Experimentation on the motion of an obstacle avoiding robot

Rakhmanov Ochilbek[1], Amina Sani[2], Nzurumike Obianuju[3] and Rukayya Umar[4]
Department of Computer Science,
Nile University of Nigeria,
Abuja, Nigeria.
Email: [1]ochilbek.rakhmanov@nileuniversity.edu.ng, [2]amina3sani@gmail.com,
[3]nzurumikeuju@yahoo.com, [4]umarrukayya114@gmail.com

ABSTRACT

An intelligent robot can be used for applications where a human is at significant risk like nuclear, space, military. The economics or menial nature of the application results in inefficient use of human workers in such areas as service industry and agriculture, for humanitarian uses where there is great risk e.g. demining an area of land mines, urban search and rescue. This paper implements an experiment on one of important fields of AI – Searching Algorithms -to find shortest possible solution by searching the produced tree. The authors concentrate on Hill Climbing Algorithm, which is one of the simplest searching algorithms in AI. This algorithm is one of the most suitable searching methods to help expert system to make decision at every state and at every node. The experimental robot traverses a maze by using sensors plugged on it. The robot used is EV3 Lego Mind-storms, with native software for programming LabView. This robot was chosen because it interacts quickly with sensors and can be reconstructed in many ways. This programmed robot calculates the best possibilities to find a way out of the maze. The maze is made of wood, and it is adjustable, as robot should be able to leave the maze in any design.

Keywords— Artificial Intelligence, Expert Systems, Robot, Obstacle Avoidance, Hill-Climbing Algorithm, Sensors

Reference Format:

Ochilbek, Rakhmanov; Sani, Amina; Obianuju, Nzurumike and Umar, Rukayya (2019), Experimentation on the Motion of an Obstacle Avoiding Robot,
Afr. J. MIS, Vol.1, Issue 3, pp. 24 - 29.

1. Introduction

As technology advances and smart devices become commonplace, Artificial Intelligence (AI) is becoming one of the most sought-after and most researched fields. Expert systems and smart machines are appearing in everyday products, with a variety of functionalities.

This paper implements an experiment on one of important fields of AI – Searching Algorithms, to find shortest possible solution by searching the produced tree. We will concentrate on Hill climbing algorithm, which is one of simplest searching algorithms in AI. This algorithm is one of most suitable searching methods to help expert system to make decision at every state, at every node. The experimental robot constructed and programmed by us, will traverse the maze by using sensors plugged on it.

The term Robot generally means a machine possessing human characteristics and being able to carry out the expected task programmed it to perform. Technically put by Rubin Murphy in his text book "an intelligent robot is mechanical creature which can function autonomously" and robotics is a branch of technology that deals with the design, construction, operation and application of robots (Robin, 2000). A robot function based on the algorithm used in programming it that is, performs the task assigned it to perform. The choice of algorithm is completely dependent on the type of function needed; lots of algorithms exist for different purposes like Binary search algorithm. A searching algorithm is widely used algorithm for route finding.

Most people think of robots in humanlike terms communicating and doing things like people would. However, these specific subsets of robots are actually not very common. A robot can be defined as a mechanical device that is capable of performing a variety of tasks on command or according to instructions programmed in advance (Govers, 2019).

With the forces of nature and the unpredictability of humanity at work, the world is in constant turmoil. The full power of nature can result in earthquakes, tsunamis and devastating storms. Humanity adds to this through war, terrorist attacks and unfortunate constant turmoil. These incidents require people to search for survivors and help remove them from the site of the incident. To search and locate survivors rescuers use various pieces of equipment (sounds poles, infrared cameras and sonars) as well as dogs. To reduce the risks to the rescuer, the search is carried out slowly and delicately but this has a direct impact on the time to locate survivors. The concept of the use of a robot to act for humans naturally lends itself to the area of searching hazardous or large environments in place of or sup-porting human searchers (Worrall, 2008).

