Journal of Simulation and Analysis of Novel Technologies in Mechanical Engineering
15 (3) (2023) 0049~0060

Research article

Designing a robot to follow a command from the analysis of the image received from the operator

Majid Amiri[1]*, Shayan Farokhi Aleh Kouhi[2], Ahmad Keshavarzi[3]

[1] *Department of Mechanical Engineering, University of Malek-e-Ashtar, Isfahan, Iran.*
[2] *Department of Mechanical Engineering, University of Shahid Mohajer, Isfahan, Iran.*
[3] *Department of Mechanical Engineering, Khomeinishahr Branch, Islamic Azad University, Isfahan, Iran*

*amiri_iut@yahoo.com

(Manuscript Received --- 27 Aug. 2023; Revised --- 28 Oct. 2023; Accepted --- 04 Nov. 2023)

Abstract

It is necessary to design a robot which can be operated without the physical presence of humans. The goal of this paper is to launch a robot that is in the control station far from the user. This can be achieved using a technique called "image processing". One way to solve this problem is to insert commands on a wireless robot using hand gestures to develop a user-friendly interface. The motion detection system works in three stages including image capture, feature extraction, and algorithmic decision-making. Once the gesture is detected, a command signal is generated and sent to the microcontrollers. Then, they are sent to the robot to make it run in the desired direction. Robots, for one application, are used to operate in situations where human intervention is impossible. These robots have a microcontroller-based system called the pick-and-place arm. This mechatronic arm moves objects from a source location to the desired location. The process is controlled with a keyboard. These robots are equipped with RF cameras to record videos at 2.4 GHz. The user can watch the live video at the base station using a computer system. It is tested in the MAT software, and the video is processed in the DSP code simultaneously.

Keywords: Digital image processing, Hand gesture recognition, Robot, MATLAB.

1- Introduction

Human-computer interaction plays a pivotal role in shaping our everyday. This interaction involves designing, evaluating, and implementing interactive computing systems for human use. Keyboards, mice, light pens, etc. are the most popular communication devices used in the everyday life of humans. These devices, though popular, are not natural ways of communication.

With the development of technology, computers nowadays can see through cameras. This makes human-computer interaction even stronger. A newly developed communication and control system is used in this robotic system that is reliable, user-friendly, and time-saving. Many research works have been conducted based on different methods of hand gesture recognition. These gestures are used in the robot to enable a more natural way of controlling the robot. Gesture recognition

also provides a good visual tool for interacting with the robotic system. It mainly involves image processing and machine learning for system or software development.

This robot has several applications. One is shown here, a pick-and-place robot. This pick-and-place arm is controlled by a 1*4 keyboard switch. The robot is equipped with an RF camera to record videos at 2.4 GHz. The live video of robot movement is received at the base station using a computer system. It is processed in MATLAB software, and the video is simultaneously processed in DSP code.

2- Relevant work

There are many techniques used to control the robot through gestures. We use the K-curvature method

3- Existing systems

In many existing systems, movements are used to control the robot. Some motion detection systems include adaptive color segmentation, acquisition and labeling by blocking, morphological filtering, and then gesture actions by pattern matching and skeletonization. It does not provide dynamics for gesture inputs due to pattern matching. Some systems use an interface device to provide real-time movements to the robot.

4- The proposed system

In this system, the user controls the robot from a control station, which can be a laptop, a computer with a good-quality webcam, or an external webcam. This webcam uses hand movements to generate commands for the robot to record the real video stream. Movement commands are given using the spinal hand. Mainly, five types of movements are used, which will be explained further. The robot moves in all possible fields in the environment using five different types of commands that are connected to forward, backward, right, left, and stop. An image frame is taken as input, and the image is processed. Then the processed image is used to extract the gesture command. This gesture command can be one of the above five instructions. A signal is generated from this command and transmitted to the robot using ZigBee (the transmitter).

