Human Following Robot Project Report

A Comprehensive Report on the Development of an Intelligent Human Following Robot System

Awais Asghar

School of Electrical Engineering and Computer Science, NUST H-12, Islamabad

427265

Muhammad Abdullah

School of Mechanical and Manufacturing Engineering, NUST H-12, Islamabad

408693

Abstract

The design, development, and analysis of a smart robot companion that can track people are communicated in depth in this project report. The project's goal was to create a dependable and adaptable robot that could track and follow a human operator on its own in a variety of settings. The robot's abilities were accomplished by a mix of software development, hardware integration, and system operation. The project's goals, methodology, major discoveries, and implications for next studies and innovations in human-robot interaction are all described in the report.

Introduction

Research and advancement on human-robot interaction has become increasingly important, with applications ranging from personal help to industrial automation. Of all the interaction modes, the capacity of a robot to follow and accompany a human operator on its own has attracted the most attention because of its potential applications in a variety of contexts, including entertainment, navigation, and surveillance. The project's main objectives in this regard were the development, application, and assessment of a human-following robot system. The goals, justification, and scope of the project are briefly summarized in this introduction, which also sets the ground for a full examination of the robot's development and results.

Literature Review

1. The Discipline of humanoid robotics is developing abruptly due to the invention of Arduino Uno-power-driven human-following robots and infrared sensors for interaction and celestial navigation. Possible developments to improve trailing and connectivity include Bluetooth and Pixy Camera, to literature. As these technologies become more and more amalgamated into daily life, ethical concerns including employment displacement and privacy must be addressed in future study. [1]

2. To improve operating efficiency, self-sufficient follower robot advancement, especially in medical and military applications, makes use of technologies including servo motors, ultrasonic sensors, and Arduino Uno. These robots can perform task-specific functions without the need for direct human involvement because they use real-time data from sensors to uphold optimal closeness and navigate situations. [2]

3. The advancement of self-directed robots depends on their capability to interact and communicate with humans, which is why techniques for tracking and following selected targets are being investigated. [3]

4. The development of adaptable robots driven by Arduino Uno microcontrollers and formulated with infrared and ultrasonic sensors holds prospective functions in several industries, such as logistics and healthcare. These robots execute jobs with more accuracy and efficiency, which has the ability to transform robotics trends and efforts in the future. [4]

5. Tracking and following certain targets is being researched since the development of self-reliant robots' hinges on their capacity to attract and converse with people. [5]

Components Used

The Components used in the project, there description and detail are given below:

1. L298N H-Bridge Motor Driver Module

The twin H-Bridge motor driver L298N enables real-time direction and speed control of DC motors. DC motors with a peak current of up to 2A and a voltage between 5 and 35V can be driven by this module. Two screw terminal sections are situated for the motors on the module, along with a further screw terminal block that houses the Ground pin, the VCC pin for the motor, and a 5V pin that can be used as an input or output.

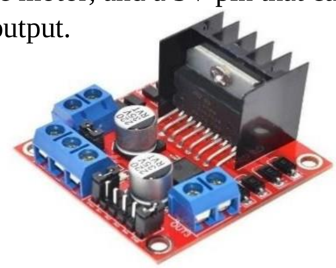

Figure 1 L298N H-Bridge Motor Driver

The voltage applied at the motor's VCC decides this. A jumper can be used to enable or disable the module's built-in 5V regulator. We may turn on the 5V regulator and use the 5V pin as an output, such as to power our Arduino board, if the motor supply voltage is up to 12V. Nevertheless, we must release the jumper if the motor voltage is higher than 12V, as those values will harm the onboard 5V regulator. Subsequently, the IC needs to be powered by a 5V source to function, the 5V pin will be used as the input in this case. [1]

2. IR Infrared Obstacle Avoidance Sensor Module for Arduino

The integrated infrared transmitter and receiver of the infrared obstacle sensor module detects the presence of obstacles in front of it by producing infrared light and looking for reflected light. Users can modify the detecting range of the module using the potentiometer that is integrated into it. The sensor responds very well and progressively in both total darkness and ambient light. There are two common types of obstacle avoidance sensors: those with three and four pins. [5]

Figure 2 IR Infrared Sensor

Working Principle: When an object has a reflecting surface, such a white color, the infrared signal from the IR transmitter bounces off in many directions, incorporating the direction of the IR recipient, which collects the signal and detects the object. The sensor cannot identify an object when the surface is absorbent because the IR signal is not reflected. This outcome would still happen in the absence of the object. The output, GND, and VCC are the three pins on the module. Additionally, we have two LEDs: one is for power, and when it is linked to electricity, it goes on. The other is used to detect obstacles. [5]

3. Arduino UNO

The Arduino Uno is an open-source microcontroller board that was created by Arduino.cc and first made available in 2010. It is based on the Microchip ATmega328P memory chip (MCU). Sets of digital and analogue input/output (I/O) pins on the microcontroller board permit it to be interfaced with different extension boards (shields) and other circuits.

