Final work for class 20139

For ID: 314679713

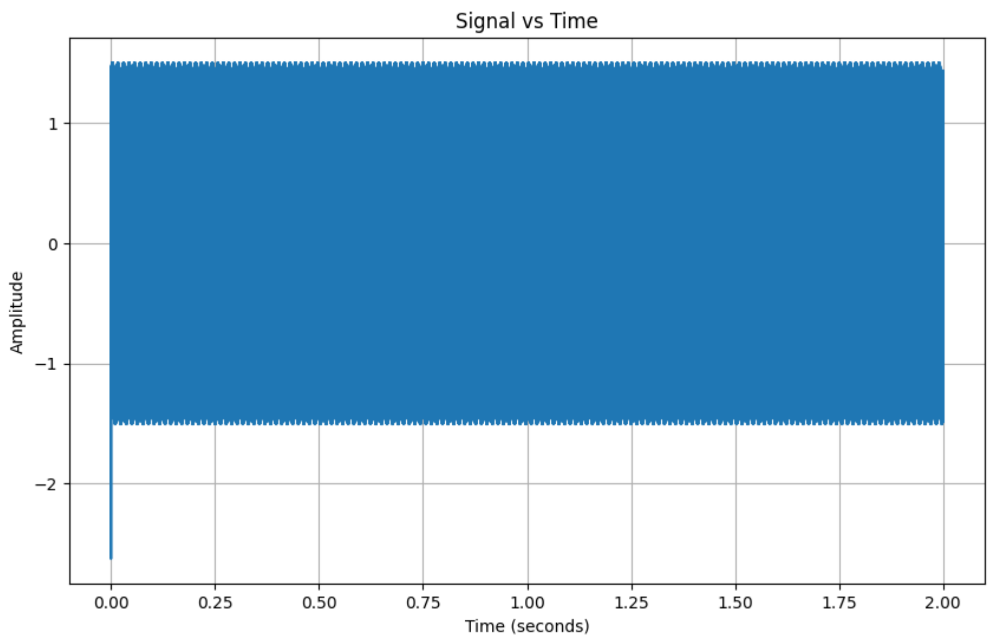
Prepared by:

Yariv Shossberger - ID: 316523406

Ori Toker - ID: 314679713

Q1

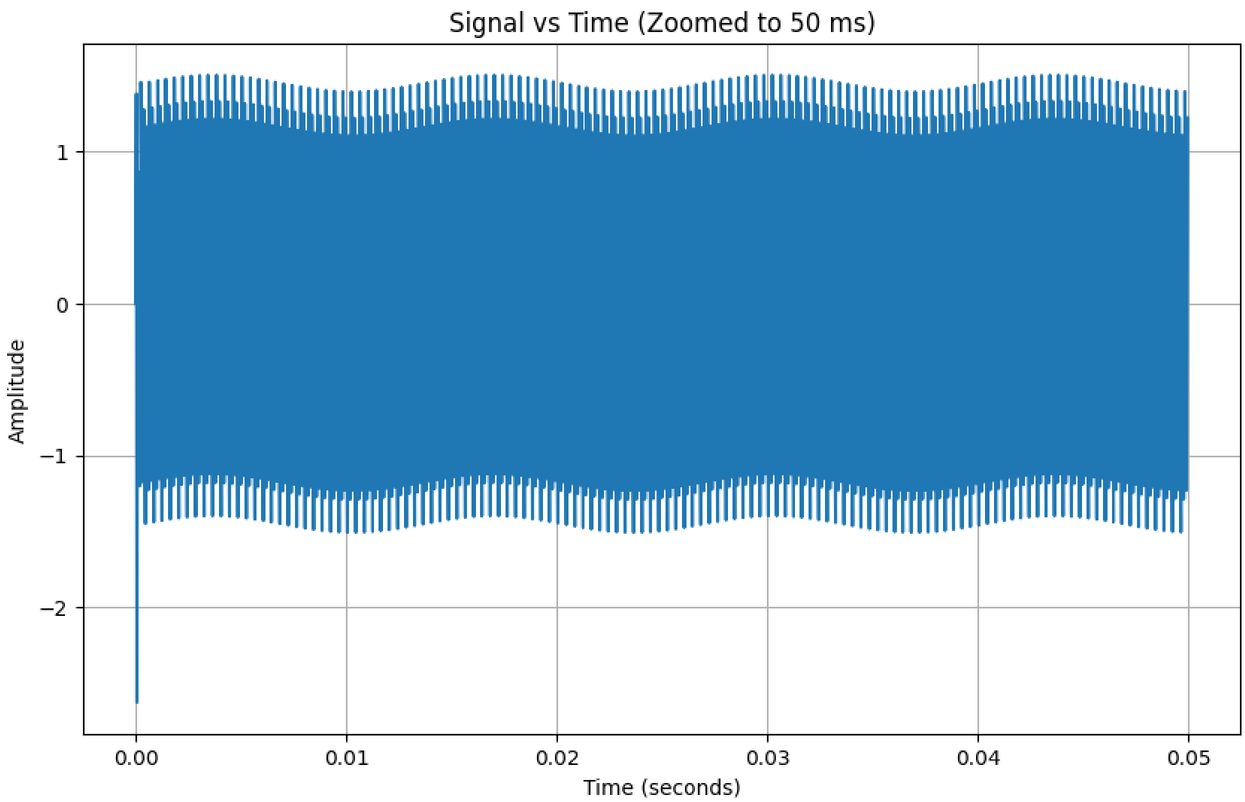
Based on the sampling frequency, the time at each data point will be calculated as $t = \frac{n}{f_s}$, where n is the index of the data point and f_s is the sampling frequency.



The plot generated indicates that the signal is displayed over a time span of about 2 seconds. However, the signal appears to be compressed or clipped, possibly due to the scale of the time axis or the amplitude of the data.

Q2

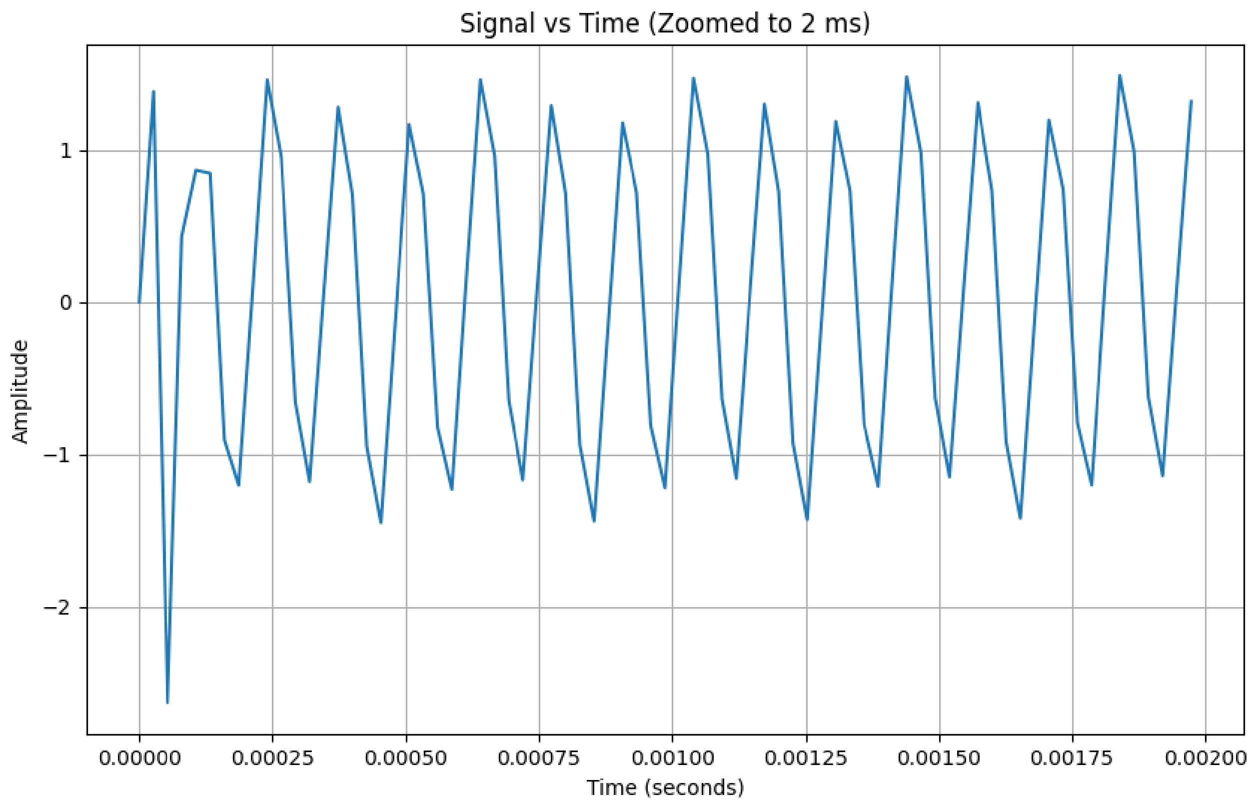
plotting a segment of the data to zoom in and inspect the signal over a smaller duration (50 ms).



<u>Key Observations:</u>

- The signal oscillates around a certain range, with visible fluctuations in the amplitude.
- There's a pattern that suggests the signal could be periodic or modulated in some way.

