

# Voice Scrambler and Descrambler System Using Frequency Inversion Technique

\*End Semester Project - EE-232: Signals and Systems

Rana Muhammad Bilal (465138), Muhammad Jahan Zaib (458924), Umer Nisar (466568)

School of Electrical Engineering and Computer Science (SEECS)

National University of Sciences and Technology (NUST), Islamabad, Pakistan

Course Instructor: Dr. Usman Zabit

**Abstract**—This project implements a voice scrambler and descrambler system using frequency inversion technique. Speech signals are scrambled through carrier modulation and Butterworth low-pass filtering, making them unintelligible. The original signal is recovered by applying the same process twice. MATLAB implementation validates the system with time-domain, frequency-domain, and spectrogram analysis.

**Index Terms**—Voice scrambler, frequency inversion, Butterworth filter, carrier modulation, MATLAB

## I. INTRODUCTION

Voice scrambling is essential for secure communication, where speech signals are intentionally distorted to prevent unauthorized listening. This project implements frequency inversion technique, which inverts the frequency spectrum through modulation and filtering. The key advantage is that applying the process twice recovers the original signal.

### A. Objectives

- Design and implement voice scrambler using frequency inversion
- Design optimal Butterworth low-pass filter for the system
- Recover original signal through descrambling process
- Analyze system performance using MATLAB

## II. THEORETICAL BACKGROUND

### A. Speech Signal Characteristics

Human speech occupies frequency range 300–3400 Hz, with most intelligible information in 300–3000 Hz band. This bandwidth constraint guides our filter design.

### B. Frequency Inversion Principle

The scrambling process modulates speech signal  $x(t)$  with carrier frequency  $f_c$ :

$$y(t) = x(t) \cdot \cos(2\pi f_c t) \quad (1)$$

In frequency domain:

$$Y(f) = \frac{1}{2}[X(f - f_c) + X(f + f_c)] \quad (2)$$

This creates upper and lower sidebands. Low-pass filtering removes the upper sideband, keeping only the inverted lower sideband. For descrambling, applying the same process:

$$z(t) = y(t) \cdot \cos(2\pi f_c t) = x(t) \cdot \cos^2(2\pi f_c t) \quad (3)$$

Using  $\cos^2(\theta) = \frac{1+\cos(2\theta)}{2}$ :

$$z(t) = x(t) \cdot \frac{1 + \cos(4\pi f_c t)}{2} \quad (4)$$

After low-pass filtering (removes  $2f_c$  component):

$$z(t) = \frac{x(t)}{2} \quad (\text{Original recovered}) \quad (5)$$

## III. FILTER DESIGN

### A. Butterworth Low-Pass Filter

The Butterworth filter is chosen for its maximally flat frequency response in the passband, minimizing signal distortion. The magnitude response is:

$$|H(j\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}} \quad (6)$$

where  $\omega_c$  is cutoff frequency and  $n$  is filter order.

### B. Filter Specifications

TABLE I  
BUTTERWORTH FILTER PARAMETERS

Parameter	Value
Filter Type	Butterworth Low-Pass
Filter Order (n)	8
Cutoff Frequency ( $f_c$ )	3000 Hz
Passband Ripple	0 dB (Maximally Flat)
Stopband Attenuation	-48 dB at 6000 Hz

### C. Filter Design Rationale

- **Order Selection:** Order 8 provides sharp transition between passband and stopband with acceptable phase distortion.
- **Cutoff Frequency:** 3000 Hz preserves speech bandwidth while removing upper sideband.
- **Butterworth Choice:** Maximally flat passband prevents in-band distortion of speech.

The normalized cutoff frequency for digital implementation:

$$\omega_n = \frac{f_c}{f_s/2} = \frac{3000}{f_s/2} \quad (7)$$

#### D. Filter Implementation

MATLAB implementation using butter function:

```
fc = 3000; % Cutoff frequency
filter_order = 8; % Filter order
[b, a] = butter(filter_order, fc/(Fs/2), 'low');
```

#### IV. SYSTEM DESIGN

##### A. System Parameters

TABLE II  
SYSTEM DESIGN PARAMETERS

Parameter	Value
Carrier Frequency ( $f_{carrier}$ )	3700 Hz
Speech Bandwidth	300–3400 Hz
LPF Cutoff	3000 Hz
Filter Order	8
Sampling Rate ( $F_s$ )	Variable (8–48 kHz)

