Masters in Robotics Engineering,
Shima Amiri Fard,

Supervisor: Prof. Carmine Tommaso Recchiuto

Research Track II
Statistical Analysis

1. Introduction

In this report, as an assignment of the Research Track II course, the statistical analysis of two different implementations is studied. The main objective of both algorithms is to drive a robot autonomously by grasping the silver tokens, and putting them behind; the robot should also avoid obstacles, golden tokens. In Figure 1, the map of the environment and the robot are shown.

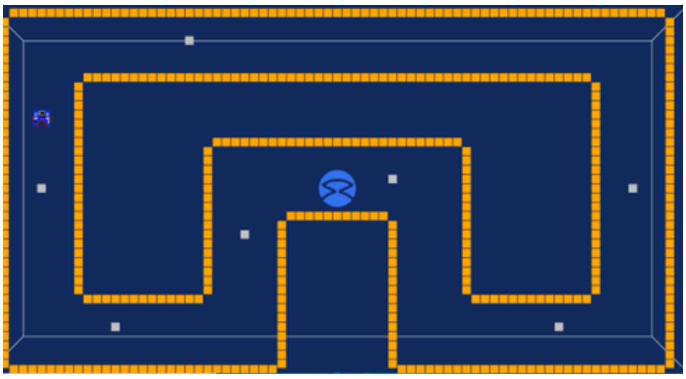

Figure 1: Robot and the simulation environment

The two algorithms are:

Algorithm A: assignment 1 of the Research Track I course, provided by Shima Amiri Fard (https://github.com/shimaamiri/ResearchTrack1_assignment1)

Algorithm B: assignment 1 of the Research Track I course, provided by professor Carmine Tommaso Recchiuto (https://github.com/CarmineD8/python_simulator/tree/rt2/robot-sim/solutions)

Given the python environment as shown in figure 1, the two algorithms compared pass instructions to the robot in the simulated environment to move thought the circuit while grabbing and dropping silver tokens and avoiding golden tokens. A clear indication of performance – considering all environment variables remain constant (token positions and robot starting point) will be the time factor. A measure of the time between each silver token grabbing and releasing events can be measured, as they distinctively define the amount of distance travelled due to their being equally spaced within the environment, and the efficiency of task execution,and overall time taken to lap the circuit. For this experiment, we have considered the results with a level of significant of 5%.

2. Methodology

To compare the algorithms, data extraction was performed for two analysis parameters: the robot's change in direction near walls, and the time taken to complete the circuit. Modifications were made to both codes to capture the necessary data accurately.

In addition to the existing tokens, an extra token was added to the environment to assess the algorithms' behavior. The positions of all tokens were adjusted, ensuring that silver tokens were not placed too close to the walls. The modifications were made in the "**sunny_side_up_arena.py**" file.

To obtain data for the first algorithm, main objective is to focus on when the robot is close to a wall. Moreover, I modified both the codes where the lines manage the change of direction close to a wall (therefore close to the golden tokens), simply by adding a + 1 each time the running program goes in that part of the code.

To obtain data for the second algorithm, main motive is to monitor the time as soon as the running program enters the robot movement part, which is usually the main function. The code is set to take the end time whenthe robot finishes to grab the last silver token. Therefore, as you can think it is not a real circuit, but as far as the final token is not randomly placed, it makes the time a good parameter to be analyzed.

2.1 . Ho and Ha Hypothesis

To compare the algorithms' performance, two hypotheses were formulated: Ho (null hypothesis) and Ha (alternative hypothesis).

Ho, null hypothesis: the robots in both codes complete the circuit in a similar time, implying their mean completion times are approximately equal.

Ha, alternative hypothesis: the robots in both codes do not complete the circuit in a similar time, indicating a difference in their mean completion times.

A total of 10 observations were recorded for each code, measuring the time taken to complete the robot tasks in seconds. The observations were automatically collected through embedded instructions in the Python script. The observations for My Code and Professor Code are presented in the table below.

Algorithm A		Algorithm B	
Collision	Trajectory	Collision	Trajectory
119.348418951	31	163.876564329	11
119.375055075	29	161.876543234	11
118.752519131	26	162.434465788	11
119.88808918	26	160.123543217	12
117.76483202	26	166.384288633	11
117.634191036	28	167.435908945	11
120.362953901	32	165.072348904	10
114.881556034	30	162.43828492	11
115.949737072	34	162.734284387	10
116.154567003	27	164.173265890	10

3. Results

This section will describe the experimental results which shows the analysis of the two algorithms i.e. My Code and Professor Code. Figure below shows the trajectory and completion time difference between code A and code B.

