Spring 2016
Control of Mobile Robotics
CDA4621
Lab 2
Feed Back Loop Control
Total: 100 points
Due Date: 2-29-16 by 8am

A. **Lab Requirements**
The lab requires use of the course robotic hardware ("Robo-Bull-2016") provided to students at no charge for the duration of the course. Required software can be downloaded free of charge from the web. All labs are to be done by teams of two students. Each student is required to submit his or her joint report through CANVAS.

- Hardware Requirements
The "Robo-Bull-2016" (Figure 1) is the main robot hardware used for the course. Note, that the components of the robot are the same as in Lab1.

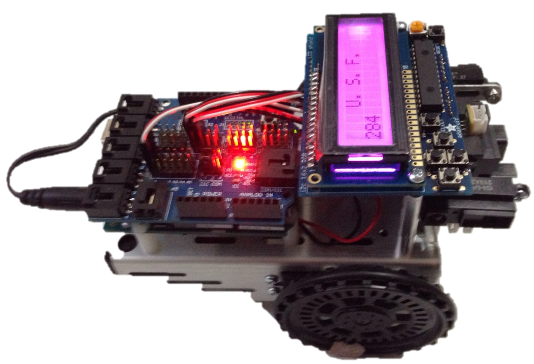

Figure 1: Robo-Bull-2016.

- Software Requirements
Arduino Software (Version 1.6.7 released on Jan 7, 2016)
https://www.arduino.cc/en/Main/Software

B. **Task Evaluation**
Each individual task is worth a specific number of points where these points are always split 50% between Task Execution and Task Report:

- Task Execution
The robot should execute the task correctly with a video clearly and completely showing the task execution (points will be taken for errors or missing aspects of task execution).

- Task Report
Each task report requires an accompanying document to be uploaded to Canvas together with ALL the files required to run the program in the robot. The task report needs to include ALL the following sections (points will be taken off if anything is missing):
1. Task description.
2. Solution describing the conceptual algorithm used to solve the task described in terms of flow charts to describe the logic of the program and block diagrams to describe the various robot components.
3. Video link to different task executions (you should split each task execution as a different video link most preferably in YouTube making sure the video is public to all). Provide at the beginning of each video your name and description of which task is being performed.
4. Description of the code used to program your robot with explanations that clearly relate to the solution previously described.
5. Images taken from the actual robot task execution (at least one image per task).
6. Conclusions where you analyze any issues you encountered when running the task and how these could be improved.

C. Task Description
The goal of this assignment is to perform close loop control using distance sensors and robot motor actuators. The lab consists of 3 tasks, each building on results from the previous task. Figure 2 presents a diagram for close loop control of robot velocity proportional to its distance to a goal $r(t)$, where the goal may be any object such as a wall or an obstacle. As the robot gets closer to the desired distance to the goal, the error $e(t)$ will modify the control signal $u(t)$ until such error becomes 0. In our case the control signal corresponds to robot motor velocity until it becomes 0 when the robot reaches a specific distance to the goal.

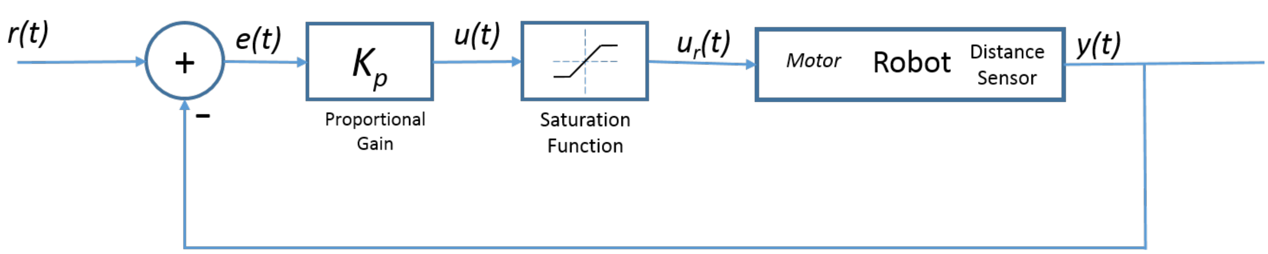

Figure 2: Close Loop Control of Robot Velocity Proportional to its Distance to a Goal.

