

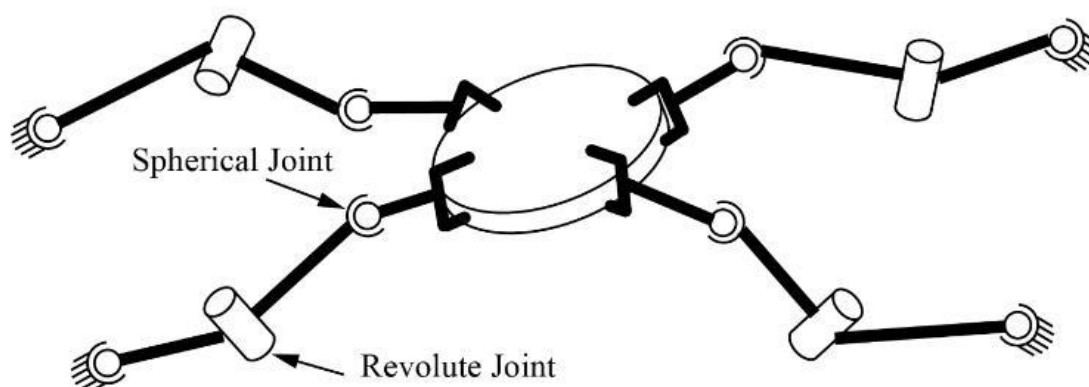
انجام این تمرین به صورت انفرادی می باشد

## هدف از انجام این تمرین

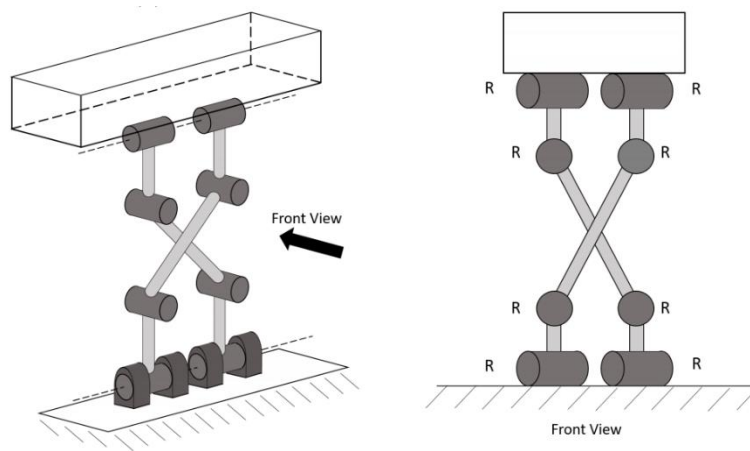
هدف این تمرین مرور برخی از مفاهیم پایه مطرح شده در کلاس درس و همچنین آشنایی با مفاهیم پایه ROS است. برای انجام بخش پیاده سازی این تمرین، در ابتدا ویدیو ارائه شده برای نصب ROS و آموزش مفاهیم پایه ای آن را در ابتدا مشاهده کنید. ROS یا سیستم عامل ربات، چارچوب نرم افزاری برای توسعه ی نرم افزار ربات است. ROS کتابخانه و ابزارهایی برای توسعه دهندگان نرم افزار، جهت ساخت کاربردهای نرم افزاری رباتیک فراهم می کند. این سیستم لایه انتزاعی سخت افزاری، راه اندازهای دستگاهها، کتابخانهها، ابزارهای بصری، ارسال پیام ها، مدیریت بسته ها و... را در اختیار کاربران قرار می دهد. لطفا نسخه ی مطرح شده در هندزان را مطابق با دستورالعمل ارائه شده در ویدیو نصب بفرمایید.

