Navigation of PIC based Mobile Robot using Path Planning Algorithm

Sangmeshwar S. Kendre
Lecturer, E&TC Dept.,
TSSM's Padmabhooshan Vasantdada
Patil Institute of Technology,
Pune, India

Pallavi V. Mulmule
Lecturer, E&TC Dept.,
TSSM's Padmabhooshan Vasantdada
Patil Institute of Technology,
Pune, India

Abhijeet Shinde
UG-Student, Computer Dept.,
TSSM's Padmabhooshan Vasantdada
Patil Institute of Technology,
Pune, India

ABSTRACT

A robot is usually an electro-mechanical machine which is guided by computer or electronic programming. Conventional line tracer robots follow path in given map. In some kind of robots path is already store in its memory and it simply follows that path. When such types of robots are left to traverse through any random maze, these robots tend to traverse all possible paths every time having no facility to remember the right path. In our robot we are providing it with the capability to traverse through any random maze and remember the right path. Thus when robot travels through the same maze again it knows which is the right path to reach destination. Also it can share this information with its other counterparts in the swarm of robots. This project depends on local path-planning algorithm using a human's heuristic and a laser range finder which has an excellent resolution with respect to angular and distance measurements is presented for real-time navigation of a mobile robot.

General Terms

Navigation of Robot, Automation of Robot et. al.

Keywords

IR Sensor; PIC; Heuristic algorithm; DC Motor; Encoder; Decoder; MPLAB IDE.

1. INTRODUCTION

Research in mobile robotics can be traced back to late 1940s, although most of the effort related to path planning is more recent and has been conducted during the 1980s. Thanks to such fields, such as artificial intelligence, mathematics, computer science and mechanical engineering that theoretical and practical understanding of some issues has received a major boost. Planning refers to a preconceived scheme or method of acting or proceeding. In other words, we can say that it defines an operative intelligence. In this algorithm, path planning is done with respect to a mobile robot in order to design or scheme its routing.

In recent work as we have seen conventional line tracer robots before, the principle followed by them is to follow lines in given map. In some kind of robots path is already store in its memory and it simply follows that path. When such types of robots are left to traverse through any random maze, these robots tend to traverse all possible paths every time having no facility to remember the right path.

What we are doing in our robot is that we are providing it with the capability to traverse through any random maze and remember the right path. Thus when robot travels through the same maze again it knows which is the right path to reach destination. Also it can share this information with its other counterparts. This project depends on the heuristic (experience-based techniques that help in problem solving) search algorithm and artificial intelligence. Our algorithm also uses TURING MACHINE's principle.

A Turing machine is a theoretical device that manipulates symbols contained on a strip of tape. Despite its simplicity, a Turing machine can be adapted to simulate the logic of any computer algorithm, and is particularly useful in explaining the functions of a CPU inside of a computer. The Turing machine was described in 1936 by Alan Turing. Turing machines are not intended as a practical computing technology, but rather as a thought experiment representing a computing machine (thus they have never actually been constructed). They help computer scientists understand the limits of mechanical computation. The Turing machine mathematically models a machine that mechanically operates on a tape on which symbols are written which it can read and write one at a time using a tape head.

2. GENERAL DESCRIPTION OF THE DESIGNED EMBEDDED ROBOT SYSTEM

The designed system is as shown in figure1 and developed around the following components:

1. PIC Microcontrollers
2. IR Sensors
3. Motor Drivers
4. DC Motors
5. Encoder / Decoder
6. Buzzer

The system model consists of two robots. As shown in figure various 7 IR sensors scan the path and send appropriate color code to Board. PIC Board takes decision depending on signals from sensors and controls peripherals like actuators (DC motors), RF module, buzzer. DC motors are used for the locomotion of robot. Once robot arrives at destination PIC Board deactivates actuators and starts sending correct path with the help of RF module to other robot. Simultaneously, buzzer plays victory song.

2.1 PIC Microcontroller

PIC18 microcontroller offers the advantages namely, high computational performance at an economical price – with the addition of high-endurance, Enhanced Flash program memory. On top of these features, the PIC18F family introduces design enhancements that make these microcontrollers a logical choice for many high-performances, power sensitive applications.