Let’s zoom in even further:



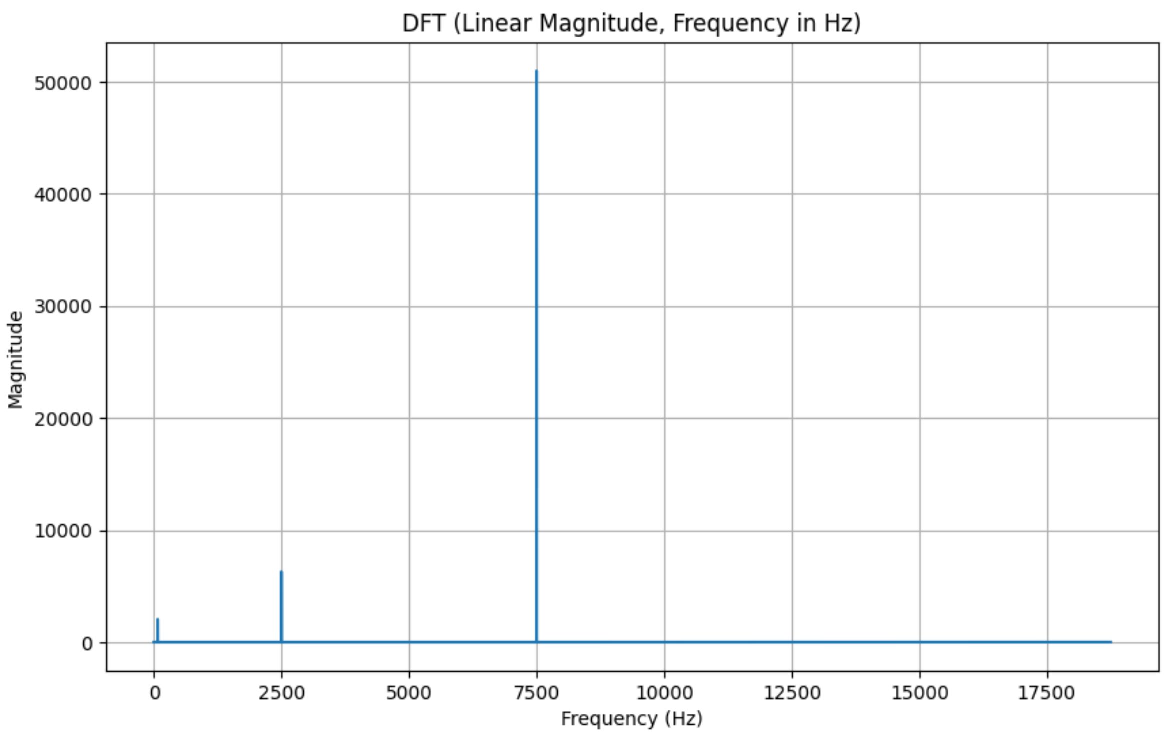
<u>Key Observations:</u>

1. **Oscillatory Nature**: The signal clearly oscillates within this 2 ms window, indicating the presence of multiple frequency components. we can see the peaks and valleys that represent the periodicity of the signal.
2. **Amplitude Variations**: The amplitude of the oscillations varies throughout the time window. The signal's amplitude fluctuates between approximately -2 and 2, suggesting that it contains varying strength in its frequency components.
3. **Transient Effects**: In the very beginning, the signal seems to show some irregularity (sharp dips and peaks), indicating the presence of initial transient behaviour. This irregularity may be due to the system's response at the start or noise in the signal.

Q3

3.1 - Linear Magnitude Scale with Frequency Axis in Hz

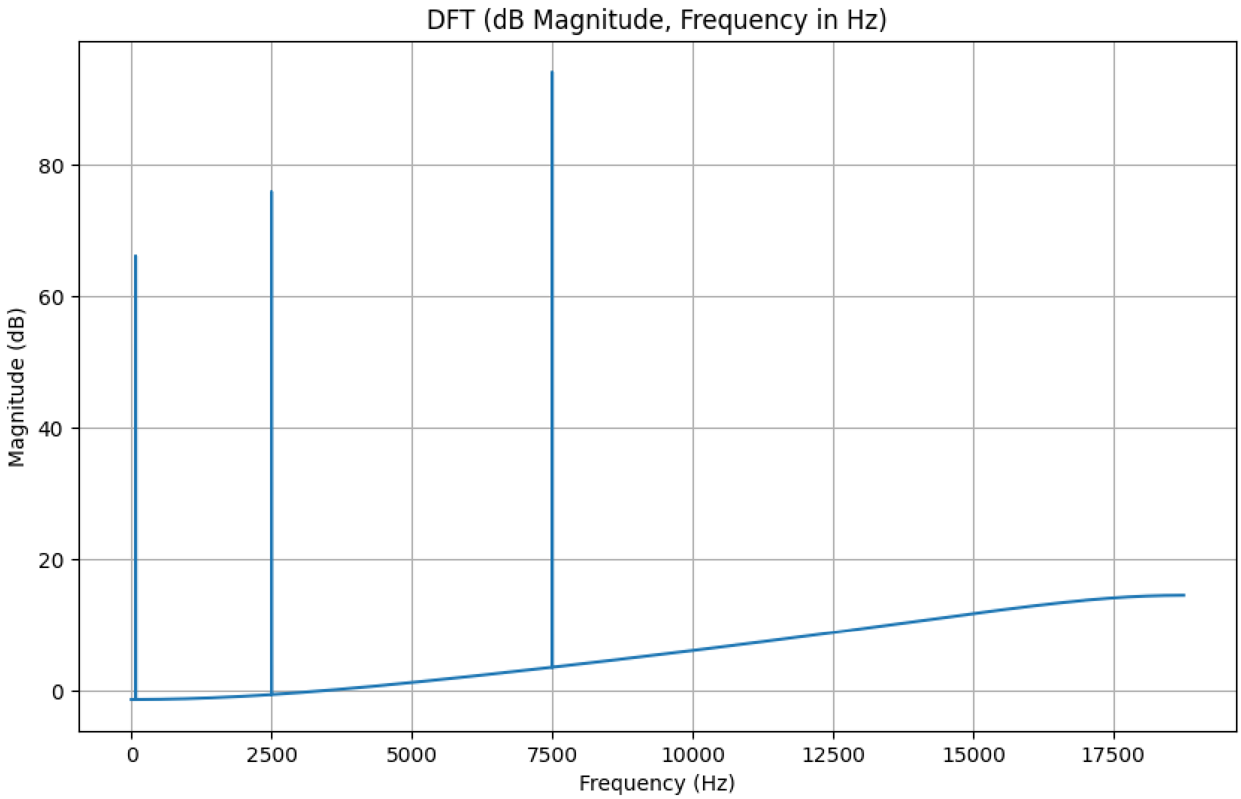
For this, we compute the DFT using the `np.fft.fft` function, and then we plot the magnitude of the result with the frequency axis labelled in Hz.



We can clearly observe a few prominent frequency components, with the most dominant peak around 7500 Hz. This suggests that the signal contains strong periodic components at this frequency.

3.2 - dB Magnitude Scale with Frequency Axis in Hz

After calculating the DFT, we convert the magnitude to decibels (dB) using $20 \times \log_{10}(\text{magnitude})$ and the frequency axis will remain in Hz.

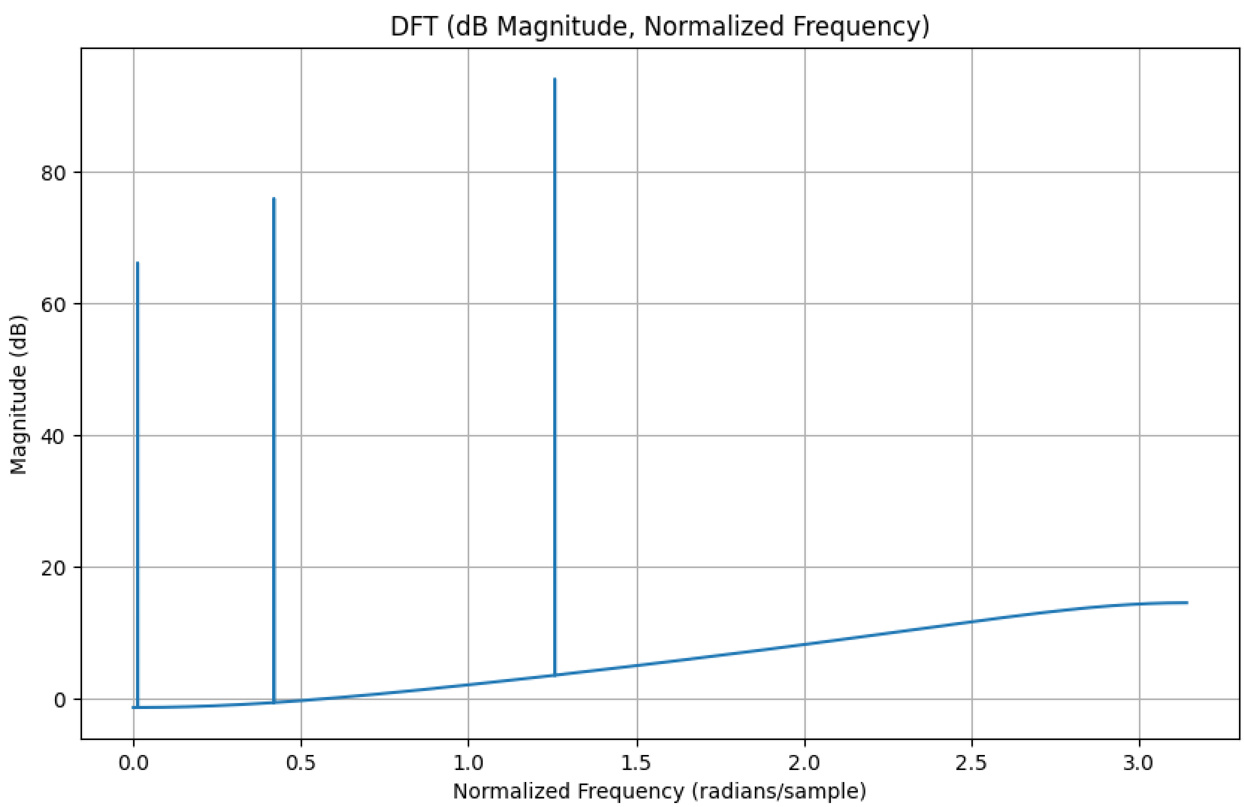


Now we can see the relative strength of the frequency components more clearly, with the most dominant component still around 7500 Hz, exceeding 80 dB.

