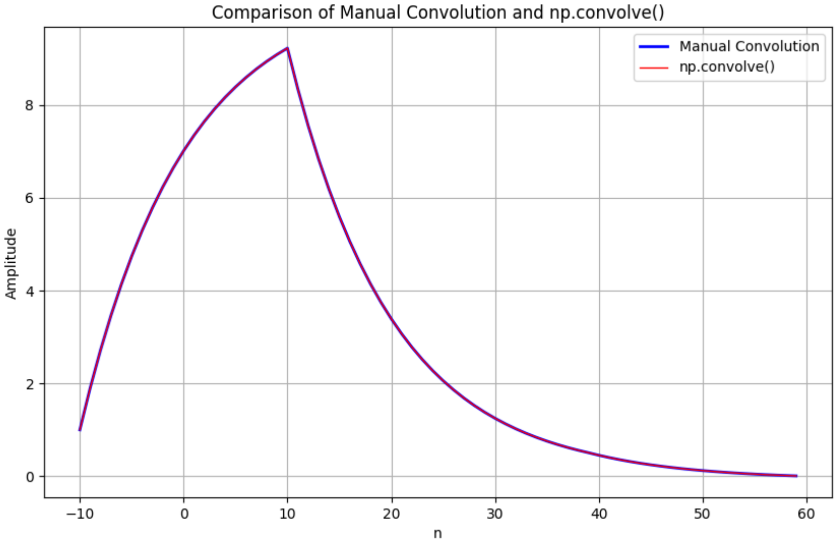
S&S OA 1

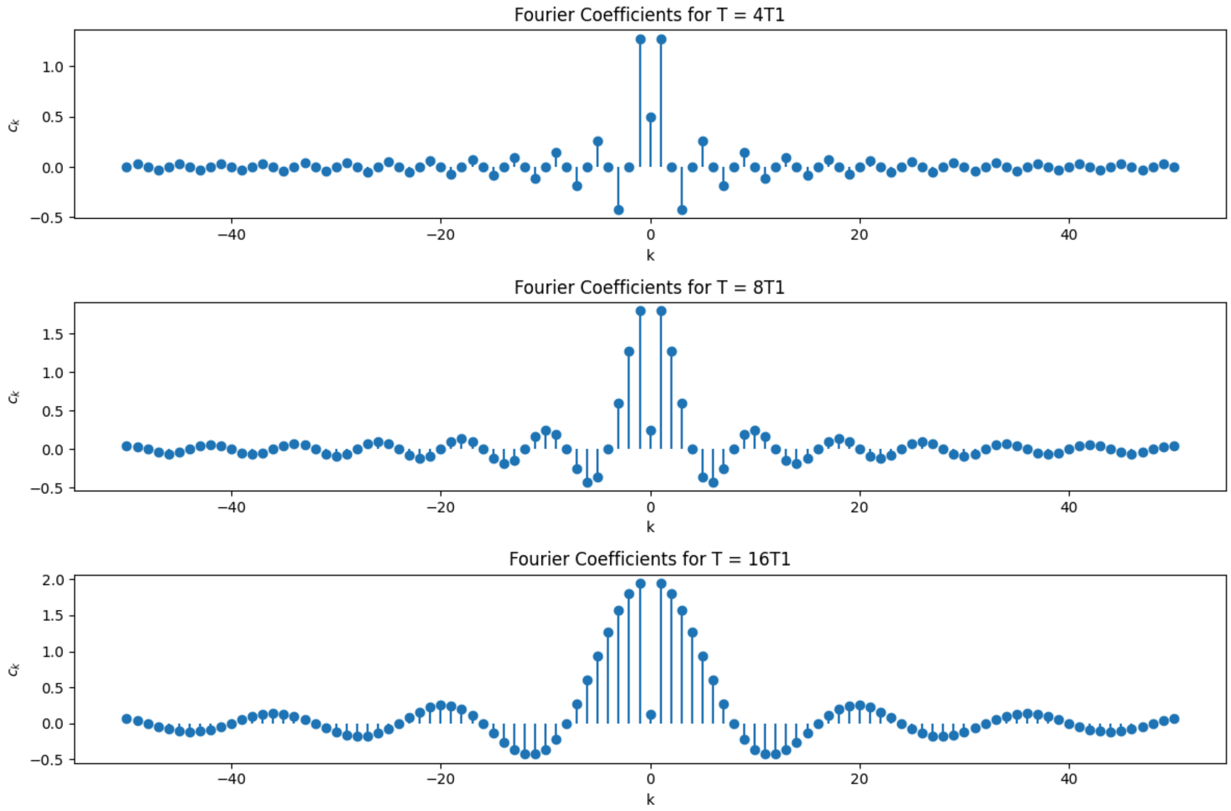
Vikranth Udandarao

2022570

1. To compare the manual convolution with the result from `np.convolve()`, both were computed and plotted on the same graph. The manual convolution was performed using nested loops to apply the convolution sum directly, while `np.convolve()` used NumPy's built-in convolution function, which is optimized for performance. In the plot, we differentiate the two methods by using distinct colors and line widths: the manual convolution is shown with a thicker