Research Track II Assignment 3

Statistical Analysis of RT1 Assignment 1

By Jack McKenna s5734072

Class: Research Track II

Professor: Carmine Recchiuto

Submission Date: 02/06/2023

Contents

Abstract

This study considers two implementations of a robotic simulation wherein the time-to-complete the simulation is compared using statistical analysis. It was found that the Null hypothesis (H_0) was rejected and thusly the Alternative hypothesis (H_a) was accepted with a significance of 5%, showing that there was a clear difference between the completion time of my colleague and I's implementation.

1. Introduction

This study follows a statistical analysis relating to the Assignment one of the class Research Track I in the first semester of the Robotics course. This assignment involved the creation of a python node to control a robotic simulator that would have a robot identify silver and gold tokens and grab and place silver tokens next to gold ones. This was carried out in a pre-set arena with six of each token and an established generation radius for both. Figure 1 shows my own implementation of the python node in the form of a flowchart.

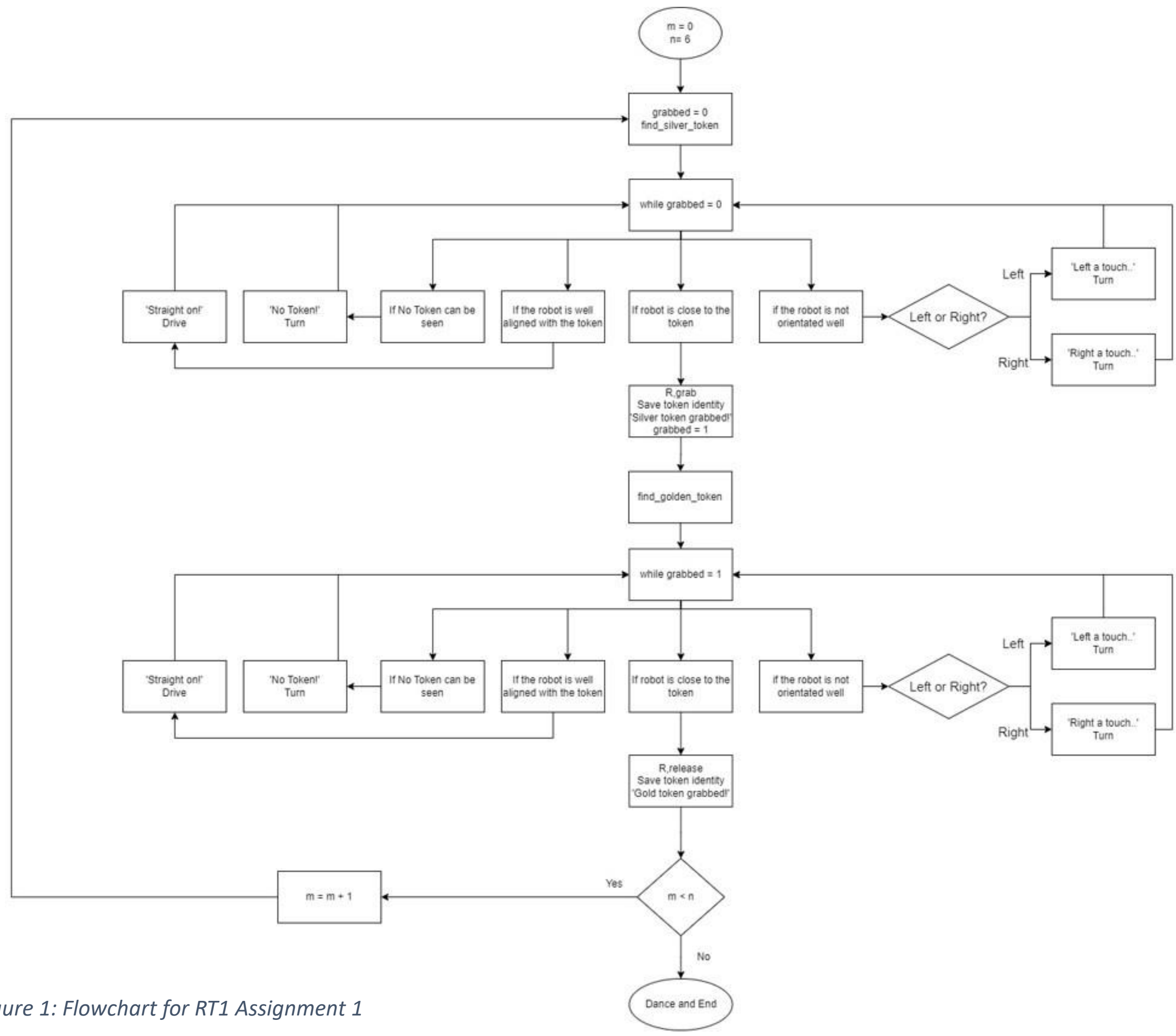

Figure 1: Flowchart for RT1 Assignment 1

Assignment 3 for the class Research Track II takes this created node and manipulates it into distinct implementations that form the basis of the statistical analysis. I chose to work with Tomasz Strzesak (Student Number: s5714359) and we decided to compare our implementations through experimentation. Inspecting the respective python nodes, Tomasz and I determined that the experiments should focus on the completion timing of the robot and compare specifically which node performs faster overall. The results will be inspected visually as well as compared through a T-test statistical analysis to determine this. A T-test is based on T-distribution and provides an appropriate test for judging the significance (or significant difference) of sample means.