- For 3.1 and 3.2, the x-axis is in Hz, which is calculated using `np.fft.fftfreq`.

3.3 - dB Magnitude Scale with Normalized Frequency (0, 2π radians/sample)

Here, the DFT magnitude is still plotted in dB, but the frequency axis is normalized **from 0 to 2π** [radians/sample].



- For 3.3, the x-axis is normalized from 0 to 2π radians/sample using `np.fft.fftfreq x2`π.

Q4

With the following transfer function of the distorting system:

$$\boxed{H(z) = \frac{z - 1.1}{z}}$$

The **frequency response** is obtained by substituting $z = e^{j\omega}$, where ω is the normalized frequency (in radians per sample).

Substitute $z = e^{j\omega}$ into the transfer function H(z):

$$\Rightarrow\ H\left(e^{j\omega}\right) = \frac{e^{j\omega} - 1.1}{e^{j\omega}}$$ (Simplify the expression)

$$\Rightarrow\ H\left(e^{j\omega}\right) = 1 - \frac{1.1}{e^{j\omega}} = 1 - 1.1e^{-j\omega}$$

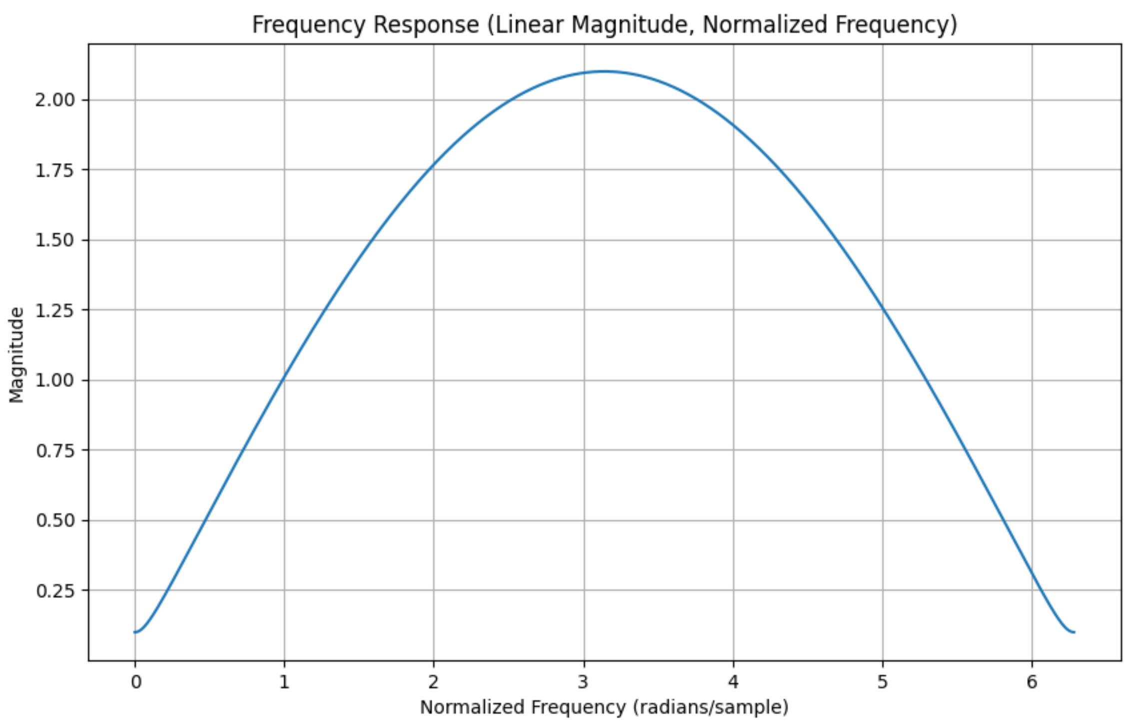
Magnitude Response: $\left|H\left(e^{j\omega}\right)\right| = \left|1 - 1.1e^{-j\omega}\right|$

Phase Response: $\text{Phase}\left(H\left(e^{j\omega}\right)\right) = arg\left(1 - 1.1e^{-j\omega}\right)$

Q5

The term "linear" refers to the fact that the scale is **direct** and **proportional** to the original value.

The linear magnitude is simply the absolute value of the complex frequency response -> $\left|H\left(e^{j\omega}\right)\right|$.



Key Observations:

- The magnitude peaks at ω=π [radians/sample], with a value of approximately 2.
- The response shows how the system amplifies frequencies near the midpoint (around π [radians/sample]) and attenuates others.

Q6

- $H_{ap}(z)$: The **all-pass** component has a magnitude of 1 for all frequencies, meaning it only affects the phase of the system and not the magnitude.

- $H_{min}(z)$: The **minimum-phase** component has all its poles and zeros inside the unit circle (i.e., stable and causal).

We can separate H(z) into all-pass and minimum-phase components by **analyzing its zeros and poles**:

- The zero of the system is at z=1.1 (outside the unit circle).
- The pole of the system is at z=0.

To create a **minimum-phase** system, we reflect the zero outside the unit circle z=1.1 **to inside the unit circle** by placing it at $z = \frac{1}{1.1} \approx 0.909$.

- **Minimum-phase component $H_{min}(z)$:**

The zero is reflected inside the unit circle, giving - $H_{min}(z) = \frac{z-0.909}{z}$

- **All-pass component $H_{ap}(z)$:**

The all-pass part compensates for the phase introduced by moving the zero. It has the form - $H_{ap}(z) = \frac{z-1.1}{z-0.909}$.

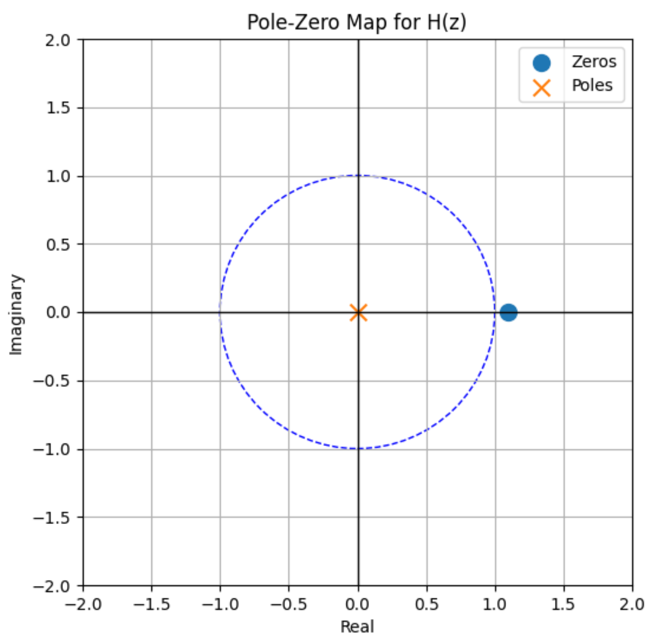
This ensures the **magnitude is 1**, while the phase adjusts for the movement of the zero.

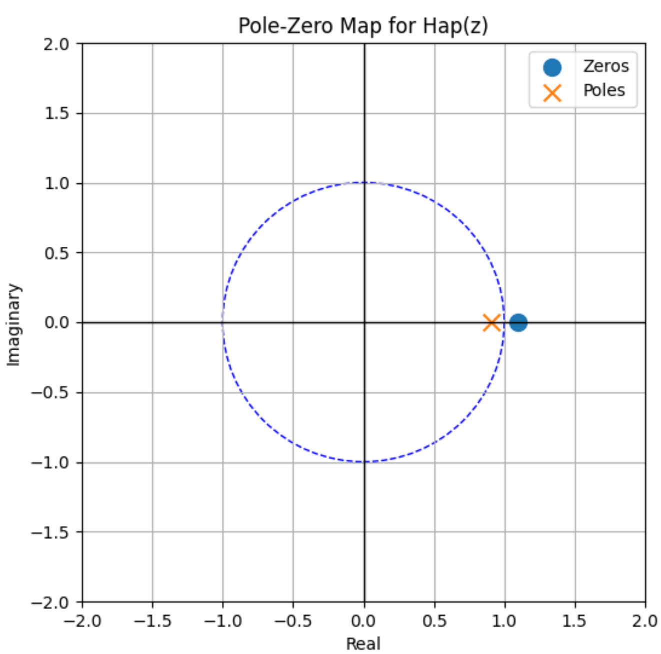
Thus, the transfer function can be written as:

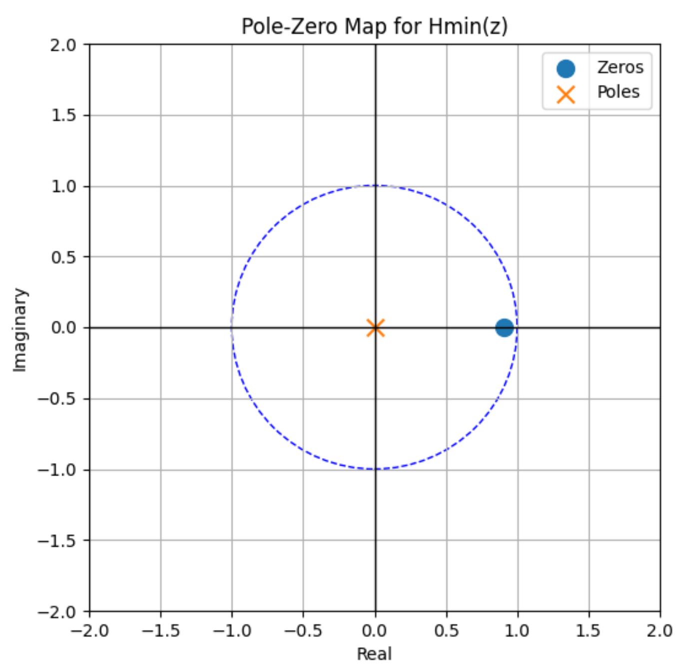
$$\boxed{\Longrightarrow H(z) = H_{ap}(z) \cdot H_{min}(z) = \frac{z-1.1}{z-0.909} \cdot \frac{z-0.909}{z}}$$

Q7

With the help of Python, we will plot the Pole-Zero map:



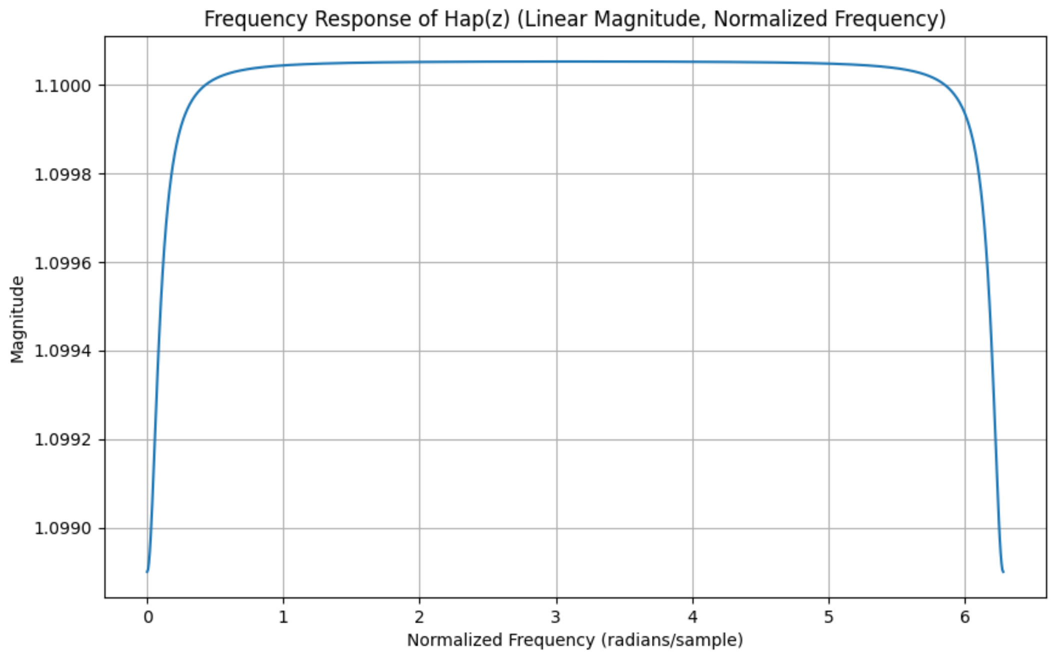


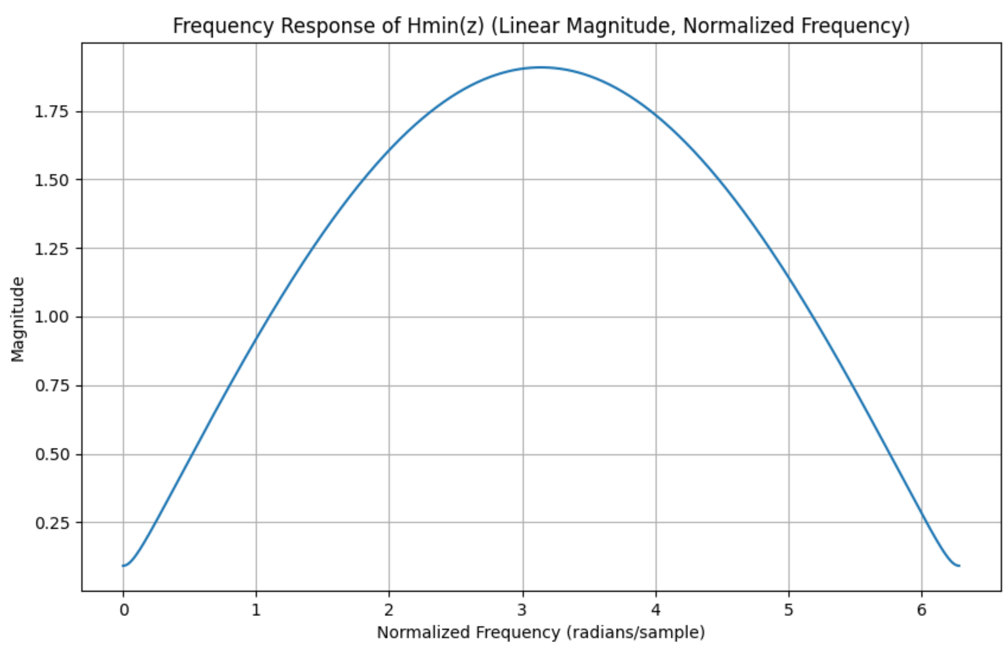


These maps visually confirm the stability of the minimum-phase system and the phase-preserving characteristics of the all-pass system.

Q8

Again, we'll compute the frequency response by substituting $z = e^{j\omega}$ in both transfer functions.





- The plot for the **frequency response of $\boldsymbol{H_{ap}(z)}$** shows that the magnitude is very close to 1 across the normalized frequency range, as expected for an **all-pass filter**. The magnitude of an all-pass filter remains approximately constant, which means it does not alter the magnitude of the input signal but affects the phase.

- The plot for the **frequency response of $\boldsymbol{H_{min}(z)}$** shows a distinct peak at ω=π [radians/sample], with a maximum magnitude of about 1.75. This response is consistent with the behavior of a **minimum-phase system**, where the magnitude varies with frequency, indicating the filtering effect of the system.

Additional Task

To determine whether the system is Generalized Linear Phase (GLP), we need to examine the phase response of the system. GLP systems have certain symmetry properties in their phase responses. However, the given system $H(e^{j\omega}) = \frac{e^{j\omega}-1.1}{e^{j\omega}}$ simplifies to:

$$\Rightarrow\ H(e^{j\omega}) = 1 - \frac{1.1}{e^{j\omega}} = 1 - 1.1e^{-j\omega}$$

Steps to Determine GLP Property

For a system to exhibit GLP properties, its frequency response should have a phase response that is linear (or nearly linear) with respect to ω.

1. **Magnitude Symmetry Check**: We observe whether $H(e^{j\omega})$ is symmetric around ω=0.

2. **Phase Linearity Check**: We check if the phase of $H(e^{j\omega})$ is linear, implying that it can be written as ϕ(ω)=−aω where a is some constant.

Analysis of $H(e^{j\omega})$

Let's rewrite $H(e^{j\omega}) = 1 - 1.1e^{-j\omega}$.

This is not in a standard linear phase form, as it does not have a straightforward linear relationship in its phase response. Specifically, the presence of the constant 1.1 and the term $e^{-j\omega}$ break the symmetry typically associated with linear phase systems.

Conclusion

The system does **not** exhibit the symmetry and linear phase behaviour required for a Generalized Linear Phase (GLP) system.

To find a corresponding Generalized Linear Phase (GLP) system for the given system, we should construct a system with a frequency response that exhibits a linear phase component while closely approximating the behavior of the original $H(e^{j\omega})$.

A GLP system typically has a frequency response in the form:

$$H_{GLP}(e^{j\omega}) = \left|H(e^{j\omega})\right|e^{-j\phi(\omega)}$$

where $\phi(\omega)$ is a linear function of ω, usually in the form $\phi(\omega)=a\omega$ or $\phi(\omega)=a\omega+b$ for some constants a and b. The phase function $\phi(\omega)$ must be linear in ω for it to qualify as GLP.

Finding a GLP Approximation

To approximate the behaviour of $H(e^{j\omega})$, we can:

1. **Determine the Magnitude**: Take $\left|H(e^{j\omega})\right|$ for the magnitude response.

2. **Apply Linear Phase**: Introduce a phase term that is linear in ω, which would give it GLP properties.

