Introduction to Unmanned Ground Vehicles

Instructor: Micho Radovnikovich
Email: mtradovn@oakland.edu
ECE 495/595 – Winter 2017

Autonomous GPS Waypoint Navigation Project

Due: Sunday, March 19th

Objectives

The purpose of this project is to provide a hands-on simulation project to solidify the theoretical and ROS programming concepts discussed so far in the course. The main task is to write ROS nodes that will guide Audibot to follow a series of GPS waypoints, which are shown in Figure 1.

Specifically, the objectives are as follows:

1. Navigate to each waypoint in order.

2. The vehicle must pass within a 1 meter radius of each waypoint for it to count.

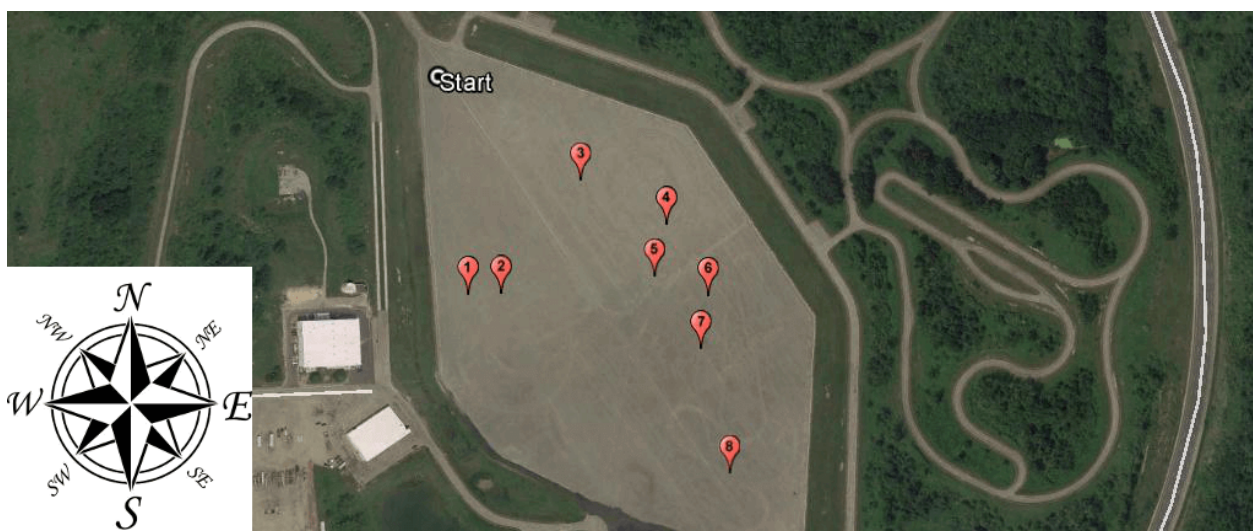3. After reaching the final waypoint, the vehicle should stop.

Figure 1: Google Earth view of the waypoints

GPS Waypoint Coordinates

The coordinates of each of the waypoints are shown in Table 1.

Table 1: GPS Waypoint Coordinates

Waypoint	Latitude	Longitude
1	42° 51′ 4.89″ N	83° 4′ 10.15″ W
2	42° 51′ 4.91″ N	83° 4′ 8.59″ W
3	42° 51′ 8.79″ N	83° 4′ 4.85″ W
4	42° 51′ 7.28″ N	83° 4′ 0.80″ W
5	42° 51′ 5.49″ N	83° 4′ 1.36″ W
6	42° 51′ 4.84″ N	83° 3′ 58.84″ W
7	42° 51′ 3.01″ N	83° 3′ 59.18″ W
8	42° 50′ 58.72″ N	83° 3′ 57.82″ W

Simulated Vehicle Interface

Your code can interact with the Audibot simulation using the following topics and parameters.

Subscribed Topics

Audibot subscribes to three different control signal topics, shown in Table 2.

Table 2: Audibot subscribed topics

Topic Name	Message Type	Description
/audibot/throttle_cmd	std_msgs/Float64	Throttle percentage command
/audibot/brake_cmd	std_msgs/Float64	Brake torque command in Newton-meters
/audibot/steering_cmd	std_msgs/Float64	Steering wheel angle command in radians

Published Topics

The topics in Table 3 are some potentially useful ones being published by the simulated vehicle. The heading is reported as it would come from a typical GPS receiver. This means that the heading is represented in degrees, ranging from 0 to 360, where 0 degrees is North, and increases clockwise.

Table 3: Simulated vehicle published topics

Topic Name	Message Type	Description
/audibot/gps/fix	sensor_msgs/NavSatFix	Vehicle's location in geodetic coordinates (50 Hz)
/audibot/gps/heading	std_msgs/Float64	Vehicle's heading in GPS frame (50 Hz)
/audibot/twist	geometry_msgs/TwistStamped	Vehicle's speed and yaw rate (50 Hz)

GPS Reference Point Coordinates

The origin of the fixed reference frame used by the Gazebo simulation is defined relative to a particular GPS coordinate. The latitude and longitude of this reference point are available on the **/audibot/gps/ref_lat** and **/audibot/gps/ref_lon** parameters.

