

Path Traversal Algorithm using Firebird V Robot

Anusha M R

Dept. of ECE, Vidyavardhaka
College of Engineering,
Mysuru, India

Albert Sunil D'Souza

Dept. of ECE, Vidyavardhaka
College of Engineering,
Mysuru, India

Rachana P

Dept. of ECE, Vidyavardhaka
College of Engineering,
Mysuru, India

Amrutha M

Dept. of ECE, Vidyavardhaka
College of Engineering,
Mysuru, India

Arjun K R

Dept. of ECE, Vidyavardhaka
College of Engineering,
Mysuru, India

Rohith K

Dept. of ECE, Vidyavardhaka
College of Engineering,
Mysuru, India

ABSTRACT

This paper aims at developing path traversal algorithm based on the co-ordinates of the axis for Firebird V robot. This algorithm does not require physical indications like markers, white lines and black lines for its path. These robots can be used in surveying pipes, small areas. Robot traces the path drawn by the user using GUI controller application. The Bluetooth module is used to establish communication between robot and computer.

General Terms

Controller application, algorithm, communication

Keywords:

Bluetooth module, Traversal, firebird V robot, GUI.

1. INTRODUCTION

A path traversing robot is basically a robot which traces the path drawn by the user on a GUI controller application. The information can be sent from system to robot through Bluetooth module. This robot can be used as a prototype for industrial robots.

Firebird V robot is controlled by ATMEGE2560 microcontroller which is based on ATMEL's AVR 8bit architecture modeled on ATMEGA8 microcontroller. The functional block diagram of Firebird V robot is as shown below.



Fig1: Firebird V Robot.[1]

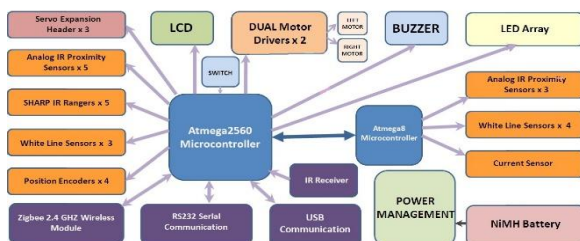


Figure 2. Functional block diagram of Firebird V Robot.[1]

Firebird V robot is driven by two powerful geared dc motors having separate motor drivers equipped with position encoders. It has 16 x 2 character LCD for displaying messages on it. It has 32 channels ADC and equipped with three white line sensors, one IR sharp sensor and eight proximity sensors. It supports various communication modes such as USB, RS232, USART and ZigBee 2.4GHz wireless.

2. LITERATURE REVIEW

COVERAGE PATH PLANNING FOR ROBOTICS by Enric and Marc [2] proposed the task of finding a path consisting of points in an area while avoiding obstacles is known as Coverage Path Planning (CPP). The proposed methods in CPP: The classical exact cellular decomposition method: It breaks the space without obstacles into non-overlapping regions known as cells. These regions can be covered easily by simple robotic motion. Each cell could be covered in a zig-zag pattern as shown in figure below.

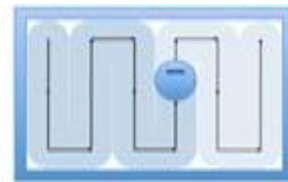


Fig 3: Typical zig-zag path.

Morse-based cellular decomposition method: This method uses sensory range information and motion template based algorithm to encounter all the critical points in the target area. This method can be used in non-polygon obstacle. The drawbacks of CPP are: In real-world robot does not have perfect knowledge about its location and encountering uncertainty in location estimates. TRAVERSAL ALGORITHM FOR COMPLETE COVERAGE (TRACC) by kavitha and balaji. in this method all the cells in the area are sensed. Blocked cells are detected. TRACC traverses to the next cell using the safest path and reach the destination by sensing all cells. The entire region is decomposed into finite cells of 11 rows and 11 columns. S is the starting point and E is the end point. C1, C2, C3,.....C121 are the decomposed cells of the region. In the below figure the cells with circle indicates blocked cells which should be avoided. Intelligent agent (IA) is capable of sensing its four cardinal neighbours F,B,L,R denotes front, back, left and right respectively for sensing.

