APPENDIX 1

SYLLABLE RECOGNITION
J COMPONENT PROJECT REPORT

by

Priyal Godha (18BIS0052)
Sai Sandeep (18BIS0039)
D Madhumita (18BIS0030)

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics
Vellore Institute of Technology Vellore
September 2019

APPENDIX 2

Declaration by Authors

This is to declare that this report has been written by us. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be plagiarized, we shall take full responsibility for it.

➔ Priyal Godha
(18BIS0052)
➔ Sai Sandeep
(18BIS0039)
➔ D Madhumitha
(18BIS0030)

Place-VIT,Vellore
Date-22 October 2019

APPENDIX 3

Acknowledgement

It is a matter of great pleasure for us to express our gratitude to people without whom we would not have been able to carry out our project successfully.

We would like to thank our faculty, ***Prof. Sibasankar Padhy****, for his continued support in all our endeavors.*

We thank our friends who helped us in this project through their guidance and encouragement.

Our thanks and appreciations also goes to the team members in developing the project and people who have willingly helped out with their abilities.

Abstract

Speech recognition is the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine-readable Format. This is a fast growing engineering technology. It has a number of benefits in different areas and provides potential benefits.This project mainly deals with the process of speech recognition.
To start, the audio signal is extracted from a source and
stored into the database. Then after sampling the signal, we used a Butterworth band pass filter to remove the noise.The signal generated by the voice of the speaker is compared with all the template signals already inbuilt in the program. To do the comparison we have used Fast Fourier transform technique (FFT) and converted the signals into frequency domain after which we found the lag between the template and the sample signal. Using the lag, we aligned the two signals and performed auto-correlation of sample with sample and cross correlation of template and sample. We plot the graphs of these two results and analyze the spectrum with which we can decide the legitimacy of the input speech signal. Using this approach we were able to identify basic syllables with considerable accuracy.

Table Of Content

Declaration ----------
1
Acknowledgment ----------
2
Abstract ----------
3
Introduction
Objective
Theoretical Approach
Experimental Procedure
Observation and results
Conclusion
Understanding our problems
And solutions
Individual contributions
Final Word
References
Appendix B

INTRODUCTION:

Speech recognition, the ability of devices to respond to spoken commands. Speech recognition enables hands-free control of various devices and equipment (a particular boon to many disabled persons), providing input to automatic translation, and creates print-ready dictation. Among the earliest applications for speech recognition were automated telephone systems and medical dictation software. It is frequently used for dictation, for querying databases, and for giving commands to computer-based systems, especially in professions that rely on specialized vocabulary.

Before any machine can interpret speech, a microphone must translate the vibrations of a person's voice into a wavelike electrical signal. This signal in turn is converted by the system's hardware—for instance, a computer's sound card—into a digital signal. It is the digital signal that a speech recognition program analyzes in order to recognize separate phonemes, the basic building blocks of speech. The phonemes are then recombined into words. However, many words sound alike, and, in order to select the appropriate word, the program must rely on the context. Both acoustic modeling and language modeling are important parts of modern statistically-based speech recognition algorithms. Hidden Markov models (HMMs) are widely used in many systems. Language modeling is also used in many other natural language processing applications such as document classification or statistical machine translation. Other models may be based on neural networking, Dynamic time warping or End to End speech recognition methods Speech recognition finds several applications in military, home automation , industrial automation, aeronautics and space research and is extensively burying deep into other fields as well .

OBJECTIVE

In this project we aim to develop a MATLAB code with which we can decide the legitimacy of the input speech signal.

Using this approach, we should be able to identify basic syllables with considerable accuracy,

Thus, the program determines what the user was probably saying and either outputs it as text or issues a computer command.

THEORETICAL APPROACH

Speech recognition functions using algorithms, with the help of acoustic and language modeling.

Acoustic Modeling is the representation of the interconnection between linguistic units of speech and audio signals, whereas, Language Modeling matches sounds with preloaded word sequences to help differentiate between words and syllables that sound similar.

To convert speech to on-screen text or a computer command, a computer has to go through several complex steps. When we speak, we create vibrations in the air. The analog-to-digital converter (ADC) translates this analog wave into digital data which the computer can comprehend.

To do this, it digitizes the sound by taking exact measurements of the wave at frequent intervals.

The system then filters the digitized sound to remove undesirable noise, and sometimes to segregate it into different bands of frequency (frequency is the wavelength of the sound waves, heard by humans as differences in pitch). It also normalizes the sound to a constant volume level.

It may also have to be temporarily aligned.

The speed of speech varies from person to person, so the sound must be accommodated to match the speed of the template sound samples already stored in the system's memory.

Next, the signal is divided into small segments as short as a few hundredths of a second, or even thousandths in the case of plosive consonant sounds.

The program then compares these segments to the known phonemes in the appropriate language.

There are 44 phonemes in the English language, while its number varies over different languages.

In the next step, The program analyses phonemes in the context of the other phonemes around them.

It runs the contextual phoneme plot through a complex statistical model and correlates them to a large library of known words, phrases, and sentences.

The program then resolves what the user was probably saying and either presents the output as text or issues a computer command

METHODOLOGY

For matching the syllables spoken by the user, here we are majorly using Graphical Analysis method. Here is how our program will recognize the speech of the user step-by-step:

We have divided the project into stages to make it more approachable

Stage 1:Database creation and calling

Stage 2: Sampling

Stage 3: Noise reduction

Stage 4: Signal alignment

Stage 5: FFT

Stage 6: Correlations and graphical analysis

Stage 7: Use Python to Convert Speech to Text

- **Take input audio and Sample:**

At first, we take the voice signal and sample it into 8000 small samples.

- **Create and call Database:**

We created a database of 23 audio files in which we spoke some words in different scenarios and wrote a matlab code through which we are able to recognise the input file.

- **Noise Reduction:**

Noise causes discrepancies and abnormalities to the input data so after sampling we have to reduce the noise to increase the efficiency of the final output. We use a Band-Pass Butterworth filter of order 7 to allow the signal components and to reject the noise.

- **Signal Alignment:**

For signal alignment, we first find the peaks of both the template and the sample signal. The lag or delay between the two signals is found by matching both the peaks. Now we shift the signal by the calculated value (delay value). Now both the signals are aligned.

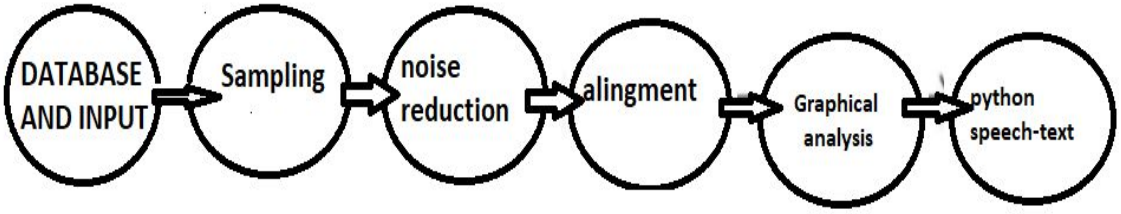
- **Fast Fourier Transform:**

Fast Fourier Transform is done because we then have a periodic function which is far easier and efficient to work with than the signal in the time domain. The auto correlation and cross correlation in the frequency spectrum leads to a faster and a more accurate result and hence we perform the Transform. So first we transform the template signal by using the inbuilt MATLAB function ‘fft’.and in similar way transform the sample signal and then plot a graph of both the functions.

- **Graphical Analysis through auto correlation and cross correlation:**

After plotting the auto correlation and the cross correlation, we analyse the output waveforms. Auto correlation is performed on the template signal and this is kept as a reference. Then, cross correlation is performed on the sample signal with the template. We know that cross correlation is same as auto correlation when the two signals given as input for the cross correlation is same, then effectively we will be performing correlation between the same signals. So, when the cross correlated output of the template and the sample becomes almost equal to the auto correlated output of the template, we can infer that both the template and the sample are the same and hence we can match the syllable.

BLOCK DIAGRAM

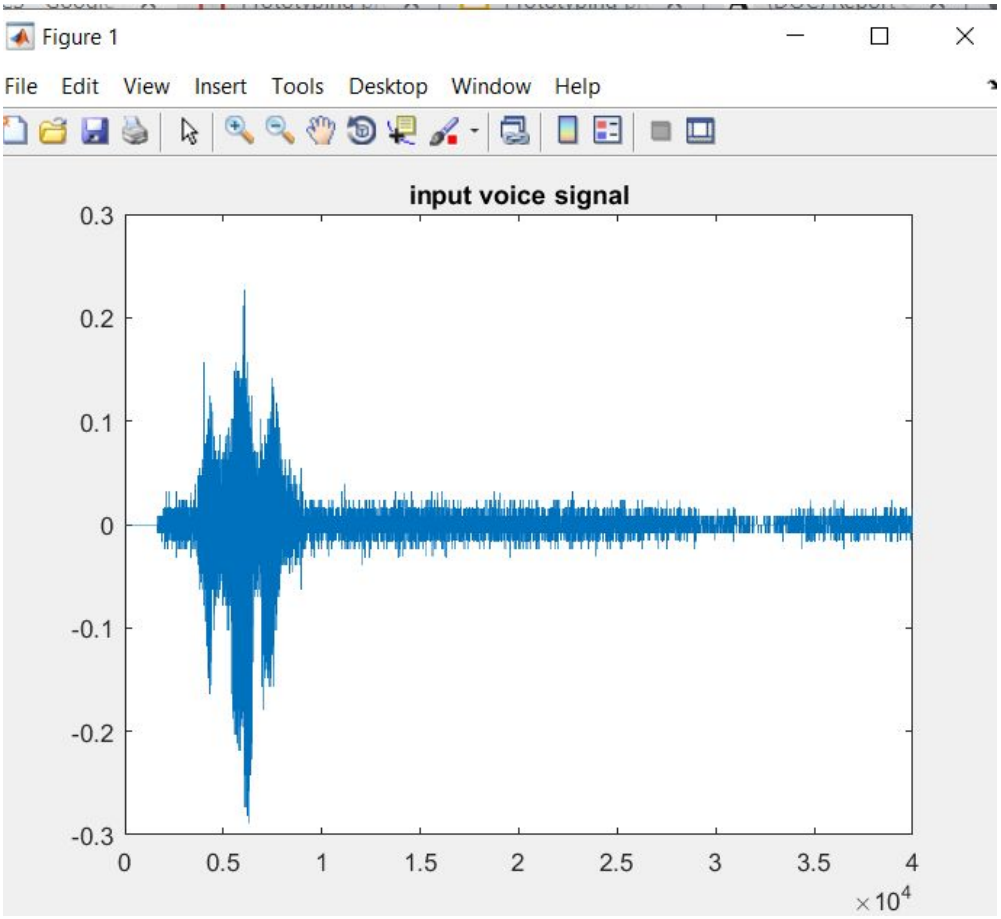


WORKING:

We have used matlab software where we have developed a code that calls the database of recorded audio files and runs that in for loop.When user speaks a word the code runs and functions as follows:
The signal is sampled and then filtered. This output is then aligned with all the template signals present in the database and is correlated with all .
Finally we get the output graphically. Python code converts Speech to text.

