 انجام این تمرین به‌صورت انفرادی می‌باشد

بسمه‌تعالی

دانشگاه صنعتی امیرکبیر دپارتمان مهندسی کامپیوتر

درس اصول علم ربات نیمسال دوم سال تحصیلی1402-1401

تمرین سری دوم

**هدف از انجام این تمرین**

هدف این تمرین مرور برخی از مفاهیم مطرح شده در کلاس درس و همچنین یادگیری کار با ربات در محیط‌های 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 3،4 و 5 را مشاهده نمایید.

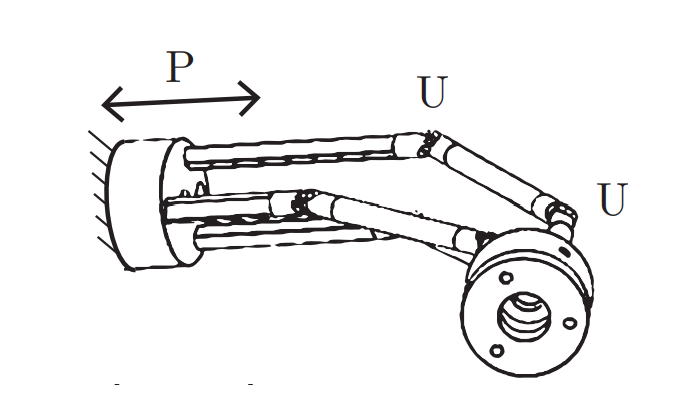
**بخش های تئوری:**

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) (12 نمره)

![A picture containing shape

Description automatically generated](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAGAAAABYCAYAAAAKsfL4AAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAAHYcAAB2HAY/l8WUAAADZSURBVHhe7dExAQAgDMCwgX/PwIGHPM1TAV3nmTD7N0gDsAZgDcAagDUAawDWAKwBWAOwBmANwBqANQBrANYArAFYA7AGYA3AGoA1AGsA1gCsAVgDsAZgDcAagDUAawDWAKwBWAOwBmANwBqANQBrANYArAFYA7AGYA3AGoA1AGsA1gCsAVgDsAZgDcAagDUAawDWAKwBWAOwBmANwBqANQBrANYArAFYA7AGYA3AGoA1AGsA1gCsAVgDsAZgDcAagDUAawDWAKwBWAOwBmANwBqANQBrADVzAWjVBKwjOw+3AAAAAElFTkSuQmCC)

Prismatic joint

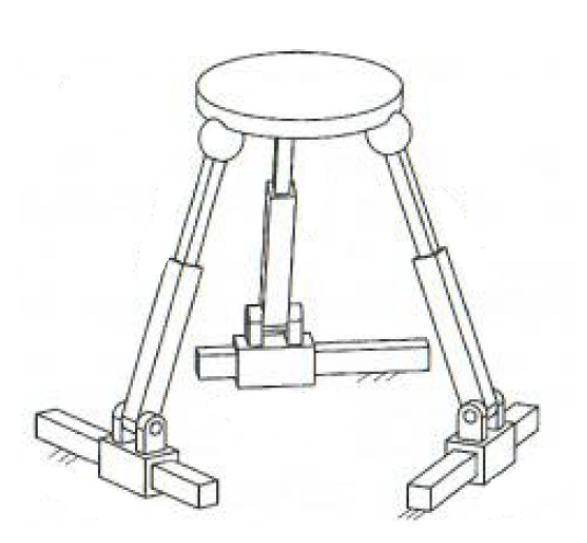


(a)

![A picture containing shape

Description automatically generated](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAGAAAABYCAYAAAAKsfL4AAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAAHYcAAB2HAY/l8WUAAADZSURBVHhe7dExAQAgDMCwgX/PwIGHPM1TAV3nmTD7N0gDsAZgDcAagDUAawDWAKwBWAOwBmANwBqANQBrANYArAFYA7AGYA3AGoA1AGsA1gCsAVgDsAZgDcAagDUAawDWAKwBWAOwBmANwBqANQBrANYArAFYA7AGYA3AGoA1AGsA1gCsAVgDsAZgDcAagDUAawDWAKwBWAOwBmANwBqANQBrANYArAFYA7AGYA3AGoA1AGsA1gCsAVgDsAZgDcAagDUAawDWAKwBWAOwBmANwBqANQBrADVzAWjVBKwjOw+3AAAAAElFTkSuQmCC)