## بخش تئوری

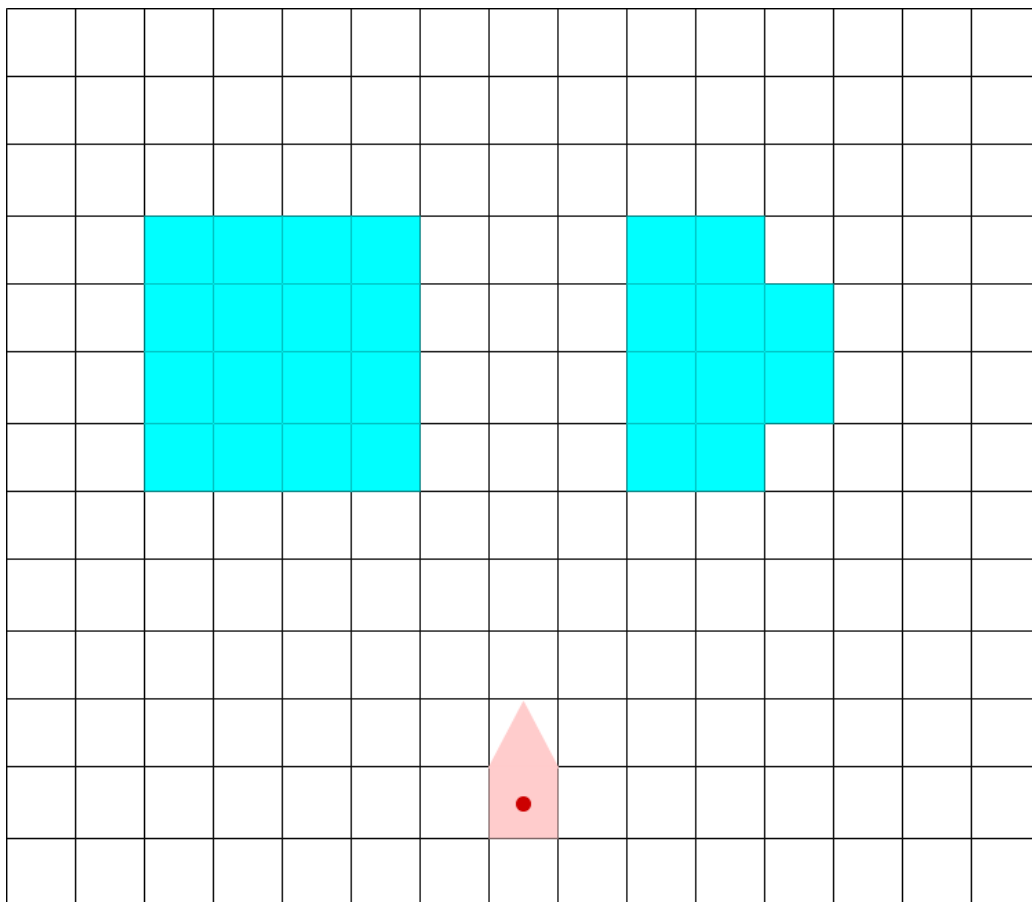
۱. ربات زیر از ۴ بازوی SRS تشکیل شده است که یک دیسک را نگه داشته اند. درجه آزادی آن را به دست آورید.  
(راهنمایی: چهار مفصل واقع شده در چهار انتهای ربات به زمین متصل شده اند.) (بارم: ۵ امتیاز)



۲. برای ربات زیر درجه آزادی را به دست آورید. (تصویر، ربات را از دو نما نشان می دهد) (بارم : ۵ امتیاز)



۳. برای ربات مشخص شده در تصویر زیر، C-space را مشخص کنید. (فرض کنید که ربات تنها امکان دوران ۹۰ درجه را دارد) (بارم: ۱۰ امتیاز)



## بخش شبیه‌سازی

### ❖ شرح سناریو

در این بخش سعی داریم با کمک ROS سیستمی طراحی کنیم که دانشجویان مهندسی کامپیوتر را بر اساس گرایشی که دارند دسته‌بندی کند. هدف از این بخش تنها آشنایی با برنامه‌نویسی سمت سرور ROS بوده و وارد برنامه‌نویسی ربات در این تمرین نمی‌شویم. هرچند در بخش دوم از شما می‌خواهیم که کنترل ربات را در شبیه‌ساز Gazebo به صورت دستی به دست بگیرید.

### ❖ توضیح گام‌های انجام سناریو

**گام اول (۵۵ امتیاز):** لازم است که فایل `random_student.py` را دریافت کرده و با کمک آنها شروع به تولید داده در گره `student_request` کنید و در تاپیک `std_request` قرار دهید. دقت کنید همانطور که در ویدیو تدریسیاری اول توضیح داده شده است، پس از ساخت پکیج مورد نظر برای گام اول، فایل `random_student.py` را به عنوان یک گره وارد پکیج ساخته شده بکنید و `custom message` مورد نظر خود را با توجه به مراحل که در ویدیو توضیح داده شده، بسازید. فیلدهای این `message` را با توجه به کد زده شده در فایل `random_student.py` تعریف نمایید.

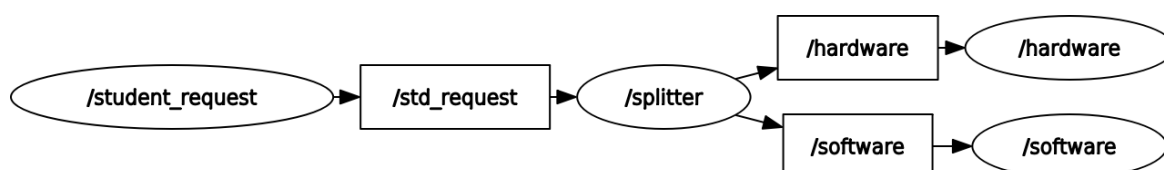
`message` تعریف شده توسط تاپیک `std_request` به گره `splitter` منتقل می‌شود. وظیفه این گره این است که با توجه به فیلد `departement` دانشجویان را به دو گروه `software` و `hardware` تقسیم کرده و در دو تاپیک `software` یا `hardware` آن‌ها را `publish` کند.