The code for both the programs is given in APPENDIX B and the Zipfile of database and the actual files of codes are provided .

OBSERVATION AND RESULTS



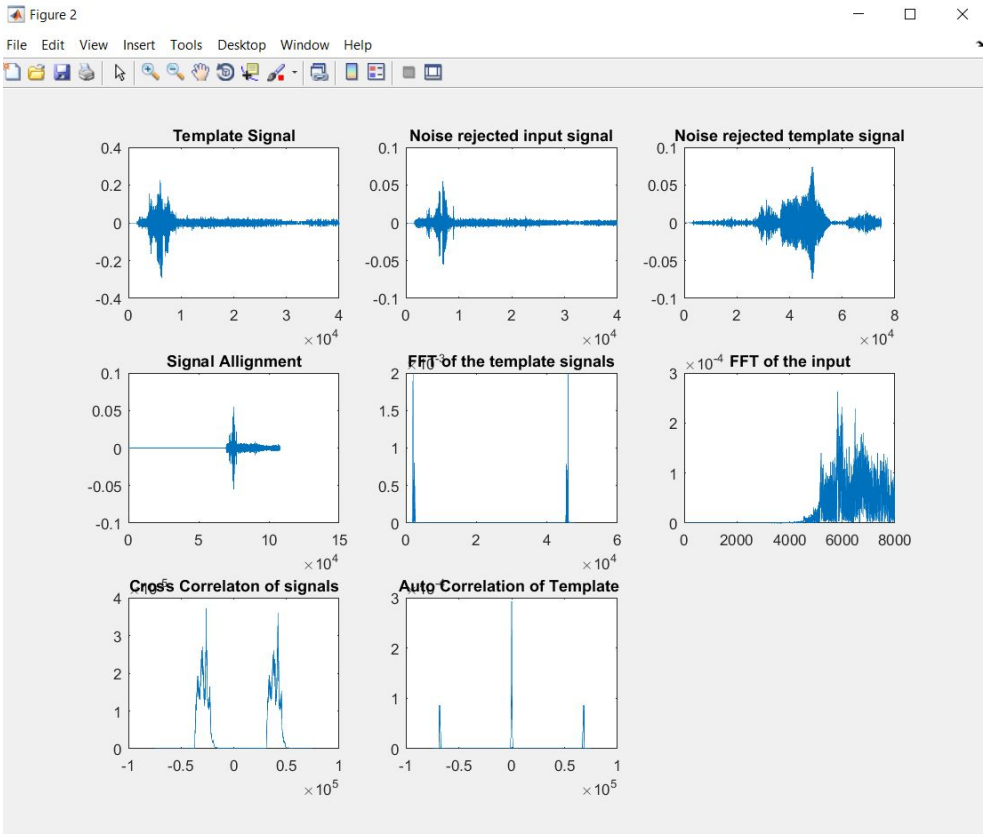


Figure 2
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

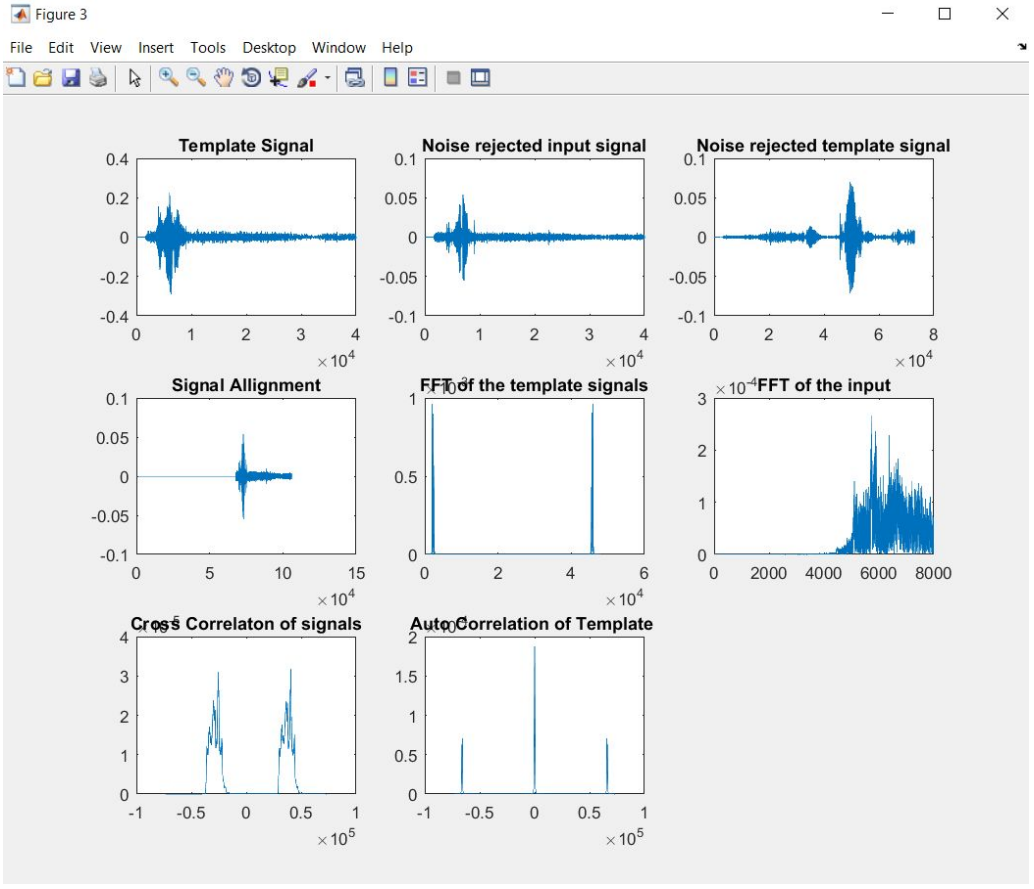


Figure 3
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

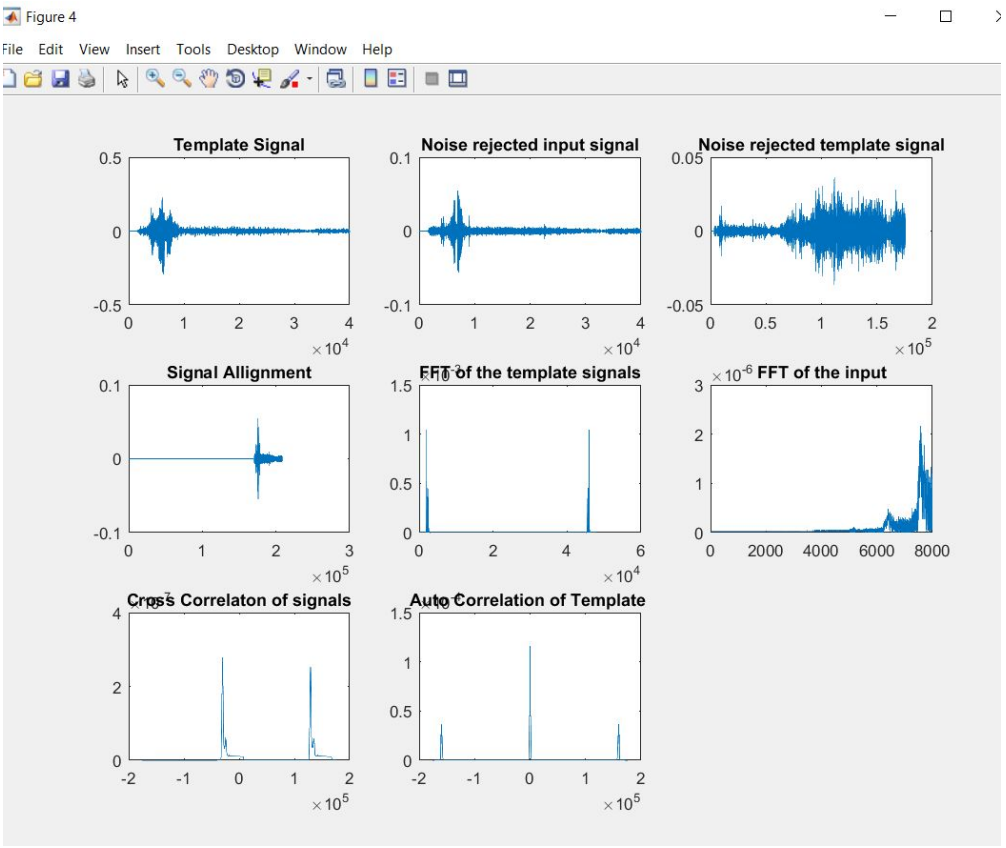


Figure 4
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

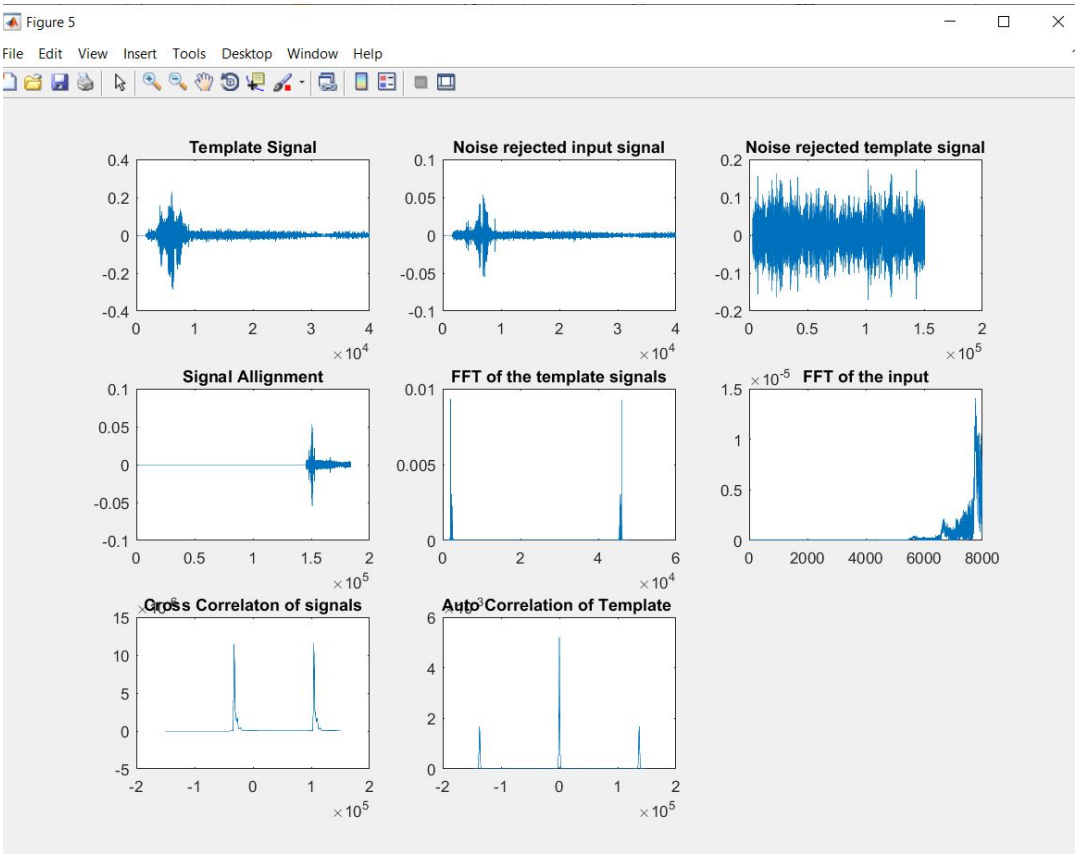


Figure 5
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

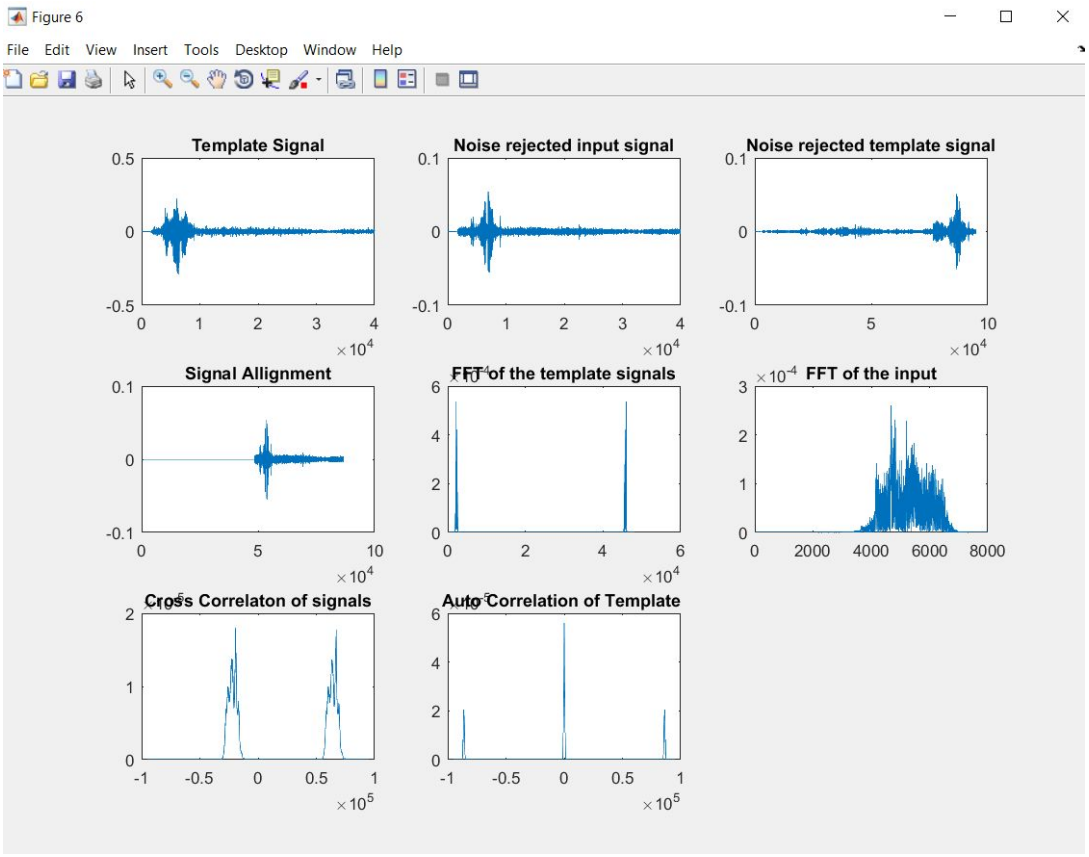


Figure 6
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

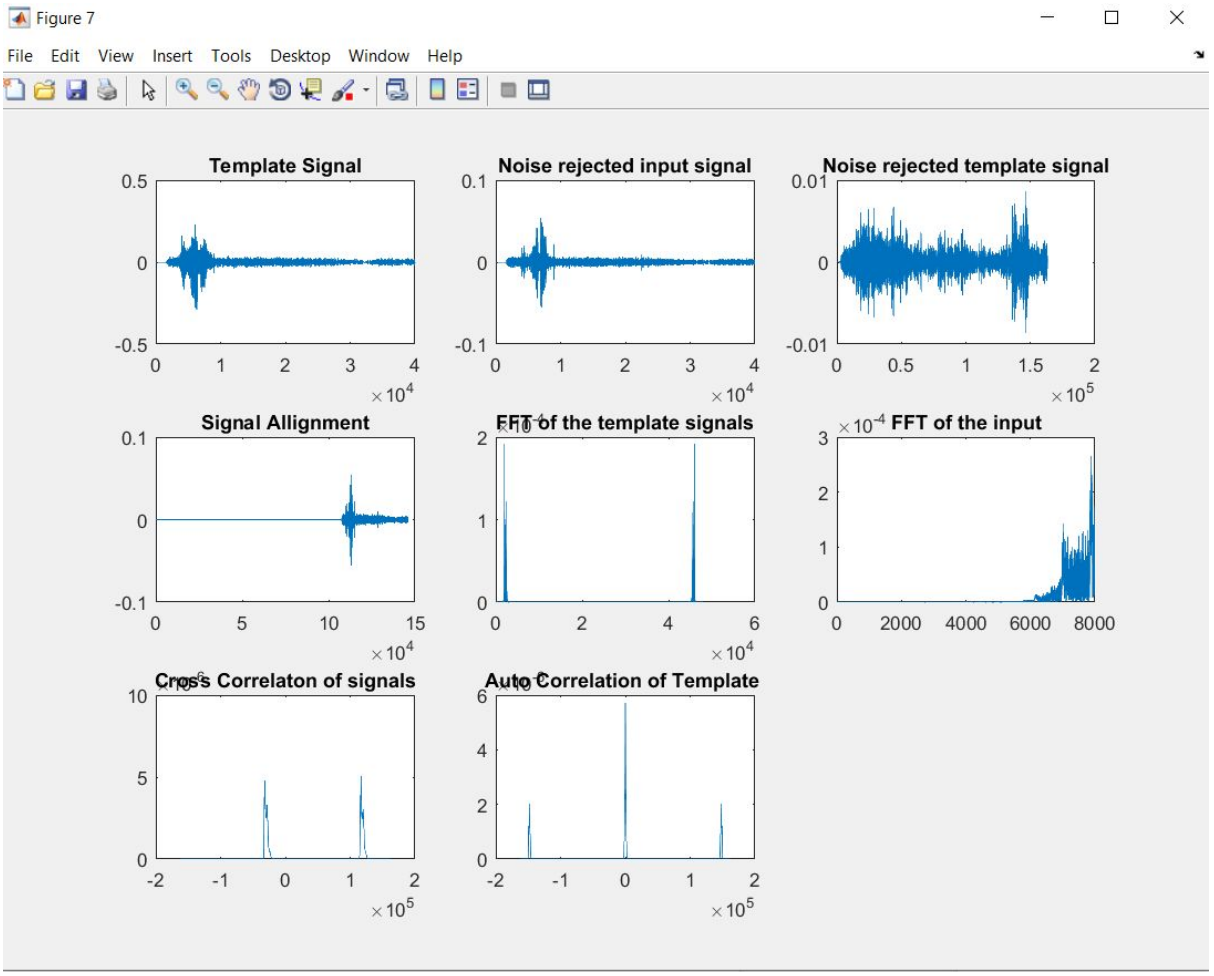


Figure 7
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

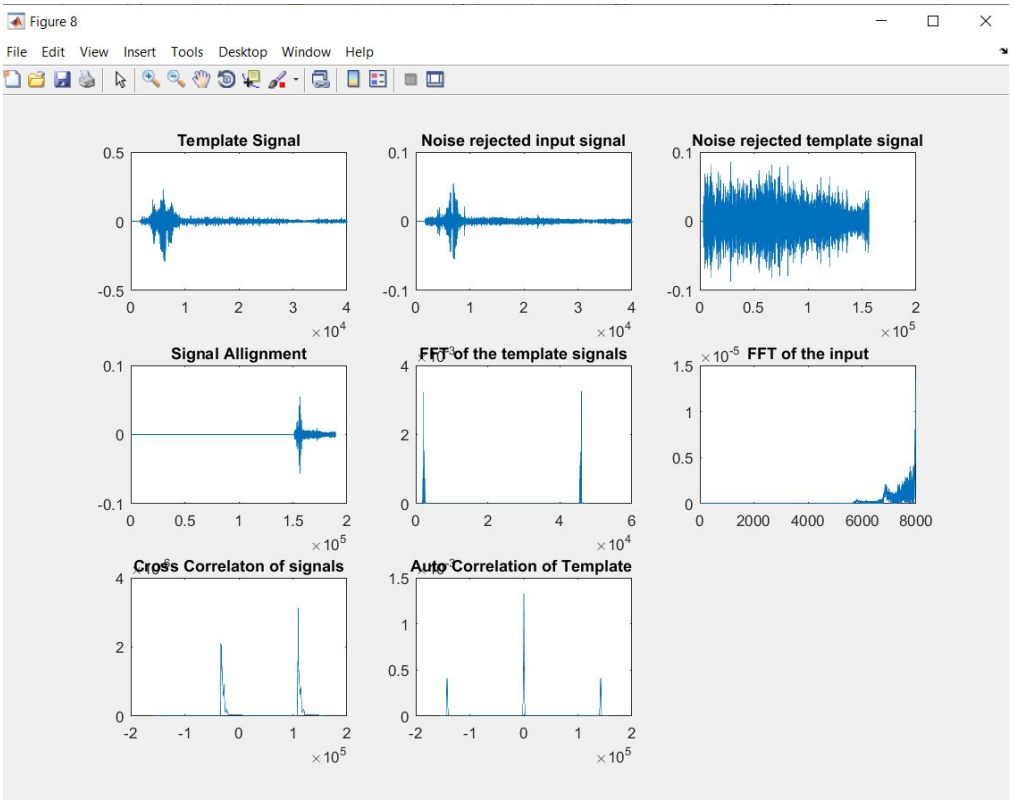


Figure 8
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

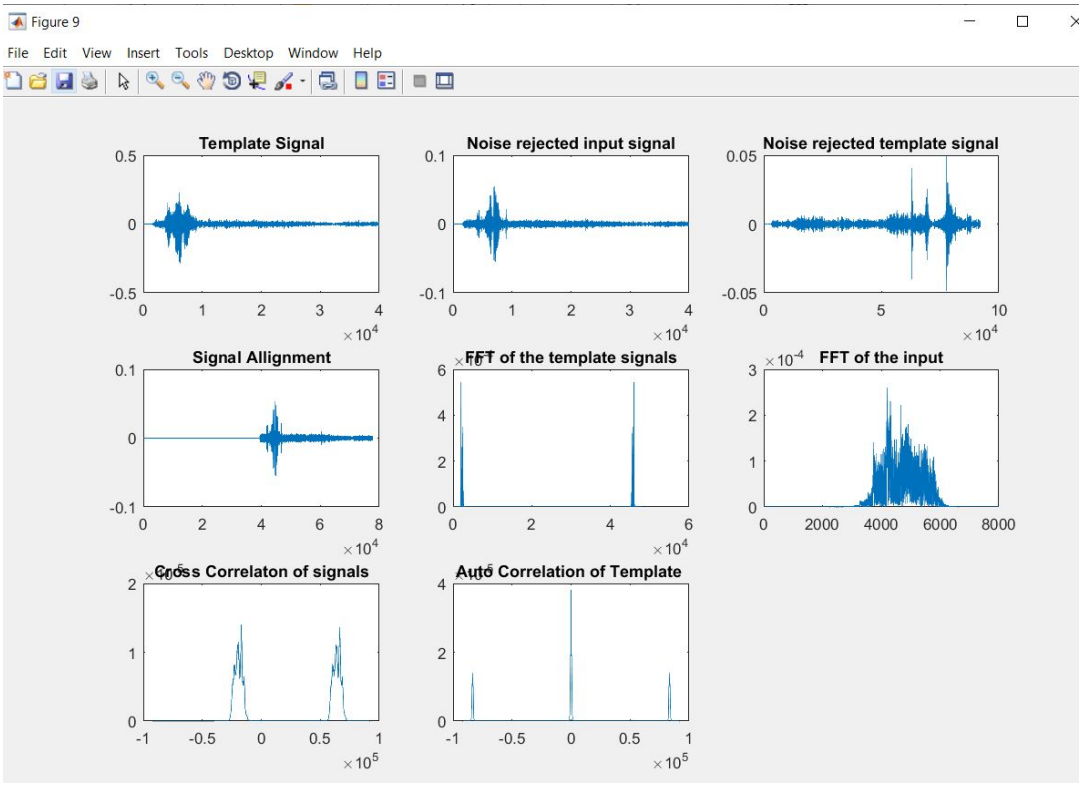


Figure 9
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

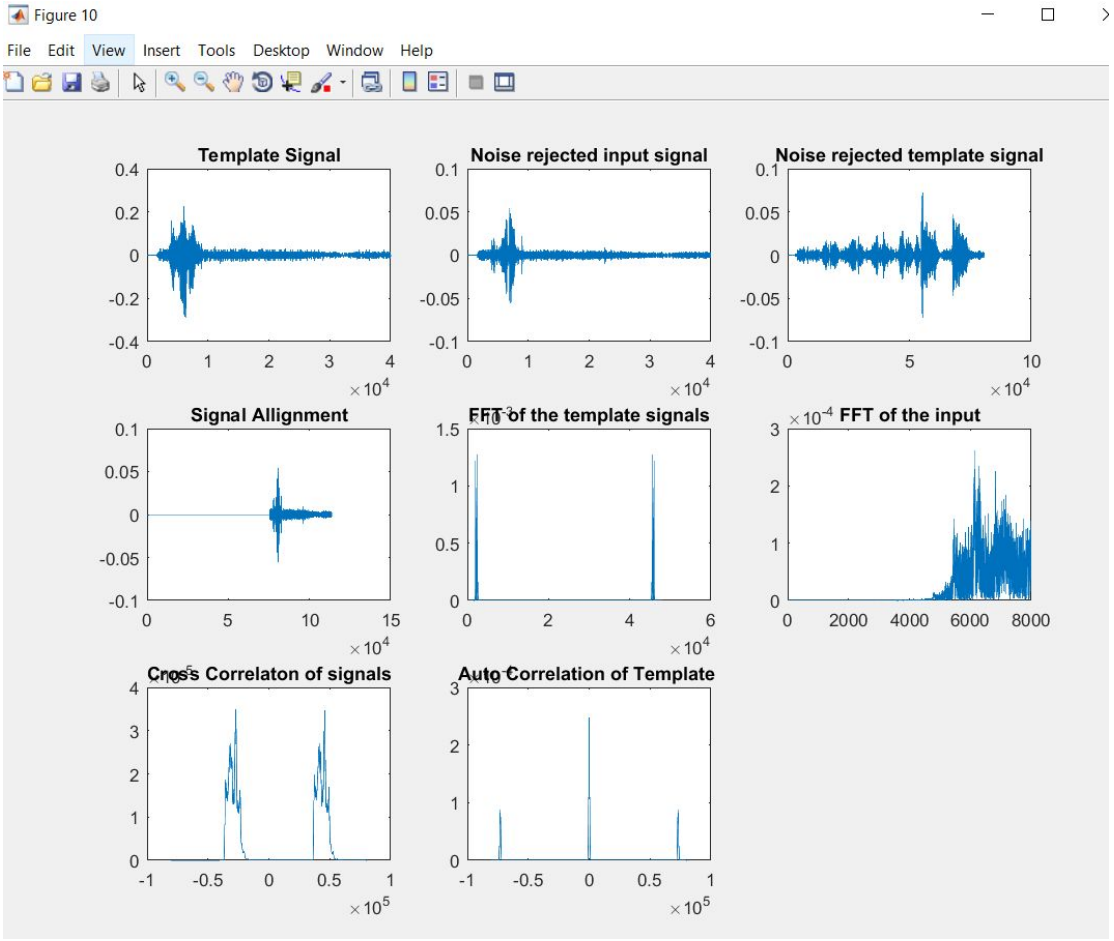


Figure 10
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

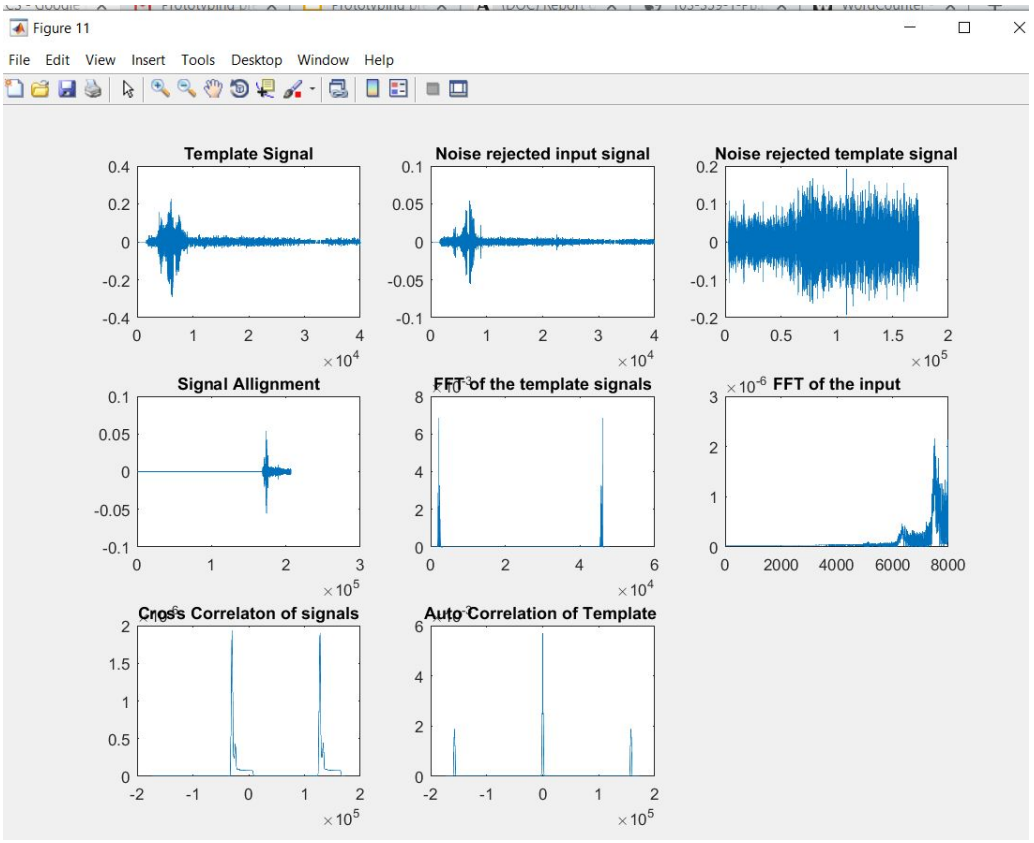


Figure 11
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

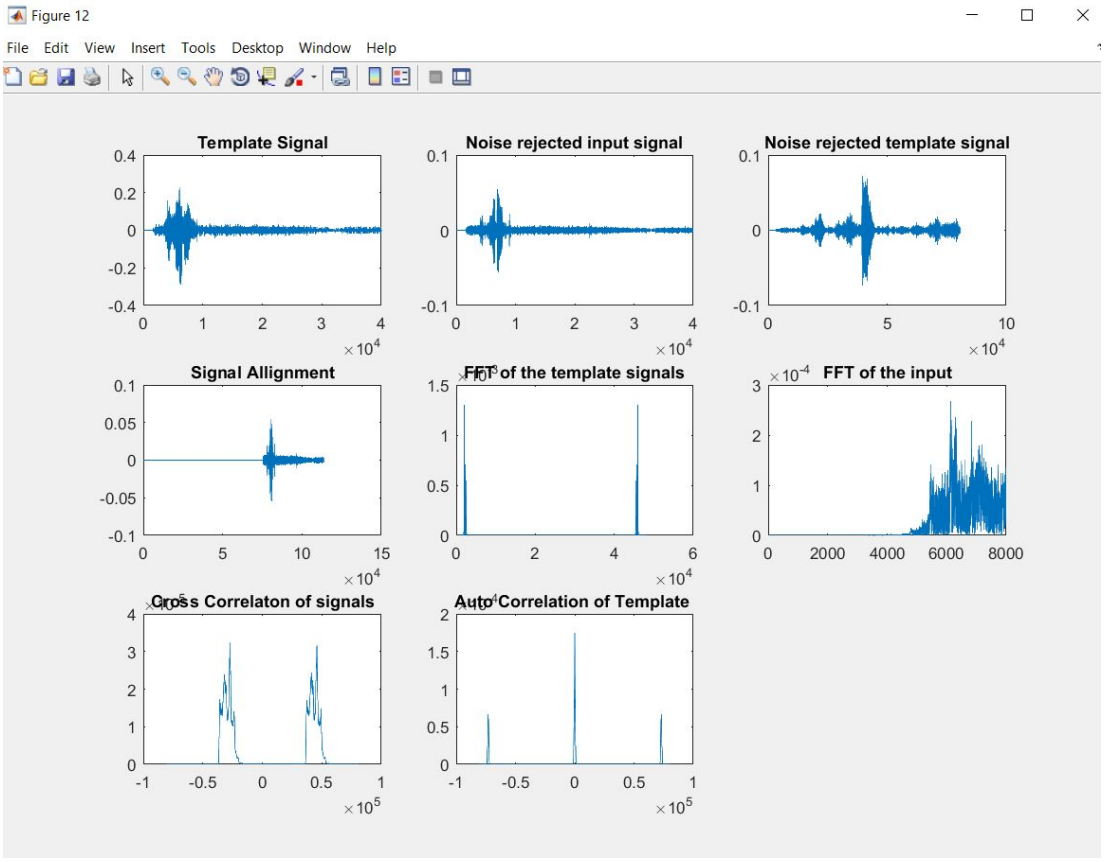


Figure 12
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

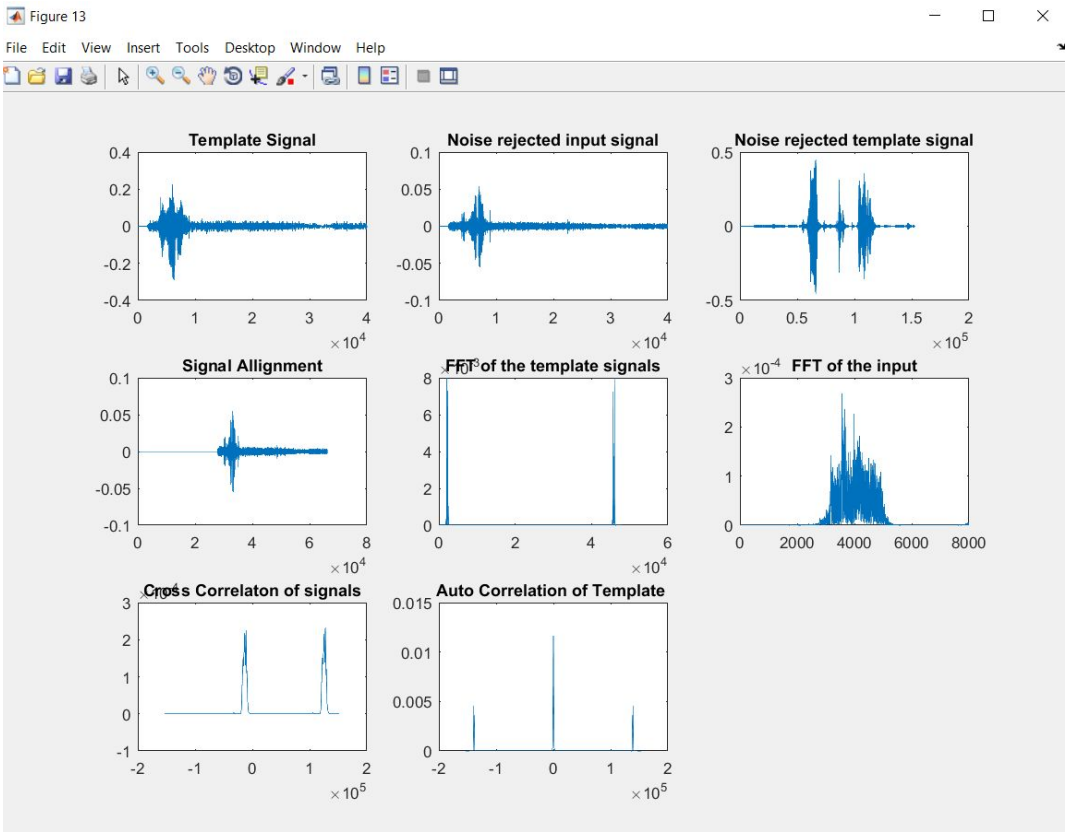


Figure 13
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

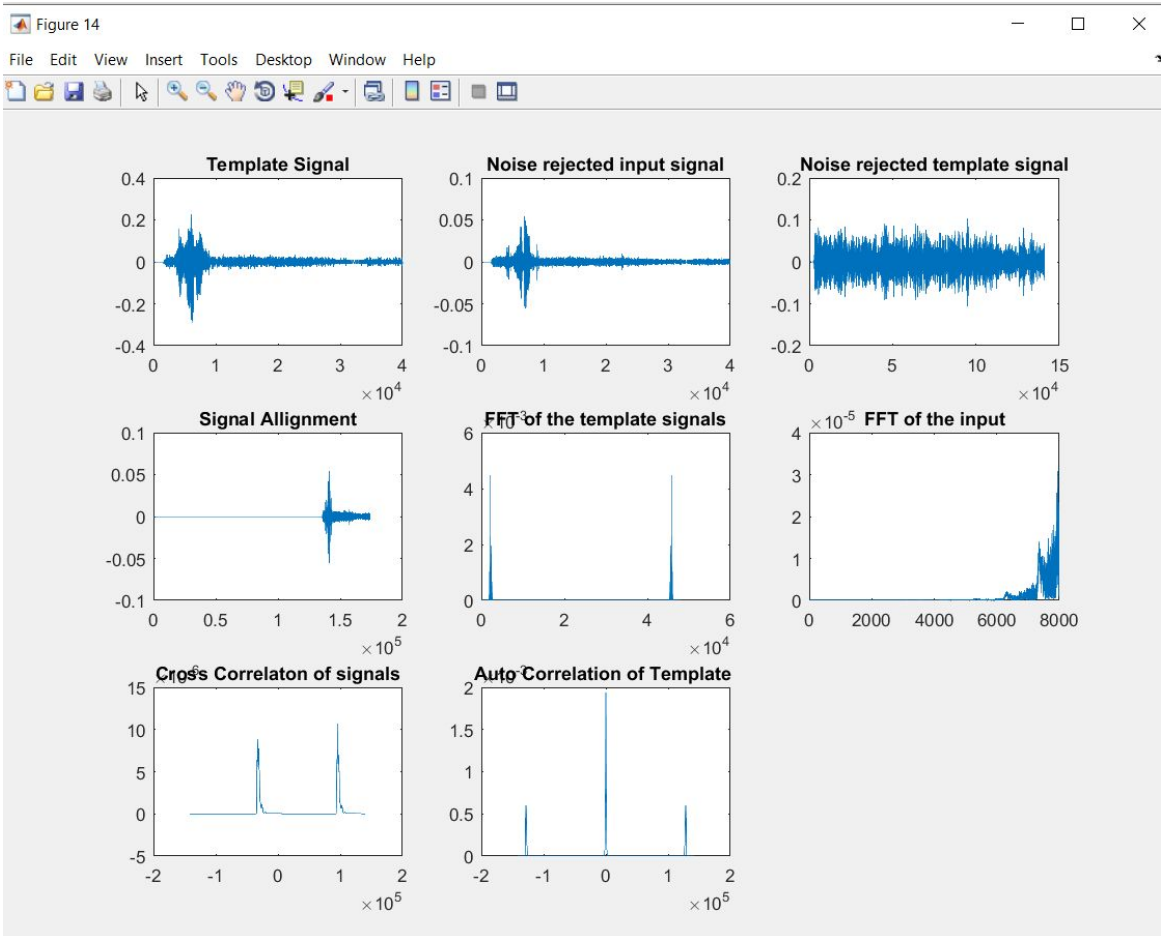


Figure 14
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

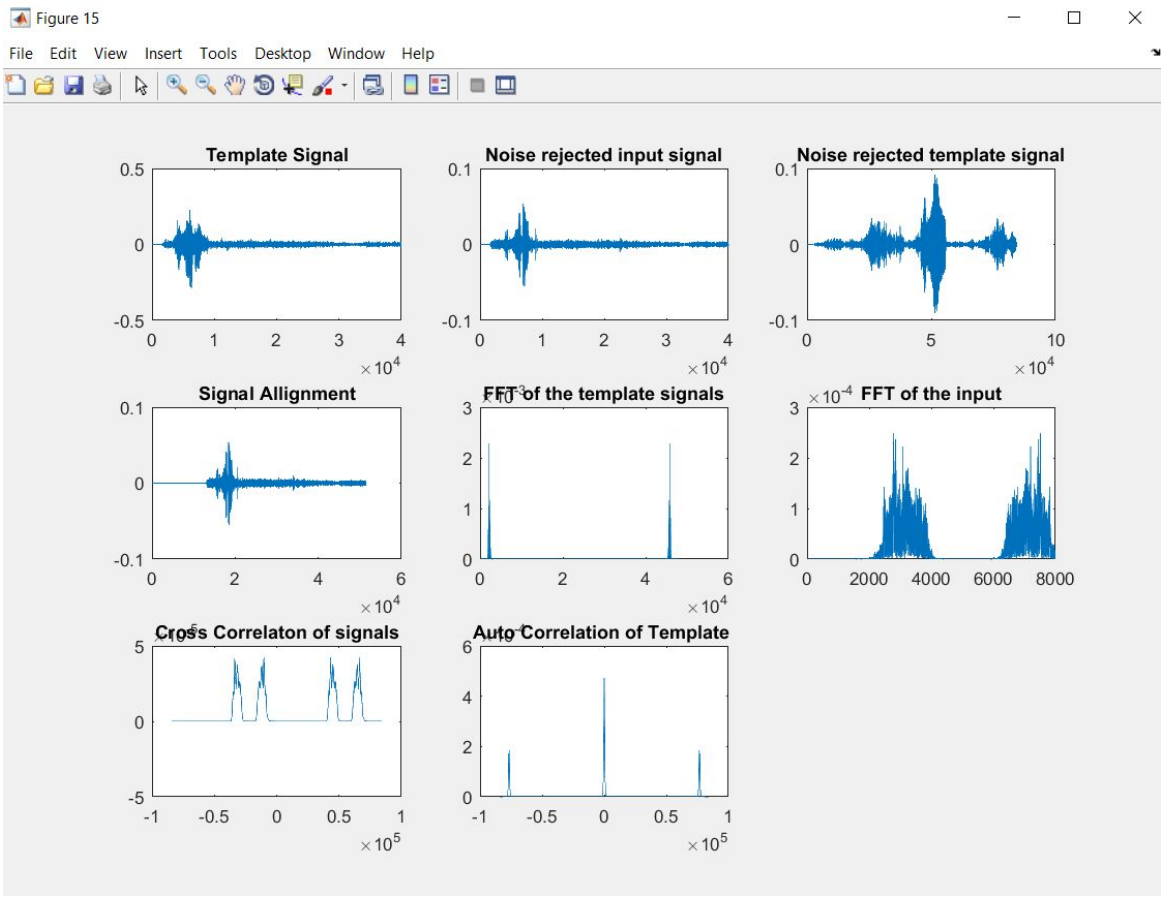


Figure 15
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

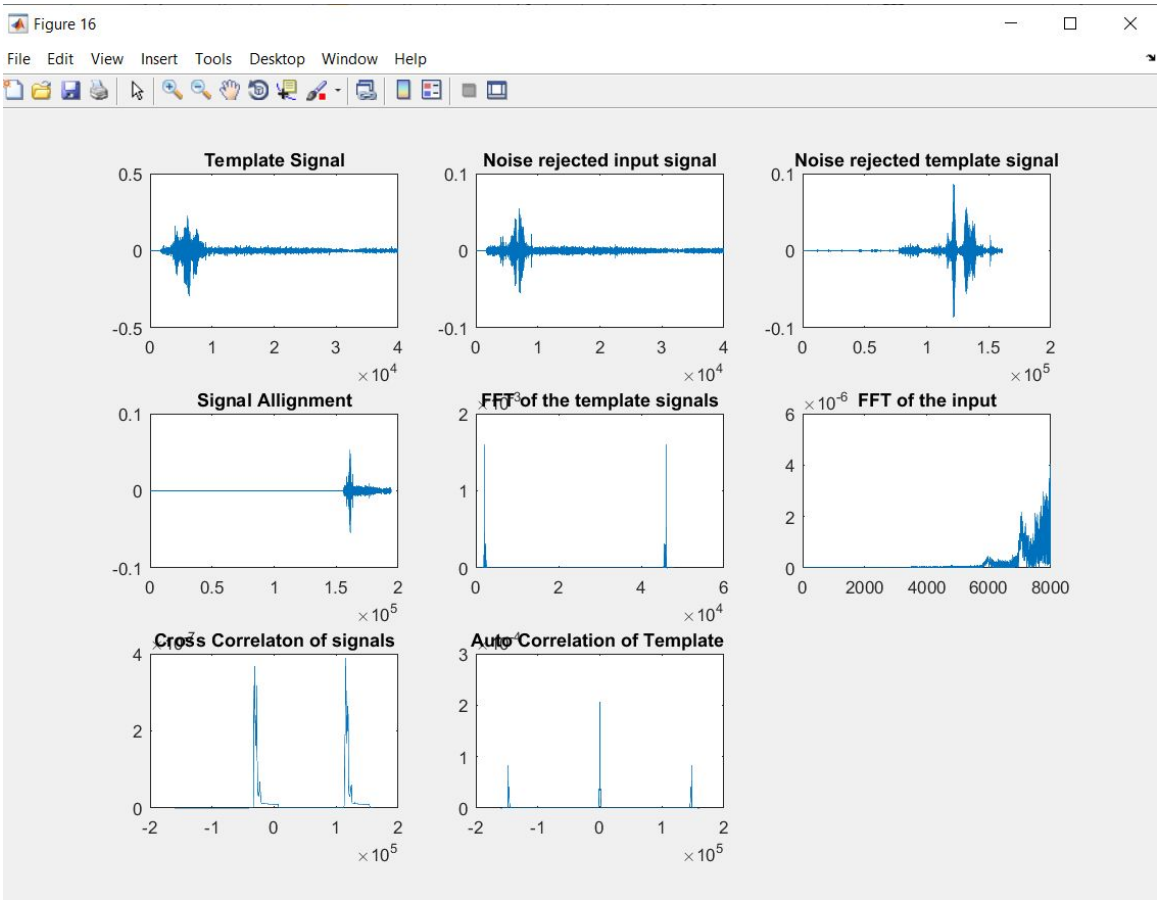


Figure 16
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

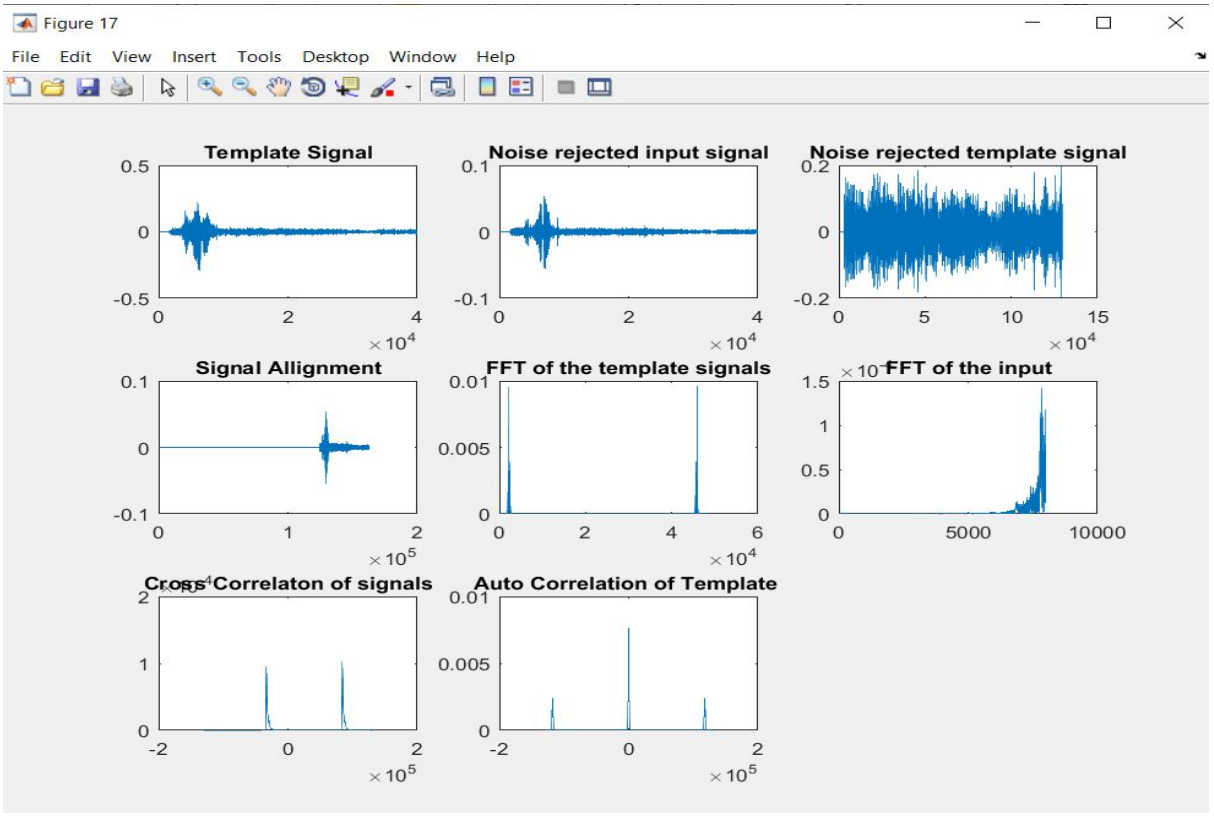


Figure 17
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

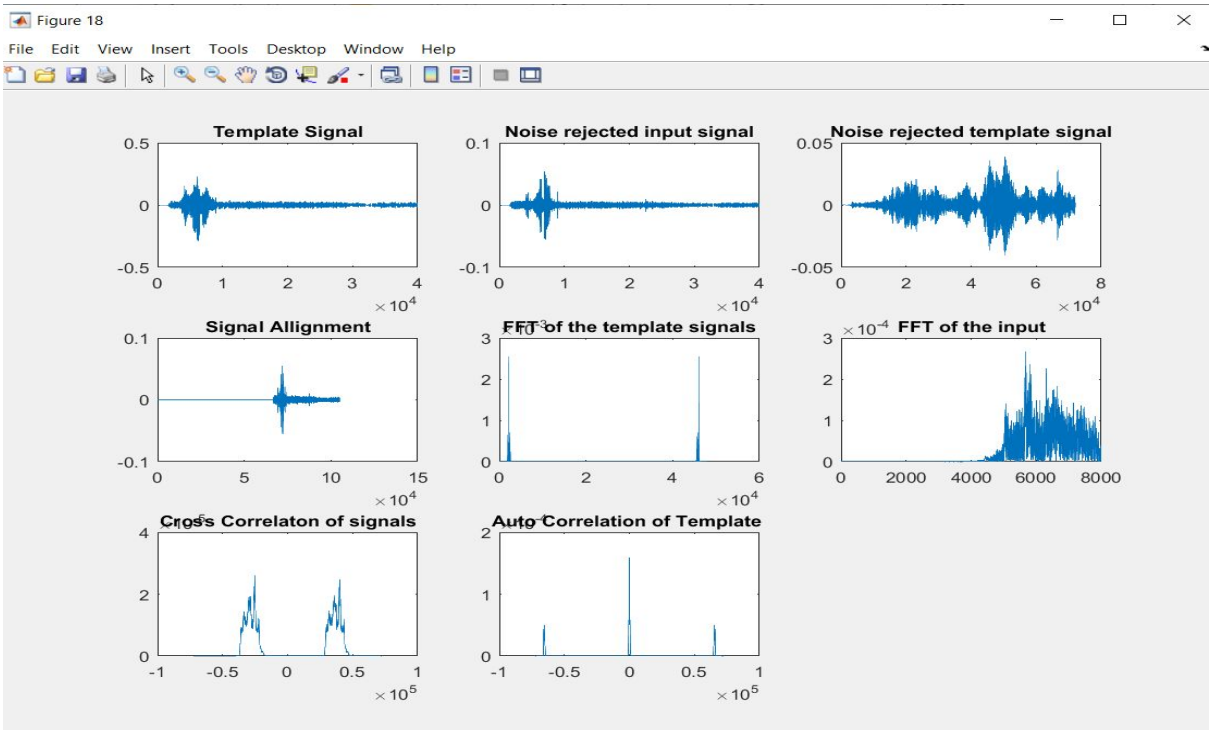


Figure 18
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

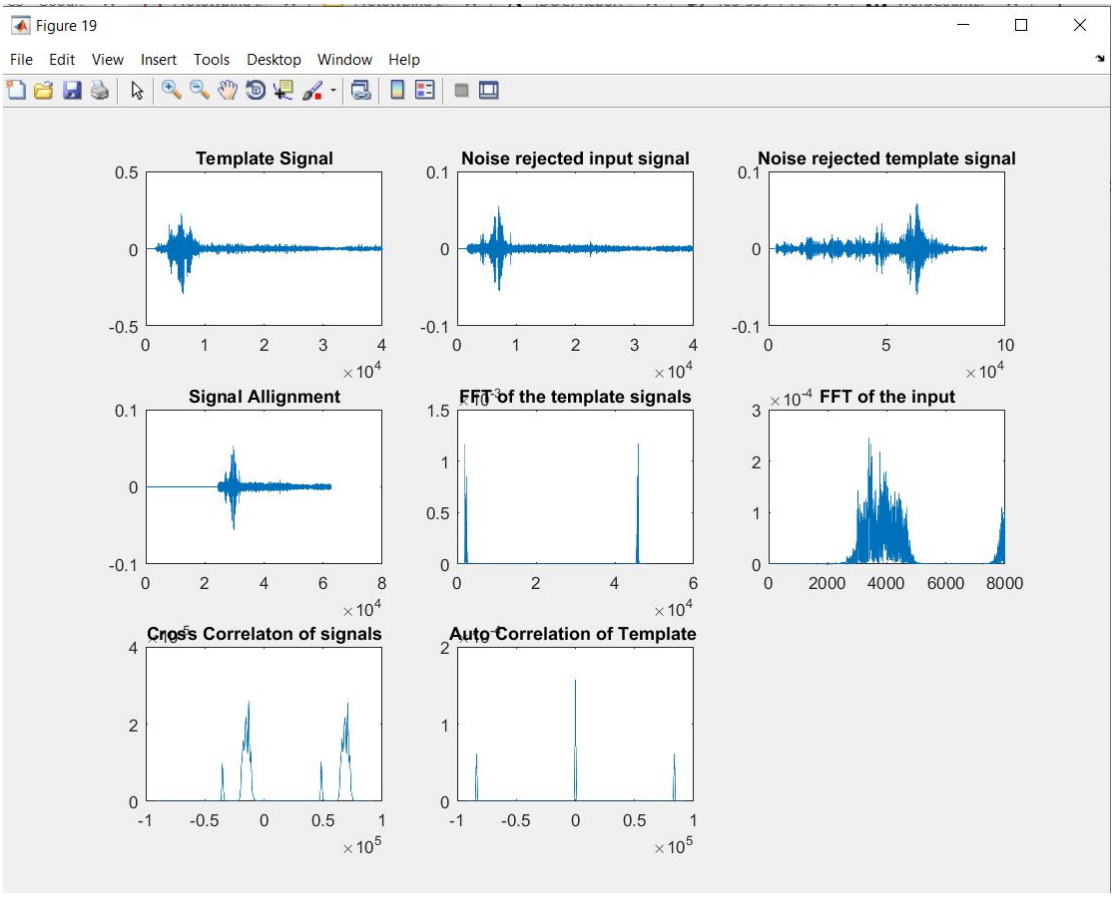


Figure 19
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

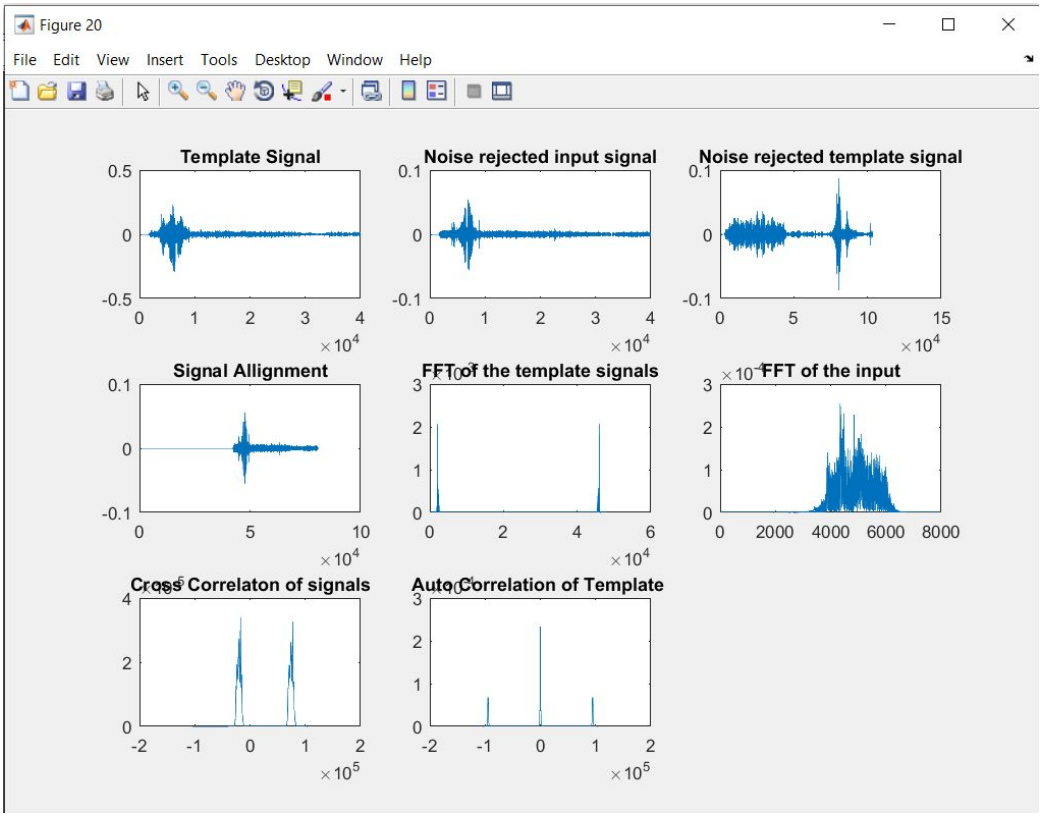


Figure 20
