	Marwari University Faculty of Technology Department of Information and Communication Technology	
Subject: Digital Signal and Image Processing(01CT0513)	**Aim:** Design Butterworth and Chebyshev filter using the bilinear transformation method.	
Experiment No: 04	**Date:**	**Enrollment No: 92200133030**

Aim: Design Butterworth and Chebyshev filters using the bilinear transformation method.

Theory:-

- The bilinear transformation method is commonly used to design analog filters and then convert them into digital filters. This method maps the analog frequency response to the digital frequency response using a bilinear transformation.
- The Butterworth and Chebyshev filters are two commonly used filter types. The Butterworth filter has a maximally flat frequency response in the passband, while the Chebyshev filter allows for a sharper transition between the passband and the stopband at the expense of ripples in either the passband or stopband.
- The steps involved in designing Butterworth and Chebyshev filters using the bilinear transformation method are as follows:
- Specify the desired filter specifications, such as the filter order, cutoff frequency, and filter type (Butterworth or Chebyshev).
- Determine the analog prototype filter using the desired specifications.
- Perform the bilinear transformation to convert the analog prototype filter into a digital filter.
- Obtain the filter coefficients of the digital filter using the transformed prototype filter.
- Plot the filter's magnitude response and impulse response.
- Save the filter coefficients (optional).
- Flowchart:
- The flowchart for the program will consist of the following steps:
- Specify the desired filter specifications, such as the filter order, cutoff frequency, and filter type.
- Design the analog prototype filter using the scipy.signal.butter or scipy.signal.cheby1 function.
- Perform the bilinear transformation using the scipy.signal.bilinear function to convert the analog filter to a digital filter.
- Obtain the filter coefficients of the digital filter.
- Plot the filter's magnitude response and impulse response.
- Save the filter coefficients (optional).
- Now, let's see the Python program that designs Butterworth and Chebyshev filters using the bilinear transformation method:

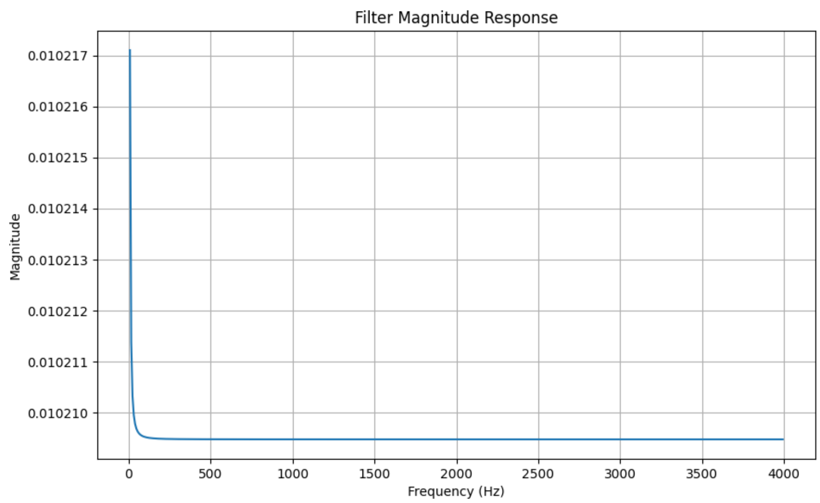
Programm:-

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.signal import bilinear, butter, cheby1, freqz, lfilter

def design_butterworth_filter(filter_order, cutoff_frequency, sampling_frequency):
    # Normalize the cutoff frequency (cutoff_frequency / Nyquist frequency)
    nyquist_frequency = sampling_frequency / 2
    normalized_cutoff = cutoff_frequency / nyquist_frequency

    # Design the analog Butterworth filter
    analog_b, analog_a = butter(
        filter_order, normalized_cutoff, analog=False, btype="low"
    )

    # Perform the bilinear transformation
    digital_b, digital_a = bilinear(analog_b, analog_a, sampling_frequency)

    return digital_b, digital_a

def design_chebyshev_filter(filter_order, cutoff_frequency, sampling_frequency, ripple):
    # Normalize the cutoff frequency
    nyquist_frequency = sampling_frequency / 2
    normalized_cutoff = cutoff_frequency / nyquist_frequency

