

```

In [13]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

# 1. Load the CSV file
data = pd.read_csv('signal_122140055.csv')
time = data['time'].values
amplitude = data['amplitude'].values

# Calculate sampling rate
dt = time[1] - time[0] # Time step (assuming uniform sampling)
fs = 1 / dt # Sampling frequency in Hz

# 2. Perform FFT
N = len(amplitude)
yf = np.fft.fft(amplitude)
xf = np.fft.fftfreq(N, dt)[:N//2] # Positive frequencies only
amplitude_spectrum = 2.0/N * np.abs(yf[:N//2]) # Magnitude of FFT

# Find the dominant frequency (excluding DC component)
idx_max = np.argmax(amplitude_spectrum[1:]) + 1 # Skip DC (index 0)
dominant_freq = xf[idx_max]
print(f"Dominant frequency: {dominant_freq:.2f} Hz")

# 3. Design and apply bandpass filter
bandwidth = 0.1 * dominant_freq
lowcut = dominant_freq - bandwidth
highcut = dominant_freq + bandwidth

# Ensure frequencies are within Nyquist Limit
nyquist = fs / 2
lowcut = max(lowcut, 0.1) # Avoid zero or negative frequencies
highcut = min(highcut, nyquist * 0.99) # Stay below Nyquist

# Design Butterworth bandpass filter
order = 4
b, a = signal.butter(order, [lowcut, highcut], fs=fs, btype='band')
filtered_signal = signal.lfilter(b, a, amplitude)

# 4. Plot the results
plt.figure(figsize=(15, 10))

# Plot original signal
plt.subplot(4, 1, 1)
plt.plot(time, amplitude, label='Original Signal')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('Original Time-Domain Signal')
plt.grid(True)
plt.legend()

# Plot FFT spectrum (full range)
plt.subplot(4, 1, 2)
plt.plot(xf, amplitude_spectrum, label='FFT Spectrum')
plt.axvline(dominant_freq, color='r', linestyle='--', label=f'Dominant Freq = {dominant_freq:.2f} Hz')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Amplitude')

```

```

plt.title('FFT Spectrum')
plt.grid(True)
plt.legend()
plt.xlim(0, fs/2) # Limit x-axis to Nyquist frequency
# Plot FFT spectrum (zoomed to 31 Hz to 33 Hz)
plt.subplot(4, 1, 3)
plt.plot(xf, amplitude_spectrum, label='FFT Spectrum')
plt.axvline(dominant_freq, color='r', linestyle='--', label=f'Dominant Freq = {d')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Amplitude')
plt.title('FFT Spectrum (Zoomed 31 Hz to 33 Hz)')
plt.grid(True)
plt.legend()
plt.xlim(31, 33) # Zoom to 31 Hz to 33 Hz
# Adjust y-axis to focus on the relevant amplitude range in this frequency band
idx_range = (xf >= 31) & (xf <= 33)
plt.ylim(0, max(amplitude_spectrum[idx_range]) * 1.1) # Add 10% margin above ma

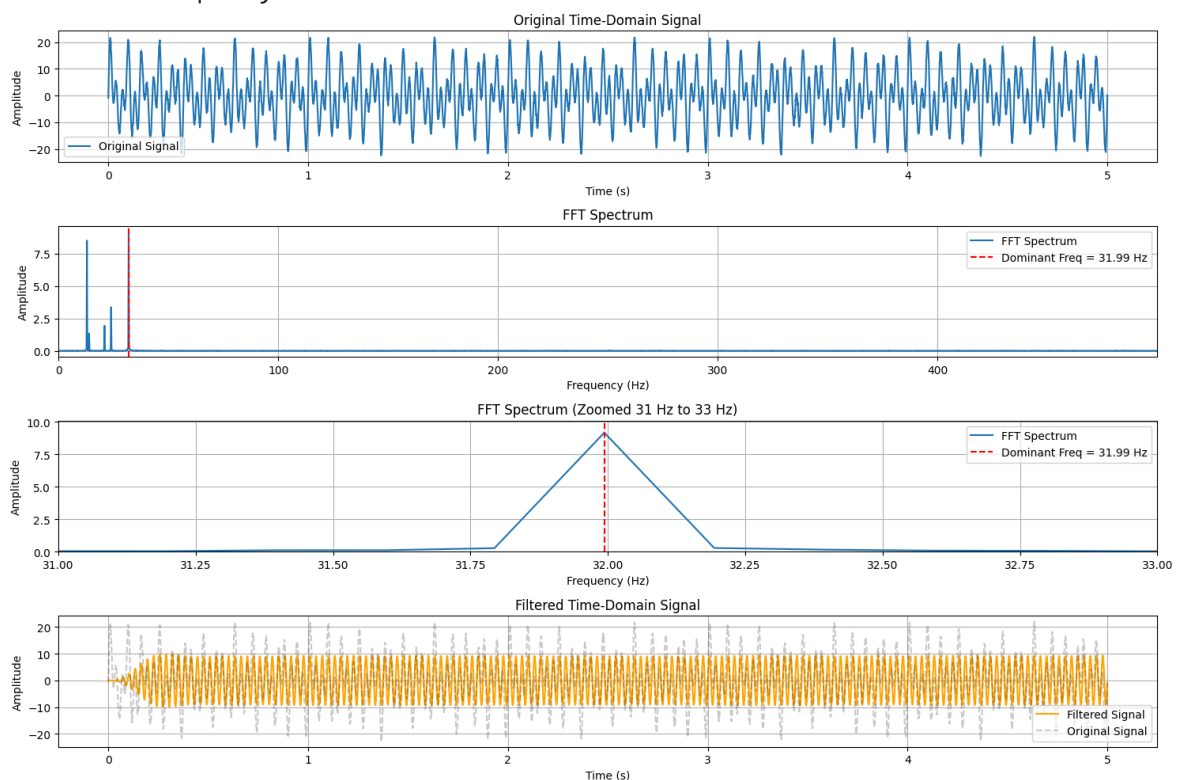
# Plot filtered signal
plt.subplot(4, 1, 4)
plt.plot(time, filtered_signal, label='Filtered Signal', color='orange')
plt.plot(time, amplitude, 'k--', alpha=0.2, label='Original Signal')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('Filtered Time-Domain Signal')
plt.grid(True)
plt.legend()

# Adjust spacing between subplots
plt.subplots_adjust(hspace=0.5)

plt.tight_layout()
plt.show()

