Lab 6: Week 7

Due on 5/20/2020 at 5 PM.

This group lab assignment has two questions. Develop the following RobotC programs.

Read the instructions carefully.

- Provide solutions to written questions in a single PDF file. In the title page of this document, identify group members, their affiliations, and percentage-wise contribution.
- Use separate c files for programming questions with filenames in the form Qnm.c, where n is the question number and m, is the part number.
    - E.g., **Q1b.c** is the solution file for the Q1 part b programming question.
- **Zip** your solution files (including the PDF) in a file named Gn.zip where n is the group number and upload on D2L. Upload only one solution per group.
    - E.g., G3.zip is the solution by group 3.
- If you have any questions, contact the instructor.

Q1: You will implement the proportional controller to control the deceleration of the robot turning motion and test the turning on the turn utility found in the robot virtual world.

a) Design the state diagram for the program that will cause the robot to perform a point turn at an angle specified by the user at the start of the program.

b) Write the pseudocode for a controller Task that will implement the proportional controller. The controller will decelerate the robot to the desired angle.

The proportional controller is given by

$$desiredMotorPower = k(desiredAngle - currentAngle)$$

c) Implement the RobotC program and test on the turn utility with a 270-degree turn. Experiment with various slew rate values, proportional gain values, and maximum motor power values.

Note:

- Make sure that you use a reasonable proportional-gain value (k=2 is a fair initial value) with the controller task running at 50 Hz.
- An error of +/- 2 degrees is acceptable.

- Use the same or a higher speed for the main Task where you have the state machine implemented.

d) Compare the result of this algorithm against the previous implementation without the proportional controller in terms of accuracy and speed of turning.

Q2: You will implement a slew rate controller to the program developed in Q1. This will control the initial acceleration of the robot motion.

a) Write the pseudocode for a slew-rate controller to control the acceleration of the robot from its still position at the beginning of the program.

   The slew rate controller is given in Table 1. The variable `slewRate` is defined by the user and determines how the robot accelerates or decelerates. The `maximumMotorPower` variable allows the user to limit the maximum motor power to be delivered to the motor during the program.

b) Implement the slew rate controller within the controller Task. Then test on the turn utility with a 270-degree turn. Experiment with various slew rate values, proportional gain values, and maximum motor power values.

   Note:
   - Implement the slew rate controller after the proportional controller.
   - Make sure that you use a sufficiently high slew rate so that the slew-rate controller can keep up with the values computed by the proportional controller. In other words, the slope of the slew rate is greater than the slope of the proportional gain. (k=2 and slewRate=1 are reasonable initial values). Controller task running at 50 Hz.

c) Compare the result against Q1 in terms of accuracy and speed of turning.

Table 1: Pseudocode for slew-rate controller

```
if (desiredMotorPower > currentMotorPower){
	if (currentMotorPower + slewRate >= maximumMotorPower){
		currentMotorPower = maximumMotorPower;
	}
	else{
		currentMotorPower += slewRate;
	}
}
else{
	currentMotorPower=desiredMotorPower;
}
```