##### B. System Flow Diagram

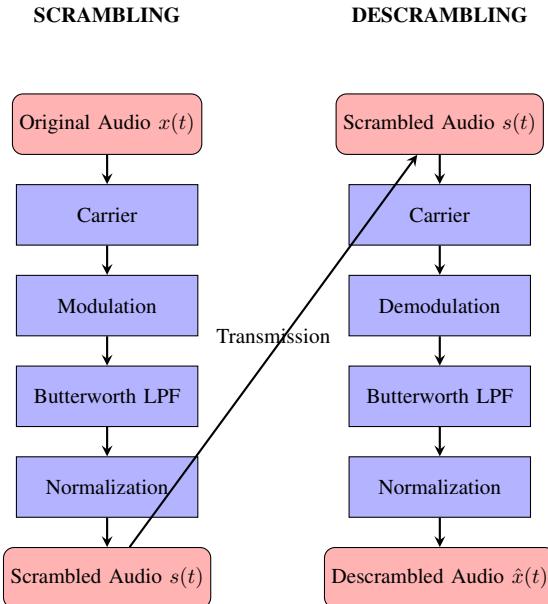


Fig. 1. System Flow Diagram

#### V. RESULTS AND ANALYSIS

##### A. Frequency Domain Analysis

Figure 2 illustrates the magnitude spectrum of the speech signal at three critical stages: the original baseband signal, the scrambled signal (inverted), and the successfully recovered descrambled signal.

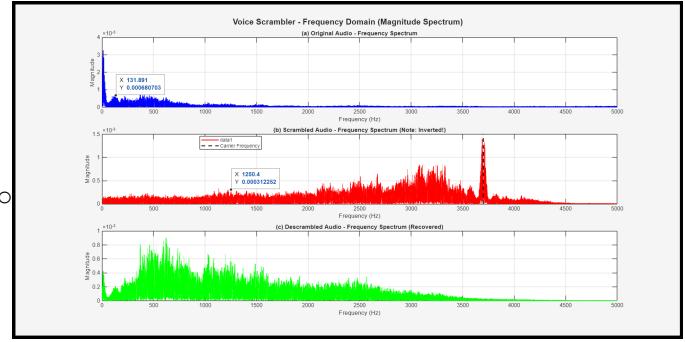


Fig. 2. FFT Analysis: Magnitude Spectrum of Original, Scrambled, and Descrambled Signals.

##### Key Observations:

- Original:** The spectrum shows high energy concentration in the 300–3000 Hz range.
- Scrambled:** The spectrum is inverted; lower frequencies are shifted to the higher end of the band and vice-versa.
- Descrambled:** The recovered spectrum shows high fidelity to the original, with the  $2f_c$  components successfully attenuated by the Butterworth LPF.

#### VI. IMPLEMENTATION

##### A. MATLAB Implementation

The system was implemented in MATLAB with the following stages:

```
% 1. Audio Acquisition
[original_audio, Fs] = audioread(filename);
if size(original_audio, 2) > 1
    original_audio = mean(original_audio, 2);
end
t = (0:length(original_audio)-1) / Fs;

% 2. Scrambling
fc_carrier = 3700;
carrier = cos(2*pi*fc_carrier*t');
modulated_signal = original_audio .* carrier;
scrambled_audio = filter(b, a, modulated_signal);

% 3. Descrambling
demodulated_signal = scrambled_audio .* carrier;
descrambled_audio = filter(b, a, demodulated_signal);
```

#### VII. CONCLUSION

This project successfully implemented a voice scrambler and descrambler system using frequency inversion technique. The 8th-order Butterworth low-pass filter with 3000 Hz cutoff effectively removed unwanted sidebands while preserving speech quality. The frequency domain analysis confirms successful scrambling through spectrum inversion and accurate signal recovery.

## REFERENCES

- [1] S. Haykin and M. Moher, "Communication Systems," 5th ed., Wiley, 2009.
- [2] A. V. Oppenheim and R. W. Schafer, "Discrete-Time Signal Processing," 3rd ed., Prentice Hall, 2009.