EE 623 Assignment 2

Neelima Pulipati
Roll No.210108032

November 25, 2024

Objective-1: To implement and compare the speech coding performances obtained with plain LPC and voice-excited LPC vocoders for wideband speech.

Approach and Methodology

Plain LPC Vocoder

The Plain LPC Vocoder emulates the speech production system by representing the vocal tract as a time-varying filter. The input speech signal is divided into small frames, typically around 20 ms in duration, which allows the signal to be analyzed and synthesized in manageable segments. For each frame, the following steps are performed:

- **LPC Coefficients Calculation:** Autocorrelation of the frame is computed, and the Levinson-Durbin recursion is used to estimate the coefficients of the linear prediction filter. These coefficients characterize the resonant properties of the vocal tract during that time frame.
- **Voiced/Unvoiced Decision:** The signal is analyzed to determine whether the frame corresponds to a voiced or unvoiced segment. This decision is based on frame energy and the zero-crossing rate. Voiced segments, like vowels, have high energy and low zero-crossing rates, while unvoiced segments, such as fricatives, exhibit the opposite characteristics.
- **Excitation Signal Generation:** For voiced frames, an impulse train, spaced by the pitch period, is used as the excitation signal. For unvoiced frames, white noise serves as the excitation. This distinction is crucial to replicate the natural differences in voiced and unvoiced speech sounds.
- **Synthesis Filter:** The excitation signal is scaled by the gain (reflecting the signal's energy) and passed through the LPC synthesis filter. This filter models the vocal tract's transfer function, generating the synthesized speech for the current frame.

The output is synthesized with a low bit rate (about 2.4 kbps). However, the synthesized speech often has a synthetic quality due to the simplicity of the excitation model.

Voice-Excited LPC Vocoder

The Voice-Excited LPC Vocoder improves on the Plain LPC Vocoder by using a residual signal (the output of the LPC analysis filter) as the excitation source. This approach enhances the quality of the synthesized speech, particularly for voiced segments, by preserving more of the original signal's characteristics. Key steps include:

- **LPC Coefficients Calculation:** Similar to the Plain LPC Vocoder, the LPC coefficients are calculated using autocorrelation and the Levinson-Durbin recursion. These coefficients model the transfer function of the vocal tract for the current frame.

- **Residual Signal Compression:** The residual signal, obtained by filtering the input frame through the LPC analysis filter, is compressed using the Discrete Cosine Transform (DCT). This step reduces the amount of data to be transmitted while preserving the essential features of the excitation signal.
- **Synthesis at the Receiver:** At the receiver, the compressed residual signal is reconstructed, and the LPC synthesis filter is applied to generate the speech signal. This method eliminates the need for pitch period and voiced/unvoiced decisions, producing more natural-sounding speech.

The Voice-Excited LPC Vocoder offers improved speech quality compared to the Plain LPC Vocoder but at the cost of increased computational complexity and bit rate (kept below 16 kbps).

Outputs and Observations

The implemented vocoders were tested on four audio files (two male and two female voices) with a sampling rate of 16 kHz. The performance metrics, including runtime and segmental signal-to-noise ratio (SNR), were recorded.

Output Table

The results for each input file are summarized in the table below:

Table 1: Performance of Plain and Voice-Excited LPC Vocoders

File	Vocoder Type	Runtime (s)	Segmental SNR (dB)	Sampling Rate (Hz)
File 1	Plain LPC	0.54	-2.05	16000
File 1	Voice-Excited LPC	2.17	-5.53	16000
File 2	Plain LPC	0.19	-1.23	16000
File 2	Voice-Excited LPC	0.69	-1.38	16000
File 3	Plain LPC	0.57	-1.91	16000
File 3	Voice-Excited LPC	2.28	-8.09	16000
File 4	Plain LPC	0.18	-0.99	16000
File 4	Voice-Excited LPC	0.75	-3.51	16000

Analysis of Results

- **Runtime:** The Plain LPC Vocoder is computationally faster compared to the Voice-Excited LPC Vocoder, as the latter requires additional steps for DCT compression and reconstruction.
- **Segmental SNR:** The Plain LPC Vocoder achieved better segmental SNR in most cases compared to the Voice-Excited LPC Vocoder. However, the voice quality in the Plain LPC Vocoder is synthetic and lacks naturalness.
- **Trade-offs:** While the Voice-Excited LPC Vocoder introduces more computational complexity and reduces segmental SNR, it provides higher quality, natural-sounding speech by avoiding synthetic artifacts.

Conclusion

The two LPC-based vocoders were implemented successfully. While the Plain LPC Vocoder is computationally efficient and achieves a lower bit rate, it compromises on speech quality. The Voice-Excited LPC Vocoder offers improved naturalness at the cost of higher complexity and bit rate. Future work can explore optimizing the DCT compression to achieve better quality at lower bit rates.

Objective 2

To implement the Code-Excited Linear Prediction (CELP) codec for narrowband speech coding at a target bitrate.

Details of CELP Algorithm

The CELP codec incorporates the following components:

- A source-filter model for speech production.
- Use of adaptive and fixed codebooks for excitation signals.
- Optimization in the perceptually weighted domain.
- Vector quantization to achieve efficient compression.

The target bitrate was determined based on the roll number as follows:

$$\text{Target Bitrate (kbps)} = \text{Last Digit of Roll Number} \mod 5$$

For roll number 210108032, the target bitrate is 13 kbps.

Implementation

The CELP implementation extended the LPC codec by introducing:

- Fixed and adaptive codebook searches for excitation signals.
- Gain quantization and adaptive codebook updates.
- Optimized encoding of LPC coefficients and excitation parameters to achieve the target bitrate.

Code for CELP Encoding

```
# CELP Encoding Function
def celp_encode(signal, fs, frame_size=160, hop_size=80, lpc_order=10,
                num_fixed_codebook=64, gain_bits=4, lpc_bits=5, pitch_bits=7, codebook=Non
    # Frame the signal and initialize variables
    frames = frame_signal(signal, frame_size, hop_size)
    encoded = []

    # Loop through each frame for LPC and codebook encoding
    for frame in frames:
        # LPC Analysis
        A, G = lpc_analysis(frame, lpc_order)

        # Codebook search and gain quantization
        residual = compute_residual(frame, A)
        adaptive_excitation = search_adaptive_codebook(residual)
        fixed_codebook_idx, fixed_gain = search_fixed_codebook(residual, codebook)

        # Encode frame parameters
        encoded.append((fixed_codebook_idx, fixed_gain))
    return encoded
```

The full code for CELP encoding and decoding is available in the attached report.

Results for CELP Implementation

The CELP codec was tested on the same set of male and female speech files. The results are summarized below:

- **Actual Bitrate:** Consistently close to the target bitrate of 13 kbps.
- **Segmental SNR:** Reported for all samples.
- **Runtime:** Encoding and decoding completed efficiently.

Summary of Results

- **File: Male Sample 1**
  - Actual Bitrate: 13.20 kbps
  - Runtime: 8.85 seconds
  - Segmental SNR: -256.21 dB
- **File: Female Sample 1**
  - Actual Bitrate: 13.18 kbps
  - Runtime: 8.48 seconds
  - Segmental SNR: -247.74 dB