Real Time Navigation of Robot with Path Memorizing Algorithm

Urvi Patel[1], Narotam Sharma[2], Professor Vishal Vaidya[3]
[1,2,3] Dept. of Instrumentation & Control Engineering, Institute of Technology,
Nirma University, Ahmadabad, Gujarat, India

***Abstract* - To solve, day to day problems and to make the human life more comfortable the growth towards the technology is increased intensively. New generation is focusing on building autonomous robots to make the world a better and comfortable place to live in. Recently, the algorithms for the autonomous vehicle have greatly evolved, providing efficient and effective solutions to cope with the variability and the complexity of real-world environments. Many companies are investing a lot of money in developing such algorithm to make driving easier and safer. The paper deals with the same idea to increase robot efficiency for remembering places and path followed during the manual control mode with the help of feedback-based control system. Path memorizing robot can prove to be very useful in tedious job of moving raw materials from one place to another in an industry as it can remember the path so it can autonomously take the material to the destination as many times as required. Also, it can be used to transport woods from the forest area to the industries without involving the man force. During all this transportation there are other factors for security of the robot which cannot be ignored. The robot should be able to avoid the obstacles coming in its way and after that it should come back to its original route to reach the destination. For this purpose, various sensors can be used to keep the robot on track and avoid the obstacle.**

Key Words*: *Autonomous Robot, Path Memorizing Algorithm, Differential Drive, Ackermann's Principle

1. INTRODUCTION

A robot is built basically to lessen the human effort involved in performing a particular task as far efficiently. A lot of effort is made to make the robot more human friendly by giving it a good user interface which includes response, recognition, autonomous navigation etc [1].

Path memorized algorithm, the term which is coined to explain the working of the robot in terms of remembering the path it follows. The idea to make robot remember the path it has followed to reach the particular destination works in the same way as in human beings. A person remembers the path it has once taken to reach the destination and follows it every time the person needs to go there, same can be done with the robot which can store the path, as memory storage is not an issue nowadays, to be followed in a memory and by teaching the robot how to reach the destination avoiding any obstacles coming in way. The path which is once stored can be altered as per the need of the user, further different paths can also be stored at a time, depending on the memory storage, to be followed by the robot.

In this paper, the algorithm consists of two stages: one where the robot is being manually controlled by a remote controller and the path it followed in this mode will be stored in the memory. The second stage follows the same path travelled in manual mode autonomously in real time. In controller stage robot is wirelessly controlled through mobile application via bluetooth connection. In auto mode when obstacle is detected it will detour around them returning on the original position.

The experiment is divided into two parts. In first part, path memorizing code is implemented on the differential drive on two wheeled robot. In second part, four wheeled robot was used and the path memorizing code was implemented on it by applying Ackerman's formula. Magnetometer and encoder are used to get the exact position of the robot and distance travelled by it, including range sensors to avoid obstacles. The algorithm is compatible with every existing technology as well as is simple and cheap to apply in daily usage.

1.1 Block Diagram

This paper proposes an algorithm which will eliminate the use of complex circuitry and sensors [1]. In this experiment, few sensors are used for feedback. It includes algorithm for two modes: (1) Manual mode (2) Auto mode.

Figure-1 given below is the basic block diagram of the system. The input commands in manual mode are given by mobile application which we developed for controlling the robot remotely. In auto mode, the direction and the path, stored in the memory of microprocessor, is used as command for direction and speed for the path to be followed. Atmega was used as the controller of the system and Raspberry Pi as processor of the system.

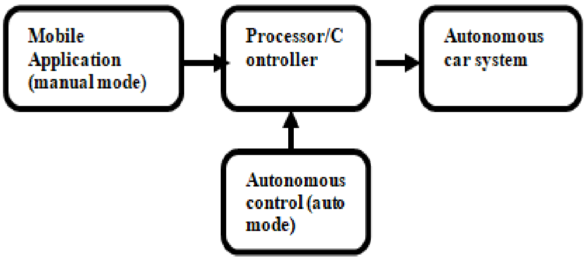
Mobile Application (manual mode)
Processor/C ontroller
Autonomous car system
Autonomous control (auto mode)

Fig-1: Block Diagram

2. FIRST PHASE USING DIFFERENTIAL DRIVE

2.1 Differential Drive

In first phase, differential drive concept is applied for taking left or right turn. It consists of 2 drive wheels mounted on a common axis, and each one can be independently driven either forward or backward or left or right [2].