This generated signal is stored in the file at the control station. As soon as the ZigBee (receiver) in the robot receives a command from the control station, it is transmitted to the PIC microcontroller. The microcontroller takes this signal as input from ZigBee and produces some output signals, which are transmitted to the drive motor. This production of the output signal depends on the input of the gesture; a different output signal is generated for all five possible inputs of the gesture. The drive motor uses the robot to start the DC motor. When the command signal is given to the robot, it executes the previous command and moves until the next command is given or there is any obstacle in its path. Fig. 1 shows the main flow of the system.

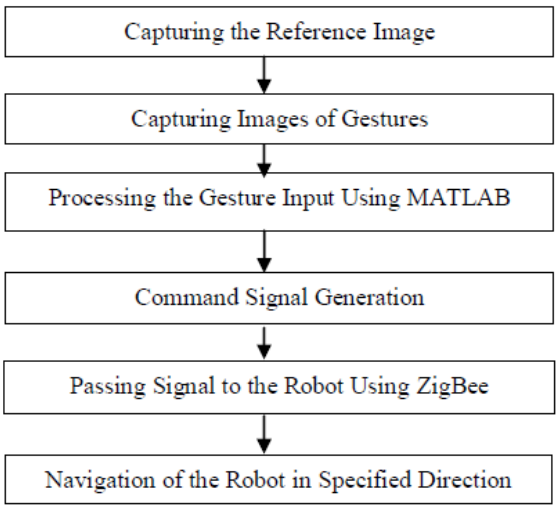

Fig. 1 Basic flowchart of the system

5- Technologies used

The MATLAB image processing toolbox is used. But fortunately, most operations do not require a toolbox. The required commands are shown in the toolbox of the image. Some of the commands used here are read and write, which help MATLAB read and write images. The user can run the MATLAB code to detect the movements in order to move the robot in the desired direction.

6- How to recognize shapes using a neural network

Fault detection can be done with different methods, but in this article, a neural network has been used because of its advantages, which are mentioned below.

**Classification*

Training a system so that it can learn a number of inputs with the help of an observer, and after training, an input sample that has not been taught to the system before is given to the system as input so that it can use that sample in the system according to the training it has seen. Put it in the appropriate category.

** Clustering*

In this sample of systems, the goal is to design a system that can place input samples in a number of categories without an observer. There is no supervisor in this system, which means that our training phase is without a supervisor.

** Regression*

In this type of system, the goal is to calculate the estimate of a problem based on the available data, that is, by having the available data; we can estimate the problem for the next set of data.

The main goal in this article is to detect shapes, and for this purpose, we must prepare a dataset consisting of shapes in different sizes, colors, and rotation angles. [8]. The whole process of diagnosing problems can be done in five steps:

1- Image preprocessing
2- Extracting the features of shapes
3- Construction of a suitable neural network with extracted features
4- Training of the relevant neural network
5- Neural network test with new samples

6-1- Image pre-processing

In this phase, we prepare the primary images that are completely mixed and have irrelevant and useless information for neural network learning and feature extraction. Shapes cannot be taught to the neural network before preprocessing, so in order for them to become a suitable input, they must go through a series of processes called preprocessing. This pre-processing includes various items, as follows:

- Noise elimination
- Separating the image from the background
- Convert image to binary image
- Standardization of the dimensions of all images
- and...

The above steps are performed by a function called getdata(). By using this loop, the image reading is done as follows:

```
for i2=1:20
   if i2==6 || i2==11 || i2==16
      j=1;
   end
      if i2<6
i=imread(['trainPic\a',num2str(j),'.jpg']);
       elseif i2<11
i=imread(['trainPic\b',num2str(j),'.jpg']);
```

```
    elseif i2<16
i=imread(['trainPic\c',num2str(j),'.jpg']);
```

Then, the read images are converted from color mode to black and white mode using the following command, so the black and white operation of the input images is a standardization action in the pre-processing stage.

```
a= rgb2gray(i);
```

Then, using the remove-back function, we separate the image; or the useful area from the background image, that is, the non-useful areas are removed, which is also a pre-processing operation as described below.

```
function [ x ] = removeBack( x )
x(:,1:5)=[];
x(:,end-5:end)=[];
x(1:5,:)=[];
x(end-5:end,:)=[];
while x(1,:)==0
   x(1,:)=[];
end
while x(:,1)==0
   x(:,1)=[];
end
while x(end,:)==0
   x(end,:)=[];
end
while x(:,end)==0
   x(:,end)=[];
end
x(:,1)=[];
x(:,end)=[];
x(1,:)=[];
x(end,:)=[]
```

6-2- Features extraction

Various parameters can be used to extract features. For example, to distinguish a square from a triangle with a simple feature of the number of sides or by using the number of angles of the shape, this can be done.

