Humanoid Robot Project: Autonomous Navigation

Summary

Using ROS 2 Humble, packages were implemented for detecting laser scanner data, using the data to generate a map, and using the generated map for navigation purposes. The respective packages can be found at the following links:

1. urg_node2: https://github.com/Hokuyo-aut/urg_node2
2. slam_toolbox: https://github.com/SteveMacenski/slam_toolbox/tree/humble
3. nav2: https://github.com/ros-navigation/navigation2/tree/humble

Current issues include the ethernet connection dropping and missing transform frames preventing map generation with slam_toolbox. After fixing these issues, navigation can be implemented using navigation2.

ROS 2 installation

The ROS 2 setup can be found on the official ROS website - it should be noted that the current setup for this project uses version humble, but any version should do provided the correct branches are used for each package.

https://docs.ros.org/en/humble/Installation.html

Workspace setup and package installation

Note: This section can be skipped by cloning the following repository, which already contains the necessary packages: https://github.com/zzaid17/hokuyo_laser_scanner_ws

---

After setting up the ROS 2 installation, a folder containing another src folder should be created to initialize the workspace - e.g. /workspace/src (this example directory will be referred to later in the document). The packages can be cloned into this workspace with the following terminal commands on Ubuntu 22.04:

```
git clone https://github.com/Hokuyo-aut/urg_node2.git
git clone https://github.com/SteveMacenski/slam_toolbox/tree/humble.git
git clone https://github.com/ros-navigation/navigation2/tree/humble.git
```

Dependencies

Rosdep can be used to handle dependencies automatically. It should be noted that rosdep seems to miss the BehaviorTree.CPP dependency, so this will need to be installed manually. Additionally, rosdep should be updated before attempting to install any dependencies.

```
git clone https://github.com/BehaviorTree/BehaviorTree.CPP.git
rosdep update
rosdep install -q -y -r --from-paths src --ignore-src --rosdistro humble
```

Source and build:

The workspace can be built using colcon:

```
colcon build --symlink-install
```

During development, it was noted that failing to source the proper bash files often resulted in compilation and build issues. Specifically, the following files must be sourced (these require folders that may not be generated until after building):

1. /opt/ros/humble/setup.bash
    a. The exact location of the /ros directory may depend on the ROS 2 installation
2. /workspace/install/local_setup.bash
3. /workspace/install/urg_node2/share/urg_node2/local_setup.bash
4. /workspace/install/slam_toolbox/share/slam_toolbox/local_setup.bash

Running nodes:

The nodes can be launched using the following commands:

```
ros2 launch urg_node2 urg_node2.launch.py
ros2 launch slam_toolbox online_sync_launch.py
```

Additionally, rviz can be used to visualize the scanner data to ensure that the map generated by slam_toolbox is satisfactory:

```
ros2 run rviz2 rviz2
```

Upon launching rviz2, the user must go to Add -> Create visualization > By topic and select the necessary topics (i.e. /map, /scan).

Additional information:

One of the issues faced was connecting the laser scanner via an ethernet interface. The initial value of the IP address for the scanner is 192.168.0.10, so the network settings on the computer being used to run the setup should match as per the following.

IP Address: 192.168.0.15
Subnet Mask: 255.255.255.0
Default Gateway: 192.168.0.1

Failure to follow this step may result in the error “could not open ethernet port” when attempting to run the scanner node.

Pending issues / TODO:

Despite using the correct network settings, network issues persist. Specifically, the ethernet connection between the scanner and the computer consistently drops after 20-30 minutes of use. The same problem was reproduced using a different ethernet cable and computer. Several other troubleshooting steps were also followed. These included restarting and reinstalling the network manager and disabling the USB auto-suspend setting (which is on by default).

The physical connection of the laser scanner was investigated and looked fine, though defective hardware could still be a factor. Another potential cause is the scanner overheating since it becomes warm to the touch after leaving it on for an extended period.

Additionally, when attempting to visualize the scanner data in rviz2, the map is not generated due to missing tf data. This requires setting up transform frames with the tf2 library, which may require a static transform broadcaster file, though this has not yielded any results so far. After successfully getting tf data, the next step would be to use the map generated by slam_toolbox to implement navigation using navigation2.