

در این آزمایش قصد داریم به mobile robot طراحی شده در آزمایش قبلی، odometry اضافه کنیم تا ربات بتواند بر اساس میزان چرخش و سرعت چرخ‌هایش مکان فعلی خود را تقریب بزند. odometry در مورد استفاده از داده‌های حسگرها برای تخمین تغییر در موقعیت، جهت‌گیری و سرعت ربات در طول زمان نسبت به نقطه‌ای است (به عنوان مثال فریم‌ای که این نقطه مرکز آن است می‌تواند odom باشد و مرکز آن در $x=0$ ، $y=0$ ، $z=0$ باشد). اطلاعات odometry معمولاً از حسگرهایی مانند wheel encoder و IMU بدست می‌آید. نهایتاً تبدیل بین فریم ربات و فریم odom باید در tf ارسال شود تا بشود مکان ربات را در دنیا تقریب زد.

در ROS، فریم مختصاتی که برای odometry استفاده می‌شود، به عنوان فریم odom شناخته می‌شود. فریم odom نقطه‌ای در جهان است که ربات برای اولین بار شروع به حرکت می‌کند. موقعیت و جهت‌گیری یک ربات در فریم odom با گذشت زمان و مسافت کم دقت‌تر می‌شود زیرا حسگرهایی مانند IMU (که شتاب را اندازه می‌گیرند) و wheel encoders (که تعداد دفعات چرخش هر چرخ را اندازه می‌گیرند) بدون خطا نیستند. به عنوان مثال، تصور کنید ربات شما به دیوار برخورد کند. چرخ‌های آن ممکن است به طور مکرر بچرخند بدون اینکه ربات به جایی حرکت کند (ما به این لغزش چرخ می‌گوییم). در این حالت odometry چرخ فکر می‌کند که ربات مسافتی را طی کرده است، در حالی که درجا زده است؛ ما باید این نادرستی‌ها را مدیریت کنیم. بنابراین نحوه ارتباط فریم‌ها به این صورت می‌شود:

1. map => odom
2. odom => chassis
3. chassis => base_laser (sensor base frames)

تبدیل اول معمولاً توسط سیستم localization and mapping ارسال می‌شود (ما در این آزمایش تبدیل دستی و استاتیک می‌دهیم). تبدیل دوم توسط سیستم odometry مثل IMU ارسال می‌شود. تبدیل سوم توسط robot_state_publisher و فایل URDF ربات ارسال می‌شود.

فعالیت ۱

این تمرین نیز در دنیای maze انجام می‌پذیرد. ابتدا یک caster wheel دیگر به انتهای ربات اضافه کنید تا ربات متعادل تر شود. سپس در مستندات gazebo صفحه مربوط به این پلاگین را پیدا کنید (detail level) را روی ۵ بذارید تا بتوانید راحت‌تر جست‌وجو کنید). تگ‌های لازم را به پلاگین اضافه کنید تا odometry را به تایپیک odom / ارسال کند بجای تایپیک پیشفرض. تایپیک را echo کنید و از خروجی ترمینال همراه با ربات اسکریت شات بگیرید.

Odometry بین کدام دو فریم در حال ارسال است؟

فعالیت ۲

مانند آزمایش قبل پل مربوط به LaserScan را وصل کنید. یک پل دیگر بسازید که تایپیک tf مربوط به ربات که gazebo استفاده می‌کند را به ros2 می‌آورد و سپس آن را به tf بازنگاشت می‌کند. (یادآوری به نحوه اضافه کردن remapping در لانچ فایل مراجعه کنید)

برای آشنا شدن با نوع پیام تایپیک در gazebo از دستور زیر استفاده کنید.

```
gz topic -i -t /topic_name
```

با استفاده از static_transform_publisher که در آزمایش قبل دیدیم، دو تا نود تبدیل ایستا بسازید:

1. gpu_lidar را به chassis وصل کنید.

2. odom را به world وصل کنید.

یک لانچ فایل بنویسید که تمام کارهای این فعالیت را انجام می‌دهد.

فعالیت ۳

در rviz2 ابتدا LaserScan را اضافه کنید و اندازه Grid را بیشتر کنید. تایپیک‌های مورد نیاز را مشخص کنید و مطمئن شوید که tf خطاری ندارد. مقدار size را در LaserScan بر روی 0.1 قرار دهید.

حال با استفاده از teleop ربات را تکان دهید، باید بتوانید حرکت فریم ربات در rviz2 را ببینید و مقادیر سنسور LIDAR بسته به جهت ربات، در سمت مناسب از جهان قرار داشته باشند.

نمره اضافه

مدل ربات را به URDF تبدیل کنید (فقط مدل ربات نه جهان) و سپس آن را با استفاده از robot_state_publisher در rviz نمایش دهید. کاری کنید که در فعالیت ۳ به جای صرفاً فریم ربات، خود مدل ربات نمایش داده شود. می‌توانید از مخزن eddiebot-ros کمک بگیرید. در این حالت آیا به همه تبدیل‌های ایستا فعالیت ۲ نیاز داریم؟

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.