

ARDUINO HUMAN FOLLOWING ROBOT USING PYTHON

¹MOHAMMED AZHARUDDIN, ²Dr. MOHAMMED ABDUL BARI

¹B.E STUDENT, ²ASSOCIATE PROFESSOR AND HOD

DEPARTMENT OF CSE

ISL ENGINEERING COLLEGE, BANDLAGUDA

Abstract: Humanoid robotics is an emerging research field that has received significant attention during the past years and will continue to play an important role in robotics research and many applications of the 21st century and beyond. In this rapid moving world, there is a need of robot such a “A Human Following Robot” that can interact and co-exist with them. Because of its human following capability, these robots can work as assistants for humans in various situations and it can also acquire or monitor certain information associated with the human subject. In this paper we present a prototype that uses Arduino Uno along with basic sensors such as ultrasonic and IR sensor. All the processing is carried out by the microprocessor while the control of the motors is carried out by the controller. This robot can further be modified by using many technologies such as Bluetooth, PixyCamera etc.

Keywords: Artificial Intelligence, Human following, Human tracking, Ultrasonic Sensor, IR Sensor, Arduino Micro Controller.

I. INTRODUCTION

Robotic technology has increased appreciably in past couple of years. Such innovations were only a dream for some people a couple of years back. But in this rapid moving world, now there is a need of robot such as “A Human Following Robot” that can interact and co-exist with them. The development of robot technology had increased significantly due to industrial, medical and military applications. In various fields with harsh environment such as underground mining, war-zones, medical, construction, space exploration etc. the work done by one is extremely dangerous. Life of individuals assisting are also put at risks. Tasks performed by humans have its own limitations in many ways. In order to perceive beyond the

human limitation in vision, speed, consistency, flexibility, quality e.tc we should make use of robots. A key requirement for these robots is the ability to detect humans and to interact with the min non-technical way. The main objective of this dissertation is to make a robot that can help humans with various tasks.. In this paper, we present a prototype of a human following robot that uses Arduino Uno and different sensors for detection and following an object. The Robot must follow the following objectives.

- The robot must be capable of accurately follow a person.
- It should be capable of taking various degrees of turns.
- The robot must be insensitive to environmental factors such as noise.
- The robot must be capable to avoid collision.

II. SYSTEM CIRCUIT DIAGRAM AND COMPONENTS

2.1 Circuit Diagram

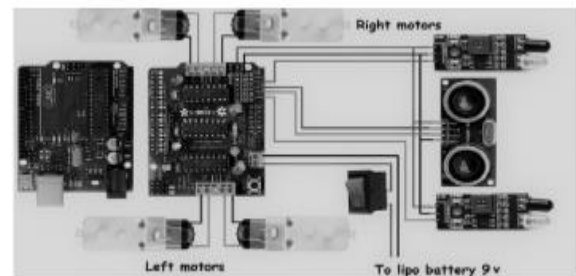


Figure 1: Human Following Robot Circuit

2.2 System Components

2.2.1 ArduinoUno

It is the brain of our project. It can give all the command to their sub ordinate components which should by operated by the human behavior . And

it also give feedback to the other components and human. So that it can be the used as a medium of communication between human and robots & vice versa. . It has specification of 8 bit CPU, 16 MHZ clock speed, 2 KB SRAM 32 KB flash Memory, 1 KB EEPROM.

2.2.2 DC Motors

DC Motor is a device that converts any form of energy into mechanical energy or imparts motion. In constructing a robot, motor usually plays an important role by giving movement to the robot. Here 4 DC motor are used to drive the robot.

2.2.3 Motor Shield

The Motor Shield is a driver module for motors that allows you to use Arduino to control the working speed and direction of the motor. . The Motor Shield can either be powered by Arduino directly or by an external 6V~15V power supply via the terminal input. Here Motor Driver Board is designed to Work with L293D IC.

2.2.4 Ultrasonic sensor

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. The working principle of this module is simple, it sends an ultrasonic pulse out at 40kHz which travels through the air and if there is an obstacle or object, it will bounce back to the sensor. By calculating the travel time and the speed of sound, the distance can be calculated.