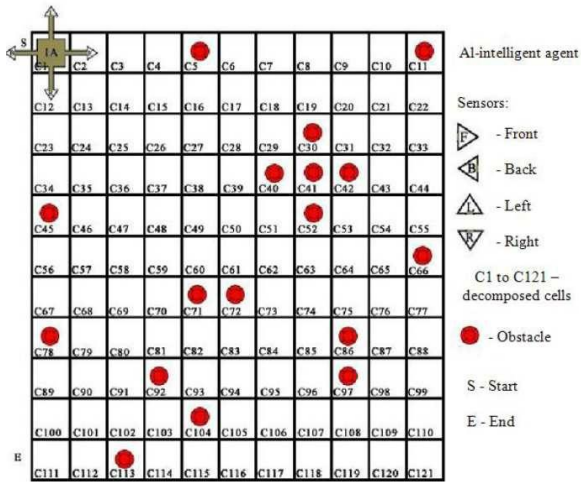


Fig 4: TRACC method.[3]

Drawbacks in TRACC method are: Redundancy, coverage time, more sensors used to keep robot in line and curved paths cannot be achieved.

3. PROPOSED APPROACH

The block diagram of proposed system is shown below.

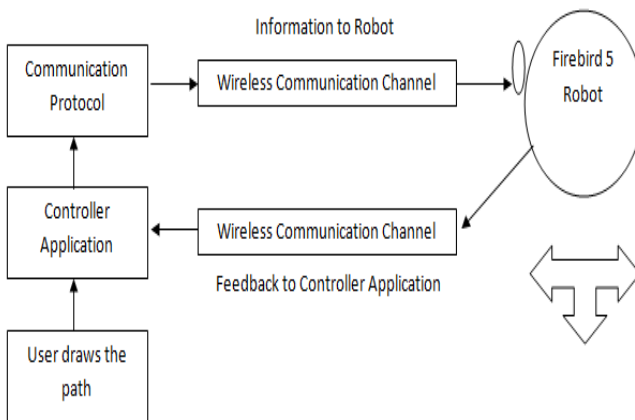


Fig 5: Block diagram of proposed system

The user draws a path on controller application running on computer. The controller application allows the user to draw paths having straight lines. It also allows user to set the scale.

The path information is transmitted using wireless communication module. The robot traverses the path with least error.

3.1 User draws the path on GUI

The user draws the path on the GUI using straight lines.

3.2 Controller application

For distance calculation,

$$D = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \text{ ----- (1)}$$

1. (x_i, y_i) represent the initial co-ordinate where robot is present.
2. (x_{i-1}, y_{i-1}) represent the next co-ordinate where robot travels.

For angle calculation,

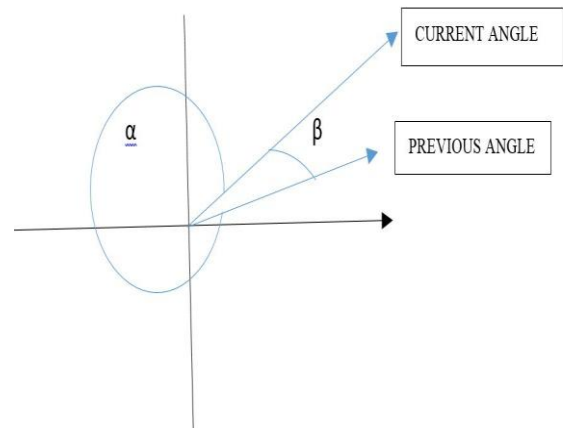


Fig 6: Angle covered when robot takes a left turn.

$$\alpha = 360 - (\text{current angle} - \text{previous angle})$$

$$\beta = \text{current angle} - \text{previous angle}$$

When current angle is greater than the previous angle robot takes a left turn.

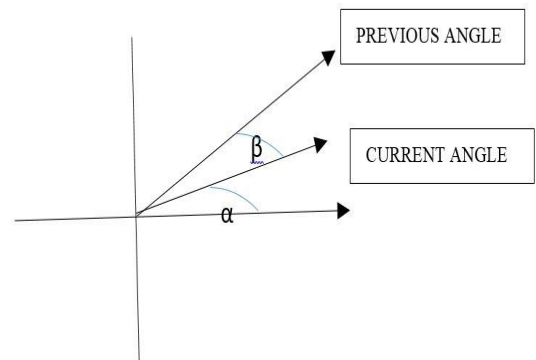


Fig 7: Angle covered when robot takes a right turn.

$$\alpha = 360 - (\text{previous angle} - \text{current angle})$$

$$\beta = \text{previous angle} - \text{current angle}.$$

When current angle is less than the previous angle robot takes a right turn.

3.3 Communication Protocol

The robot is communicated serially through Bluetooth. The MATLAB establishes the serial communication without doing manual works. The data can be sent in simple steps from the Bluetooth device.