2. Hypothesis

In scientific fields, hypotheses are created to suggest new experiments and observations and make predictions about the results of these. With statistical analysis specifically, Null and Alternative hypotheses are considered. The null hypothesis assumes that both methods A and B are equally as good as one another. The alternative hypothesis favours one over the other. The null hypothesis is typically sought to be disproved whereas the alternative approved.

In the case of this study, the hypotheses can be outlined as followed:

- **Null Hypothesis (H_0)** – There is no significant difference between the time-to-complete of Tomasz and I's implementations.
- **Alternative Hypothesis (H_a)** – There is a clear difference between the time-to-complete of Tomasz and I's implementations.

We can also make a general hypothesis wherein it is believed my implementation will overall run faster than Tomasz's, purely due to inspection of each other's python code. This hypothesis would agree with the alternative if proven true.

3. Methodology

Using the file *two_colours_assignment_arena.py* which is in the data arena files for the simulation environment, manipulation of the arena can be achieved. Parameters such as the number of tokens and the radius at which they spawn can be manipulated to allow the user to simulate different scenarios. In our experimentation, to test our hypotheses, we decided to have the robot complete the token stacking objective over 30 iterations, time each one, and then plot a graph of the results. To achieve this, the following parameters were set constant or variable:

- **Constants:**
  - Number of tokens = 3
  - Silver token radius = 0.9
  - Home zone size = 2.5
- **Variable:**
  - Gold token radius

It was important to eliminate human error when timing these simulations, so the python time function was employed to print an accurate timing to the terminal after the robot had finished the simulation. Figure 2 shows the appended timing code to the assignment file.

```
import time
start = time.time()
end = time.time()
total_time = end - start
print('End')
print(str(total_time))
```

Figure 2: Timing Code

The radii of the gold token for each iteration were generated in an excel file with the command:

$$= RAND() * (([Upper\ limit] - [Lower\ limit]) - [Lower\ limit])$$

Where:

$$Upper\ limit = 2.8$$
$$Lower\ limit = 0.3$$

These limits were determined by inspection as a higher radius would push the boxes outside of the reachable zone and a lower radius would lead the boxes to overlap and cause issues for the robot. We each ran the 30 simulations and recorded the results on Table 1 within the appendix which shows the generated gold token radius list as well as the timings acquired over the 30 iterations.

4. Analysis and Results

Figure 3 displays a graph of the times recorded over the 30 iterations generated using Table 1.

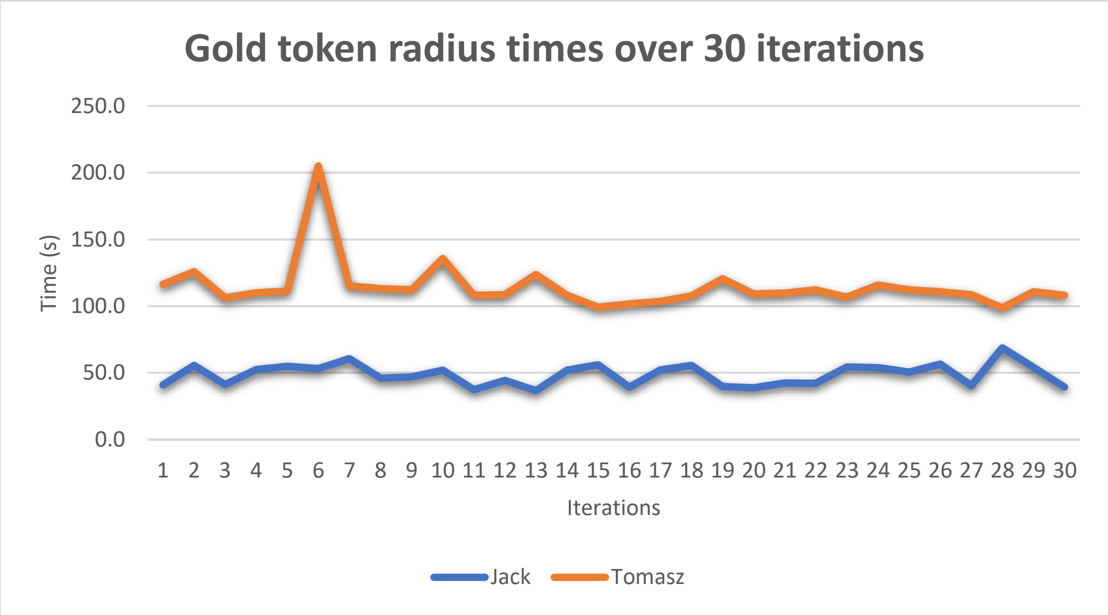

Figure 3: Gold token radius times over 30 iterations graph

Using the data collected, the first step is to calculate the mean value for both sets of data using equation (1).

