"SYSTEM AUTOMATION USING SPEECH & GESTURE RECOGNITION"

A REPORT

Submitted by

ASHUTOSHANAND SINHA

100430111042

In fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

SHANTILAL SHAH GOVERNMENT ENGINEERING COLLEGE

BHAVNAGAR, 360340

Gujarat Technological University, Ahmadabad

October, 2017

SHANTILAL SHAH GOVERNMENT ENGINEERING COLLEGE

Certificate

This is to certify that the project report titled **"SYSTEM AUTOMATION USING SPEECH & GESTURE RECOGNITION"** submitted by**ASHUTOSHANAND SINHA**(Enrollment No: 100430111042) has satisfactory completed his work of subject **PROJECT 1** During the academic year 2016-2017 authentic work carried out by them under my supervision and guidance.

Guided by:

ANKIT PONKIA

Lecturer

Shantilal Shah Engineering College

Head Of Department

S.B Parmar

SHANTILAL SHAH GOVERNMENT ENGINEERING COLLEGE

ABSTRACT

System Automation is one of the biggest area in industrial applications. The use of System Automation is to handle a system or a machine very easily and effectively. GESTURE & SPEECH recognition is an easiest & interesting way for Automation and controlling system. Gestures recognition deals with the goal of interpreting human gestures via mathematical algorithm. Gestures made by users with the help of a color band and/or body pose , in two or three dimensions, get translated by software/image processing into predefined commands .The computer then acts according to the command. There have been a lot work already developed in this field either by extracting hand gesture only or extracting hand with the help of color segmentation. In this project, both gesture extraction and Speech Extraction used for better, faster, robust, accurate and real-time applications. For hand gesture detection, the default background is captured and stored for further processing. Comparing the new captured image with background image and doing necessary extraction and filtering, gesture portion can be extracted. For Speech recognition , the sound recorded is analyzed and filtered and comparing against threshold value. Then applying different mathematical algorithms different gestures and Sound samples detected. All this work done using MATLAB software. Laptop automation is one of the bigest example of the gesture and speech recognition.By interfacing a portion of Master hand or/and color to mouse of a Computer, the computer can be controlled same as the mouse. And then many virtual (Augmented reality)or PC based application can be developed (e.g. sudoku game, Paint).In current time robotics is a very big criterian for resurch and industrial application devlopment. so devlopement of many more functioning in roborics criterian using gesture and speech recognition very easy.(e.g. A four wheel robot can understand gesture and speech inputs data and perform a functions. like ; move in fron side , move in beck side ,move in left side , move in right side , ect.)

Acknowledgement

A project on this magnitude would have been difficult to complete without constant support and encouragement from our respected teachers and successful completion of our project. Our special thanks to guide **Prof.ANKITPONKIA** for valuable guidance and continued interest show in our project. We also thanks to all teachers of department and other class members to have applied us and who guided us in our project as and when required.

Guidance is seeded from various teachers involved in the project. Equal complete participation is done by all members of the team.

INDEX

LIST OF FIGURES

1.INTRODUCTION

(1.1) System Automation is one of the biggest area in industrial applications. The use of the System Automation is to handle a system or a machine very easily and effectively. In the present scenario , so many ways are available for handling a system or we can say so many ways are available for system automation .like

1) computer interface controlling.
2) cell phone interface controlling.

Innovation of our project idea" SYSTEM AUTOMATION USING GESTURE AND SPEECH RECOGNIZATION ". So We are started our exploration to finding an easiest way for Automate the system .

Finally we getting , "GESTURE " & "SPEECH" recognition is an easiest way to Automate and control system ,because Gesture and Speech is a non-verbal type communication and peoples are use this type of communication easily.

Education is not require for this type of communication.So we decided to make an laptop which can operate using speech and gesture recognition .so everyone operate laptop without learning about laptop

.

Although once basic module will be design then we can add many smart function and Automate different system to increase it's artificial intelligence.

(1.2) GESTURES

It is hard to clinch on a useful definition of gestures because of its wide variety of applications. A statement concerning it can only specify a particular domain of gestures. Many researchers had tried to define gestures but their actual meaning is still uncertain. Bobick and Wilson have defined gestures as the motion of the body intended tocommunicate with other agents. As per the context of the project, a gesture is defined as an expressive movement of hand which has a particular message, that is communicated precisely between a sender and a receiver.

A sender and a receiver should have the same set of information for a particular hand gesture for a successful communication. A gesture can be categorized as dynamic and static. A dynamic gesture is intended to change over a period of time whereas a static gesture is observed at the spurt of time. A waving hand meaning goodbye is an example of dynamic gesture and a still hand sign is an example of tatic gesture. All the static and dynamic gestures are interpreted over a period of time to understand a full message

This complicated process is termed as gesture recognition. Gesture recognition deals with a process of recognition and interpretation of a stream of continuous sequential gesture from the given set of input data.

(1.3) HUMAN COMPUTER INTERFACE SYSTEM

Computer is used by many people either at their work or home or in their spare-time. Special input and output devices have been designed over the years with the purpose of easing the communication gap between computers and humans. Keyboard and mouse are frequently used for this purpose.

Each new device is seen as an attempt to increase the intelligence-level of computer and making humans able to perform more complicated communication with the computer. This has been possible due to the result oriented efforts made by computer professionals for creating successful human computer interfaces . As the complexities of human needs have turned into many folds and continues to grow so, the need for Complex programming ability and intuitiveness are critical attributes of computer programmers to survive in a competitive environment.

The computer programmers have been immensely successful in easing the communication between computers and human. With the emergence of every new product in the market; it attempts to ease the complexity of jobs performed. For instance, it has helped in facilitating tele operating, robotic use, better human control over complex work systems like cars, planes and monitoring systems.

Earlier, Computer programmers were avoiding such kind of complex programs as the focus was more on speed than other modifiable features. However, a shift towards a user friendly environment has driven them to revisit the focus area .

The idea is to make computers understand human interactions and develop a user friendly human computer interfaces (HCI).Making a computer understand speech, facial expressions and human gestures are some steps towards it. Gestures are a type of non-verbally exchanged information. A person can perform innumerable gestures at a time.Since human gestures are perceived through vision, it is a subject of great interest for computer vision researchers.

The project aims to understand and utilize human hand gestures by creating an HCI. Coding of these hand gestures into machine language requires a complicated programming algorithm.

(1.4) Gesture Recognition:

Gesture recognition is a topic in computer scienceand language technology with the goal of interpreting human gestures via mathematical algorithms. Gestures can originate from any bodily motion or state but commonly originate from the face or hand. Current focuses in the field include emotion recognition from the face and hand gesture recognition. Many approaches have been made using cameras and computer visionalgorithms to interpret sign language. However, the identification and recognition of posture, gait,proxemics, and human behaviors is also the subject of gesture recognition techniques.

Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces or even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse.

Gesture recognition enables humans to communicate with the machine (HMI) and interact naturally without any mechanical devices. Using the concept of gesture recognition, it is possible to point a finger at the computer screen so that the cursor will move accordingly. This could potentially make conventional input devices such as mouse, keyboards and even touch-screensredundant.

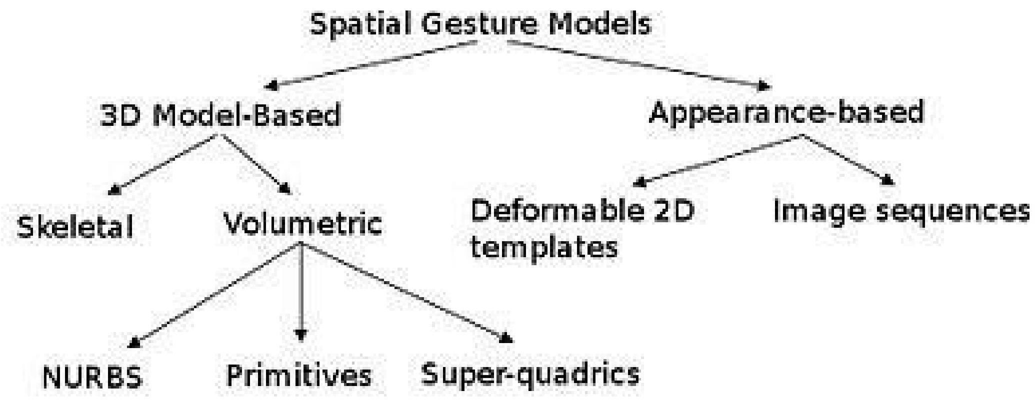


Fig.(1) <u>Various Gesture Models</u>

Gesture recognition can be conducted with techniques from computer vision and image processing.The literature includes ongoing work in the computer vision field on capturing gestures or more general human pose and movements by cameras connected to a computer

Depending on the type of the input data, the approach for interpreting a gesture could be done in different ways. However, most of the techniques rely on key pointers represented in a 3D coordinate system. Based on the relative motion of these, the gesture can be detected with a high accuracy, depending of the quality of the input and the algorithm's approach.

2. DETAILS OF VARIOUS COMPONENTS USED IN PROJECT

2.1) ARDUINO

An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components to facilitate programming and incorporation into other circuits. An important aspect of the Arduino is the standard way that connectors are exposed, allowing the CPU board to be connected to a variety of interchangeable add-on modules known as *shields*. Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I²C serial bus, allowing many shields to be stacked and used in parallel. Official Arduinos have used the mega AVR series of chips, specifically the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560.