Requirements

In addition to fulfilling the objectives listed on the first page, your system needs to also do the following:

> **Waypoint markers**

Publish markers to indicate the positions of each waypoint. The markers should be thin disks with a diameter of 2 meters. Configure Rviz to display the markers. It is recommended to publish the markers using a **visualization_msgs/MarkerArray** message.

> **Traversed path in Rviz**

Publish a **nav_msgs/Path** message that contains all the points the vehicle has traversed in the past. Configure Rviz to display the path.

> **Save Rviz configuration**

After configuring Rviz to show the path and waypoint markers, be sure to save the configuration so you don't have to add them every time you run the simulation.

> **Launch entire system from sim_project.launch**

There will be a launch file provided in the **gps_sim_project** package called **sim _ project.launch**. The Gazebo simulator and the benchmark test script will already be in the launch file. Add whichever nodes you write to this launch file.
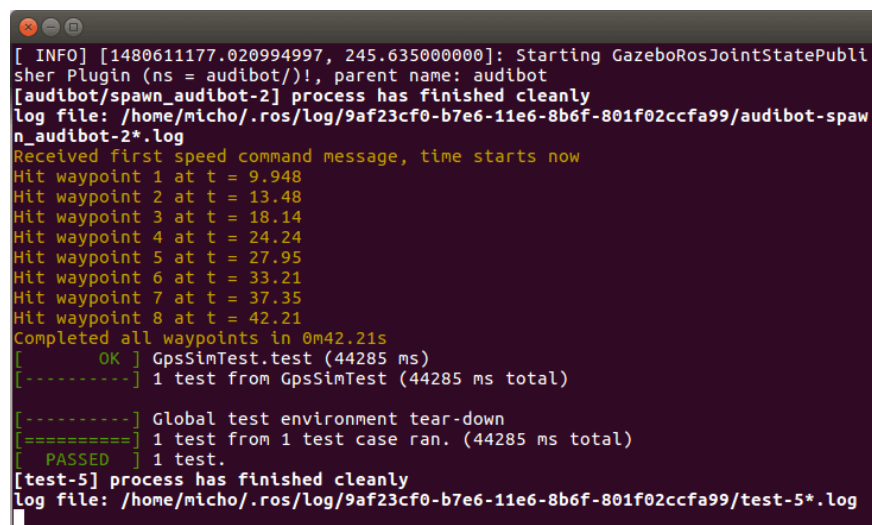
> **Benchmark testing**

Your system must pass the benchmark test (see next section).

Benchmark Testing

A test script will be provided to evaluate the performance of the GPS navigation system. The test script will check the following:

> Verify that the vehicle comes within a 1 meter radius of each waypoint in order.

> Time the entire run, starting from when the simulation is finished loading until the final waypoint is reached. The script will report the current time at each waypoint.

If the test passes, the terminal window running the system should look something like Figure 2.


```
[ INFO] [1480611177.020994997, 245.635000000]: Starting GazeboRosJointStatePubli
sher Plugin (ns = audibot/)!, parent name: audibot
[audibot/spawn_audibot-2] process has finished cleanly
log file: /home/micho/.ros/log/9af23cf0-b7e6-11e6-8b6f-801f02ccfa99/audibot-spaw
n_audibot-2*.log
Received first speed command message, time starts now
Hit waypoint 1 at t = 9.948
Hit waypoint 2 at t = 13.48
Hit waypoint 3 at t = 18.14
Hit waypoint 4 at t = 24.24
Hit waypoint 5 at t = 27.95
Hit waypoint 6 at t = 33.21
Hit waypoint 7 at t = 37.35
Hit waypoint 8 at t = 42.21
Completed all waypoints in 0m42.21s
[       OK ] GpsSimTest.test (44285 ms)
[----------] 1 test from GpsSimTest (44285 ms total)

[----------] Global test environment tear-down
[==========] 1 test from 1 test case ran. (44285 ms total)
[  PASSED  ] 1 test.
[test-5] process has finished cleanly
log file: /home/micho/.ros/log/9af23cf0-b7e6-11e6-8b6f-801f02ccfa99/test-5*.log
```

Figure 2: Test script successfully run.

What to Submit

Here are the deliverables for the project:

1. **Test your system against the benchmark test script**

   The provided test script is the same one that will be used to grade the project, so be sure that the test runs successfully before turning in the project.

2. **Commit and push all your source files to your repository on BitBucket**

   This includes all C++ source files, cfg, srv, YAML and launch files that make the system work. Additionally, push an Rviz configuration file that is loaded upon running **sim_project.launch**.

3. **A short write-up on how your system works**

   A full, formal report isn't necessary, but please write a short report documenting your approach and how the code works. Be sure to include a screen shot of a **rqt_graph** of the system. You can email me the report or push it into your repository; either way works.

Grading Rubric

The grade for the project will be derived according to the following criteria:

Table 4: Project grading criteria

Percentage	Criterion
40%	Benchmark test passes
10%	Waypoints completed in less than 1 minute
10%	Waypoints completed in less than 50 seconds
5%	Traversed path displayed in Rviz
5%	Waypoint markers displayed in Rviz
25%	Report sufficiently describes system
5%	**rqt_graph** screen shot in report
bonus 10%	Waypoints completed in less than 43 seconds