3.4 Firebird V Robot Direction control

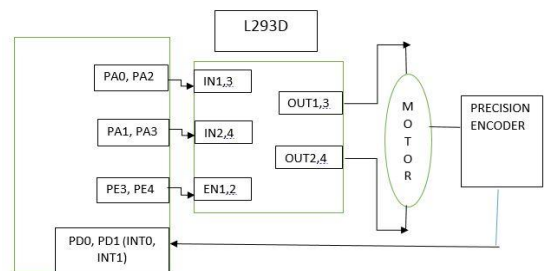


Fig 8: Block diagram of motor movement.

Robot motors are controlled by L293D, microcontroller. It can control direction and velocity of the two motors. Changes in the direction of robot can be made using suitable logic levels applied to L293D direction pins. Pulse width modulation [PWM] is used to control the velocity of wheel and to enable the pins of L293D IC. The control logic are applied in this project are forward (0x06), soft right (0x02), soft left (0x04).

3.5 Motor movement

Case 1: Robot is moving forward or backward (encoder resolution is in mm)

1. Wheel diameter= 5.1cm.
2. Wheel circumference=
 $5.1\text{cm} \times 3.14 = 16.014\text{cm} = 160.14\text{mm}$.
3. Number slots on the encoder disc= 30
4. Position encoder resolution=
 $163.2\text{ mm} / 30 = 5.44\text{mm} / \text{pulse}$.

The minimum distance travelled by robot is 5.44mm.

Case 2: Robot is turning with one wheel stationary while other wheel is rotating clockwise or anti clockwise. Center of rotation is center of the stationary wheel (encoder resolution is in degrees)

In this case only one wheel is rotating and other wheel is stationary so robot will complete its 360° rotation with stationary wheel as its center.

1. Radius of Circle formed in 360° rotation of Robot = Distance between Wheels= 15 cm.
2. Distance Covered by Robot in 3600 Rotation = Circumference of Circle traced = $2 \times 15 \times 3.14$
 $= 94.20\text{ cm}$ or 942 mm.
3. Number of wheel rotations of in 3600 rotation of robot = Circumference of Traced Circle / Circumference of Wheel = $(942 / 160.14) = 5.882$.
4. Total pulses in 360° rotation of robot= $30 \times 5.882 = 176.46$
5. Position encoder resolution in degrees= $360/176 = 2.045\text{degrees/count}$

The minimum angle rotation by robot is 2.045.

3.5.1. Enabling the robot movement through interrupts

1) EIMSK(External interrupt mask register)

7	6	5	4	3	2	1	0
INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Fig 9: EIMSK Register [4]

The individual bits of this interrupt is used to enable the interrupts in EICR register. We have enabled 5th and 6th bit of EICR to enable 4th and 5th bit of EICR

2) EICR (External interrupt control register)

EICR register is mainly used to select the trigger type of the interrupt like rising edge or falling edge. This is adjusted by setting the bits in EICRA and EICRB registers. EICRA register represents 4 interrupts (INT3 – INT0) and EICRB represents 4

interrupts (INT7 – INT4).The below figure shows the bit representation of EICRA register

a) EICRA-External interrupt Control Register

7	6	5	4	3	2	1	0
ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Fig 10: EICRA Register [4]

The trigger type for EICRA is selected based on the following table

Table 1.Trigger Type Of Eicra Register [4]

ISCn1	ISCn2	Description
0	0	The low level of INTn generates an interrupt request.
0	1	Any logical change on INTn generates an interrupt request.
1	0	The falling edge between two samples of INTn generates an interrupt request
1	1	The rising edge between two samples of INTn generates an interrupt request.

The external interrupts 3 to 0 are activated by the external pins INT3:0 if the corresponding interrupt mask in EIMSK is set. The level and the edges on the external pins that activate the interrupts are as defined in the table above where n can be 0,1,2,3...The interrupt can be disabled by clearing the particular bit in the interrupt enable register(EIMSK).The below figure shows the bit representation of EICRB register.

b) EICRB – External Interrupt Control Register B

7	6	5	4	3	2	1	0
ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

The bit operation in this register is same as that of EICRA register but this register represents different set of interrupts (INT7 –INT4) and these interrupts are enabled or disabled by different set of bits in EIMSK register. The trigger type of EICRB is selected is based on the following table

Table 2.Trigger Type Of Eicrb Register [4]

ISCn1	ISCn2	Description
0	0	The low level of INTn generates an interrupt request.
0	1	Any edge of INTn generates asynchronously an interrupt request.
1	0	The falling edge of INTn generates asynchronously an interrupt request.
1	1	The rising edge of INTn generates asynchronously an interrupt request.

In our project we are making use of interrupts 4 and 5 for left and right movement. These two interrupts are activated by setting the particular bits in EIMSK register. The trigger type

required like rising edge or falling edge is set in the EICRB register by considering the table for required interrupts.

