|  |  |  |
| --- | --- | --- |
| C:\Users\mohammad\Downloads\image(1).png | به نام خدا | C:\Users\mohammad\Pictures\amirkabirLogo.png |
| **دانشگاه صنعتی امیرکبیر**  **دانشکده‌ مهندسی کامپیوتر**  **اصول علم ربات**  **تمرین سری سوم** | | |

|  |  |
| --- | --- |
| محسن حسینی | نام و نام خانوادگی |
| 9731016 | شماره‌ دانشجویی |
| 13/03/1401 | تاریخ ارسال گزارش |

­

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

[سوال 1 – سنسورها 3](#_Toc103811499)

[سوال ۲ – مکان یابی 3](#_Toc103811500)

[سوال 3 – ناوبری کور 3](#_Toc103811501)

[سوال 4 – قطب نما 3](#_Toc103811502)

[سناریو 1 4](#_Toc103811503)

[سناریو 2 5](#_Toc103811504)

# سوال 1 – سنسور­ها

دو سنسور Active:

* GPS
* LiDAR

دو سنسور Passive:

* Inclinometer
* 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 3 – 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.

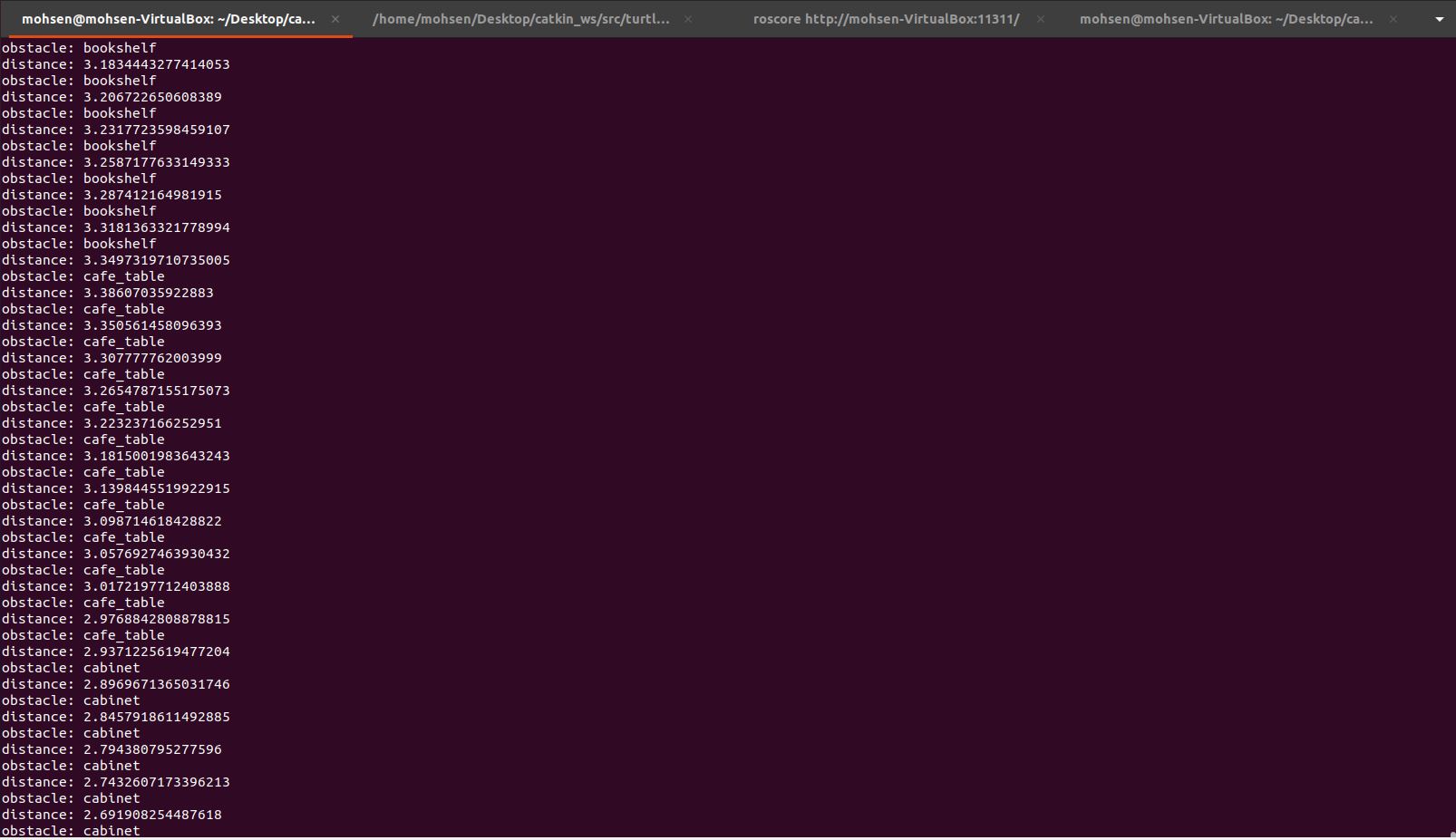
# سوال 4 – قطب­ نما

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 1

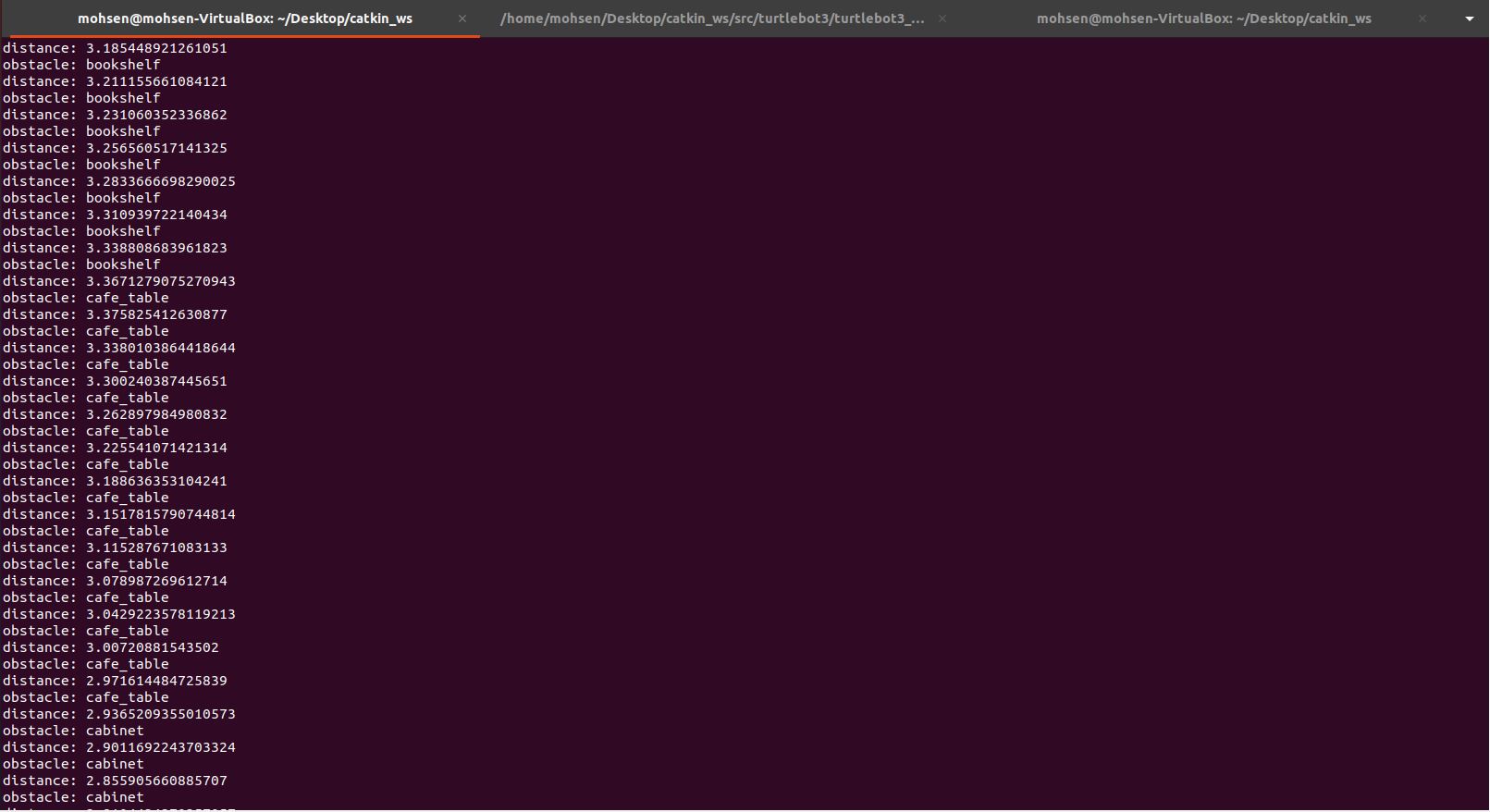
الف) در این قسمت ابتدا یک فایل launch مناسب ایجاد می­کنیم. در ادامه باید یک فایل پیام مخصوص ایجاد کنیم. در انتهای باید کد مربوط به نود خود را بنویسیم که تمام فایل­ها ضمیمه شده است.