An Obstacle Avoiding Robot is an intelligent robot, which can automatically sense and overcome obstacles on its path. Obstacle Avoidance Robot has a vast field of application. They can be used as services robots, for the purpose of household work and so many other indoor applications, they also have great importance in scientific exploration and emergency rescue, there may be places that are dangerous for humans or even impossible for humans to reach directly, then we should use robots to help us. In those challenging environments, the robots need to gather information about their surroundings to avoid obstacles (Pantech Solution).

AI involves using methods based on the intelligent behavior of humans and other animals to solve complex problems (Robin, 2000). In other words, it is a way of making a computer, a computer-controlled robot, or a software think intelligently, in the same manner that human think. Hill Climbing is a heuristic search used for mathematical optimization problems in the field of Artificial Intelligence. Given a large set of inputs and a good heuristic function, it tries to find a sufficiently good solution to the problem (van Putten, 2006).

The robot used is EV3 Lego Mind-storms. The reason we chose this robot is that it interacts quickly with sensors and can be reconstructed in many ways. To program the robot, we use LEGO MIND-STORMS Education EV3 1.4.0 software (Kevan, 2003; Joe and Evelyn, 2016; Leo, undated). This programmed robot calculates the best possibilities to find a way out of maze. The maze is made of wood, and it is adjustable, as robot should be able to leave the maze in any design.

There are many experiments and researches done in this field. For instance, Haissam and Hussein (2014) used both concepts of Hill climbing search algorithm and obstacle avoiding, with difference that they conducted experiment through simulation, while the experiment in the present paper is real life trial.

2. REVIEW OF RELATED LITERATURE

Since this work focuses on two aspects: (i) designing the robot and (ii) programing it using a search algorithm, our literature review focuses on the above mentioned aspects, LEGO robots and Hill Climbing searching algorithm.

Lego Robot has been used in different domain for different purposes like teaching. For instance, it can be used in teaching to increase students' engagement in learning (Renata and Robertas, 2013). Another work was done on disabled children where Lego robot was used in ascertain whether low costs robots provide a means by which disabled children can exhibit understanding of cognitive concepts (Cook et. al. , 2011).

Hill climbing algorithm is a local search algorithm that seeks to find optimal path from a starting position to a goal, it starts with a stochastic solution and climbs the hill by accepting better connecting solution than the one at hand the algorithm stops when it gets to either global maximum in the case of maximization or global minimum for Minimization (Haissam and Hussein, 2014; Mohammed and Asaju, 2017). A simple description for this algorithm was given in (Ben, 2004); hill climbing is an example of an informed search method because it uses information about the search space to search in a reasonably efficient manner. If one tries to climb a mountain in fog with an altimeter but no map, one might use the hill climbing to Generate and Test approach.

Haissam and Hussein (2014) used Hill climbing search algorithm in their work to implement computer simulated artificial intelligent agent that moves in 2D and 3D environments, a set of algorithms that helps the agent finds shortest path to a known target location while avoiding all obstacles. In addressing hill-climbing limitations, map generation algorithm was introduced which assigns costs to every location in the map. They concluded by stating that the algorithm has produced a complete and logical solution for the agent in 2D and 3D environments.

Dash et. al. (2012) proposed an algorithm for path planning of Robot with regards to the existing problem of path planning of Robot in the area of robotics. Their algorithm uses the Neural Network training technique. The output resulted in positive impact which is finding the next position of the robot is fast because of the neural network Technique and use of sensor system. The conclusion indicated implantation of the system is costly.

Liang (2011) relates to existing path planning of robot problem by proposing a path planning algorithm in which A* search calculates the grid from starting point to goal. This is in contrast to Dash et. al. (2012) where map planning algorithm was used to assign the cost to every location in the map.

Over the last few years, several other researchers have presented variety of approaches to solve the maze in different ways, using different methods and different sensors e.g. Van Putten (2006). In Kevan (2003), a robot to solve the maze using touch sensor was presented.

3. EXPERIMENTATION ON THE ROBOT

This section is presented in 4 parts; compiling the Lego robot, experimenting with Hill climbing algorithm, EV3 programming software and maze construction.

