RT2: Report Assignment 1 pt. 2

Matteo Maragliano

April 2022

Chapter 1

Introduction

The 2^{nd} part of the first assignment of Research Track 2 consisted in the realization of a statistical analysis on two different type of robot driving:

- one realized during RT1 Assignment 1;
- one provided by the Professor to have a term of comparison.

The analysis consisted in collecting a certain number of data from some experiments on the robots' behaviour and then states which one is "better" in that specific term.

Our experiments were based on two different measurements:

- lap time: the time the robot used to complete a single lap of the circuit;
- distance: the effective distance covered by the robot during each lap.

Just for a reminder: the goal of the robot is to drive along the arena circuit, grab some silver tokens found along the path and put them backwards before continuing the lap.

Chapter 2

Realization

In order to minimize the additional lines to add to the original code it was provided a sort of *library* containing the functions to compute the statistics.

The program is able to write into some dedicates files *.txt* chosen through a parameter passed during the program launch.

The analysis is divided into two phases:

- Statistics using 2 arenas, collecting 30 simulations for both controllers and both arenas, to perform a statistics on the arena used during the assignment;
- Statistics using the first arena to detect which algorithm behaviour is better. In this case it was collected an amount of 10 data, each one in a different environment removing in each simulation the $1^{st}, 2^{nd}, 3^{rd}, 4^{th}5^{th}, 6^{th}, 7^{th}, 1^{st}$ and $4^{th}, 3^{rd}$ and $4^{th}, 4^{th}$ and 7^{th} respectively and doing 1 lap for each of this set; the data acquired have been processed.

2.0.1 Lap Time

The lap time test was computed by the function *time()* which allows to store the instant of time it is called into. This is called at the beginning and at the end of the lap, it is computed the difference and all these value are written into the file.

2.0.2 Distance Travelled

The distance travelled was a little more tricky to be computed.
We started from the assumption that the robot started always from the same position in the arena.

The function used compute the Euclidean distance between two position, the *previous* and the *current* one of the robot; after that the *previous* is updated with the *current* value. All these values are added to a global variable storing the total distance travelled in each lap.
Once a lap is completed, the distance value is written into the dedicated file.

Chapter 3

Results

All the results obtained were analysed by the help of MATLAB software in order to have a reliable and objective analysis.

3.0.1 Hypothesis

We started from a null hypothesis H_0 which states both robot controller are equally good inside two specific arenas.
We wanted to see if $\mu_{assignment} = \mu_{robot-sim}$ in both cases of the analysis.

So, we tried to reject this hypothesis by demonstrating one is better in term of lap time (speed) or distance travelled (efficiency in driving): both after checking their are good for our statistical analysis.

3.0.2 Procedure

All the data were imported into the MATLAB software by an appropriate parse to have them into a numeric form starting from a text one.

Data elaboration

We computed:

- μ: the mean value of all the data recorded;
- σ: the standard deviation of all the data recorded.

For the standard deviation we used a dedicated MATLAB function, while the mean value was computed manually since easier to do.

Results visualization

Once μ and σ are obtained we plotted the results:

- one plot with the data not sorted and the mean value μ computed;
- one plot with a histogram and its relative normal distribution obtained with μ and σ computed;
- some boxplot for the comparison.

Plot with 30 simulations for the first and second arena:

Data not sorted and mean value:

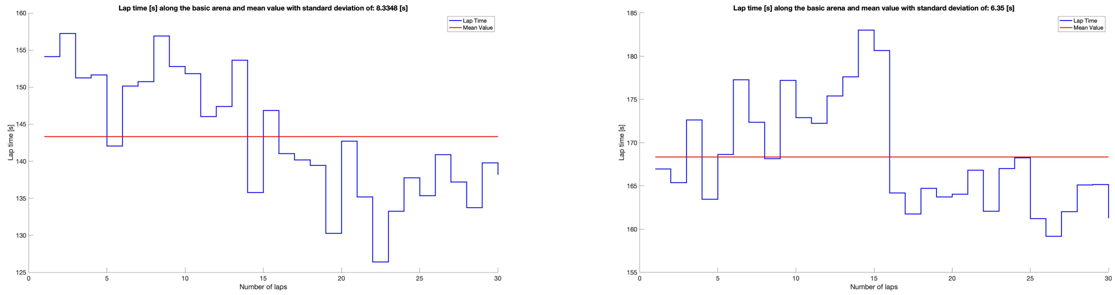

Figure 3.1: Data visualization, not sorted, and mean value for the first arena for lap time.

