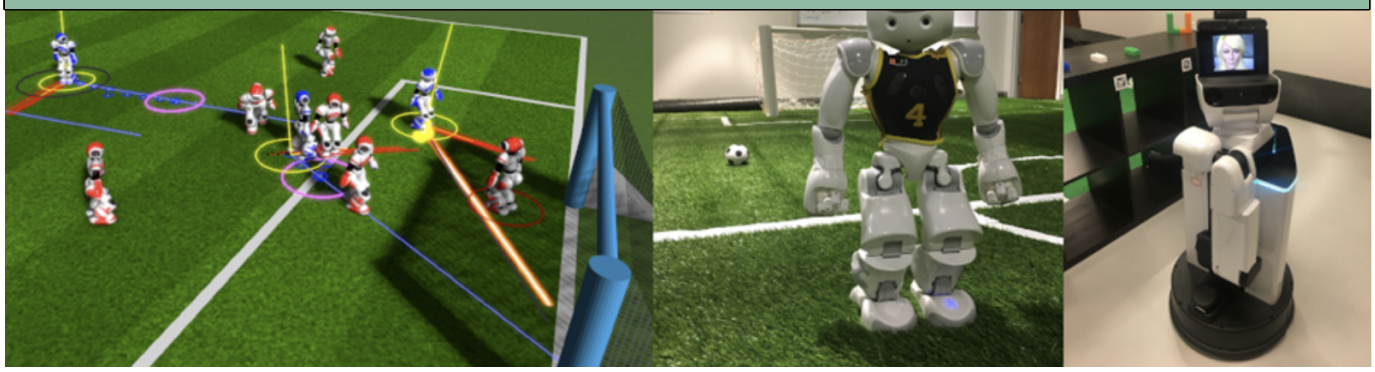
Fall 2022 - CSC752 Autonomous Robotic Systems - Assignment 3

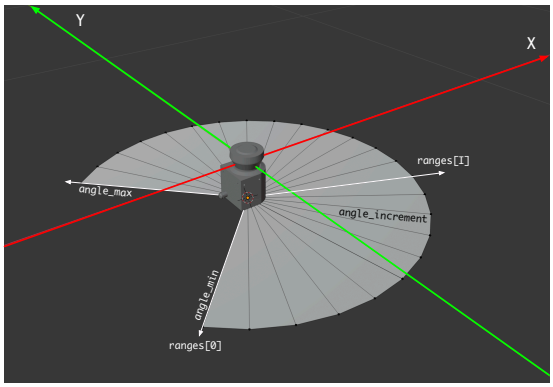


Due date: 10/06/2022, 11:00am, before class starts. This assignment is worth 20 points.

The goal of this exercise is to create an entire control loop for the HSR. You will extract the position of an object in the scene from the laser scan and then control the robot such that it drives towards that object.

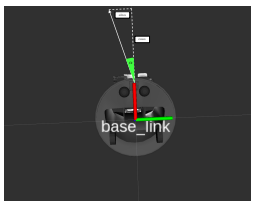
- Adapt your launch file from the last exercise such that:
  1. `$(find [USERNAME]_hrsb_behavior_controller)/worlds/singleObject.world` is loaded as the world. Check it out from svn://svn.cs.miami.edu/classes/csc752.231/worlds/singleWorld.world or download the `singleObject.world` file from our homepage, move it to your `worlds` folder and add and check-in the file to your repo.
  2. Create a new ROS node that makes the HSR move in front of the object.
- Write a simple controller that drives the HSR towards the bookshelf and stops before the shelf. You can use the code from the callback method of the laser scan topic for assignment 2.
- You might need to add further data structures, e.g., `geometry_msgs` or `actionlib_msgs` as a dependency to your `CMakeLists.txt` and `package.xml`. Which ones depends on your approach to solve the problem.
- Add a RobotModel plugin to RViz to visualize the HSR robot.
- Add a TF display plugin to RViz.
- Implement a TF listener to transform the extracted point from the laser frame to the odom frame. Refer to figures 1a–1c for the visual explanation. Publish the point in the odometry frame as a RViz marker.
- Add and commit the complete directory "assignment3" with all necessary files to the repository.

Approach: Extract the position of the bookshelf from the laser scan with respect to the robot and make the robot move towards that position. Extracting the position requires a little bit of basic math and making the robot move can be done by are more than one way: (a) by using a topic `/goal`, (b) by using the `hsrb_interface` (this is what we did with `ihsrb`, (c) moving the robot using `tf`, and (d) by using an actionlib, e.g. `move_base_msg`. All of these options have descriptions in either the hsr.io documents or on the Internet as ROS examples.



(a) Angles for the Hokuyo laser scanner of the HSR from angle_min to angle_max with its increments. Further details in `hokuyo_laser.gazebo.xacro`.

(b) The `base_link` coordinate system of the HSR. X (red) is forward, Y (green) is left and Z (blue) is up. The `base_range_sensor_link` (the scanner_link) is in line with the `base_link` and not rotated.



(c) For twisting the robot you need to calculate angles. This figure shows how to do this.

Figure 1: Material to better understand the TF system of the HSR.