Autonomous Path Finding Robot

Prof. Vikrant A. Agaskar
Assistant Professor,
dept. Computer Engineering VCET
Mumbai, India

Mr. Pranav Mahesh Bhosale
dept. Computer Engineering VCET
Mumbai, India

Mr.Pankaj Prakash Dhanve
dept. Computer Engineering VCET
Mumbai, India

Mr.Vikas Khedkar
dept. Computer Engineering VCET
Mumbai, India

***Abstract*— This paper presents the detailed steps for designing and developing an autonomous robot. Autonomous robot creation has been a hot topic in the AI field. With recent technological developments, autonomous robots are gaining growing interest around the globe, and there are a number of ongoing research and development initiatives in both industry and academia. Location of obstacles in complex ground system is unknown. Route finding in such environment is actually really hot problems for robots. In this paper we present the design and implementation of a multi-sensor-based route to find autonomous robot framework for exploration. With the goals of seeking direction in a complex terrain environment with different obstacles, a novel path finding method integrating line tracking and obstacle avoidance algorithms is proposed and implemented. Experimental results indicate that our robot system with our path finding algorithm can effectively solve path finding problem in complex ground environment and avoid collision with the obstacles**

I. INTRODUCTION

Automated robots have been a long and interesting area of research, one area of practical importance is an automatic robot course. We need a good algorithm guiding path for the given application. Much research has been done on this subject and has shown promising results. This work aims to get a first taste of the topic and other developments in robotics. Our main goal is to find the correct and safe way to the destination to avoid all obstacles and to give live feed on your smartphone phone. For For example, if the robot is held at a specific location in a room and the coordinates of a particular destination are given, it should move in that direction, thus avoiding all obstacles in its way and reaching its destination. Within this project we have Arduino UNO . We used sensors in the provided path to direct the robot and to coordinate and provide live feed for the path.

II. PROBLEM STATEMENT

An independent robot is such a machine who has no in advance information about route and which needs a few calculation to be utilized for making the headings for route. Way arranging accomplishes something comparative if there should be an occurrence of self-ruling robots with the utilization of electronic sensors and control framework (calculation). The field of automated mapping addresses the issue of robot route where the utilization of GPS isn't accessible or conceivable. It considers the capacity of a robot to review its environment, assemble a virtual guide and move along an ideal way. The robot utilizes Laser Detection and Ranging (LIDAR), ultrasound and different sensors for information assortment to comprehend its environment. This information is utilized to fabricate a virtual guide of its environment and fabricate a way on the fly as the robot continues. This is alluded to as Mapping. A robot that explores utilizing this guide must have the option to precisely figure its situation concerning the tourist spots in the guide, and find itself right now. This is known as Localization. To move along an ideal way both mapping and restriction is important.

III. MOTIVATION

Autonomous robotics is one of the most key topics of this generation of research. It has a wide range of applications, such as construction, manufacturing, waste management, space exploration, and military transportation. One of the main areas of research, in order to achieve successful autonomous robots, is path finding. Path finding in robotics is defined as navigation that shall be collision free and most shortest or optimum for the autonomous vehicle to manoeuvre from a source to its destination. This thesis concentrates on building a path finding algorithm for an all-terrain vehicle (ATV) used for travelling in an open field or forest. The novelty of this algorithm is that it does not simply create a path between a source to its destination, but it makes sure that the vehicle covers the entire field area when navigating from the source to its destination. An autonomous robot is such a person who has no beforehand data about navigation and which needs some algorithm to be used for creating the directions for navigation. Path planning does something similar in case of autonomous robots with the use of electronic sensors and control system (algorithm).

IV. LITERATURE SURVEY

1. Domain Explanation – Robotics

An interdisciplinary part of designing and science that incorporates mechanical building, electronic building, data building, software engineering, and others. Apply autonomy manages the structure, development, activity, and utilization of robots, just as PC frameworks for their control, tangible criticism, and data handling.