Eq. 1-5 summarizes the diagram in Figure 1:

$$u(t) = K_p * e(t) \qquad \text{(Eq. 1)}$$
$$u_r(t) = f_{Sat}(\, u(t)) \qquad \text{(Eq. 2)}$$
$$u_r(t) = f_{Sat}(\, K_p * e(t)\,) \qquad \text{(Eq. 3)}$$
$$e(t) = r(t) - y(t) \qquad \text{(Eq. 4)}$$
$$u_r(t) = f_{Sat}(\, K_p \left(r(t) - y(t)\right)\,) \qquad \text{(Eq. 5)}$$

where:

$r(t) = desired\ distance\ to\ the\ goal$
$y(t) = distance\ from\ robot\ to\ the\ goal$
$e(t) = distance\ error$
$K_p = proportional\ gain\ or\ correction\ error\ gain$
$u(t) = control\ signal\ corresponding\ to\ robot\ velocity$
$f_{sat} = Saturation\ Function\ (See\ Figure\ 3)$
$u_r(t) = control\ signal\ corresponding\ to\ saturated\ robot\ velocity$
$u_r(t) \rightarrow$ RServo.write(u_r)

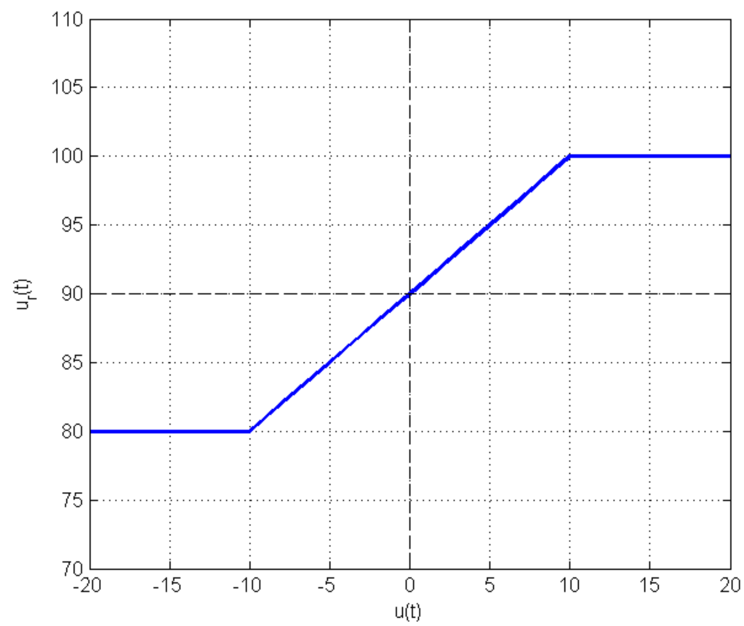

Figure 3: Saturation Function showing cutoff for max positive motor velocity at $u(t) = 100$, and min negative motor velocity at $u(t) = 80$, where $u(t) = 90$ corresponds to 0 velocity.

1. Wall Distance (20 points)

You need to program the above equations (1-5) in order to implement the close loop control program where the control signal velocity $u(t)$, corresponding to robot motor velocity, will move the robot from a distance of 10 inches to a distance of 5 inches from the wall ($r(t)=5$), as shown in Figure 4. The distance will be measured using the front short distance sensor. The task has to be performed using 5 different values of K_p(0.5, 1, 3, 5, and 20). For each value of K_p you will need to provide a graph showing Time vs. Distance from Goal, in addition to all sections required in the task report

