ME3720 Lab 2

February 18, 2022

1 Introduction

The objective is to get experience with robotic motion planning. The undersea environment is a subsea blowout preventer (BOP) structure. Your goal is to use the SRS Fusion to survey the BOP and report back the results. The SRS Fusion has an integrated camera as the primary extraceptive sensing system.

The survey will consist of a series of objectives each given as a 3D pose (position and orientation) that is the waypoint goal. When you achieve a waypoint goal, the next instruction (goal) will be given. The first and last objective will always be the same, the rest of the goals will randomly change with each iteration of the subsea BOP world.

Each waypoint goal has an AprilTag. These tags are similar to QR codes. When the camera has the April Tag in the field of view a background image detection process will use the AprilTag placard to provide the next instruction. The final objective is to autonomously navigate the fusion into the subsea housing bay located at the top of the structure. The team that accomplishes the goal with a combination of the best time and least distance traveled will be declared the winner and gain 5 points. There is a total of 30 points for the lab.

2 Blowout Preventer Environment

The BOP is a UUV_Simulator model `https://uuvsimulator.github.io/packages/uuv_simulator/docs/features/gazebo_worlds/#subsea-bop-panel` centered on [0,0,-60], with a height off the bottom of approximately 13 meters. The (x,y) BOP boundary coordinates are [4,4], [4,-4], [-4,-4], [-4,4], they are shown in figure 1. This can be used to help define the workspace for motion planning.

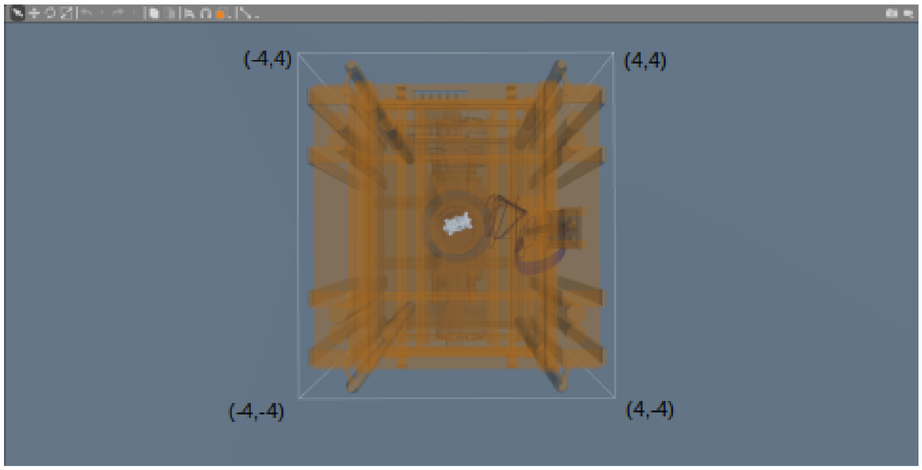

Figure 1: Overhead view of the BOP.

3 Getting Started

This lab uses the same approach as the first. A MATLAB m-script sending/receiving messages with the VMWare virtual environment running a physics-based simulation model with Ubuntu, ROS, Gazebo, and the UUV_Simulator. There are three parts to the lab:

1. **Week one**: Get the Fusion to navigation three-dimensional waypoints within the subsea bop environment where at the goal waypoint the vehicle must rotation to a given heading.
2. **Week two**: 3D motion planning in the subsea bop environment. Use either MATLAB or other motion planning scripts for navigating within the bop environment.
3. **Week three**: Run the complete scenario with the AprilTags providing random instructions for motion planning. The goal is to try and improve time and distance results of the total run. Complete the lab with a writeup of the results.

To get started from within the Ubuntu VM use the following commands:

1. ≫ roslaunch uuv_gazebo_worlds subsea_bop_panel.launch

2. ≫ roslaunch fusion_description upload.launch mode:= default x:=0 y:=0 z:=-10

3. ≫ roslaunch perc_apriltags perc_apriltags_random.launch

4. ≫ roslaunch fusion_control joy_wrench.launch

The last roslaunch command permits the Fusion to be controlled by a game control console. It is optional, it is not required for the lab, but can be useful for understand the control for operating in the subsea bop environment. Figure 2 is an image capture of the subsea environment with the images that are being captured and displayed by the Fusion camera.

