

Project Phase#0

ناوبری خودمختار (autonomous navigation) یک زمینه تحقیقاتی است که در چند سال گذشته شاهد پیشرفت های چشم گیری بوده است. یک مثال از کاربرد آن در صنعت و کارخانه ها است که در آن ها بخش قابل توجه ای از هزینه ها مربوط به جابجایی اشیا توسط نیروی کار انسانی است. هزینه جابجایی مواد کارخانه را می توان با استفاده از وسایل نقلیه رباتیک خودگردان کاهش داد. همچنین، این ربات های خودمختار می توانند در مکان هایی که برای کارگران خطرناک است، با حداقل مداخله انسانی حرکت کنند. مسائل مختلفی در ناوبری خودمختار مانند نقشه برداری، مکان یابی و برنامه ریزی مسیر وجود دارند. هدف این پروژه بحث در رابطه با و نشان دادن مفهوم طراحی و شبیه سازی یک ربات متحرک است که نهایتاً باید بتواند با اجتناب از موانع با استفاده از SLAM به مکان مورد نظر به صورت مستقل برسد.

زمینه پژوهشی (SLAM) Simultaneous Localization and Mapping سعی دارد این مسائل را با استفاده از انواع حسگرها از جمله: اسکنرهای لیزری، رادارها، دوربین ها، encoderها، GPS و IMU ها حل کند. متداول ترین حسگر ادراکی مورد استفاده برای مکان یابی و نقشه برداری در محیط های صنعتی، اسکنر لیزری است. روش های SLAM با استفاده از اسکنرهای لیزری عموماً قوی ترین روش ها در زمینه SLAM در نظر گرفته می شوند و می توانند موقعیت یابی دقیقی را در حضور موانع دینامیکی و محیط های متغیر ارائه دهند.

مسئله ناوبری ربات را می توان به 3 زیر کار یا فاز مختلف تقسیم کرد:

- نقشه برداری Mapping
- مکان یابی Localization
- برنامه ریزی Planning

نقشه برداری و مکان یابی با هم مرتبط و از وظایف اصلی SLAM هستند. با اینکه اکتشاف مستقل محیط توسط ربات از اهداف بلندپروازانه این حوزه است، SLAM پیچیدگی برنامه ریزی در حین ساخت نقشه را کنار می گذارد و فرض می کند که ربات توسط teleoperation یا شیوه دیگری کنترل می شود.

Mapping:

ایجاد نقشه یک فرآیند بسیار مهم در جهت یابی ربات است، زیرا معمولاً در مرحله مکان یابی نیز استفاده می شود. نقشه یک بازسازی فضایی از محیط اطراف ربات است. این بازسازی که به آن مدل نیز گفته می شود، با استفاده از حسگرها و دوربین های ربات همراه با odometry آن ایجاد شده است. این حسگرها نقش مهمی در به دست آوردن اطلاعات محیطی دقیق تر برای پردازش و نقشه برداری دارند. سنسورهای متنوعی مانند سونار، مادون قرمز (IR)، اسکنر لیزری و LiDAR را می توان برای نقشه برداری از محیط استفاده کرد.



از انجایی که در این درس فقط دارای سنسور RGBD هستیم، از شیوه Visual SLAM استفاده خواهیم کرد. نقشه‌های ایجاد شده انواع مختلفی دارند از جمله:

- Occupancy Grid Maps
- Feature-Based
- Maps and Topological maps

که هرکدام بسته به نوع سنسور و محیطی که قرار است نقشه برداری شود، استفاده‌های خود را دارد.