Fig(2). Arduino

A handful of other processors have been used by Arduino compatibles. Most boards include a 5 volt linear regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variants), although some designs such as the LilyPad run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions. An Arduino's microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an externalprogrammer.

2.1.1)Arduino hardware

Like its predecessor, the Arduino open-source hardware platform is based on Atmel's versatile ATmega MCU family (Figure 2). ATmega's modified Harvard-style AVR processor core combines a rich instruction set with 32 general purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in a single instruction cycle. ATmega MCUs are available with either 4 K or 8 Kbytes of in-system programmable flash (with read-while-write capabilities) plus various combinations of EEPROM (256/ 512/ 512/ 1 Kbytes) and SRAM (512/ 1 K/1 K/ 2 Kbytes).

2.1.2) PIN DIAGRAM:.

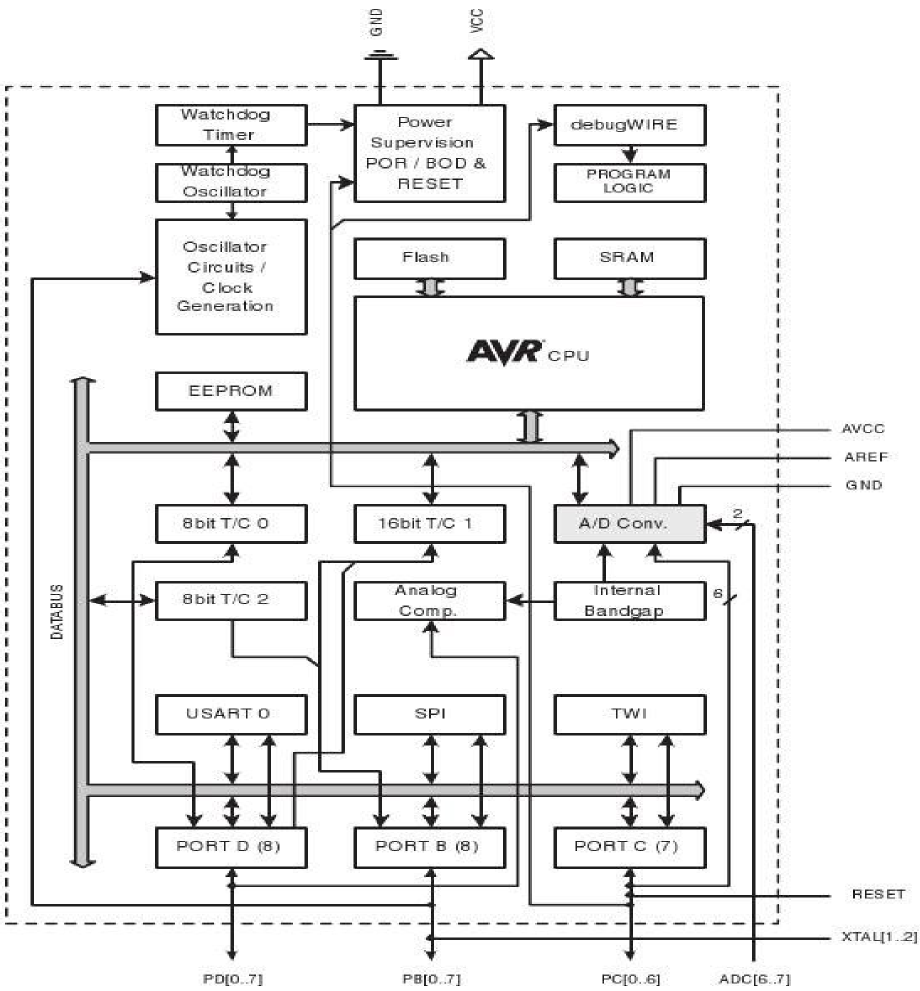


Fig.(3)Arduino Pin Diagram

2.1.3) AVR Architecture

First, lets refresh on the computer architecture that the Arduino's processor utilizes. The AVR ATmega line of MCU's use what is called a Harvard Architecture. This is where program code and program data are physically separated. Bottom line, as aArduino (or ATmega) programmer, you are always dealing with, at least, TWO memories. This is a bit different than programming on a PC or Mac where the programmer is dealing with Von Neumann Architecture (in reality, PCs/Macs use a hybrid but from a programmers perspective, we can think of them as Von Neumann). The Von Neumann Architecture combines program and data in one memory.For the purposes of this tutorial, it is very important to recognize that your Arduino has two memories: Program Memory and Data Memory. The picture below (source: Atmel AVR ATmega 328p Manual) shows the AVR Core Architecture. The green box represents the Flash Program Memory. This is where your code is loaded when you upload your program over USB. The light blue box is the Data SRAM. This is where your program data is kept when power is applied to your board.

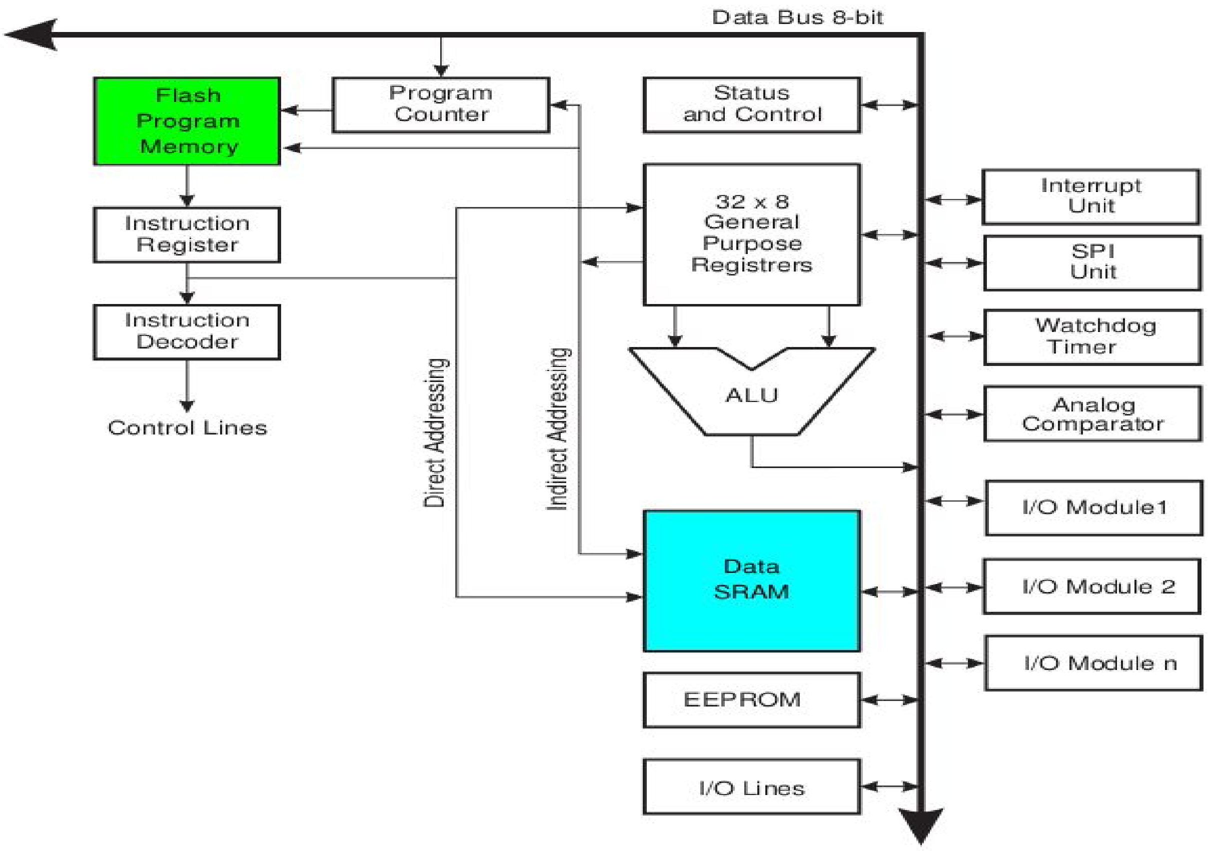


Fig.(4) <u>AVR Architecture</u>

The AVR CPU is complimented with a powerful set of hardware functions and I/O capabilities. Although some MCUs have a slightly different configuration, with most devices include:

- 23 general purpose I/O lines
- a 6- or 8-channel 10-bit ADC
- six PWM Channels

- a serial programmable USART
- a byte-oriented 2-wire Serial Interface
- an SPI serial port
- three flexible Timer/Counters (with compare modes)
- a real Time Counter with Separate Oscillator.

The Arduino Nano is a slightly more compact board based on the ATmega328 or ATmega168 that was designed specifically to be dropped onto breadboards (Figure 4a). For applications where space is at an even greater premium, the Arduino Mini (Figure 4b) is also available. Other variants feature breadboard areas for small circuits or on-board communication capabilities such as Bluetooth or Ethernet (Figure 4c). One of the most unique members of the Arduino family is the LilyPad, a board designed specifically for use in “smart apparel” and e-textiles (Figure 4d). The washable/wearable processor board can be sewn to fabric and connected to similarly-mounted power supplies, sensors and actuators using conductive thread.