The features that we chose in this article are as follows:

- Number of angles
- The output of the edge detection algorithm
- Number of sides
- Image volume relative to the entire figure
- The output of the hough transform algorithm
- and ...

In the next step, edge detection is done using the "canny" function, which is one of the features of shapes in the neural network. After the edge shape is detected, it is possible that the edge-detected shape has noise and is not standard as a result, so we use morphological algorithms to make it more standard [7]. Edge detection in the Canny method consists of six steps:

The first step is to filter the original image and remove noise from the image. For this purpose, the Gaussian filter can be used with a simple mask, which is exclusively used in the Canny algorithm.

The second step is to find strong edges using the gradient magnitude at each point, for which a Sobel mask is usually used.

$|G|=|Gx|+|Gy|$

The third step is to obtain the direction of the edges of the image using the gradient value in the x and y directions, which was calculated in the previous step. The following formula is used to calculate the direction of the edges.

theta = invtan (Gy / Gx)

The fourth step is assigning acceptable directions in the image to the obtained directions. We allow only four directions for each pixel in the image: 0, 45, 90, and 135 degrees. Therefore, we map the

obtained directions to one of these four directions.

The fifth stage is non-maximum suppression. This step checks the direction of the edges and removes the edges that cannot be considered edges. This step gives a thin line in the final image.

The sixth step is thresholding in edge detection. In Canny, a method called hysteresis is used. For this purpose, we define two upper and lower thresholds. Any pixel that has a gradient greater than the upper limit is accepted as an edge, and if it has a value less than the lower limit, it is rejected, and if it has a value between these two limits, it is accepted if one of its neighbors is accepted.

Compared to other algorithms, the Canny algorithm is highly accurate, but it has computational complexity and is a bit slow.

Algorithms for filling image holes are among the morphological operators on images. Morphology operators on images include a wide set of algorithms that process images based on their shape. In morphological operation, the result of the operation of an image is exactly the same size as the original image. In a morphological operator, the value of each pixel in the output image is determined by comparing that pixel in the input image with its neighbors [6].

The stretch operator adds pixels to the boundaries of an object in the image, and the shrink operator removes pixels from the image. The number of pixels that are added to or removed from the image depends on the size and shape of the neighborhood structure that we define. A neighborhood structure could be, for example, a 5x5 square.

In this step, we extract the characteristics of the shapes using labeling. In this step, operations such as determining the number of holes in a shape (hole operator), obtaining the center of the shape (centroid), showing the number of pixels in the shape (area), and specifying the number of sides in neural networks (boundaries) are all done using the "region props" function.

In the next step, using the resize function, we reduce the dimensions of the input images for easy training of the neural network, which actually matches the size of different shapes because the images with large dimensions are If the number of neurons is too high, they will not have the ability to train the neural network, and this act itself is a kind of standardization.

6-3- Construction and training of neural networks

An artificial neural network is a practical method for learning various functions such as functions, with real values, functions with discrete values, and functions with vector values. Neural network learning is immune to training data errors, and such networks have been successfully applied to problems such as speech recognition, image recognition and interpretation, and robot learning. A neural network is a method of calculation based on the interconnected connection of several processing units is made. The network consists of an arbitrary number of cells, nodes, or neurons that connect the input set to the output.