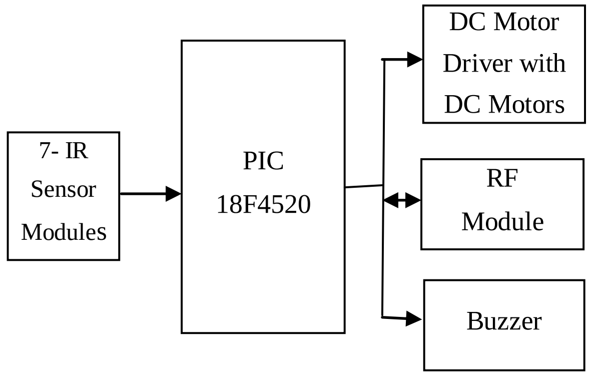


Fig.1. Robots (Master & Slave) Block Diagram

2.2 IR Sensor Modules

The QRD1114 reflective sensor consists of an infrared emitting diode and an NPN silicon phototransistor mounted side by side in a black plastic housing. The on axis radiation of the emitter and the on-axis response of the detector are both perpendicular to the face of the QRD1114. The phototransistor responds to radiation emitted from the diode only when a reflective object or surface is in the field of view of the detector. Monitoring and controlling of the system involves sensing the changes in the reflected IR light according to the color of the surface.

2.3 Motor Driver L293D

L293 and L293D are quadruple high-current half-H drivers. The L293D is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

2.4 ASK Transmitter

The ST-TX01-ASK is an ASK Hybrid transmitter module. ST-TX01-ASK is designed by the Saw Resonator, with an effective low cost, small size, and simple-to-use for designing. Transmitter Module is used send data to another robot.

2.5 ASK Receiver

The ST-RX02-ASK is an ASK Hybrid receiver module. An effective low cost solution for using at 315/433.92 MHZ. The circuit shape of ST-RX02-ASK is L/C. Receiver is used to receive data transmitted by another robot.

2.6 Encoder

The HT 12E Encoder ICs are series of CMOS LSIs for Remote Control system applications. They are capable of Encoding 12 bit of information which consists of N address bits and 12-N data bits. Each address/data input is externally trinary programmable if bonded out.

2.7 Decoder

The HT 12D ICs are series of CMOS LSIs for remote control system applications. These ICs are paired with each other. For proper operation a pair of encoder/decoder with the same number of address and data format should be selected. The Decoder receive the serial address and data from its corresponding decoder, transmitted by a carrier using an RF transmission medium and gives output to the output pins after processing the data.

3. ALGORITHM

In computer science, a heuristic algorithm, or simply a heuristic, is an algorithm that is able to produce an acceptable solution to a problem in many practical scenarios, in the fashion of a general heuristic, but for which there is no formal proof of its correctness. Alternatively, it may be correct, but may not be proven to produce an optimal solution, or to use reasonable resources. Heuristics are typically used when there is no known method to find an optimal solution, under the given constraints (of time, space etc.) or at all.

Two fundamental goals in computer science are finding algorithms with provably good run times and with provably good or optimal solution quality. A heuristic is an algorithm that abandons one or both of these goals; for example, it usually finds pretty good solutions, but there is no proof the solutions could not get arbitrarily bad; or it usually runs reasonably quickly, but there is no argument that this will always be the case.

For instance, say you are packing odd-shaped items into a box. Finding a perfect solution is a hard problem: there is no known way to do it that is significantly faster than trying every possible way of packing them. What most people do, then, is "put the largest items in first, then fit the smaller items into the spaces left around them." This will not necessarily be perfect packing, but it will usually give a packing that is pretty good. It is an example of a heuristic solution.