:Localization

مکان‌یابی ربات فرآیند تعیین محل قرارگیری یک ربات متحرک با توجه به محیط آن است، مکان‌یابی به این سوال پاسخ می‌دهد که «ربات الان کجاست؟». اگر یک مدل ریاضی از حرکت داشته باشیم، یک ربات متحرک مجهز به حسگر برای نظارت حرکت خود (مانند رمزگذارهای چرخ و حسگرهای اینرسی) می‌تواند تخمینی از مکان خود نسبت به جایی که شروع کرده است محاسبه کند. این همان odometry هست که در تمرین‌ها با آن آشنا شدیم و در یک ربات مسطح می‌توان آن را به عنوان تخمینی از موقعیت ربات نسبت به سیستم مختصات جهان نام برد. ربات را می‌توان به منظور کاهش خطای سیستماتیک ناشی از مقادیر پارامترهای نادرست کالیبره کرد، اما خطاهای غیر سیستماتیک ناشی از عوامل محیطی مانند لغزنده بودن کف و شتاب‌های بیش از حد، قابل اجتناب نیست. در SLAM از روش loop closure به منظور تصحیح خطای odometry استفاده می‌شود. شیوه حلقه بسته تلاش دارد تا بفهمد که آیا ربات قبلاً منطقه‌ای را که در آن قرار دارد دیده است یا خیر. این روش برای جلوگیری از افزودن اطلاعات اضافی (اطلاعات یک منطقه از قبل دیده شده) و در نتیجه اجتناب از هزینه محاسباتی غیر ضروری نیز انجام می‌شود.

Planning:

مرحله برنامه ریزی شامل یافتن مسیر بهینه بین موقعیت فعلی و موقعیت هدف با اجتناب از هرگونه مانع است. برنامه ریزی را می توان به دو دسته تقسیم کرد:

- Off-line path-planning (Global planning): در برنامه ریزی خارج از خط، قبل از برنامه ریزی ربات باید محیط را بشناسد و یک نقشه از محیط را دارا باشد. برخی از این نوع الگوریتم ها عبارتند از: الگوریتم Dijkstra و الگوریتم A*. در برنامه ریزی آنلاین، ربات باید خود را به طور مداوم مکان یابی کند. برای انجام این کار معمولاً از فناوری LiDAR استفاده می شود.
- On-line path-planning (Local planning): در این روش ربات نقشه ای از محیط را از پیش ندارد و باید در طی برنامه ریزی آن را بسازد. برای محیط های متغیر و پویا از این روش استفاده می شود. Vector Field Histogram و Dynamic Window Approach از انواع آن هستند.

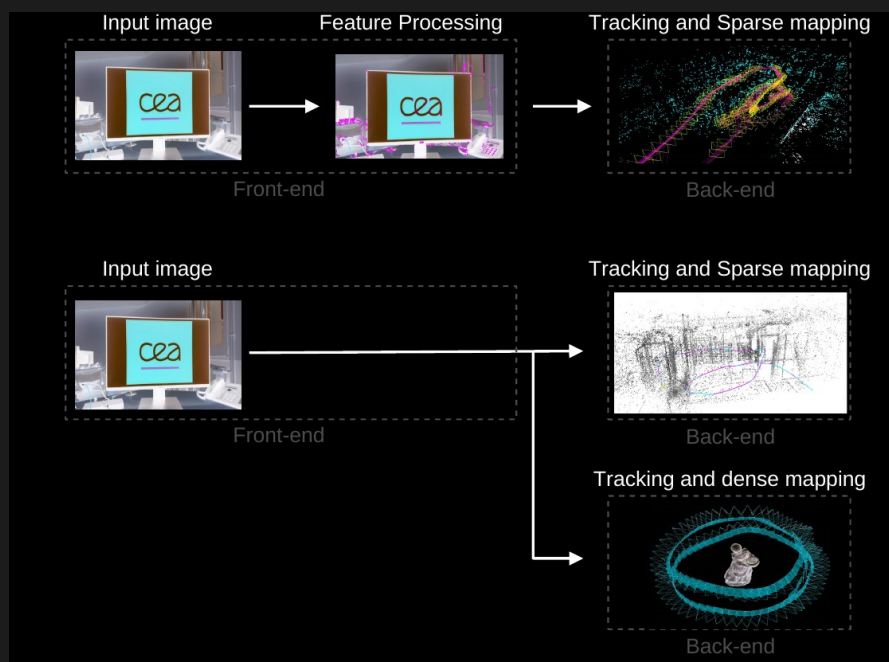
بنابراین SLAM در تلاش برای مکان یابی در حین نقشه برداری است.