Spherical joint



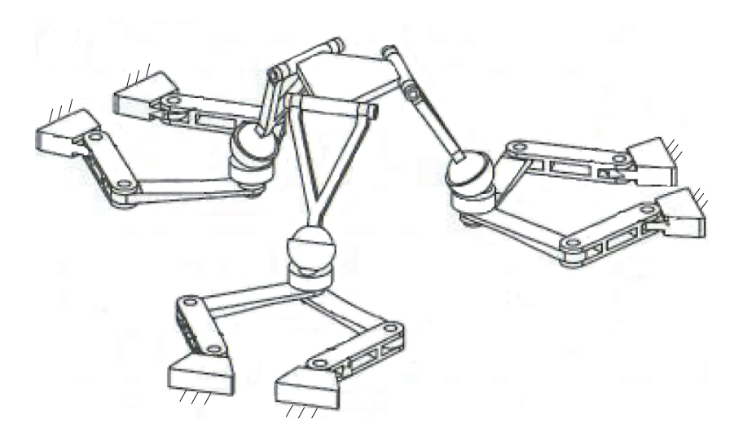
(b)

Combined Prismatic and Revolute joint

Dof = 2

Spherical joint

Prismatic joint



(c)

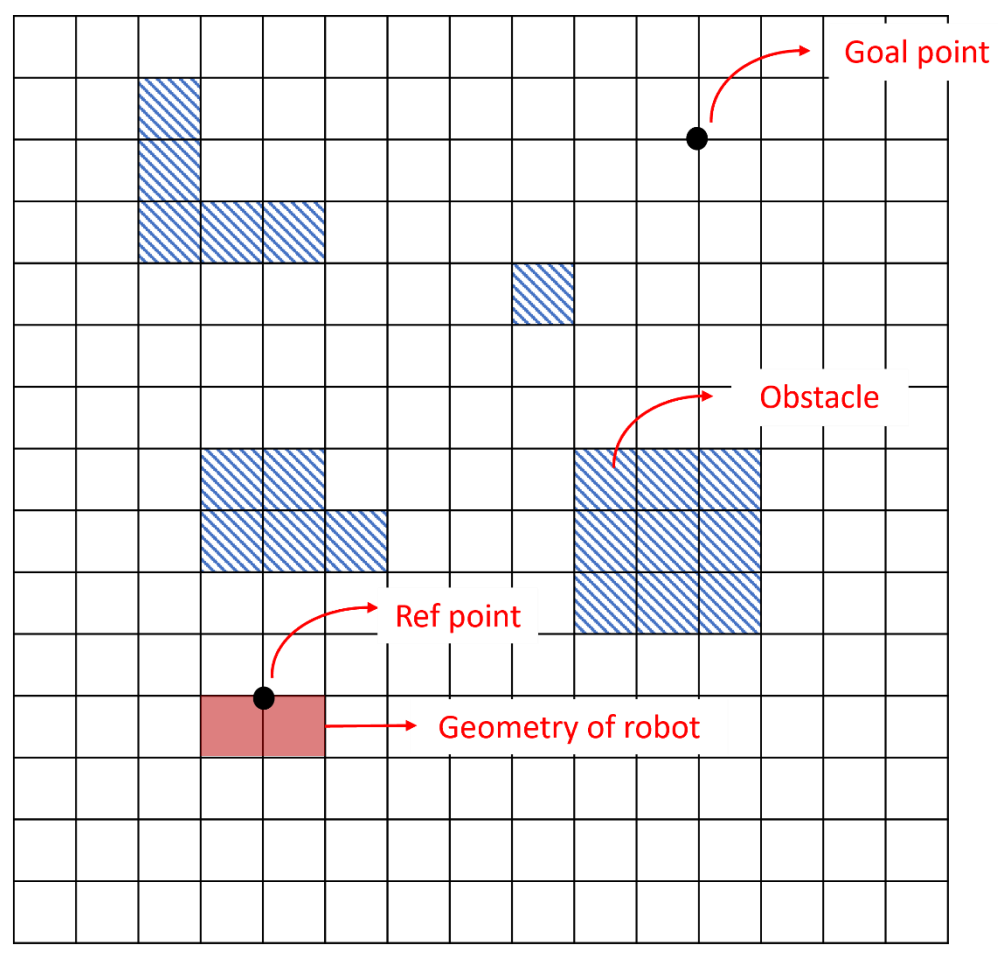
Spherical joint

Revolute joint

Revolute joint

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

1. برای ربات مشخص شده در تصویر زیر، C-space را مشخص کنید. ( فرض کنید که ربات تنها امکان دوران 90 درجه را دارد) (8 نمره)



Chart

Description automatically generated

1. تفاوت ربات های holonomic و non-holonomic را شرح دهید. (5 نمره)

**بخش های شبیه سازی:**

* **گام اول** (25 نمره) **:**

**این گام در محیط empty map انجام می‌شود.**

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:

1. 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 10- تا 10 به عنوان x و y مقصد بعدی بسازد. توجه کنید این نقطه باید با محل کنونی ربات دست کم 5 متر فاصله داشته باشد.

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

1. حرکت به سمت جلو با سرعت خطی ثابت

2. دوران در حالت ایستاده با سرعت زاویهای دلخواه

در آغاز کار، ربات باید درخواستی به سرویس GetNextDestination بفرستد. پس از دریافت مقصد بعدی به اندازه‌ی مناسب دوران کرده تا هدف دقیقا 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 5 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 (0, 0) قرار دهید. سپس به ازای سرعت خطی های 1، 2 و 4 متر بر ثانیه اجرا بگیرید.