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

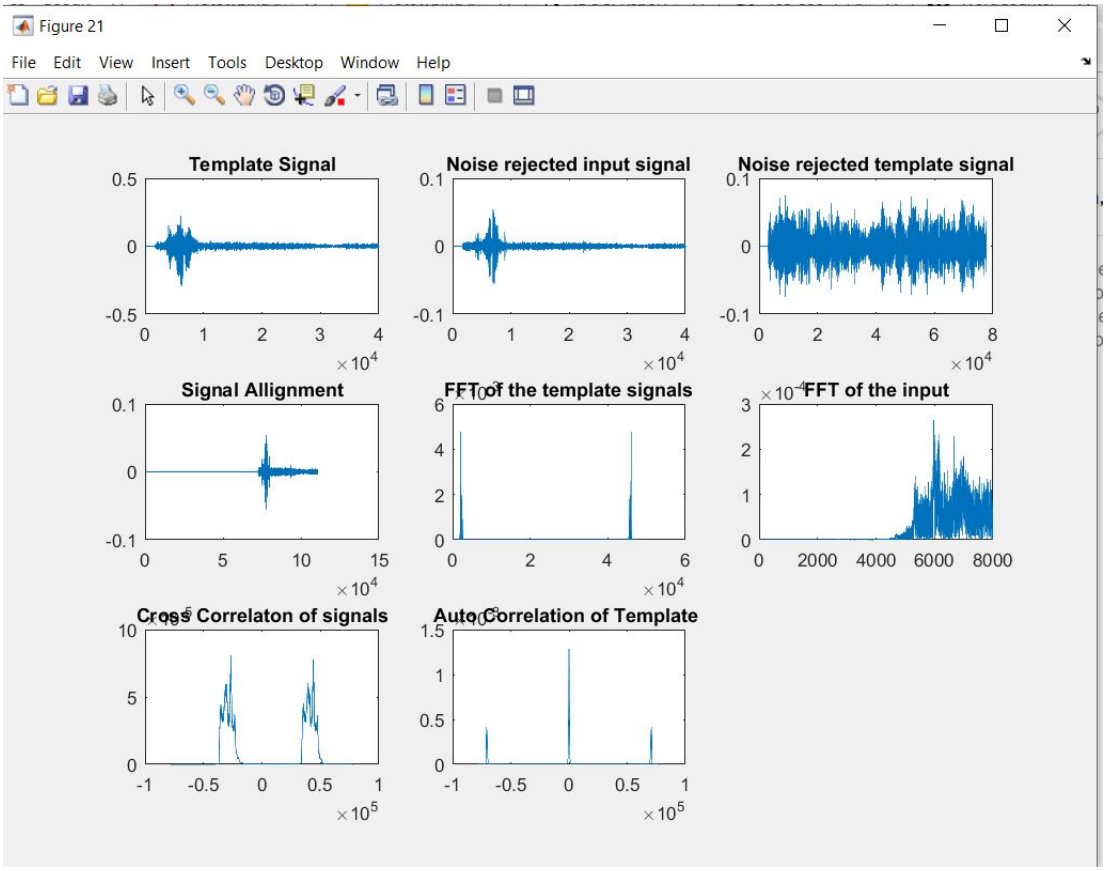


Figure 21
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

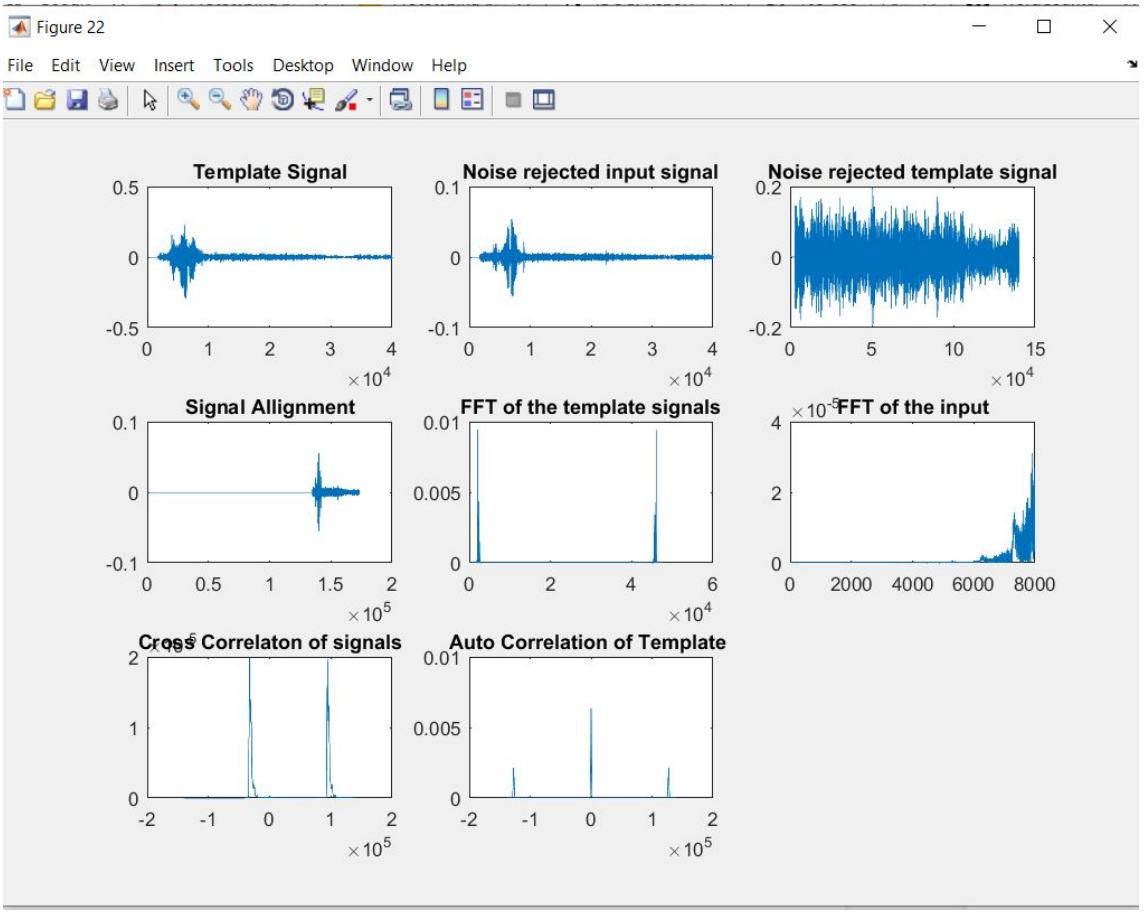


Figure 22
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

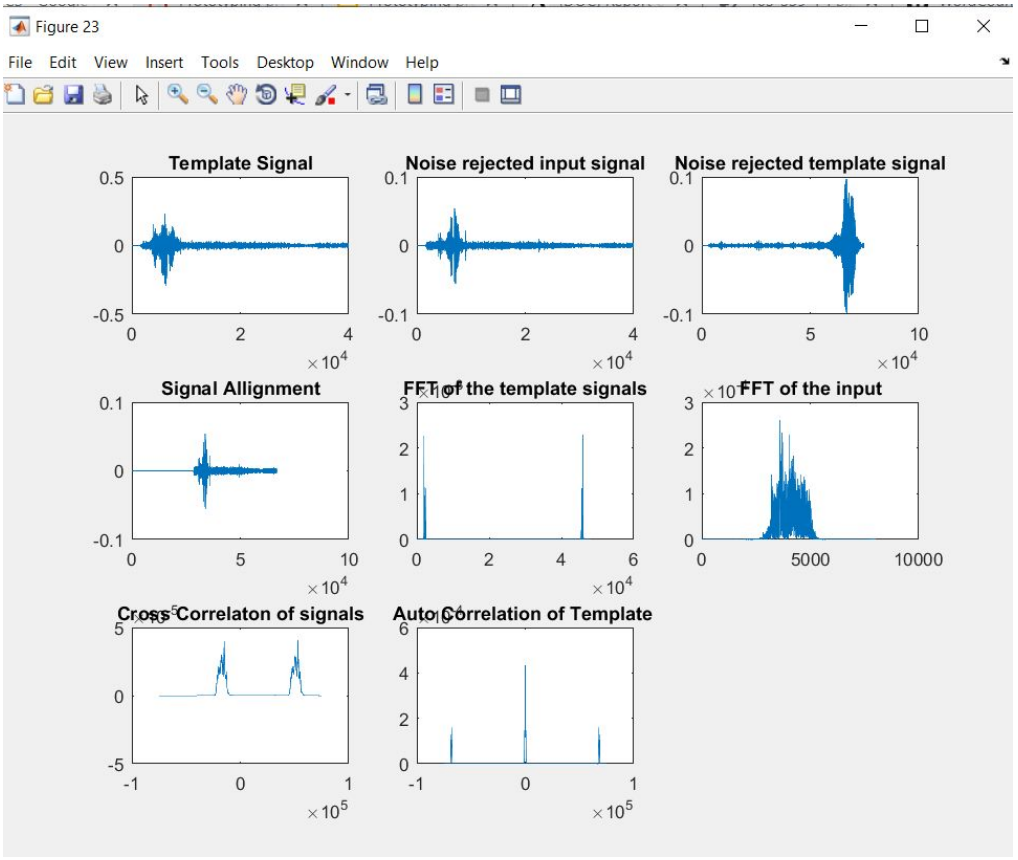


Figure 23
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

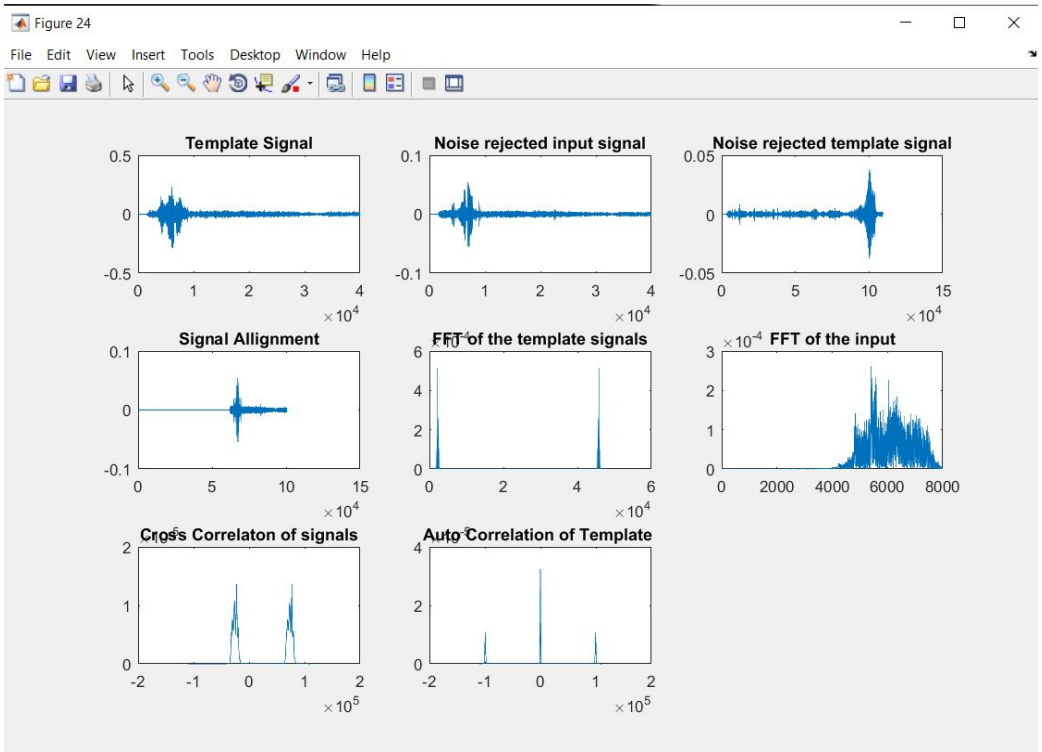


Figure 24
File Edit View Insert Tools Desktop Window Help
Template Signal
Noise rejected input signal
Noise rejected template signal
Signal Allignment
FFT of the template signals
FFT of the input
Cross Correlaton of signals
Auto Correlation of Template

Understanding our problems and overcoming:

1-We got the idea of this project from a python project where we convert speech to text. Then had to research on how to work with it in matlab.

2-We had issues with using butterworth filters and had to do a detailed research on how to use them. Then further we had issue with correlation then overcame that.

