**Algorithm Description Document**

**1. Leader**

The obstacle avoidance algorithm can be broken down into five stages: identifying the obstacle, deviating from the obstacle, moving past the obstacle, returning to the course, and then finally returning to a neutral state. At any point during this entire process, the obstacle avoidance can be terminated by pressing the “Obstacle” button in the app.

In the first stage (Stage 0), the robot must identify if there is an obstacle in front that it must navigate around. During the core loop, the robot will use the sonar to check if there are any objects in front of it within a certain distance. If there isn’t the robot will move forward a few steps and then evaluate again. If there is an object, then the obstacle has been identified and the robot turns left before starting the next stage.

During the second stage (Stage 1), the robot will turn its sonar to the right and track if it can still see the obstacle. As long as the robot sees the obstacle, it will continue moving forward towards the left. The robot tracks how far it has to deviate with a simple counter for each step, which becomes important for the fourth stage. Once the robot can no longer see the obstacle, then the robot has likely deviated far enough from the path to have the ability to move past the object. At this point, the robot turns right and transitions to the next stage.

Once the third stage (Stage 2) has been reached, the robot must now move past the object. Because the robot needed to turn left at a certain distance in front of the obstacle, it may be a significant distance away from the obstacle to actually be able to track it. For that reason, the robot will now move forward until it can once again see the obstacle to its right. Once it has been identified, the robot will continue to move parallel to the obstacle. The robot will finally turn right and transition to the next stage once it no longer sees the obstacle to the right.

The fourth stage (Stage 3) has the robot return to its original course by taking the same number of steps back that it originally had taken deviating from the course in the second stage. For each step, the robot will subtract from the move counter until it reaches zero. Next, the robot will turn left and enter its final stage. Finally, in the fourth stage (Stage 4), the robot returns to its neutral state by re-centering the radar and turning off the obstacle avoidance flag.

In the second and third stages (Stage 1 and Stage 2), the robot also utilizes course correction. Because the motors on the wheels are not consistent, the robot has some difficulty moving in a straight line. For this reason, the robot will compare its current distance from the obstacle with the previous distance during each forward step. If the robot determines it is moving away from the obstacle, then it will attempt to correct its trajectory by turning slightly towards the obstacle. Likewise, if it’s getting closer to the obstacle, the robot will adjust by turning slightly away. This helps account for some of the variability in how the robot is angled when it’s moving around the obstacle.

**2. Follower**

The follower will look straight if there is an object detected and is within its maximum threshold. If the leader is at ‘equilibrium’ then the follower will stop. This means the robot is at the right distance to follow. If the leader surpasses the equilibrium and the ‘follow-threshold’ but is lower than the mid threshold the follower will begin to move at regular speed.

The follower will increase its speed if the leader is between mid and maximum threshold. That means the distance is far and the follower needs to close that distance. If the distance is below equilibrium and above the minimum-follow-threshold, then the robot will slow its speed due to the leader being too close to avoid collision. If the follower distance is less than the minimum threshold, the follower will stop because it is too close to the leader robot. After these actions are taken, the follower will keep looking straight to detect and gauge the leader’s distance in order to constantly adjust itself. These actions will continue as long as the follower senses the leader.

If the object/leader is within reach, the follower will begin looking at 7 different intervals. It will look left, two intervals between the left and center, center, right, and two intervals between the center and right. These grabs look for an object/leader to detect and follow, avoiding blind spots in the meantime. Depending on the sum of these variables the object will calculate a 'deciding' action to take:

* sts1 = 1 if there is object on its left
* sts2 = 3 if there is an object in the center
* sts3 = 5 if there is an object on its right
* Else all these values will equal 0, meaning no object was detected on its corresponding side.

If the number is 4, two objects were located, left and center. The follower will choose the distance that is least from it to turn and follow. If the number 8, two objects were located, right and center. The follower will choose the distance that is least from it to turn and follow. If the number is 9, that means 3 objects were detected from center, right, and left. The follower will choose the distance that is from it to turn and follow. Depending on the decided variable, the robot will turn right or left, then look straight for it to detect an object/leader to follow. The robot will continue to look and search until it has tagged an object within its threshold.