blue line, and the `np.convolve()` result is depicted with a thinner red line. Both methods yield nearly identical results, which visually overlap on the plot, indicating that the manual approach correctly implements the convolution process.



2.



3. .

Prominent Frequency Components:

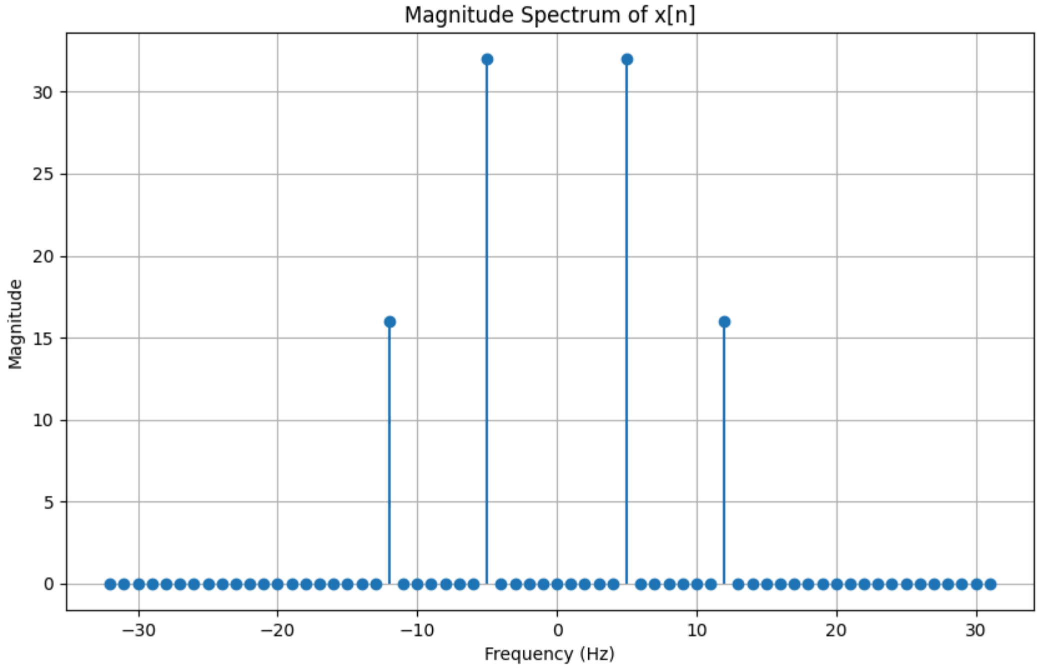
- The FFT magnitude spectrum clearly shows two primary peaks:
  - One at **5 Hz**.
  - The second at **12 Hz**.
- These frequencies correspond exactly to the frequencies f1=5 Hz and f2=12 Hz that were used to construct the signal in the time domain.