The neural network supports parallelism, and in the human brain, the nerves work as an electrochemical process because if the neurons of the brain were to work in series, how slow would the human reaction be for any event?. Fortunately, this is not the case in the brain, and brain neurons have become increasingly parallel, and this

property is also present in artificial neural networks, which increases the computational speed of these networks. It is believed that the human brain consists of 10- 11 neurons; each neuron is connected to 104 other neurons. It is insignificant. Nevertheless, a person is able to their image 0.1 seconds. This extraordinary power must be obtained from the parallel processing distributed among a large number of neurons. [5]

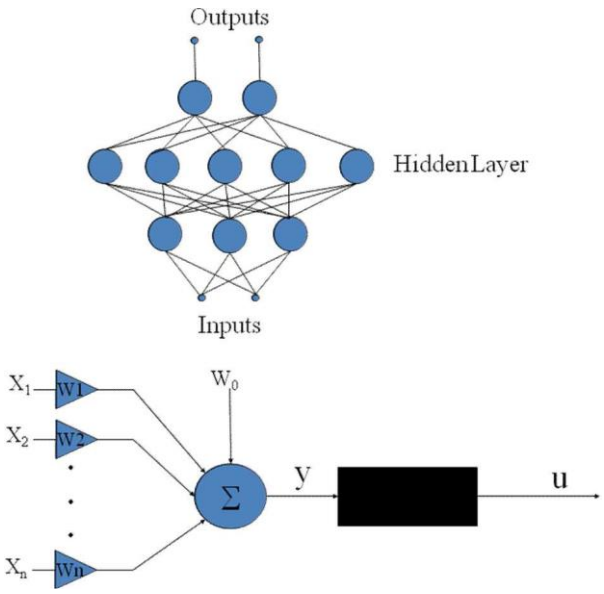

Fig. 2 Multilayer perceptron

To train our system, we need a perceptron neural network because it performs classification easily. It is a system with a number of inputs and outputs and a number of hidden layers.

6-3-1- Perceptron

A type of neural network is built based on a computing unit called a perceptron. A perceptron takes a vector of inputs with real values and calculates a linear combination of these inputs. If the resulting value is greater than a threshold value, the output of the perceptron will be equal to 1, otherwise it will be equal to -1.

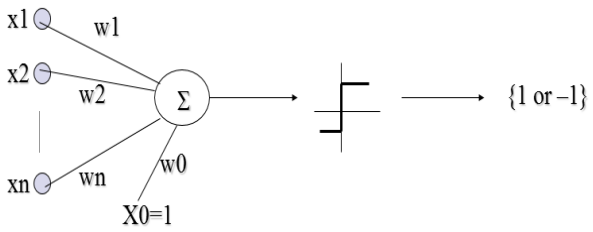

Fig. 3 Perceptron schematic

6-3-2- Learning a perceptron

The perceptron output is defined by the following relationship:

$O(x1, x2, \ldots, xn) = 1$
if $w0 + w1x1 + w2x2 + \ldots + wnxn > 0$
otherwise
$O(x1, x2, \ldots, xn) = -1$

Neural network inputs depend on the number of selected features. Since the inputs of each neuron are either one or zero, we have to convert the features into binary. In this system, we have 2500 input neurons, and the number of our outputs is equal to the number of shapes that the system is supposed to learn (currently 4), and we have 10 hidden layers. So we need a weight matrix; the weight matrix is equal to 4*2500. During training, the weight matrix changes, and finally we reach a fixed value. [4]

6-3-3- Perceptron ability

Preceptron can be considered hyperplane decision surface in the n-dimensional space of samples. The perceptron produces a value of 1 for the samples on one side of the screen and -1 for the values on the other side**.**

Decision boundary (WX = 0)

6-3-4- Add bias

Adding bias makes the perceptron network easier to use.

So that we don't need to use another rule to learn the bias, we consider the bias as an

input with a constant value of 1 and assign the weight W0 to it.

$$\hat{y} = b + \sum_i x_i w_i$$

$$\hat{y} = w_0 + \sum_{i=1} x_i w_i$$

6-3-5- Perceptron training

How do i learn the weights of a single perceptron in such a way that the perceptron creates the correct values for training examples?

