Juan Pablo Vielma A01203239
Danny Avendaño A01207115
José Juan Gálvez A01272733

1. What problems did you come across during the lab?

During the lab, we tried out to simulate different behaviors. Some of these behaviors were moving in circles, moving back & forth, drawing a square and an infinite spiral. Our first challenge was making the robot stop, because using the instruction “motor.stop()” didn’t do any good. It turns out that this wasn’t completely our fault, we were using a defective robot and replacing it solved our issue. Our next challenge was when trying to draw an “infinite” spiral. Our first try just drew a circle (which was expected), we tried using different speeds on each wheel, which were constant. Moving forward on the spiral, we increased the speed of one wheel for 1 unit every 50ms. To our surprise, we discovered that if you initialize motor speed to zero, order the motor to move forward and then increase the speed, nothing happens. Since motor speed is set to zero, the instruction “motor.forward()” didn’t go through. Nevertheless, we figured out the problem and solved it by initializing the motor’s speed to 1 instead of 0.

Our final program was drawing a pattern in the form of a lemniscate symbol, better known as an “infinite symbol” or well, drawing an “8”. The robot is expected to change directions once it detects a collision. One issue we encountered was changing behavior immediately after the collision was detected. This is happening because the function that detects collision is validated before/after a certain movement cycle is completed. To solve such problem, we figured out that the solution was using threads. Using a thread that works as a listener to detect collision can help us detect and validate collisions concurrently. On one thread we check for collisions, if such collision happens, another behavior will take place. Once it has finished running, control will be given back to the main thread and normal execution will continue. Our biggest challenge applying threads to our project was struggling with static methods when trying to compile the code.

2. What modification would be required to turn your behaviour into a solution for any real problem? i.e. Where could it be used and how?

In order to give more functionality to the robot, we can use the sensors (Collisions, sounds, vibrations, GPS, etc). This would give us the opportunity to adapt the behavior of our robot to solve real life problems. For example, in an earthquake, a rescue robot must have the ability to identify where the walls are so it can know what is the best path to follow. In addition, it must listen and recognize human sounds so it knows what to do and how to act. Besides, a good feature that this kind of robot could have, would be a connection to the internet and big data centers. This would be helpful in order to analyze data so the robot can compare within a database another kind of disaster and use AI so it could know which are the best actions to do and stay connected with other robots so they can work as a group. This could improve the service being better and more efficient.

Another application in which this could be use is in self-driving cars