Area mapping for autonomous disinfection robot

D M Anton[1], R C Milas[1], A M Stiube[1] and L E Stiube[1]

[1] Faculty of Managerial and Technological Engineering, University of Oradea, 410087 Oradea, Romania

E-mail: danielanton0227@gmail.com

Abstract. This article presents an autonomous obstacle avoidance robot with infrared sensors. With the help of these robots, the automotive industry has developed, using various types of sensors, both parking using ultrasonic sensor models, to Lidar sensors that map the immediate proximity of the car in real time. These sensors, through different trail simulations, adjusts the direction of the car to avoid various obstacles, or to avoid the following traffic jams.

1. Introduction

The concept of Artificial Intelligence emerged in 1956, at the Dartmouth Conference [1] organized by Marvin Minsky, Jhon McMarthy and two senior scientist, Claude Shannon and Nathan Rochester of IBM, this dream becoming a reality in recent years, getting more and more in every field. Artificial intelligence by definition states that it resolves new situations or problems based on experience, experience gained on the basis of continuous learning. Artificial Intelligence has begun to take shape in the field of computer science that develops technical systems to solve technical problems related to human intelligence. This artificial intelligence is based on acquired knowledge, and it has no predominant idea to limit it in thinking and developing new ideas based on previous ones [2].

Artificial Intelligence is used also for labyrinth robots, a family of devices that offers the benefits of all PIC18 microcontrollers, namely high performance, cost-effective computers with high-strength addition and flash memory. In addition to these features, the PIC18F2455, 2550, 4455, 4550 series introduces the enhancement design that makes these microcontrollers a logic for many high-performance, application-sensitive users.

2. Robot Maze architecture

This wiring diagram shows the connections of the motors, the sensors, the power supply of the control board, each pin separately.

In figure 1 is presented the electrical scheme of the Pixel Board, designed in Eagle.

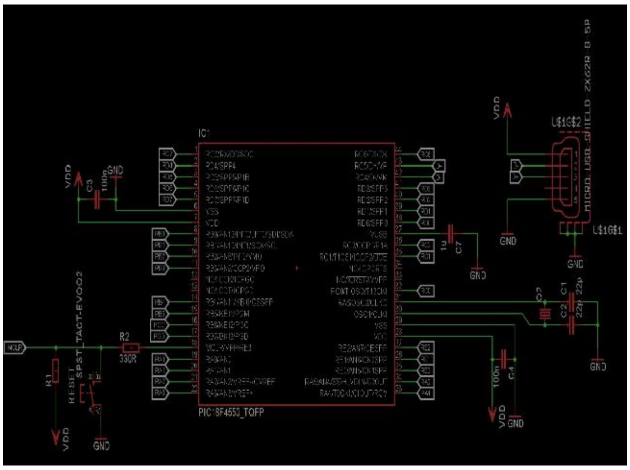

Figure 1. The electrical scheme of the Pixel Board route.

3. Motor driver and motor characteristics

To reduce the size and weight of the robot, it was used a L293D driver, which controls the robot motors at 1000 rpm at the output of the gearbox. This driver supplies two motors because the plate PixelBoard is not capable of delivering a lot of continuous power, so we attach a motor driver that feeds directly to the DC source. This high-frequency bridge is designed to provide bidirectional current up to 600mA at voltages from 4.5V to 36V. Devices are designed to drive inductive loads such as relays, solenoids, DC and bipolar motors, and other high voltage or voltage loads in positive consumption applications. The DC motors [4] of the robot are equipped with reducers, because at the output of the motor we have a speed of 30000 rpm and the torque is very small to move the robot, the transmission ratio is 30: 1, the free running speed is 1000 rotations per minute at 12V. Maximum torque is equivalent to 63.6 mNm. Stall current is reaching up to 800mA in load, no load current settling at 100mA, this being different depending on the weight robot and the frictional force that appear between the wheels and the surface they are running on.

Running the robot is done with the wheels on which the bicomponent silicon care is added and it is very elastic, with a fairly high resistance, if it is stretched, it does not return. Wheel models have an outer mold for the contour of the wheels, designed and printed on the 3D printer. Silicon casting is done with the syringe because it has to take the air out between the mold and the tire and the space required to be filled with silicone does not allow us to introduce silicone quickly because of its sizes.

The analogue distance sensors used are Sharp, with the perception of the distance from 10 cm to 80 cm. These reflective sensors show us the distance with accuracy up to about 0.4 cm in our robot up to a wall, usually these walls are white and the reflection rate is about 90%, but in some cases the walls are gray , this reflection decreases to about 20% because it absorbs part of the light emitted by the diode, and the sensors only receive some of this infrared light emitted by the diode. At the maximum reading threshold, there is a hysteresis of a few millimeters, at minimum the values are almost unreviewable. In figure 2 is presented the actual build of the maze robot, and in figure 3 the electrical scheme.

Figure 2. Maze robot.

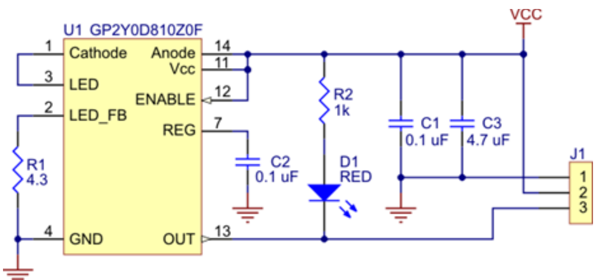

Figure 3. The electric sensor distance diagram. [Pololu]

4. Robot design and process of filament deposition

This chassis has been engineered in SolidWorks, where all the control electronics, sensors, motors, encoders, wheels and battery are mounted, as presented in figure 4.