At a conceptual level, when using the Arduino software stack, all boards are programmed over an RS-232 serial connection, but the way this is implemented varies by hardware version. Serial Arduino boards contain a level shifter circuit to convert between RS-232-level and TTL-level signals. Current Arduino boards are programmed via USB, implemented using USB-to-serial adapter chips such as the FTDIFT232. Some variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. (When used with traditional microcontroller tools instead of the Arduino IDE, standard AVR ISP programming is used.)

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins, six of which can producepulse width modulated signals, and six analog inputs. These pins are on the top of the board, via female 0.10-inch (2.5 mm) headers. Several plug-in application shields are also commercially available.

2.2) **Camera :**A portable camera interfaced to a device or an inbuilt camera in any device, capable of sending data, is used to capture the image of a hand gesture and transmit via networking devices , hardwired or wireless ,to main control computer, where the image is processed and necessary task is executed.

2.3) **Control Computer (laptop) :** A computer device that will receive the data and process using MATLAB and generate control signals to carry out necessary tasks as programmed.

2.4) **Microphone :** A portable microphone interfaced to a device or an inbuilt mic in any device, capable of sending data, is used to capture the voice of a user and transmit via networking devices , hardwired or wireless ,to main control computer, where the voice is processed and necessary task is executed.

A **microphone** (colloquially called a **mic** or **mike**; both pronounced /ˈmaɪk/)is an acoustic-to-electric transducer or sensor that converts sound into an electrical signal. Microphones are used in many applications such as telephones, tape recorders, karaoke systems, hearing aids, motion picture production, live and recorded audio engineering, FRS radios, megaphones, in radio and television broadcasting and in computers for recording voice,speech recognition, VoIP, and for non-acoustic purposes such as ultrasonic checking or knock sensors.

Most microphones today use electromagnetic induction(dynamic microphone), capacitance change (condenser microphone), piezoelectric generation, or light modulation to produce an electrical voltage signal from mechanical vibration.

3.SOFTWARE USED IN PROJECT

"SOFTWARE"

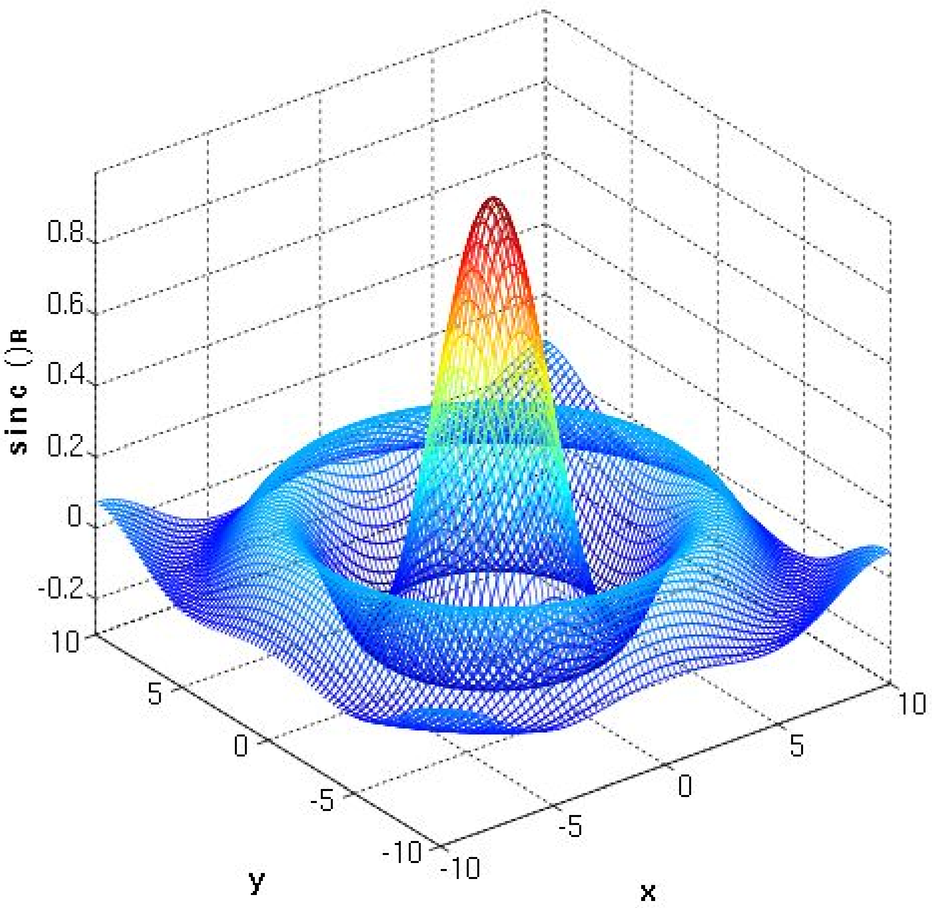
3.1) Arduino 1.0.3

The Arduino integrated development environment(IDE) is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring projects. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a "sketch". Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier.

3.2) Matlab

MATLAB (**mat**rix **lab**oratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the Mu PAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded system


0.8
0.6
0.4
0.2
0
-0.2
sinc ()R
10
5
0
-5
-10
y
-10
-5
0
5
10
x

4. LITERATURE REVIEW

4.1 Important Matlab Functions

IV.1.1 Create Video object :

Syntex:

vid=videoinput('winvideo',1,'YUY2_640x480');

Description :

Generate 640x480 Size Video Object . Thise Object Store Data Ahich Is Capture by Priority 1 “Winvideo” Camera .

IV.1.2 Get Snap Shot :

Syntex:

me=getsnapshot(vid);

Description :

Cpture Image From Live video Object Nd store Into ‘me ’ Object.

4.1.3 rgb2gray : Convert RGB image or colormap to grayscale.

Syntax :

I = rgb2gray(RGB)
newmap = rgb2gray(map)

Description :

I = rgb2gray(RGB) converts the truecolor image RGB to the grayscale intensity image I. rgb2gray converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.newmap = rgb2gray(map) returns a grayscale colormap equivalent to map. If the input is an RGB image, it can be of class uint8, uint16, single, or double. [11] The output image I is of the same class as the input image. If the input is a colormap, the input and output colormaps are both of class double.

4.1.4 Im2bw : Convert image to binary image, based on threshold

Syntax :

```
BW = im2bw(I, level)
BW = im2bw(X, map, level)
BW = im2bw(RGB, level)
```

Description :

BW = im2bw(I, level) converts the grayscale image I to a binary image. The output image BW replaces all pixels in the input image with luminance greater than level with the value 1 (white) and replaces all other pixels with the value 0 (black). Specify level in the range [0,1]. This range is relative to the signal levels possible for the image's class. Therefore, a level value of 0.5 is midway between black and white, regardless of class. To compute the level argument, you can use the function graythresh. If you do not specify level, im2bw uses the value 0.5.BW = im2bw(X, map, level) converts the indexed image X with colormap map to a binary image.BW = im2bw(RGB, level) converts the true color image RGB to a binary image.If the input image is not a grayscale image, im2bw converts the input image to grayscale, and then converts this grayscale image to binary by thresholding

4.1.5 medfilt2 *:* 2-D median filtering

Syntax :

```
B = medfilt2(A, [m n])
B = medfilt2(A)
```

Description :

Median filtering is a nonlinear operation often used in image processing to reduce "salt and pepper" noise. A median filter is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges.B = medfilt2(A, [m n]) performs median filtering of the matrix A in two dimensions. Each output pixel contains the median value in the m-by-n neighborhood around the corresponding pixel in the input image. medfilt2 pads the image with 0s on the edges, so the median values for the points within [m n]/2 of the edges might appear distorted.B = medfilt2(A) performs median filtering of the matrix A using the default 3-by-3 neighborhood.

4.1.6 **bwareaopen** *:* Morphologically open binary image (remove small objects)

Syntax :

BW2 = bwareaopen(BW, P)

Description :

BW2 = bwareaopen(BW, P) removes from a binary image all connected components (objects) that have fewer than P pixels, producing another binary image, BW2. The default connectivity is 8 for two dimensions, 26 for three dimensions, and conndef(ndims(BW), 'maximal') for higher dimensions. BW can be a logical or numeric array of any dimension,

4.1.7 **imfill** *:* Fill image regions and holes

Syntax :

BW2 = imfill(BW)

Description :

BW2 = imfill(BW) displays the binary image BW on the screen and lets you define the region to fill by selecting points interactively by using the mouse. To use this interactive syntax, BW must be a 2-D image. Press Backspace or Delete to remove the previously selected point. A shift-click, right-click, or double-click selects a final point and starts the fill operation. Pressing Return finishes the selection without adding a point. The input image can be numeric or logical, and it must be real and non-sparse. It can have any dimension. The output image has the same class as the input image.