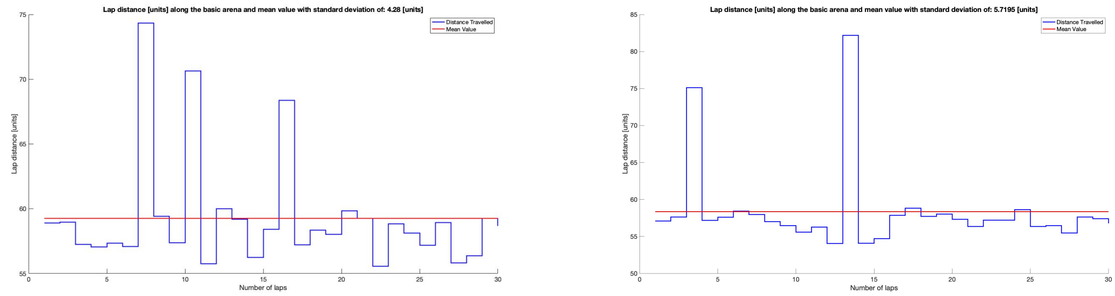

Figure 3.2: Data visualization, not sorted, and mean value for the first arena for distance travelled.

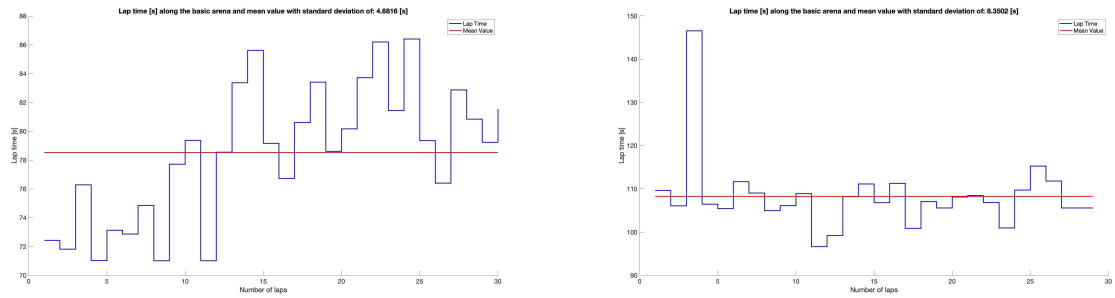

Figure 3.3: Data visualization, not sorted, and mean value for the second arena for lap time.

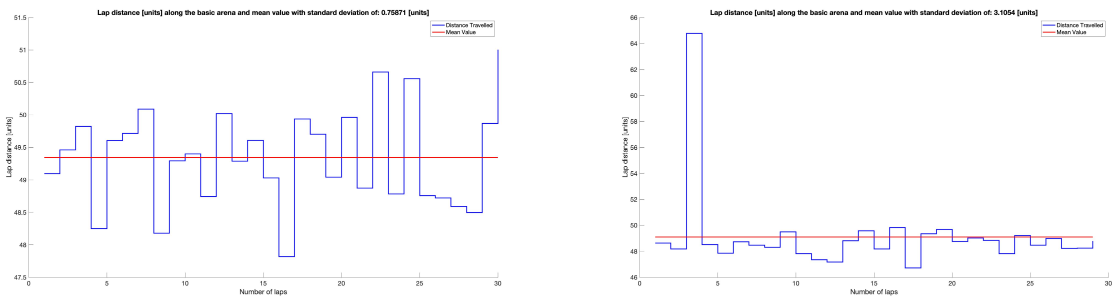

Figure 3.4: Data visualization, not sorted, and mean value for the second arena for distance travelled.