اسکرین شات از اجرا در سیستم من:



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 طبیعتا مانند قسمت قبل خواهد بود):



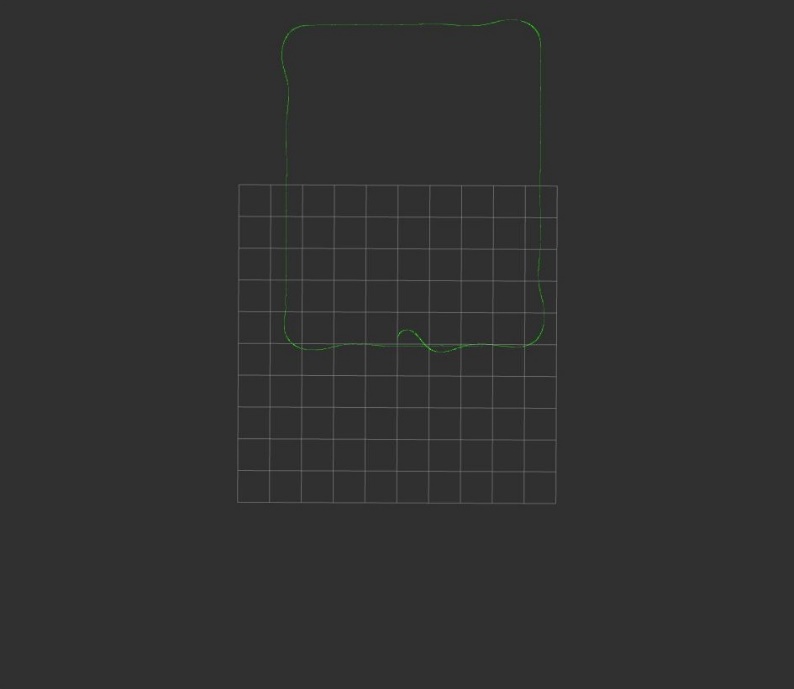
ج) در این قسمت ابتدا نود closest\_obstacle نزدیک­ترین مانع را به کمک نود distance\_calculator پیدا می­کند (ارتباط این دو از طریق سرویس GetDistance انجام می­شود). در ادامه نود closest\_obstacle نزدیک ترین مانع و فاصله تا آن را (از طریق پیام ClosestObstacle) منتشر می­کند. نود lidar وظیفه خواندن این پیام ها و پیام های LaserScan را دارد. با توجه به اعداد دریافتی از این دو تاپیک کنترل ربات را برعهده می­گیرد.

نکته­ای 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 می­شوند.