Step 1: Calculate $\left|H\left(e^{j\omega}\right)\right|$

The magnitude response of the original system is:

$$\left|H\left(e^{j\omega}\right)\right| = \left|1 - 1.1e^{-j\omega}\right|$$

This can be evaluated as:

$$\left|H\left(e^{j\omega}\right)\right| = \sqrt{1^2 + (1.1)^2 + 2 \cdot 1 \cdot 1.1\cos(\omega)} = \sqrt{2.21 - 2.2\cos(\omega)}$$

Step 2: Construct the GLP System

To make this system GLP, we replace the phase of $H\left(e^{j\omega}\right)$with a linear phase term. Thus, the corresponding GLP system can be expressed as:

$$H_{GLP}(e^{j\omega}) = \left|H(e^{j\omega})\right|e^{-j\alpha(\omega)}$$

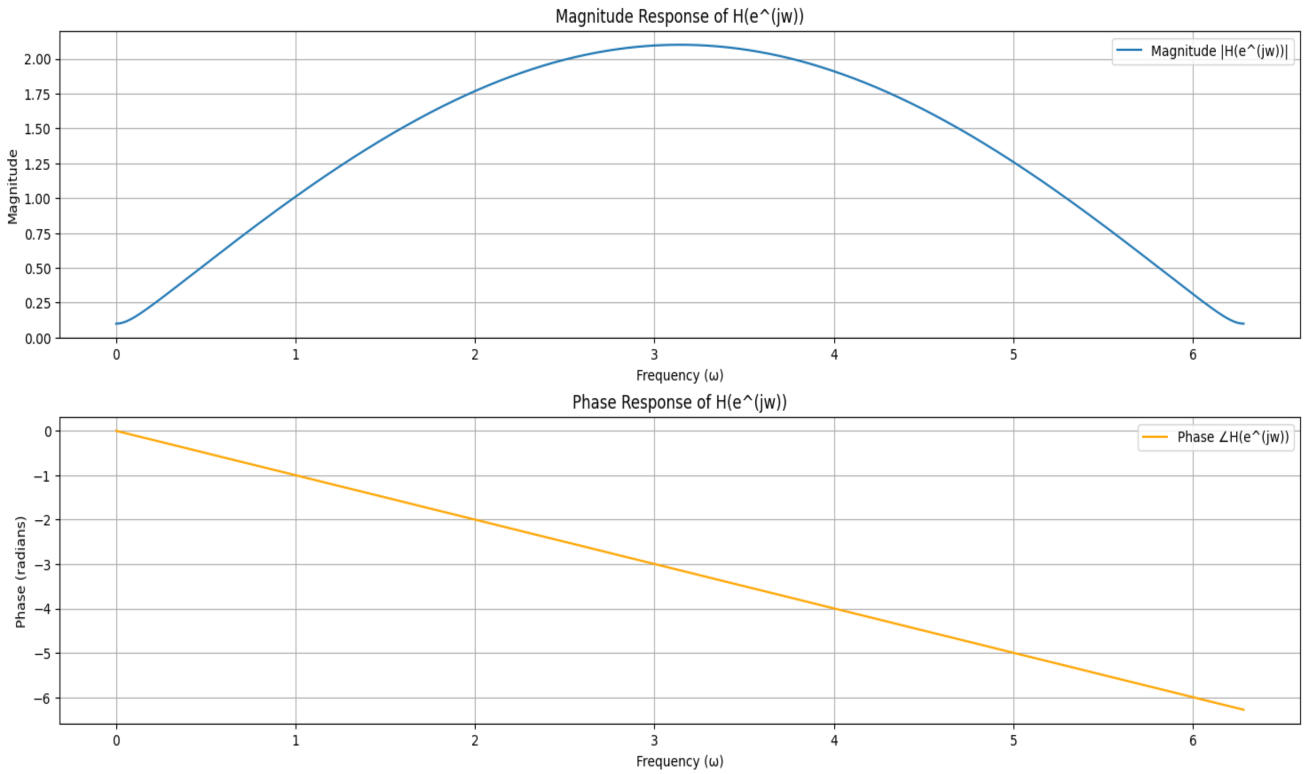
where α is chosen based on desired delay or phase shift characteristics.

The exact value of α depends on how closely we want the GLP system to approximate the phase characteristics of the original system, but typically α is chosen to approximate the phase of the original system near ω=0.

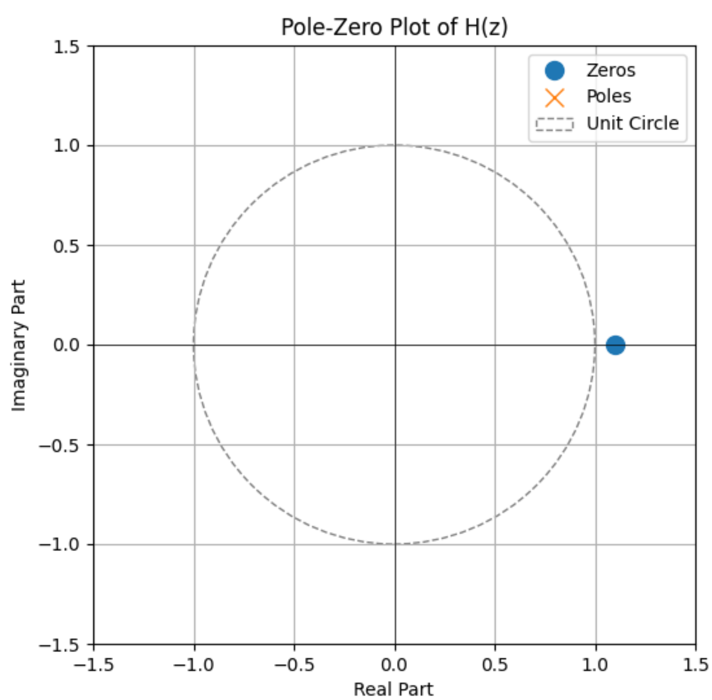
Therefore, we will choose α=1.

So, our corresponding GLP system will be:

$$H_{GLP}\left(e^{j\omega}\right) = \sqrt{2.21 - 2.2\cos(\omega)}\, e^{-j\omega}$$



Here is the system's Pole-Zero map:



The differences between $H_{GLP}(e^{j\omega})$ and $H\left(e^{j\omega}\right)$ lie not only in their mathematical form but in the underlying symmetry that allows the GLP system to maintain phase linearity, **providing predictable and uniform group delay, which the original non-GLP system lacks**.

A non-constant group delay can cause:

- **Phase Distortion**: The non-uniform delay across frequencies causes phase distortion, which can lead to signal distortion, especially for wideband signals.
- **Reduced Signal Fidelity**: In applications where the waveform shape is critical, such as audio or video, non-constant group delay can degrade the signal quality.
- **Unpredictable Timing**: Varying delay across frequencies makes timing less predictable, which can introduce issues in synchronized systems.

The GLP system's constant group delay is advantageous in applications requiring high fidelity and phase accuracy, as it preserves the signal shape. However, designing such filters can be complex and computationally intensive. On the other hand, non-GLP systems are often simpler and more efficient but suffer from phase distortion due to non-uniform group delay, which can degrade signal quality in certain applications. The choice between GLP and non-GLP filters ultimately depends on the specific requirements of the application, balancing design complexity with signal integrity needs.

Q9

The **correction system** typically refers to an inverse system that compensates for the distortions introduced by the original system.

Let's start with the Transfer Function -

we were given the transfer function of the distorting system as:

$$H(z) = \frac{z - 1.1}{z}$$

The correction system should be the **inverse** of this transfer function:

$H_{\text{correction}}(z) = H(z)^{-1} = \dfrac{z}{z - 1.1}$ (Multiply both sides by (z − 1.1) to eliminate the denominator)

$Let's\ define - H_{\text{correction}}(z) = \dfrac{Y(z)}{X(z)} = \dfrac{z}{z - 1.1}$

$\Longrightarrow zY(z) = zX(z) + 1.1Y(z) \Longrightarrow \mathrm{y[n+1] = x[n+1]} + 1.1y[n] \Longrightarrow \mathrm{y[n] = x[n]} + 1.1y[n-1]$

Multiplication by z corresponds to a **time advance** , such that zY(z) corresponds to y[n+1].

converting into the time domain (using the inverse Z-transform)

Moving one sample in time

Therefore, the **Difference Equation**:

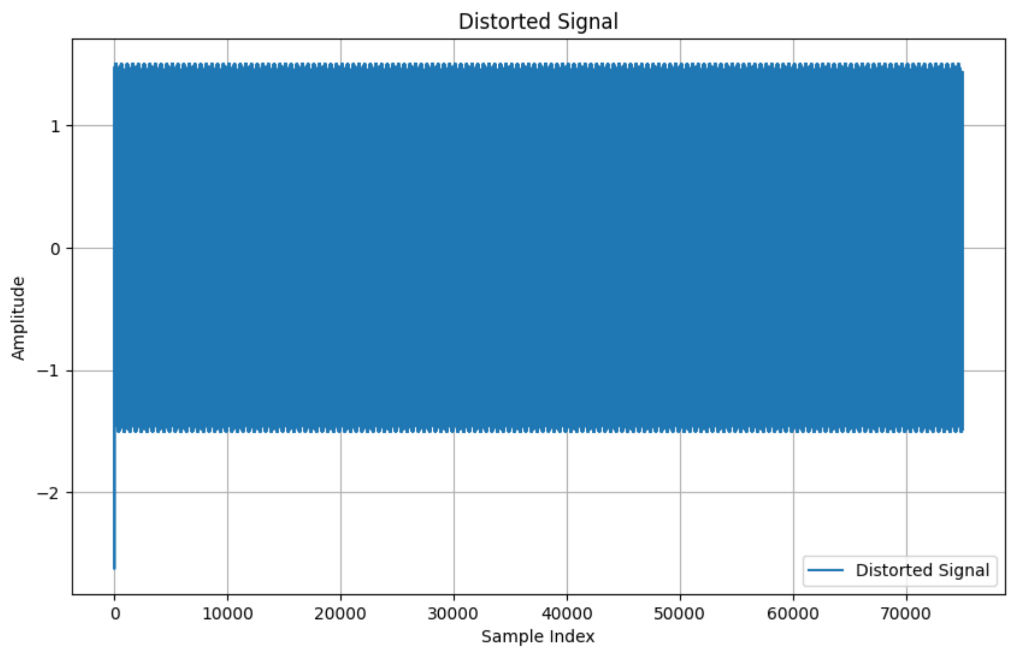
$$y[n] = x[n] + 1.1y[n-1]$$

We got a recursive equation with positive feedback as expected.
The original system distorted the signal by introducing feedback with the coefficient −1.1, The correction system is designed to **cancel** this effect by using positive feedback with +1.1, which cancels out the distortion introduced by the original system.