Probability density function with μ and σ obtained:

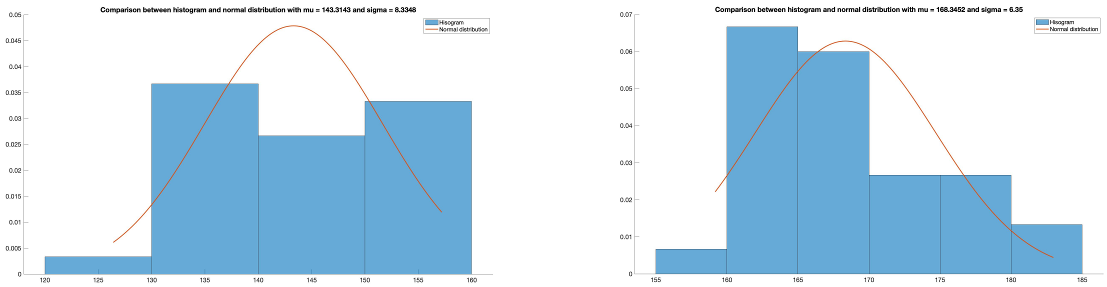

Figure 3.5: Data histogram and probability density function visualization for the first arena for lap time.

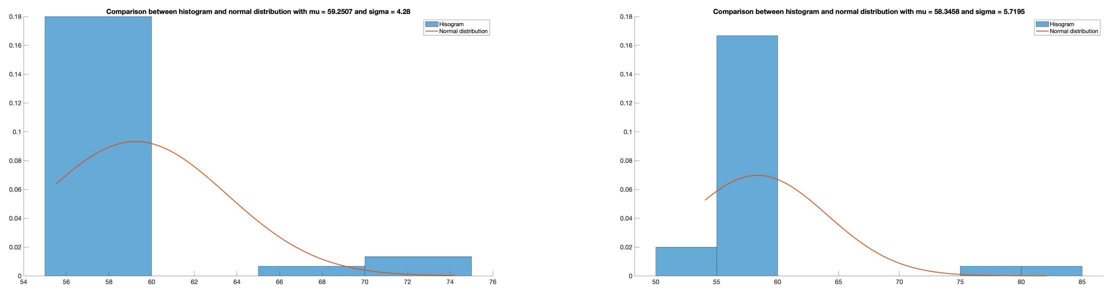

Figure 3.6: Data histogram and probability density function visualization for the first arena for distance travelled.

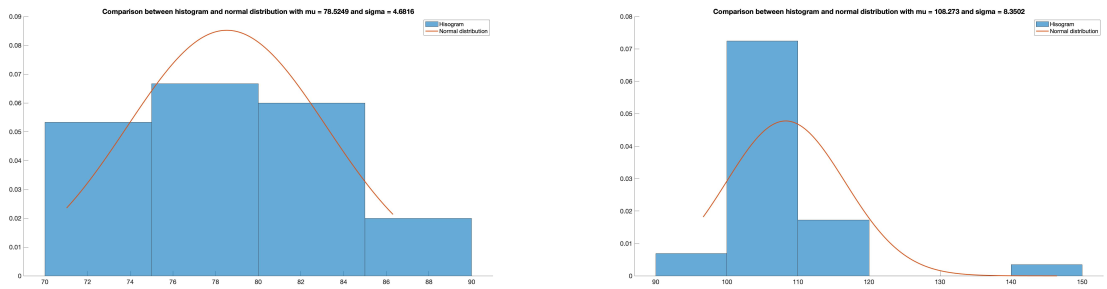

Figure 3.7: Data histogram and probability density function visualization for the second arena for lap time.