موارد زیر را در گزارش خود قرار دهید:

1. ویدیوی کارکرد ربات برای هر سرعت خطی.
2. نمای مسیر ربات در شبیه ساز Rviz به ازای هر سرعت خطی.
3. 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** (35 نمره) **:**

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

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

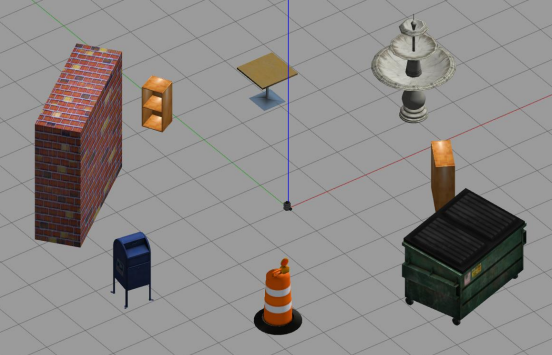
1**.** Sensor Node : این نود باید به فاصله ی ربات را تا موانع با استفاده از سنسور LiDAR و تاپیک LaserScan بخواند و همواره مشخصات نزدیک ترین مانع را شامل فاصله و زاویه بر روی تاپیک ClosestObstacle بفرستد. این تاپیک نیز به یک custom message با دو متغیر مانند شکل زیر نیاز دارد .

|  |
| --- |
| float64 distnace float64 direction |

با استفاده از این سنسور میخواهیم زاویه چرخش مناسب ربات را برای فاصله گرفتن از نزدیکترین مانع محاسبه کنیم. به عناون مثال اگر کمترین مقدار فاصله متعلق به ایندکس 30ام آرایه ی ranges خوانده شده از LiDAR باشد، ربات باید 150 درجه به سمت راست خود بچرخد.

2. 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 2 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 نقطه ی شروع ربات را (0, 0) قرار داده و yaw angle آن را 0.75 بگذارید.