Equation 1: Mean Equation

$$\bar{X} = \frac{\sum_{i=1}^{i=n} X_i}{N}$$

Where, $X = set\ data, N = number\ of\ iterations.$

Tomasz and I's means were calculated to be:

- $\bar{X}_J = 48.7\ s$
- $\bar{X}_T = 114.5\ s$

Next, we can calculate the standard deviation of our sets using equations (2).

Equation 2: Standard Deviation Equation

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(X_i - \bar{X})^2}{N}}$$

Where, $X = set\ of\ data, \bar{X} = mean\ of\ set, N = number\ of\ iterations.$

Tomasz and I's Standard deviations were calculated to be:

- $\sigma_J = 8.059$
- $\sigma_T = 18.733$

We can also calculate the variance which is simply the standard deviation squared.

- $S_J^2 = 64.947$
- $S_T^2 = 350.925$

Clearly, upon inspection, we can decide that my python code generally runs faster than Tomasz's. Having a smaller mean and standard deviation as well as looking at Figure 3 we can confirm the general hypothesis to be accurate. However, a statistical analysis is still required to reject or confirm the null or alternative hypotheses. As stated previously, a T-test was performed to do this.

We can calculate the pooled variance using equation (3).

Equation 3: Pooled variance equation

$$\widehat{\sigma_{pooled}^2} = \frac{(N_1 - 1)S_J^2 + (N_2 - 1)S_T^2}{(N_1 - N_2) - 2} = 207.888$$

Where, $S^2 = variance, N = number\ of\ iterations.$

And thusly the pooled standard deviation using equation (4).

Equation 4: Pooled standard devation equation

$$\widehat{\sigma_{\overline{X_1} - \overline{X_2}}} = \sqrt{\widehat{\sigma_{pooled}^2}\left(\frac{1}{N1} + \frac{1}{N2}\right)} = 3.723$$

Where, $N = number\ of\ iterations.$

We can then calculate the T-value with equation (5).

Equation 5: T-value calculation

$$t_{\overline{X_1} - \overline{X_2}} = \frac{\overline{X_1} - \overline{X_2}}{\sigma_{\overline{X_1} - \overline{X_2}}} = 2.866$$

To complete the statistical analysis, we can refer to the standard T-table. Using the following criteria:

- Confidence level 95% (5% significance)
- Degrees of freedom (DOF = (30+30)-2 = 58)

We can extract from the table a critical value $CV = 2.000$.

Comparing the T-values we can see that the critical value is less than our calculated T-value, $CV < t_{\overline{X_1} - \overline{X_2}}$. This means we can confidently reject the null hypothesis (H_0) and accept the alternative hypothesis (H_a).

5. Conclusion

To conclude, the Null hypothesis (H_0) has been rejected due to $CV < t_{\overline{X_1}-\overline{X_2}}$, this means that there is a significant enough difference between Tomasz and I's implementations, which accepts the Alternative hypothesis (H_a). It can also be seen that the general hypothesis, wherein my implementation overall runs faster than Tomasz's, is accurate, this is shown clearly from Figure 3 as well as through Equations 1 and 2 where my mean time is over 60 seconds faster than Tomasz's and my standard deviation is 10 points less.

Appendix

Table 1: Results of timing iterations

Iteration	Jack			Tomasz	
	Radius Number	Time (s)		Radius Number	Time (s)
1	1.2	41.0		2.4	116.24
2	2.4	55.6		0.7	125.82
3	1.5	41.2		1.2	106.21
4	2.2	52.5		1.3	110.2
5	0.3	54.8		1.8	111.23
6	0.6	53.2		1.0	205.14
7	2.8	60.6		2.2	115.23
8	0.7	46.1		2.3	113.23
9	0.4	47.0		2.1	112.23
10	0.3	52.1		0.7	135.84
11	0.8	37.3		1.7	108.22
12	1.2	44.4		1.9	108.74
13	0.8	36.7		2.6	123.78
14	0.6	52.1		1.9	108.24
15	2.5	56.1		1.4	99.2
16	1.3	39.4		1.6	101.73
17	2.1	52.3		1.8	103.7
18	2.4	55.7		2.4	107.72
19	1.6	39.8		1.2	120.7
20	1.3	38.8		1.3	109.2
21	1.6	42.5		1.6	109.74
22	1.5	42.1		2.0	112.25
23	2.3	54.5		1.4	106.71
24	2.1	54.0		2.5	115.73
25	0.6	50.7		2.5	112.28
26	2.5	56.7		2.6	110.75
27	1.7	40.4		2.4	108.74
28	1.0	68.8		1.5	98.69
29	0.6	54.2		1.5	110.73
30	1.3	39.4		2.4	108.23
	Mean	48.7		**Mean**	114.548333
	Standard Deviation	8.059363085		**Standard D.**	18.7332469

The T-table can be found through the following link under "Table of selected values":

https://en.wikipedia.org/wiki/Student%27s_t-distribution#Confidence_intervals