

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.io import loadmat
from scipy.signal import butter, filtfilt

# --- 1. Load MATLAB data ---
mat = loadmat("earthquake_in_3.mat") # Update path as necessary
signal = mat["earthquake_in3"]
Fs = 400 # Sampling frequency (Hz)
nyquist_freq = Fs / 2

# --- 2. Bandpass Filter Function ---
def apply_bandpass_filter(x, lowcut, highcut, Fs, order):
    nyq = 0.5 * Fs
    wn = [lowcut / nyq, highcut / nyq]
    b, a = butter(order, wn, btype='band')
    xf = filtfilt(b, a, x, axis=0)
    return xf

# --- 3. Filter Z component between 40–42 Hz ---
z_channel = signal[:, 0]
z_filtered = apply_bandpass_filter(z_channel, lowcut=40, highcut=42, Fs=Fs, order=4)

# --- 4. Time axis ---
n_samples = len(z_channel)
time = np.arange(n_samples) / Fs

# --- 5. Plot Original vs Filtered Z ---
plt.figure(figsize=(12, 6))
plt.plot(time, z_channel, label="Original Z", alpha=0.5)
plt.plot(time, z_filtered, label="Filtered Z (40–42 Hz)", linewidth=1)
plt.title("Z Component - Time Domain")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# --- 6. Manual Fourier Analysis Function ---
def furi(x, Fs):
    N = len(x)
    ffx = np.fft.fft(x)
    An = -1/(2*N) * np.imag(ffx[:N//2])
    Bn = 1/(2*N) * np.real(ffx[:N//2])
    phase = np.arctan2(Bn, An)
    Cn = np.sqrt(An**2 + Bn**2)
    freq = np.linspace(0, Fs/2, N//2, endpoint=False)
    p = (2 * np.abs(ffx[:N//2]) / N)**2
    return freq, Cn, p, phase

# --- 7. Compute Power Spectra ---
freq_orig, _, p_orig, _ = furi(z_channel, Fs)
freq_filt, _, p_filt, _ = furi(z_filtered, Fs)

# --- 8. Plot Power Spectrum ---
plt.figure(figsize=(12, 6))
plt.plot(freq_orig, p_orig, label="Original Z")
plt.plot(freq_filt, p_filt, label="Filtered Z (40–42 Hz)")
plt.title("Z Component - Power Spectrum")
plt.xlabel("Frequency (Hz)")
plt.ylabel("Power")
plt.xlim(0, 100)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# --- 9. Output Nyquist frequency ---
print("Nyquist frequency:", nyquist_freq, "Hz")

```