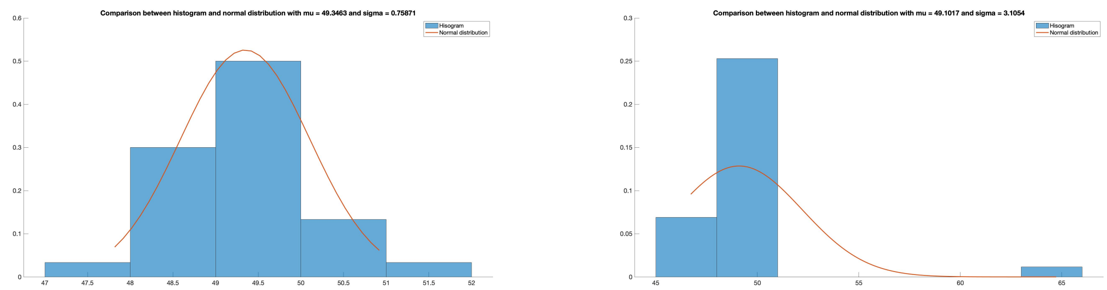

Figure 3.8: Data histogram and probability density function visualization for the second arena for distance travelled.

Plot with 10 simulations, one for each different obstacles location

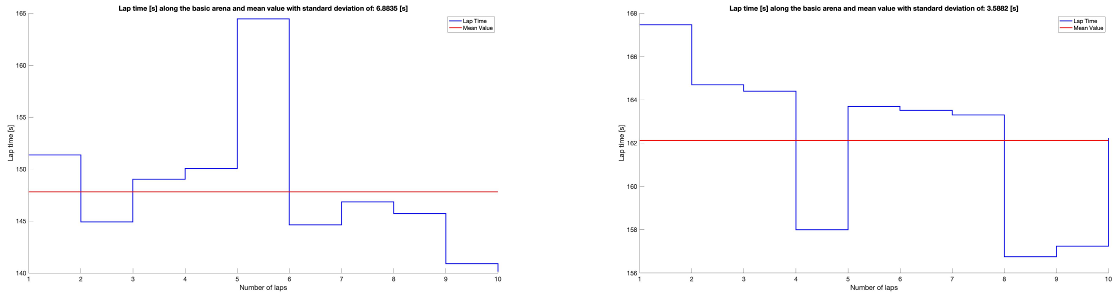

Figure 3.9: Data visualization, not sorted, and mean value for lap time.

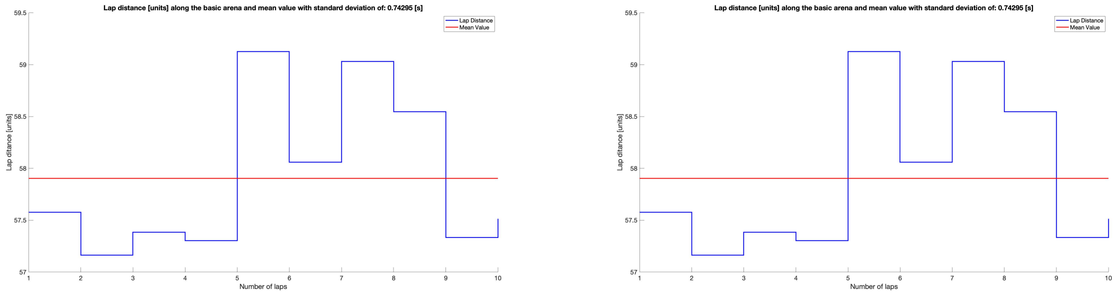

Figure 3.10: Data visualization, not sorted, and mean value for distance travelled.