3-We had to work on creating the databases for 7 different audio files in 3 different scenarios and link everything from python to matlab.

Individual Contribution:

SANDEEP(18BIS0039)- Creating the database and linking the database to matlab.Getting the input and writing the python code to convert text to speech

PRIYAL(18BIS0052) -Learning the Concept of the project and Matlab Code for Fourier transformation and Correlation and documentation.

MADHUMITA(18BIS0030)-Noise reduction concept, using butterworth filter and MATLAB Code for the same and documentation.

(Everyone has equal contribution in every single part of Code and understanding.)

CONCLUSION

This project is a basic approach to implement speaker and speech recognition.

After recognition of various syllable this

idea can be further implementedtherecognizeentiresentencesfromvariousspeakers.This can also be used to identify parts of speech from an entire sample. Current research in this domain consists to make it more efficient and fast.

References:

- ***https://searchcustomerexperience.techtarget.com/definition/speech-recognition***
- ***https://in.mathworks.com/matlabcentral/***
- ***https://www.geeksforgeeks.org/speech-recognition-in-python-using-google-speech-api/***
- ***https://ieeexplore.ieee.org/document/5495783***

Appendix B

Matlab code:

```
clc
clear all
%cd database

datafiles = dir('database')
filename='hi.wav';
[aud,fs] = audioread(filename);
recObj=audiorecorder(fs,8,2);
%record user's voice for 2 secs
disp('start speaking');
pause(1);
recordblocking(recObj,5);
disp('end of recording');
play(recObj);
y=getaudiodata(recObj);
y = y(:,1);
figure(1)
plot(y)
title('input voice signal');
audiowrite(filename,y,8000);

for i = 3:25
   %database files
   temp_file=datafiles(i).name
   %create a recorder object
   figure(i-1)
   cd database
   [aud0,fs0] = audioread(temp_file);
   recObj0=audiorecorder(fs0,8,2)
   y0=getaudiodata(recObj);
   y0 = y0(:,1);
   subplot(331)
```

```
plot(y0)
title('Template Signal')

%use a butterworth filter of order 7 to reduce the noise of the input
%voice.
%reeading the audio file
cd ..
[f2,fs2]=audioread('hi.wav');