Amplitudes of the Peaks:

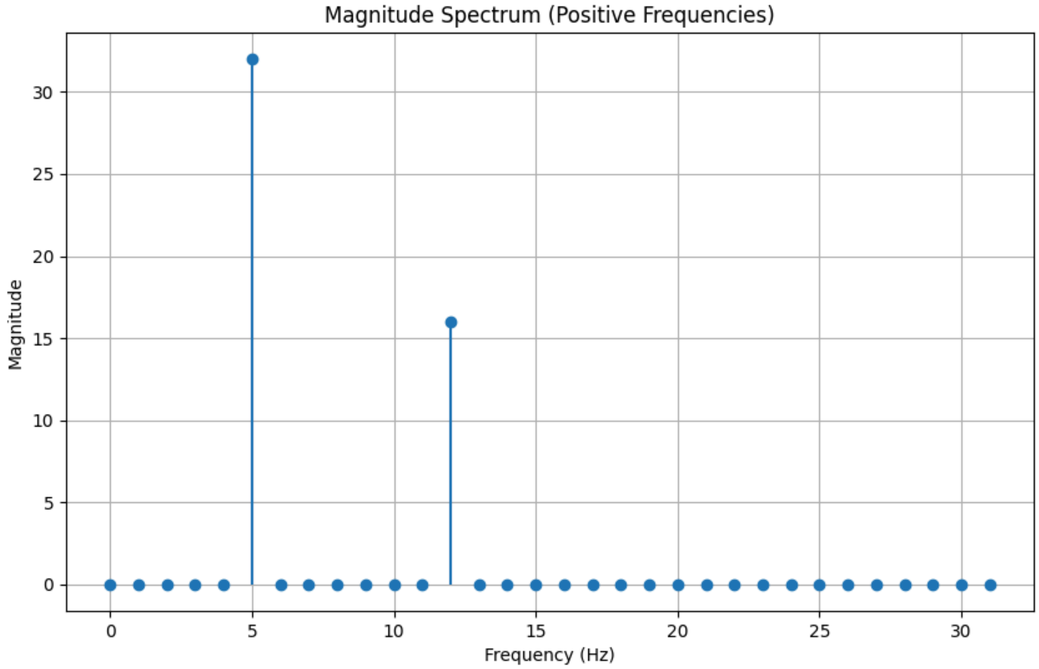
- The peak at **5 Hz** has a magnitude of approximately **32**. This is consistent with the fact that the cosine term has an amplitude of **1**. In the FFT, the magnitude of this peak is scaled by the number of samples NNN, which results in a larger peak.
- The peak at **12 Hz** has a magnitude of approximately **16**. This corresponds to the second cosine term where the amplitude is **0.5**. The FFT shows a peak half as large as the one at 5 Hz, which matches the expected amplitude scaling.

Symmetry of the Spectrum:

- The full FFT result (first plot) is symmetric around 0 Hz, with peaks appearing at both positive and negative frequencies. This is a direct consequence of the fact that the signal x[n]x[n]x[n] is real-valued, leading to conjugate symmetry in the frequency domain.
- The second plot focuses on the **positive frequencies**, showing only the significant frequency components at **5 Hz** and **12 Hz**.

