

CSCE 439/839: Robotics: Algorithms and Applications, Fall 2015

Homework 1

Started: Wednesday, August 26, 2015
Due: Beginning of class Friday, September 11, 2015

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, 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 and turn in a pdf using handin! 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.

Note regarding checkoffs: There are a number of checkoffs associated with this assignment, you can wait and get them all checked off at once or as you complete them.

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 (10pts). *Installing ROS². You should install **ROS Indigo** (Indigo is one of the releases). I would either recommend installing Ubuntu 14.04.2 natively on your computer and then install ROS or you can do all of this inside of a virtual machine such as VirtualBox (open source) or VMWare (free to CoE students or the free vmware-player can be used). VMWare is slightly preferred as VirtualBox occasionally has had bugs that make it difficult to use with the radios we will use in this class. **Make sure that you install Indigo as I will assume you are using this version of ROS throughout this assignment and course.***

Checkoff: Bring your computer³ to office hours and have the TA or instructor sign below to indicate that you have completed this step.

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

²You should explore <http://www.ros.org> as there are many more helpful hints and documentation that are not referenced in this homework.

³If you do not have a laptop, please talk to me now.

Problem 3. *Turtle Sim.* Complete section 1.1 the beginner tutorials found at <http://www.ros.org/wiki/ROS/Tutorials>. This goes through a “turtle” tutorial that you will need to use to answer the below questions. **Note:** Some of the links may refer to old tools, so make sure to follow the links that point you towards the Indigo documentation. In general, if you see a tool that starts with `rx` you should instead use the same version of the tool that starts with `rqt_`. For instance, where it refers to `rxconsole` use `rqt_console` instead. Also, we will be using `catkin`, not `roscpp`.

a) (5pts). Use `rostopic pub` from the command line to write “CSE” with your turtle. Attach a picture showing your final “CSE” as drawn in TurtleSim.

b) (5pts). Give all the commands you used to write CSE in TurtleSim.

c) (5pts). Show the `rqt_plot` of the x and y position, speed, and angle (so four lines) of your turtle as it writes CSE. Augment the plot to show where drawing of each of the letters occurred.

Problem 4. *Creating New ROS Nodes*

a) (10pts). Now create a new ROS node (see <http://www.ros.org/wiki/ROS/Tutorials> for details, you can use either C++ or Python) that publishes messages to write CSE (similar to the manual `rostopic pub` from problem 3a). Include a picture of the results. Also, describe how you dealt the timing between sending messages. Remember that you need to comment the code appropriately for full credit (and turn it in using `handin`).

b) (5pts). Create a launch file that starts both the TurtleSim node and your new node. Include a copy of the launch file in your report.

c) (5pts). **839 Only:** Create **one** new node that writes CSE using three different turtles. Include a picture of the results.

d) (5pts). **839 Only:** Create three different nodes (or they could be one node instantiated three different times) that each control a different turtle to write CSE (one turtle per letter). Include a picture of the results.

Problem 5. *In the kit I gave you in class you have an Arduino with an attached Joystick. In this problem, you will program the Arduino to read the joystick and then send this data to a ROS node you will create to ultimately drive your turtle with the joystick. Refer to <https://www.sparkfun.com/products/9760> for details on this joystick and the various links to libraries to read it.*

a) (5pts). *Program your Arduino to read the inputs of the joystick and buttons (you can refer/use some of the referenced libraries on the sparkfun page, but it is also fairly trivial to implement yourself). Make sure to include your code in `handin` and tell me here if you wrote your own code or if you used an existing library (include a link to it in your report if you used an existing library). You should print out the values of the joystick axes and buttons over the serial port in a human readable format that is **easier** to read than the sample joystick code (e.g. add annotations to say what each value is). Include a few lines of this output in your writeup and annotate it to indicate what is going on.*

Checkoff: Show the output of this code to the instructor.

Problem 6. *Creating a ROS interface to the Arduino.*

a) (10pts). *Create a binary protocol that includes at least a start byte (a unique byte at the start of each packet) and a checksum to transmit all of the joystick information from the Arduino. Note that by binary protocol, I mean that you should transmit the byte `0xCD` instead of the string `'205'`. You can use any type of checksum you would like (e.g. just the sum of the bytes, XOR, etc). **Describe** your protocol in this report.*

b) (5pts). *Create a ROS node to process the data sent from the Arduino. You can use the `read_serial.py` python script (on the course website). This creates a node that publishes each byte it receives on a topic called `\rx`. You can subscribe to this topic to get the data, but you should create a new node (in C++ or python) to do the processing of the data received on this topic. Make your joystick node compatible with the `ps3joy` ROS node (<http://www.ros.org/wiki/ps3joy/Tutorials/WritingTeleopNode>). By this I mean that your node should publish the same topics that the `ps3joy` node does.*

c) (5pts). *What is the fastest rate⁴ that you can transmit joystick and button readings from the Arduino to ROS? How did you figure out this rate?*

⁴You do not need to try to get the fastest baud rate or smallest packet size, mainly I'm interested in how fast you can send with your particular configuration before bytes start getting dropped, etc.

d) (5pts). *Figure out how to use the joystick to drive around the turtle in TurtleSim. You can do this by following a similar approach to that done in the ps3joy node. Include a picture drawn by you while controlling the turtle with your joystick. Be creative in the picture you draw.*

e) (5pts). *Now create a launch file to launch all the nodes needed to drive around the turtle with your Arduino joystick.*

f) (5pts). *Include a printout of the rqt_graph of your overall system. Remember, that all problems that ask for figures or launch files should also include a description to receive full credit.*

g) (10pts). **839 Only:** *Write additional ROS and Arduino code to enable sending a boolean value on the topic ArduinoLED that will cause the LED on the Arduino to turn on or off based on the boolean value. This will require modifying your ROS node, sending a serial command (modification of the read_serial.py code), and processing this command on the Arduino. Make sure to include a header and checksum on the binary packet you create (to allow for future expansion). Configure one of the buttons on your controller to turn on and off the LED.*

Checkoff: Show turning on and off the LED with the controller to the instructor.

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