There are two different ways to do this, which are as follows:

- Perceptron law
- Delta law

6-3-6- Perceptron learning algorithm

We assign random values to the weights. We apply the perceptron to each educational example. If the example is evaluated wrong, we correct the values of the perceptron weights and follow the following process:

Are all educational examples evaluated correctly?

- Yes , end of the algorithm.
- No, we will go back to step 2.

Perceptron law

For an educational example, X = (x1, x2, ..., xn), in each step the weights change according to the Percepton law as follows:

wi = wi + Δwi

where:

Δwi = η (t - o) xi

t = target output

o = output generated by the perceptron

η = constant called the learning rate (e.g., 0.1)

It has been proven that for a set of linearly separable examples, this method converges, and the perceptron will be able to correctly separate the examples. [1]

C is a widely used programming language for embedded processors and controls. Embedded C uses the most standard C syntax and semantics, for example, main (), variable definition, data type declaration, conditional statements (if, switch, case), loops (while for), functions, arrays, strings, etc. Here, we used two embedded C codes to control the drive of two DC motors, which are used to control the movement of the robot, and we used another code to control the drive of the two DC motors of the pick and place robot arm.

MPLABX TOOLl

MPLAB is a free integrated development environment for developing embedded programs on PIC and Ds PIC microcontrollers and is developed by Microchip Technology. This tool is used to execute embedded C code.

Design

Our design is mainly divided into two parts. The first part focuses on recognizing the hand gesture command. Commands are generated at the control station and sent to the robot via ZigBee within the ZigBee range. The robot moves in the specified direction according to the specified command. The second part focuses on the control of the pick and place arm and the use of the 1 * 4 keyboard switch. The following section shows the steps taken in the design.[3]

A. Record the reference image

The reference image is a simple background image that needs to be captured using a webcam before the user performs gestures. This image is required because it contains some background

constraints to correctly identify the palm with minimal noise in the image, so the reference image should be a plain background.

B. Motion recording

The end result of the hand gesture recognition system is to generate a command and these commands are used to control the robot. There are mainly five possible gesture instructions that can be given to the robot, namely forward, backward, right, left and stop. These gesture commands are given by the user based on the number of fingers. Hand gesture commands are given against a reference image using a webcam.

C. Hand Gesture Recognition

Once the reference image is captured, the user can run MATLAB code for motion detection to move the robot in the desired direction. The image frame is taken as input from the webcam at the control station, and further processing is performed on each input frame for palm recognition. These recognized gestures are further transmitted to the receiver serially via a USB-to-UART converter.

D. ZigBee

It is simpler and cheaper than other wireless personal area networks (WPANs), such as Bluetooth or Wi-Fi. This device is used to send a signal from the transmitter to the receiver, which is located at a distance of 100-200 meters through a wireless antenna. ZigBee has a defined rate of 250 kbps. Here it is used to send the command signal from the control station to the robot. This command signal is then sent to the PIC microcontroller. Fig. 4 shows ZigBee.

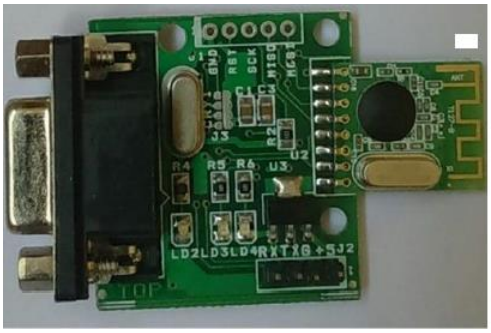

Fig. 4 ZigBee

E. Microcontroller: PIC 16CXX

All PIC microcontrollers use an advanced RISC architecture. The PIC16F8X devices have enhanced core features, an eight-level deep stack, and multiple sources of internal and external interrupts. A total of 35 instructions (minus the instruction set) are available. In addition, a large set of patents is used to achieve a very high level of performance. PIC16F8X microcontrollers typically achieve a 2:1 code compression and up to a 4:1 speed improvement (at 20 MHz) over other 8-bit microcontrollers in their class. The PIC16F8X has 68 bytes of RAM, 64 bytes of data EEPROM, and 13 I/O pins. A timer or counter is also available. Fig. 5 shows PIC microcontroller.