```

Dominant frequency: 31.99 Hz



Analisis dan *Filtering* Sinyal Menggunakan *FFT*

Gambaran Umum Proses

Proses ini melibatkan analisis sinyal dalam domain waktu untuk diekstrak menjadi representasi domain frekuensi menggunakan *Fast Fourier Transform (FFT)*. Komponen frekuensi dominan diidentifikasi dan disaring menggunakan *bandpass filter*. Visualisasi yang disediakan meliputi:

1. **Sinyal Asli dalam Domain Waktu:** Data amplitudo mentah terhadap waktu.
2. **Spektrum *FFT* (Rentang Penuh):** Hasil dekomposisi spektral yang menunjukkan kandungan frekuensi dari sinyal.
3. **Spektrum *FFT* (Zoom 31 Hz hingga 33 Hz):** Fokus pada area frekuensi di sekitar komponen dominan.
4. **Sinyal yang Telah Difilter:** Rekonstruksi sinyal setelah penyaringan, mempertahankan hanya komponen dominan.

Analogi

Analogi yang digunakan adalah seperti menghadiri sebuah pesta di mana banyak orang berbicara secara bersamaan. Tujuannya adalah untuk memfokuskan pendengaran pada suara terkuat di tengah kebisingan:

- **Sinyal Asli**
Seperti campuran percakapan yang kacau, sinyal domain waktu menangkap tumpang tindih berbagai frekuensi dalam satu waktu. Mewakili hasil superposisi seluruh percakapan yang terjadi secara bersamaan. Fluktuasi amplitudo terhadap waktu mencerminkan campuran berbagai komponen frekuensi.
- **Spektrum *FFT*:**
Transformasi ke domain frekuensi melalui *FFT* memisahkan sinyal menjadi frekuensi-frekuensi komponennya. Spektrum amplitudo menunjukkan puncak signifikan di 31,99 Hz, yang mengindikasikan keberadaan komponen dominan di antara puncak-puncak kecil lainnya akibat kebisingan latar.
- **Spektrum *FFT* (Zoom 31–33 Hz):**
Memfokuskan pengamatan pada frekuensi sempit (31–33 Hz) memungkinkan identifikasi yang lebih akurat terhadap komponen utama. Aktivitas frekuensi di sekitar area tersebut sangat minimal, memperkuat keyakinan bahwa inilah frekuensi yang perlu disaring.
- **Sinyal yang Telah Difilter:**
Aplikasi *bandpass filter* berperan seperti headphone dengan fitur *Active Noise Cancelling (ANC)*, hanya membiarkan frekuensi target melewati, sehingga menghasilkan sinyal yang lebih bersih dan fokus.

Lampiran

- Percakapan dengan LLM [GROK AI](#)
- CSV File (Gunakan Akun Email ITERA) [CSV File](#)