THE MOVING OF A ROBOT IN THE LABYRINTH

Ilie Borcoși, *„Constantin Brancusi" University of Targu Jiu, Romania*

Abstract: In this paper is described the how a robot moves in a labyrinth, and also the possibility of finding the way to the exit of the labyrinth.

Keywords: robot, labyrinth, route finding, crossing labyrinth

1. Introduction

The utility of robots is much known nowadays. They are used in many areas. They are industrial, military, miners and medical robots. Without the help of robots some activities would be impossible, for example accuracy laser operations [1].

Execution of industrial applications where robots are used introduces next particularities [2]:

- The robots must replace people in dangerous activities;
- The robots must replace people in activities that they can't or won't execute them;
- Robots should replace people in the activities which they can perform more economically.

Although at the moment most robots are controlled directly by humans or they are operating in a static environment, there is an interest from humans which becoming more pronounced for the robots which can operate at a high level of autonomy in a dynamic environment.

So, in the last decade, robotics research passed from the stationary robotic systems to autonomous mobile robots. On the basis of recent progress in the field of fundamental algorithms for mapping, navigation and perception robots, the mobile robots are ready to be used in difficult environments [3].

These robots need a hardware and software navigation system in order to guide themselves in a dynamic environment.

The mechanical structure of a robot must controlled for it perform its tasks. Control of a robot involves three distinct phases: perception, processing/cognition and action [4].

Perception is done using sensors which take information regarding environmental and robot (for example position of its elements, the speed, the acceleration). This information is then processed to be „understood", stored and/or used then in developing action, to calculate the correct answer. This answer is sent as an electrical signal towards motors / actuators which move the mechanical components.

A robot would be useless if it can't do anything: to move, to transform the environment, to explore, that means perform a particular action.

Phase information processing from sensors can have varied complexity. This may include data translation from the sensor in commands to the motors. Sometimes it is necessary to filter the useful information received by sensors from the perceived noise. For this purpose can be used sensory fusion. Sensory fusion is a new technology that involves combining signals from multiple sources of sensory for the requested information to be precise and unique as in the case of the human eye [5]. An example would be stereoscopic view by using two cameras mounted at a certain distance to simulate the view into space. Phase processing / cognition can be removed in some cases and get teleoperation robots (operation remote by humans).

2. Problem formulation

Evidencing the possibility of movement of a robot in an environment with obstacles will do next by crossing a linear maze using the shortest way (optimizing by the route through the maze). So, it shows the ability of the robot

avoid obstacles, to adapt to the environment and to operate in an efficient mode.
Linear labyrinth, in the case of robots, is a tortuous route with many intersections (and many possibilities to cross), usually built of a black line on a white or light colored background. The robot must follow the lines maze and to find the shortest route between an entry point "START" and an exit point "FINISH" from the labyrinth (fig. 1).

Application of the rule left or right hand is done considering that:

- The robot will always prefer to take the left (or right) instead to go straight ahead when this is possible;
- The robot will always prefer to go straight instead to do the right (or left);
- The robot will be faced with eight distinct situations during crossing labyrinth (fig. 2).

Moving by line of the robot is made using 8 sensors in-line (fig. 3). Permanently, sensors in the middle will be above the line. The robot has

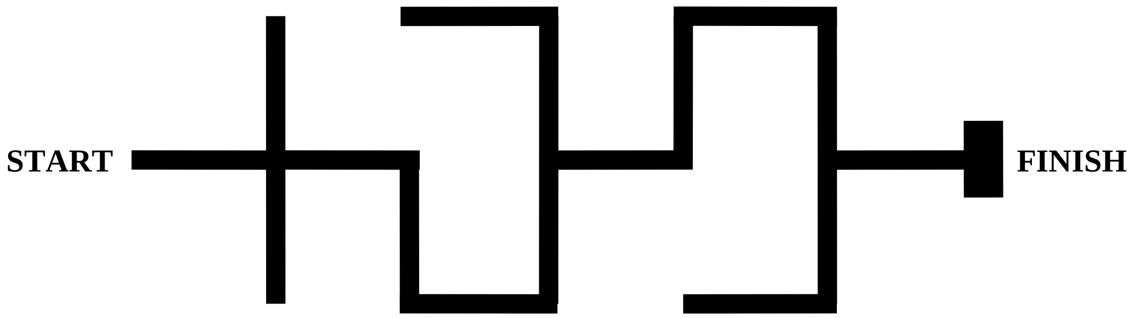

Figure 1.