3.1 Assembly of the robot

All necessary parts of EV3 are listed in Figure 1 (Joe and Evelyn, 2016).

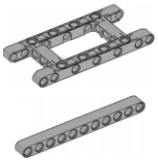
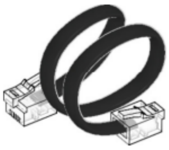
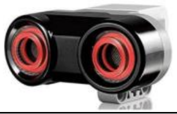
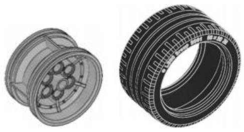
A- EV3 Brick: serves as the control center and power station for your robot	*B- Large Motor*: lets you program precise robotic action
C- Construction frames and beams 	D- Connection cables 
E- Ultrasonic sensor: Generates sound waves and reads their echoes to detect and measure distance from objects 	F -Tire design for robot 

Figure 1: parts of LEGO robot.

The robot in Figure 4 was assembled for our experiment. During the assembly, all necessary steps were followed as required (Carnegie Mellon's Robotics Academy, 2014). Various robot constructions can be found on LEGO magazines and the design of the robot can depend on imagination of the designer (Joe and Evelyn, 2016).

Figure 4 is labeled with letters, the parts from Figure 1. Briefly, A- EV3 brick is control center, and block program from Figure 5 is uploaded to A through computer. B- large motors - is attached to A with cables, D, and gets directions from A on when to run motor or not. This motor will make robot move with help of the wheel – F. So once the movement of the robot is arranged, it left to ultrasonic sensors, E, to sense the danger distance length decided by Figure 3, and send signal to A, to decide if robot should continue to move or further action is required. These further actions are presented in Figure 3 and Figure 5.

3.2 Hill Climbing

Figure 2 illustrates basically how the Hill climbing algorithm works, basically.

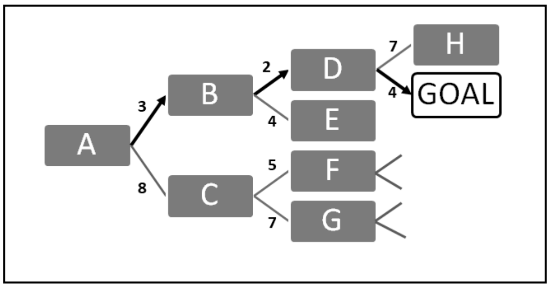
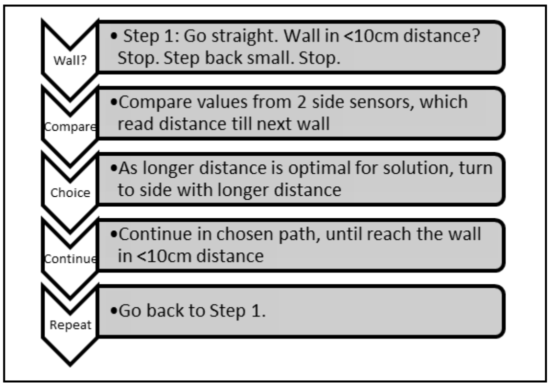


Figure 2: Searching for cheapest path

In our case, longer distance from wall is an optimal choice. Robot has 3 ultrasonic sensors (Figure 4); 1 in front- to sense obstacle in front, 2 by the sides, right left – they compare distance between the robot and the wall. Robot will go straight, when the gap is 10cm between robot and wall from front side, it will stop. Side sensors will compare distance to the respective walls, and robot will be ordered to continue to the side which has longer distance to the wall. Figure 3 illustrates the work of algorithm in robot. In this figure, robot needs to step back small from the wall, to take turn. We used word small, and this small distance we programmed 5cm length, but it may vary slightly, depending on robot position and ground friction.



- Step 1: Go straight. Wall in <10cm distance? Stop. Step back small. Stop.
- Compare values from 2 side sensors, which read distance till next wall
- As longer distance is optimal for solution, turn to side with longer distance
- Continue in chosen path, until reach the wall in <10cm distance
- Go back to Step 1.