Fig. 5 PIC microcontroller

F. Arduino

Arduino is designed based on the ATMega328 microcontroller. The ATmega328 is an 8-bit CMOS microcontroller with a low-power RISC

architecture. By executing powerful instructions in a single clock cycle, the ATmega328 delivers close to 1 MIPS (million instructions per second) per MHz allowing the system to be designed to optimize power consumption at processing speeds. The microcontroller is programmed using the Arduino programming language (based on wiring) and the Arduino development environment (based on processing). Fig. 6 shows the Arduino.

Fig .6 Arduino board

G. RF433 MHz receiver and transmitter

It is often necessary to switch electrical devices remotely without a direct line between the transmitter and receiver. A remote control system based on an RF transmitter and RF receiver can be used to control the output load from a remote location. An RF transmitter, for example, uses radio frequency to transmit signals at a specific frequency and bandwidth. An RF receiver can receive these signals only if it is tuned to a predetermined signal or data pattern. Fig. 7 shows the sign of a receiver. [2]

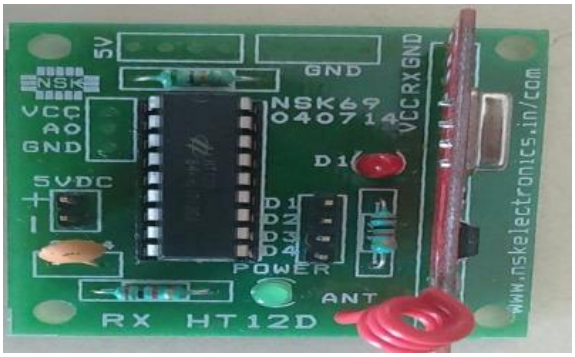

Fig .7 RF433 MHz receiver

H. Motor driver: circuit L293D

Here we use two L293D circuits. One of them is used to start the robot and the other is used to move the pick and place arm. In the first circuit, all input pins are connected to the PIC 16CXX digital pins, and the four outputs are connected to the DC robot motor. In the second circuit, all the four input pins are connected to the Arduino digital pins, and the four output pins are connected to the DC motor of the pick and place arm. In both circuits, pins are used to enable input and output pins and Vcc is used to supply external power to DC motors. Fig. 8 shows the L293D pin diagram.

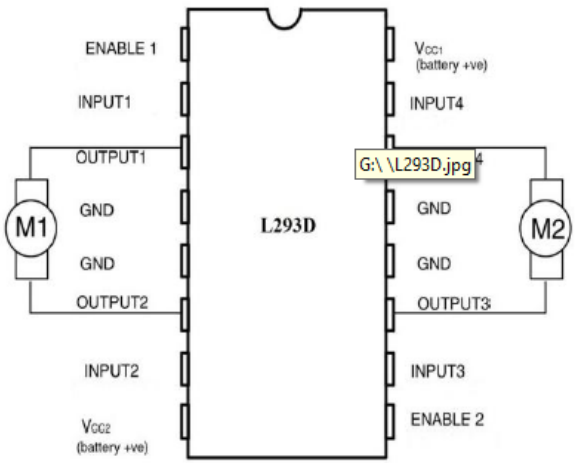

Fig. 8 L293D Pin Diagram

I. DC motors

A DC motor is mechanically coupled directly to the electric motor. The stator is stationary and therefore current flows

through it. The current in the rotor is changed by the switch. DC motors are more suitable for equipment that uses 12V DC systems in cars than conveyor motors, although they require precise speed control to control speeds above and below rated speeds. The speed of a DC motor can be controlled by changing the current. The Fig. 9 shows the DC motor.

Fig .9 DC motor

6-4- Implementing and testing the neural network

A. Transferring the movement of gestures

The image frame is taken as input from the webcam at the control station and further processing is performed on each input frame for palm detection. It includes background constraints for palm detection with minimal noise in the image.