    # Design the analog Chebyshev filter
    analog_b, analog_a = cheby1(
        filter_order, ripple, normalized_cutoff, analog=False, btype="low"
    )

    # Perform the bilinear transformation
    digital_b, digital_a = bilinear(analog_b, analog_a, sampling_frequency)

    return digital_b, digital_a

def plot_filter_response(digital_b, digital_a, sampling_frequency):
    # Compute the frequency response of the filter
    frequency, magnitude_response = freqz(digital_b, digital_a, fs=sampling_frequency)

    # Plot the magnitude response
    plt.figure(figsize=(10, 6))
    plt.plot(frequency, np.abs(magnitude_response))
    plt.title("Filter Magnitude Response")
    plt.xlabel("Frequency (Hz)")
    plt.ylabel("Magnitude")
    plt.grid(True)
    plt.show()

    # Compute and plot the impulse response
    impulse = np.zeros(100)
    impulse[0] = 1  # Create a unit impulse
```

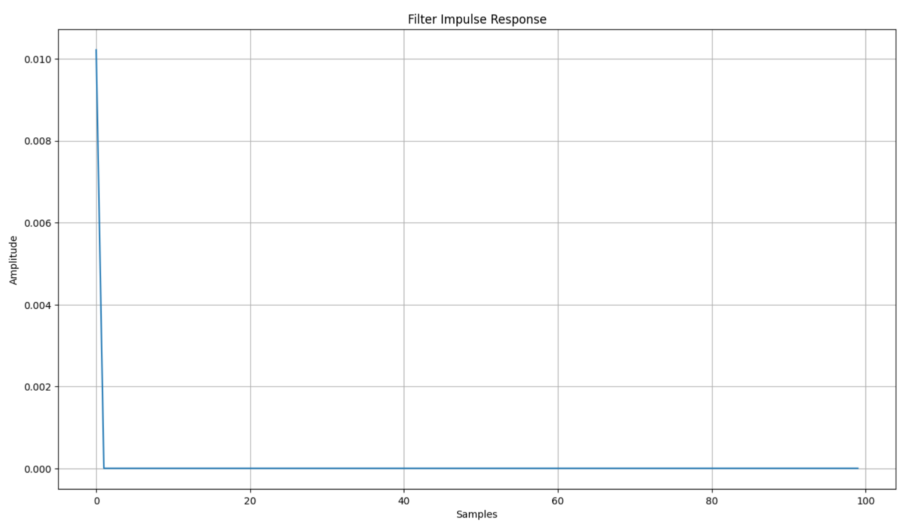
```
    impulse_response = lfilter(digital_b, digital_a, impulse)

    plt.figure(figsize=(10, 6))
    plt.plot(impulse_response)
    plt.title("Filter Impulse Response")
    plt.xlabel("Samples")
    plt.ylabel("Amplitude")
    plt.grid(True)
    plt.show()

# Specify the desired filter specifications
filter_order = 4  # Filter order
cutoff_frequency = 1000  # Cutoff frequency in Hz
sampling_frequency = 8000  # Sampling frequency in Hz
ripple = 0.5  # Ripple factor for Chebyshev filter

# Design the Butterworth filter
digital_b, digital_a = design_butterworth_filter(
    filter_order, cutoff_frequency, sampling_frequency
)
# Plot the Butterworth filter's magnitude response and impulse response
plot_filter_response(digital_b, digital_a, sampling_frequency)

# Design the Chebyshev filter
digital_b, digital_a = design_chebyshev_filter(
    filter_order, cutoff_frequency, sampling_frequency, ripple
)
# Plot the Chebyshev filter's magnitude response and impulse response
plot_filter_response(digital_b, digital_a, sampling_frequency)