Figure 3: Decision steps using Hill Climb algorithm

In this way it will jump to the node which is giving better option. When it comes to exit from one line to another, it will evaluate it again longer distance to next wall and will follow it. In this way, it will try to choose best options to arrive to optimal solution.

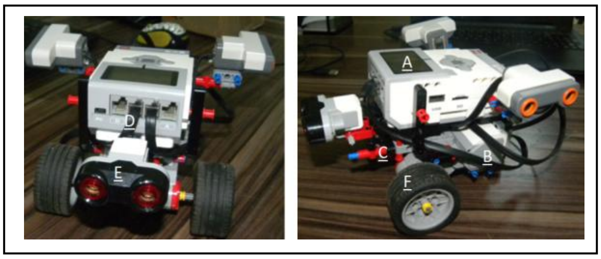


Figure 4: the assembled LEGO robot, ready for trial. Lettered parts are mentioned in Figure 1.

3.3 EV3 Programming

There are many ways to control EV3 robot, using well known languages; C, C++ and others. But to make it easier for education, LEGO preferred to develop a simple to use user interface, called Lab-VIEW (Joe and Evelyn, 2016).

This software can encode the robot using some simple blocks. At the back-end it uses C/C++ programming language. With object oriented programming, all directives are converted to some shapes, like in Figure 5. So user will just choose some blocks, to order robot what to do, and software will automatically program it in necessary way. Orders are basic, such as turn right or turn left, compare sensors etc. It is up to the user to program useful stuff so it can run smoothly. Figure 5 shows the implementation of algorithm, described in Figure 3, EV3 Lab View programming. Each part of the code is explained.

3.4 Maze design and construction

The definition of maze is a complicated system of paths or passages that people (robot in our case) try to find their way through, as it is described by Cambridge dictionary. Our maze is constructed from wood, with adjustable walls, so we can construct another maze. Size of board is proposed to be 120cm by 240cm. Road width will be 30cm. Height of walls 15cm to enable the robot sensor to adjust accordingly.

Figure 6 shows how a sample maze will be constructed:

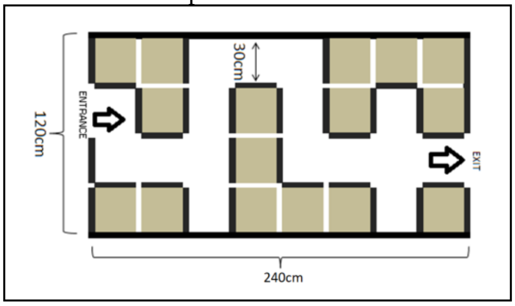


Figure 6: sample maze design

Complex designs are avoided as they can push robot for continuous loop, which need backtracking and cancellation of previous path, and this requires a different searching algorithm with more complex sensors. The main idea behind maze design was that robot should always have only 2 options to choose with one correct. Thus, in maze with intermediate level, finding longer path should bring you closer to solution.

For our experiment, we will try to use a maze with at least 2 right turns and 2 left turns, to understand if robot is choosing best option always. Figure 7 shows our real constructed maze for trial. Total length the robot is supposed to cover is 350cm