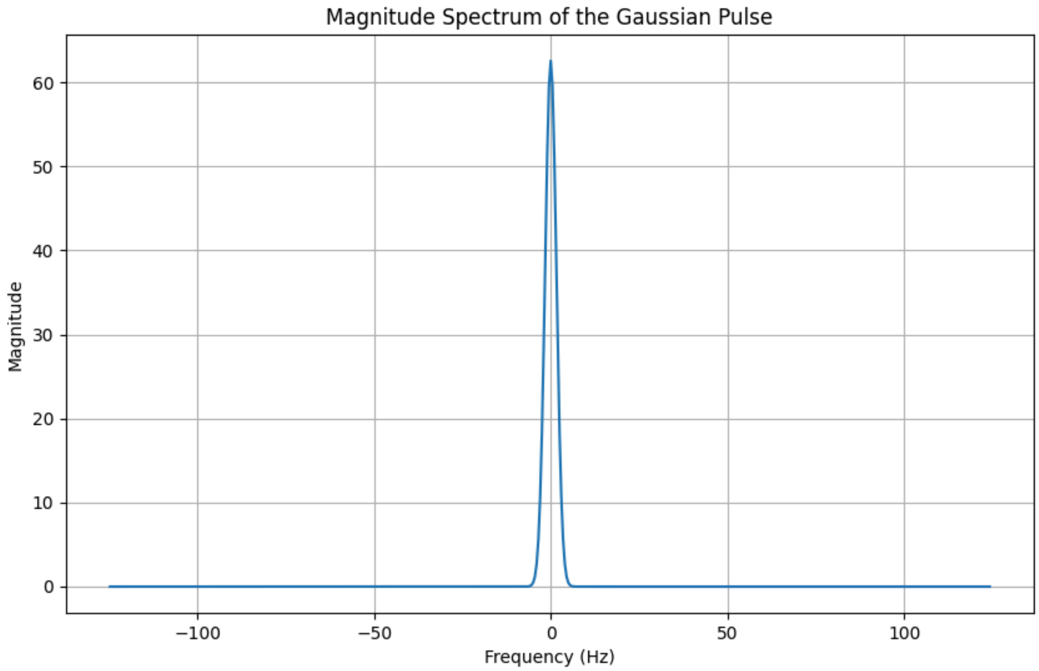


Magnitude Spectrum of x[n]



Magnitude Spectrum (Positive Frequencies)

4. …



The Gaussian pulse x(t) is a smoothly varying signal in the time domain. Its magnitude spectrum, obtained through the Fast Fourier Transform (FFT), reveals several important characteristics about the frequency content of the pulse.

Gaussian Shape in the Frequency Domain:

a. The FFT of a Gaussian pulse results in another Gaussian-shaped magnitude spectrum in the frequency domain. This property arises because the Fourier Transform of a Gaussian function is also a Gaussian. The spectrum is centered at **0 Hz**, reflecting that the pulse is a low-frequency signal with most of its energy concentrated around zero frequency.

Dominance of Low-Frequency Components:

b. The bell-shaped spectrum shows that the energy of the Gaussian pulse is primarily concentrated at **low frequencies**. The amplitude decreases rapidly as we move away from 0 Hz. This indicates that the Gaussian pulse is composed mainly of low-frequency components, which corresponds to its smooth and continuous nature in the time domain.

Width of the Frequency Spectrum:

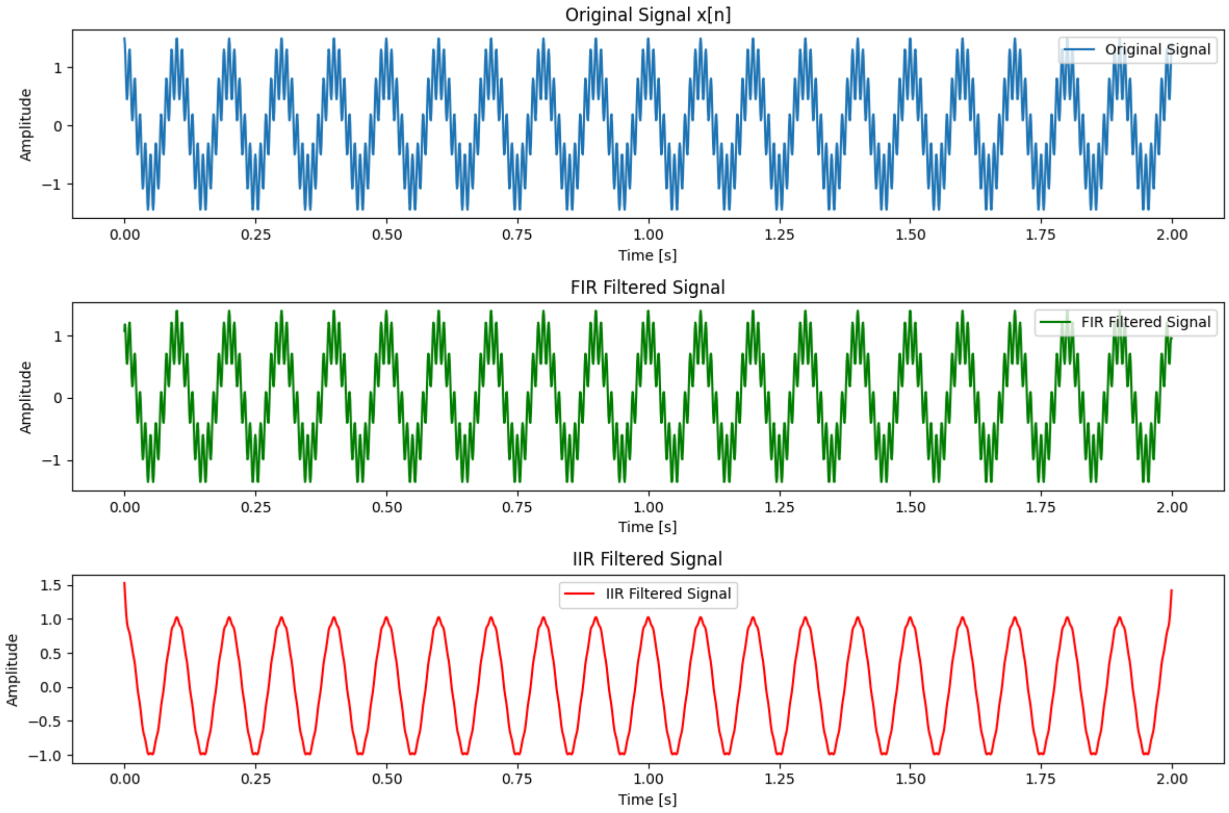
c. The width of the magnitude spectrum depends on the parameter σ\sigmaσ, which controls the width of the Gaussian pulse in the time domain. In this case, with σ=0.1\sigma = 0.1σ=0.1, the spectrum is relatively **wide** in the frequency domain. According to the time-frequency uncertainty principle, a narrower pulse in the time domain (small σ\sigmaσ) results in a broader frequency spectrum.

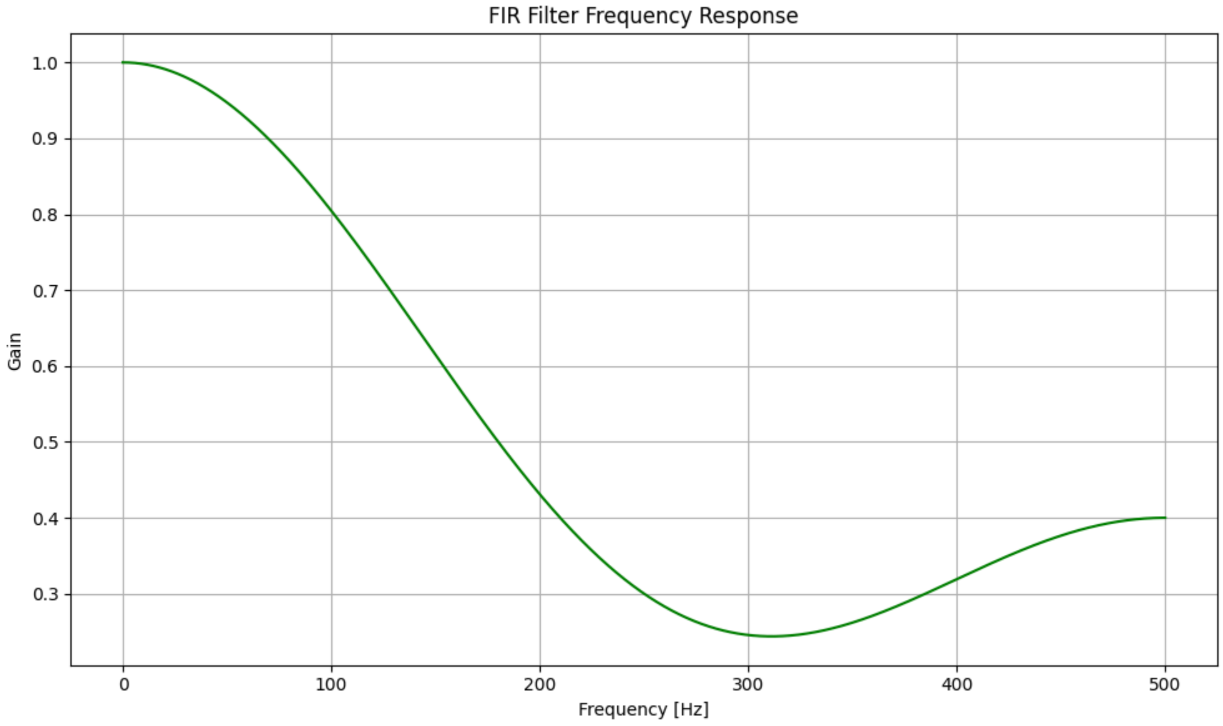
d. If σ\sigmaσ were larger, the time-domain pulse would be wider, and the corresponding frequency spectrum would be narrower, concentrating more energy near 0 Hz.

