بسمه‌تعالی

دانشگاه صنعتی امیرکبیر دپارتمان مهندسی کامپیوتر

درس اصول علم ربات نیمسال دوم سال تحصیلی۱۴۰۳-۱۴۰۲

تمرین سری صفر

انجام این تمرین به‌صورت انفرادی می‌باشد

هدف از انجام این تمرین

هدف این تمرین آشنایی با مفاهیم پایه ROS است. برای انجام بخش پیاده سازی این تمرین، در ابتدا ویدیو ارائه شده برای نصب ROS و آموزش مفاهیم پایه‌ای آن را در مشاهده کنید. ROS یا سیستم عامل ربات، چارچوب نرم افزاری برای توسعه‌ی نرم افزار ربات است. ROS کتابخانه و ابزارهایی برای توسعه دهندگان نرم افزار، جهت ساخت کاربردهای نرم افزاری رباتیک فراهم می کند. این سیستم لایه انتزاعی سخت‌افزاری، راه انداز های دستگاه‌ها، کتابخانه‌ها، ابزارهای بصری، ارسال پیام ها، مدیریت بسته ها و... را در اختیار کاربران قرار می دهد. لطفا نسخه‌ی مطرح شده در هندزان ترم جاری (ros نسخه neotic) را مطابق با دستورالعمل ارائه شده در ویدیو نصب بفرمایید.

هدف از این تمرین تنها آشنایی با برنامه نویسی سمت سرور ROS می‌باشد. در این تمرین میخواهیم با استفاده از یک سناریو ساده با برنامه نویسی سمت سرور، ایجاد node و topic و همچنین publish/subscribe کردن پیام ها به وسیله topic آشنا شویم.

❖ شرح سناریو

موبایل رباتی مشابه شکل ۱ در اختیار داریم. این ربات قابلیت چرخش درجا (rotate) و حرکت در مسیر مستقیم را دارد و همچنین سنسوری در بالای ربات قرار گرفته است که فاصله بر حسب سانتیمتر را از ۴ جهت روبرو، پشت، راست و چپ ربات در اختیار ما قرار می‌دهد. هدف این است که با استفاده از دو قابلیت ذکر شده ربات (چرخش درجا و فاصله سنجی از طرفین) و به کمک ابزارهایی که ROS در اختیار ما قرار می‌دهد برنامه‌ایی طراحی کنیم که ربات ابتدا نزدیک ترین مانع به خود را با استفاده از داده سنسور ها تشخیص داده و سپس در جهتی چرخش کند تا پشتش به آن مانع باشد. سپس به سمت دورترین مانع حرکت کند. (تا زمانی که فاصله اش از آن مانع کم شود). به عبارتی اگر نقطه ۰ مبدا قرار داشته باشد و یک مانع در نقطه (۰،۱) باشد که نزدیکترین مانع به ربات است. باید در جهت محور منفی چرخیده تا زمانی که به سمت دورترین مانع قرار گیرد. اگر مانع دورتر در نقطه (۵،۰) باشد باید ربات ۹۰ درجه به سمت راست بچرخد. فرض کنید جهت ربات در ابتدای کار همیشه رو به نزدیک ترین مانع است.

شکل ۱) **turtlebot-burger**

همانطور که در معیارهای برنامه نویسی تمیز(clean-code) گفته شده است که هر تابع یا method باید دقیقا یک وظیفه مشخص انجام دهند در ROS هم هر گره یا node که ایجاد می‌کنیم باید دقیقا یک وظیفه مشخص داشته باشد.

از این رو برای طراحی سیستمی با مشخصات گفته شده کل سیستم را به ۳ بخش اصلی تقسیم کرده و برای هرکدام از این بخش ها یک node ایجاد می‌کنیم. این سه بخش اصلی عبارت هستند از ۱) سنسور تشخیص فاصله ۲) کنترلر ۳) موتور ها.

وظیفه هر کدام از ۳ node نام برده شـــده در زیر قرار داده شـــده اند همچنین می‌توانید گراف مربوط را در شـــکل ۲ مشاهده نمایید.

۱. سنسور تشخیص فاصله: در این node در هر iteration چهار عدد فاصله را از فایل داده شده میخواند. ترتیب اعداد به ترتیب برابرند با چپ، بالا، راست، پایین. این اعداد فاصله ربات از موانع موجود در چهار طرف ربات می‌باشند. این node باید مقادیر فاصـله را در قالب یک custum-message درون topic مخصـوص خود با نام distance قرار دهد. دقت شود که رنج دید سنسور فاصله ۱۰ سانتیمتر تا ۲۰۰ سانتی متر می‌باشد.
۲. کنترلر: وظیفه این گره تشـــخیص بهترین جهت چرخش و کنترل چرخ های ربات با اســـتفاده از مقادیر فاصـــله می‌باشـد. برای این کار این گره ابتدا باید تاپیک distance را subscribe کند تا به مقادیر فاصـله دسـترسـی داشـته باشـد. پس از پیدا کردن نزدیک ترین مانع ابتدا باید جهت چرخش مناسـب را پیدا کنید تا پشـت به مانع باشـید. سپس با استفاده از همان داده ها دورترین مانع را پیدا کرده و به سمت آن حرکت کنید.
۳. در نهایت کنترلر باید پیام مناسـب که شـامل مقدار چرخش بر حسـب درجه و جهت چرخش و سـرعت حرکت و جهت حرکت هست را در قالب custome-message در topic های مناسب مربوط به هر موتور قرار دهد.