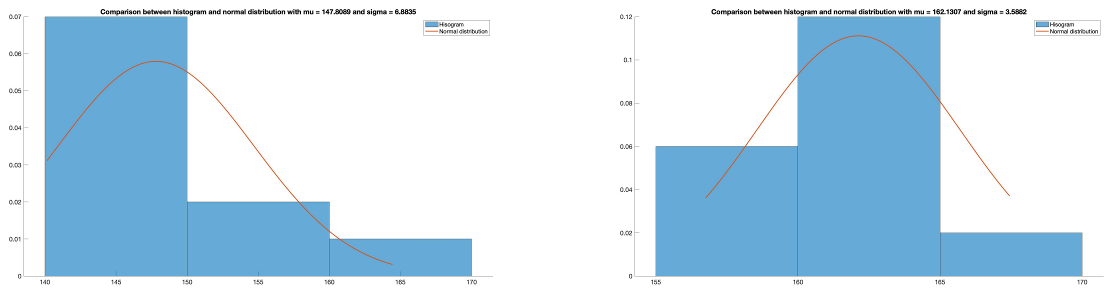

Figure 3.11: Data histogram and probability density function visualization of lap time.

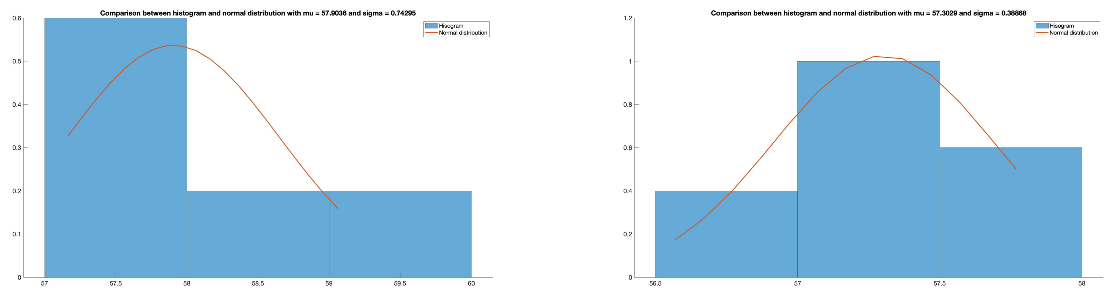

Figure 3.12: Data histogram and probability density function visualization of lap time.

All the graphs have their comparison with a probability density function obtained with the μ and σ computed; the function is drawn following:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

where:

- f(x): probability density function;
- μ: mean value;
- σ: standard deviation.

3.0.3 Lilliefors test

Since we wanted to know if the data acquired belong to a normal distribution, we applied a normality test based on the Kolmogorov-Smirnov.

The *Lilliefors test* is aimed at proving the null hypothesis that the sample belongs to a normal distribution.
However, without specifying which normal distribution; i.e., it does not specify the expected value and variance of the distribution.

Null hypothesis H_0

The null hypothesis for this test is that all the data are from a normal distribution, however, without specifying which normal distribution; i.e. it does not specify the expected value and variance of the distribution.

The function used is *lillietest()* and it returns two values as output:

- 1: if the null hypothesis has to be rejected (the data do not belong to a normal distribution);
- 0: if the null hypothesis has not to be rejected (the data belong to a normal distribution).

With the data acquired into the two arenas we obtained:

Normal distribution: H_0 accepted	Lap time assignment controller (Arena 1)
	Lap time robot-sim controller (Arena 1)
	Lap time assignment controller (Arena 2)
	Distance travelled assignment controller (Arena 2)
Not normal distribution: H_0 rejected	Lap time robot-sim controller (Arena 2)
	Distance travelled assignment controller (Arena 1)
	Distance travelled robot-sim controller (Arena 1)
	Distance travelled robot-sim controller (Arena 2)

Once obtained these results we computed two different tests to see if the two distribution were comparable or not:

- *T-Test* for the one who accepted H_0
- *Wilcoxon Rank Test* for the one who rejected H_0.

Both the function used returned a boolean value:

- 0: the two data behave the same way;
- 1: the two data behave in a different way;

To see if one is better than the other we compared the mean value through the use of boxplots.