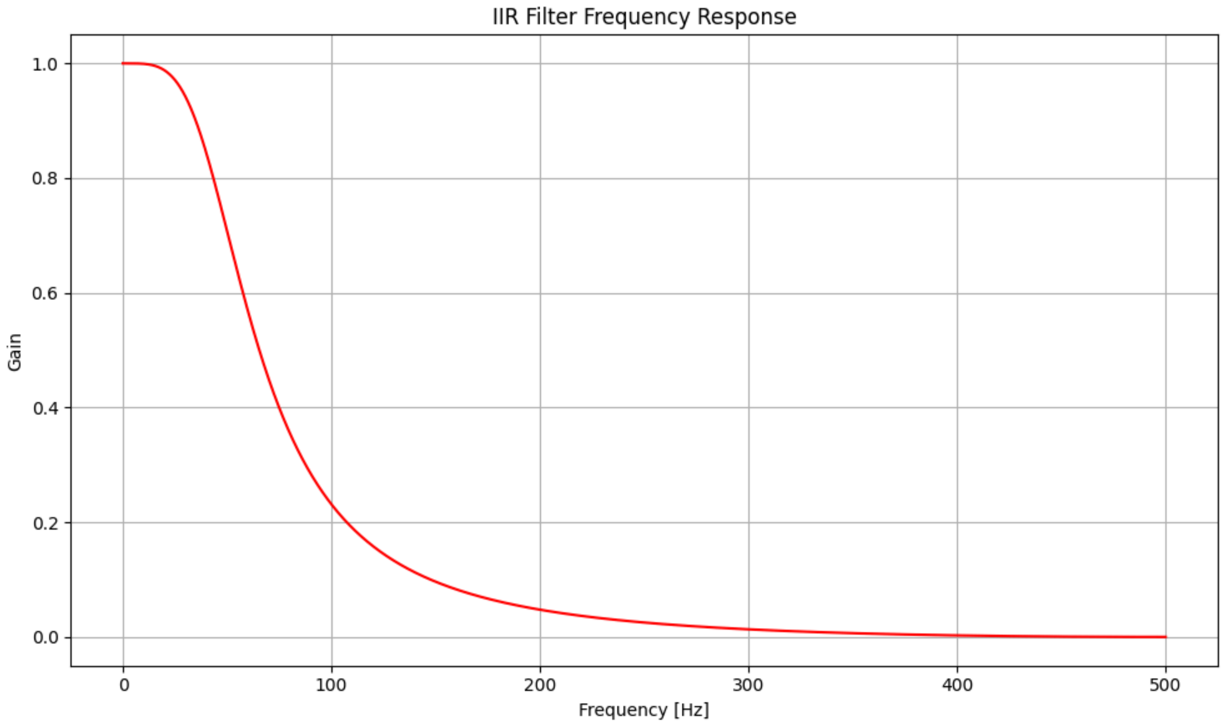
Interpretation of Frequency Components:

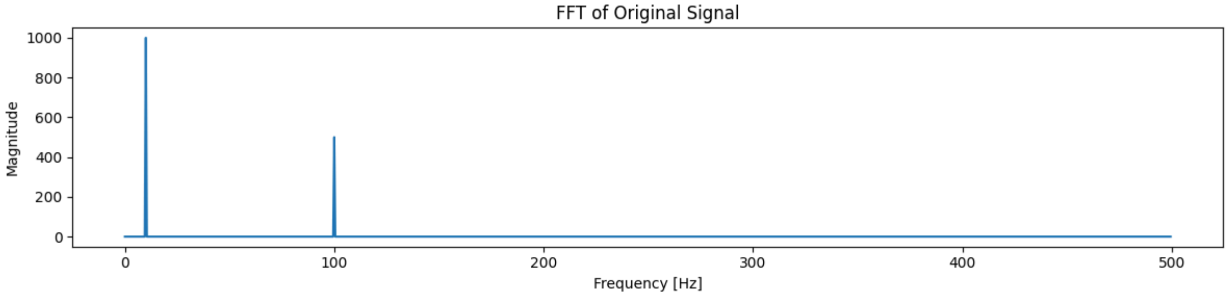
e. The central peak around 0 Hz indicates the strong presence of **DC (low-frequency)** components, while the rapid decay of the spectrum indicates minimal contributions from higher frequencies. This reflects that the Gaussian pulse is smooth and doesn't exhibit rapid changes, which would require higher frequency components.

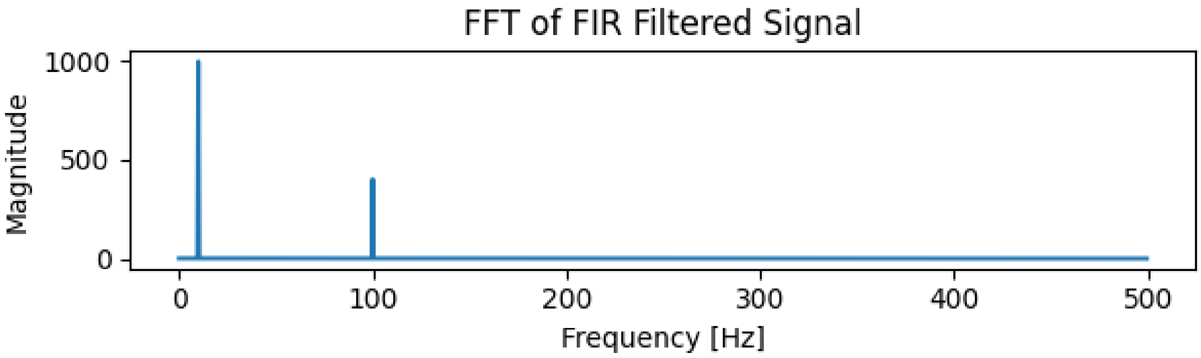
5.

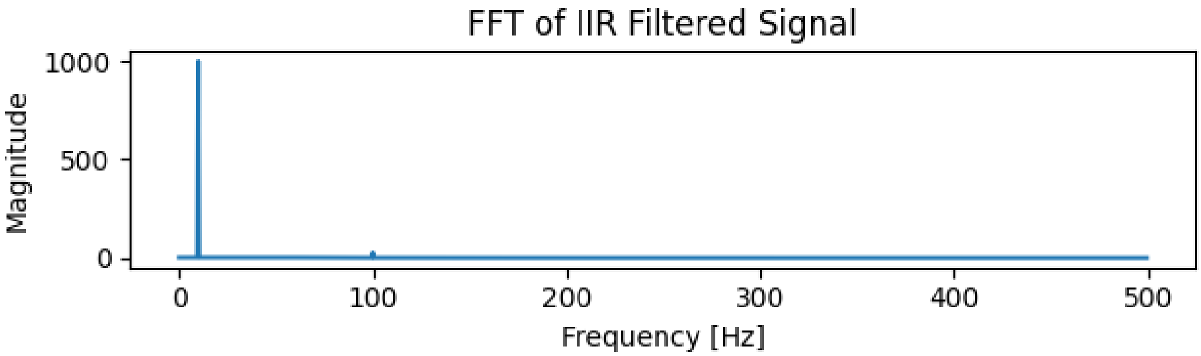

Original Signal x[n]
Original Signal
Amplitude
1
0
−1
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Time [s]
FIR Filtered Signal
FIR Filtered Signal
Amplitude
1
0
−1
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Time [s]
IIR Filtered Signal
IIR Filtered Signal
Amplitude
1.5
1.0
0.5
0.0
−0.5
−1.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Time [s]


