

CSCE 439/839: Robotics Homework 2

Instructor: Carrick Detweiler
carrick_at_cse.unl.edu
University of Nebraska-Lincoln
Fall 2015

Started: Friday, October 16, 2015
Due: Weds, November 4th, 2015 start of class

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. Answer all questions with **complete sentences** and describe your reasoning where appropriate for full credit. This homework is due on the date listed above before the start of class.

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, while you do not need to turn in a printout of your code for this assignment (instead you should submit via handin), 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 bring a printout (typed, not hand written) of your assignment class on the day it is due! 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 and a PDF version of your report by visiting <http://cse.unl.edu/~cse439/handin/>.** Failing to electronically turn in your code (and PDF) will result in a 10 point penalty on this assignment. Points may also be deducted for coding errors, poor style, or poor commenting.

Name:

Problem 1 (5pts¹). *(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?*

Problem 2. *ROS: In this problem, you will create a ROS node that will compute statistics for the Huskers' Football team. For these questions, refer to the hw2 code (`rawStats.py` and `offensivePlay.msg`).*

a) (5pts). *Describe how you create a ros package for this node and where you place the files. What topics does this node publish on?*

b) (5pts). *Create a new node, called `offensiveStats` (C++ or Python) that subscribes to messages of type `offensivePlay`. Write the code in this node to compute the pass completion percentage (number completed passes divided by total number of passes attempted). Describe the steps required to create the node and subscribe to the message.*

¹Each HW counts equally in your overall grade, even if homeworks have different point totals.

c) (5pts). Create a new message type that contains the pass completion percentage as well as the total passing and rushing (non-passing yards gained) yards for this game. Describe the process for creating a new message. How do you ensure that this message (and others) are compiled when you run `catkin_make`?

d) (5pts). Write the code to create a publisher, populate, and send the above message. Make sure to comment your code.

e) (5pts). What command would you use to plot the total yards completed? Give the command, the proper arguments, and include the plot.

f) (5pts). Without changing the `rawStats` node, launch a new pair of nodes that publish raw stats on the topic `\plays\opponentOffensivePlay` and also generates the offensive stats for the opponent. Hint: look into “`remap`” in the ros documentation.

g) (5pts). Create a launch file that will launch all four nodes.

Problem 3. Sensing

a) (5pts). What is the difference between precision and accuracy? Which is more important for the gyroscope on the hovercraft? Why?

b) (5pts). Describe how you could obtain dead-reckoned position information from an accelerometer. What would the largest position error be after moving 100m if the accelerometer has a 3% error.

c) (5pts). **839 Only:** The gyro on the fanboat tends to drift. What additional sensor(s) could you add to the hovercraft to correct for this drift. How would this work?

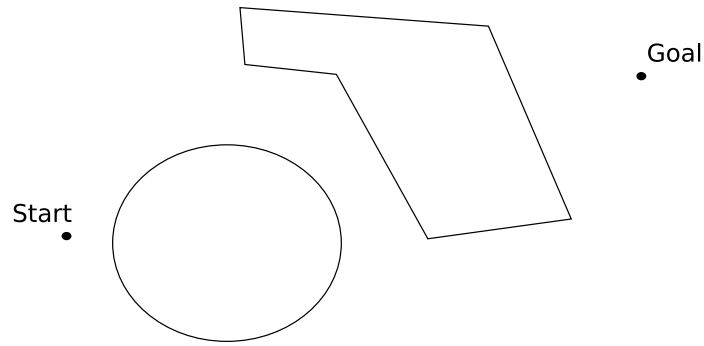


Figure 1: Bug algorithms

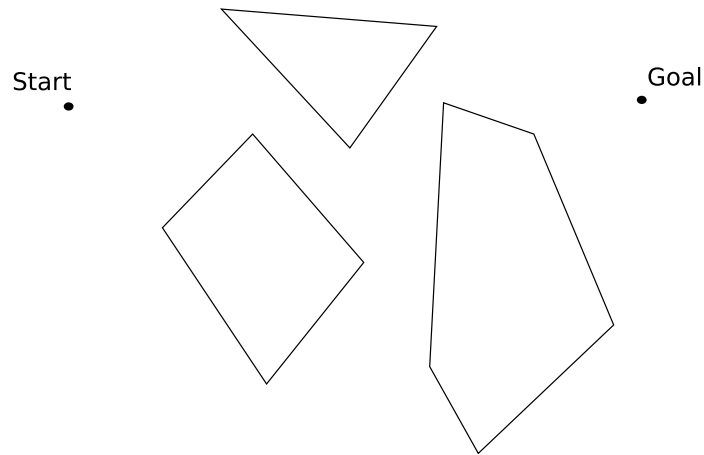


Figure 2: Visibility Graph

Problem 4. Path Planning

a) (5pts). In Figure 1, draw the path the Bug1 algorithm would take. Explain the algorithm.

b) (5pts). Draw the visibility graph for Figure 2 and indicate the path that the robot follow from the start to the goal. Explain.

c) (5pts). In Figure 1, draw the path Tangent Bug would take (in a different color/style line or on a separate plot). Explain the algorithm.

d) (5pts). **839 Only:** In Figure 1, one of the Bug algorithms outperformed the other. Draw a different figure where the performance is reversed. Explain what happens.

Problem 5. PID

- a) (5pts). Write the equations for a PID controller and explain the impact of each of the P, I, and D terms.
- b) (5pts). In a controller running at 10Hz with $P=10$, $I=0.1$, and $D=4$ (without any loop time compensation), what would you set the parameters to if the loop rate was increased to 20Hz? Why?
- c) (5pts). **839 Only:** List and explain 2 different problems that can occur with PID controllers and how you can mitigate these problems in a PID controller.

Problem 6. Kalman Filter: In this problem you should use the `rocketShip.py` code. Your goal is to develop a Kalman filter that will estimate the position, velocity, and acceleration of the rocket at a given time. You are given a control input in terms of acceleration on the topic `control/acceleration` and sensed values for the acceleration and velocity on `sensors/acceleration` and `sensors/velocity`, respectively. For your debugging purposes, you also have ground truth information, but you should not incorporate this into your filter design.

- a) (5pts). Write the linear equations that define the process and measurement in the matrix notation used in class.
- b) (5pts). Write the Kalman Equations (process and measurement update equations) for this system using the actual numbers and variables in the matrices (e.g. do not just say $x = A\dots$, write the vector elements of x).
- c) (5pts). Implement your Kalman filter to estimate the position, velocity, and acceleration of the rocket. Discuss your implementation and challenges you faced. You can use C++ or python and any matrix libraries (although this isn't required), but you should write your own Kalman code and not use a Kalman library.
- d) (5pts). How did you tune the parameters of your Kalman filter (R and Q)? Discuss the impact of changing the values of these parameters and support it with plots showing different responses.
- e) (5pts). Approximately how long can you stay within 50 units of the ground truth position with your estimate? Include data from multiple runs to support your answer.
- f) (5pts). **839 Only:** If you do not use the control input in your equations how much better/worse does it perform? Discuss why this is the case and support this with data.

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