2.2.5 IR Sensor

IR sensor is an electronic device, that emits the light in order to sense some object of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. These types of radiations are invisible to our eyes, but infrared sensor can detect these radiations. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode.

III. WORKING AND DESIGN

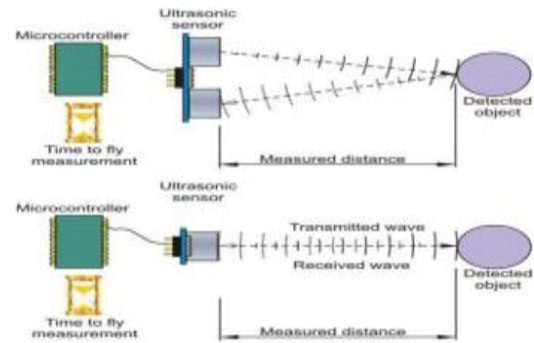


Figure 2: Ultrasonic Sensor Principle

3.1 Ultrasonic and IR Sensor Principle

This ultrasonic sensor is placed at the top of robot and pair of IR sensors are attached on either side of the ultrasonic sensor. We used ultrasonic sensor for obstacle avoidance and to maintain a specific distance for the object. The ultrasonic sensor works accurately works accurately within a range of 4 meters. Ultrasonic sensors operate by calculating the times differences. Infrared sensors detect the object's distance with infrared radiations when the beam from transmitter detects an object it returns to the receiver with an angle after reflection also known as method of triangulation this also helps in calculation of distance travelled by robot and eliminate any further error in the robotic movement due to displacement. IR sensor controls the movement of motors and ultrasonic sensor detects the obstacle and stops the motors.

IV. RESULTS

Different experiments were conducted and the performance of the human following robot was tested. Test was performed on the ultrasonic and infrared sensor. It was noted that the sensor was working accurately within a range of 4 meters. Then we performed the test to check whether the robot maintains a specific distance with the target object. Then we checked the serial communication between Arduino, motor shield and various motors. On the basis of results obtained from these tests and experiments, we made the necessary changes in the processing and control algorithm. After the completion, we observed that the results produced were very satisfying the robot was perfectly following the

person wherever it goes. Hence the objective of implementing a good Human-Robot interaction was achieved.

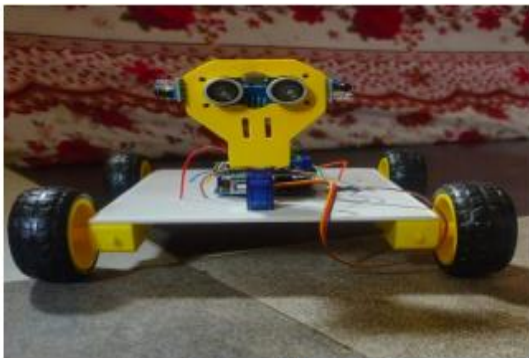


Figure 3: Human Following Robot

V. APPLICATIONS

Looking deeply into environment or our surroundings, we were able to interpret that there is a need for such a robot that can assist humans and can serve them. Such a robot can be used for many purposes. With a few modifications, the robot can act as a human companion as well. The tasks these kinds of robots can perform are limitless, including assisting in carrying loads for people working in hospitals, libraries, airports, etc.

VI. FUTURE WORK

There are many interesting applications of this research in different fields, whether military or medical. A wireless communication functionality can be added to the robot to make it more versatile and control it from a large distance. This capability of a robot could also be used for military purposes. By mounting a real-time video recorder on top of the camera, we can monitor the surroundings by just sitting in our rooms. We can also add some modifications to the algorithm and the structure as well to fit it for any other purpose. Similarly, it can assist the public in shopping malls. So there it can act as a luggage carrier, hence no need to carry up the weights or to pull that. Similarly, ample amount of modifications could be done to this prototype for far and wide applications.