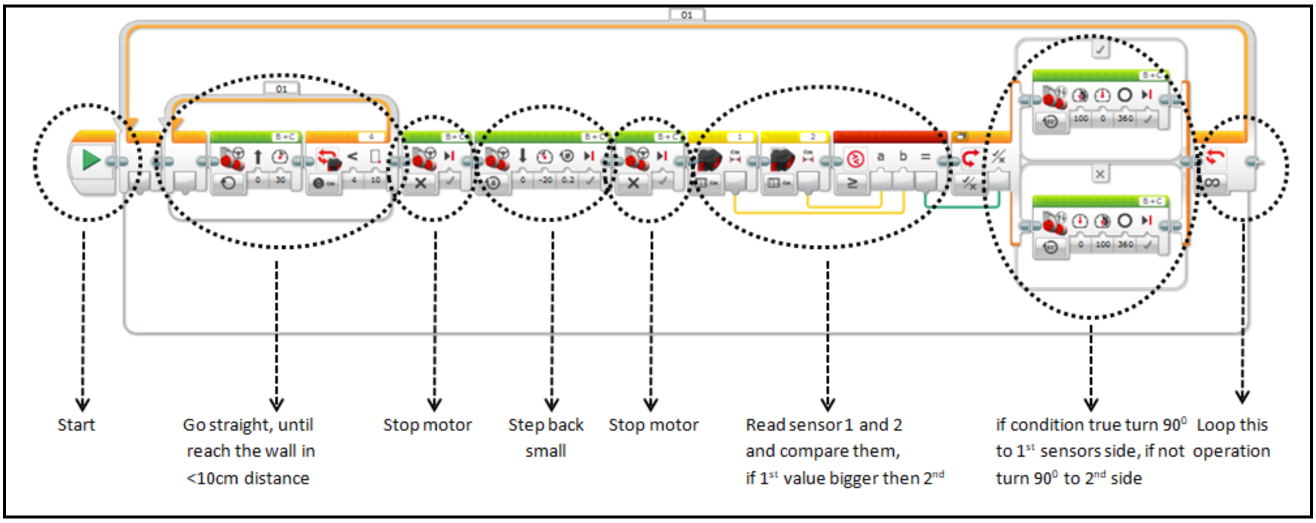

Start
Go straight, until reach the wall in <10cm distance
Stop motor
Step back small
Stop motor
Read sensor 1 and 2 and compare them, if 1st value bigger then 2nd
if condition true turn 90^0 to 1st sensors side, if not turn 90^0 to 2nd side
Loop this operation

Figure 5: Lab view block programming of Hill climb design, described in Figure 3.

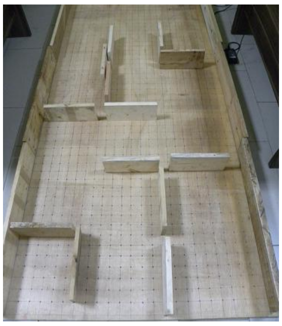


Figure 7: Wooden maze.

4. Results and Discussion

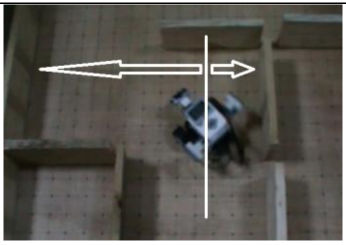
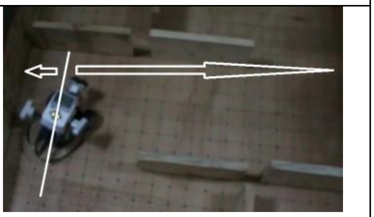
Robot turning to left, after deciding that distance to the left obstacle is longer than the distance to the right	
Robot turning to right, after deciding that distance to the right obstacle is longer than the distance to the left	

Figure 8: Pictures from trial

The total distance it covered was 3.5 meters, as we mentioned above. To help robot to make correct decisions, we programmed it to go slower. This can be done by adjusting speed of the robot during Lab View programming part, in Figure 5. Yet, number of turns is important as well. Table 1 shows results of first 4 trials. We didn't continue with trial as the fact that robot is traversing the maze was established.

№	Result	Reason
Trial 1	Failed at 2nd turn	Sensor misread
Trial 2	Failed at 4th turn	Incomplete turn
Trial 3	Successfully left maze	
Trial 4	Successfully left maze	

Table 1: Results of the trials

It turned 6 times in constructed maze. Total time needed to leave the maze was 37 seconds, in trial 3 and nearly same in trial 4. One can adjust it to go faster. But that is not our objective. Our main goal for this experiment was to see implementation of Hill climbing algorithm.

5. Conclussion

In this paper, the authors assembled an EV3 robot and program it to use Hill climbing algorithm to traverse the maze, with complexity level of intermediate. We can conclude that it was successful one, as we achieved the aim, our robot traversed the maze and found exit.