%determine total number of samples in audiofile
N2=size(f2,1);
%buiding a butterworth bandpass filter for removing noise components
n2=7;
beginFreq2=2000/(fs2/2);
endFreq2=3000/(fs2/2);
[b2,a2]=butter(n2,[beginFreq2,endFreq2],'bandpass');
fout2=filter(b2,a2,f2);
subplot(332)
plot(fout2)
title('Noise rejected input signal');

%use a butterworth filter of order 7 to reduce the noise of the template
%voice
%reading the audio file
cd database
[f1,fs1]=audioread(temp_file);
cd ..
%determine total number of samples in audio file
N1=size(f1,1);
%building a butterworth andpass filter for removing noise components
n1=7;
beginFreq1=2000/(fs1/2);
endFreq1=2500/(fs1/2);
[b1,a1]=butter(n1,[beginFreq1,endFreq1],'bandpass');
fout1=filter(b1,a1,f1);
subplot(333)
```

```
plot(fout1)
title('Noise rejected template signal');

%determining the leg and allinging the signals
del=finddelay(fout1,fout2);
if (del>0)
   fout1=alignsignals(fout1,fout2(:,1));
   subplot(334)
   plot(fout1);
   title('Signal Allignment');
else
   fout2=alignsignals(fout2(:,1),fout1);
   subplot(334)
   plot(fout2);