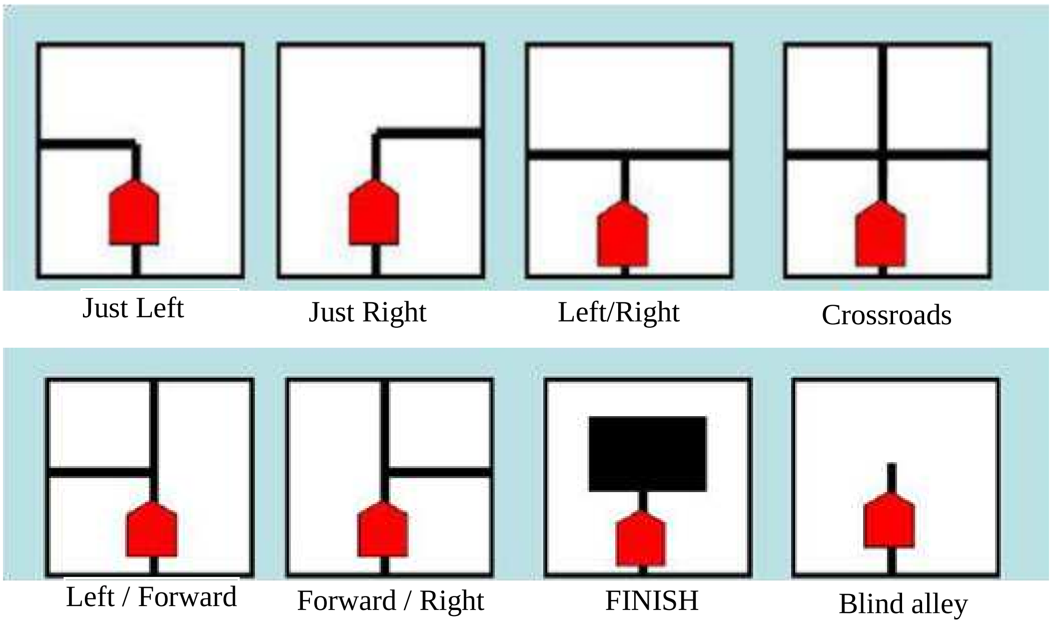

Figure 2.

2. Problem formulation

The completion of labyrinth is made by applying the left or right hand rule and the robot can reach blind alley (when the route is not continued).

two wheels, each of them operated by one driver and the third wheel is only for support.
If it is considered that line sensors provide logical 1(they are active) when they are above the line, then character string provided by sensors will be:

- 1000.0000, 1100.0000 = Line far to the left

- 0110.0000 = Line close to the left
- 0011.0000 = Line very close to the left
- 0001.1000 = The center line
- 0000.1100 = Line very close to the right
- 0000.0110 = Line very close to the right
- 0000.0011, 0000.0001 = Line far to the right
- 1111.1111 = T-type or X-type intersection
- 1111.1000, 1111.0000, 1110.0000 = Move left
- 0001.1111, 0000.1111, 0000.0111 = Move right
- 0011.1100 = Finish of labyrinth
- 0000.0000 = Blind alley
- 1011.1100, 1001.1001, etc = Combinations that should not appear

Generally, the robot will perform next actions:

- Follow the line, searching for the next intersection

- When you encounter an intersection, decide what kind of intersection is

- After which he determined what type of intersection met, depending on the method applied, it will decide which direction to follow.

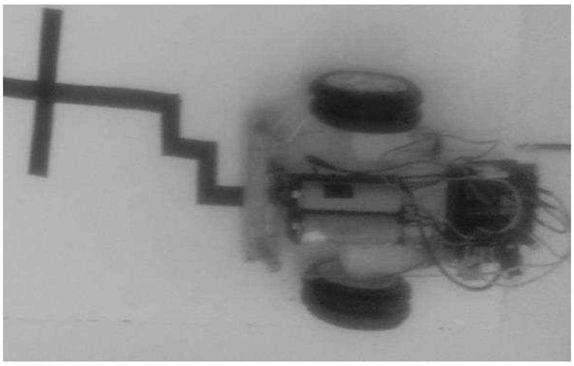

Figure 3.

These steps are repeated until the robot will detect the exit of the labyrinth.

The robot will turn "slow", "medium" or "fast" depending on position the line followed: "very close", "near" or "far".

The robot can encounter difficulties when it reaches a crossroad (when the robot has more than one option for direction) and it will need to make the right decision to continue the route depending on the type of intersection and the method used (rule of left/right hand).

