Reflective Summary

1st Matthew Walther
Ritchie School of Engineering and Computer Science
University of Denver
Denver, Colorado, United States of America
matthew.walther@du.edu

***Abstract*—This document summarizes the task that the robot was given, a brief overview of the original learning in simulation, and a reflection on the changes made to make the robot work in a real world environment while explaining some of the difficulties that come with real world implementation compared to a simulated environment.**

***Index Terms*—Q-Table, Reinforced Learning, Q-Learning, Simulation to Real World**

I. Introduction

For this project the robot was taught how to right wall follow through reinforced learning. It was trained with a Q-Table consisting of three actions being move forward, turn left, and turn right plus three main states left, forward, and right that were each split into 3 sections of range being close, medium, and far. Combined these states and actions formed the Q-Table that the robot would build while learning in a simulated environment. After learning the robot was then places in a real world environment that resembled the simulation that it had learned in. When in the real world environment the robot would utilize its learned Q-Table to navigate the room efficiently without crashing into the box walls used to build the space.

II. Discussion

A. Differences between Sim and Real World

There are many differences between a simulated environment and a real world environment, one such difference that needed to be accounted for was the difference in size of the environments the robot would navigate.The environment given to the robot in simulation was quite large compared to the real world environment constructed to test its learning by almost twice the size. Due to this change the code for the robot had to be adjusted to account for it as it would occasionally pick up the wall on the other side of the environment which would effect its greedy decision action choice. Another major difference was the hardware present in the robot in the real world which would operated at a different speed compared to the simulated version, having different robots to perform the testing on led to having two different Lidar's on the different models which opperated at different speeds that would return a different sized array of scan readings that would affect how the states could be interpreted for deicison making.

B. Changes Made between Project 3 and Project 4 Code

Due to the differences in the environment and access to topics for the robot parts of the code for the robot to navigate the environment in test mode had to be changed for project 4. First any use of gazebo topics such as setting the model's position and physics pausing or un-pausing had to be removed since it was no longer a simulated environment. Then due to the difference in environment size the sectioning of ranges for the states left, right, and forward had to be adjusted so that it could properly compile states to choose an action as it was detecting a wall on the opposite side of the room that should have been ignored and would effect the state being constructed which in turn changed the action taken. After the ranges for each state section were adjusted the next part changed was the actual section sizes themselves. First was the forward section being detected which originally was 90° section consisting of 0° to 44° and 315° to 359° was changed to a 45° section defined as 0° to 9° and 325° to 359°. The next state section changed was the right which detected a 90° section consisting of 225° to 314° was changed to detecting a 13° section from 270° to 283°. Once the sections and their range divisions were adjusted for the new environment the code for selecting the index of each angle returned from the Lidar scan had to be changed since there were two different ways the scan data could return depending on which model turtlebot was used, one would return a full 360° reading at a slower speed or the other would return around 270 scans with a radian increment for the angle between each scan index. To account for this change the angle increment was taken and converted to degrees to then be divided into the angle used to define the state section range in order to find the scan data index. Once that change had proved to work for either turtlebot model the state definition code was changed to better decide on which state the robot was in, this part was changed from finding the average distance in that side's section to finding the minimum value in that section while throwing out potential invalid reading that would be less that the minimum range possible of the laser scan message. This change helped the initial change of each section's range of close, medium, and far work more effectively in the real world environment.

C. Challenges of Real-World Implementation

With implementing code for a robot in the real-world it comes with challenges, some challenges present in this implementation were the possibility of invalid readings being

returned from the Lidar laser scan that could effect state building, another one was the challenge of testing the robot in the different environment from the one it learned from while in simulation. Finally was the limitations of hardware when running code made for simulation, this was present in testing when the sleep function was removed for the real world but caused the code to operate faster than the robot was sending data back causing the code to tend to operate at a different pace than robot leading to collisions with the environment walls and getting lost in space.

D. Impact of Simulation on Real-World Application

With simulation it can serve as a proof of concept for a robot if it successfully completes its task which for this project was wall following. Having it complete the course in simulation proved that it can work and that with some adjustments to the code it was also able to work and be applied to the real world.

III. Observations and Conclusions

Overall I had seen many differences between implementation in a simulation and the real world which can lead to having to make a variety of changes to the original code. In conclusion the differences between them can lead to thinking about new ways that the issue can be addressed while also showing you things that might have been overlooked while working in a simulated environment like it did for me. The changes made to the code between project 3 and project 4 took quite some time to get working but in the end made the robot operate more effectively and if I take those changes back to the simulation will probably end up working well there as well but may need changes as well showing how different those environments can be.