FIR Filter Frequency Response
Gain
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
100
200
300
400
500
Frequency [Hz]


IIR Filter Frequency Response
Gain
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500
Frequency [Hz]


FFT of Original Signal
Magnitude
1000
800
600
400
200
0
0
100
200
300
400
500
Frequency [Hz]





FIR Filter (Finite Impulse Response)

1. **Time Domain Behavior**:
   - The FIR filter attenuates the high-frequency component at **100 Hz** while retaining the **10 Hz** component, as seen in the FIR-filtered signal plot.
   - There is some visible **delay** introduced in the signal due to the convolution process, which is characteristic of FIR filters. This delay is caused by the symmetric nature of the FIR filter, and it is proportional to the filter length.
2. **Frequency Domain Behavior**:
   - In the frequency domain, the **100 Hz component** is clearly attenuated in the FFT of the FIR-filtered signal, while the **10 Hz component** is preserved. However, the attenuation at 100 Hz is not complete, and some leakage is visible.
   - The **frequency response** of the FIR filter shows a gradual roll-off after the **50 Hz cutoff frequency**, and as a result, the filter does not entirely remove components near the cutoff, which is a common issue in finite-length FIR filters. This makes the filtering less sharp than with an IIR filter.

IIR Filter (Infinite Impulse Response)

1. **Time Domain Behavior**:
   - The IIR filter effectively removes the high-frequency **100 Hz** component and retains the **10 Hz** component with minimal distortion. Compared to the FIR filter,

    the IIR filter introduces significantly less delay and closely matches the low-frequency content of the original signal.
   - The IIR filter achieves this with fewer coefficients, making it computationally more efficient than the FIR filter. However, recursive IIR filters can introduce phase shifts, although using zero-phase filtering (`filtfilt`) here minimizes this issue.
2. **Frequency Domain Behavior**:
   - The FFT of the IIR-filtered signal shows **complete suppression of the 100 Hz** component, while the **10 Hz component** is preserved cleanly, indicating the IIR filter's sharper frequency cutoff.
   - The **frequency response** of the IIR filter shows a much steeper roll-off after **50 Hz**, which results in better attenuation of higher frequencies compared to the FIR filter. The IIR filter performs better in suppressing unwanted frequencies while retaining the desired low-frequency components.

Comparison of FIR and IIR Filters:

1. **Attenuation Efficiency**:
   - Both the FIR and IIR filters effectively suppress the high-frequency component at **100 Hz**, but the **IIR filter** achieves sharper attenuation, as evidenced by the frequency response and FFT plots.
   - The FIR filter's gradual roll-off leads to some residual presence of the 100 Hz component in the frequency spectrum, whereas the IIR filter eliminates it more effectively.
2. **Delay and Phase Distortion**:
   - The **FIR filter** introduces more delay in the time domain due to its convolution process. The symmetric FIR filter also minimizes phase distortion but at the cost of delay.
   - The **IIR filter**, being recursive, introduces minimal delay while still suppressing the high-frequency content, making it more suitable for real-time applications that require lower latency. Phase distortion can be a concern in IIR filters, but the use of zero-phase filtering (`filtfilt`) avoids this.
3. **Computational Efficiency**:
   - The **FIR filter** requires more coefficients and thus more computational resources to achieve similar performance compared to the IIR filter.
   - The **IIR filter** is more computationally efficient, as it uses fewer coefficients to achieve a sharper cutoff and better attenuation of high frequencies.