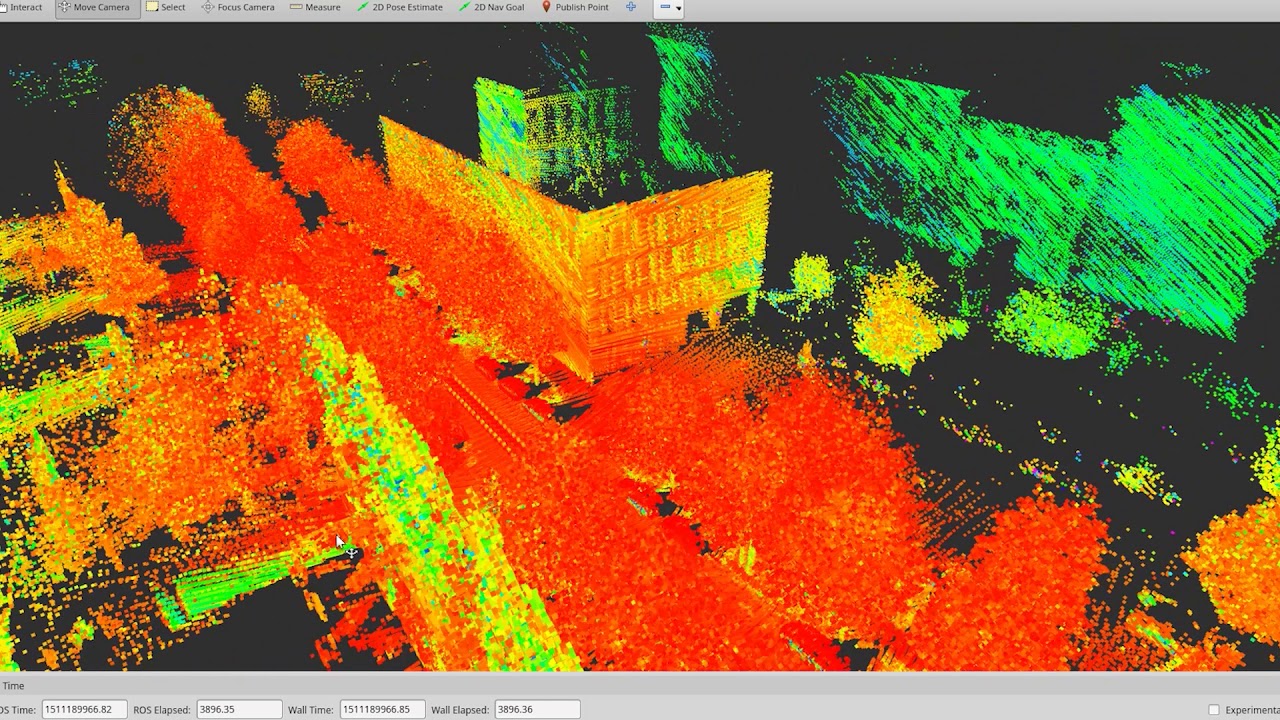
Second step word map – obstacle-detect.world

موارد زیر را در گزارش خود قرار دهید:

1. ویدیوی کارکرد ربات

2. نمای مسیر ربات در شبیه ساز Rviz

* **گام سوم** (15 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 پکج خود را اجرا کنید. با این دستور Gazebo و RViz اجرا می شوند. در یک ترمینال دیگر turtlebot3\_teleop\_key.launch را از پکج 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).

A picture containing diagram

Description automatically generated

موارد زیر را در گزارش خود قرار دهید:

1. تصویری از کارکرد ربات

2. 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.

1. برای این کار به سرویس laser\_scan\_assembler از پکج 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 پکج sensor\_msgs باز میگرداند. دستور اجرای نود این سرویس را به انتهای launch file خود اضافه کنید.

دقت کنید که این نود دو پارامتر دریافت می کند:

1. max\_scans : که تعداد بیشینه اسکن ها را در یک پیام PointCloud2 مشخص می کند. مقدار آنرا "400" بگذارید.