VII. CONCLUSION

A successful implementation of a prototype of a human-following robot is illustrated in this paper. This robot does not only have the detection capability but also the following ability as well. While making this prototype, it was also kept in mind that the functioning of the robot should be as efficient as possible. Tests were performed under different conditions to pinpoint the mistakes in the algorithm and to correct them. The different sensors that were integrated with the robot provided an additional advantage. The human-following robot is an automobile system that has the ability to recognize obstacles, move, and change the robot's position toward the subject in the best way to remain on its track. This project uses an Arduino, different types of sensors to achieve its goal. This project challenged the group to cooperate, communicate, and expand their understanding of electronics, mechanical systems, and their integration with programming.

REFERENCES

- [1]. K. Morioka, J.-H. Lee, and H. Hashimoto, "Human-following mobile robot in a distributed intelligent sensor network," *IEEE Trans. Ind. Electron.*, vol. 51, no. 1, pp. 229–237, Feb. 2004.
- [2]. Y. Matsumoto and A. Zelinsky, "Real-time face tracking system for human-robot interaction," in *1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings, 1999*, vol. 2, pp. 830–835 vol.2.
- [3]. T. Yoshimi, M. Nishiyama, T. Sonoura, H. Nakamoto, S. Tokura, H. Sato, F. Ozaki, N. Matsuhira, and H. Mizoguchi, "Development of a Person Following Robot with Vision Based Target Detection," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006*, pp. 5286–5291.
- [4]. H. Takemura, N. Zentaro, and H. Mizoguchi, "Development of vision based person following module for mobile robots in/out door environment," in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2009*, pp.

[5]. Muhammad Sarmad Hassan, MafazWali Khan, Ali Fahim Khan,"Design and Development of H

uman Following Robot", 2015,Student Research Paper Conference,Vol-2, No-15.

[6]. N. Bellotto and H. Hu, "Multisensor integration for human-robot interaction," IEEE J. Intell. Cybern.Syst., vol. 1, no. 1, pp

CODE

```
#!/usr/bin/env python
from __future__ import print_function
import xbox
from Arduino import Arduino
import time
# Define Arduino variables
servo1 = 9
servo2 = 6
servo3 = 5
servo4 = 3
servo5 = 11
IN1 = 2
IN2 = 4
IN3 = 7
IN4 = 8
ENA = 10
ENB = 12
board = Arduino()
board.Servos.attach(servo1)
board.Servos.attach(servo2)
board.Servos.attach(servo3)
board.Servos.attach(servo4)
board.Servos.attach(servo5)
board.pinMode(IN1, "OUTPUT");
```

```
board.pinMode(IN2, "OUTPUT");
board.pinMode(IN3, "OUTPUT");
board.pinMode(IN4, "OUTPUT");
board.pinMode(ENA, "OUTPUT");
board.pinMode(ENB, "OUTPUT");
board.pinMode(13, "OUTPUT")
angle1 = 90
angle2 = 90
angle3 = 90
angle4 = 90
angle5 = 0
# Format floating point number to string format -
x.xxx
def fmtFloat(n):
    return '{:6.3f}'.format(n)
# Print one or more values without a line feed
def show(*args):
    for arg in args:
        print(arg, end="")
# Print true or false value based on a boolean,
without linefeed
def showIf(boolean, ifTrue, ifFalse=" "):
    if boolean:
        show(ifTrue)
    else:
        show(ifFalse)
# Move servo with respect to the left analog stick
X values limited to the min-max angle set
def leftXmove(servonum,minAngle,maxAngle):
    angle = angle1
    if int(joy.leftX()) > 0:
        angle = angle - 2
```

```

if angle <= minAngle:
    angle = minAngle
return angle
elif int(joy.leftX()) < 0:
    angle = angle + 2
if angle >= maxAngle:
    angle = maxAngle
return angle
else:
    return angle

# Move servo with respect to the left analog stick
X values limited to the min-max angle set
def leftYmove(servonum,minAngle,maxAngle):
    angle = angle2
if int(joy.leftY()) > 0:
    angle = angle - 2
if angle <= minAngle:
    angle = minAngle
return angle
elif int(joy.leftY()) < 0:
    angle = angle + 2
if angle >= maxAngle:
    angle = maxAngle
return angle
else:
    return angle

# Move servo with respect to the right analog
stick Y values limited to the min-max angle set
def
rightYmove(servonum,minAngle,maxAngle):
    angle = angle3