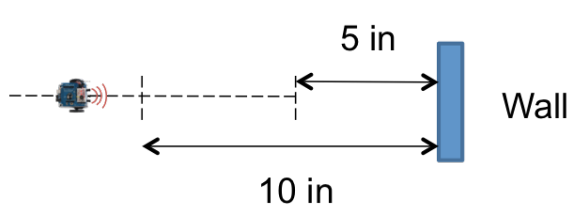

Figure 4: Wall Distance Task

2. Wall Following (40 points)

Task 2 requires the robot to move around the wall while keeping a min 5 inches of distance from the wall, as shown in Figure 5. Use the same close loop control implementation as in Task 1 with appropriate proportional gains to control front distance to the wall where robot needs to turn.

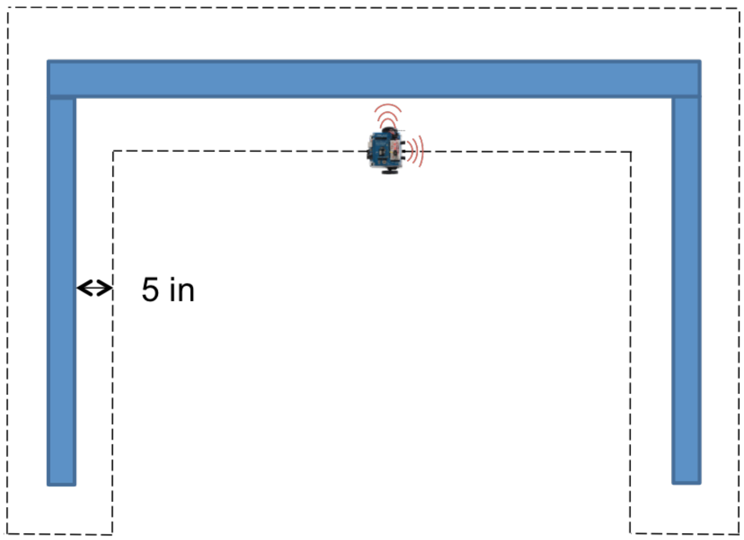

Figure 5: Wall following task

3. Corridor Navigation (40 points)

Task 3 requires the robot to navigate centered between the walls, as shown in Figure 6. Use the same close loop control implementation as in Task 1 with appropriate proportional gains when reading distances to front walls where robot needs to turn.

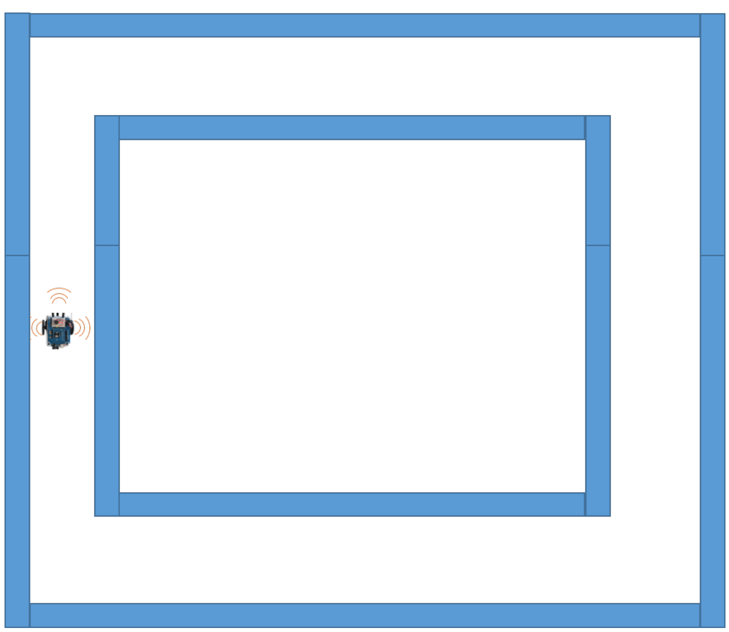

Figure 6. Corridor navigation task