In education, it is well proven fact that experimental approaches usually leading to better understanding of the concepts and theorems. Observing visually, how the robot is reacting, helps to understand the way the algorithm works, with its advantage and disadvantages.

Comparing our experiment to previous research trials, we aimed the same goal, to tackle the problem, in other words, to make the robot to complete given job successfully. But as we mentioned before, the outline of this experiment was to implement theory of searching algorithm. Once the algorithm steps are programmed into the robot, it is easy to observe the advantage and disadvantage of the algorithm.

We should underline the fact that the maze level was intermediate (relatively not hard), so was the implementation of the searching algorithm. In other words, the in maze construction, we avoided the conditions like death end, backtracking, and others. If one wants to solve maze with higher difficulty, some of solutions can be using better searching algorithm and more effective robot construction. For instance, our sensors were fixed, but the robot may have dynamic sensor reader, to tackle more difficult maze types. In same manner, an algorithm with backtracking options would help in various conditions.

Acknowledgement

We would like to thank our supervisor Prof. Nwojo Agwu, Nile University of Nigeria, for his support and guidance during the experiment and paper preparation. The authors will also like to express appreciation to the anonymous reviewers whose comments helped to restructure the paper.

References

[1] Govers, Francis X. (2018). Artificial Intelligence for Robotics. *Packt Publish*, Chapter 1.

[2] Worrall, K. J. (2008). Guidance and search algorithms for mobile robots: application and analysis within the context of urban search and rescue. *University of Glasgow*, PhD thesis.

[3] van Putten, B. J. S. (2006). Design of a maze solving robot using Lego MINDSTORMS. *(DCT rapporten; Vol.2006.057). Eindhoven: Technische Universiteit Eindhoven.*

[4] Kevan, A. (2003). A Lego Mindstorms Maze Solving Robot. *University of Leeds, Master Thesis.*

[5] Pantech Solution. Obstacle avoidance robot. https://www.pantechsolutions.net/robotics-projects/obstacle-avoidance-robot (visited on 06/05/2019).

[6] Robin, M. (2000). Introduction to AI Robotics. *MIT Press*.

[7] Renata, B. & Robertas, D. (2013). Teaching of Computer Science Topics Using Meta-Programming-Based GLOs and LEGO. *Informatics in Education,* vol. Vol. 12, no. No. 1, pp. 125–142.

[8] Cook, A.M; Adams, K; Volden, J. and Harbottle, N. (2011). Using Lego Robots to Estimate Cognitive Ability in Children who have Severe Physical Disabilities. *Disabil Rehabil Assist Technol.* 2011;6(4):338-46. .

[9] Haissam E. & Hussein B. (2014). A simulated motion planning algorithm in 2D and 3D environment using Hill Climbing. *International Journal of Artificial Intelligence & Applications (IJAIA)*, Vol. 5, p. 1-.

[10] Mohammed A. & Asaju L. (2017). ß-Hill Climbing algorithm for Sudoku game. *2nd Proceedings of Palestinian International Conference on Information and Communication Technology*.

[11] Ben, C. (2004). Artificial Intelligence Illuminated. *Jones and Barlett Publishers.*

[12] Dash, T., Mishra, G. and Nayak, T. (2012). A Novel Approach for Intelligent Robot Path Planning. *Proceedings of National Conference in AIRES-*, Andhra University, p. 388.

[13] Liang, J. (2011). A path planning algorithm of mobile robot in known 3D environment. *Procedia Engineering*. 15:157-62.

[14] Joe, O. and Evelyn, L. (2016). LEGO Mindstorms EV3 Programming Basics. *Washington State Library, Library Development Team*.

[15] Lego. User Guide (Official release). *ENGB Version 1.3.2.* www.legoeducation.com/mindstorms (Last visited: 27/05/2019)

[16] *Carnegie Mellon's Robotics Academy* (2014). Introduction to Programming EV3. Carnegie Mellon's University. https://www.cmu.edu/roboticsacademy/PDFs/Curriculum/Intro-to-EV3/EV3-teachers-guideWEB.pdf. (Last visited: 27/05/2019)