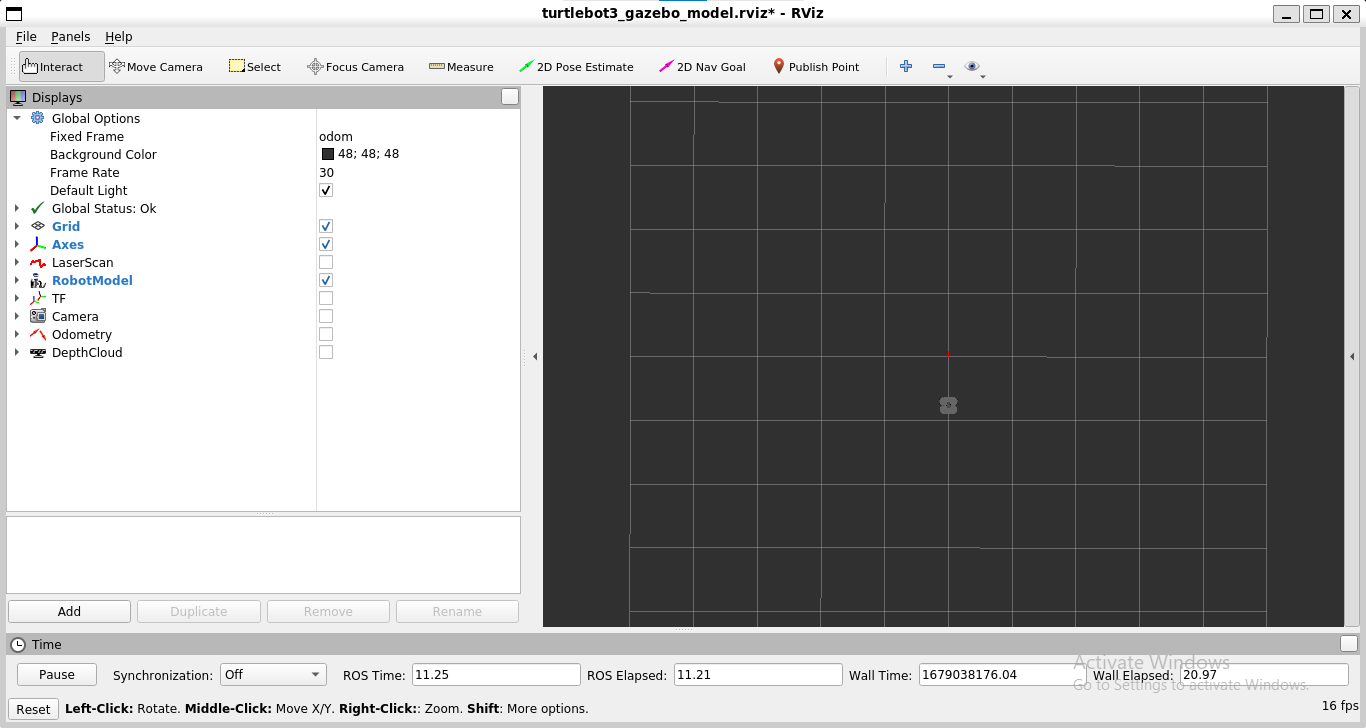
2. fixed\_frame : که فریم publish کردن پیام ها مشخص می کند. برای کاربرد ما این مقدار را "odom" ست کنید.

2. 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 پکج laser\_assembler استفاده نمایید.

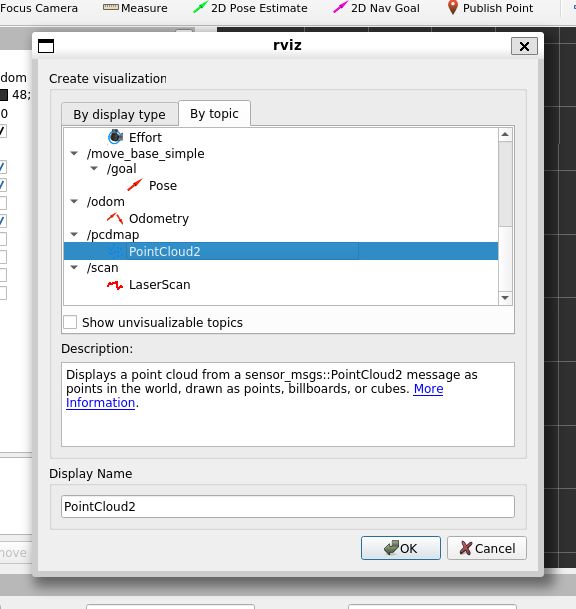
برای اضافه کردن سرویس به launch file و نوشتن نود خود می توانید از این لینک و ویدیو ی آن بهره ببرید:

<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.