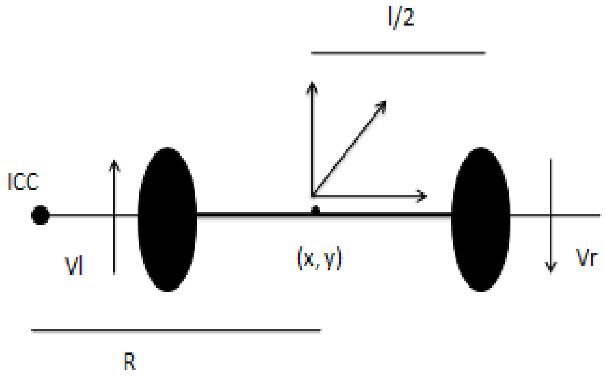

Fig-2: Differential Drive Principal

During the manual control when the robot is moving in forward direction, the speed and the direction of both wheels will be same. Variation in the wheel speed leads to change in the direction of the robot. The point about which the robot rotates is called instantaneous center of curvature (ICC). For the movement of the robot in any direction the rotational velocity of both the wheels remain same. The equations for finding the individual velocity of two wheels will be as follows:

$\Omega(R + l/2) = V_r$…………………… (1)

$\Omega(R - l/2) = V_l$……………………… (2)

Where l is the distance between the centers of the two wheels, V_r, V_l are the right and left wheel velocities along the ground, and R is the signed distance from the ICC to the midpoint between the wheels [2].

2.2 Application in project

When velocities of both the wheels are same and both wheels are moving in same direction the robot moves in straight line. When the velocities of both wheels are same but the direction is opposite the robot rotates at the same position with ICC as a center. And when the velocities are different the robot will move in the direction or wheel with higher velocity. Velocities of the wheels can be calculated by equation (1) and (2).

In this phase, we used Atmega as controller and processor. The path travelled during manual mode is stored in the EEPROM of Atmega. Also, the time for which the robot moves in forward direction is stored in the EEPROM along with angle of the turn. The memory storage capacity of EEPROM is limited, which limits the number of paths which can be stored in the EEPROM.

Magnetometer records the angle at which the robot turned right or left, which is stored in the EEPROM. During auto mode for smooth navigation and turning of robot, the stored values are continuously compared with the current value from magnetometer to precisely control the angle at which the robot turns. The forward movement of the robot is time based. The time for which the robot moves in forward direction is stored; this value will be used as timer value in auto mode for forward movement.

Data stored EEPROM is array of strings indicating the path travelled by the robot. For instance, if the car has travelled in forward direction for 30 seconds and then turned left in manual mode, then the array will consist of two commands which will be executed during auto mode. First command will be for forward direction with time, for which it moved in forward direction, “f30”, and second command will be “r70” with the angle at which the robot turned to right.

3. SECOND PHASE USING ACKERMAN’S PRINCIPAL

3.1 Ackerman’s principal

After the successful completion of first phase, it has been moved on to a four wheeled robot. Four wheeled robot can be used in any surrounding and can easily transport heavy material to a destination point. The limitation of memory storage is solved in this phase by using Raspberry Pi. Raspberry Pi uses SD card which will provide enough memory storage to store multiple paths, further Raspberry Pi will provide an advantage of remote excess in the robot. In first phase, we only used magnetometer for angle feedback. In this phase we used various sensors for obstacle avoidance along with an algorithm for following the path precisely. As the four wheeled robot is used in this phase differential drive equations can't be used to find the speed of the wheels.