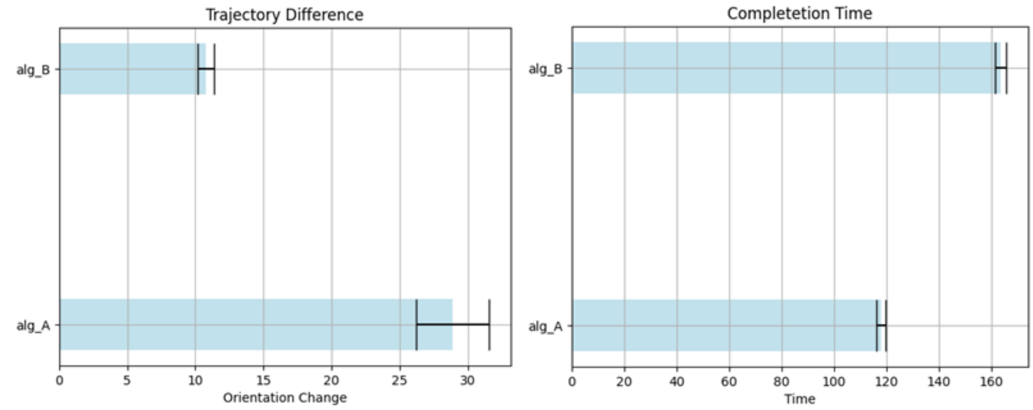

Figure 2: Left. difference between trajectory of both algorithms. Right. difference between completion time of both algorithms

Figure 3 compares the time needed for the robot to reach the goal and number of collisions for both algorithms.

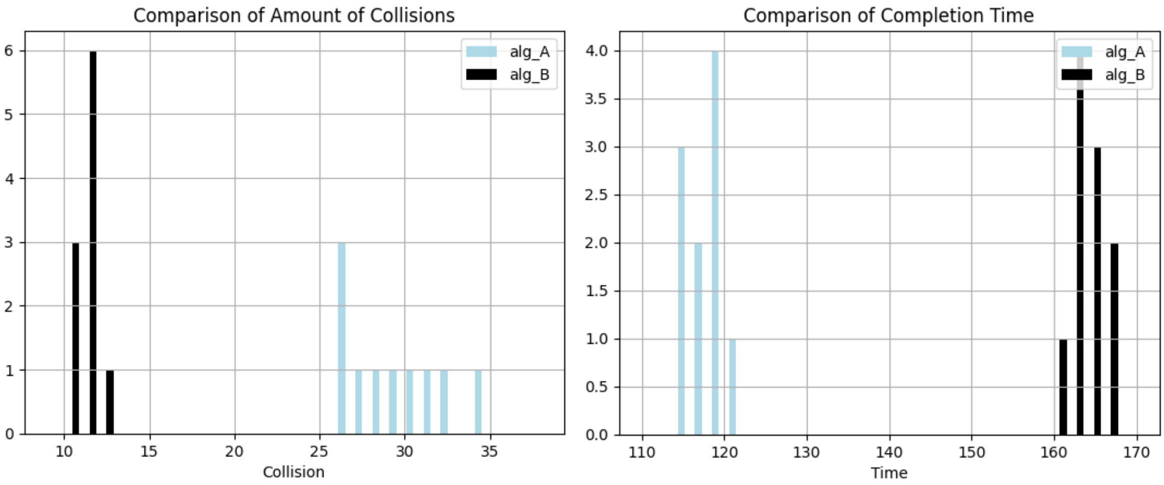

Figure 3 Left. Comparison of number of collisions. Right. Comparison of completion tim

3.1 . T-Test Analysis

For Statistical Analysis and evaluation of our results, the T-test has been selected. Using theSciPy library, the T-test method was used to give required results corresponding to a two-tailed test.

A T-test, specifically the two-sample T-test, was selected as the statistical method for comparing the performance of the two algorithms, My Code and Professor Code. The T-test, also known as Student's Test, is based on the t-distribution and is commonly used to assess the significance of differences between sample means.

The choice of the T-test is appropriate for this analysis due to the following reasons:

Small Sample Size: The study involved a relatively small number of observations for each code. In such cases, the T-test is considered suitable as it does not require a large sample size to produce reliable results.

Independent Samples: The observations collected for My Code and Professor Code are independent of each other, as they represent separate runs of the algorithms. The T-test is well-suited for comparing the means of two independent samples.

The two T-test analysis is performed in **Jupiter** between the two data and expected thatthey’re going to be deeply different. This is a test for the null hypothesis that two independent samples have identical average values. This test assumes that the data have identical variances by default. The results indicate significant differences between them.

	Algorithm A	**Algorithm B**
Collision Static	34.03	-25.59
P Value	5.05e-40	2.75e-33

The p-values obtained were lower than the specified significance level, and the negative t-values led to the rejection of the null hypothesis, providing evidence in support of the alternative hypothesis.

4. Discussion

The statistical analysis reveals notable differences between algorithm A and algorithm B in guiding the robot within the environment. Algorithm A demonstrates superior performance with lower lap times and a greater distance from wall, golden tokens. Therefore, this minimizes the occurrence of collisions. Additionally, algorithm A exhibits faster completion of laps compared to the algorithm B. These findings align with the initial assumptions and are further supported by the statistical tests conducted.