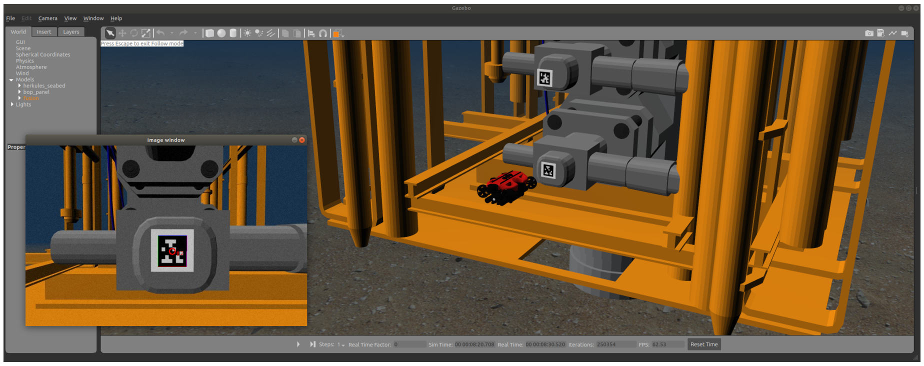

Figure 2: Gazebo/UUV_Simulator subsea BOP environment with the Fusion camera view and the detection AprilTag.

4 Phase II

Review the motion planning algorithms available in MATLAB. Start with typing “Motion Planning” in the MATLAB Search Documentation window. Become familiar with each of the algorithms. There are several options (not all listed):

1. plannerRRT
2. plannerRRTStar
3. plannerAStarGrid
4. plannerHybridAStar
5. Frenet Trajectory Generation

Each group will select an approach. This will need to be combined with the previous lab's PID controllers for depth and heading to navigate the random waypoints given from the AprilTags. In general the MATLAB script pseudo code is:

- Initialize system parameters
- Plan_path to the initial goal
- Navigate to the initial goal

  Handle errors in navigation
- When near the objective, orient the heading vehicle towards the AprilTag
- Repeat until you are finished:

  Receive the instructions from the April Tag

  Plan_path to the goal

  Navigate to the goal

  Handle errors in navigation
- Record the time and distance upon completion

When the Fusion AUV camera is in front of the AprilTag and within range, a ROS nav_msgs/Odometry message is sent with the pose (position and orientation of where the Fusion needs to be for the next AprilTag objective. This position should be used as the next mission objective. The name of the objective is in the child.frame_id field of the odometry message. The name of the rosmessage is 'apriltag_wypt_msg'.

4.1 Final Run

To fairly judge the best time for completion of the BOP survey conduct the following series of objectives.

1. Start on top of the BOP at position (-3,-3) see figure 3.
2. First objective is target 0 (see figure 4). This is the large top April tag on the north side.
3. Second objective is target 5 (see figure 5). This is the bottom April tag on the south side of the BOP.
4. Third objective is target 10 (see figure 6). This is the bottom April tag on the BOP panel on the east side.

5. Fourth objective is target 1 (see figure 7). This is middle April tag on the north side.

6. The final objective is the open oil pipe on the top of the BOP at position (0,0). (see figure 8).

Note that for this final run you will need to detect the Apriltag but you will not be using the Odometry message to determine the next objective. Be sure to record your time and your rosbag file and turn this in as part of the project.

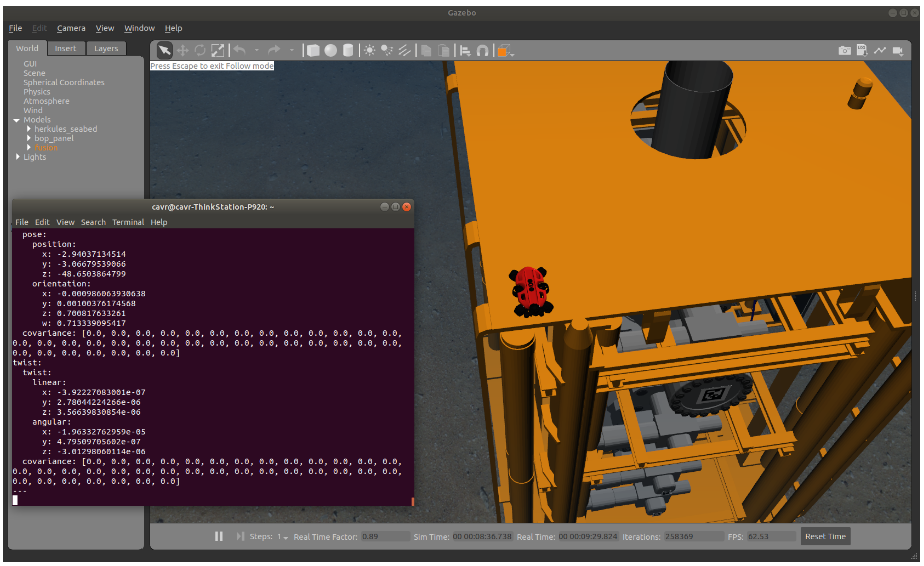

Figure 3: Start position on top of the BOP at (-3,-3).

