Extra Task 2: Multiplicative Noise and Signal Storage Optimization

1. Introduction

In this extra task, we explore the effect of multiplicative Gaussian noise on a voice signal and evaluate the reconstruction accuracy after applying sinc interpolation. Unlike additive noise, multiplicative noise alters the signal amplitude proportionally, posing different challenges for reconstruction fidelity.

2. Methodology

A recorded voice signal was multiplied pointwise with Gaussian-distributed random noise. The resulting noisy signal was then sampled and reconstructed using previously developed sinc interpolation methods. The absolute reconstruction error was also visualized to analyze the degradation.

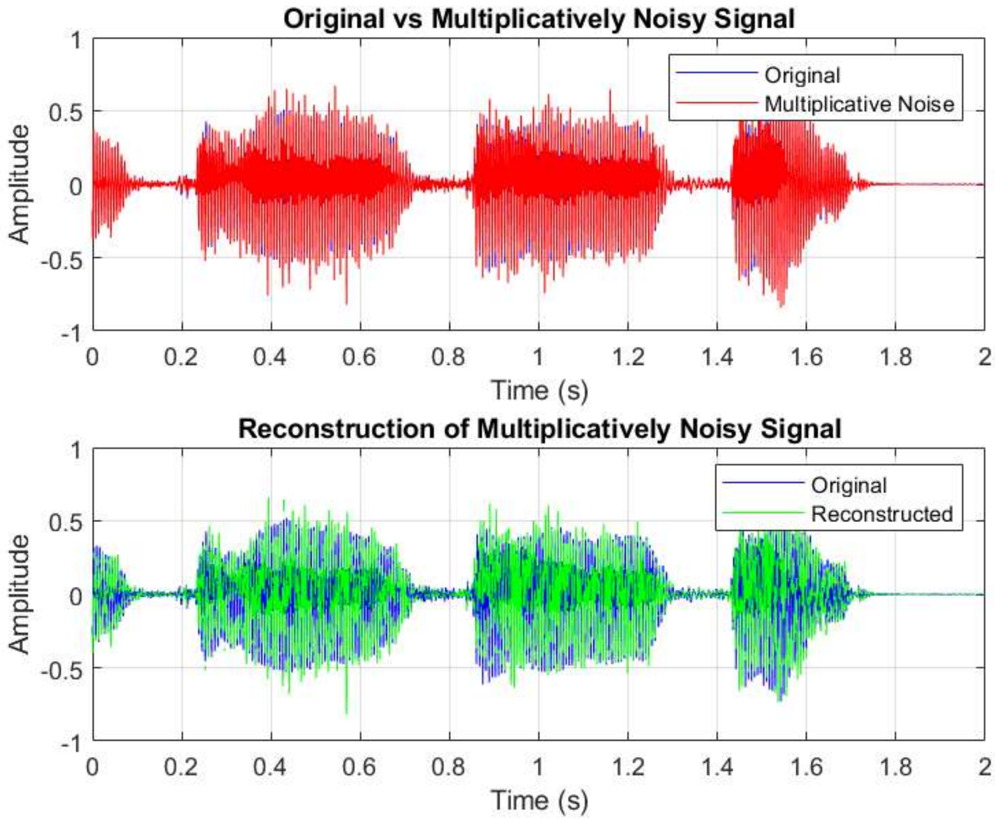
3. Results

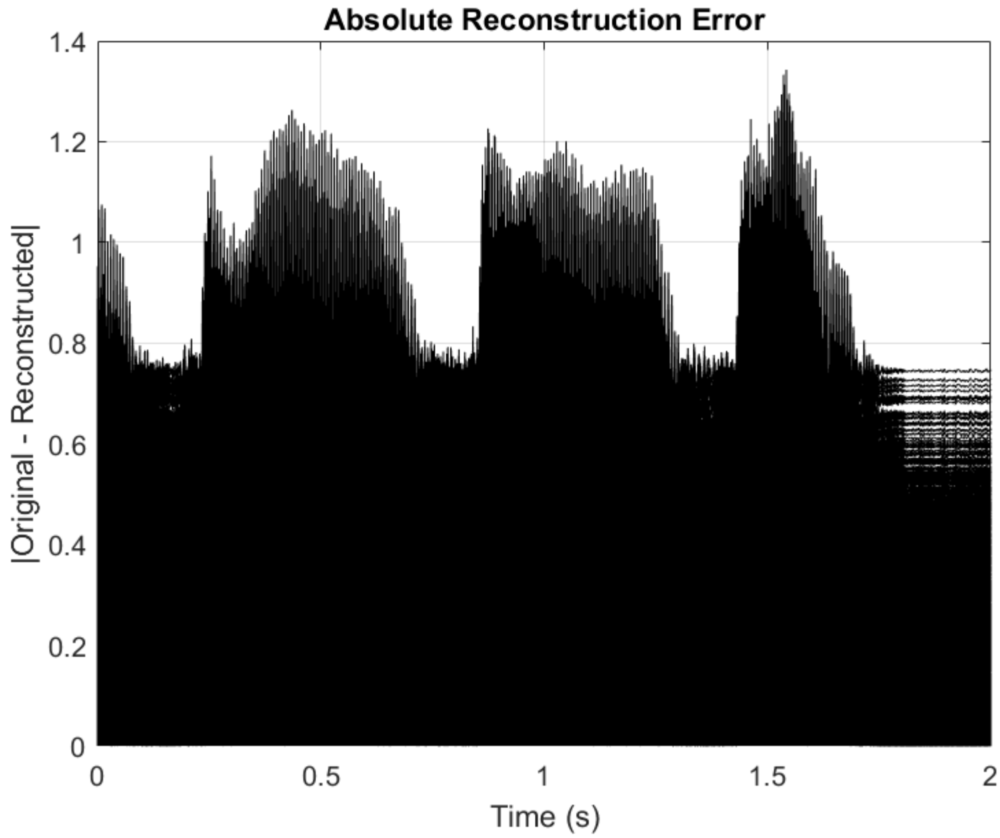
The following figures illustrate the key stages:

• Original vs Multiplicative Noise signal

• Reconstructed vs Original signal

• Absolute Reconstruction Error


Original vs Multiplicatively Noisy Signal
Original
Multiplicative Noise
Amplitude
Time (s)
1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Reconstruction of Multiplicatively Noisy Signal
Original
Reconstructed
Amplitude
Time (s)
1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2



4. Optimal Signal Storage Strategy

To store signals efficiently, one optimal solution is to apply a combination of non-uniform quantization and compression. Since real-world signals like voice tend to have clustered amplitudes (e.g., around zero), non-uniform quantization preserves more information where needed. Post-quantization, entropy coding such as Huffman or Arithmetic coding can further reduce storage size. Additionally, storing only the difference (delta encoding) or compressing in the frequency domain using DCT/FFT can yield high compression with tolerable error for many applications.

5. Conclusion

This task highlights the challenges posed by multiplicative noise in signal processing. By comparing reconstruction quality and error metrics, we gain deeper insights into noise robustness. Furthermore, optimal storage methods like non-uniform quantization and entropy compression can help maintain quality while reducing storage requirements.