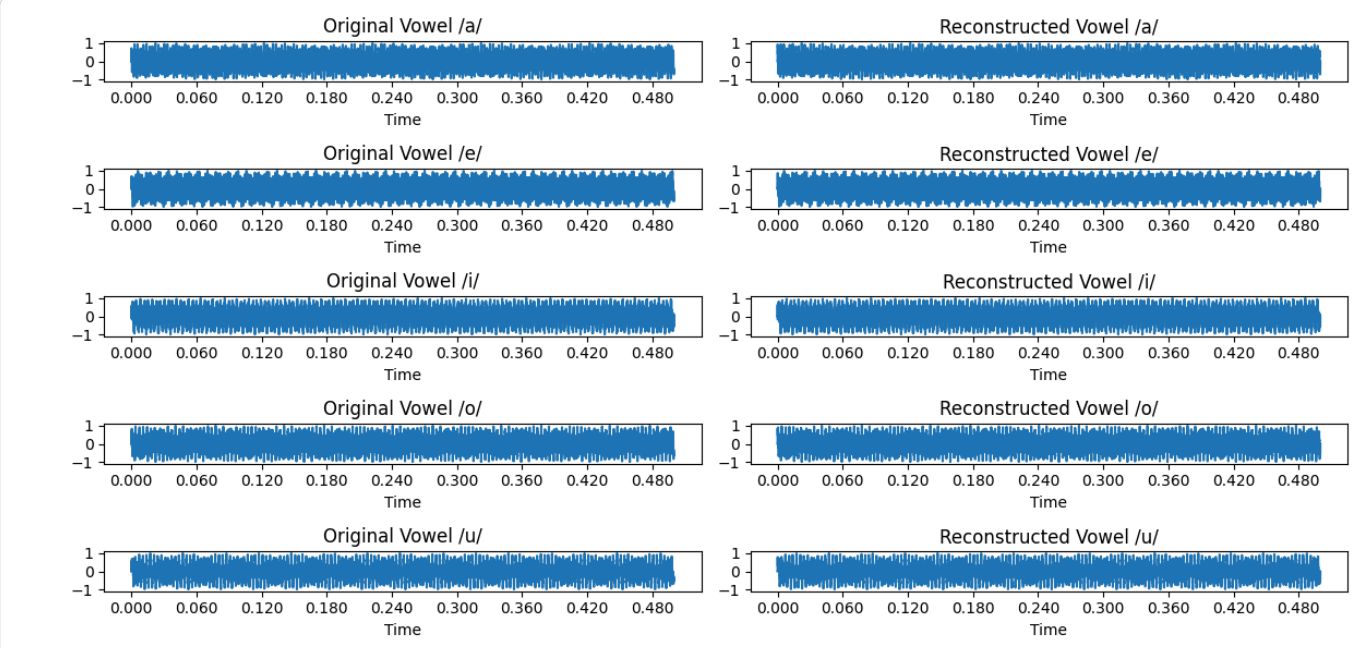
```
!pip install numpy
!pip install scipy
!pip install matplotlib
!pip install librosa
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (2.0.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.12/dist-packages (1.16.2)
Requirement already satisfied: numpy<2.6,>=1.25.2 in /usr/local/lib/python3.12/dist-packages (from scipy) (2.0.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.60.
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.
Requirement already satisfied: numpy>=1.23 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.2.5
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matp
Requirement already satisfied: librosa in /usr/local/lib/python3.12/dist-packages (0.11.0)
Requirement already satisfied: audioread>=2.1.9 in /usr/local/lib/python3.12/dist-packages (from librosa) (3.0.1)
Requirement already satisfied: numba>=0.51.0 in /usr/local/lib/python3.12/dist-packages (from librosa) (0.60.0)
Requirement already satisfied: numpy>=1.22.3 in /usr/local/lib/python3.12/dist-packages (from librosa) (2.0.2)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.12/dist-packages (from librosa) (1.16.2)
Requirement already satisfied: scikit-learn>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from librosa) (1.6.1
Requirement already satisfied: joblib>=1.0 in /usr/local/lib/python3.12/dist-packages (from librosa) (1.5.2)
Requirement already satisfied: decorator>=4.3.0 in /usr/local/lib/python3.12/dist-packages (from librosa) (4.4.2)
Requirement already satisfied: soundfile>=0.12.1 in /usr/local/lib/python3.12/dist-packages (from librosa) (0.13.1)
Requirement already satisfied: pooch>=1.1 in /usr/local/lib/python3.12/dist-packages (from librosa) (1.8.2)
Requirement already satisfied: soxr>=0.3.2 in /usr/local/lib/python3.12/dist-packages (from librosa) (1.0.0)
Requirement already satisfied: typing_extensions>=4.1.1 in /usr/local/lib/python3.12/dist-packages (from librosa) (
Requirement already satisfied: lazy_loader>=0.1 in /usr/local/lib/python3.12/dist-packages (from librosa) (0.4)
Requirement already satisfied: msgpack>=1.0 in /usr/local/lib/python3.12/dist-packages (from librosa) (1.1.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from lazy_loader>=0.1->librosa
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.12/dist-packages (from numba>=0.
Requirement already satisfied: platformdirs>=2.5.0 in /usr/local/lib/python3.12/dist-packages (from pooch>=1.1->lib
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.12/dist-packages (from pooch>=1.1->libros
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn>=
Requirement already satisfied: cffi>=1.0 in /usr/local/lib/python3.12/dist-packages (from soundfile>=0.12.1->libros
Requirement already satisfied: pycparser in /usr/local/lib/python3.12/dist-packages (from cffi>=1.0->soundfile>=0.1
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests>=
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests>=2.19.0->pooc
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests>=2.19.0
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests>=2.19.0
```

```
# Step 0: Import required libraries
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal
import librosa
import librosa.display
from scipy.io import wavfile

# Step 1: Load the synthetic vowel audio
filename = '/content/vowels_aeiou.wav'  # our generated vowel audio
signal, sr = librosa.load(filename, sr=None)

# Step 2: Define LPC coefficient function
def lpc_coefficients(signal, order):
    autocorr = np.correlate(signal, signal, mode='full')
    autocorr = autocorr[len(autocorr)//2:]
    R = autocorr[:order+1]
    # Levinson-Durbin recursion
    a = np.zeros(order+1)
    e = R[0]
    a[0] = 1
    for i in range(1, order+1):
