Bug Algorithm 1: Report

Angelo Espinoza

Centro de Investigación en Matemáticas. CIMAT, Zacatecas, Mexico

October 6, 2021

1 Introduction

The Bug Algorithm 1 (Bug1) [1] is a path-planning algorithm that assume local knowledge of the environment and a global goal.

Essentially, the Bug1 consists of three principal behaviours:

1. Move in a straight line towards goal (motion-to-goal)
2. Circumnavigate the obstacle and store the closest point to goal (boundary-following)
3. Return to the closest point and head towards goal again

The robot is assumed to be a point with perfect positioning and it can also measure the distance $d(x, y)$ between two points x and y.

Let $q_0^L = q_{\text{start}}$ and the m-line be the straight line segment that connects $q_i^L = q_{\text{goal}}$. If the robot encounters an obstacle, let q_1^H be the point where the robot first encounters an obstacle and call this point *hit point*. The robot circumnavigates the obstacle until it re-encounters the *hit point* q_1^H. After that, the robot decides the closest point to goal, by comparing with all the other ones. Therefore, it circumnavigates once more to that closest point and leaves. The exact point where it leaves is called *leave point*, denoted by q_1^L. In the exact moment that the robot leaves it heads towards the the goal and move straight to it again.

2 Development

2.1 The simulator

For the simulation of the algorithm, we have chosen the **RobotBASIC** [2] simulator. It is a 2-D simulator that counts with multiple features for the simulation of mobile robotics, such as bumper, infrared, and ultrasonic sensors, global positioning (GPS), among others. All of this, makes **RobotBASIC** suitable for the purposes of this simulation. It is worth to point out that it also counts with its own IDE to compile the code. As the name suggests, it is based on Visual Basic, however it implements its own features to make easier the job of working with robotics.

2.2 Constraints

Given that the robot is a differential robot and not a point, this represents constraints at the time of simulating it. For instance, at the moment of implementing the algorithm it was adapted to work with such robot. This implies that the robot will have to adapt to environment and instead of using the bumper sensors, it will use the ultrasonic sensors, otherwise it has been proved that the robot will tend to collide in the spike-shaped obstacles, since it gets too close to the obstacle at the time of turning around them.

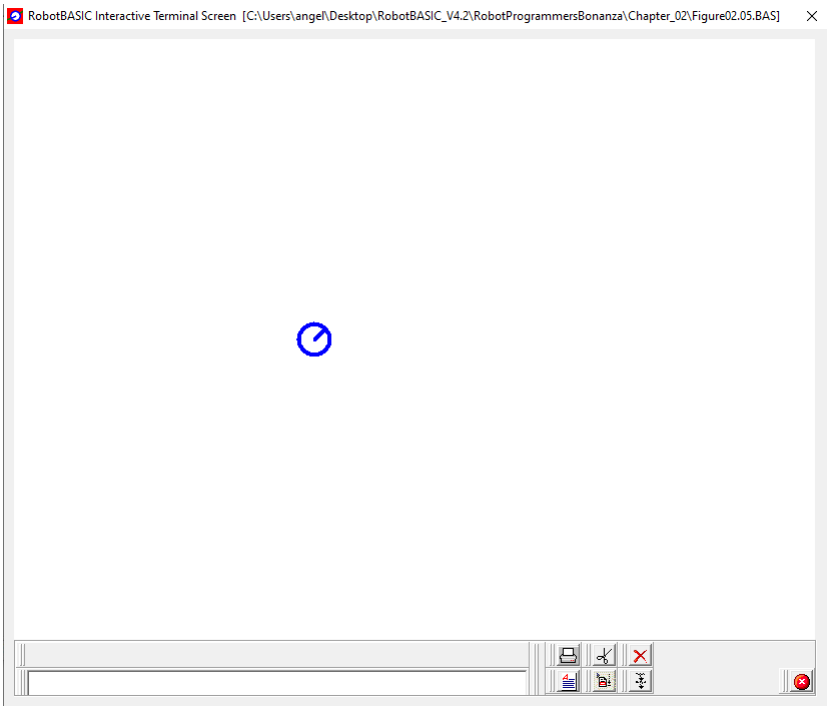

Figure 1: Robot positioned at 300 in x and 300 in y

2.3 Maps

RobotBASIC allows to the users to draw different environments in which a robot can navigate through, All of this, with the purpose of testing different behaviours of the robot in different worlds.

In this simulation, the robot has been tested on 4 different maps, there are 2 convex maps (Figure 2), and 2 non-convex maps (Figure 3).

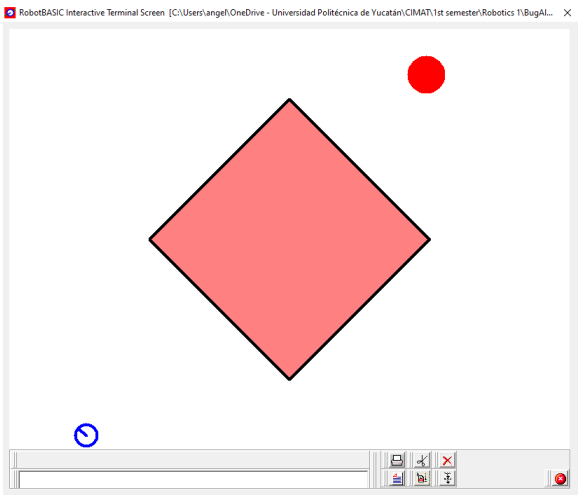

Figure 2: Convex map with the robot (blue) in an initial position and the target (red)