The robot that it has reached an intersection T-type or X-type (cross type) when all sensors are activated (equivalent to the string 1111.1111). To distinguish the two intersections, robot will activate a subroutine "STEP" which allows moving robot in front one step (few millimeters/centimeters depending on route width). If notices that only the two middle sensors are active (they have value logical 1), then it is an intersection X-type (cross type), otherwise (when they have logic value 0) the intersection is T-type. Then the robot will move back one step, using the same subroutine.

If sensors in the left/right are active it make a turn left/right or an intersection left/right and then it forward. To recognize if it has to make a turn or cross an intersection robot activates subroutine "STEP" and it will read information from sensors. If it senses that only the two middle sensors are active, then it is an intersection left / right and must forward. Otherwise (when they have logic value 0), robot will have to make a turn left / right after it will move back one step, using the same subroutine.

To find the shortest route for traversing labyrinth, robot must complete it twice. At first crossing will take some wrong decisions, but it will save them and avoid them in the second round.

For completing the labyrinth of figure 1 is applied the right hand rule. If the robot starts from point "START", scroll algorithm is described below. It is considered that every time when the robot reaches a blind alley, this means that it previously turned wrong and must be changed and stored way that at the second crossing robot "knows" and avoid those sections of the route.

- After point "START" robot meets the first intersection such as "Left / Right / Forward" (information 1111.1111 from the eight sensors is read). Applying the right hand rule robot is forced to go to the right. Robot control software stores the fact that it went to the right with the letter R (Right).
- The robot reaches at the end of the line or at a blind alley (information 0000.0000 is read) and it is forced to turn (180o). This maneuver is stored with „U”, because the robot performed a U-turn type. At this moment, the robot's memory contains information RU.
- After it returned, it will arrives again at the crossroads, but this time it will turn right and will store this information.
- Because of the turn back (type U), the last movement takes the robot to a route blocked. The route chosen is wrong: instead be turned to the right in the intersection, the robot should have gone in front / forward (F). Can establish a rule: every time when are stored

the sequence of movements "Right - U Turn - Right" (RUR), will replace memory of this sequence with the movement "Forward" (F).

- The next turning is "just right". But this is not an intersection in according to previous definitions whereas the robot can go only one way and it is not necessary to memorize this turn.
- Next intersection is a "Right / Forward". Applying the right hand rule, the robot will necessarily turn to the right and will store this by the letter R. **FR** is the current memory status.
- Next intersection is a "Left/ Forward". In this case, the robot will move forward and will store this with F. **FRF** is the current memory status.
- Further, the robot will reach at a blind alley and will perform a U-turn type. **FRFU** is the current memory status.
- Once again it reached the intersection, the robot will do the right according to the rule. **FRFUR** is the current memory status.
- Similar to RUR turn, FUR turn is wrong, because it includes a blind alley. So, we will replace it with L, because at the second crossing labyrinth, the robot will turn left. It will be erased from memory FUR and will be replaced by L.
- Thus, the robot came out of the labyrinth exit "FINISH", having memorized the sequence of letters FRL. It will use this information from memory to perform the shortest route at the second or next passes it through the labyrinth.

4. Conclusions

The algorithm has the advantage of enabling optimal navigation of the labyrinth from the second pass through the labyrinth.

A disadvantage of this algorithm is the fact that can't be used in an environment in which objects or the route changes. Navigating such an environment requires a much larger number of sensors, like sensors very complicated and algorithms required for an environment „in transit" which, at it turn is exponentially more complex, of field of artificial intelligence.

REFERENCES

[1].http://www.puterea.ro/special/litiaza-renala-operatie-clasica-vs-operatii-realizate-de-roboti-da-vinci-si-avicenna-robotii-care-salveaza-vieti-117917.html

[2]. www.robotics.ucv.ro/ROBOTI.INDUSTRIALI/Domenii%20de%20aplicatie%20...

[3]. Boboc, Interacțiunea naturală om-robot pentru aplicații de robotică asistivă, Teza de doctorat, Universitatea Transilvania din Brasov.

[4].https://www.cs.cmu.edu/~mihaib/articole/roboti/roboti-html.html

[5]. http://oftalmix.ro/2016/03/01/strabismul-pentru-medici-strabismul-2/

[6]. https://en.wikipedia.org/wiki/Ismail_al-Jazari

[7].https://en.wikipedia.org/wiki/Karel_%C4%8Capek

[8].https://en.wikipedia.org/wiki/KUKA

[9].https://en.wikipedia.org/wiki/Liezi

[10]https://en.wikipedia.org/wiki/Leonardo%27s_robot