vSLAM

همانطور که از نام آن پیداست، Visual SLAM (یا vSLAM) از تصاویر به دست آمده از دوربین ها و سایر حسگرهای تصویر مانند دوربین های تک چشمی، دید استریو، دوربین های همه جهته یا RGB-D برای مکان یابی و نقشه برداری محیط استفاده می کند. درک نحوه کار vSLAM آسان است. هدف چنین سیستمی تخمین حرکت دوربین با توجه به جابه جایی پیکسل ها در تصاویر متوالی است.

می توان این شیوه را با هزینه کم و با دوربین های نسبتاً ارزان اجرا کرد. علاوه بر این، از آنجایی که دوربین ها حجم زیادی از اطلاعات را ارائه می دهند، می توان از آن ها برای تشخیص نقاط عطف یا landmark ها (موقعیت های اندازه گیری شده قبلی) استفاده کرد. تشخیص نقطه عطف همچنین می تواند با بهینه سازی مبتنی بر گراف ترکیب شود و در اجرای SLAM به انعطاف پذیری دست یابد.

تکنیک‌های اولیه vSLAM با دوربین‌های تک‌چشمی انجام می‌شدند و نقاط ویژگی (feature points) تصویر را شناسایی می‌کردند که با نام «feature-based approach» نیز یاد می‌شوند. در طول زمان vSLAM به سمت استفاده از «رویکردهای مستقیم» رفته است، یعنی الگوریتم‌هایی که می‌توانند از کل تصویر برای ردیابی ربات استفاده کنند. شیوه شناخت ویژگی در قسمت بالا و شیوه مستقیم در قسمت پایین شکل زیر نمایش داده شده است:



vSLAM همچنان یک فناوری در حال ظهور است؛ شیوه‌های یادگیری عمیق، مولتی مودال، تقویتی و ... پیوسته برای حل مسائل SLAM در پژوهش‌های نوین بکار گرفته می‌شوند.

در این فاز قصد داریم یک آشنایی خیلی ساده از فرایند SLAM داشته باشیم. بخش وسیعی از شیوه‌های SLAM ای که امروزه استفاده می‌شوند مبتنی بر اسکن لیزری هستند، ولی از آنجایی که به این سنسورها دسترسی نداریم، می‌توانیم تصویر عمقی را به اسکن لیزری تبدیل کنیم (در تمرین‌ها این کار انجام شده است).

این شیوه قدرت و کیفیت اسکن لیزری واقعی را ندارد و نمی‌تواند نتایج مطلوب را به ما دهد. با این حال قبل از رفتن به سمت visual SLAM در فازهای بعدی، در این فاز سعی می‌کنیم با استفاده از شبیه‌ساز ابتدا eddie را در یک محیط جدید حرکت دهیم تا نقشه‌ای از آن را بسازد و سپس با استفاده از نقشه ذخیره شده مسیریابی انجام دهیم.

برای تمام فعالیت‌ها فقط از یک workspace استفاده کنید. در صورت استفاده از workspace های متفاوت به ازای فعالیت‌های مختلف، نمره منفی داده می‌شود.

فعالیت ۱

جدید ترین تغییرات مخزن [arashsm79/eddiebot-ros](https://github.com/arashsm79/eddiebot-ros) را در ws خود pull کنید و با rosdep وابستگی‌ها را نصب کنید.

با استفاده از دستور

```
ros2 launch eddiebot_gazebo eddiebot_gz_sim.launch.py world:='maze_tight'
use_sim_time:='true'
```

گزیبو را اجرا کنید و شبیه‌سازی را شروع کنید.