The intention of Ackermann geometry is to prevent the need for tires to slip sideways when following the path around a curve. Rather than preceding "turntable" steering, where both front wheels turned around a common pivot, each wheel gained its own pivot, close to its own hub. With the help of this slipping of tires can be avoided sideways and more accurate driving experience can be obtained [3].

To move the car smoothly during turning, the front wheels should rotate around the center of the imaginary line drawn from two front wheels and two wheels at the back of the car. The basic idea of the Ackermann’s principal is that the inner wheel should move at slower speed than the outer wheels. Thus, the inner wheels will steer at larger angles than that of outer wheels [3]. The figure is shown is shown below:

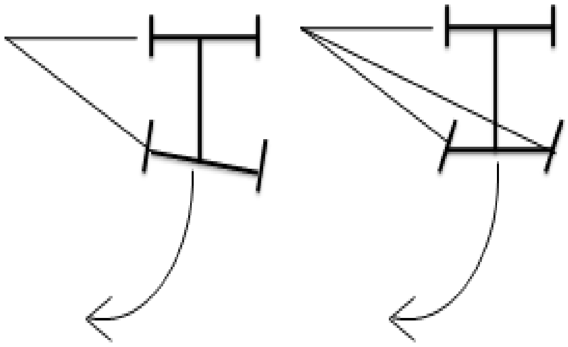

Fig-3: Ackermann’s Principal Diagram

3.2 Communication between Atmega and Raspberry Pi

Serial Communication between Atmega and Raspberry Pi is done with USB port. The use of Raspberry Pi will eliminate the requirement of computer for control and tracking of robot. Here, two way communications is required; which is from Atmega to Raspberry Pi and from Raspberry Pi to

Atmega; for writing commands to memory of Raspberry Pi and reading commands from memory of Raspberry Pi.

Firstly, the serial port in Raspberry Pi is to be located for connection. After that, to enable the communication between Raspberry Pi and Atmega through USB, for Atmega a built-in Serial library was needed and for Raspberry Pi the installation of pySerial module was needed.

During manual mode, the command from the mobile application received in Atmega is sent serially to python in Raspberry Pi. Python writes the received data into text file. Main advantage of using Raspberry Pi is the storage of multiple path file along with this, the files can be retrieved as per the requirement. This enables us to store multiple paths at a time.

In auto mode, the file in which commands is stored is opened and send to the processor by python. This data sent to Atmega, controls the path travelled by robot in auto mode.

3.3 Manual Mode

In manual mode, the robot is controlled by the mobile application, as shown in Fig.3. , developed during first phase of the project. The data of path travelled by the robot is obtained by two sensors encoder and magnetometer. The data obtained from the sensor is stored in the text file in the Raspberry Pi so it will be stored for as long as the user wants and user can also change the data.

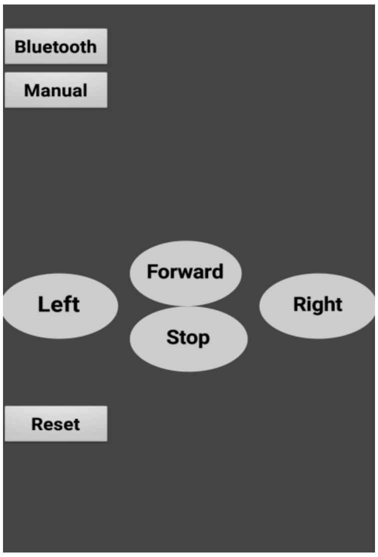

Fig-4: Mobile Application

As seen in Fig.4. , there are various options for connecting with robot and controlling it. By selecting the angle of turning, it will be saved in the text file. There are switches for Forward, Left, Right, Stop and Reset. Manual mode button is for selecting in which mode we want the car to run Manual or Auto. The Bluetooth button is to set a bluetooth connection between the robot and Mobile used.

Forward path travelled by the robot is obtained from the encoder counts. These counts are converted to distance and stored in the memory for forward motion. Now, for controlling the movement of steering wheel of the robot gear assembly was used and potentiometer was attached to the assembly. Whenever the steering wheel will rotate the value of potentiometer will change, this value is mapped with the angle, so the value stored in the text file will be at the angle at which robot turned.

