Statistical Analysis Report

Sayna Arghideh

Research Track 2

Prof. Carmine Tommaso Recchiuto

Introduction:

This report evaluates the performance and efficiency of the robot navigation algorithm implemented in the provided Python scripts (run.py and the main robot control script). The robot's task is to identify and collect gold tokens within a simulated environment, using its sensors to navigate and manipulate its surroundings.

Objectives:

-Evaluate the robot's performance in collecting gold tokens.

-Determine if the robot's navigation strategy effectively minimizes the time and effort required to collect tokens.

Hypotheses:

-Null Hypothesis (H0): The robot's navigation strategy does not significantly impact the time and effort to collect tokens.

-Alternative Hypothesis (Ha): The robot's navigation strategy significantly reduces the time and effort to collect tokens.

Experimental Setup:

The simulation environment requires the following dependencies:

-Python 2.7

-Pygame

-PyPyBox2D

-PyYAML

The simulation runs the robot script, which controls the robot's movements and actions to collect gold tokens. The provided run.py script facilitates running the simulation with different robot control scripts.

Methodology

1-Robot Navigation Algorithm:

-The robot continuously searches for gold tokens using the (‘find_token_gold’) function.

-Upon detecting a token, the robot adjusts its position based on the token's distance and rotation.

-The robot collects the token if it is within a predefined distance threshold (‘d_th’).

2- Performance Metrics:

-Time to Collect Tokens: Measured from the start of the simulation to the collection of the last -token.

-Number of Adjustments: Number of times the robot adjusts its direction or position.

3-Data Collection:

-Simulations were run multiple times to gather data on the robot's performance.

-Each run recorded the time taken to collect all tokens and the number of adjustments made.

Results

The results of the simulations are summarized in the table below:

Run	Time to Collect Tokens(s)	Number of Adjustments
1	135	18
2	140	20
3	130	17
4	145	21
5	138	19

Statistical Analysis

Descriptive Statistics:

- Mean Time to Collect Tokens: $\bar{X} = \frac{135+140+130+145+138}{5} = 137.6$ seconds
- Standard Deviation (Time): $\sigma = \sqrt{\frac{\sum(X-\bar{X})^2}{N-1}} \approx 5.77$ seconds
- Mean Number of Adjustments: $\bar{A} = \frac{18+20+17+21+19}{5} = 19$
- Standard Deviation (Adjustments): $\sigma = \sqrt{\frac{\sum(A-\bar{A})^2}{N-1}} \approx 1.58$

T-Test for Time:

Null Hypothesis (H0): The mean time to collect tokens is not significantly different from a benchmark time of 150 seconds (assumed).

Alternative Hypothesis (Ha): The mean time to collect tokens is significantly less than the benchmark time.

1-Test Statistic

$$t = \frac{\bar{X} - \mu_0}{\sigma/\sqrt{N}} = \frac{137.6 - 150}{5.77/\sqrt{5}} \approx -5.38$$

Degrees of Freedom (DoF): $N - 1 = 4$

Critical Value (one-tailed, $\alpha = 0.05$): $t_{critical} = -2.132$

Conclusion: Since $t = -5.38 < t_{critical} = -2.132$, we reject the null hypothesis. The robot's navigation strategy significantly reduces the time to collect tokens compared to the benchmark.

T-Test for Adjustments:

Null Hypothesis (H0): The mean number of adjustments is not significantly different from a benchmark of 25 adjustments.

Alternative Hypothesis (Ha): The mean number of adjustments is significantly less than the benchmark.

Test Statistic:

$$t = \frac{\bar{A} - \mu_0}{\sigma/\sqrt{N}} = \frac{19 - 25}{1.58/\sqrt{5}} \approx -8.50$$

Degrees of Freedom (DoF): $N - 1 = 4$

Critical Value (one-tailed, $\alpha = 0.05$): $t_{critical} = -2.132$

Conclusion: Since $t = -8.50 < t_{critical} = -2.132$, we reject the null hypothesis. The robot requires significantly fewer adjustments than the benchmark.

Discussion

The statistical analysis indicates that the robot's navigation algorithm effectively reduces the time and number of adjustments needed to collect gold tokens. These results suggest that the implemented strategy is efficient and robust within the simulated environment.

Recommendations

Further Testing: Conduct additional simulations with varying numbers of tokens and different obstacle configurations to validate the robustness of the navigation algorithm.

Algorithm Optimization: Investigate potential optimizations to further reduce the time and adjustments required.

Conclusion

The robot's navigation strategy significantly improves the performance in collecting gold tokens, as evidenced by the statistical tests conducted. These findings support the effectiveness of the implemented algorithm in the given simulation setup.