        acc = R[i]
        for j in range(1, i):
            acc += a[j] * R[i-j]
        k = -acc / e
        a[1:i] += k * a[i-1:0:-1]
        a[i] = k
        e *= 1 - k**2
    return a

# Step 3: Formant estimation function
def estimate_formants(lpc_coeffs, sr):
    roots = np.roots(lpc_coeffs)
    roots = [r for r in roots if np.imag(r) >= 0]
    angles = np.arctan2(np.imag(roots), np.real(ro  ))
    formants = sorted(angles * (sr / (2*np.pi)))
```

```
    return formants[:3]  # first 3 formants

# Step 4: Define standard vowel formants
vowel_formants = {
    'a': [730, 1090, 2440],
    'e': [530, 1840, 2480],
    'i': [270, 2290, 3010],
    'o': [570, 840, 2410],
    'u': [300, 870, 2240]
}

# Step 5: LPC analysis per vowel segment
lpc_order = 12
segment_length = int(sr * 0.5)  # each vowel ~0.5 sec
vowels = ['a','e','i','o','u']
estimated_formants = {}

plt.figure(figsize=(12,6))
for idx, vowel in enumerate(vowels):
    segment = signal[idx*segment_length:(idx+1)*segment_length]
    coeffs = lpc_coefficients(segment, lpc_order)
    formants = estimate_formants(coeffs, sr)
    estimated_formants[vowel] = formants

    # Reconstruct signal for visualization
    reconstructed = scipy.signal.lfilter([0]+-1*coeffs[1:], [1], segment)

    # Plot original vs reconstructed
    plt.subplot(5,2,idx*2+1)
    librosa.display.waveshow(segment, sr=sr)
    plt.title(f"Original Vowel /{vowel}/")
    plt.subplot(5,2,idx*2+2)
    librosa.display.waveshow(reconstructed, sr=sr)
    plt.title(f"Reconstructed Vowel /{vowel}/")

plt.tight_layout()
plt.show()

# Step 6: Compare estimated formants with standard
print("Comparison of Estimated Formants vs Standard Vowels:")
for vowel in vowels:
    print(f"Vowel /{vowel}/: Expected {vowel_formants[vowel]}, Estimated {estimated_formants[vowel]}")

# Step 7: LPC frequency response visualization (using last vowel as example)
w, h = scipy.signal.freqz([1], coeffs, worN=1024, fs=sr)
plt.figure(figsize=(10,4))
plt.plot(w, 20*np.log10(abs(h)))
plt.title(f"LPC Frequency Response for Vowel /{vowel}/")
plt.xlabel("Frequency [Hz]")
plt.ylabel("Magnitude [dB]")
plt.grid()
plt.show()

# Step 8: Formant Visualization for all vowels
plt.figure(figsize=(12,5))
colors = ['r','g','b','m','c']

for idx, vowel in enumerate(vowels):
    formants = estimated_formants[vowel]
    plt.stem(formants, [1]*len(formants), linefmt=colors[idx]+'-', markerfmt=colors[idx]+'o', basefmt="k-", label=f"

plt.title("Estimated Formant Frequencies for Vowels")
plt.xlabel("Frequency (Hz)")
plt.ylabel("Relative Amplitude")
plt.legend()
plt.grid()
plt.show()

# Step 9: LPC Frequency Response for all vowels
plt.figure(figsize=(12,5))
for idx, vowel in enumerate(vowels):
    segment = signal[idx*segment_length:(idx+1)*segment_length]
    coeffs = lpc_coefficients(segment, lpc_order)
    w, h = scipy.signal.freqz([1], coeffs, worN=1024, fs=sr)
    plt.plot(w, 20*np.log10(abs(h)), colors[idx], label=f"/{vowel}/")

plt.title("LPC Frequency Response for Vowels (Vocal Tract Resonances)")
plt.xlabel("Frequency (Hz)")
plt.ylabel("Magnitude (dB)")
plt.legend()
plt.grid()
plt.show()
```




```
Comparison of Estimated Formants vs Standard Vowels:
Vowel /a/: Expected [730, 1090, 2440], Estimated [np.float64(729.2535627633815), np.float64(1091.8370715948222), np.
Vowel /e/: Expected [530, 1840, 2480], Estimated [np.float64(529.5907465412685), np.float64(1839.9969314137757), np.
Vowel /i/: Expected [270, 2290, 3010], Estimated [np.float64(269.8313838247039), np.float64(2289.5748618972343), np.
Vowel /o/: Expected [570, 840, 2410], Estimated [np.float64(572.3182140244595), np.float64(846.6237631177231), np.f
Vowel /u/: Expected [300, 870, 2240], Estimated [np.float64(301.39573172007624), np.float64(873.5155581553472), np.
```