The direction is obtained from the manual command given by the user in manual mode; for example, if the robot has travelled in a straight line and then turned, then the data stored in memory will be distance for straight motion and "l" for left command. Also the value of the potentiometer is stored in the text file to turn the steering wheel during auto mode.

3.4 Auto Mode

Once the path is stored in the text file in Raspberry Pi, it can be used repeatedly. The robot will follow the path, the user commands. Thus, it can be used for applications which require transporting raw materials in industry from one place to other whenever required. In auto mode, the robot moves as per the command and to safely reach the destination it will require avoiding the obstacles coming in the way. Thus, obstacle avoidance sensors are added to the robot.

To avoid the obstacles, we applied obstacle avoidance algorithm for smooth navigation of the robot in auto mode. The robot movement is based on the data stored in the text file in Pi. To ensure that robot moves exact the same distance and turns at exact the same angle as stored, the stored data is compared with the real time data obtained from the sensors used to record the data making it closed loop system. In manual mode, the real time data is obtained to store it as command, while in auto mode the real time data is used for feedback to form a closed loop system.

The data stored in the file are string commands for every movement. For example, for forward movement the "f3" is stored in the file, where f is for forward direction and 3 is the distance travelled by the robot in forward direction. Similarly, when the robot turns left "l90" is stored, where l is for left turn and 30 is the angle at which the robot turned left. The potentiometer value of the steering wheel is also stored in the command for more accurate results while taking a turn in auto mode.

To avoid the obstacles in auto mode range sensors are used to detect the obstacles coming in the way. Range sensors are easy to use and implement to avoid the obstacles coming in the way. Long distance and short distance range sensors were used to detect the obstacles at different distance.

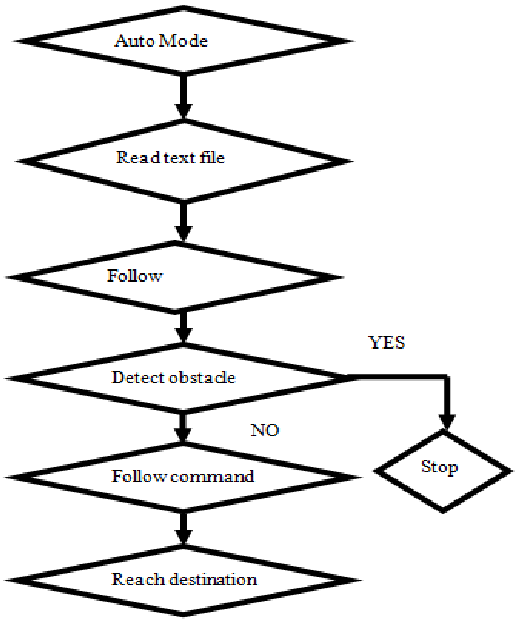

Fig-5: Flow Chart of Auto Mode

4. CONCLUSION

In this paper, the basic difference between horse driven car system and automobile driven car is explained. Further the system work as a Path memorized which actually remember the path driven and try to replicate it accordingly. The System is been tested on two type of car system one which follows the differential principle and another one which is nowadays more familiar to a normal car "Ackermann's Principle".

The system uses sensors which are easily available and easy to use. It makes the system simple and cost effective solution to tedious work for transferring material in industries.

The system can also expand to make it safer for the public in real life by adding more feedback like cameras, lidar etc.

REFERENCES

[1] Kelvin D'Souza, Steffi Joseph, Ashley Jose, Sushant Chavan, Yogesh Gholap, "A Novel Approach to Navigation of a Robot using a Memorizing Algorithm" International Journal of Applied Information Systems, Vol.2– No.7, May 2012

[2] Gregory Dudek, Michael Jenkin: Computational Principles of Mobile Robotics. Cambridge University Press 2000

[3] Retrieved from http://what-when-how.com/automobile/the-ackermann-principle-as-appliedo-steering-automobile/