System based on following heuristic search algorithm:

```
1. Initialize the variables
        For i=0 to 50 do
            Final_path[i] ='*'
        End For
    Dead_end=0
    Diff=0
    Found=0
    i=0
2. If  (FRONTS=Destination_Color)Then
            Turn on the sounds and play music.
            Found=1
            Goto Step -7
3. Else if (LEFTS=true or RIGHTS=true) then
    If Dead_end=0 then
            If LEFTS=true then
            Turn the right DC motor on
            Turn the left DC motor off
            Final_path [i++] =L
    Else if FRONTS=true then
            Turn the right and left DC motors on
            Final_path [i++] =S
    Else if RIGHTS=true then
            Turn the left DC motor on
            Turn the right DC motor off
            Final_path [i++] =R
    Else
            Dead_end=0
            If LEFTS=true then
                    Turn the right DC motor on
                    Turn the left DC motor off
            If Final_path [i-1] =L then
                    Final_path [--i] =S
                    i++
            Else if Final_path [i-1] =S then
                    Final_path [--i] =R
                    i++
            Else if Final_path [i-1] =R then
                    Final_path [--i] ='*'
```

```
                Dead_end=1
        End if

        Else if FRONTS=true then
        Turn the right and left DC motors on
                If Final_path [i-1] =L then
                Final_path [--i] =R
                i++
        Else if Final_path [i-1] =S then
                Final_path [--i] ='*'
                Dead_end=1
        End if
        Else
                Turn the right DC motor off
                Turn the left DC motor on
                Final_path [--i] ='*'
                Dead_end=1
        End if
    End if
4. Else if (FRONTS=true)then
    Turn on the both RIGHT and LEFT DC motors on
5. Else if (LEFTS=false and Right=FALSE and
    FRONTS=false)
        Dead_end=1
        REVERSE TURN OF THE ROBOT
   End if
6. If Found=1 then
        Goto step-7
    Else
        Goto step-2
    End if
7. Transfer  Final_path[] to other Robots
8.  Stop
```

4. DEVELOPING ENVIRONMENT AND RESULTS

4.1 MPLAB IDE

MPLAB IDE is a Windows® Operating System (OS) software program that runs on a PC to develop applications for Microchip microcontrollers and digital signal controllers. It is called an Integrated Development Environment, or IDE, because it provides a single integrated "environment" to develop code for embedded microcontrollers as shown in Fig.2.

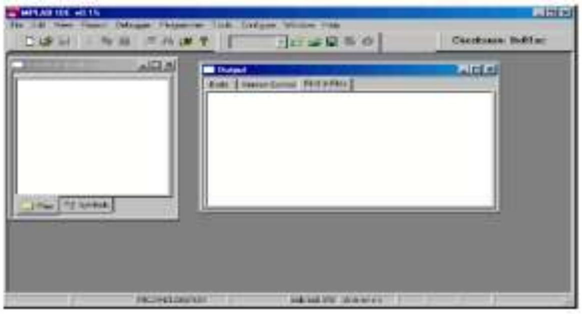
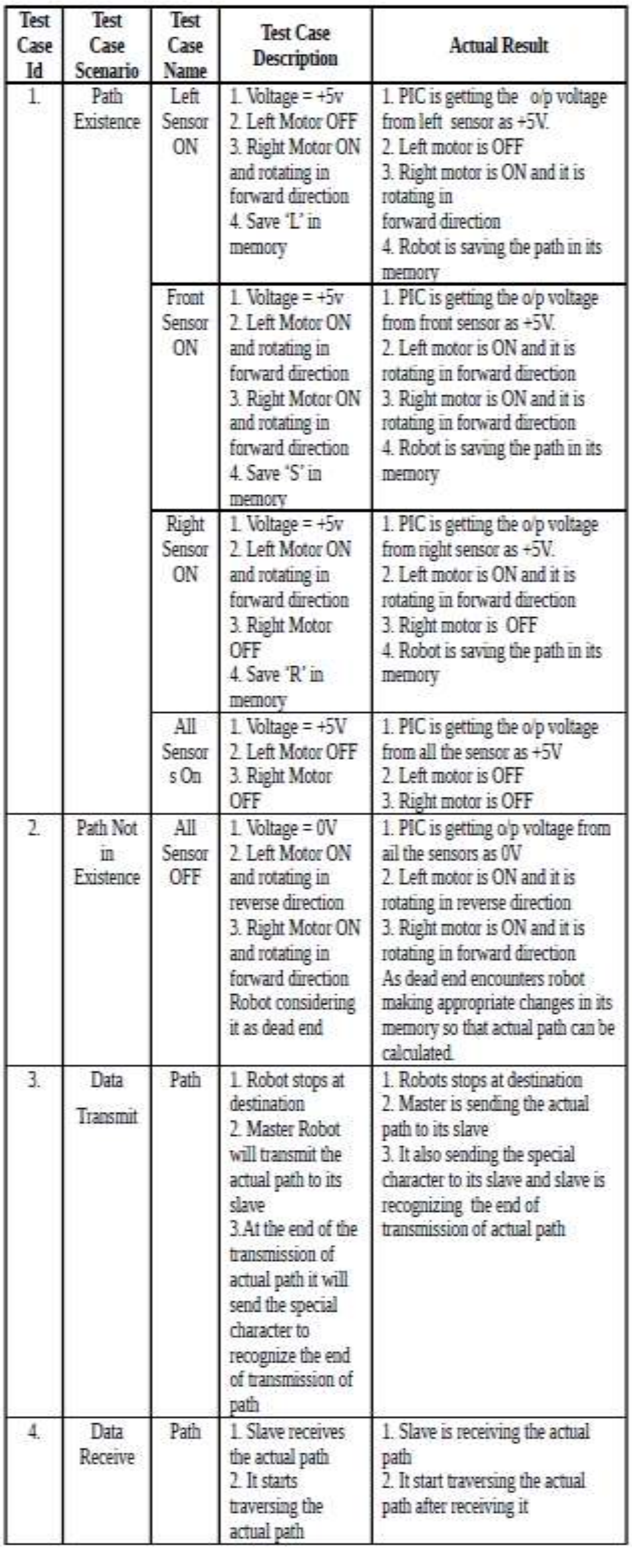


