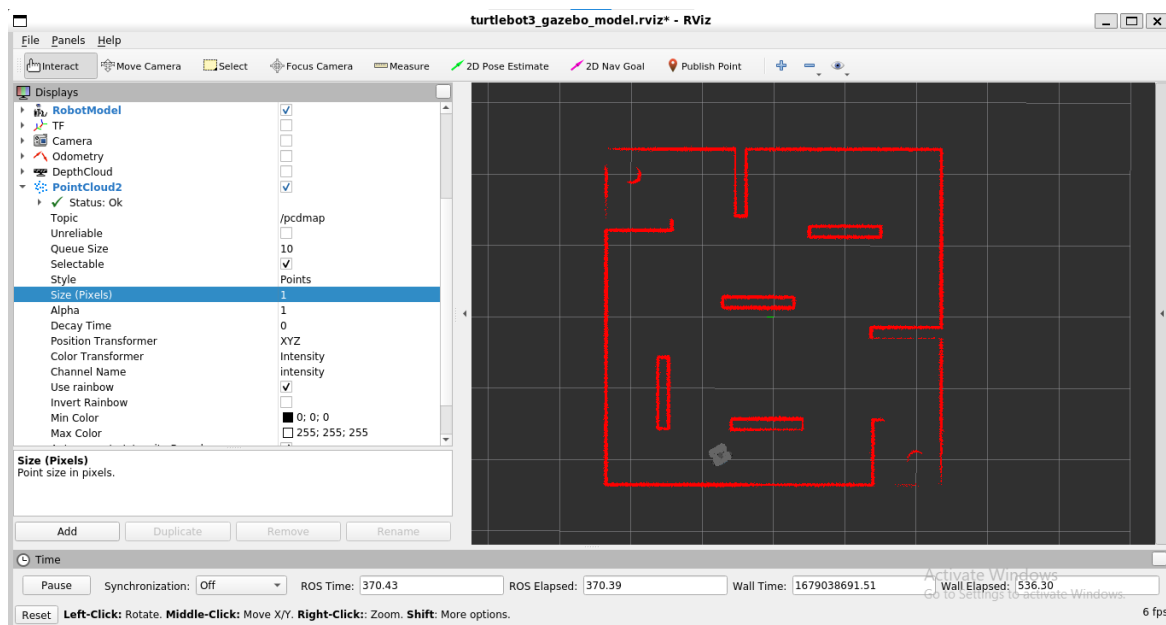


در منوی باز شده، در بخش By Topic بر روی PointCloud2 کلیک نمایید.



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 فولدری 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

___ your package/s

___ video Folder

___ first step 1 m/s

___ first step 2 m/s

___ first step 4 m/s

___ second step

نکات تکمیلی در باب تحویل تمرین

۱. مدت زمان هر فیلم باید ۱ دقیقه باشد. می‌توانید از دوربین موجود در gazebo استفاده کنید. در مجموع ۵ دقیقه فیلم برای این تمرین. حجم فایل تحویلی نهایتاً باید ۱۰۰ مگابایت باشد. برای کاهش حجم ویدیوهای 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 ۱۴۰۳/۰۱/۳۱، ۱۱:۵۹ شب، می‌باشد و سیاست‌های تاخیر مطابق با موارد ذکر شده در شیوه‌نامه لحاظ خواهد شد.