# Lab 02

# Avoid-Obstacle, Follow-Object and Random Wander Worksheet

Robot Name \_\_\_\_\_\_\_\_Arkin\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team Member Name \_\_\_\_\_\_\_\_Ben Williams\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team Member name \_\_\_\_\_\_\_\_Donald Woodruff\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Purpose

**In your own words, state the purpose of lab 02 in the following space.**

To use sonar and lidar sensor data in combination with feedback control schemes to control a mobile robot to avoid or follow objects.

## Part I – Random Wander (Layer 1)

**What was the general plan you used to implement the random wander and obstacle avoidance behaviors?**

The robot heads in a direction randomly until it detects an object either in front of the robot or beside the robot. If there is an object in front of the robot, the robot will quickly turn to avoid driving into the obstacle. If there is an object beside the robot, the robot will gradually turn away from that object.

## Part II – Test the Sonar Sensors

**Based upon your testing, what is the maximum and minimum range for the sonar sensors on your robot?**

The sonar sensors seemed to function from 5 to 30 cm, however they would report values that often quite wrong.

**What is one advantage of using a sonar sensor instead of an infrared sensor?**

The sonar sensors seemed to have a larger max range when compared to our infrared sensors.

**In the following space, discuss the accuracy of the sensor**.

The sonar sensors on our robot would often give values very different from what we expect. The sensors output values would often fluctuate significantly despite the distance to an object in front of them not changing.

## Part III – Test the Infrared Lidar Sensors

**Based upon your testing, what is the maximum and minimum range for the infrared lidar sensors on your robot?**

The infrared lidar sensors had a range of 2cm to 20cm.

**What is one advance of using an infrared lidar instead of a sonar sensor?**

The infrared lidar sensors seemed to be much more accurate than the sonar sensors

**In the following space, discuss the accuracy of the sensor.**

The sensor was much more accurate than the sonar sensors from our testing.

## Part IV – Collide Behavior

**What distance did you use for the collide behavior?**

For collide, we used 14cm to trigger the collision behavior.

**Did you use the lidar, sonar or a combination of the two sensors? How did you decide?**

We used exclusively lidar sensors for our collide behavior, because the sonar sensor readings were too inconsistent to be useful.

**How could you implement collide behavior for sensors other than the front sensors?**

For the left and right sensors, if the robot detects a close object, the robot will speed up the wheel on the side closest to that object, causing the robot to gradually turn away from the object. For objects detected behind the robot, the robot will stop if it is moving backwards and towards the object.

## Part V – Runaway Behavior (Layer 0)

**How did you represent the feel force function on your robot?**

The sensor readings from all of the robots lidar sensors were summed to create a repulsive vector, which was used to turn the robot away. The robot then turns to this angle and continues it forward motion.

**How did the robot respond with respect to the force felt, note that it could move in reverse, or it could turn a certain angle, think about what makes most sense for potential field navigation.**

The robot would respond by moving away from the forces felt. Depending on the direction, it may make either a gradual or a sharper turn.

**How does your robot handle getting the robot unstuck when the vectors sum to zero?**

The robot turns to a 45 degree angle, to allow the sensors to see new areas and it may then be able to escape.

## Part VI – Follow Behavior (Alternate Layer 0)

**In your own words, describe how to implement proportional control to follow an object,**

The distance between the mobile robot and the object is measured, then the motor speeds are controlled proportionally to the distance measured. If the robot is too close to the object, it will move away. If it is too far, it will move closer. If it is within the desired distance, it will stop moving.

**Are there situations where the robot would get stuck when following? If so, how would the robot correct for that?**

If the robot finds itself stuck in the same position for a long set amount of time, it could turn or move a prescribed distance to get itself unstuck.

## Part VII – Subsumption Architecture – Smart Wander Behavior (Layers 0 and 1)

**What is the difference between on-off and proportional control?**

On and off simply turns the motors on and off with no speed adjustment, whereas proportional control will change the speed of the motors depending on the error in the desired value, such as the error of a desired distance from an object.

**How did feedback control improve the random wander and avoid obstacle state machine?**

Using feedback control allows for more finite adjustments of the motor control and thus robot position while moving with respect to some target, such as moving away from an object.

## Part VIII – Subsumption Architecture – Smart Follow Behavior (Layers 0 and 1)

**You robot currently runs either a smart wander or smart follow behavior. How could you create a state machine that integrates smart follow and smart wander? What type of input would trigger the avoid versus the follow behavior?**

The state machine would switch between smart follow and smart wander depending on whether the robot senses an object near it. If there is an object, it was trigger smart follow, otherwise it will continue to smart wander.

## Conclusions

1. **How well did your software design plans match the reality of what you implemented on the robot?**

There were lots of aspects to the implementation that we did not account for in our design plan, however we roughly followed the design plan well.

1. **How well did your software design plans match the reality of how the robot performed? Compare it to the theory you learned in class.**

Our software design plan did not encapsulate a large majority of the functionality that was needed for our actual robot.

1. **How did you create a modular program and integrate the two layers into the overall program?**

We implemented a two layered state machine to control the robot.

1. **Did you use the sonar and IR sensors to create redundant sensing on the robot’s front half?**

Through testing, we discovered that the sonar sensors are highly inaccurate, so we used only the lidar sensors for sensing.

1. **How could you create a smart wander routine to entirely cover a room? Similar to what a Roomba does.**

The robot could move in a certain direction until it hits an object, then turn 90 degrees, move a short distance, then turn 90 degrees again. This would allow for the robot to move back and forth across the floor.

1. **What kind of errors did you encounter with obstacle avoidance behavior?**

We had difficulties with sensors reading consistently, which caused it to be difficult to avoid obstacles as consistently as we would like.

1. **How could you improve obstacle avoidance behavior?**

Adding sensor redundancy by using both sonar and lidar sensors would help greatly with ensuring accurate sensor readings.

1. **Were there any obstacles that the robot could not detect?**

Obstacles that were small were often hard for the robot to detect.

1. **Were there any situations when the range sensors did not give you reliable data?**

The sonar sensors often gave inaccurate and inconsistent data. The lidar sensors worked better, but would occasionally gives readings of 0 even when something was placed in front of it, which had to be dealt with through the robot’s programming.

1. **How did you keep track of the robot’s states in the program?**

A state machine was used to keep track of the robot’s states. Certain triggers were used to transition the robot between states.

1. **Did the robot encounter any “stuck” situations? How did you account for those?**

If the robot got stuck, the robot would turn 45 degrees and take new sensor readings.

1. **What should the subsumption architecture look like for the addition of the go-to-goal and avoid- obstacle behaviors?**

Go to goal should run until an object is detected, at which point the object should move alongside the object while still avoiding it. Once the object is no longer blocking the path towards the goal, go to goal should resume.

1. **What did you learn?**

We learned a lot about handling sensor inaccuracies, how to implement on-off and proportional control, how to recover from stuck positions, and how to use dual cores to read sensor data in the background.

1. **What did you observe?**

We observed that the polling rate of the sensors seemed to be surprising low, which sometimes made it difficult for the robot to react fast enough to objects in it’s path.

1. **What questions do you still have?**

How could poor sensing data be dealt with better?

## Appendix

Insert your properly commented, modular code in the following space.