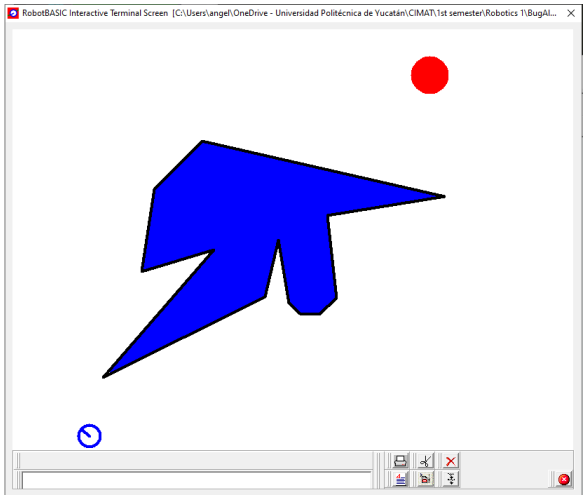

Figure 3: Non-convex map with the robot (blue) in an initial position and the target (red)

2.4 Implementation

All the code was written based on the algorithm presented in the book *Principles of Robot Motion: Theory, Algorithms, and Implementation* [1].

The results of the simulation based on the principal behaviours are presented in Figure 4, Figure 5, and Figure 6.

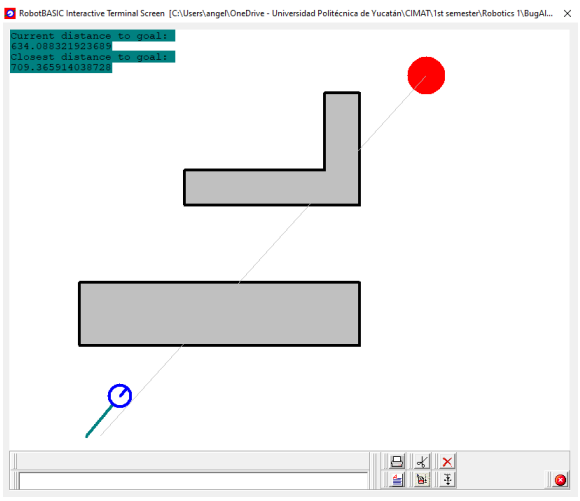

Figure 4: Robot performing motion-to-goal behaviour

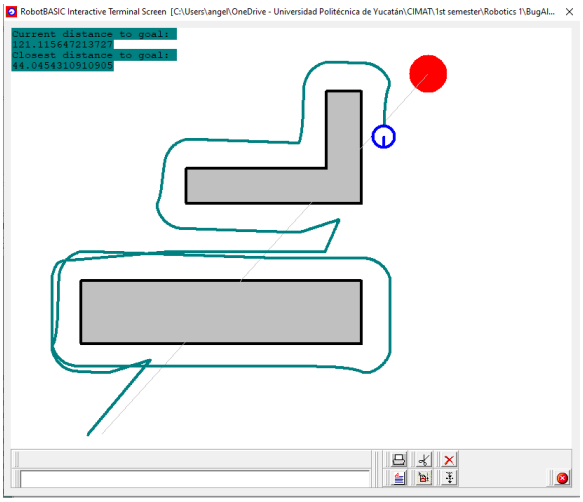

Figure 5: Robot performing boundary-following behaviour

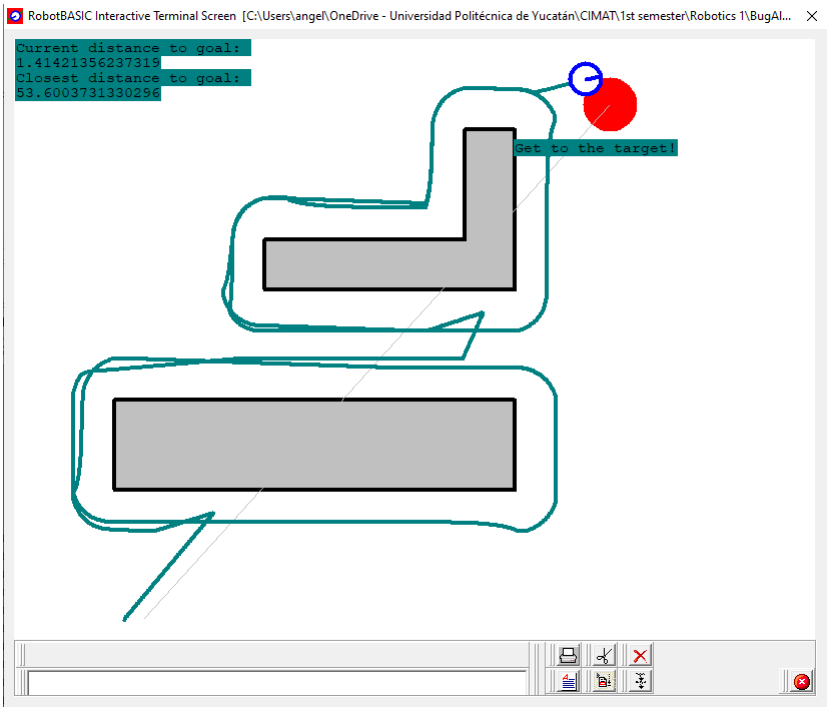

Figure 6: Robot successfully achieving the target

3 Conclusion

In this report, we have been able to show a basic motion planning algorithm (Bug1). We have also roughly shown the **RobotBASIC** simulator and proved that it is perfectly suitable for the simulation. Finally, we have implemented it on a 2-D simulator with working results.

References

[1] H. M. Choset and K. Lynch, *Principles of robot motion: Theory, algorithms, and implementation.* PHI Learning private Limited, 2005.

[2] J. Blankenship and S. Mishal, *Robot Programmer's Bonanza.* McGraw Hill, 2008.