```

```

if int(joy.rightY()) > 0:
    angle = angle - 2
if angle <= minAngle:
    angle = minAngle
return angle
elif int(joy.rightY()) < 0:
    angle = angle + 2
if angle >= maxAngle:
    angle = maxAngle
return angle
else:
    return angle

# Move servo with respect to the right analog
stick X values limited to the min-max angle set
def
rightXmove(servonum,minAngle,maxAngle):
    angle = angle4
if int(joy.rightX()) > 0:
    angle = angle - 2
if angle <= minAngle:
    angle = minAngle
return angle
elif int(joy.rightX()) < 0:
    angle = angle + 2
if angle >= maxAngle:
    angle = maxAngle
return angle
else:
    return angle

# Move servo when A button is pressed
def buttonA(servonum,minAngle,maxAngle):
    angle5 = minAngle

```



```
if int(joy.A()) > 0:
    angle5 = maxAngle
    board.Servos.write(servonum,angle5)
else:
    board.Servos.write(servonum,minAngle)
# Move motors when right trigger is used
def rightTrigmove():
    if int(joy.rightTrigger()) > 0:
        board.digitalWrite(IN1,"LOW")
        board.digitalWrite(IN2,"HIGH")
        board.digitalWrite(IN3,"LOW")
        board.digitalWrite(IN4,"HIGH")

# Move motors when left trigger is used
def leftTrigmove():
    if int(joy.leftTrigger()) > 0:
        board.digitalWrite(IN1,"HIGH")
        board.digitalWrite(IN2,"LOW")
        board.digitalWrite(IN3,"HIGH")
        board.digitalWrite(IN4,"LOW")

# Move motors when left bumper is used
def leftBumper():
    if int(joy.leftBumper()) > 0:
        board.digitalWrite(IN1,"HIGH")
        board.digitalWrite(IN2,"LOW")
        board.digitalWrite(IN3,"LOW")
        board.digitalWrite(IN4,"HIGH")

# Move motors when right bumper is used
def rightBumper():
    if int(joy.rightBumper()) > 0:
        board.digitalWrite(IN1,"LOW")
```

```
board.digitalWrite(IN2,"HIGH")
board.digitalWrite(IN3,"HIGH")
board.digitalWrite(IN4,"LOW")

# Instantiate the controller
joy = xbox.Joystick()

# Show various axis and button states until Back
button is pressed

print("Xbox controller sample: Press Back button
to exit")

while not joy.Back():
    # Poll controller leftX values and move servo1
    accordingly
    angle1 = leftXmove(servo1,0,180)
    board.Servos.write(servo1,angle1)
    # Poll controller leftY values and move servo2
    accordingly
    angle2 = leftYmove(servo2,0,120)
    board.Servos.write(servo2,angle2)
    # Poll controller rightY values and move
    servo3 accordingly
    angle3 = rightYmove(servo3,0,180)
    board.Servos.write(servo3,angle3)
    # Poll controller rightX values and move
    servo4 accordingly
    angle4 = rightXmove(servo4,0,180)
    board.Servos.write(servo4,angle4)
    # Move claw conected to servo5 when A is
    pressed
    buttonA(servo5,0,90)
    # Poll right trigger and move motors forward
    rightTrigmove()
    # Poll left trigger and move motors backward
    leftTrigmove()
    # Poll right bumper and turn right
```

```
rightBumper()

# Poll left bumper and turn left

leftBumper()

    if      (int(joy.rightTrigger()) |
int(joy.leftTrigger()) | int(joy.rightBumper()) |
int(joy.leftBumper()) == 0):

        board.digitalWrite(IN1,"LOW")

        board.digitalWrite(IN2,"LOW")

        board.digitalWrite(IN3,"LOW")

        board.digitalWrite(IN4,"LOW")

# Close out when done

joy.close()
```

Journal of Engineering Sciences