3.6 GUI

Graphical User interface (GUI) is a type of user interface that allows user to interact with electronic devices with images rather than text commands. A GUI represents the information and actions available to user through graphical icons and visual indications such as secondary notation, as opposed to text-based interface, typed command labels or text navigation. The term GUI is historically restricted to scope of two dimensional display screens with display resolutions able to describe generic information. In MATLAB, Graphical User Interface Development Environment (GUIDE) tool is used to create the GUI.

In our project we have used five pushbuttons namely:

1. Connect
2. Disconnect
3. Clear
4. Reset
5. Move

1. Connect: When the user presses the connect button, it connects the Bluetooth of the system to the Bluetooth module [H-C-2010-06-01] which is mounted on the robot.

Once the Bluetooth is paired, the connected message will be displayed on the status bar.

2. Disconnect: When the user presses the connect button, it disconnects the Bluetooth of the system and the Bluetooth module which is mounted on the robot.

When the Bluetooth is not paired, the disconnected message will be displayed on the status bar.

Axes path canvas: It is a blank space provided for the user in the GUI, to draw paths using straight lines. This axes internally calculates the coordinates of the path.

3. Clear: It is used to clear the path drawn on the GUI axes.
4. Reset: The scale in the axes can be set to default by using this button.
5. Move: When the user presses this button, the robot traces the path drawn on the axes canvas.

Following steps of algorithm:

Step1: Initializing the wheels interrupts.

Step2: Initializing the motors.

Step3: Configuring Bluetooth by UART (serial port).

Step4: Calling an interrupt for the rotation of wheels and for monitoring the Bluetooth connection

Step5: User draws the path on the GUI

Step6: Disable the other interrupt and store the value in a register.

Step7: First data is always setting the movement or direction stored in UDR3.

Step8: Second data is the value which is the actual data for the robot movement.

Step9: Rotation of the wheel is counted by the microcontroller and the counter is started counter=counter+1

Step10: Counter is started, once it reaches the value which is calculated in above mentioned cases.

4. RESULTS

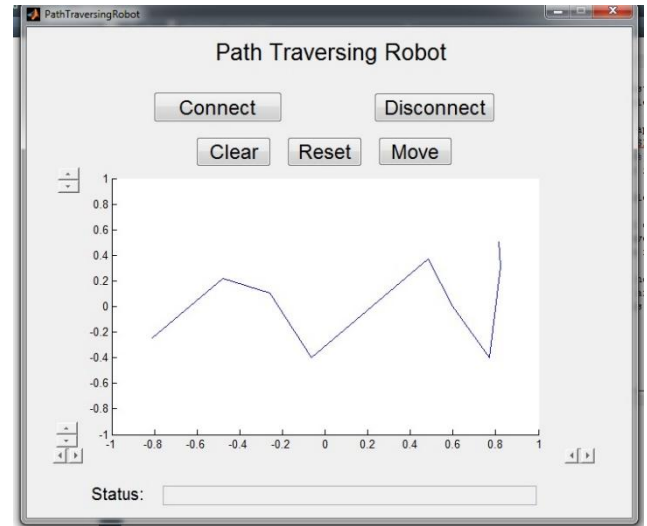


Fig 9: Path drawn on GUI

The above figure shows developed GUI for our project. On pressing the connect pushbutton connection is established between robot and computer via Bluetooth module and the status is displayed on status bar as shown in the above GUI display. Based on the path drawn by user on the axes of GUI, the robot traverses the path accordingly.

5. CONCLUSION

This paper proposes idea of robot movement using an algorithm which indicates the distance travelled by the robot based on the path drawn on GUI along with angle consideration. This helps the user during surveying smaller regions like tunnel. This robot can be used as reference for surveying in various industrial applications.

6. REFERENCES

- [1] <https://www.scribd.com/doc/229674761/Fire-Bird-v-ATMEGA2560-Hardware-Manual-2010-03-26>
- [2] Enric Galeran, Marc carrears "A survey on coverage path planning for robotics" 'Elsevier', 20th September 2013.
- [3] Journal by Kavitha Tyagarajan and Coimbatore Ganeshsankar Balaji, "Traversal algorithm for complete coverage (TRACC)" on 15-12-2015.
- [4] <https://www.scribd.com/doc/229674761/Fire-Bird-v-ATMEGA2560-Software-Manual-2010-03-26>
- [5] The Essential Guide to User Interface Design <http://ps.fragne1.edu.in/~dipalis/prgdwnl/eguid.pdf>
- [6] Reference book on Data Structure by Mark Allen Weiss, "Data Structure and Algorithm Analysis in C++"third edition.