3.0.4 Parametric Test: *T-Test*

To analyze the behaviour of the two mean values μ of the controller we plotted the result into some boxplots to see the mean value location and take a conclusion:

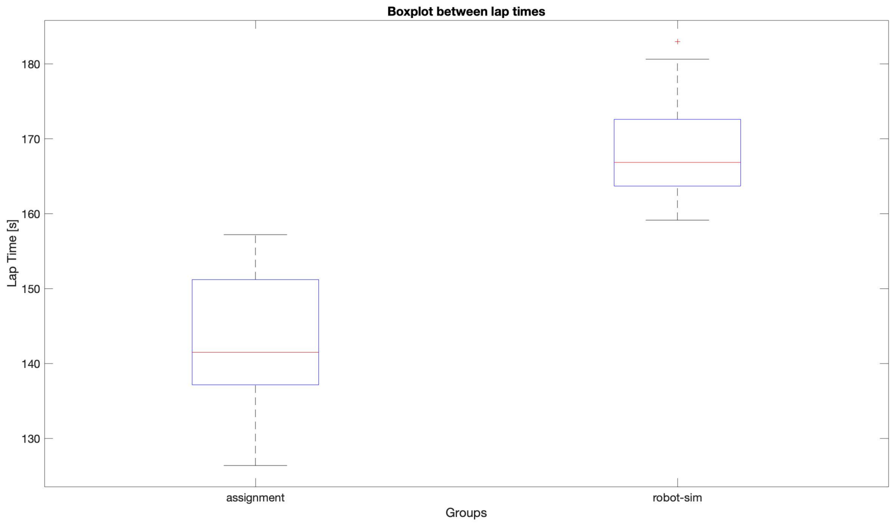

Figure 3.13: Boxplot of lap time for assignment and robot-sim controller in the first arena

3.0.5 Non-Parametric Test: *Wilcoxon Rank Test*

Since we have some data that, according to the Lilliefors test, do not belong to a normal distribution, we can still check the validity of our null hypothesis H_0 which states $\mu_{assignment} = \mu_{robot-sim}$.

The result we obtained is:

- Rejection of H_0 for lap time in Arena 2;
- Rejection of H_0 for distance travelled in Arena 1;
- Rejection of H_0 for distance travelled in Arena 2.

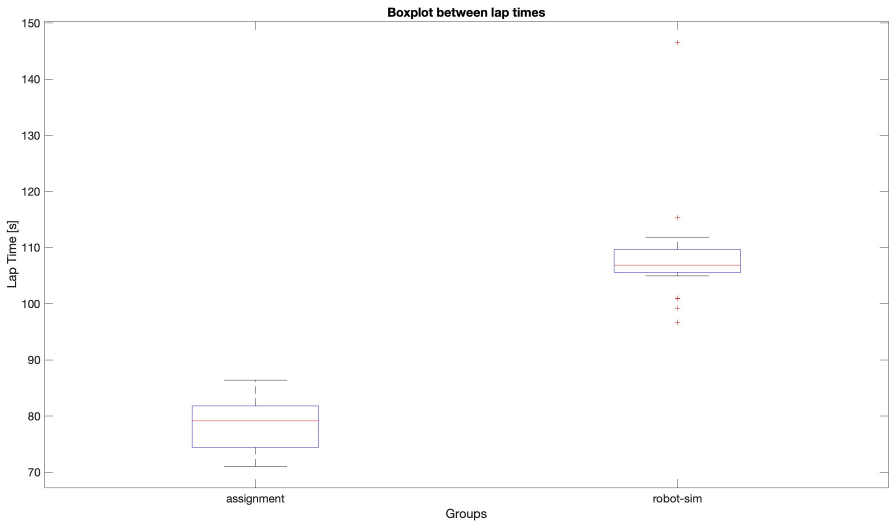

Figure 3.14: Boxplot of lap time for assignment and robot-sim controller in the second arena