# سناریو 2

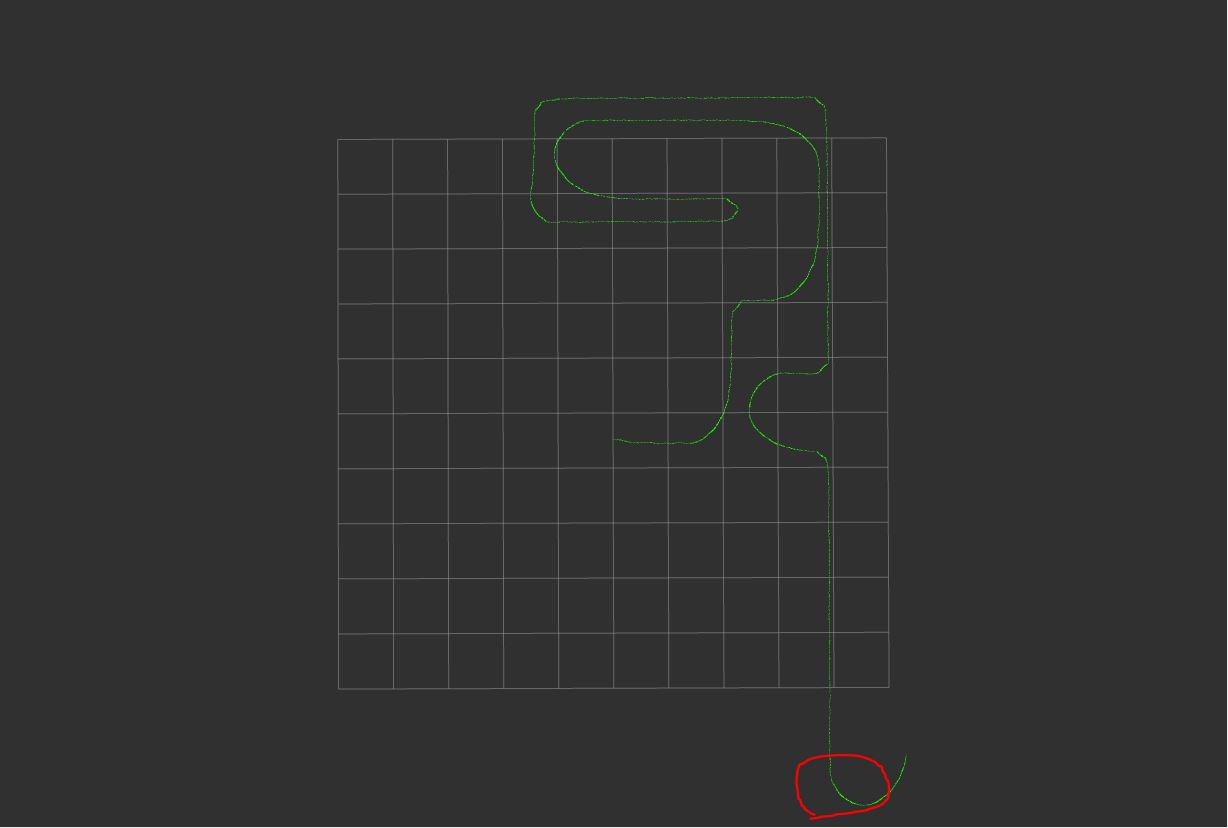
الف) ابتدا کدهای تدریس یار را در پکیج جدید قرار می­دهیم و سپس فایل لانچ را با توجه به پارامتر های جدید و دنیای جدید تغییر می­دهیم. در ادامه نیاز است تا مانند تمرین قبل فایل monitor.py را به پکیج اضافه کنیم تا مسیر ربات را در شبیه­ساز Rvis مشاهده کنیم. در آخر با ایجاد تغییرات خروجی را بهبود می­بخشیم و کنترلر را tune می­کنیم.

مسیر ربات در شبیه­ساز Rvis:



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 Rvis (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 Rvis:

