CHAPTER: 4 IMPLEMENTATION AND RESULTS

The implementation is done by using the MATLAB. The description of the tool and implementation detail and the results are described in this chapter.

4.1 MATLAB OVERVIEW

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.[116] You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas. MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications [117].

4.2 FEATURES OF MATLAB

1. High-level language for technical computing
2. Development environment for managing code, files, and data.
3. Interactive tools for iterative exploration, design, and problem solving
5. 2-D and 3-D graphics functions for visualizing data.
6. Tools for building custom graphical user interfaces.
7. Functions for integrating MATLAB based algorithms with external applications and Languages, such as C, C++, FORTRAN, Java™, COM, and Microsoft® Excel®
4.3 HISTORY OF MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation [118].

Cleve Moler, the chairman of the computer science department at the University of New Mexico, started developing MATLAB in the late 1970s. He designed it to give his student’s access to LINPACK and EISPACK without them having to learn FORTRAN. It soon spread to other universities and found a strong audience within the applied mathematics community. Jack Little, an engineer, was exposed to it during a visit Moler made to Stanford University in 1983. Recognizing its commercial potential, he joined with Moler and Steve Bangert. They rewrote MATLAB in C and founded Math Works in 1984 to continue its development. These rewritten libraries were known as JACKPAC. In 2000, MATLAB was rewritten to use a newer set of libraries for matrix manipulation, LAPACK.

MATLAB was first adopted by researchers and practitioners in control engineering, Little's specialty but quickly spread to many other domains. It is now also used in education, in particular the teaching of linear algebra and numerical analysis, and is popular amongst scientists involved in image processing. [118]

It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

1. Math and computation
2. Algorithm development
3. Modeling, simulation, and prototyping
4. Data analysis, exploration, and visualization
5. Scientific and engineering graphics
6. Application development, including graphical user interface building.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning [119]. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a
program in a scalar non-interactive language such as C or FORTRAN. The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science [119]. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others [120]

4.4. **Implementation**

The proposed algorithm is simulated using the MATLAB. This implementation takes the word spoken through microphone as the input and output the corresponding word. In between various graphs are used to represent the progress of the process. The input words are taken by using the different people speech.
The results for the word ‘hello’ are shown from the figure 28 to the figure 32. The figure 28 show the original sound. The word spoken through the microphone is recorded by the MATLAB function and then represented in the form of the signal shown in figure 28.

Figure 28: Original Sound
The original speech may be of low pitch. In other words, the power of the original input signal can be low. So the signal is boost up in the pre-emphasis phase. The figure 29 shows the signal of the speech after the pre-emphasis of the original speech signal.

![Figure 29: Sound after Pre-emphasis](image-url)
The signal can be represented in the form of spectrogram. The spectrogram basically represents the frequency content of the signal. The figure 30 represents the spectrogram of the speech signal after pre-emphasis phase. The amount of frequency can be known by the color bar shown the figure.

Figure 30: Spectrogram Of Noisy Speech
The harmonic regeneration is used to reduce the noise. The harmonic regenerator wiener filter is used to reduce the noise in this work. The spectrogram of the reduced noise signal is shown in the figure 31. This process has improved the quality of the speech signal.
The speech signal corresponding to the spectrogram shown in figure 31 is given in the figure 32. It can be seen that fluctuations have been reduced resulting improved quality speech.

![Figure 32: Speech After Noise Reduction](image)

The volterra prediction is used to predict the word. The third order volterra predictor is used for the long term prediction. The output of the volterra is compared with the output of the trained signal; the closest match is given as the output.

Similarly for the word ‘RAKESH’ and for ‘JAI MATA DI’ the graphs are shown in tables below. The graph with their brief description:
Table 2 Evaluation For word 'RAKESH'

<table>
<thead>
<tr>
<th>Original Sound</th>
<th>Sound after Pre-emphasis</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Original Sound" /></td>
<td><img src="image2.png" alt="Sound after Pre-emphasis" /></td>
</tr>
</tbody>
</table>
Spectrogram Of Noisy Speech

Spectrogram after Noise Reduction
Spectrogram after Noise Reduction

Speech After Noise Reduction
Table 3 Evaluation For word 'JAI MATA DI'

<table>
<thead>
<tr>
<th>Original Sound</th>
<th>Sound after Pre-emphasis</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Original Sound" /></td>
<td><img src="image2.png" alt="Sound after Pre-emphasis" /></td>
</tr>
</tbody>
</table>
Spectrogram Of Noisy Speech

Spectrogram after Noise Reduction
Spectrogram after Noise Reduction

Speech After Noise Reduction
In the training phase 50 words have been recorded. Each word is recorded 5 times. So for 50 words recording is done 250 times. Then proposed model is applied on these 250 words and result is saved into a database. Dynamic time warping is used to account for signals that have different durations. This is useful for trying to match speech signals because of the stretching and compression of the different phonetic portions of the speech signals. A linear time alignment comparison is not sufficient. Dynamic time warping finds the minimum distortion between frames of the input signal and the template signal. The Euclidean distance measure is more sensitive to distortions between signals on the time axis. Dynamic time warping tries to take care of this problem by shifting the time axis in order to detect signals that are out of the phase with each other. The system is tested for speaker dependent and speaker independent. In speaker dependent system, the template included for reference for the same speaker was used. Experiment was done on the speech patterns from the five speakers and in speaker independent system; the patterns of the speaker were not included. Finally the percentage word error rate was calculated for both types of systems. The % word error rate is calculated using the following method.

\[
\text{\% word error rate} = \frac{\text{no. of mismatch words}}{\text{total no. of words uttered}} \times 100
\]

Percentage word error rate for speaker Independent systems is given below in the table:-

**Speaker Independent System**

<table>
<thead>
<tr>
<th></th>
<th>Wavelet transform</th>
<th>Hmm Based</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample 1</td>
<td>.067</td>
<td>.061</td>
<td>.059</td>
</tr>
<tr>
<td>Sample 2</td>
<td>.065</td>
<td>.060</td>
<td>.057</td>
</tr>
<tr>
<td>Sample 3</td>
<td>0.069</td>
<td>.061</td>
<td>.058</td>
</tr>
<tr>
<td>Sample 4</td>
<td>0.07</td>
<td>.063</td>
<td>0.059</td>
</tr>
</tbody>
</table>

For speaker dependent systems, the percentage word error rate is tabulated below:
The above table shows that the % word error rate is reduced in the proposed system. Hence, improved QOS in the speech recognition.