B. Palm detection

After capturing the image frame from the webcam, some basic operations are performed on this frame to prepare it for further command recognition processing. The palm recognition process is done.

1) Feature extraction

An image frame is captured as input via webcam. The RGB input frame is converted to a black and white image. After obtaining a black and white image, we use the Median filter to remove noise from the input image and then convert the image into a binary sequence. Then we delete the unwanted part of the image. We need to remove unnecessary pixels (0) from the original image. In this image frame, palm detection is done at a binary threshold.

C. Command recognition using a special method:

After finishing the pre-processing of an input frame, further processing is done on the extracted image according to the specified technique. The method of commanding movements is as follows:

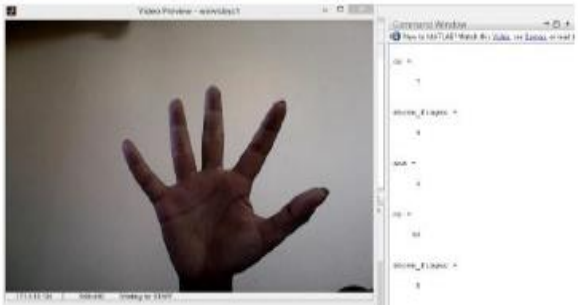

Fig. 10 Posing method

The robot in all possible fields in the environment uses five different types of commands, which are: forward, backward, right, left and stop for the number of fingers 1, 2, 3, 4 and 5 as shown in the Fig. 10 above. It is working.

D. Specific signal generation:

After identifying the gesture command, a certain value of the command signal, unique for each gesture command, is produced. This signal value is written in the file using MATLAB functions.

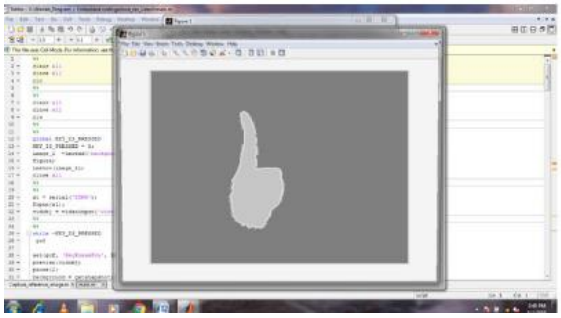

Fig .11 Gesture tracking

ZigBee-E

As soon as the command is generated in the control station, a file with the word tagged is written with it. This file is read by ZigBee after regular review. As a wireless communication, ZigBee communicates with the control station, which is also in the same network. Both the control station and ZigBee are provided with the same IP address. ZigBee has access to information from the control station.

F. Microcontroller

As shown in the Fig. 12, PICF816XX and Arduino ZigBee are connected to the PIC pin via expansion pins. When the communication process begins, ZigBee tries to connect itself with the control station using the provided IP address of the control station. As soon as ZigBee is connected to the control station, it continuously sends the command signal to the receiver. These received signal values are then passed to the PIC, which derives the command and calls specified for that command. The PIC sends five digital signals as input to the L293D drive motor (1). We also use Arduino to control the selection and location using push buttons. A command signal from Arduino is given as input to the L293D drive motor (2).

G. Motor driver: L293D

We use two L293D drive circuits. L293D (1) takes the digital signal as input from the PIC and gives the digital output to the robot's DC motors, which help the robot move in the environment. The power source of the circuit is its PIC board. One L293D drive circuit can only control two DC motors at the same time. The L293D circuit (2) is used to drive two DC motors from the Pick & Place arm. This L293D (2) takes input from the Arduino.

H. DC motors in the robot

This is the final product of the robot, which consists of all the ZigBee, PIC and L293D hardware and the starter motor driver. As shown in Fig. 13, the robot selects the location in the robot chassis, which has a power supply provided by rechargeable batteries. Slow Two DC motors are connected to this robot chassis. It is controlled by the movements of the user at the control station. The pick and place

robot is controlled by a 1x4 keyboard circuit.