Q10

Steps to Pass the Signal through the Correction System:

1. **Read the Input Signal**: we already have the signal from the previous steps, which we need to pass through this system.



2. **Apply the Difference Equation**: For each sample of the input signal x[n], we will compute the output y[n] using the formula.

$$y[n] = x[n] + 1.1y[n-1]$$

3. **Iterate Over the Signal**: The equation requires knowledge of the previous output y[n−1], so we'll need to initialize y[0]=0.

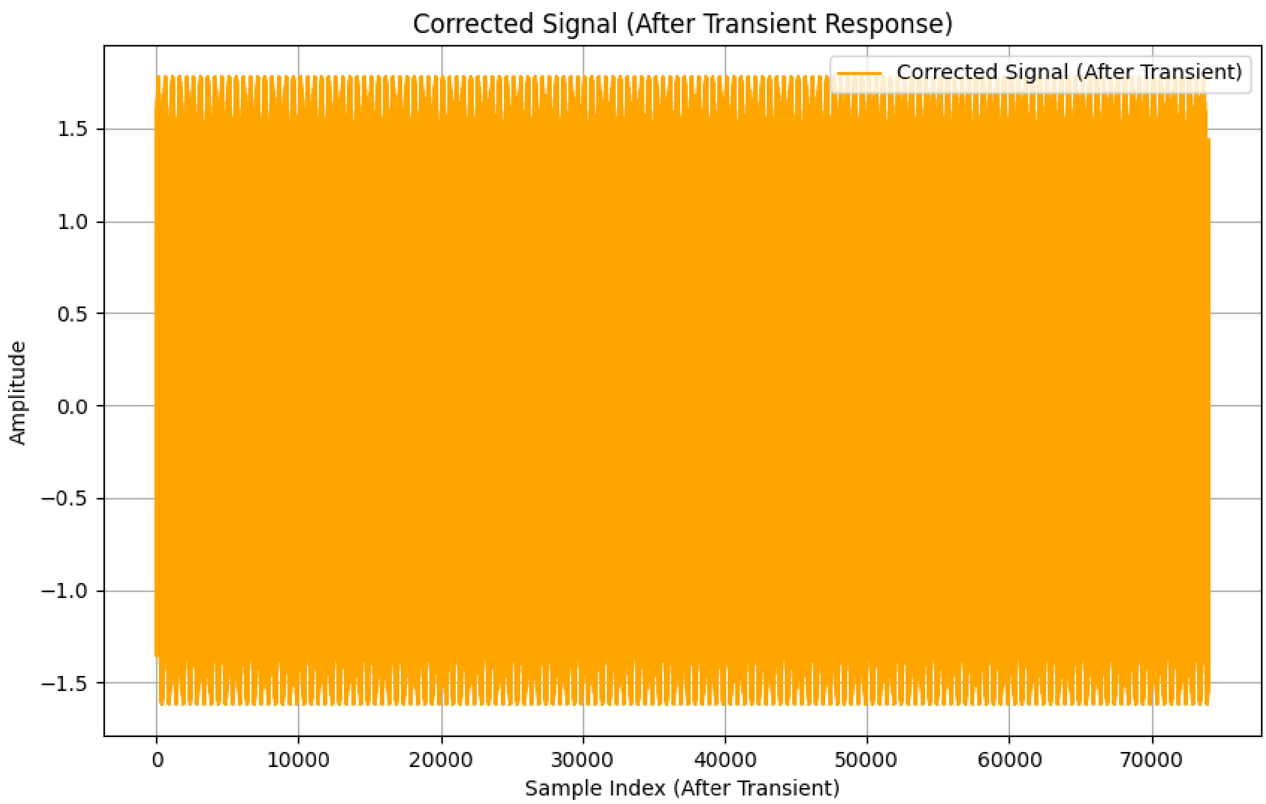
Q11

The **transient response** is the part of the output that occurs right after the system starts operating and before it settles into a steady state. In practical terms, this means that we will ignore the initial portion of the signal where there may be large fluctuations or unstable behavior.

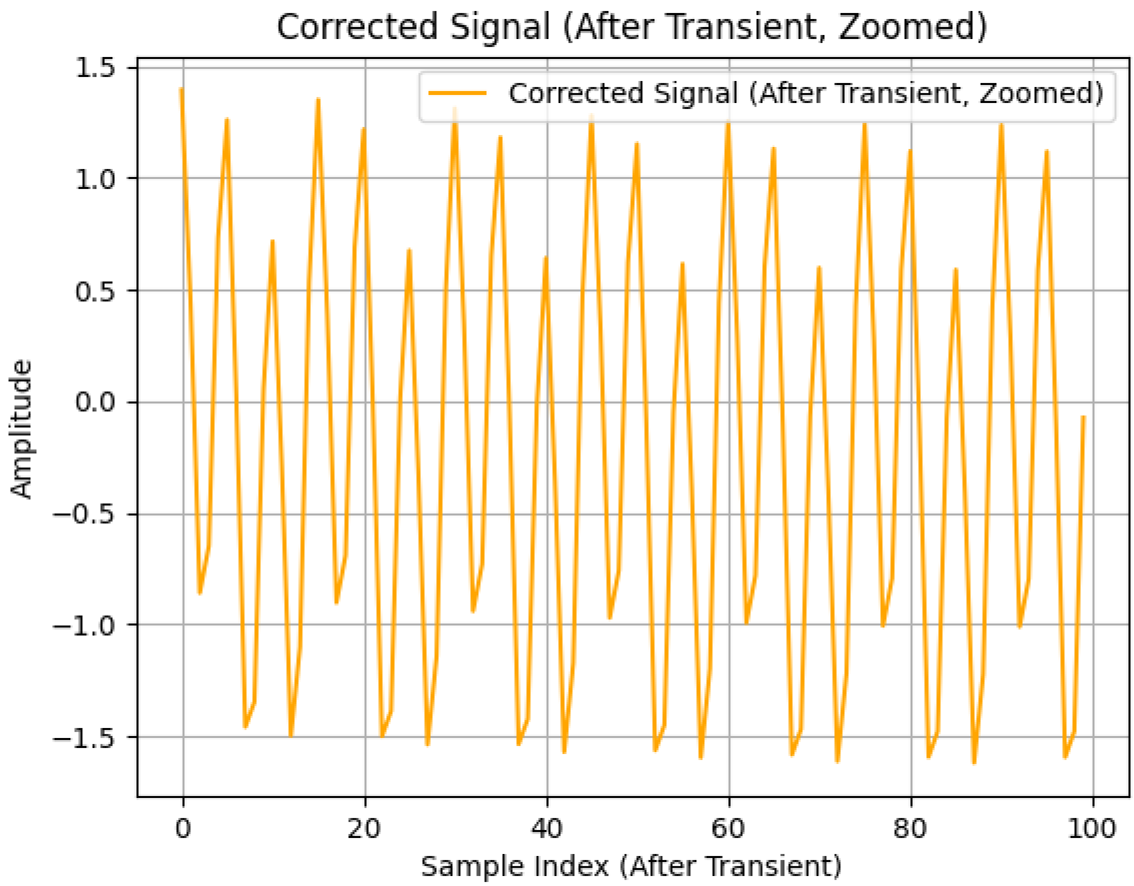
Steps to Plot the Output Signal After the Transient Response:

1. **Identify the Transient Region**: We will discard the first part of the signal (the first 1000 samples), which represents the transient response.

2. **Plot the Steady-State Part**: Once the transient portion is removed, we'll plot the remaining part of the signal.



Let's study it further by zooming in on the corrected signal:

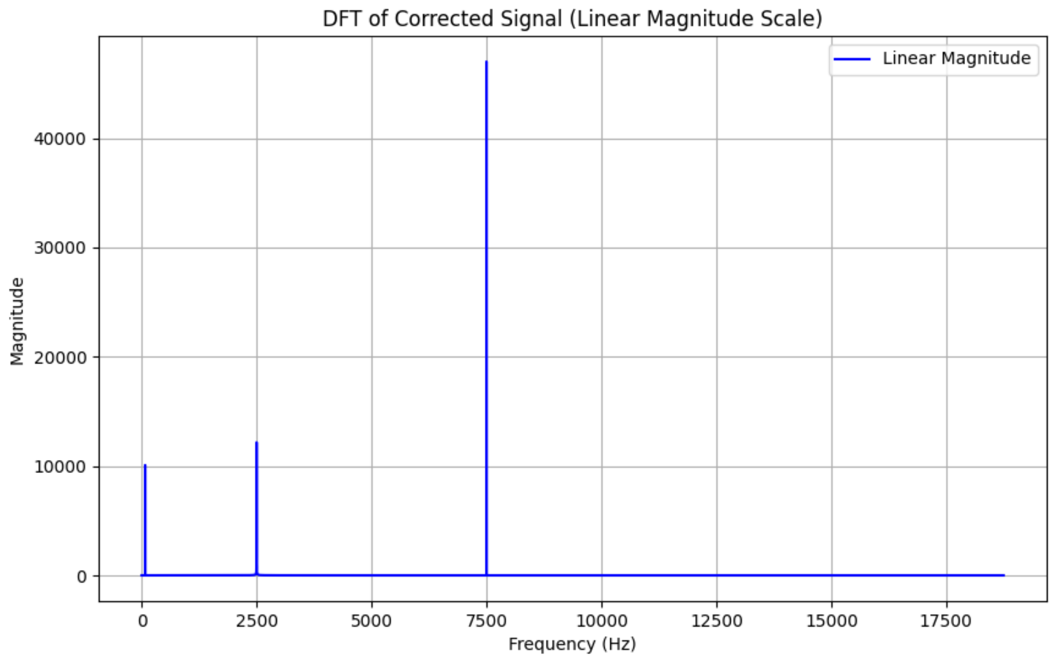


Key Observations:

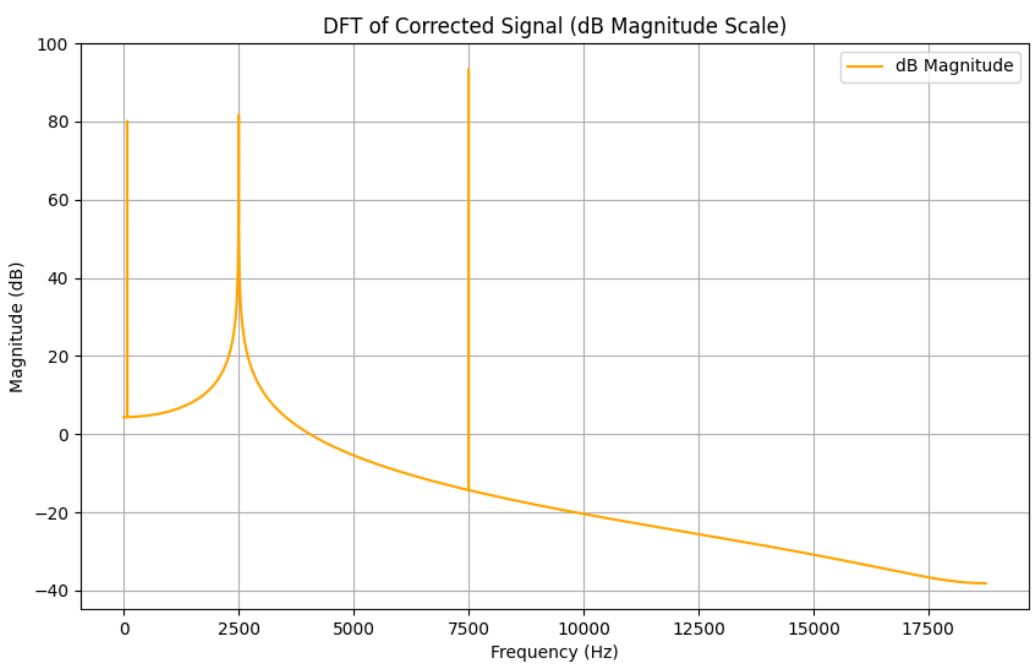
1. **Steady Oscillation**: After the transient portion is removed, the signal exhibits a steady, oscillatory behaviour. The waveform appears to be much more regular compared to the earlier zoomed-in signal (from Q2), which included the initial transient response.
2. **Amplitude Consistency**: The amplitude of the signal is more consistent in this zoomed-in portion, fluctuating between approximately -1.5 and 1.5. This suggests that the correction system has stabilized the signal, reducing irregular fluctuations that may have been present in the original signal.
3. **Frequency**: The oscillations are periodic, with regular peaks and troughs. This indicates that the dominant frequency components have been retained, and the transient and unwanted parts of the signal have been mostly removed.

Q12

We will use the Fast Fourier Transform (FFT) to compute the DFT of the corrected signal in linear and dB magnitude scale.



The linear scale plot shows the magnitude of the DFT across the frequency axis in Hertz (Hz). You can clearly observe the dominant frequency components, with the largest peak occurring at around 7500 Hz.



The dB scale plot shows the magnitude of the DFT in decibels (dB). This scale helps visualize the relative strength of the frequency components more clearly, especially for smaller magnitude components. The dominant frequency peaks are well represented, with the main peak near 7500 Hz and smaller peaks around 2500 Hz.

These plots confirm that the correction system retains specific frequency components that are strong in both linear and dB scales.

Interpretation:

- The strongest frequency component is around 7500 Hz, which is likely the fundamental frequency of the signal.
- The peaks around 2500 Hz and lower frequencies suggest the presence of harmonic or additional components in the signal, which are still present after correction.

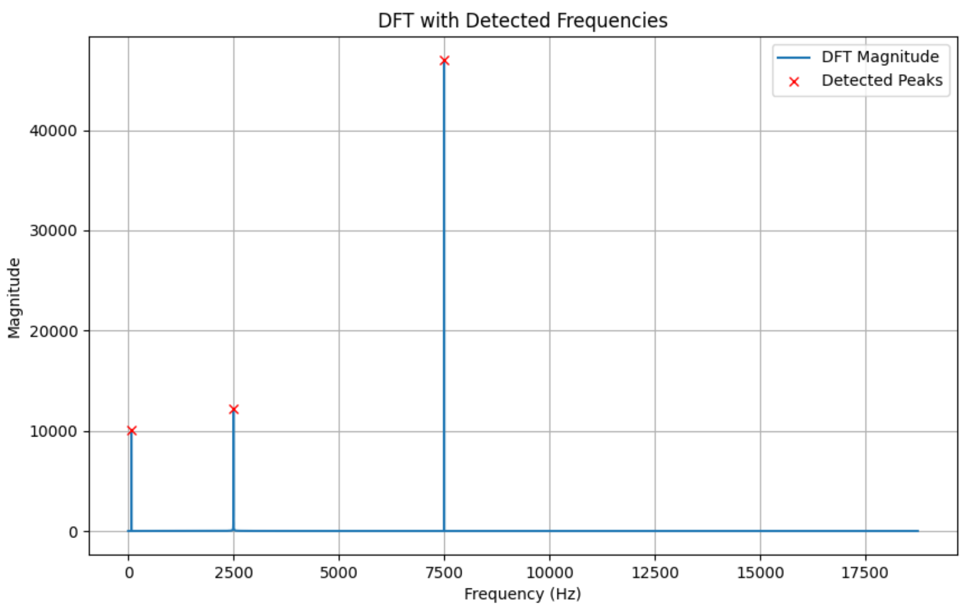
Q13

To find the frequencies present in the input signal from its DFT, we can use the `find_peaks()` function from the `scipy` library. This function will help identify the prominent peaks in the DFT magnitude, which correspond to the dominant frequencies in the signal.

Steps:

1. **Calculating the DFT**: First, we'll compute the DFT of the input signal.
2. **Using `find_peaks()`**: Applying `find_peaks()` function to the magnitude of the DFT to detect the prominent frequencies.
3. **Setting a Threshold**: We'll set a threshold for peak detection to avoid noise and detect only significant frequency components.

The variable `properties` in the `find_peaks()` function is optional and stores additional information about the peaks detected. You can use it to access detailed properties of the peaks, such as their heights, prominences, or widths, depending on the arguments passed to `find_peaks()`.




```
Detected frequencies (Hz): [  75.          2499.83108108 7500.        ]
```

Q14

To design a **200th-order Bandpass Finite Impulse Response (FIR) filter** that passes only the second frequency using a **Hamming window**, and to set the bandwidth to 10% of the center frequency, we will follow these steps:

1. Determine Filter Specifications:

- **Second Frequency (f_center)**: Identify the second dominant frequency from the previous peak detection (with the find_peaks() function in our case 2499.83 Hz).
- **Bandwidth (BW)**: The bandwidth should be **10% of the center frequency**. This defines how wide the filter's passband should be.

$$BW = 0.1 \times f_{center} = 249.98Hz$$

- **Sampling Frequency (fs):** Given earlier as 37,500 Hz.

2. Calculate Normalized Frequencies:

Since the design functions in Python typically use normalized frequencies (where 1 corresponds to the Nyquist frequency, $\frac{f_s}{2} = 18750Hz$), we will calculate the normalized passband edge frequencies f_{low} and f_{high} :

$$f_{low} = f_{center} - \frac{BW}{2} = 2374.84Hz\,, \qquad f_{high} = f_{center} + \frac{BW}{2} = 2624.82Hz$$

$$Normalized\ f_{low} = \frac{f_{low}}{f_s/2} \approx 0.1266, \qquad Normalized\ f_{high} = \frac{f_{high}}{f_s/2} \approx 0.14$$

3. Design the Filter Impulse Response:

To calculate the filter's impulse response, we can use the `firwin()` function from `scipy.signal`, which designs FIR filters using a windowing method. The Hamming window will be applied to the impulse response.

4. Use a Hamming Window:

The Hamming window is applied to the ideal impulse response to control the filter's characteristics. It is designed to minimize the side lobes of the filter's frequency response.

Filter Order: The filter has a 200th order, meaning it uses 201 coefficients (M+1).

The mathematical formulas for the impulse response of the filter and the Hamming window:

1. Ideal Bandpass Filter Impulse Response (No Windowing)

The ideal **Bandpass Filter** is constructed by combining two rectangular filters (one low-pass and one high-pass). The impulse response of an ideal bandpass filter is obtained by subtracting the impulse response of a low-pass filter from a high-pass filter.

Ideal Low-Pass Filter Impulse Response:
For a low-pass filter with cutoff frequency f_c , the impulse response $h_{low}(n)$ is given by:

$$\boxed{h_{low}(n) = \frac{2f_c}{f_s} \cdot sinc\left(2\pi f_c\left(n - \frac{M}{2}\right)\right)}$$

Where:
- f_c is the cutoff frequency,
- f_s is the sampling frequency,
- n is the sample index (ranging from 0 to the filter order M),
- $sinc(x) = \frac{sin(x)}{x}$ is the sinc function.
- M is the filter order.

