



Department of  
Computer Engineering

به نام خدا



Amirkabir University of Technology  
(Tehran Polytechnic)

دانشگاه صنعتی امیرکبیر  
دانشکده مهندسی کامپیوتر  
اصول علم ربات

تمرین سری سوم

نام و نام خانوادگی	محسن حسینی
شماره دانشجویی	۹۷۳۱۰۱۶
تاریخ ارسال گزارش	۱۴۰۱/۰۳/۱۳

**فهرست گزارش سوالات** (لطفاً پس از تکمیل گزارش، این فهرست را به‌روز کنید).

- سوال ۱ - سنسورها ..... ۳
- سوال ۲ - مکان یابی ..... ۳
- سوال ۳ - ناوبری کور ..... ۳
- سوال ۴ - قطب نما ..... ۳
- سناریو ۱ ..... ۴
- سناریو ۲ ..... ۵

## سوال ۱ – سنسورها

دو سنسور Active:

- GPS
- LiDAR

دو سنسور Passive:

- Inclinator
- Camera

همانطور که در بالا اشاره شده است، GPS از نوع فعال است زیرا سیگنال‌هایی را که از ماهواره‌های مخصوص ارسال می‌شود را دریافت و تحلیل می‌کند. پس توسط سیگنال‌های ماهواره‌ای انرژی به محیط می‌دهد.

## سوال ۲ – مکان یابی

برای مکان یابی داخل ساختمان GPS گزینه مناسبی نیست. زیرا امواج ماهواره‌ای باید به صورت مستقیم دریافت شوند و این امکان در فضای بسته وجود ندارد و یا با خطای بالایی همراه است.

## سوال ۳ – ناوبری کور

از ترکیب قطب نما و شیب سنج استفاده می‌شود.

از قطب نما برای پیدا کردن هدینگ روبات حول محور Z استفاده می‌شود و از شیب سنج برای پیدا کردن هدینگ روبات حول محور X استفاده می‌شود.

