CSCE 452/752 Robotics and Spatial Intelligence, Fall 2024

Project 2

Assigned: 2024-09-25
Due: 2023-10-10, 11:59pm

Overview and purpose

This assignment asks you use the `sim2` simulation, an enhanced version of `sim1` from the previous project, to control the simulated robot to visit collection of sites efficiently and accurately. The purpose is to give some experience using ROS in a non-trivial context, along with some experience controlling the motion of (simulated) robots.

Logistics

This assignment is intended to be completed in groups of size 3. Nevertheless, it may be completed in teams of size 1, 2, or 3. Expectations and grading standards will be the same regardless of team size. The instructor will not mediate conflicts between team members.

You should assign yourself to a team within the Canvas system —Use the People link within the course page— by noon on **Tuesday, October 1**. After this date, teams for this project will be final. **If you are choosing to work alone, this step is still required**; simply add yourself to a group with no other members. After this date, teams for this project will be final.

Exactly one member of each team should submit.

Reference material

In addition to the concepts you used in Project 1 (nodes, topics, publishers, subscribers, etc.), you'll need to use **services**, **parameters**, and **bags**. You will likely want to consult the tutorials and reference documentation on those subjects.

Provided software

Download the newly-updated version `sim` package from this link and extract it into the `src` subdirectory in your ROS workspace.

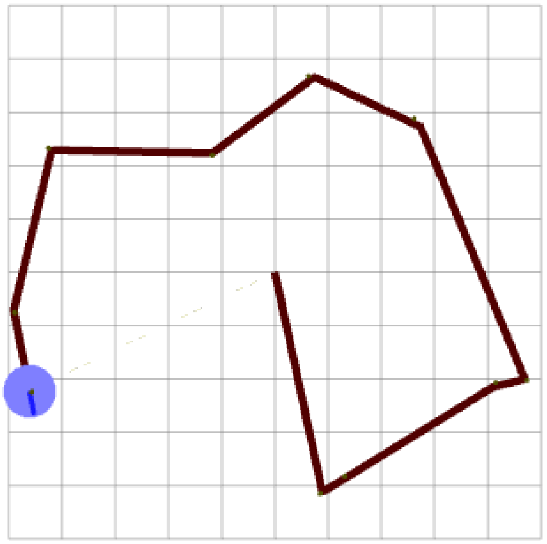
ROS2 `sim` package

You should use this package to complete this assignment, but should not modify anything in the `sim` package.

Your task

Create a ROS node that uses the `sim2` simulator to draw a line that **visits each point in a given list of targets**. The robot's movement should be not accurate (i.e. minimal extraneous movement) and fast.

Here is an example of what a visualization of the completed project might look like:



Some details about how `sim2` handles the targets:

- The `sim2` node accepts a string **parameter** called `targets` that gives the name of the file from while the target points will be read. The `sim2` node will look first in the current directory and then in the `share` directory of the `sim` package for this file. By setting this parameter, you can test your program with the sample target files provided with the `sim` package and also with your own hand-crafted targets. You will want to learn how to include parameter values on the `ros2 run` command line.
- The targets file is simply a list of (x, y) points in YAML format.
- The visualization markers published by `sim2` include small cylinders at each of the target locations. Your goal is to get the center of the robot to each of these locations. When a target is successfully visited, its color in this visualization will change.

- You may discover that `sim2` publishes some other information that you'll find helpful. You are encouraged to explore its behavior.
- The `sim2` node provides a `reset` service that returns everything to is initial state. This may be useful for testing your own program multiple times without restarting `sim2`.

Some details about how your program must work:

- Before moving the robot, your program should clear any existing drawing using the `reset` service offered by `sim2`.
- The robot **does not** need to visit the points in the order in which they are given. On the contrary, your program should make intelligent choices about the order to visit the points to keep the travel time small.
- You should move the robot by publishing messages on the `cmd_vel` topic. Your program must publish at least 10 messages per second on this topic. This is a minimum; you may publish more rapidly than this if you like. You can check this using the `ros2 topic hz` command.
- You should visualize the results using `rviz` as in Project 1, including displays for the Robot Model and the Marker Array of simulation markers. It is recommended to use a white background.

The results will be evaluated based on adherence to these requirements, along with the speed of execution, the correctness of the results, and quality of your code.

What to submit

~~Specific submission requirements, which will include recordings of your program's behavior using the `ros2 bag` tool, will be posted soon.~~

You should submit a **single zip archive** containing everything listed below:

- A **PDF** report including:
    1. A concise description of how the system works. This should be a couple of paragraphs, describing your overall approach at a conceptual level. Some questions to consider: What choices did you make in designing the program, and why? What nodes does your system contain, and what role does each one play? What specific constants are part of your method, and how did you select values for those constants?
    2. A short answer to these questions: Did the results meet your expectations? Why or why not?
- The source code for your program(s). This will span at least one source code file. Be sure to consult the source code style guide on the course website.

- Recordings created by `ros2 bag record` of **all topics** throughout the **entire execution** of your program for **each** of the five input files included in the archive below:

  Project 2 Inputs

  It is essential for your submitted bags to contain **all topics** because this information will play a primary role in evaluating the correctness of your solution. **We cannot award credit for messages published on topics that are not recorded in the rosbags you submit.**

  The directory of each bag file must be named with your two digit group number for this project, followed by a dash, followed by the number of the input bag file. For example, if you are in Group 3, your submission should include bag directories called `03-1`, `03-2`, and so on. **Incorrectly named bags cannot be graded.**

  Each rosbag recording is a **directory** which contains both a db3 file and a yaml file of metadata. You may rename this directory, but **you should not modify the names or contents of the db3 or yaml files after recording**, or you risk corrupting the rosbag. **Corrupted rosbags cannot be graded.**

You may (but are not required to) also include:

- A short video demonstrating your system in some way that may be helpful in grading.

[Submit via Canvas.]