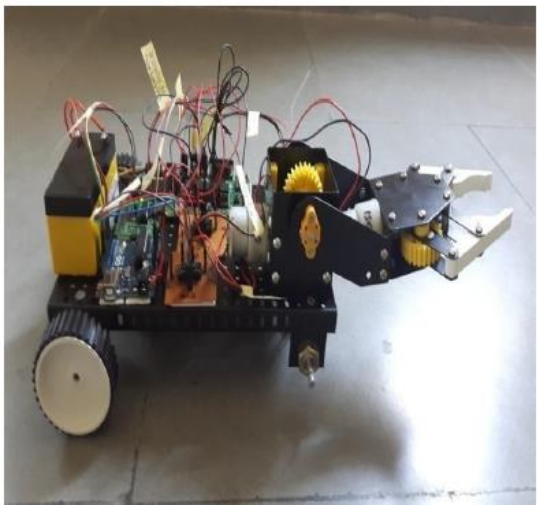

Fig. 12 Robot 13

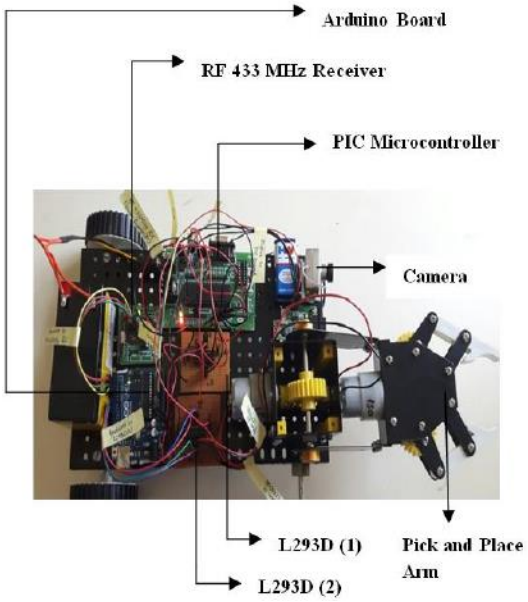

Fig .13 Final connection of the robot

7- Conclusion

The purpose of this article is to investigate robot control in a new way. Gesture control, which is a more natural way of controlling the device, makes controlling robots more efficient and easier. Finger counting techniques are used to input movements, where the number of each finger specifies a specific command for the robot to move in a specific direction in the environment. After recognizing the gesture, the command signal is generated and transferred to the robot and it moves in a specific direction. The pick and place arm is used to select objects where human intervention is not possible.

References

[1] Ahmed, T. (2012). A neural network based real time hand gesture recognition system. *International journal of computer applications*, *59*(4).

[2] Dhawan, A., & Honrao, V. (2013). Implementation of hand detection based techniques for human computer interaction. *arXiv preprint arXiv:1312.7560*.

[3] Kaura, H. K., Honrao, V., Patil, S., & Shetty, P. (2013). Gesture controlled robot using image processing. *International Journal of Advanced Research in Artificial Intelligence*, *2*(5).

[4] Jain, M., Aditi, A. L., Khan, M. F., & Maurya, A. (2012). Wireless gesture control robot: an analysis. *International Journal of Advanced Research in Computer and Communication Engineering*, *1*(10), 855-857.

[5] Swetha, N. (2013). Design and implementation of accelerometer based robot motion and speed control with obstacle detection. *Int. J. Sci. Eng. Technol*, *2*, 749-755.

[6] MaruthiSagar, N. V., Kumar, D. S., & Geethanjali, N. (2014). MEMS based gesture controlled robot using wireless communication. *International Journal of Engineering Trends and Technology*, *14*(4), 185-188.

[7] Khan, R. Z., & Ibraheem, N. A. (2012). Hand gesture recognition: a literature review. *International journal of artificial Intelligence & Applications*, *3*(4), 161.

[8] Malima, Ozgur, & Cetin. (2006). A fast algorithm for vision-based hand gesture recognition for robot control. *2006 IEEE 14th Signal Processing and Communications Applications*, 1-4.