Madelaine Struwe
Dr. Hayes
Intro to Robotics
4 October 2017

Lab 3 Report

1. The controller from part 2 might fail when the goal is an infinite distance away, or constantly changing. For our controller we put in the goal position for Sparki to go somewhere. So, if the goal position was to change without it being updated in the Sparki program, it would never reach the goal and only go to its pre-programmed goal.
2. Position error determines the distance between the robot and its goal, from a straight line. It tells us how far the robot needs to go before it is at its goal.
3. Heading error determines the direction the robot is facing. It will help us turn the robot from its starting position to the direction it needs to face to get to the goal.
4. Bearing error determines the angle the robot is facing and what angle it needs to face to be in the right orientation of the goal. It helps the robot face the correct direction.
5. We have the robot constantly update its position as it moves. After each update, it recalculates the position and bearing error at the beginning of each loop. The robot moves a small amount each time, and takes time to recalculate its position and bearing error.
6. For the position we tested, our implementation worked. However, we did not get the chance to test every state and goal so it is possible it does not fully work.
7. The way we implemented things in our code, we did not use equations with theta prime and x prime.
8. Because of our implementation, we do not have gain constants, and cannot alter them to see what would happen.
9. With higher gain constrains, the robot will be further away from the goal position.
10. If there was an object between the robot and its goal, it would run into the object not knowing or caring it was there, to try and get to its goal. When it is too high, the robot can go off track and never reach the goal.
11. Using the ultrasonic sensor to find an obstacle, we could have Sparki see if an object is 15cm away from it, if an object is there, turn clockwise until the object is not there. Then move forward 15cm and turn back twice as many degrees, counter-clockwise, it turned originally to face the same goal it was going to before it had to move around the object.
12. If the obstacle avoiding robot encounters a U-shaped object, it should sense the object, turn clockwise until it no longer sees the U-shaped object. Then move forward, if it sees it again, it should turn clockwise again until it no longer sees it. This process should be repeated until it has cleared the object. After doing so, it should rotate back twice the amount of degrees it turned to avoid the obstacle in the counter-clockwise direction, to find the goal position.
13. The names of everyone in the group is, Sherry Nguyen, Jason Evarts, and Madelaine Struwe.
14. We spent about 4 hours on the programming for this lab.