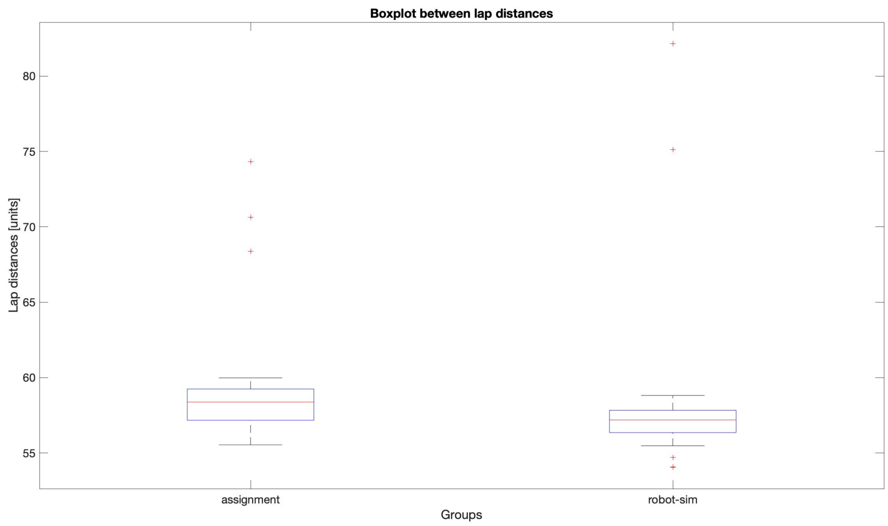

Figure 3.15: Boxplot of lap distance for assignment and robot-sim controller in the first arena

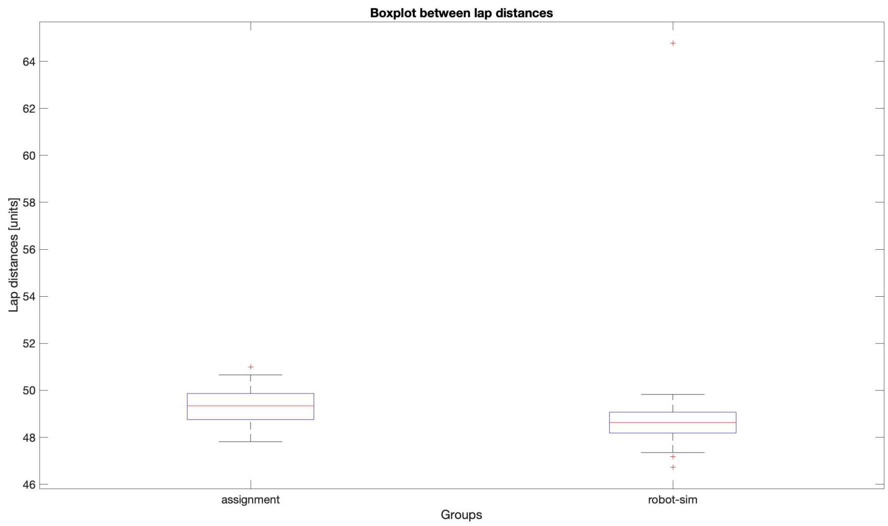

Figure 3.16: Boxplot of lap distance for assignment and robot-sim controller in the second arena

Comparison for the general algorithm

The same procedure adopted before has been used to detect which algorithm perform better in general by using 10 iterations with 10 different obstacle location.

We used the same approach for the parametric and non parametric test with the result of:

- Lap time:
  - T-Test
  - Rejection of H_0
- Lap distance:
  - Wilcoxon Rank Test
  - Acceptation of H_0

The resultant plots obtained are:

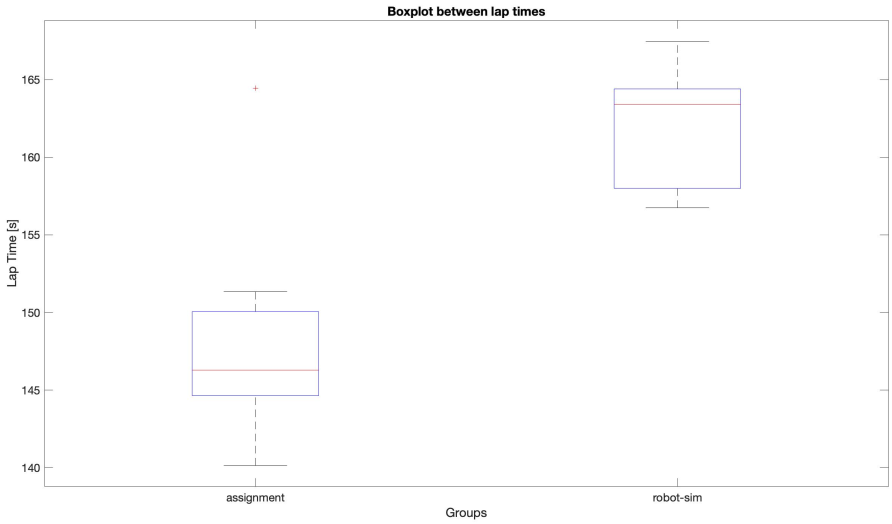

Figure 3.17: Boxplot of lap time for assignment and robot-sim controller in the general simulation

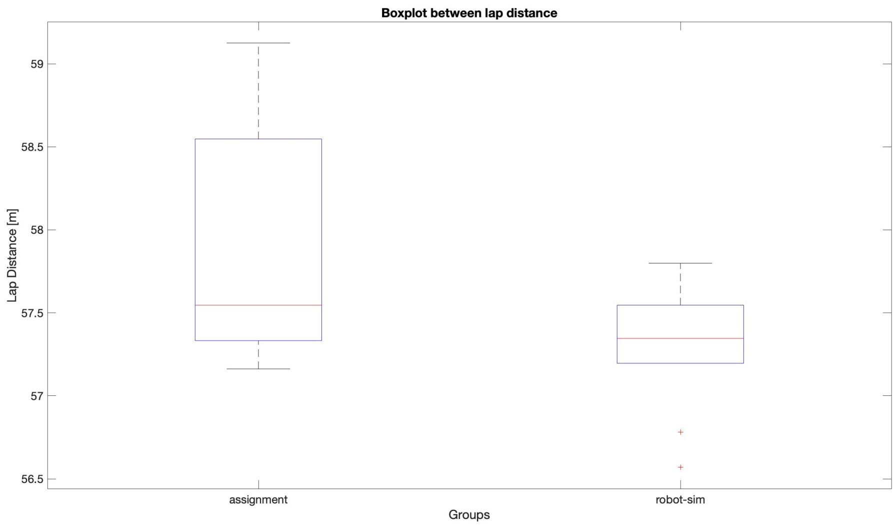

Figure 3.18: Boxplot of lap distance for assignment and robot-sim controller in the general simulation

Chapter 4

Conclusion

It is important to know that during the simulation and the acquiring data process the PC may not be totally dedicated to it and, according the its RAM and CPU usage there can be few errors: sometimes we had for example to stop the simulation and start it again because the robot hit a wall of turned too much around its z-axis or stopped in the middle of the circuit thus bringing errors in simulation and in the data.

The test can be expanded by introducing more statistical analysis for example the number of time the robot hits a wall of the number of time the robot chooses a wrong direction with respect to the total number of laps analysed.

Thanks to these tests we can state which controller resulted to be better.

Hypothesis verification

Since we started from a null hypothesis of the same greatness for both the controllers, in which we stated $\mu_{assignment} = \mu_{robot-sim}$, we can now check if it was correct.

Since we want to know which algorithm performs better in general we have to discuss the last result.

In this case we have:

- Acceptation of H_0 for the lap distance that means the two algorithms performs in the same way in terms of distance covered in the circuit;
- Rejection of H_0 for the lap time that means the two algorithm do not behave the same way; we can now check the μ value of the lap time in the different cases:

$$\mu_{assignment} < \mu_{robot-sim}$$

From this observation we can conclude that, in general, the *assignment*-algorithm performs in a better way with respect to the *robot-sim* one because they behave the same way in terms of distance covered but $\mu_{assignment} < \mu_{robot-sim}$ in terms of lap time.