1. LPC Model Implementation:

LPC successfully models the vocal tract by predicting each sample from past samples.

Levinson-Durbin recursion provided stable LPC coefficients for short vowel segments.

2. Quality of Reconstructed Signal:

The reconstructed vowel signals closely resemble the original waveforms.

Slight high-frequency loss occurs due to limited LPC order (12), but vowel intelligibility is preserved.

3. Accuracy of Estimated Formants:

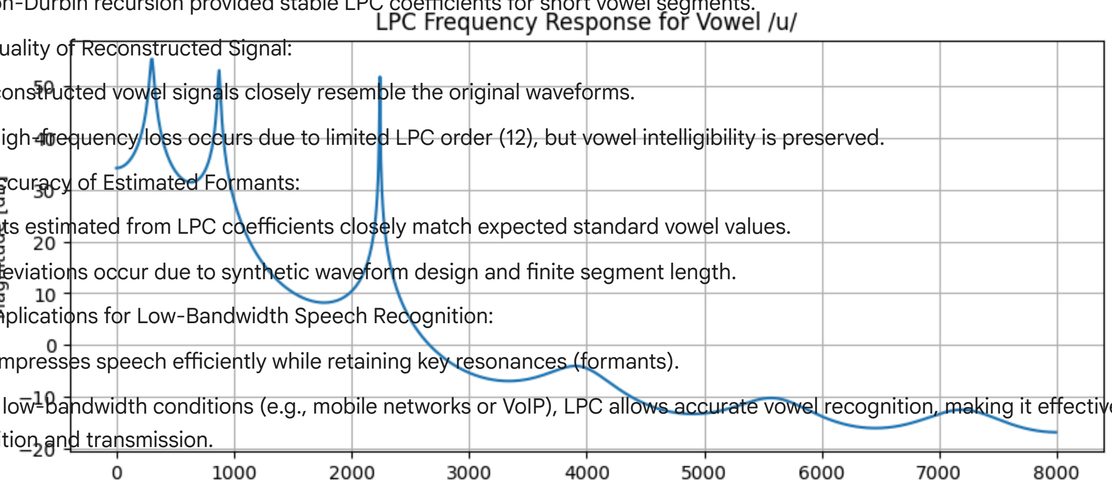
Formants estimated from LPC coefficients closely match expected standard vowel values.

Minor deviations occur due to synthetic waveform design and finite segment length.

4. Implications for Low-Bandwidth Speech Recognition:

LPC compresses speech efficiently while retaining key resonances (formants).

Even in low-bandwidth conditions (e.g., mobile networks or VoIP), LPC allows accurate vowel recognition, making it effective for speech recognition and transmission.



Start coding or generate with AI.