5 April Tags

AprilTags (`https://april.eecs.umich.edu/software/apriltag`) are fiducial markers that can be used for determining vehicle pose (position and orientation) relative to the tag. If the position of the AprilTag is known, one can determine the position of the vehicle. We are not using this functionality for this lab. Figure 2 shows an example of an AprilTag within the subsea BOP environment. In the left side is a separate image that shows the camera field of view when detection of the tag occurs.

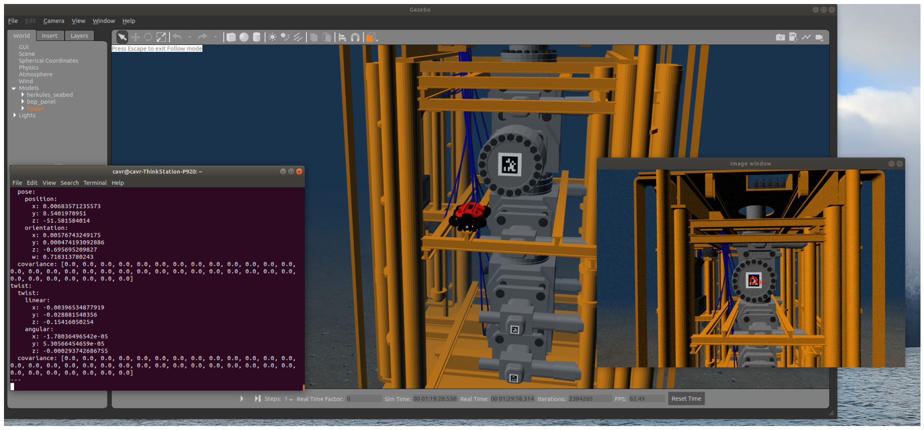

Figure 4: First objective target 0.

6 Software

For the software setup there are three choices:

1. Use the cloudlab web application at cloud.nps.edu.
2. Download the OVF.zip file from the Sakai Resources section (located at Box at: https://nps.box.com/s/g3tn9vtgh62j4hry728rdnzowieregbs). Uncompress the file in a directory and use VMware to open the OVF file. This will load the VMware Ubuntu OS with all the necessary upgrades.
3. Update your existing system with the following instructions.

If you are using a laptop with MATLAB on your native OS and ROS/Gazebo/UUV_Simulator on a VMware Ubuntu OS then you will need to update ROS software packages. The ROS packages are located on gitlab.nps.edu. Take the following steps:

1. $\gg$ cd /code/remus_ws/src/
2. $\gg$ git clone git@gitlab.nps.edu:me3720/perception/perc_apriltags.git
3. $\gg$ cd /code/remus_ws
4. $\gg$ catkin_make -DCATKIN_WHITELIST_PACKAGES="perc_apriltags"

Next update the uuv_simulator package:

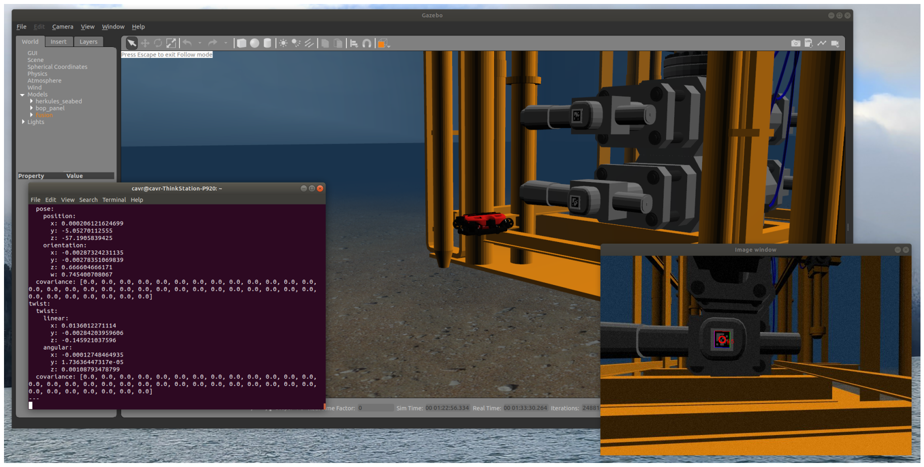

Figure 5: Second objective target 5.

1. ≫ cd /code/remus_ws/src/
2. ≫ sudo rm -r uuv_simulator
3. ≫ git clone git@gitlab.nps.edu:me3720/simulation/uuv_simulator.git
4. ≫ cd /code/remus_ws
5. catkin_make -DCATKIN_WHITELIST_PACKAGES=”uuv_simulator”

Next update the ros_msg_converter ROS package.