Fig.2. MPLAB IDETest Cases

4.2. Test Cases

We carried out different tests on designed system. The results obtained are summarized below in Table.I. The obtained results are as per the expectation. The designed robot passed all test cases.

Table.I. Test Cases of Designed Embedded Robot System



Test Case Id	Test Case Scenario	Test Case Name	Test Case Description	Actual Result
1.	Path Existence	Left Sensor ON	1. Voltage = +5v 2. Left Motor OFF 3. Right Motor ON and rotating in forward direction 4. Save 'L' in memory	1. PIC is getting the o/p voltage from left sensor as +5V. 2. Left motor is OFF 3. Right motor is ON and it is rotating in forward direction 4. Robot is saving the path in its memory
		Front Sensor ON	1. Voltage = +5v 2. Left Motor ON and rotating in forward direction 3. Right Motor ON and rotating in forward direction 4. Save 'S' in memory	1. PIC is getting the o/p voltage from front sensor as +5V. 2. Left motor is ON and it is rotating in forward direction 3. Right motor is ON and it is rotating in forward direction 4. Robot is saving the path in its memory
		Right Sensor ON	1. Voltage = +5v 2. Left Motor ON and rotating in forward direction 3. Right Motor OFF 4. Save 'R' in memory	1. PIC is getting the o/p voltage from right sensor as +5V. 2. Left motor is ON and it is rotating in forward direction 3. Right motor is OFF 4. Robot is saving the path in its memory
		All Sensors On	1. Voltage = +5V 2. Left Motor OFF 3. Right Motor OFF	1. PIC is getting the o/p voltage from all the sensor as +5V 2. Left motor is OFF 3. Right motor is OFF
2.	Path Not in Existence	All Sensor OFF	1. Voltage = 0V 2. Left Motor ON and rotating in reverse direction 3. Right Motor ON and rotating in forward direction Robot considering it as dead end	1. PIC is getting o/p voltage from ail the sensors as 0V 2. Left motor is ON and it is rotating in reverse direction 3. Right motor is ON and it is rotating in forward direction As dead end encounters robot making appropriate changes in its memory so that actual path can be calculated.
3.	Data Transmit	Path	1. Robot stops at destination 2. Master Robot will transmit the actual path to its slave 3.At the end of the transmission of actual path it will send the special character to recognize the end of transmission of path	1. Robots stops at destination 2. Master is sending the actual path to its slave 3. It also sending the special character to its slave and slave is recognizing the end of transmission of actual path
4.	Data Receive	Path	1. Slave receives the actual path 2. It starts traversing the actual path	1. Slave is receiving the actual path 2. It start traversing the actual path after receiving it

4.3 Result Photos

Master Robot using IR sensors scans all available paths and after reaching destination decides the shortest path using algorithm and sends this information through transmitter to slave robot as shown in fig.3.

