CSCE 496/896: Robotics: UAS, Fall 2020
Homework 2

Started: Monday, Sept 14
Video Submission 1: Sunday, Sept 20
Video Submission 2: Friday, Sept 25
Video Submission 3: Friday, Oct 2
Due: Sunday, Oct 4

Instructions: This homework is an individual assignment, collaboration is not allowed. If you discuss any problems with others, please note this on the assignment as described in the syllabus. Also note any materials outside of lecture notes, course textbooks, and datasheets that you used. Show your work and describe your reasoning to get partial credit if your solution is incorrect.

You should also make sure that you properly label and **describe** any figures or plots that you include in your writeup. You will not receive full credit if you do not explain these. In addition, you should refer to your code where needed to answer the questions (e.g. say "See file mynode/launch/test.launch for this problem, which ...").

You must turn in a pdf of your assignment on Canvas. Make sure to answer questions in complete sentences and explain answers as needed. **You must also turn in your code for all parts of this problem on Canvas.** Failure to electronically turn in your code will result in a 10 point penalty on this assignment. Points may also be deducted for coding errors, poor style, or poor commenting.

If you use any code from an online or other source you must cite the source in the comments. Otherwise it is considered plagiarism, which we check for!!

Note Regarding Video Submissions: There are a number of video submissions associated with this assignment. You must submit a short video to the associated video assignment on Canvas showing your system in action by the specified video submission date to receive credit for this. Using your phone to record a video of your computer with you narrating is sufficient for this.

Overall Homework Description: In this homework you will start by creating a ball in Gazebo and get it to roll around by applying forces. You will then implement a PID controller that will allow your drone to follow the ball when it *knows* exactly where the ball is. The next step will be to configure the camera on the drone in simulation, find the ball using the camera, and program the drone to follow the ball around using only information from the camera.

Name:

Problem 1. *(5 pts)*[1] *(To be completed at end of assignment) Approximately how much time did the total assignment take? Which sub-problem took longest and how much time did it take? Are there any questions that need clarification?*

[1]Each HW counts equally in your overall grade, even if homeworks have different point totals. This one is out of 115 points for 496 and 115+20 points for 896.

Problem 2. *Creating Objects in Gazebo*

a). *(5 pts) Start by reading the tutorial on how to build worlds in Gazebo: `http://gazebosim.org/tutorials?tut=build_world`. Manually create a ball in the world. Describe the steps and include an image of your ball in the world.*

b). *(5 pts) Now we want to apply forces to the ball to move it around. Read `http://gazebosim.org/tutorials?tut=apply_force_torque&cat=tools_utilities` and figure out how to apply a force to cause the ball to lift off the ground. Describe the steps. What is the minimum amount of force needed to lift the ball of the ground? Why is this the minimum amount of force?*

c). *(5 pts) Now we want to programmatically move the ball. You can do this from the command line with the following command:*

```
rosservice call /gazebo/apply_body_wrench "body_name: 'unit_sphere::link'
reference_point: {x: 0.0, y: 0.0, z: 0.0}
wrench:
  force: {x: 100.0, y: 0.0, z: 0.0}
  torque: {x: 0.0, y: 0.0, z: 0.0}
duration: {secs: 10, nsecs: 0}"
```

Now write a short ROS program[2] to, you guessed it, have the ball trace your initials (or at least approximately). You will most likely need to apply equal and opposite forces to get it to stop between letters. Describe your approach and make sure to include this basic code in your submission.

Video Submission: (10 pts) *This is due by Sunday, Sept 20. You must submit a video showing your ball moving around to write your initials. Narrate the video to say what you are doing and describe your code as well.*

d). *(5 pts) Now investigate the ROS messages that are available and figure out how to determine the position of the ball. What topic(s) do you need to look at to find the position of the ball?*

[2] Refer to the "Writing a Simple Service" ROS tutorial if you do not recall how to use services.

e). *(5 pts) Now write a small ROS program that subscribes to the messages, extracts the position of the ball and republishes the position on a topic called* `/ballPosition` *with the ROS message type of* `geometry_msgs/Pose`*. Describe your approach and make sure to include this basic code in your submission.*

f). *(5 pts) Use rqt_plot to show the position of the ball as you write your initials. (Hint: you can also use rosbag to record and replay so that you have time to adjust the plot.)*

g). *(5 pts) Read this tutorial:* *http://gazebosim.org/tutorials?tut=color_model* *on changing the colors of objects in Gazebo. Set the color based on the first letter of your last name:*