1. ≫ cd /code/remus_ws/src/
2. ≫ sudo rm -r ros_msg_converter
3. ≫ git clone git@gitlab.nps.edu:me3720/simulation/ros_msg_converter.git
4. ≫ cd /code/remus_ws
5. catkin_make -DCATKIN_WHITELIST_PACKAGES=”ros_msg_converter”

7 Conclusion

The report should include a full description of your approach and results. Why did you select the motion planning approach and what parameters were selected to improve performance? Make sure to save your run with a rosbag file. I’ll discuss how to do this in class. Show the total path through completion by saving the bag file and include it as part of your report.

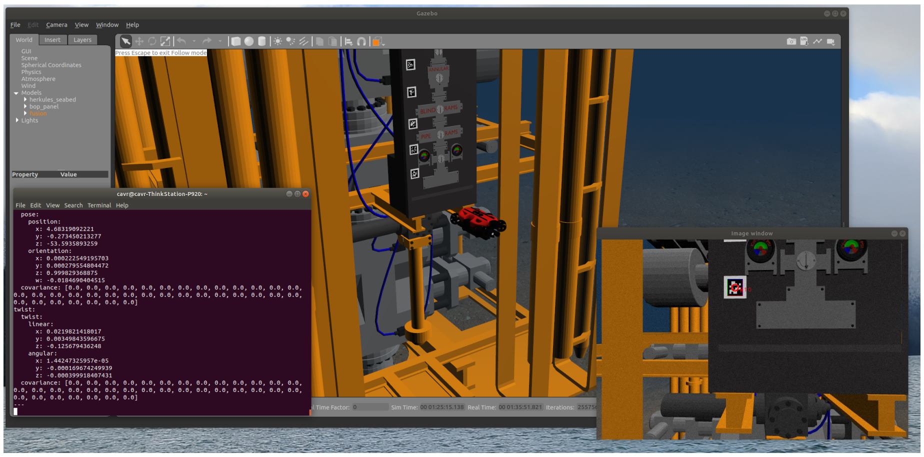

Figure 6: Third objective target 10.

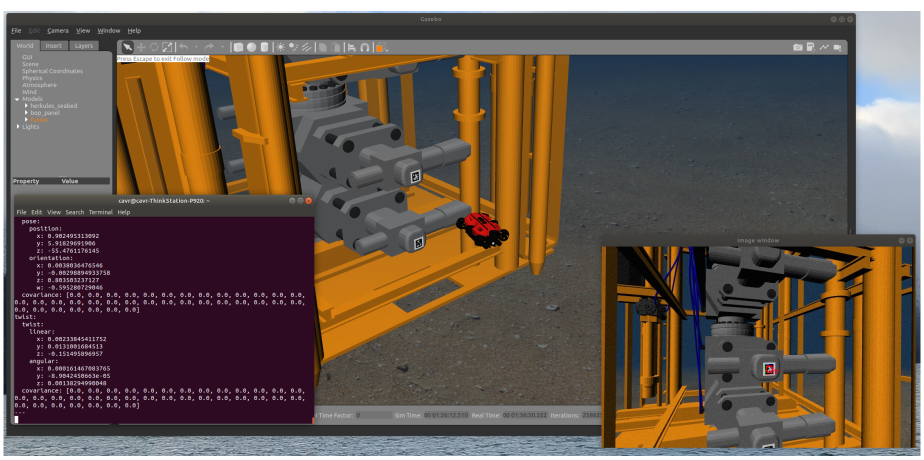

Figure 7: Fourth objective target 1.

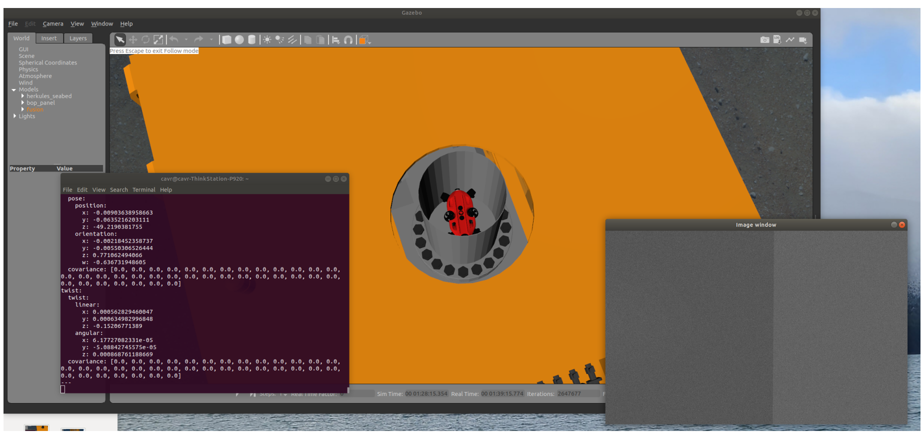

Figure 8: Final objective.