The robot is made of heated 40 W power filament and a 24 V supply, it heats up to 220^0C, the filament is pushed by a stepper motor, and depending on the engine speeds the ones three axes (x, y, z) automatically adjust the pushing speed of the filament by the motor printer, and on the y-axis, the height of a single layer deposited from a passage of the nozzle extruder, which is a 0, 4 mm through filament care is extruded into layers of at least 0.1 mm high. In figure 5 is shown the scheme, designed in Eagle, with short caption (caption centred).

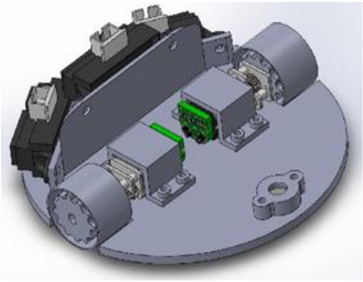

Figure 4. Design robot assembly.

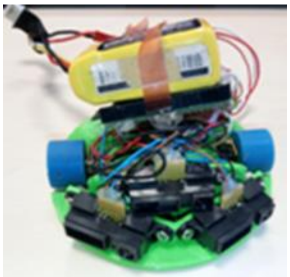

Figure 5. Figure with short caption.

5. Applying the Dijkstra algorithm

Schedule a test area to apply this algorithm based on known distances after completing the route and finding the shortest route by distance and time. Determining the matrix of walls (sqm) this matrix will contain all the walls that the robot will bypass, the starting point and the finish. The declaration of maturity will be done on rows and columns, as in figure 6, all later being the robot working environment.

The other algorithm, also called another engagement algorithm, is sometimes used with the left or right turn function, these being chosen from the beginning for the robot. This is used in order not to get into the problem in which the robot resumes the same route, the other Pledge algorithm is used, which will arbitrarily choose the chosen direction [5].

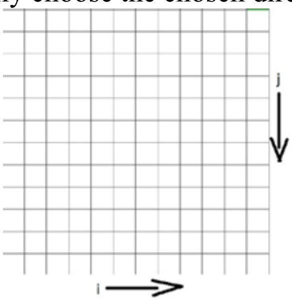

Figure 6. Defining rows and colonies.

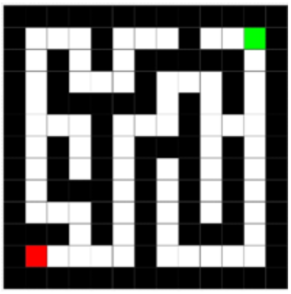

Figure 7. Labyrinth of the robot

The drawing of the interface is based on a matrix in which the walls to be circumnavigated are marked with 1, and with 0 is marked the route where the robot can go willingly to accomplish the task that was imposed at the beginning, namely to choose the optimal route. The starting point and the finish point, the outline of the labyrinth, and the robot's own labyrinth in which to operate all the distances, are set, to find the shortest route [6].

Pentru programarea robotului, pe baza acestui alt algoritm, a fost utilizată generarea de grafică statică, cu implementarea lor bazată pe cod, precum și coduri open source, care au fost ulterior modificate pentru robot [7,8,9].

Operation of the labyrinth, at each secondary artery blockage, returns when the artery breaks away from the main artery and continues the path until the robot reaches the end point.

In figure 8 and 9, the robot traverses the labyrinth completely to map the maze in order to get a precise map of it, in order to record all possible solution and finding the best suitable one.

At this point, the robot finds a second node from the first second and continues its route until it returns to the main road.

Figure 8. The first nodes traveled, to choose the route

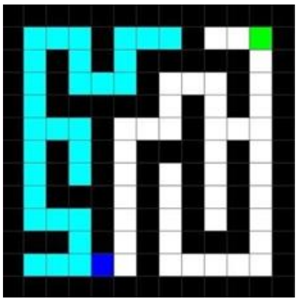

Figure 9. Go through a complete route to a dead end, and return to the last node

In figure 10, the robot has arrived at the desired point, noting the shortest trajectory. The optimal route is due to the two basic rules that make it always choose the shortest and quickest route.

Figure 10. Completion of the route with the shortest route

6. Conclusion

As a conclusion, the maze robot is collecting data from environment where it is placed and utilize these informations to generate an environment map. The algorithm is using the map to generate the optimum path, for the future to traverse all the spaces. This article is the basis for a for a work in progress PhD's work, named Prototipying an autonomous mobile robot family for UV-C neon disinfection. For future work, it is proposed to add fixed state LIDAR and rotative type LIDAR as well, for mapping the environment by generating a cloud dataset.

References

[1] J A Perez, F Deligianni, D Ravi and G Z Yang 2017 *Artificial intelligence and robotics* EPSRC UK-RAS Network

[2] M Singh, R Kumar, V Giradkar, P Bhole and M Kumari 2016 Artificially intelligent maze solver robot *International Research Journal of Engineering and Technology (IRJET)*

[3] E H Kivelevitch and K Cohen 2010 Multi-agent maze exploration *Journal of Aerospace Computing Information and Communication*

[4] https://www.pololu.com/

[5] https://en.wikipedia.org/wiki/Maze-solving_algorithm

[6] https://www.programiz.com/dsa/dijkstra-algorithm

[7] https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/

[8] https://cp-algorithms.com/graph/dijkstra.html

[9] https://github.com/Auone31/Dijkstras_Algorithm