   title('Signal Allignment');
end


%determine the FFT of the template signal
x=fft(fout1,N1);
axis_x1=(fs1/N1).*(0:N1-1);
x=abs(x(1:N1)./(N1/2));
%determine the FFT of the input signal
y=fft(fout2);
axis_x2=(fs2/N2).*(0:N2-1);
y=abs(y(1:N2)./(N2/2));
%plolting the FFT of the signals
subplot(335)
plot(axis_x1,x)
title('FFT of the template signals')
subplot(336)
plot(axis_x2,y)
title('FFT of the input')
```

```
 %Correlation of the signals
    y=y';
    y=y(1,:);
    y=y';
    x=x';
    x=x(1,:);
    x=x';
    %cross correlation
    z=xcorr(x,y);
    mi=max(z);
    l=length(z);
    t=-((l-1)/2):1:((l-1)/2);
    t=t';

    %auto correlation

    p = xcorr(x);
    m1=max(p);
    l1=length(p);
    t1=-((l1-1)/2):1:((l1-1)/2);
    t1=t1';

    %plotting the Correlation of the signals
    subplot(337)
    plot(t,z);
    title('Cross Correlaton of signals')
    subplot(338)
    plot(t1,p);
    title('Auto Correlation of Template ')
end
```

Code Reference : -

All the codes are uploaded in GitHub.

Link : - https://github.com/unknown-guy-1610/Signal_Analysis