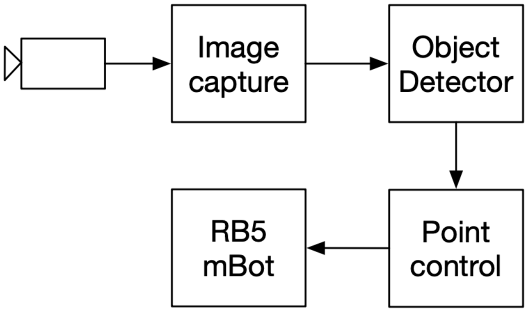
Homework 2 - CSE 276A - Intro to Robotics

Due: Tuesday, 31 October 2024, before midnight

The objective of this exercise is to use the on-board camera to drive to specific locations in your environment. Using feedback from the detected landmarks, you should be able to improve your localization performance and through this drive more effectively. The general structure of the anticipated system is shown below.



In HW1, you have already built part of the structure, an open-loop point controller that controls the RB5 mBot. You will need to turn the open-loop controller into a close-loop controller by incorporating visual feedback. The visual feedback comes from known landmark positions and your estimate of the relative pose of your vehicle regarding these landmarks.

You have **two options**:

1. Use artificial landmarks such as AprilTags (https://github.com/AprilRobotics/apriltag).
2. Use a custom recognition package to learn your own set of landmarks, such as TV monitors and coke-cans. You will get 20% extra credit with this option, but **fair warning it is significant work**.

Here are some hints and links that may help you with this assignment:

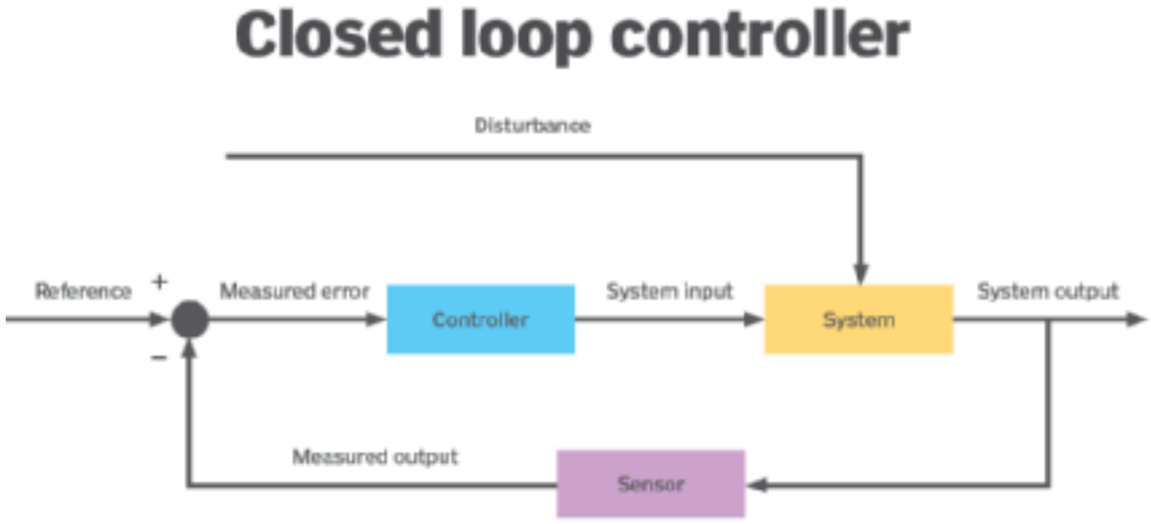
1. As the RB5 cameras have optical distortion, it is important to calibrate your camera before you do processing.
2. If you are using ROS, a good guide to calibration / camera usage is available from here. Otherwise the OpenCV camera calibration guide can be helpful: link. You can then setup the objects at ”calibrated distances” and use them for servoing / navigation.
3. The AprilTag ROS package has a tutorial on how to use the nodes for estimating the TF between your position and tags. Here is the link to the AVL site about use of AprilTags on the RB5. Note that you do not need to install the packages mentioned in this post since we have installed them for you.

The desired car motion is moving through 4 way points. $(0, 0, 0) \rightarrow (1, 0, 0) \rightarrow (1, 2, \pi) \rightarrow (0, 0, 0)$. During recording the video, you must have some markers (such as a tiny tag) for each way point on the ground, so we can tell how accurately the robot actually moves. In the environment, you will need to set up some landmarks such as AprilTag markers or coke-cans based on your selection of landmarks. Thus, the robot can estimate its position based on them. During the experiment, you must record the localization error at each waypoint.

As part of your solution, provide a zipped folder (hw2 your name) with the following:

1. A report (3 pages max) with the following information.

   (a) The description of your closed-loop controller. You should draw a similar graph like below, and give a short description of each node and the information being passed on the edges.



   (b) The description of your landmark. You need to talk about where you place landmarks, and give a short reason why you position them that way.

   (c) A description about how to estimate the car pose based on the landmarks and how to handle the noise in the estimate of landmark poses. You should motivate your choice of a solution.

   (d) During motion, in some short time, the camera may not detect any landmark. You must talk about how you handle the case when there is no landmark in the camera field of view or you can't detect any landmark.

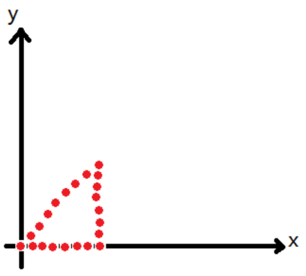
   (e) If you select the second option for landmark detection, you need to provide the **motivation** and **details of the detection algorithm**. Besides that, you also need to describe the robustness and the accuracy of your method.

   (f) A short analysis of how smooth your car moves.

   (g) The total moving distance in units you select. For example, if the distance between the first landmark and second landmark is 0.5m, then your unit is 0.5m (**The unit should not be less than 0.5m**). By the way, the total moving distance (include rotation) is part of your performance.

   (h) The location error(both rotation and translation) on each waypoint.

   (i) A 2D trajectory of the car. That is, you need to draw a dot into an image for each 0.5 second like the following. (or 1 second based on your speed.) Those positions should be estimated by your algorithm instead of your eyes.



    (j) Your reflection on the overall performance.

2. A short video that demonstrates the performance of your system. Upload the video to YouTube as an unlisted video (but not private) and put the link in the document. Describe in the report your sensor coordinate, vehicle coordinate, world coordinate, and the start pose for the robot.

3. Code: all the related ROS packages in the src folder of the ROS workspace.