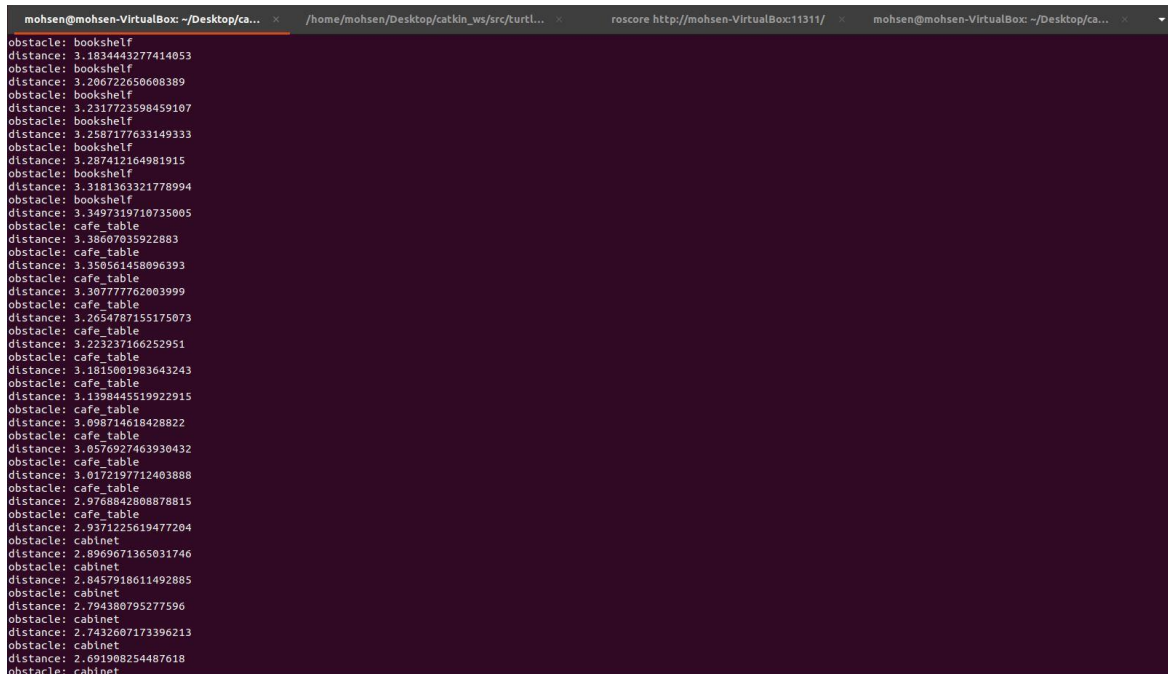
## سوال ۴ – قطب نما

از قطب نما برای پیدا کردن هدینگ روبات حول محور Z استفاده می‌شود. بدین صورت که با خواندن شمال فعلی از قطب نما و دانستن هدینگ ابتدایی و جهت شمال در زمان شروع، می‌توان هدینگ حول محور Z را بدست آورد.

## سناریو ۱

الف) در این قسمت ابتدا یک فایل launch مناسب ایجاد می‌کنیم. در ادامه باید یک فایل پیام مخصوص ایجاد کنیم. در انتهای باید کد مربوط به نود خود را بنویسیم که تمام فایل‌ها ضمیمه شده است.

اسکرین شات از اجرا در سیستم من:



```
mohsen@mohsen-VirtualBox: ~/Desktop/ca... x /home/mohsen/Desktop/catkin_ws/src/turtl... x roscore http://mohsen-VirtualBox:11311/ x mohsen@mohsen-VirtualBox: ~/Desktop/ca... x
obstacle: bookshelf
distance: 3.1834443277414053
obstacle: bookshelf
distance: 3.286722650608389
obstacle: bookshelf
distance: 3.2317723598459107
obstacle: bookshelf
distance: 3.2587177633149333
obstacle: bookshelf
distance: 3.287412164981915
obstacle: bookshelf
distance: 3.3181363321778994
obstacle: bookshelf
distance: 3.3497319710735005
obstacle: cafe_table
distance: 3.38607035922883
obstacle: cafe_table
distance: 3.350561458096393
obstacle: cafe_table
distance: 3.307777762003999
obstacle: cafe_table
distance: 3.2654787155175073
obstacle: cafe_table
distance: 3.223237166252951
obstacle: cafe_table
distance: 3.1915801983643243
obstacle: cafe_table
distance: 3.1398445519922915
obstacle: cafe_table
distance: 3.098714618428822
obstacle: cafe_table
distance: 3.0576927463930432
obstacle: cafe_table
distance: 3.0172197712403888
obstacle: cafe_table
distance: 2.9768842808878815
obstacle: cafe_table
distance: 2.9371225619477204
obstacle: cabinet
distance: 2.8969671365031746
obstacle: cabinet
distance: 2.8457918611492885
obstacle: cabinet
distance: 2.794380795277596
obstacle: cabinet
distance: 2.7432607173396213
obstacle: cabinet
distance: 2.691980254407618
obstacle: cabinet
```

ب) در این قسمت علاوه بر فایل‌های قسمت قبلی نیاز به یک نود جدید و فایل سرویس داریم. هردو در پوشه Codes قرار داده شده است.