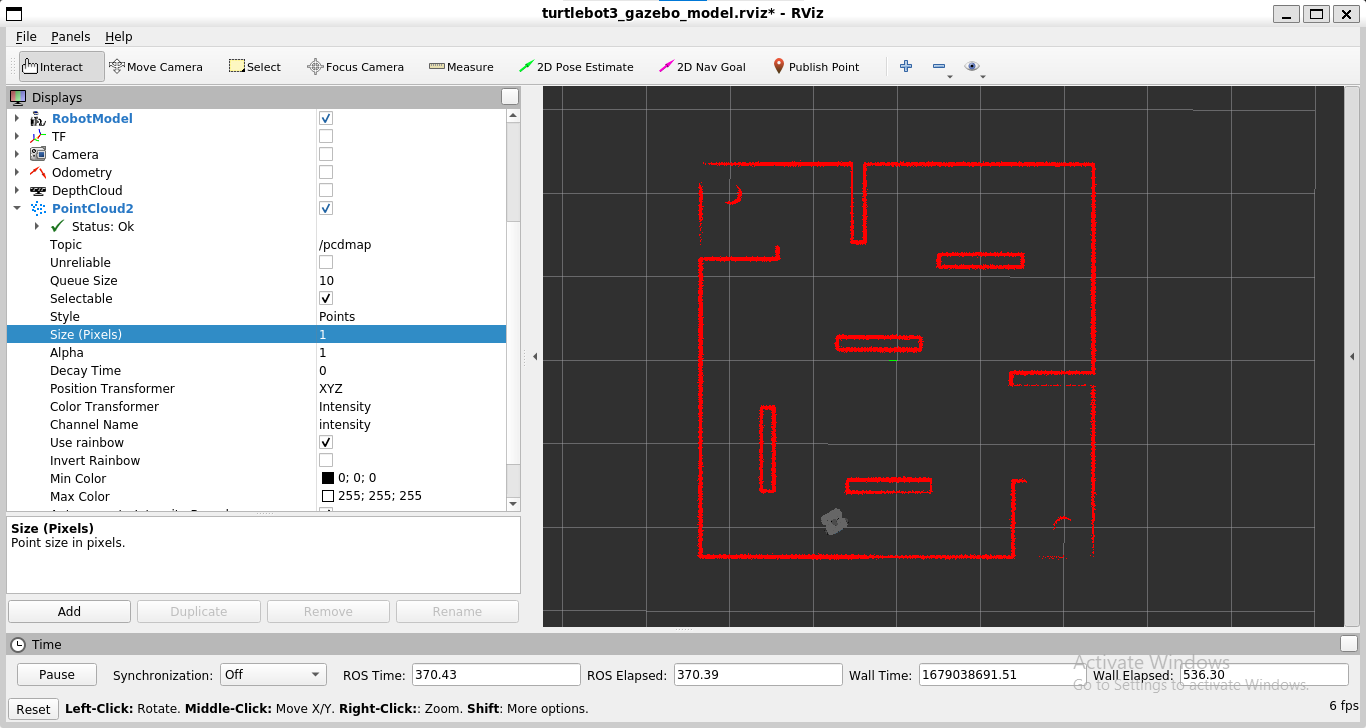


در منوی باز شده، در بخش By Topic بر روی PointCloud2 کلیک نمایید.



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 پکج 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:

1. ویدیوی کارکرد ربات
2. شکل نهایی

* **نحوه تحویل**

فایل zip نهایی شما باید شامل موارد زیر با نام گذاری دقیقا 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. مجموعا 5 فیلم برای این تمرین باید تهیه شود. 5 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

|\_\_ third step

**نکات تکمیلی در باب تحويل تمرين**

1. مدت زمان هر فیلم باید 1 دقیقه باشد. در موجوع 5 دقیقه فیلم برای این تمرین. حجم فایل تحویلی نهایتا باید 100 مگابایت باشد. برای کاهش حجم ویدیو های ضبط شده می‌توانید از نرم افزار handbrake بهره ببرید.
2. فرمت فایل زیپ نیز به صورت HW2\_StudentNumber باشد.
3. افراد می‌بایست تمارین را به صورت انفرادی انجام دهند.
4. 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.
5. 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.
6. تاریخ تحویل تمرین 19/12/1401، 11:59 شب، می‌باشد و سیاست‌های تاخیر مطابق با موراد ذکر شده در شیوه‌نامه لحاظ خواهد شد.