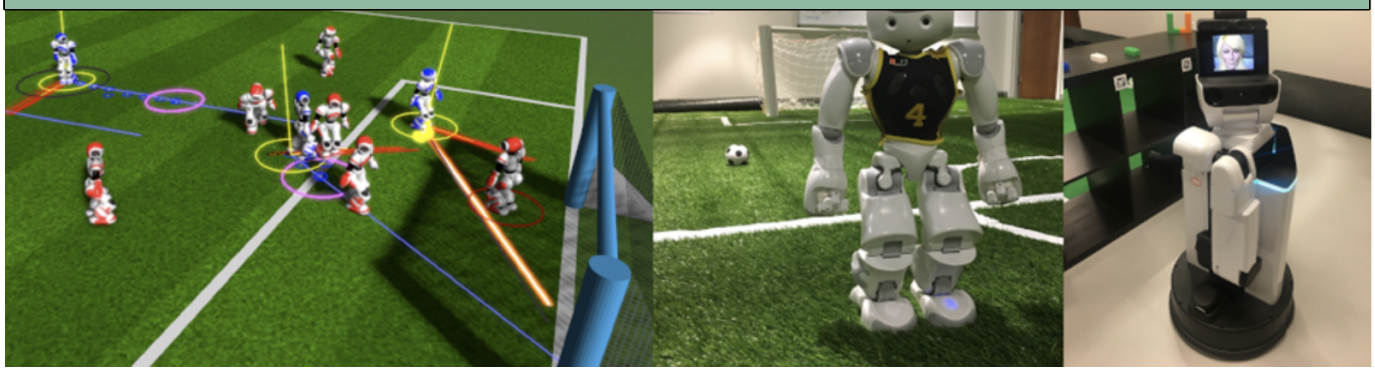
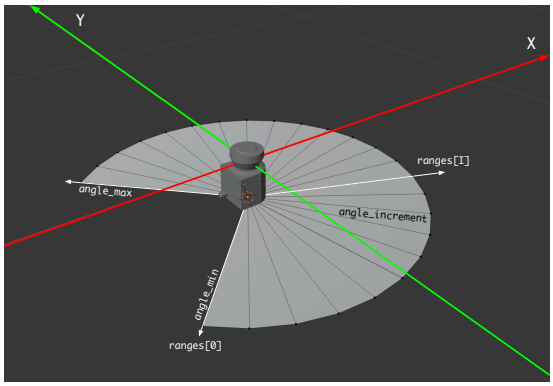
# Fall 2024 - CSC398 Autonomous Robots - Assignment 3



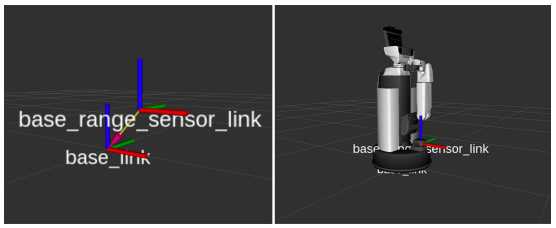
**Due date: 09/26/2024, 12:30 pm, before class starts. This assignment is worth 20 points.**

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. What are we asking you to do?

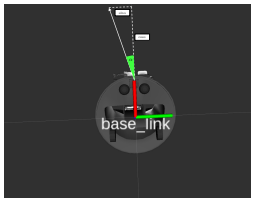
1. We made some small modifications to the simulation world. `git pull` (from your catkin workspace) to update.
2. `cd` into the `src/` directory off your catkin workspace. Clone the new assignment: `git clone` https://classroom.github.com/a/b6GPCI69.
3. Create a new package from scratch. Use the command `catkin_create_pkg` to create a new package with the name of your repository (has to be the exact name!) and the dependencies `rospy` (or `roscpp` if you prefer to program in C++), `std_msgs`, `sensor_msgs`, and `geometry_msgs`. Refer to the tutorial "Creating a ROS Package" if you are stuck: http://wiki.ros.org/ROS/Tutorials/CreatingPackage.
4. Create a new ROS node that drives the HSR towards the minimum distance point (at the green object) and stops before that object. You can use the code from the callback method of the laser scan topic from assignment 2.
5. Once the robot reaches its destination, get the x,y position of the robot (use /base_link) with respect to the world. For this, implement a TF listener to transform the position of the robot to the `/map` frame. Refer to figures 1a–1c for the visual explanation. Publish the point in the odometry frame as a RViz marker.
6. Add a TF display plugin to RViz. Disable all frames and re-enable /base_link and /map.
7. Add, commit, and push the entire package to github classroom.




(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.