Figure 3 Arduino UNO

The board may be programmed using the Arduino IDE (Integrated Development Environment) and a type B USB cable. It includes six analogue and fourteen digital I/O pins, six of which can be used for PWM output. It can be powered by a rectangular 9-volt battery or a barrel connector that takes voltages ranging from 7 to 20 volts. It shares the same headers as the Leonardo board and the same microprocessor as the Arduino Nano board. The Arduino website has the hardware reference design, which is made accessible under a Creative Commons Attribution Share-Alike 2.5 license. There are also available layout and production files for certain hardware variants. [5]

4. Arduino 4WD Smart Robot Car Chassis Kit with DC Motor

An easy-to-assemble and operate robot chassis platform is the 4-Wheel Robot Chassis Kit. Everything you need to give your robot a quick four-wheel-drive platform with lots of opportunity to grow and add more sensors and controllers is included in the Arduino chassis kit. All you need to do is plug in your motor driver and Arduino or Raspberry Pi to begin programming your robot. This clever robot automobile has a roomy inside with pre-drilled holes for adding sensors and electronics in any configuration you want. With this robot chassis, you may speed up developing a robot by quickly preparing your mechanical base in a matter of minutes.

Figure 4 (4WD) Smart Robot Car Chassis

The most often used robot platforms are wheeled robots because they are simple to operate, maintain, and use. This kit is the most basic robot platform since it is easy to assemble and program. [5]

5. 12-volt Rechargeable Cells (3 cells 4 volts each)

These are rechargeable batteries with a voltage rating of 4 volts each, totaling 12 volts when connected in series. They are commonly used as power sources for electronic devices, robots, and vehicles requiring a higher voltage supply. The rechargeable nature of the cells allows for repeated use and recharging, making them ideal for portable and mobile applications. [1]

Figure 5 12-volt Rechargeable Cells

Code:

The code was first developed according to the requirements and then additional modifications were performed in order to ensure the proper working and intelligent performance of the robot. The Arduino Uno (Rev-3) was used as the microcontroller and a C++ code was developed to connect different parts and sensors with the motor. Arduino acted as the central processing unit which collected data from sensors and then sent output signals to the motors to perform actions as described by the code. So, in order to perform certain actions by the robot a detailed set of instructions must be given to Arduino which could then implement those instructions and a proper interaction between hardware and software is achieved. [4]

Syntax:

The syntax of the code developed in C++ language is given below. (Note this code is specific for the Human Following Robot).

```
// Define constants for the robot
#define Forward_Speed 100
#define Turn_Speed 100

#define Left_IR 3
#define Right_IR 2
#define Middle_IR 4

// Pins for left motor
int enableLeftMotor = 10;
int leftMotorPin1 = 9; //In2 8
int leftMotorPin2 = 8; //In1 9

// Pins for right motor
int enableRightMotor = 5;
int rightMotorPin1 = 7; //In4 7
int rightMotorPin2 = 6; //In3 6

// Setup function
void setup() {

  // Initializing serial communication
  Serial.begin(9600);

  // Setting motor pins as outputs
  pinMode(enableRightMotor, OUTPUT);
  pinMode(rightMotorPin1, OUTPUT);
  pinMode(rightMotorPin2, OUTPUT);
  pinMode(enableLeftMotor, OUTPUT);
  pinMode(leftMotorPin1, OUTPUT);
  pinMode(leftMotorPin2, OUTPUT);


  // Setting IR sensor pins as inputs
  pinMode(Right_IR, INPUT);
  pinMode(Left_IR, INPUT);
  pinMode(Middle_IR, INPUT);
  // Stopping motors initially
  stopMotors();

}

// Main loop function
void loop() {

  // Reading values from IR sensors
  int rightIRSensorValue = digitalRead(Right_IR);
  int leftIRSensorValue = digitalRead(Left_IR);
  int Middle_IR_Value = digitalRead(Middle_IR);
  // Controlling the robot based on sensor inputs
```

```
  if (rightIRSensorValue == LOW && leftIRSensorValue == HIGH) {
    turnRight();//Right Sensor Senses Human
  } else if (rightIRSensorValue == HIGH && leftIRSensorValue == LOW) {
    turnLeft();//Left Sensor Senses Human
  } else if (Middle_IR_Value==LOW) {
    goForward();//Middle Sensor Senses Human
  } else if ( leftIRSensorValue == LOW && rightIRSensorValue == LOW ){
    goback();
  }
   else {
    stopMotors();
  }
}
void goback(){
  digitalWrite(rightMotorPin1, LOW);
  digitalWrite(rightMotorPin2, HIGH);
  digitalWrite(leftMotorPin1, LOW);
  digitalWrite(leftMotorPin2, HIGH);
  analogWrite(enableRightMotor, 75);
  analogWrite(enableLeftMotor, 75);
}
// Function to control both motors to go forward
void goForward() {
  digitalWrite(rightMotorPin1, HIGH);
  digitalWrite(rightMotorPin2, LOW);
  digitalWrite(leftMotorPin1, HIGH);
  digitalWrite(leftMotorPin2, LOW);
  analogWrite(enableRightMotor, Forward_Speed);
  analogWrite(enableLeftMotor, Forward_Speed);
}

// Function to turn left
void turnLeft() {
  digitalWrite(rightMotorPin1, HIGH);
  digitalWrite(rightMotorPin2, LOW);
  digitalWrite(leftMotorPin1, LOW);
  digitalWrite(leftMotorPin2, HIGH);
  analogWrite(enableRightMotor, Turn_Speed);
  analogWrite(enableLeftMotor, Turn_Speed);
}

// Function to turn right
void turnRight() {
  digitalWrite(rightMotorPin1, LOW);
  digitalWrite(rightMotorPin2, HIGH);
  digitalWrite(leftMotorPin1, HIGH);
  digitalWrite(leftMotorPin2, LOW);
  analogWrite(enableRightMotor, Turn_Speed);
  analogWrite(enableLeftMotor, Turn_Speed);
}

// Function to stop both motors
void stopMotors() {
  analogWrite(enableRightMotor, 0);
  analogWrite(enableLeftMotor, 0);
}
```