- *Red for A-G*
- *Blue for H-M*
- *Green for N-Z*

Describe how you achieved this.

h). *(5 pts) Create a new launch file based on* `mavros_posix_sitl.launch` *that automatically loads your colored ball in ROS. Include the key elements of your launch file in the report.*

Problem 3. *Getting Images from Gazebo*

a). *(5 pts) Follow the tutorial in the* `sim_camera_guide.pdf` *file posted on the website to get the camera running on your drone in Gazebo. Describe how you set this up and include a figure showing the camera view from* `rqt_image_view`*.*

b). *(5 pts) Now figure out how to orient the camera so that it is facing straight down. Make sure to create copies of any files you modify when you do this so that you can always revert to the default configuration. Describe how you did this and show the results.*

Video Submission: (10 pts) *Rerun your initial writing code from problem 2c) with the drone positioned above the ball as it moves around. Record a video showing the view from the drone camera as the ball moves around to write your initials. This is due by Friday, Sept 25. Narrate the video to say what you are doing and describe your code as well.*

Problem 4. *Tracking with Ball Detector*

In the code download for this lab, there is code to detect balls (called `ballDetector`*). In the* `launch` *directory, there are two launch files. The first,* `ballDetector.launch`*, launches the ball detector code. The second,* `displayDebugImages.launch`*, will display three different images you can use for debugging. If you find that the ball detector slows down your system, you can disable the debug images and information by commenting out the* `#define BALLDETECTOR_DEBUG` *line in the* `ballDetector.cpp` *file.*

The ball detector works by first converting the image to HSV color space. It then performs thresholding to filter out all pixels that are between a low and high HSV threshold. These values can be configured in the launch file (see `ballDetector.launch` *for an example) and can be dynamically changed by running the command* `rosrun ball_detector configGUI.py`*. The ball detector then searches for the largest group of connected pixels that have similar height and width (as a crude approximation to find circular groups of pixels). The largest group that meets this criteria is selected as the "ball" in the image and a message is sent out with this ball location.*

The debug images display the HSV image, thresholded image (white pixels indicate those that are between the low and high thresholds), and a marked up image with white pixels indicating the boundaries of connected groups. In addition, in this image the final ball location is marked with a circle. Use these images and the configGUI to set good HSV low and high thresholds.

a). *(5 pts) Without modifying the ball detector code or the gazebo/mavros code, configure the system so that the ball detector consumes the images produced by the drone. You will need to use the "remap" command in the launch file to achieve this. Make sure to create a new launch file for this problem so that you can submit copies of all your other launch files for the other problems. Describe how you accomplished this. (Note: Launch files can also call other launch files to simplify things.)*

b). *(5 pts) Now tune the ball detector to identify the ball as viewed from the drone's downward facing camera. Describe how you did this and the final HSV parameters used.*

c). *(10 pts) Next, write code to control the position of the drone so that it hovers over the ball when it is stationary. Start with the drone offset from the ball location so that it has to move to center itself over the drone. Describe your approach and include figures showing how well this works.*

Note 1: You cannot use the `ballPosition` topic or any other topic that gives you the actual position. *You must use the camera to find the ball. However, you can assume that the ball is visible to the camera when it starts, but you need to show that it can move over it no mater where the ball starts in the frame.*

Note 2: If your computer cannot keep up with the processing, you can do a few things. You can run Gazebo at slower than realtime so that it has more time to process. You can also turn down the update rate for the physics simulation in gazebo or possibly disable the physics modeling all together. You might also try allocating more resources to the virtual machine. If you are still having trouble, please talk to the instructor.

Video Submission: (10 pts) *This is due by Friday, Oct 2. You must submit a video showing your drone hovering over the ball using the camera to control the position. Narrate the video to say what you are doing and describe your code as well.*

d). *(10 pts) Now make the ball move around and have the drone follow the ball based on the image processing. Describe how you achieved this and if you had to do anything different than you did for the stationary ball. How fast can you move the ball and still achieve good tracking? Make sure to include description and figures to support your approach.*

e). *(10 pts)* **839 Only:** *Repeat problem c) (moving over a stationary ball), but this time create a search pattern so that it can find the ball when it starts outside of the initial frame of the camera. Make sure to describe your approach and include figures showing how well this works.*

f). *(10 pts)* **839 Only:** *Repeat problem d) (ball following), but with the drone camera tilted up 30 degrees. Describe the results and how they differ from when the camera is looking straight down.*

Do not forget to fill in the amount of time you spent on this assignment in Question 1.