CSE 397/498-013
Introduction to Mobile Robotics

Homework: Wall Following & Feedback Control
Report Due Date: Tuesday, 8 Nov 05 submitted via Blackboard
PRIOR TO THE START OF CLASS (**NO ANALOG SUBMISSIONS**)

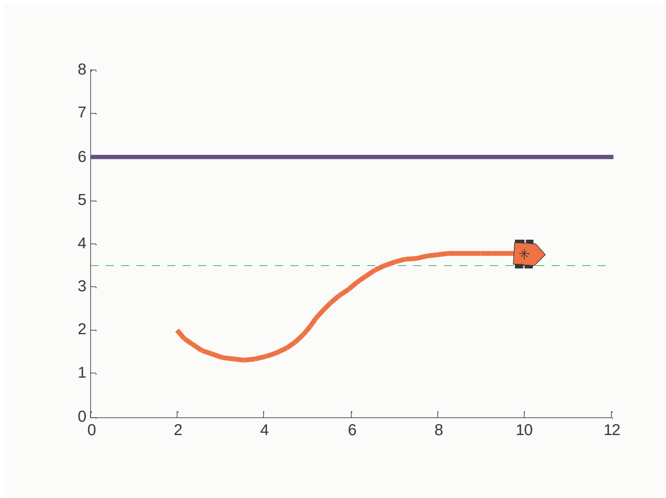

Figure 1: WLS Wallfollower trajectory under PD controller

A. Objectives:

1. Implement least-squares and robust wallfollowers.
2. Experiments in robot motion control

B. Robot Model:

1. Again, we are using a differential drive kinematic model.
2. We will assume a similar error model to the last assignment. However, the outlier rate will be 25% (vice 20%) and the standard deviation of the Gaussian noise will be modeled as $\sigma=0.1*\rho$ where ρ is the normal distance to the wall.
3. Assume a constant robot velocity of 0.5 m/s
4. Assume a MAXIMUM angular velocity of 0.25 rad/s
5. The sensor update rate is 2Hz
6. The maximum sensor range is 6 meters.
7. Assume a left-handed wallfollower with a desired standoff distance from the wall (blue line above) of 2.5 meters (dashed green line).

C. Requirements:

1. You MUST use Matlab to complete this assignment.
2. This is an individual assignment. Each student is required to submit his/her own work in order to receive credit.
3. As stated on the web page, if you turn this assignment in late without coordinating with me first you will receive a 0.

D. The Assignment:

1. **Least-Squares & RANSAC Wallfollowers:**
    a. You are to implement a least-squares wallfollower `LS_WallFollower(kP,kV)` based upon the `WLS_Line` function from the previous homework assignment. *kP* and *kV* correspond to the proportional and derivative gains for your controller. This will control the relative position and orientation of your robot with respect to the wall described in Figure 1.
    b. To regulate the position and orientation of the robot, you will implement the PD controller discussed in class (see lecture 10). You will need to choose appropriate gains for the controller design.
    c. You will first estimate the relative position and orientation of the robot with respect to the wall. These inputs will be fed to the PD Controller which in turn will generate an output (ω) that will be used to control the orientation of the robot.
    d. Updates to robot position and orientation occur at 2Hz.
    e. The initial robot position will be taken as input from the user mouse. The initial orientation will be generated at random from $\theta \in \left[-\pi/4, \pi/4\right]$
    f. Repeat steps 1.a-1.e by implementing a RANSAC_WallFollower(`kP,kV`) that will filter outliers from the sonar measurements. You can use the same criteria for outlier rejection as in the previous assignment (DON'T FORGET TO TAKE INTO ACCOUNT THE NEW SENSOR MODEL!).
    g. When you have determined that both implementations are performing properly, save images from sample trial runs with identical initial configurations for both wallfollowers (Figures 1-2).

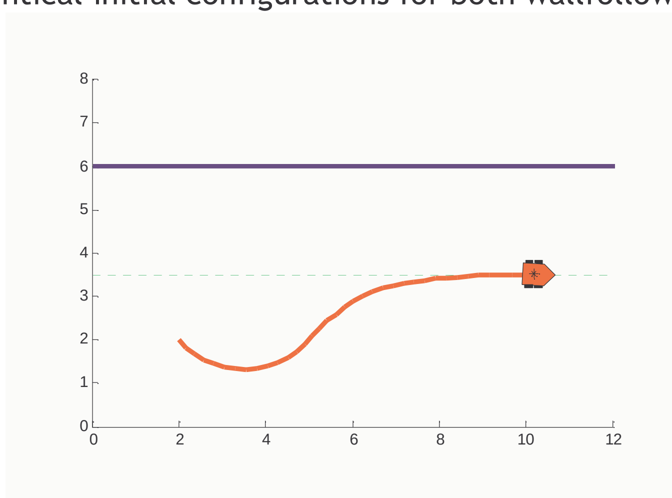

Figure 2: RANSAC Wallfollower with PD controller.

    h. Discuss differences in the performance of the two approaches, to include convergence properties, settling time, overshoot, etc.

2. **Controller Design:**
    a. In this exercise, we will attempt to optimize controller performance under ideal conditions. This means:
        i. No outliers.
        ii. No sensor noise.
    b. You are to perform a Monte Carlo simulation on the initial robot configurations to estimate optimal controller gains for the wallfollowing task.
    c. The initial robot position will be (2,y,θ) where $y \in [1,5]$ and $\theta \in \left[-\pi/4, \pi/4\right]$
    d. Discretize *y* and *θ* by 0.1 meters and *π*/64 radians, respectively.
    e. Choose the gains so that the PD controller is critically damped.
    f. For each gain set starting at kP=0.1, conduct a trial across each possible initial pose as discretized in 2.d.
        i. For each pose, record both the settling time and the overshoot for the controller.
        ii. For each gain set, record the mean settling time and overshoot over all initial poses (you may assume that the controller will settle by *x>=15*).
    g. Increase kP by 0.1 and repeat 2.d-f above.
    h. Iterate to obtain the controller gain set with minimum settling time.
    i. Plot the results of gain vs. settling time and include this in your report. What pair of gains did you determine was optimal for your wallfollower?

    NOTE: There are 41x33 points, and will need to test on the order of 10 gain sets. That comes to something like 13,530 trials. Unless you want to wait all day for this to run, TURN OFF ALL THE PLOT COMMANDS IN YOUR MAKE/MOVE ROBOT FUNCTIONS, ETC.

E. Additional Questions:

1. Continue to increase the initial error in position and orientation for the robot beyond those covered for the design trials. What do you observe about the controller performance? Are there configurations where the controller will fail to converge to the proper distance from the wall? If so, explain why these occurs.
2. What happens if you set *kP*=0 for the controller? What about *kV*=0? Explain why.
3. Increase the maximum angular velocity of the robot from 0.25 to 0.5 rad/s. How does this affect controller performance.

F. Turn in: A write-up, to include images from all simulation trials, answers to all questions, as well as your Matlab source code.