با استفاده از

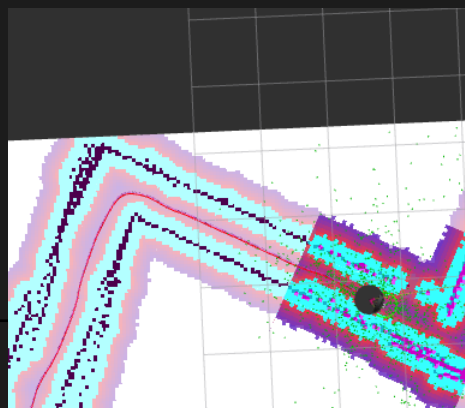
```
ros2 launch eddiebot_rviz view_robot.launch.py use_sim_time:=true
```

rviz اجرا کنید.

سپس به پوشه eddiebot_nav/maps رفته و دستور زیر را اجرا کنید:

```
ros2 launch eddiebot_nav slam.launch.py sync:=false use_sim_time:=true
```

با استفاده از Teleop در Gazebo ربات را به آرامی تکان دهید و سعی کنید محیط را در rviz نقشه برداری کنید. (دقت کنید نقشه درست شده بی نقص نیست و شامل ناهمگانی‌های زیادی خواهد بود زیرا slam_toolbox برای اسکنرهای لیسری واقعی با برد و زاویه دید زیاد طراحی شده است و همچنین از حسگر IMU برای odometry استفاده نمی‌شود)



نیازی به نقشه برداری کل دنیا نیست، نقشه برداری تا انتهای پایین نقشه کافیست.

در نهایت با دستور:

```
ros2 service call /slam_toolbox/save_map slam_toolbox/srv/SaveMap "name: data: 'maze_tight'"
```

می‌توانید نقشه تولید شده را ذخیره کنید. (در محلی که slam را اجرا کردید ذخیره می‌شود).

فعالیت ۲

همه برنامه‌ها را ببندید. Gazebo و rviz را با دستورات قبلی دوباره اجرا کنید. با دستور زیر

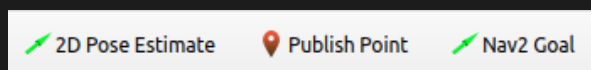
```
ros2 launch eddiebot_nav localization.launch.py use_sim_time:=true  
world:="maze_tight.yaml"
```

نودهای مربوط به مکان‌یابی و با دستور

```
ros2 launch eddiebot_nav nav2.launch.py use_sim_time:=true
```

نودهای مربوط به [navigation2](#) را اجرا کنید. مطمئن شوید که شبیه‌سازی در حال اجرا است.

در rviz ابتدا با استفاده از 2d pose estimate مکان و جهت ربات (با نگه داشتن کلیک) را مشخص کنید و سپس با Nav2 Goal یک مکان از نقشه را به دلخواه مشخص کنید تا برنامه‌ریزی برای آن انجام شود و ربات به حرکت در بیاید.



حرکت ربات در Gazebo و rviz را مشاهده کنید. (نیازی نیست ربات به انتهای مسیر برسد و یا مسیریابی بدون خطا انجام شود)

فعالیت ۳

با محتویات لانچ فایل‌ها و پکیج‌های استفاده شده آشنا شوید.

نمره اضافه

برای بهبود مسیریابی و نقشه برداری می‌توانید فایل‌های کانفیگ مربوطه را مطالعه و تغییر دهید. با توضیحات کامل دلیل تغییرات خود را نوشته و دلیل بهبود را توضیح دهید.

تحويل

از نتیجه تمرین‌ها اسکرین‌شات گرفته و در یک فایل PDF با نام و نام خانوادگی و شماره دانشجویی قرار دهید و همراه با تمام کدها در یک فایل ZIP در سامانه آموزش مجازی دانشگاه ارسال کنید.