# Save the filter coefficients (optional)
filter_path = "filter_coefficients.txt"
np.savetxt(filter_path, np.vstack((digital_b, digital_a)), delimiter=",")
print(f"Filter coefficients saved at: {filter_path}")
```

Output :-

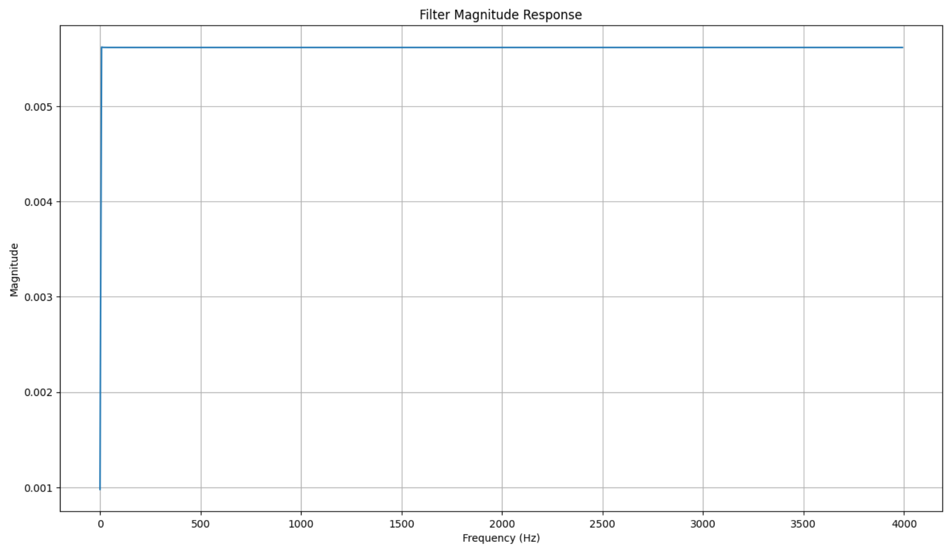
Butterworth Filter Magnitude Response :-



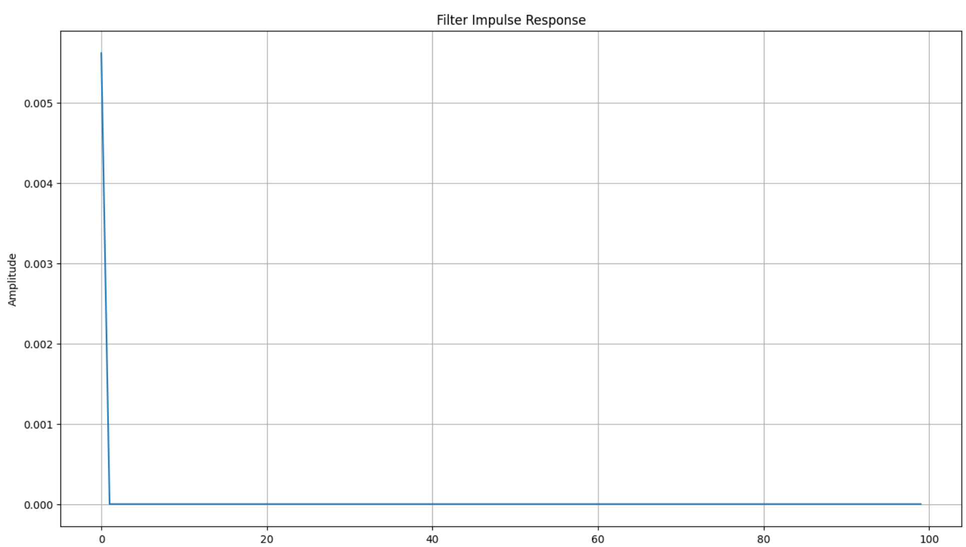
Butterworth Filter Impulse Response :-



Chebyshev Filter Magnitude Response :-



Chebyshev Filter Impulse Response :-



Marwadi University	**Marwari University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Signal and Image Processing(01CT0513)	**Aim:** Design Butterworth and Chebyshev filter using the bilinear transformation method.	
Experiment No: 04	**Date:**	**Enrollment No: 92200133030**

Filter Coefficients :-

5.621724098249910630e-03,-2.248408570661841743e-02,3.372191305732654548e-02,-2.247846538775357186e-02,5.618913938795535787e-03
1.000000000000000000e+00,-4.000320181991981805e+00,6.000960591637167774e+00,-4.000960637301628431e+00,1.000320227656443350e+00