نکته اول: ربات دارای ۲ عدد موتور dc متصل چرخ، مانند شکل ۱ می‌باشد.

نکته دوم: با توجه به شـکل ۱ اگر ربات بخواهد در جای خود حرکت چرخشـی انجام دهد بدیهتا چرخ ها در جهت مخالف همدیگر باید بچرخند. بنابراین مقادیر موجود در topic موتور ۱ و ۲ با هم متفاوت هستند.

نکته سوم: این node هم subscriber است و هم publisher. شکل ۲

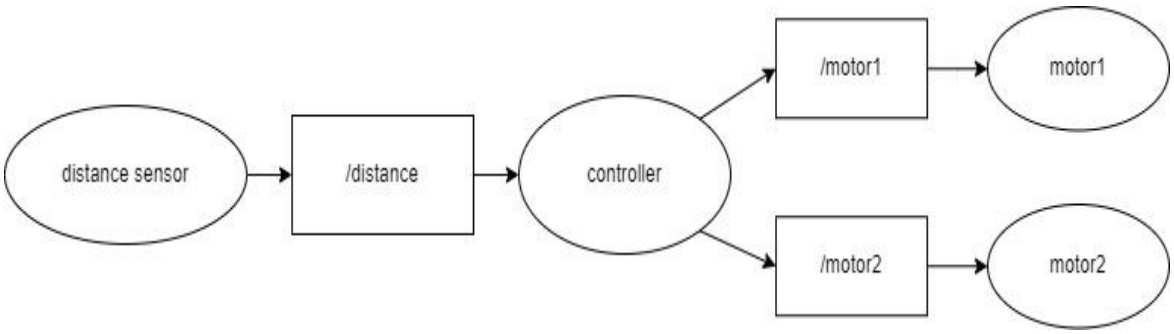
نکته چهارم: از آنجایی که داده های اولیه رندوم اسـت شـما تمامی پردازش ها را برای اولین داده های رندوم انجام داده و شـروع به حرکت میکنید تا زمانی که از نزدیک ترین مانع ۱۰ سـانتی متر فاصـله بگیرید. سـپس این کار را مجدد برای نزدیک ترین مانع جدید انجام میدهید.

۴. موتور: این node تنها از topic مربوط به خود subscribe می‌کند و با اســـتفاده از loginfo مقادیر دریافتی را درون ترمینال چاپ می‌کند.

در آخر برای اطمینان از درستی کارکرد برنامه و نمایش گراف نهایی می‌توانید دستور زیر را اجرا کنید:

```
rqt_graph
```

در انتها خروجی دستور بالا باید مشابه با تصویر زیر باشد.



شکل ۱) گراف node و topic های سناریو اول

نکته آخر: برای راحتی کار فرض می‌کنیم که چرخش ربات هیچ زمانی نمی‌گیرد. به محض اینکه node کنترلر تصمیم می‌گیرد که ربات به چه اندازه و به چه جهتی بچرخد، چرخش صورت می‌گیرد. همچنین جهت قرارگیری یا state ربات باید همواره در node کنترلر update و ذخیره شود. جهت قرار گیری اولیه ربات دلبخواهی است.

❖ نحوه تحویل

کل فولدر پکیج مربوط به این تمرین را به همراه فایل pdf گزارش کار zip کرده و درون سامانه بارگزاری نمایید.

در فایل گزارش کار باید صحت انجام درست و کامل سیستم به خوبی با عکس و اسکرین شات از ترمینال هایی که در آن node های مختلف را run کردید مشخص شده باشد. همچنین عکس گراف نهایی سیستم(خروجی rqt_graph) را در فایل گزارش کار قرار دهید

نکات تکمیلی در باب تحویل تمرین

۱. تحویل گزارش در یک فایل pdf مطابق با قالب قرار گرفته در سایت کورسز و با نام‌گذاری HW0_StudentNumber می‌بایست تحویل داده شود.
۲. فرمت فایل زیپ نیز به صورت HW0_StudentNumber باشد.
۳. افراد می‌بایست تمارین 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. برای انجام تمرین و پروژه تنها زبان برنامه‌نویسی مجاز پایتون می‌باشد.

۶. 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.

۷. تاریخ تحویل تمرین ۱۴۰۲/۱۲/۲۹، ۱۱:۵۹ شب، می‌باشد و سیاست‌های تاخیر مطابق با موراد ذکر شده در شیوه‌نامه لحاظ خواهد شد.