اسکرین شات از اجرای در سیستم من (خروجی طبیعتاً مانند قسمت قبل خواهد بود):

```
mohsen@mohsen-VirtualBox: ~/Desktop/catkin_ws
distance: 3.185448921261051
obstacle: bookshelf
distance: 3.21155661084121
obstacle: bookshelf
distance: 3.231060352336862
obstacle: bookshelf
distance: 3.256560517141325
obstacle: bookshelf
distance: 3.2833666698290025
obstacle: bookshelf
distance: 3.310939722140434
obstacle: bookshelf
distance: 3.338808083961823
obstacle: bookshelf
distance: 3.3671279075270943
obstacle: cafe_table
distance: 3.375825412630877
obstacle: cafe_table
distance: 3.380103864418644
obstacle: cafe_table
distance: 3.380240387445651
obstacle: cafe_table
distance: 3.262897984980832
obstacle: cafe_table
distance: 3.225541071421314
obstacle: cafe_table
distance: 3.188636353104241
obstacle: cafe_table
distance: 3.1517815790744814
obstacle: cafe_table
distance: 3.115287071083133
obstacle: cafe_table
distance: 3.078987269612714
obstacle: cafe_table
distance: 3.0429223578119213
obstacle: cafe_table
distance: 3.00720881543502
obstacle: cafe_table
distance: 2.971614484725839
obstacle: cafe_table
distance: 2.9365209355010573
obstacle: cabinet
distance: 2.9011692243703324
obstacle: cabinet
distance: 2.855905660885707
obstacle: cabinet
```

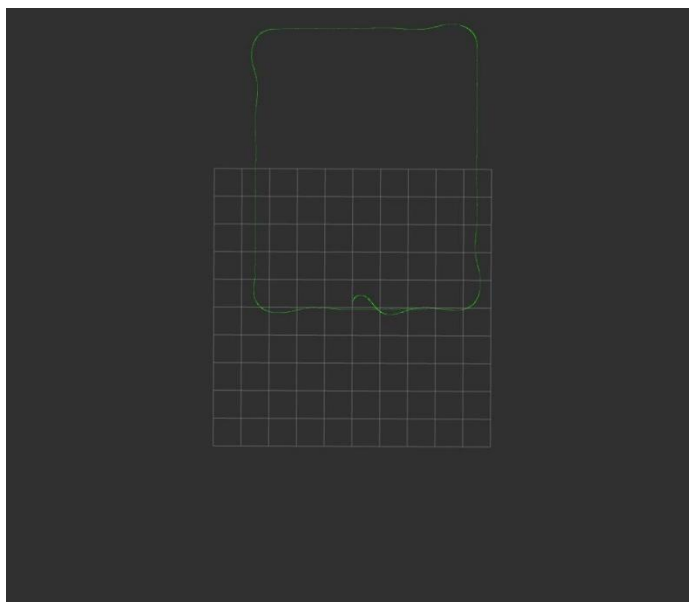
ج) در این قسمت ابتدا نود `closest_obstacle` نزدیک ترین مانع را به کمک نود `distance_calculator` پیدا می کند (ارتباط این دو از طریق سرویس `GetDistance` انجام می شود). در ادامه نود `closest_obstacle` نزدیک ترین مانع و فاصله تا آن را (از طریق پیام `ClosestObstacle`) منتشر می کند. نود `lidar` وظیفه خواندن این پیام ها و پیام های `LaserScan` را دارد. با توجه به اعداد دریافتی از این دو تاپیک کنترل ربات را برعهده می گیرد.

نکته ای که باید اشاره شود این است که به دلیل سخت بودن استفاده از `teleop` در این بخش (با جناب ستاک در این باره ایمیل رد و بدل شده است)، از آن استفاده نکردم و به صورت دستی سرعت خطی ربات را کنترل کردم. ولی همچنان چرخش ها دقیق و منظم انجام می شوند.