These advances are utilized to create machines that can fill in for people and duplicate human activities. Robots can be utilized much of the time and for bunches of purposes, yet today many are utilized in hazardous conditions (counting bomb recognition and deactivation), producing forms, or where people can't endure (for example in space, submerged, in high warmth, and tidy up and control of perilous materials and radiation). Robots can take on any structure yet some are shown up. This is said to help in the acknowledgment of a robot in certain replicative practices for the most part performed by individuals. Such robots endeavor to duplicate strolling, lifting, discourse, comprehension, or some other human movement. A considerable lot of the present robots are roused ordinarily, adding to the field of bio-motivated apply autonomy.

The idea of making machines that can work self-governingly goes back to old style times, yet investigation into the usefulness and potential employments of robots didn't develop significantly until the twentieth century. Since the beginning, it has been as often as possible expected by different researchers, innovators, specialists, and experts that robots will one day have the option to emulate human conduct and oversee undertakings in a human-like design. Today, mechanical technology is a quickly developing field, as innovative advances keep; examining, structuring, and fabricating new robots fill different down to earth needs, regardless of whether locally, financially, or militarily. Numerous robots are worked to do tasks that are risky to individuals, for example, defusing bombs, discovering survivors in unsteady destroys, and investigating mines and wrecks. Apply autonomy is likewise utilized in STEM (science, innovation, building, and arithmetic) as an instructing aid.[1] The approach of nanorobots, minute robots that can be infused into the human body, could change medication and human health.[2]

Apply autonomy is a part of building that includes the origination, structure, assembling, and activity of robots. This field covers with hardware, software engineering, man-made consciousness, mechatronics, nanotechnology and bioengineering.[3]

A couple of way discovering/following calculations have been presented in most recent two decades. A portion of the celebrated calculations are the Pure Pursuit [4]. [5] and the Follow the Carrot [5]. These calculations use the spatial data for calculation of the directing directions and help a robot in following a given way or stripe. In any case, both these calculations don't regularly think about the real bends of the objective way/stripe. Because of this deficiency of these the two calculations, they furnish with less exactness. Vector interest [6] is another way following strategy that depends on a hypothesis initially introduced by R. S. Ball in the year 1900. The Ball's hypothesis is normally utilized to show the development of an item as for the focused on facilitate framework. Another methodology was additionally displayed by T. Hellstrom [7]. The displayed approach was utilizing the recorded direction what's more, controlling directions to choose the heading in which the robot moves. One sort of vehicles are Automated Guided Vehicles (AGVs) [1], [3] those are furnished with sensors and by utilizing the sensors, such vehicles can follow a way. Such robot vehicles can be utilized for different purposes particularly in processing plants in assembling plants, transportation reason, and so forth. Such vehicles are modest in cost as well as increasingly successful and proficient. All the above examined approaches, strategies and calculations are not productive as far as following the bended lines or ways. We intend to conquer this inadequacy of the current methodologies and present another strong procedure.

V. CURRENT SYSTEM

Wheeled robots will be robots which utilize mechanized wheels to move themselves to explore the ground. This methodology is more straightforward than utilizing tracks or legs and they are simpler to configuration, construct, and program in level, not - so-rough territory utilizing wheels for development. Likewise, they are preferred controlled over different sorts of robots. The disservice of wheeled robots is that they can't explore well over impediments, for example, rough territory, sharp decays or low-grinding zones. Wheeled robots are generally mainstream in the shopper showcase, with minimal effort and straightforwardness gave by their differential controlling. Robots can have any number of wheels, however there are three wheels enough for static and dynamic equalization. Extra wheels can add to adjust; be that as it may, extra instruments will be required to keep all the wheels in the ground, when the territory isn't level.

VI. PROPOSED SYSTEM

We have made a plastic frame which comprises of 2 wheels on the posterior and a Ball Bering in the front side. As a matter of first importance we have utilized different sensors to coordinate the given robot utilizing raspberry pi a specific way. We have coded in Embedded C language as we thought that it was productive for this specific task. Presently the directions are sent by Bluetooth and those directions can be sent to the Bluetooth chip by utilizing your cell phone. Presently our

Arduino gives a yield of 5V just so we have utilized a LN2 moto driver to run our engines . There are two engines ,they are appended to the two wheels . For alter course they
move inverse way , Since we've utilized the engine driver both the engines get the necessary force. To control the engine driver we're utilizing a fundamental 12 V battery which is set at the base of the pursuit. We have utilized MPU sensors for guiding the way to explicitly take the yaw development for the all bearing . We have introduced a Ball bearing in the front increment the productivity of the robot ,to diminish its going span to least.

VII . Implementation

1 Design Consideration

1.1 Arduino Uno:

The Arduino UNO Microcontroller interfaces the project's hardware part with the software part. It is used to calculate the voltage as well as the battery temperature and send it over its Bluetooth module to the android smartphone. One integral part of the system is the microcontroller.

1.2 MPU 6050:

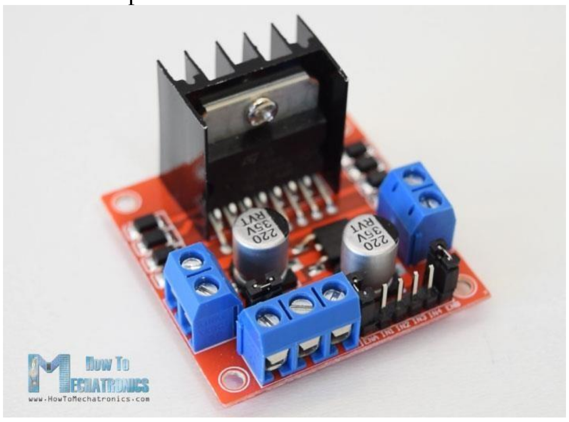
The MPU-6050 InvenSense sensor features a MEMS accelerometer and a MEMS gyro in a single chip. It is very precise, since it includes 16-bit analogy for each channel to digital conversion hardware. Thus it simultaneously absorbs thex, y, and z channels. For interface with the Arduino the sensor uses the I2C-bus.
The MPU-6050 isn't costly, especially since it incorporates both an accelerometer and a gyro.

1.3 MPU 9250

The MPU-9250 is InvenSense ®'s new nine axis MEMS sensor. This substitutes for the famous EOL'd MPU-9150. InvenSense ® lowered power consumption and decreased the size of the MPU-9150 by 44 percent. The MPU-9250 uses 16- bit analog to digital converters (ADCs) to digitize all 9 axes. "Gyro noise performance is 3x better, and compass full scale range is more than 4x better than competitive offerings."

1.4 :L298N Motor driver

The L298N is a dual H-Bridge motor driver which simultaneously allows the control of speed and direction of two DC motors. The module can operate DC motors with voltages between 5 and 35V, with a maximum current of up to 2A

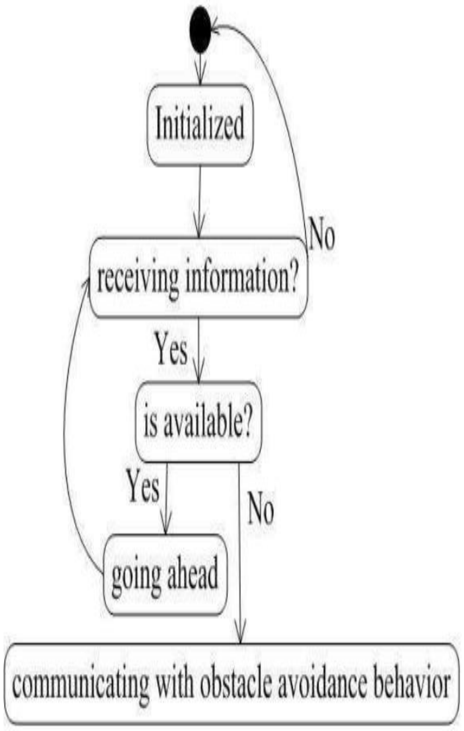


1.5 Ultrasonic sensor HC-S

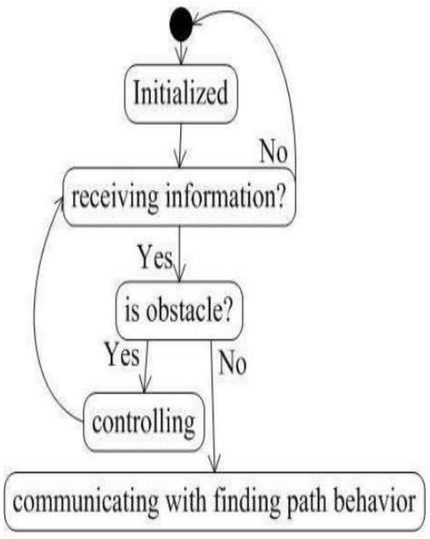
HC-SR04 is an ultrasonic sensor used specifically to calculate the target object's distance. It calculates accurate distance using non-contact technology–a system that does not require any physical contact between sensor and object. Transmitter and receiver are two main parts of the sensor where the former converts an electrical signal to ultrasonic waves, and later converts it back to electrical signals. Such ultrasonic waves are nothing but sound signals that can be calculated at the receiving end and shown there.

Design Details

1.2 State Diagram

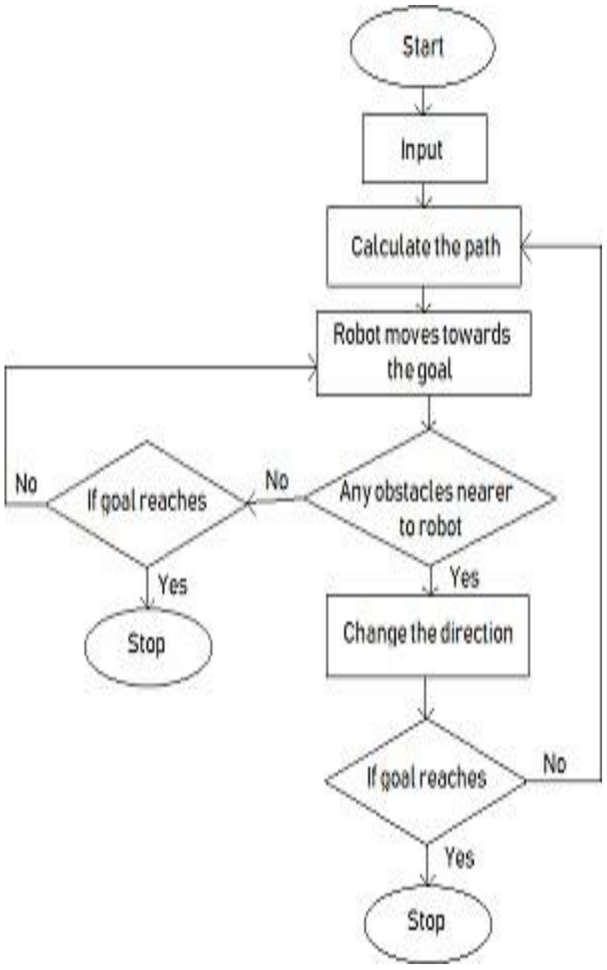


State Transition diagram for the finding path behaviour

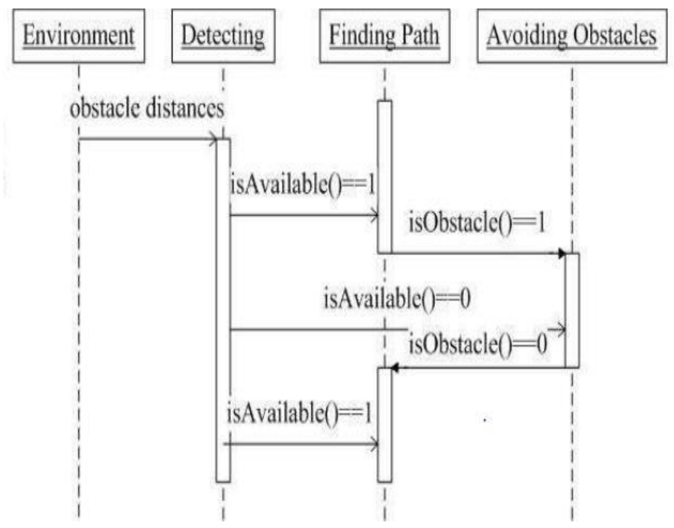


The state Diagram for the obstacle avoidance Behaviour

1.3 Activity Diagram



4.2.3 Sequence Diagram



3. Implementation Methodology

Autonomous robot is a robot that is able to carry out certain work independently without human assistance. Navigation autonomous is one of autonomous robot capabilities for traveling from one stage to another. Implementation of Autonomous robot navigation to navigate an unknown area, requires that the robot explore and map the environment and look for the path to reach a certain point. Path Finding Robot is a mobile robot which uses differential steering wheels .We have used various sensors to direct the given robot using Arduino UNO in a particular direction. We have coded in Embedded C language as we found it efficient for this particular project. Now the coordinates are sent by Bluetooth and those coordinates can be sent to the Bluetooth chip by using your mobile phone. Now our Arduino UNO gives an output of 5V only so we have used a LN2 moto driver to run our motors . There are two motors ,they are attached to both wheels . For change direction they move in opposite direction , Since we've used the motor driver both the motors get the required power. To power the motor driver we're using a basic 12 V battery which is placed at the bottom of the chase. We have used MPU sensors for directing the path to specifically take the yaw movement for the all bearing . We have installed a Ball bearing in the front to increase the efficiency of the robot ,to reduce its turning radius to minimum.
As we have too find the suitable path there can be obstacles in the path which makes our robots path difficult thus it also has an obstacle avoiding part if the obstacle comes in the near range of the robot it changes the path and then continues with its regular path to reach its destination.

4. Code

```
/*
 * for clockwise pin 5 is high and 6 is low
 * for anticlockwise pin 10 is high and 9 is low
 */
const int trigPin
= 11; const int
echoPin = 12;
// defines variables
#include <math.h>
#include <Wire.h>
#include
<MPU6050.h>
float pitch
= 0; float
roll = 0;
float yaw =
0; double
temp=1.0;
unsigned long
timer = 0; float
timestep = 0.01;
double m=-1;
int
steps=0
; int
counter
=0;
//double
M=-1;
MPU605
0 mpu;
int
outputA
=8;   int
outputB
=9;   int
aState;
int
aLastState
=0; int
flag=0;
int
bkl=
1;
void
setup
()
{
pinMode(trigPin, OUTPUT); // Sets the trigPin as an
Output pinMode(echoPin, INPUT); // Sets the
echoPin as an Input pinMode(4,OUTPUT);
pinMode(5,OUTPU
T);
pinMode(6,OUTPU
T);
pinMode(7,OUTPU
T);
pinMode(10,OUTP
UT);
digitalWrite(7,HIGH
);
pinMode(outputA,IN
PUT);
pinMode(outputB,IN
PUT);
Serial.begin(115200)
;
aLastState = digitalRead(outputA);
while(!mpu.begin(MPU6050_SCALE_2000DPS,
MPU6050_RANGE_2G))
  {
    Serial.println("Could not find a valid MPU6050
sensor, check wiring!");
    delay(500);
  }
  mpu.calibrateGy
  ro();
  mpu.setThreshol
  d(3);
}
void loop()
{
  int
distance;
timer =
millis();
while(Serial.available()>0)
{
  delay(10);
double
a=Serial.parseInt();
Serial.println(a);
double
b=Serial.parseInt();
Serial.println(b);
char bd =
Serial.read();
double z= b/a;
m = a tan(z);
m=m*57.2958; // degree
conversion double distance =
sqrt((a*a) + (b*b));
steps=int((distance)*1.8);
if(m<0)
  m=m-
  (2*m);
   if(a>0
&& b>0)
m=int(90
    -m);

if(a<0     &&
b>0)
m=int(270+
m);     if(a<0
&&       b<0)
m=int(270-
m);     if(a>0
&&       b<0)
m=int(90+m
);
}
Vector norm =
mpu.readNormalizeGyro(); yaw = yaw
+ norm.ZAxis * timeStep; temp=-(yaw
+norm.ZAxis * timeStep);
if(flag!=bkl)
{
if(temp<=m)
{
Serial.println(m)
;
analogWrite(10,
0);
analogWrite(4,2
00);
analogWrite(5,0
);
analogWrite(6,2
00); delay(9);
analogWrite(4,0
);
analogWrite(5,0
);
analogWrite(6,0
);
analogWrite(10,
0);
}
if(temp>=m)
{
  while(counter!=steps)//1.8 is the multiplying factor
  {
 aState = digitalRead(outputA); // Reads the "current"
state of the outputA
  // If the previous and the current state of the output A
are different, that means a Pulse has occured
analogWrite(10,LOW);
  analogWrite(4,128);
  digitalWrite(5,HIGH);
  analogWrite(6,LOW);
  delay(10);
  analogWrite(10,LOW);
  analogWrite(4,LOW);
  analogWrite(5,LOW);
  analogWrite(6,LOW);
  counter++;
     Serial.print("Position: ");
  Serial.println(counter);
  aLastState = aState;
  distance=sonar(trigPin,echoPi
  n); Serial.print("distance:");
  Serial.println(distance);
  if(distance<10)
  {
    Serial.println("entere
d"); analogWrite(10,0);
analogWrite(4,200);
analogWrite(5,0);
analogWrite(6,200);
delay(1000);
analogWrite(4,0);
analogWrite(5,0);
analogWrite(6,0);
analogWrite(10,0);
delay(10);
analogWrite(10,LOW)
; analogWrite(4,128);
digitalWrite(5,HIGH);
analogWrite(6,LOW);
delay(500);
analogWrite(10,LOW)
;
analogWrite(4,LOW);
analogWrite(5,LOW);
analogWrite(6,LOW);
delay(10);
analogWrite(10,200);
analogWrite(4,0);
analogWrite(5,200);
analogWrite(6,0);
delay(1200);
analogWrite(4,0);
analogWrite(5,0);
analogWrite(6,0);
analogWrite(10,0);
counter++;
}
}
}
}
flag=flag+1;
}
int sonar(int trigPin,int echoPin)
{
 int distance,duration;
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin,
HIGH); distance=
duration*0.034/2;
return(distance);
}
```

VIII. Limitation

This paper focuses on building an all-terrain vehicle. A Route finding algorithm is used to navigate in an open field or woodland. The innovation of this algorithm is that it does not necessarily create a path between a source and its destination, but ensures that the vehicle covers the entire field area when traveling from the source to its destination

IX. Conclusion

Path finding robot that uses Arduino to send these types of robots to enemy terrain or any appropriate terrain to find the appropriate route. There is a use of ultrasonic sensors and Bluetooth module in our project that would connect to Arduino, so we can send coordinates from our mobile to the robot that will move forward to avoid all obstacles.

X. Acknowledgements

The Co-authors would like to thank Ass. Prof Vikrant A. Agaskar for his Idea and guidance. We would also like to thank all his friends and family, without their enthusiasm this project would not have been possible.

XI. REFERENCE

[1] J.A. Fernandez and J. Gonzalez, The NEXUS open system for integrating robotics software, Robotics and Computer-Integrated Manufacturing, Vol.15, pp. 431 – 440, 1999.
[2] K. Konolige and K. Myers, The Saphira Architecture for Autonomous Mobile Robots, SRI International.
[3] T. Balch, TeamBots, 2000, www.teambots.org.
[4] S. Lenser, J. Bruce, and M. Veloso, CMPack: A Complete Software System for Autonomous Legged Soccer Robots. Proceedings of AGENTS '01, ACM Press, 204-211. 2001
[5] T. Balch and R.C. Arkin, Behavior-based formation control for multi-robot teams. www.cc. gatech.edu/ai/robot-lab/.
[6] M. Mataric, Behavior-based control: Example from navigation, learning, and group behavious. Journal of Experimental and Theoretical AI, 9(2-3), 1997
[7] Josh R. de Leeuw and Kenneth R. Livingston, A Self-Organizing Autonomous Prediction System for Controlling Mobile Robots. International Conference on Automation, Robotics and Control System, pp123-129, 2009