Figure 6: Code to perform Operations.

Description:

In the first section of code the different pins of infrared sensors and motor drivers are initialized with specific integer values these values corresponds with the actual number of pins in the Arduino Uno board to which each specific sensor and motor driver pins are connected. In the Void Setup() function each pin specified above is described with its pin mode i.e., whether output or input. In the Void Loop() function first sensors are asked to read the input and then based on the input values different conditions are implemented (if-else statements are used) based on these conditions certain output actions are asked to be performed by the motor driver (turn left or right).

In general, if the left sensor reads some value at input the robot will turn left, if it reads some value at the right sensor the car will turn right, if it reads input value in the middle sensor car will move forward and if both left and right sensors are activated at the same time car will move backward.

1. First, #define function is used which means the value of variable given by using this function cannot be changed later in the code. Then, different variables are given different integer values same as given in the Arduino board.
2. Void Setup() function, in this function the modes of pins and different integer or float values are initialized, pinmode(variable-name,Mode) this is used to give a specific role to each variable. e.g., infrared sensor sends input values so its mode should be INPUT.
3. Digitalread() function is used in the void loop() section which reads the value from the input sensor but there can be only two possibilities HIGH or LOW, it will only give a digital value.
4. If-Else statements are later used which are very famous in programming languages if a certain condition is met a certain action is performed otherwise a different action can also be performed. Many different possibilities from if else statements are also used nowadays e.g., nested if-else or elseif statements.
5. In the last section of code different user-defined functions are made which are used in the if else statements e.g., goback(), goforward(), turnleft() etc. these functions are developed according to the digital values given to the pins of motor driver for that purpose digitalwrite() function is used which gives a digital value high or low to the pin.
6. Another function analogwrite (variable,PWM-value) is also used this used uses pulse width modulation (PWM) technique to give a digital output in analog form by changing the width or period of signal.

Pictures:

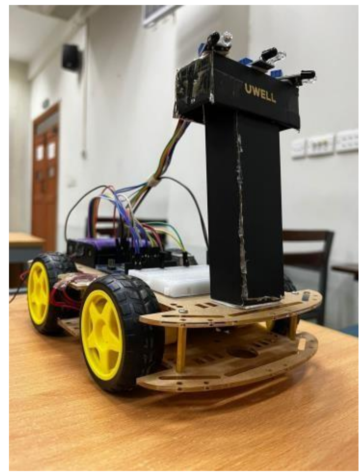

Figure 7: Front View

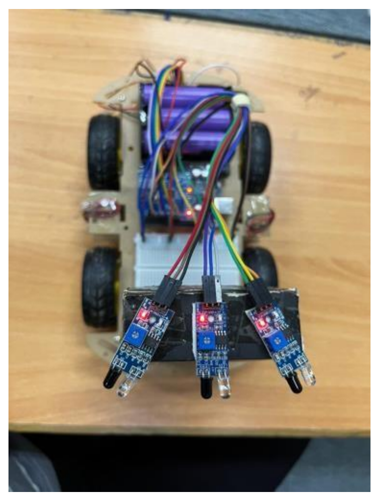

Figure 8: Top View

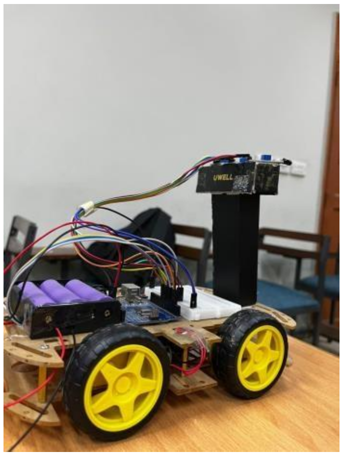

Figure 9: Side View

Methodology:

The following procedure and approach was adopted to fully develop an intelligent human following robot.

1. Plan: The general overview and approach to developing the robot was discussed in detail and the apparatus required were listed down and ordered. [1]
2. Apparatus Assembly (Hard): The parts of the card or the hard structure of the robot was then made from wood shop and it was assembled in a car like structure. Different sections in the lower portion of the car were designed for motor driver.
3. Dc motors setup: Dc motors were connected to rubber tyres, 4 tyres connected with 4 Dc motors were made and then assembled in the wood structure. Motor driver was also installed in the structure.
4. Sensors Setup: 3 infrared sensors were installed in the above section of the car the sensor at the left was tilted in the left direction so it only reads values from the cars left view and right sensor was also tilted in the right direction.
5. Wiring Setup: Wires from each sensor and motor drivers were integrated with the Arduino Uno pins wires were connected according to already decided wiring scheme and PWM pins were connected with the enablers of the motor driver with helped in controlling speed of the motors. 5V and ground connections were provided using the breadboard and the voltage supply for the motors was given via motor driver.
6. Apparatus Inspection: After the apparatus was complete a final inspection was done to ensure all the components and wiring connection are in place and the wires were also arranged in somewhat organized way.
7. Code Uploading: The already developed code was then copied to the Arduino Compiler, Arduino was connected to Laptop and board connection was provided in the compiles and after checking errors in the code it was finally uploaded.
8. Working Inspection: Finally, the Human Following Robot was checked whether it is functioning properly or not. Some modifications were later made like changing motor speed using PWM function and adding an additional function in the code which helped the car to also go back. [3]

Discussions:

The development of different self-operating robots is because of microcontrollers and different sensors and output giving components. Generally, all robots work on the same principle. Data is collected and it is analyzed according to certain conditions and later defined actions are performed. The main key point of such robots is that they can perform on their own without any maintenance. These different task performing robots are developed because of advancement in software hardware interaction techniques and better codes. However, these robots also have some limitations. They can be too expensive to design so their feasibility might be a question mark. There are talks around the globe that the tasks performed by human's labor might get replaced by robots consuming a lot of jobs. These robots are highly predictable they cannot perform under circumstance where immediate and different action is required instead, they can only perform according to instructions given to them. [4]

However, some of the limitations can be catered using modern Artificial Intelligence and Machine learning techniques. Our project for the human following robot is only a prototype for next generations of robots certain actions are performed by it implementing the input data and giving output according to requirement. This project helped a lot in making us familiar with different electronic components like I-R sensor, Motor driver etc. We also learned that slight changes in apparatus and code can lead us to make a different action performing robot like Line Following Robot.

Conclusion:

Project for the Human Following Robot increased our understanding of different sensors and components. By developing a fully working robot from scratch we gained valuable insights in how to properly compile any apparatus and how to fully develop a code. Different functions of code in C++ language were learnt during the project syntax for the code was developed from scratch which boosted our programing skills. We became familiar with the Arduino Uno, a microcontroller which will be used greatly in our future project and even in the FYP, different pins and digital-analog functions of Arduino were learn while working on this project. In short, as students this simple project took our understanding of electronics and coding to a whole new level.

References:

S. A. R. P. A. S. Dhiren Sati, "IJARSCT," April 2021. [Online]. Available: https://ijarsct.co.in/Paper1025.pdf.

M. T. Gandla, "Human Following Robot," 8 October 2021. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3917847.

S. Hassan, "Design and development of human following robot," July 2015. [Online]. Available: https://www.researchgate.net/publication/354631715_Design_and_development_of_human_following_robot.

T. R. W. B. L. C. Kharga Bahadur Kharka, "HUMAN FOLLOWING ROBOT USING ARDUINO UNO," July 2021. [Online]. Available: https://www.irjmets.com/uploadedfiles/paper/volume3/issue_7_july_2021/15119/1628083587.pdf.

S. M. A. P. A. K. P. A. V. R. Sonali Gaikwad, "Human Following Robot using Arduino," April 2023. [Online]. Available: https://ijarsct.co.in/Paper9186.pdf.