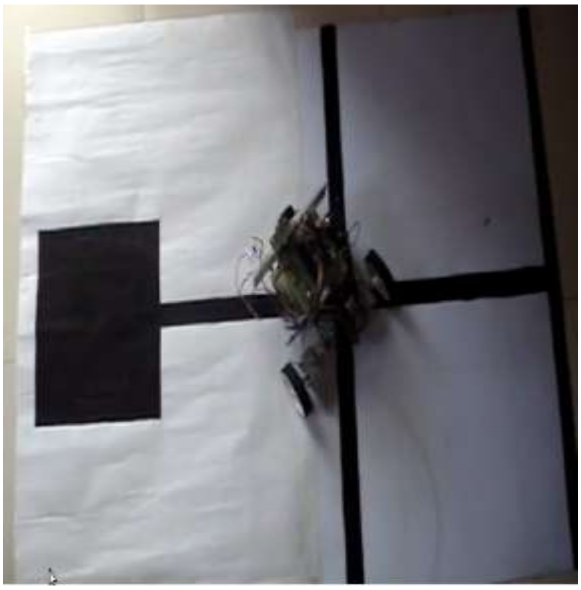


Fig.3. Master Robot travels through all paths & reaches to destination.

Slave Robot receives shortest path information from master robot and starts traveling through given path and reaches to destination as shown in fig.4.

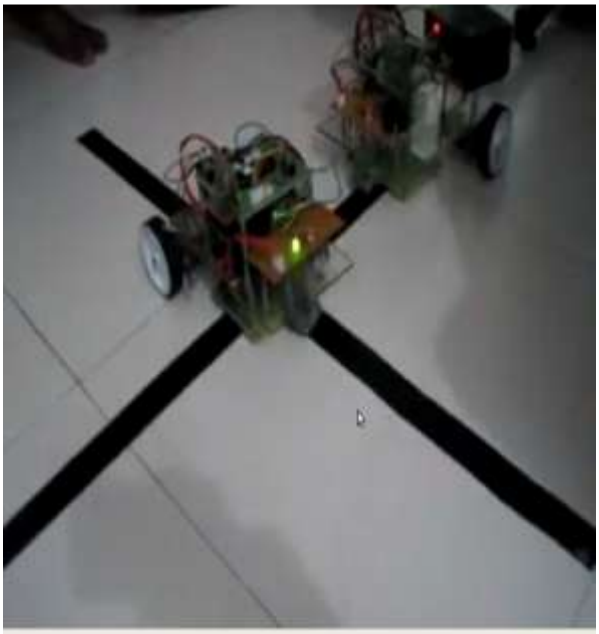


Fig.4. Slave Robot travels through shortest path & reaches to destination.

5. CONCLUSION

Momentum is building in intelligent robotics. During the past few years, there have been significant leaps in self learning and swarm robotic performance in the areas of navigation, defense and security. Researchers have developed robotic swarms that have the potential to co-ordinate with each other. Our work is to implement shortest path planning heuristics algorithm using robot. PIC microcontroller is used to control the motion of the robot. Finally, the master robot travels through all available paths and decides the shortest path and this information is transmitted to slave robot for traveling through given path.

6. REFERENCES

[1] Ellips Masehian, and Davoud Sedighizadeh, “Classic and Heuristic Approaches in Robot Motion Planning – A Chronological Review” Proceedings of world academy of science, engineering and technology volume 23 august 2007 issn 1307-6884.

[2] Antonelli, G.; Chiaverini, S. and Fusco, G.; “A Fuzzy-Logic-Based Approach for Mobile Robot Path Tracking”, IEEE Trans. Fuzzy Sys. Vol. 15, Issue 2, (2007) pp. 211-221.

[3] Gen, M.; Runwei C. and Dingwei W. "Genetic algorithms for solving shortest path problems", Proc. IEEE Int. Conf. on Evolutionary Comput. (1997) pp. 401 – 406.

[4] A. Atramentov and S. M. LaValle. Efficient nearest neighbor searching for motion planning. In submitted to 2002 IEEE International Conference on Robotics and Automation, 2002.

[5] James Bruce,Manuela Veloso,”Real-Time Randomized Path Planning for Robot Navigation”

[6] Gordon MaComb, Myke Predko, “ROBOT BUILDER'S BONANZA”, Third Edition, TAB

[7] Owen Bishop, “Robot Builder's Cookbook - Build and Design Your Own Robots”, Newnes

[8] Bruno Apolloni, “Machine Learning and Robot Perception”, Springer