MILESTONE 2 – COVER PAGE

Team Number: THURS-12

Please list full names and MacID's of all *present* Team Members

Full Name:	MacID:
AMREEN RAHMAN	rahmaa52
YUVRAJ SANDHU	sandhuy
SARAH YOUSSEF	yousss6
ZHIKAI ZHANG	zhanz377

MILESTONE 2 – PRELIMINARY DISCUSSION

Team Number: 12

1. Discuss what you think will happen when you execute your code for each scenario
   → Do you expect there to be any differences in performance between scenarios? Why or why not?
   → Do you expect one scenario to result in a faster navigation of the track?
   → Do you expect one scenario to fail (i.e., not successfully navigate the track)?

Time-Controlled Environment: If we control the time variable, the drone will go smoother because the drone will move at a predetermined time instead of moving through the track / turns with only the distance sensor. They should complete the track in the same amount of time as their speed would be the same in both experiments.
Distance-Controlled Environment: Since the code is relying on the information sent from the distance sensor, the robot would have a higher margin of error due to the code being limited to the values coming from the sensor. As well as the movement of the robot could be much more erratic and unpredictable compared to the time-controlled environment. Thus, this environment is much more likely to fail.

MILESTONE 2 – TIME-CONTROLLED SIMULATION (PLANNING)

Team Number: 12

Plan your computer program for the time-controlled scenario in the table below. The **track dimensions** can be found in Figure 2 of the P0 Project Module.

→ List one discrete motion per row. Examples are shown in the table (you should delete them before planning your own program.

Move forward for 2.285 seconds
Rotate -22.03 deg and move forward for 1.226 s
Rotate -11.015 deg and move forward for 1.261 s
Rotate -11.015 deg and move forward for 1.261 s
Rotate -11.015 deg and move forward for 1.261 s
Rotate -11.015 deg and move forward for 1.261 s
Rotate -11.015 deg and move forward for 1.261 s
Rotate –22.03 deg and move forward for 1.226s

MILESTONE 2 – TIME-CONTROLLED SIMULATION (EXECUTION)

Team Number: 12

Complete this worksheet *after* having written and executed your code for the first scenario (time-controlled simulation).

1. Copy-and-paste your Python code in the space below (***only*** your code)

```
my_qbot.forward(2.29)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(4.58)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
```

```
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(1.3355)
my_qbot.rotate(-11.05)
my_qbot.rotate(-11.05)
my_qbot.forward(2.29)
```

2. Does the Q-bot successfully navigate the track (Yes or No)?

Yes (Follows the Yellow Line)

3. If you answered **No** to Question 2, describe approximately when and where the Q-bot got stuck and/or hit something

4. If you answered **Yes** to Question 2, enter the number of attempts required to successfully navigate the track

For us to be able to successfully complete the track, we had 2 attempts using different methods, 1- attempting to finding arc length and 2- finding degree of arc but doing it the wrong way, after finally getting it right on our 3rd attempt using degree of the arc.

5. If you answered **Yes** to Question 2, enter the elapsed time (in seconds)

68.87 seconds

MILESTONE 2 – DISTANCE-CONTROLLED SIMULATION (PLANNING)

Team Number: 12

Plan your computer program for the distance-controlled scenario in the table below.

→ List one discrete motion per row. Examples are shown in the table (you should delete them before planning your own program.

Travel forward until its 0.5 m from the wall
Rotate –45.05 CW
Travel forward until its 0.5 m from the wall and rotate –40.5 CW
Travel Forward until its 0.35 m from the wall and then rotate –60.05 CW
Travel Forward until its 0.35 m from the wall and then rotate –30.05 CW
Travel Forward until its 0.45 m from the wall and then rotate –30.05 CW
Travel Forward until its 0.55 m from the wall and then rotate –45.5 CW
Travel Forward until its 0.25 m from the wall and then rotate –60.5 CW
Travel Forward until its 0.3 m from the wall and then rotate –45.5 CW
Travel Forward until its 0.9 m from the wall and then rotate –35.5 CW

MILESTONE 2 – DISTANCE-CONTROLLED SIMULATION (EXECUTION)

Team Number: 12

Complete this worksheet *after* having written and executed your code for the second scenario (distance-controlled simulation).

2. Copy-and-paste your Python code in the space below (and the code you typed in)

```
my_qbot.travel_forward(0.5)
my_qbot.rotate(-45.05)
my_qbot.travel_forward(0.5)
my_qbot.rotate(-40.05)
my_qbot.travel_forward(0.35)
my_qbot.rotate(-60.05)
my_qbot.travel_forward(0.35)
my_qbot.rotate(-30.05)
my_qbot.travel_forward(0.45)
my_qbot.rotate(-30.05)
my_qbot.travel_forward(0.55)
my_qbot.rotate(-45.05)
my_qbot.travel_forward(0.35)
my_qbot.rotate(-60.05)
my_qbot.travel_forward(0.25)
my_qbot.rotate(-45.05)
my_qbot.travel_forward(0.4)
my_qbot.rotate(-30.05)
my_qbot.travel_forward(0.85)
```

3. Does the Q-bot successfully navigate the track (Yes or No)?

Yes

4. If you answered **No** to Question 2, describe approximately when and where the Q-bot got stuck and/or hit something

6. If you answered **Yes** to Question 2, enter the number of attempts required to successfully navigate the track

13

7. If you answered **Yes** to Question 2, enter the elapsed time (in seconds)

45.4 Seconds

MILESTONE 2 – GROUP DISCUSSION

Team Number: 12

Discuss your findings in the spaces below

5. Discuss how the Q-bot performed in each scenario

Time-Controlled Environment: Performed perfectly and followed the coded instructions well by executing it exactly and following the yellow line track almost perfectly. Also ran smoother compared to the distance experiment as we predicted before executing the experiments.
Distance-Controlled Environment: Performed slower than the Time-Controlled environment and had a high margin of error in following the yellow line. Although the robot had a much smoother and time efficient path to complete the track.

6. Explain the reasons for any differences in Q-bot performance between each scenario

The performance of the Q-bot is smoother and more accurate in the time-controlled environment, because it is easier for the bot to reposition itself, rotate and move forward, which is the reason why the Q-bot perfectly follows the yellow line in the time-

controlled scenario. However, it is harder in the distance-controlled environment because the Q-bot is constantly measuring the distance to the wall, which results in errors. As well the distance-controlled Q-bot would not have a consistent path around the track despite the code being unchanged. It would occasionally be off-set from its designated path and it is likely due to the distance sensor receiving inaccurate data.

In addition, there is a 0.39 second delay in the distance-controlled environment.

***** Recognizing that the Q-labs environment is meant to simulate the real-world, all the real-world factors that would impact the Q-bot's movement as if it were being driven in your living room are part of the simulation *****