4.1.8 **bwboundaries** *:* Trace region boundaries in binary image

Syntax :

B = bwboundaries(BW)

Description :

B = bwboundaries(BW) traces the exterior boundaries of objects, as well as boundaries of holes inside these objects, in the binary image BW. bwboundaries also descends into the outermost objects (parents) and traces their children (objects completely enclosed by the parents). BW must be a binary image where nonzero pixels belong to an object and 0 pixels

constitute the background. The following figure illustrates these components. bwboundaries returns B, a P-by-1 cell array, where P is the number of objects and holes. Each cell in the cell array contains a Q-by-2 matrix. [11] Each row in the matrix contains the row and column coordinates of a boundary pixel. Q is the number of boundary pixels for the corresponding region. BW can be logical or numeric and it must be real, two-dimensional, and nonsparse. L and N are double. A is sparse logical.

4.1.9 **regionprops** *:* Measure properties of image regions

Syntax :

STATS = regionprops(BW, properties)

Descriptions :

STATS = regionprops(BW, properties) measures a set of properties for each connected component (object) in the binary image, BW. The image BW is a logical array; it can have any dimension. [11] STATS is a structure array with length equal to the number of objects in BW, . The fields of the structure array denote different properties for each region, as specified by
properties.

Shape Measurement –

Area, EulerNumber, Orientation, BoundingBox, Extent, Perimeter, Centroid, Extrema, PixelIdxList, ConvexArea, FilledArea, PixelList, ConvexHull, FilledImage, Solidity, ConvexImage, Image, SubarrayIdx, Eccentricity, MajorAxisLength, EquivDiameter, MinorAxisLength

Pixel Value Measurement –

MaxIntensity, MinIntensity, WeightedCentroid, MeanIntensity, PixelValue

4.1.10 WaveRead : Read Wave File (Read Predefine Wave File)

Syntex

```
[xsignal4,Fs4,Nbits4]=wavread('s4.wav');
```

Description :

Read “S1.Wave” File from Memory And store in xsignal4 name variablr With Frequency 44100 And 16 Bit Formet.

4.1.11 FFT : Genarate Fast Foutrioer Transform Of Speech Signal

Syntex

```
Y4 = fft(x4,NFFT4);
```

Description :

Above Instructon Genrate Fast Fourier Form Of Signal x4 With NFFT4 Points.

4.1.12 MAX : For Finding Maximum Value in Matrix And It’s Position

Syntex

```
Y4 = fft(x4,NFFT4);
```

Description :

Above Instructon Genrate Fast Fourier Form Of Signal x4 With NFFT4 Points.

4.1.12 LIVE RECORDING OF SIGNAL:

Syntex

```
A=audiorecorder(44100,16,1);
recoordblocking(A,2);
x=getaudiodata(A);
```

Description :

For Live Speech Signal Recording We Use Audiorecorder. Here IN Audiorecorder 44100 is Fs , 16 is Nbits And 1 Is Device Id.

Recordblocking is work When Speech signal is detected by Mick. So Here We show that We have 2 Second Recording time

Getaudiodata is uased for converting Audio File Data to Double Type Data .

5.WORKING OF PROJECT

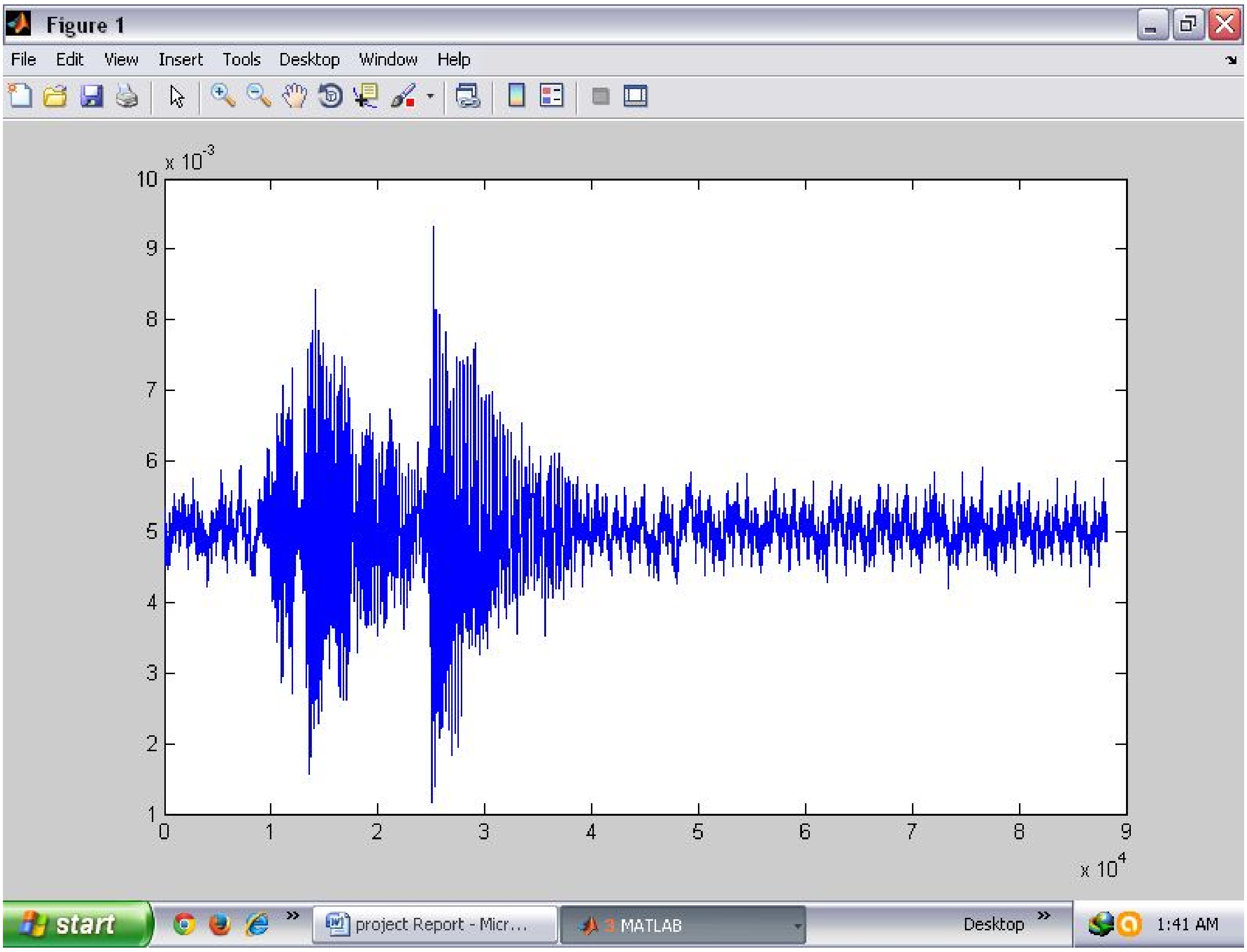
5.1) ***Shutdown & Hibernate Operations:***

5.1.1) SHUTDOWN (Time Domine Operation)

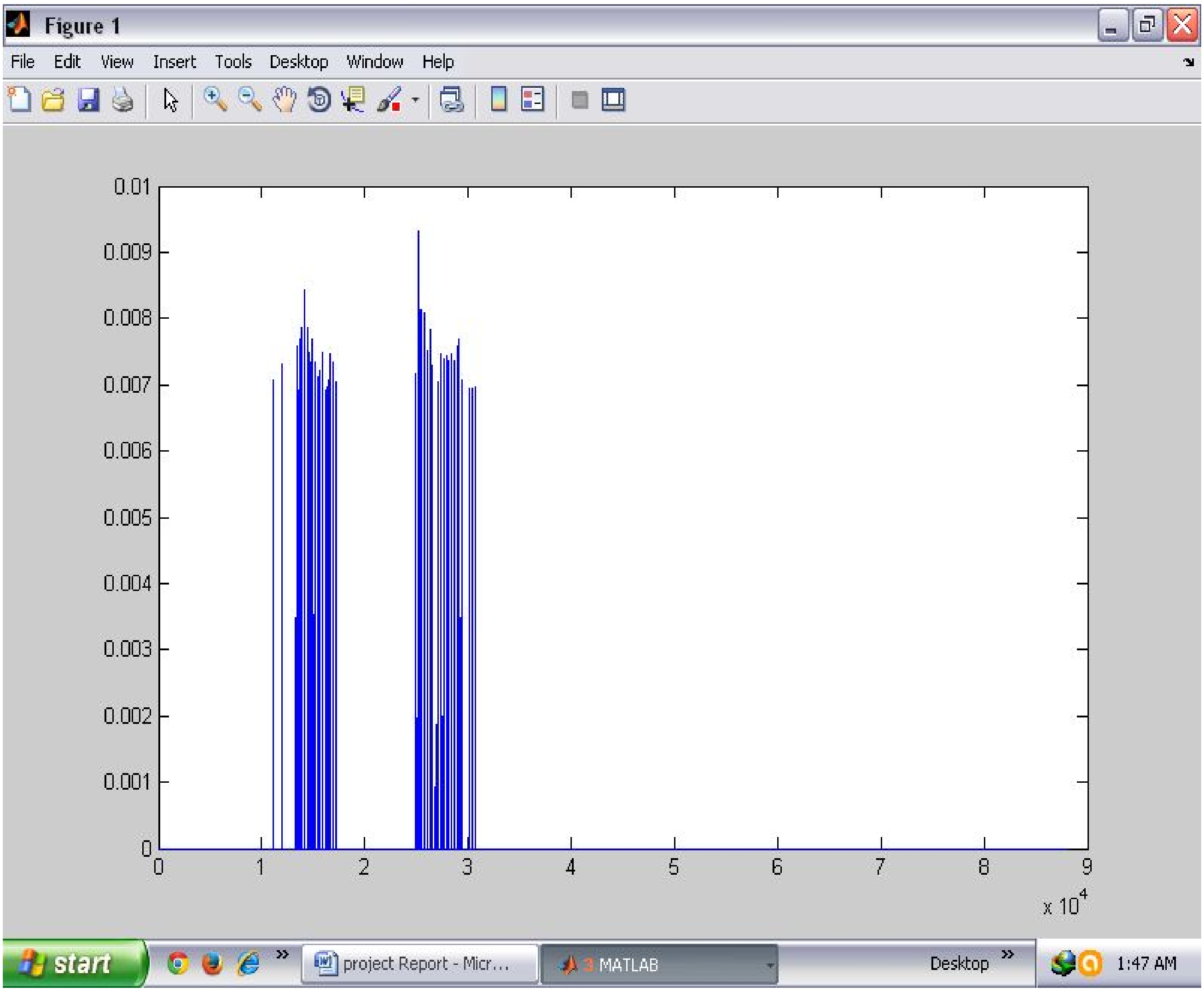
To recognize the" SHUTDOWN" speech , a audio object is created in Matlab and voice is recorded using instructions as follows:

A=audiorecorder(44100,16,1);

Recordblocking(A)



Fig(5) voice recorded with Noise (Time Domine)

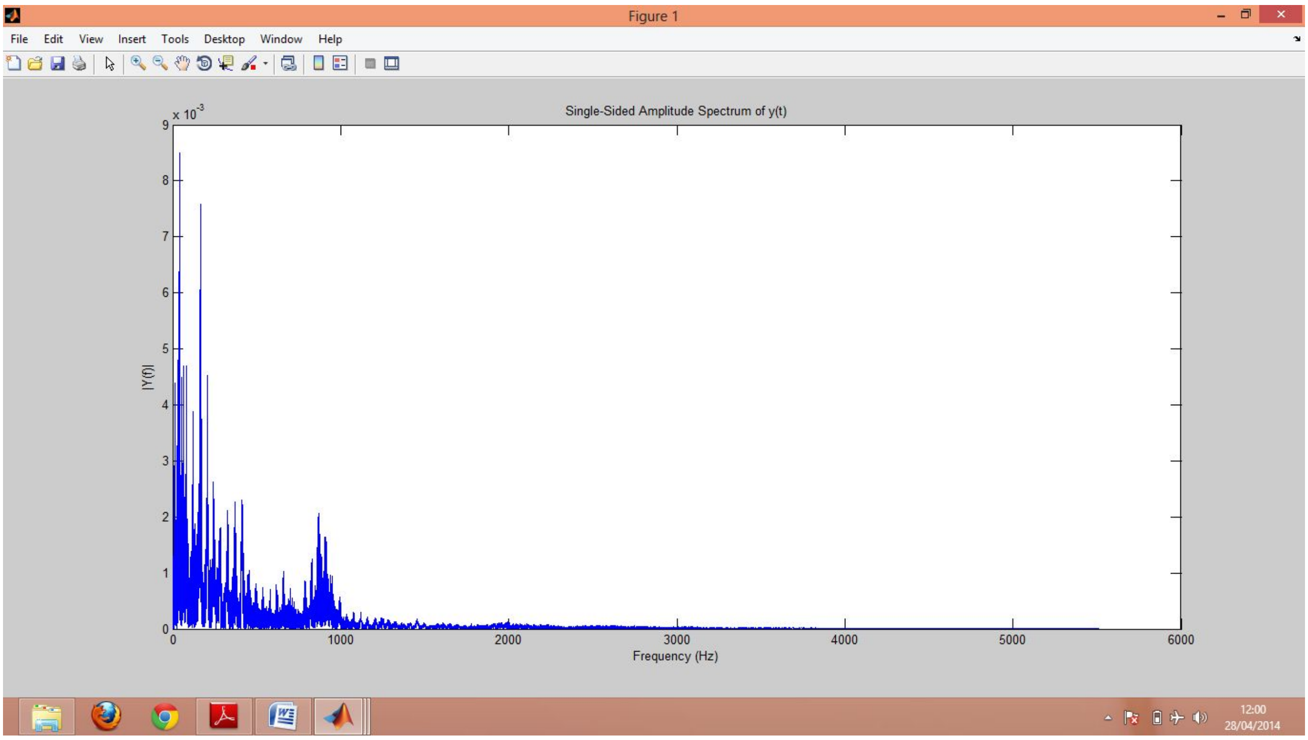


Fig(6) Voice After Noise Removal (Time Domine)

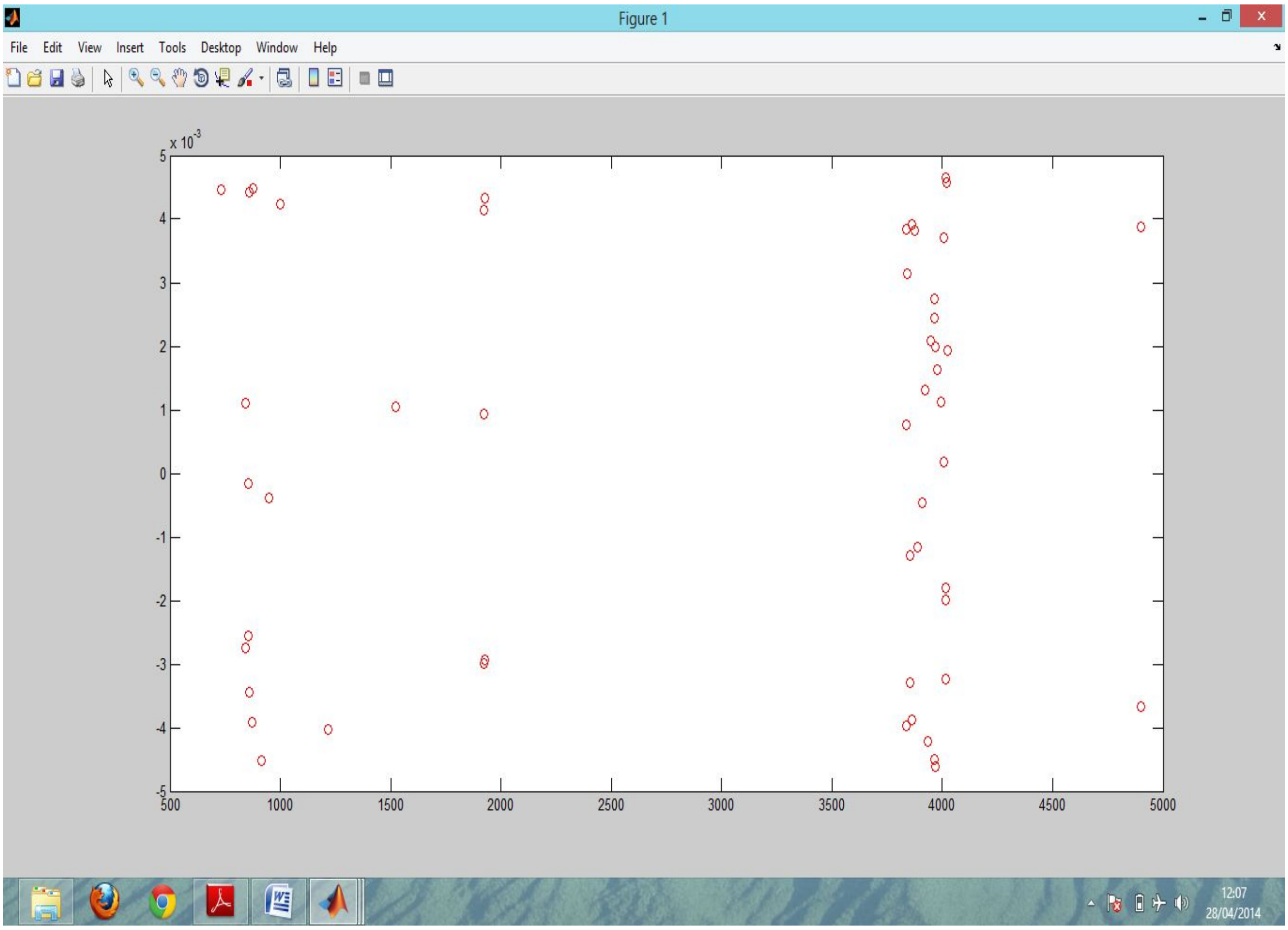
5.1.2) SHUTDOWN (Frequncy Domine Operation)

For Finding FFT (Fast Fourier Transform) Of Speech Signal

```
[xsignal1,Fs1,Nbits1]=wavread('s1.wav');
        x1=xsignal1;
L1=length(x1);
T1 = 1/Fs1;
t1 = (0:L1-1)*T1;
NFFT1 = 2^nextpow2(L1);
Y1 = fft(x1,NFFT1)/L1;
f1 = Fs1/8*linspace(0,1,NFFT1/2+1);
YY1 = 2*abs(Y1(1:NFFT1/2+1));
```



<u>Fig(7) FFT Form Of Voice Signal with Noise</u>



Fig(8) **<u>50 Maximum Points Of FFT Form Of Voice Signal</u>**

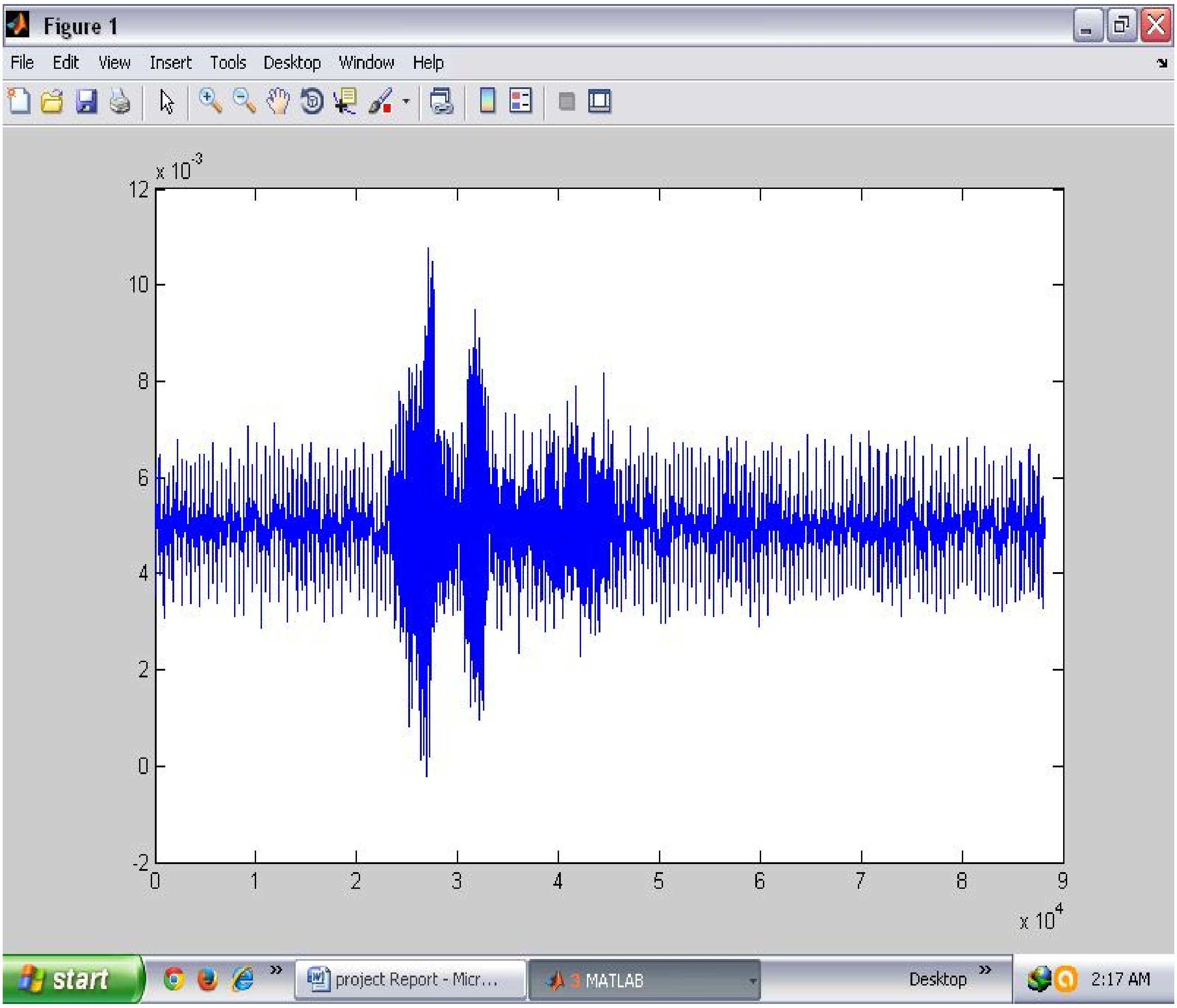
5.1.3) HIBERNATE

To recognize the" HIBERNATE" speech , a audio object is created in Matlab and voice is recorded using instructions as follows:

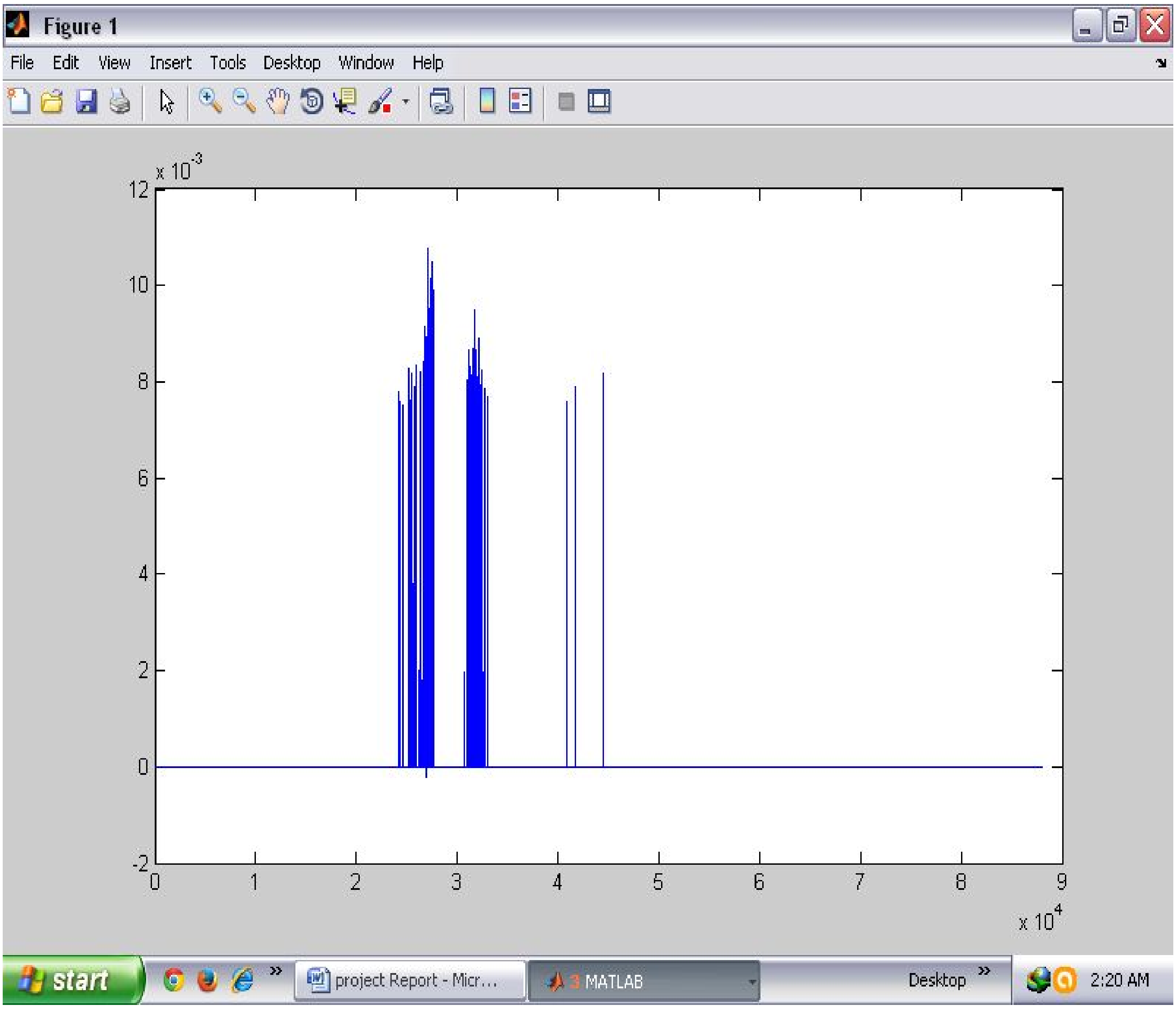
A=audiorecorder(16,44100,1)

;

Recordblocking(A)



Fig(9) voice recorded with Noise

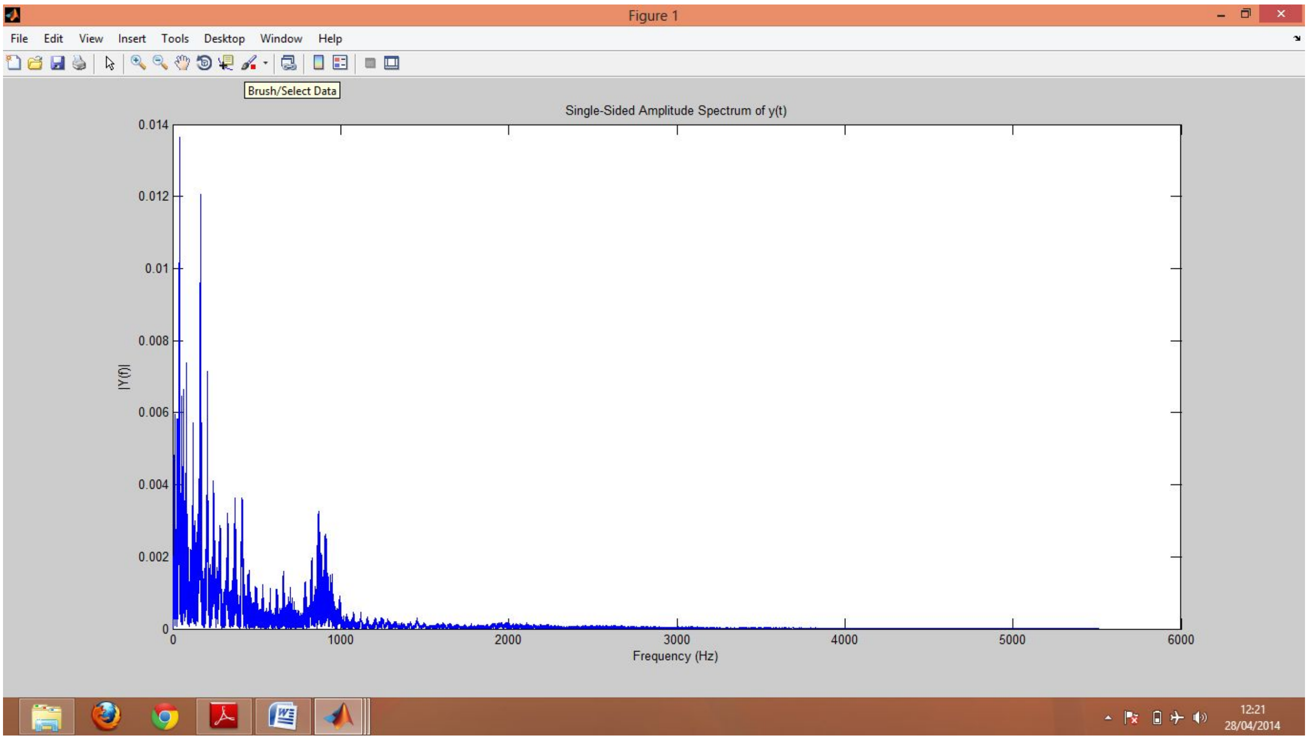


Fig(10) Voice After Noise Removal

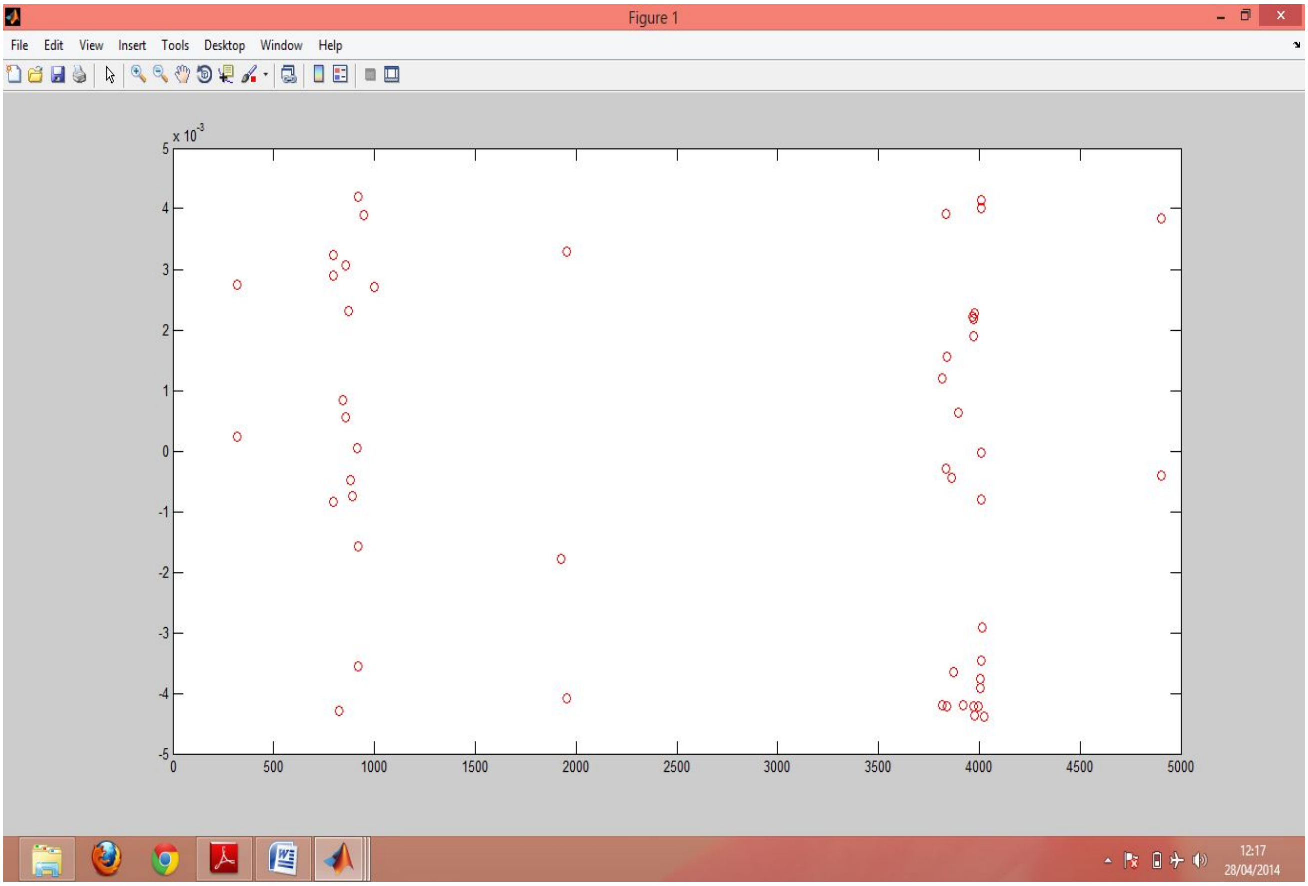
5.1.4) HIBERNATE (Frequncy Domine)

For Finding FFT (Fast Fourier Transform) Of Speech Signal

```
[xsignal1,Fs1,Nbits1]=wavread('s1.wav');

x1=xsignal1;

L1=length(x1);

T1 = 1/Fs1;

t1 = (0:L1-1)*T1;

NFFT1 = 2^nextpow2(L1);

Y1 = fft(x1,NFFT1)/L1;

f1 = Fs1/8*linspace(0,1,NFFT1/2+1);

YY1 = 2*abs(Y1(1:NFFT1/2+1));
```



Fig(11) FFT Form Of Voice Signal with Noise



Fig(12) 50 Maximum Points Of FFT Form Of Voice Signal

5.2) Java Function :

JMOUSEEMU(POS,CLICK)

JMOUSEEMU(POS) moves mouse cursor to POS = [X,Y] (in pixels) on the current figure. If there is no figure open, POS specifies the position with respect to the bottom-left corner of the screen.

JMOUSEEMU(POS,CLICK) defines the button click option. CLICK is a string argument with five possible values [{'none'} | 'normal' | 'extend' | 'alternate' | 'open'].

Selection Type Property

'none' - No mouse click (default)

'normal' - Click left mouse button

'extend' - Shift-click left mouse button

'alternate' - Control-click left mouse button

'open' - Double click any mouse button

If only mouse click without cursor movement is desired, leave POS empty.

In addition to the 5 CLICK options for the single-command mode, 2 additional click options are available to enable mouse dragging:

'drag_on' - Click left mouse button and hold

'drag_off' - Release held left mouse button

Every 'drag_on' must be paired with subsequent 'drag_off', and no other mouse click commands is allowed during dragging.

IMPORT JAVA FUNCTION LIBRARY

There are Two Matlab Instruction for import Java Librar

import java.awt.Robot;

*import java.awt.event.**

mouse = Robot;

Uses of java library

- Dubole Click , Single Click
- Contoling Mouse Using External Device
- Opening Any Function Or Library Using Maatlab

- Read OR Write any Files on Computer. etc

5.3) Sudoku:

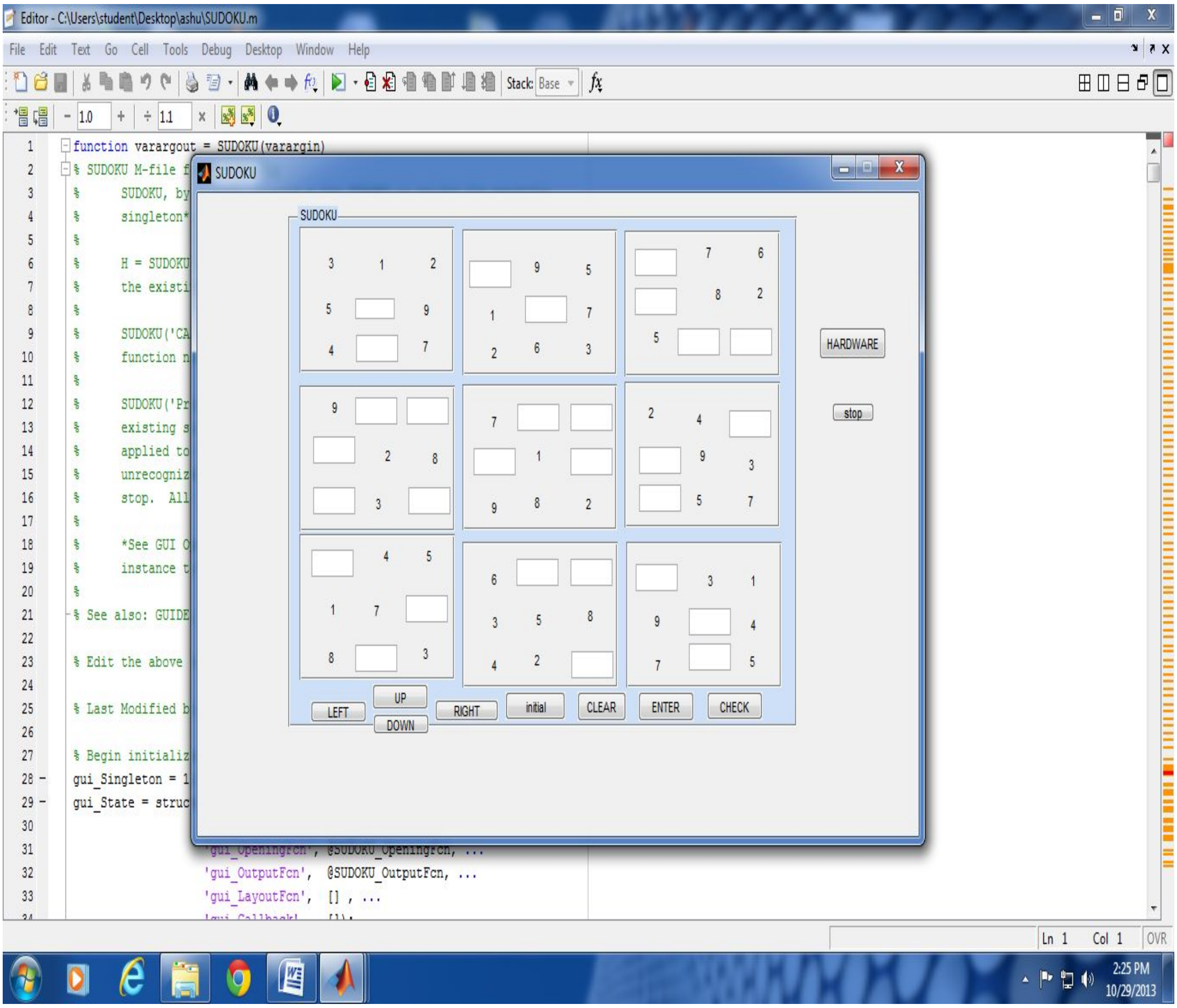


Fig. (13) Sudoku GUI

6s.PROGRAMMING IN MATLAB

5.1) Shutdown:

```
% s1=audiorecorder(44100,16,1);
% recordblocking(s1,2)   %recording samples
% shut=getaudiodata(s1);    %getting audio data into samqle_1
% plot(shut)
% shut = wavread('s2.wav');
% Noise Removal qractically
i=1;
shutnoise_mask = (shut(i:end)<0.0065)& (shut(i:end)>0.002);
shut_rem_noise=~(shutnoise_mask).*shut;
plot(shut_rem_noise)

% maximum Recognition
maxvalshut=max(shut_rem_noise);%max val
minvalshut=min(shut_rem_noise);%min val
a=shut_rem_noise;
i=1;
j=1;
for i=1:length(a)
if a(i)>0
break;
end
```

```
end
m1_start=i;
a1 = a(i:end,1);

a2=a1(end:-1:1);
% plot(a2);
for i=1:length(a2)
if a2(i)>0
break;
end
end
%m2_end=i;
a3 = a2(i:end,1);

a4=a3(end:-1:1);
figure
plot(a4)

m1=max(a4(1:6000));

n1=max(a4(6000:12000));

o1=max(a4(12000:end));

closeall

if m1 > 0.008
if n1 < 0.008
if 01 > 0.008
jmouseemu([20,20],'normal');
jmouseemu([320,60],'none');
```

```
end
end
end
```

5.2) Hibernate:

```
s1=audiorecorder(44100,16,1);
%  while(1)
recordblocking(s1,2)   %recording samples
hibernate=getaudiodata(s1);    %getting audio data into samqle_1
% plot(hibernate)
% hibernate = wavread('s2.wav');
% Noise Removal qractically
i=1;
hibernatenoise_mask = (hibernate(i:end)<0.0073)& (hibernate(i:end)>0.002);
hibernate_rem_noise=~(hibernatenoise_mask).*hibernate;
plot(hibernate_rem_noise)

% maximum Recognition
maxvalhibernate=max(hibernate_rem_noise);%max val
minvalhibernate=min(hibernate_rem_noise);%min val
a=hibernate_rem_noise;
i=1;
```

```
j=1;
for i=1:length(a)
if a(i)>0
break;
end
end
m1_start=i;
a1 = a(i:end,1);

a2=a1(end:-1:1);
% plot(a2);
for i=1:length(a2)
if a2(i)>0
break;
end
end
%m2_end=i;
a3 = a2(i:end,1);

a4=a3(end:-1:1);
figure
plot(a4)

m1=max(a4(1:5000));

n1=max(a4(5000:6000));

o1=max(a4(8000:10000));

g1=max(a4(12000:end));
```

```
closeall

if m1 > 0.0065
if n1 == 0
if 01 > 0.0065
if g1 > 0.0065

jmouseemu([20,20],'normal');
jmouseemu([350,60],'normal');
jmouseemu([320,90],'extend');


end
end
end
end
```

6.Applications

6.1) GESTURE BASED APPLICATIONS

Using the hand gestures and hand movement, many applications can be developed. It can also be interfaced with computer and can work as a mouse. Similarly a no. of mouse interfacing applications can be developed. Many virtual applications can be developed that creates a virtual reality. This works only when the application runs. All the computer applications can be interfaced with hand gesture control and can be controlled directly from hand gesture.

Mouse Interfacing :Its an example of dynamic gestures, where hand movement(essentially finger in this case) is tracked. Mouse pointer is interfaced to this movement and mouse clicks are initiated through static gestures.

Virtual Calculator: Virtual Calculator is like an augmented reality. It is designed in Matlab in the same interfacing program. Depending on the position of finger- peak of Master hand, application gives input to the calculator. It checks for 5 frames. If for 5 frames, the finger-peak remain in a particular region or number region, then that number or symbol is given as input.

PC Calculator: PC Calculator is the original calculator application given in windows platform by Microsoft. In PC Calculator, finger-peak of Master hand is interfaced with mouse cursor. If the mouse cursor remains within +-20 pixels for 5 frames, then Left Click event activated by java Robot class and if the cursor remains within +-20 pixels for 8 frames, then Right Click event activated by java Robot class. By this method, input is given to the calculator.

PC Paint : Above paint is done with the help of hand and without using mouse. In above application, two color pins used (Red, Blue). Red is used for left click and cursor movement and blue color is used for dragging. If the cursor remains within +-20 pixels for 5 continuous frames, then Left click gets triggered

Tele presence : There may raise the need of manual operations in some cases such as system failure or emergency hostile conditions or inaccessible remote areas. Often it is impossible for human operators to be physically present near the machines. Telepresence is that area of technical intelligence which aims to provide physical operation support that maps the operator to carry out the necessary task.

Remote Controlling of Hardware: Using dropbox, google drive; a captured image or video can be sent to any location. Processing the image, the given command or gesture can be found out. Then interfacing the computer to any hardware e.g motor through Atmega or any other medium, any hardware interfacing application can be executed.

6.2) SPEECH BASED APPLICATIONS

Aerospace (e.g. space exploration, spacecraft, etc.) .

Automatic translation.

Automotive speech recognition (e.g., On Star, Ford Sync).

Court reporting(Real time Speech Writing).

Home automation.

\

7.CONCLUSION& FUTURE WORKS

7.1) CONCLUSION:

- The proposed technique is simple , efficient and faster (SEF). .
- Using different colours and with different hand gestures, different types of interfacing can be done.
- In case of hand-mouse interfacing, the computer can be controlled from a distant point depending on the resolution of camera. If the camera is linked to the computer either by direct connection or by online; then system can be controlled remotely
- Using the cloud feature the hand gesture command can be sent to the main computer from a faraway area.
- In Hardware interfacing applications , many application can be developed in a similar way like emergency window closing application. This can have a vast utilisation in future smart homes

7.2) Future work :-

After completion of this project we want to

- Develop a computer for a deaf and blind people. In this compute we develop a system which is really easy for operating using gesture recognition and speech recognition, main advantage of this idea is , many job areas are open for deaf and blind people using this kind of technology. Because they also control the machine using speech and gesture recognition
- Develop a smart controlling house applications for an own house. To develop a house application more user friendly.

8.REFRENCES

HTTP://EN.WIKIPEDIA.ORG/WIKI/MICROPHONE

WWW.MATHWORKS.IN/PRODUCTS/**MATLAB**/

ARDUINO.CC/EN/GUIDE/INTRODUCTION

HTTP://EN.WIKIPEDIA.ORG/WIKI/ARDUINO