در انتها نیز گره‌ها با نام‌های `software` و `hardware` وظیفه چاپ اطلاعات دریافتی را دارند.

در آخر برای اطمینان از درستی کارکرد برنامه می‌توانید دستور زیر را اجرا کنید:

```
rqt_graph
```

خروجی دستور می‌بایست مشابه با تصویر زیر باشد :



**گام دوم ( ۲۵ امتیاز ) :** در این بخش می‌خواهیم کنترل turtlebot را در Gazebo به دست بگیریم. پیش از شروع این بخش دستورات زیر را اجرا کنید:

```
cd ~/catkin_ws/src/  
git clone -b noetic-devel https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git  
git clone -b noetic-devel https://github.com/ROBOTIS-GIT/turtlebot3.git  
git clone -b noetic-devel https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git  
cd ~/catkin_ws && catkin_make
```

این دستورات، پکیج‌های مورد نیاز برای کار با turtlebot را نصب می‌کنند.

پس از نصب پکیج‌ها، برای این سناریو موارد زیر باید انجام شوند:

- پکیج مربوط به turtlebot انواع مختلفی از این ربات را با فیزیک متفاوت در اختیار ما قرار می‌دهد. برای این سناریو از ربات waffle استفاده کنید.
- مختصات اولیه ربات را در empty world نقطه‌ی (۳،۲) در نظر بگیرید.
- حال به کمک گره teleoperation حرکت ربات به وسیله کیبورد را بدست بگیرید.
- در آخر سه مورد بالا را یکبار دیگر با ایمپورت کردن محیط funky-maze.world پیاده سازی کنید با این تفاوت که نقطه شروع را می‌توانید به صورت دلخواه غیر از مبدا انتخاب کنید. (تنها تفاوت در محیط ربات اعمال می‌شود که جای empty world از این فایل برای محیط ربات در Gazebo استفاده کنید)

**راهنمایی:** برای مورد چهارم باید در مسیر turtlebot3\_simulations/turtlebot3\_gazebo/launch یک فایل launch. مشابه فایل turtlebot3\_empty\_world.launch ایجاد کنید و به جای empty.world، فایل funky-maze.world را قرار دهید. حال می‌توانید با دستور roslaunch، محیط مورد نظر را بالا بیاورید.

تمامی دستورات استفاده شده در گام‌های این سناریو، به همراه اسکرین شات‌های مربوط به هر بخش، داخل گزارش قرار بگیرد.

## نکات تکمیلی در باب تحویل تمرین

۱. تحویل گزارش در یک فایل pdf مطابق با قالب قرار گرفته در سایت کورسز و با نام‌گذاری HW1\_StudentNumber می‌بایست تحویل داده شود.
۲. تحویل کدها در پوشه‌ای با نام Codes صورت گیرد. در نهایت pdf گزارش و پوشه‌ی Codes را zip نموده و یک فایل zip بارگذاری نمایید. فرمت فایل زیپ نیز به صورت HW1\_StudentNumber باشد.
۳. افراد می‌بایست تمرین را در صورت انفرادی بودن به تنهایی و در صورت گروهی بودن تمرین، با اختیار خود انفرادی یا گروهی انجام دهند. توجه فرمایید که لزوم تحویل تمرین به صورت گروهی نام‌نویسی در شیت مندرج در شیوه نامه تا تاریخ 15 اسفند 1400 است و پس از آن امکان ثبت نام اعضا وجود ندارد.
۴. در صورت احراز مشابهت در گزارش یا در کد توسط تدریس‌یاران، نمره هر دو گروه بدون هیچ‌گونه اغماضی صفر لحاظ خواهد شد.
۵. 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.
۶. برای انجام تمرین و پروژه تنها زبان برنامه‌نویسی مجاز پایتون می‌باشد.
۷. **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.**
۸. تاریخ تحویل تمرین ۱۴۰۰/۱۲/۲۷، ۱۱:۵۹ شب، می‌باشد و سیاست‌های تاخیر مطابق با موارد ذکر شده در شیوه‌نامه لحاظ خواهد شد. خواهشمندیم برای کسب اطلاع شیوه‌نامه‌ی آموزشی تمرینات درس را به طور کامل مطالعه فرمایید.
۹. در صورت وجود هرگونه ابهام یا سوال با مسئولان تمرین جناب آقای عسکری و آقای آخوندی از طریق ایمیل های زیر در ارتباط باشید.

[a.akhoundi79@gmail.com](mailto:a.akhoundi79@gmail.com)

[askaria079@gmail.com](mailto:askaria079@gmail.com)

**موفق باشید**