**Lab 3: Inverse Kinematics**

**CSCI 3302: Introduction to Robotics**

Report due 9/25/17 @ 11:59pm

The goals of this lab are to

* Understand how to calculate the wheel speed given a desired velocity
* Experience basic feedback control
* See the need for higher-level reasoning / path-planning

You need:

* A functional Sparkiduino development environment
* A Sparki robot
* A working implementation of the Odometry Lab code
* The ArcBotics line following poster

**Overview**

A robot’s forward kinematics allow you to compute the final position of a robot given individual joint positions. If we wanted to be able to perform the opposite process – figuring out the joint positions required to move a robot to a desired position/configuration, we will need to utilize *inverse kinematics*. A robot’s inverse kinematics provide a joint configuration that will achieve a desired pose (in Sparki’s case: x, y, ). In this lab you will implement a feedback controller that navigates the robot from its current position to a provided goal position.

**Instructions**Each group must develop their own software implementation and turn in individual lab reports. **You are encouraged to engage with your lab partners for collaborative problem-solving**, but are expected to turn in your own write-ups. If your group does not finish the implementation by the end of the class, you may continue this lab on your own time as a complete group.

**Submit your code (.ino file) and a PDF write-up** answering each question (please put the number of each question next to your answers, rather than turning in your answers as an essay).

**Part 1: Compute the Inverse Kinematics of Sparki**

1. Use the forward kinematic relationship for a differential wheel platform from the book ( and as a function of left and right wheel angle change) to calculate its inverse, that is left and right wheel angle change given and .

**Part 2: Position Controller**

1. (Position Error) Calculate the Euclidian distance between your current location and the goal position.
2. (Heading Error) Calculate the angle between the orientation of the robot and the direction of the goal position.  
   *For example, if the line between the robot’s position and the goal position is oriented 90 degrees to the left of the robot, is 90 degrees. If the goal is directly in front of the robot, is 0 degrees (and so on). You should use the function atan2() for doing this.*
3. (Bearing Error) Calculate the angle between the orientation of the robot and the goal orientation.
4. Create a controller that orients the robot in the correct heading, drives the proper distance, and orients to the proper bearing.
5. Verify that your controller works by driving to locations around the map!

**Part 3: Testing and Feedback Controller**

1. Calculate a forward speed that is proportional to , e.g.
2. Calculate a rotation speed that is proportional to and , e.g. and make sure the direction of is consistent with your coordinate system
3. Create a proportional feedback controller that uses this signal to drive to the goal.
4. Verify that your controller works by driving to locations on the map!

**Part 4: Lab Report**

Create a report that answers each of these questions:

1. Does your implementation work? If not, describe the problems it has.
2. What are the equations for the final controller from your implementation?   
   (Wheel speeds as a function of the desired end pose)
3. What happens if you decrease your gain constants (the 0.1 and 0.01 values)?
4. What happens if you increase these gain constants? What if they become too large?
5. What is the role of the position error?
6. What is the role of the bearing error?
7. Why include heading error?
8. What would happen if an obstacle was between the robot and its goal?
9. (Briefly) How would you implement simple obstacle avoidance using the ultrasonic sensor?
10. Describe what would happen if the obstacle-avoiding robot encountered a U-shaped object like this:
11. What are the names of everyone in your lab group?
12. Roughly how much time did you spend programming this lab?
13. (*Optional, confidential, not for credit*) A brief description of any problems (either technical or collaborative) encountered while performing this lab (e.g., issues with the clarity of instructions, clarity of documentation, lab colleague’s behavior, etc.)