## سناریو ۲

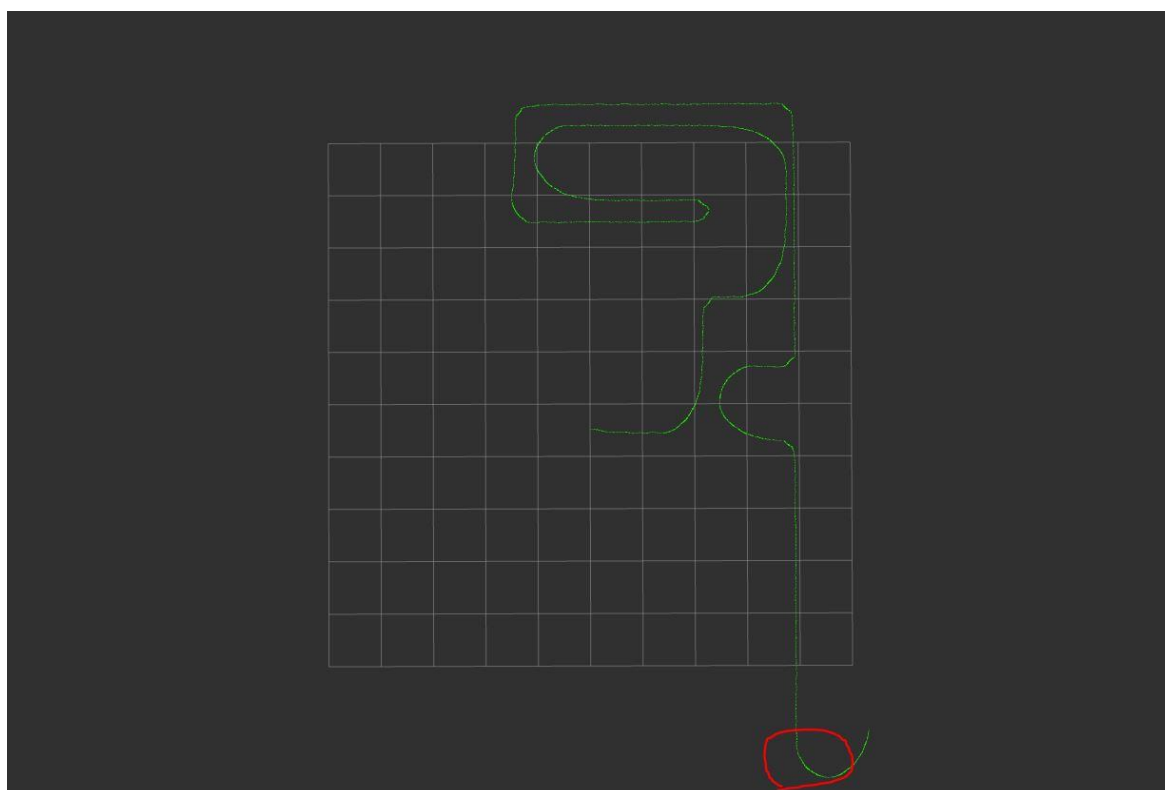
الف) ابتدا کدهای تدریس یار را در پکیج جدید قرار می دهیم و سپس فایل `lanj` را با توجه به پارامتر های جدید و دنیای جدید تغییر می دهیم. در ادامه نیاز است تا مانند تمرین قبل فایل `monitor.py` را به پکیج اضافه کنیم تا مسیر ربات را در شبیه ساز `Rviz` مشاهده کنیم. در آخر با ایجاد تغییرات خروجی را بهبود می بخشیم و کنترلر را `tune` می کنیم.

مسیر ربات در شبیه ساز `Rviz`:



ب) در این بخش برخلاف بخش قبلی از کنترلر PID استفاده نکرده ام (به دلیل پیچیدگی بالا). در نود `maze_solver` عملیات کنترل ربات رخ می‌دهد. ابتدا باید به سمت چپ برویم تا یک دیوار را پیدا کنیم. پس از پیدا کردن دیوار آن را دنبال می‌کنیم تا از `maze` خارج شویم. باقی اطلاعات به صورت کامنت در کد قرار داده شده است.

مسیر حرکت ربات در شبیه‌ساز Rviz (خروج از سمت پایین راست انجام شد که با قرمز مشخص شده است):



ج) در این قسمت از کد بخش قبلی برای پیمایش کنار دیوار استفاده می‌کنیم و تنها فرقی که وجود دارد این است که اگر بین ربات و هدف فاصله خالی مشخصی باشد، بجای دنبال کردن دیوار، هدف را دنبال می‌کند.

مسیر طی شده تا رسیدن به هدف در شبیه‌ساز Rviz:

