NAVBOT THE AUTONOMOUS ROBOT

Craioveanu David, FONTA Alin-Vasile

Faculty of Industrial Engineering and Robotics, Specialization Robotics, Year of study: I, e-mail: davi.craioveanu@gmail.com
Scientific supervisor: Conf.dr.ing. Andrei Mario IVAN

ABSTRACT: This article presents to presents a robotic system developed for laboratory applications. The system employs computer vision techniques using OpenCv to detect and track objects based on color. The robot's capabilities include object detection, movement control, and obstacle avoidance using ultrasonic sensors.
KEY WORDS: 3D printing, computer vision, 3D design

1. Introduction

The rapid advancement of robotics technology has brought about significant changes in various industries, revolutionizing processes and pushing the boundaries of what is possible. From manufacturing and healthcare to agriculture and exploration, robots have found their way into numerous fields, streamlining operations and augmenting human capabilities. One area where robots have proven particularly valuable is in laboratory work.

Robots in the industrial and laboratory sectors share common characteristics. They are designed to perform tasks autonomously or with minimal human intervention, offering advantages such as increased precision, repeatability, and productivity. In industrial settings, robots have become integral components of manufacturing processes, carrying out assembly tasks, material handling, and quality control. They operate with high speed and accuracy, contributing to improved efficiency and cost-effectiveness.

In laboratory environments, robots play a pivotal role in scientific research, experimentation, and analysis. They are employed in diverse scientific disciplines, including chemistry, physics, biology, pharmaceuticals, and materials science. Lab robots offer several key advantages, including the ability to handle hazardous materials, execute repetitive tasks with consistency, and work in controlled and sterile conditions.

The robot at hand, named "NavBot," has been designed to serve as a versatile assistant in laboratories, catering to various disciplines, from chemistry and physics to small workshops. Its primary objective is to provide valuable support and streamline operations in these environments, offering ease of use and utility.

NavBot incorporates computer vision [1] to facilitate object recognition. Through an integrated graphical user interface (GUI), the robot can identify and track different objects, shapes, or even living organisms, based on predefined parameters. This capability enhances data collection and analysis, expediting research processes and increasing accuracy.

To ensure efficient navigation and obstacle avoidance, NavBot is equipped with an ultrasonic sensor. This sensor enables the robot to measure distances accurately and adapt its path to avoid potential collisions with objects or obstacles. By prioritizing safety, NavBot mitigates risks and minimizes disruptions during experiments and laboratory operations.

Beyond its application in laboratory environments,[2] NavBot has the potential to extend its impact to various fields of work. In pharmaceutical manufacturing, for example, the robot can contribute to tasks such as compound synthesis, high-throughput screening, and quality control. Its precision and consistency enhance productivity and reliability in the pharmaceutical development process.

NavBot's capabilities also find value in biomedical research, where it aids in automation for genomics, proteomics, and drug discovery. By handling large sample volumes and executing repetitive tasks, the robot accelerates research timelines and increases throughput, ultimately contributing to advancements in healthcare.

2. Robot design

In creating this prototype, we also considered environmental concerns by choosing materials carefully. Although we were somewhat limited, we wanted to use materials that were as environmentally friendly as possible. For this reason, the base of the robot is made from an old, reused toy.

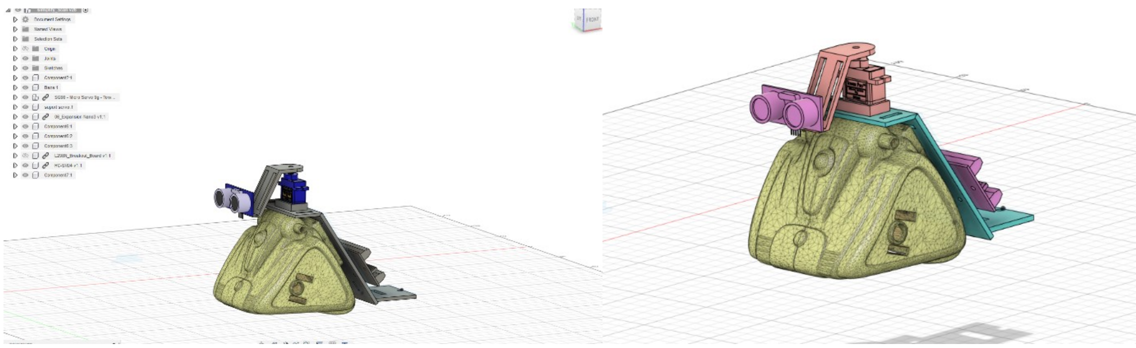

Fig.1 NavBot 3D model

Fig.2 Robot 3D scanning

2.1 3D printing of the robot components

For the realization of the robot components, we used 3D printing technology. This process allowed us to create a 3D model of the device, enabling us to analyze and make adjustments before the final manufacturing.

The component elements were printed using the ANYCUBIC MEGA 3D printer, [3] utilizing PLA (Polylactic Acid) filament. PLA is a popular biodegradable and environmentally friendly material derived from renewable resources such as cornstarch or sugarcane. It is known for its ease of use, low warping, and minimal odor during printing.

The ANYCUBIC MEGA printer has a spacious printing area measuring 300mm x 300mm x 305mm, providing ample space for large-sized components. PLA filament offers excellent print quality, allowing for precise and detailed designs.

To ensure optimal printing results, we set the extrusion temperature to around 200-220°C. Additionally, we adjusted the printing speed to a moderate level of 50 mm/s, striking a balance between print time and quality. These settings allowed us to achieve a smooth and consistent layer-by-layer construction of the components.

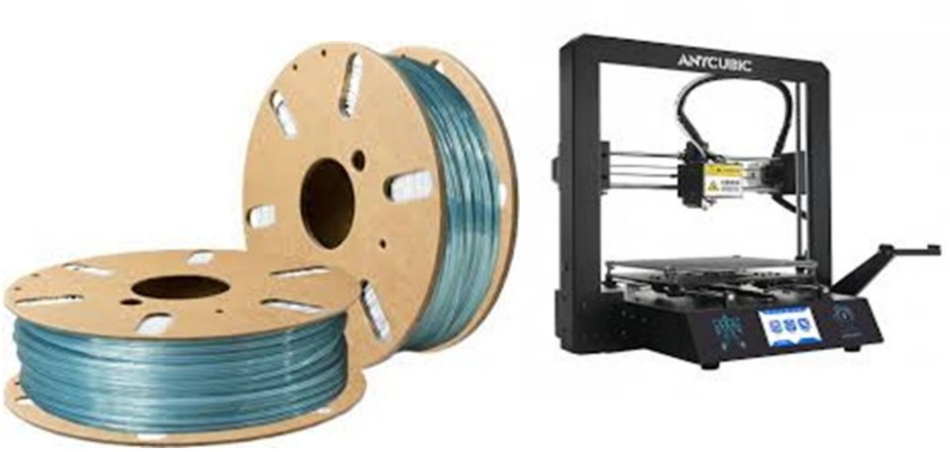

Fig.3 The 3D printer and filament used

2.3 Electrical part of the robot

The robot operates through an Arduino Nano board, to which a servo motor is connected to move the ultrasonic sensor. It also has a Bluetooth module, HC-05, which receives processed images from the computer facilitate a faster operation. The movement is achieved with the help of two DC motors.

Fig.4 Electrical components used

3. The code

This code captures a region of the screen and processes the image received by the laptop from the wireless camera mounted on the robot to detect a certain color. It then identifies the largest object of that color and determines the robot's control action based on its position in the image. If the object is in the center of the image, the robot moves forward. If the object is off-center, the robot turns towards it. If the object is too small, the robot searches for it, and if it is not found, the robot keeps searching.
The code also has a graphical user interface (GUI) window with buttons to end the program or continue running it. It is written in Python uses the OpenCV library to process images, the mss library to capture the screen, the serial library to communicate with the robot, and the tkinter library to create the GUI window.

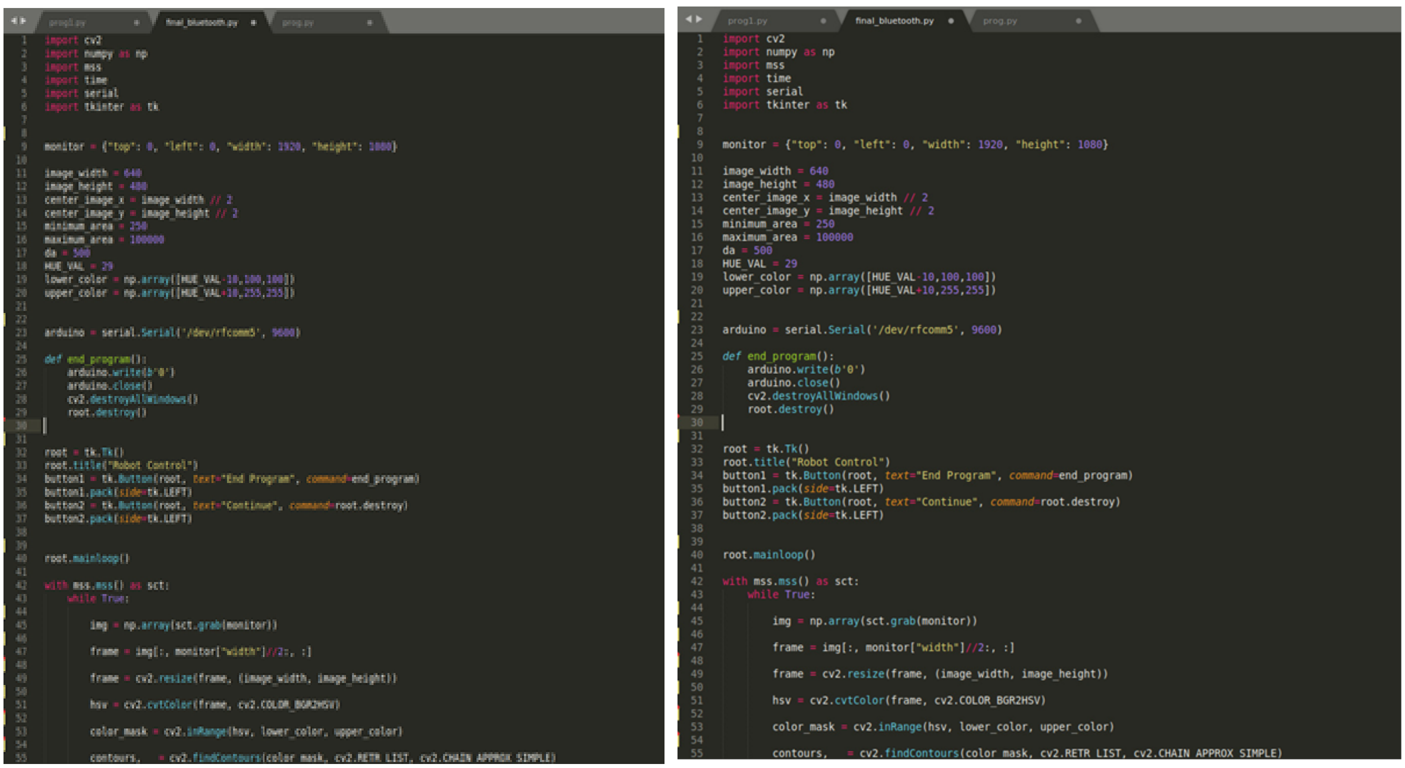

Fig.5 The code that runs on laptop

The Arduino code [4] is quite straightforward:
The code defines pin numbers for a robot's motors and a distance sensor. It sets the pins as inputs or outputs, and starts serial communication with the computer(laptop).
In the main loop, it waits for a command from the computer. If the command is 'f', 'b', 'l', or 'r', the robot moves in the corresponding direction: forward, backwards, left, right. If the command is '0', the robot enters an autonomous mode where it avoids obstacles using the distance sensor.
The distance function measures the distance to an object using the sensor, and the “autonom” function controls the robot's movements in autonomous mode based on the distance measured so that the obstacle could be avoided.

File Edit Sketch Tools Help

program_bluetooth_arduino

```
#include <SoftwareSerial.h>
SoftwareSerial bluetooth(12,11);
int leftMotorPin1 = A1;
int leftMotorPin2 = A0;
int rightMotorPin1 = 4;
int rightMotorPin2 = 5;

int trigPin = A3;
int echoPin = A2;

#define MAX_DISTANCE 20
void setup() {
  pinMode(leftMotorPin1, OUTPUT);
  pinMode(leftMotorPin2, OUTPUT);
  pinMode(rightMotorPin1, OUTPUT);
  pinMode(rightMotorPin2, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  Serial.begin(9600);
  bluetooth.begin(9600);
}

void loop() {
  //Serial.println(distanta());
  //delay(200);
  if (bluetooth.available()) {
    char command = bluetooth.read();
    if(command == '0'){Serial.println("haha intru in mod autonom");
    autonom();
    }
    if (command == 'f') {
      forward();
    } else if (command == 'b') {
      backward();
```

program_bluetooth_arduino

```
  }
  if (command == 'f') {
    forward();
  } else if (command == 'b') {
    backward();
  } else if (command == 'l') {
    left();
  } else if (command == 'r') {
    right();
  } /*else if (distanta() <= 15) {
    oprire();*/
  }
 }*/
}

void forward() {
  digitalWrite(leftMotorPin1, HIGH);
  digitalWrite(leftMotorPin2, LOW);
  digitalWrite(rightMotorPin1, HIGH);
  digitalWrite(rightMotorPin2, LOW);
}

void backward() {
  digitalWrite(leftMotorPin1, LOW);
  digitalWrite(leftMotorPin2, HIGH);
  digitalWrite(rightMotorPin1, LOW);
  digitalWrite(rightMotorPin2, HIGH);
}

void left() {
  digitalWrite(leftMotorPin1, LOW);
  digitalWrite(leftMotorPin2, HIGH);
  digitalWrite(rightMotorPin1, HIGH);
  digitalWrite(rightMotorPin2, LOW);
}
```

File Edit Sketch Tools Help

program_bluetooth_arduino

```
}

void left() {
  digitalWrite(leftMotorPin1, LOW);
  digitalWrite(leftMotorPin2, HIGH);
  digitalWrite(rightMotorPin1, HIGH);
  digitalWrite(rightMotorPin2, LOW);
}

void right() {
  digitalWrite(leftMotorPin1, HIGH);
  digitalWrite(leftMotorPin2, LOW);
  digitalWrite(rightMotorPin1, LOW);
  digitalWrite(rightMotorPin2, HIGH);
}

void oprire() {
  digitalWrite(leftMotorPin1, LOW);
  digitalWrite(leftMotorPin2, LOW);
  digitalWrite(rightMotorPin1, LOW);
  digitalWrite(rightMotorPin2, LOW);
}
int distanta(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH);
  float distance = microsecondsToCentimeters(duration);

  if(distance<100){return distance;}
  }
long microsecondsToCentimeters(long microseconds){
```

File Edit Sketch Tools Help

program_bluetooth_arduino §

```
  delayMicroseconds(5);
  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH);
  float distance = microsecondsToCentimeters(duration);

  if(distance<100){return distance;}
  }
long microsecondsToCentimeters(long microseconds){
  return microseconds / 29 / 2;}

  void autonom(){
  if (distanta() > 0 && distanta() <= MAX_DISTANCE) {
    Serial.println("oprire");
    oprire();
    delay(500);


    Serial.println("Left");
    left();
    delay(1000);


    Serial.println("forward");
    forward();
  } else {
    Serial.println("forward");


    forward();
  }


    }
```

Fig.6 The code that runs on Arduino

4. Conclusion

In conclusion, robots like NavBot have transformed laboratory environments by bringing automation and efficiency to scientific research and experiments. Their ability to recognize objects, navigate with precision, and perform tasks independently makes them valuable assets in multiple disciplines. As technology continues to advance, robots like NavBot will play an increasingly vital role in driving scientific discoveries, enhancing productivity, and shaping the future of laboratory work.

5. Bibliography

[1]. Howse, J. (2013), OpenCV Computer Vision with Python , Packt
[2]. Hurst, J. şi Mortimer, J. (1997), Laboratory Robotics: A Guide to Planning, Programming and Applications
[3] *** *Imprimanta 3D ANYCUBIC MEGA* https://www.eshop.formwerk.ro/cumpara/anycubic-i3-mega-s-1604?utm_source=portal&utm_medium=web&utm_campaign=google_xml&gad=1&gclid=EAIaIQobChMI693ypYrV_gIVygCiAx3DZQLwEAQYASABEgLiafD_BwE
[4]. Magda, Y. (2018), Advanced Programming For Arduino Geeks