Ideal High-Pass Filter Impulse Response:

The high-pass filter can be constructed by subtracting the low-pass filter response from a Dirac delta function, which has value 1 at $n = 0$:

$$h_{high}(n) = \delta(n) - h_{low}(n)$$

Ideal Bandpass Filter Impulse Response:

For a bandpass filter with a low cutoff frequency f_{low} and a high cutoff frequency f_{high} , the impulse response is:

$$h_{bandpass}(n) = h_{low,f_{high}}(n) - h_{low,f_{low}}(n)$$

This formula combines two low-pass filters: one with cutoff frequency f_{high} and one with cutoff frequency f_{low}.

Therefore, we get:

$$\boxed{h_{bandpass}(n) = \frac{2f_{high}}{f_s} \cdot sinc\left(2\pi f_{high}\left(n - \frac{M}{2}\right)\right) - \frac{2f_{low}}{f_s} \cdot sinc\left(2\pi f_{high}\left(n - \frac{M}{2}\right)\right)}$$

2. Hamming Window:

The Hamming window is used to taper the ideal impulse response to reduce side lobes in the frequency response, resulting in better stopband attenuation.
The Hamming window is defined as:

$$\boxed{w(n) = 0.54 - 0.46cos\left(\frac{2\pi n}{M}\right)}$$

Where:
- n is the sample index (from 0 to M),
- M is the filter order.

3. Windowed Impulse Response:

Once the ideal impulse response of the bandpass filter $h_{bandpass}(n)$ is calculated, it is multiplied by the Hamming window to obtain the actual impulse response of the filter:

$$h_{\text{windowed}}(n) = h_{bandpass}(n) \cdot \mathrm{w(n)}$$

This ensures that the filter is smoothed, reducing the sharp transitions that would cause unwanted side lobes in the frequency response.

The bandwidth and Cutoff frequencies were already calculated in section 1 and 2 at the beginning of the question.

Q15

Now let's place all values:

$$\Rightarrow h_{bandpass}(n) = 0.14sinc\big(5250\pi(n-100)\big) - 0.1266sinc\big(4750\pi(n-100)\big)$$

$$\Rightarrow w(n) = 0.54 - 0.46cos\left(\frac{\pi n}{100}\right)$$

So, the windowed impulse response will be:

$$h_{\text{windowed}}(n) = h_{bandpass}(n) \cdot \mathrm{w(n)}$$

The **Transfer Function $H(z)$** is the Z-transform of the filter's impulse response $h_{\text{windowed}}(n)$. The general form of the transfer function for an FIR filter is:

$$H(z) = \sum_{n=0}^{M} h_{\text{windowed}}(n) \cdot z^{-n}$$

Then the transfer function $H(z)$ becomes (M=200):

$$H(z) = \sum_{n=0}^{200} h_{\text{windowed}}(n) \cdot z^{-n}$$

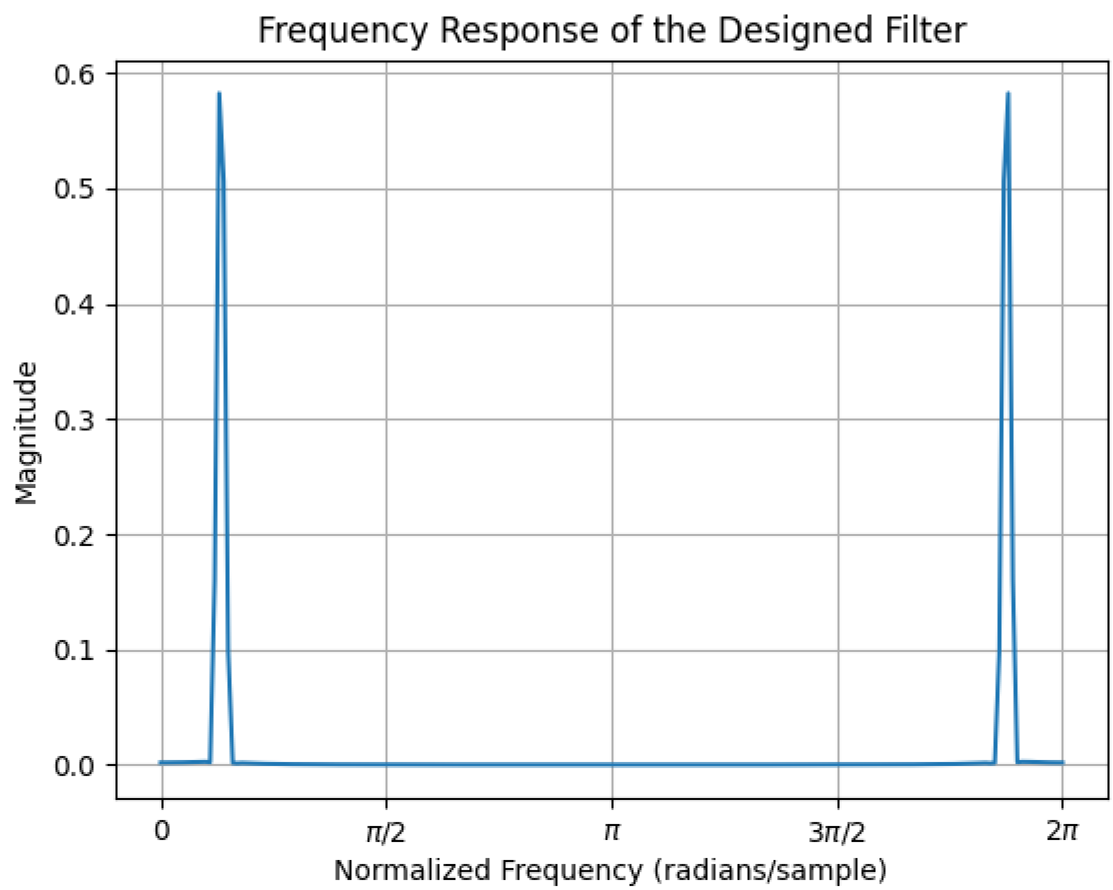
For the frequency response we will substitute $z = e^{j\omega}$, and we will get:

$$H\left(e^{j\omega}\right) = \sum_{n=0}^{200} h_{\text{windowed}}(n) \cdot e^{-j\omega n}$$

Q16

Steps for Frequency Response Calculation:

1. **Manual DFT Calculation**: We'll compute the Discrete Fourier Transform (DFT) of the impulse response manually.
2. **Magnitude Calculation**: The magnitude of the frequency response is derived from the DFT result.
3. **Frequency Mapping**: We'll map the DFT result to the corresponding normalized frequency values (radians per sample from 0 to 2π).



The plot shows the frequency response of the designed bandpass filter, with peaks near the normalized frequency range where the passband should be located. The behavior at 0 and 2π radians/sample corresponds to the filter's designed frequency characteristics, with two prominent peaks.

Q17

The difference equation for the designed FIR filter is based on the impulse response coefficients. Since this is an FIR (Finite Impulse Response) filter, the difference equation will not have any feedback terms. The general form for the difference equation of an FIR filter is:

$$y[n] = b_0 \cdot x[n] + b_1 \cdot x[n-1] + b_2 \cdot x[n-2] + \cdots + b_M \cdot x[n-M]$$

Where:

- $y[n]$ is the filter's output.
- $x[n]$ is the filter's input.
- $b_0, b_1, \ldots, b_M$ are the filter coefficients (which are the impulse response values).
- M is the filter order (in our case, M=200).

Applying the Coefficients: The coefficients $h[n]$ are calculated from the filter design.

So, the difference equation is:

$$y[n] = h[0] \cdot x[n] + h[1] \cdot x[n-1] + h[2] \cdot x[n-2] + \cdots + h[200] \cdot x[n-200]$$

Or:

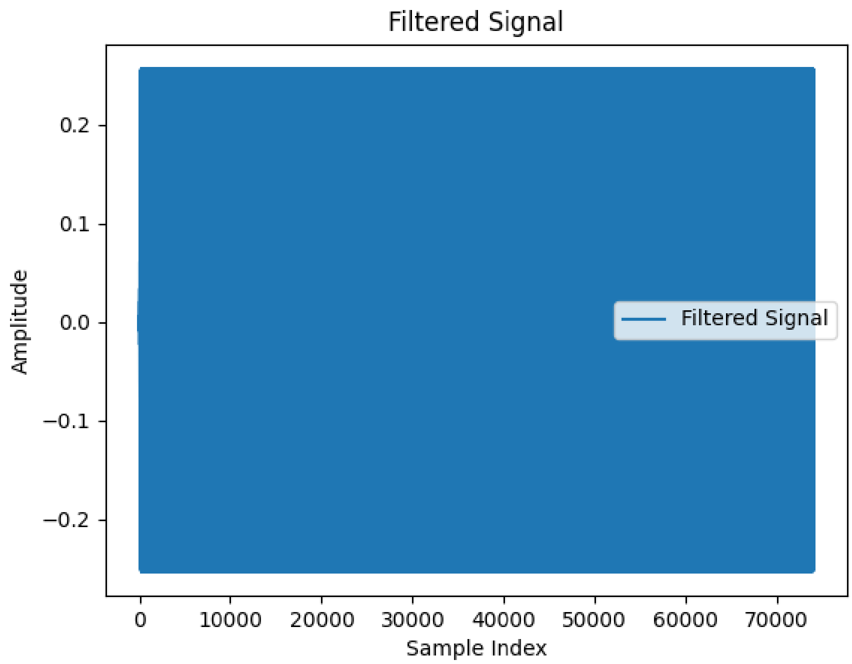
$$\boxed{y[n] = \sum_{k=0}^{200} h_{\text{windowed}}[k] \cdot x[n-k]}$$

This equation applies to your designed filter, where the impulse response $h[k]$ is determined by the bandpass filter design (including the Hamming window).

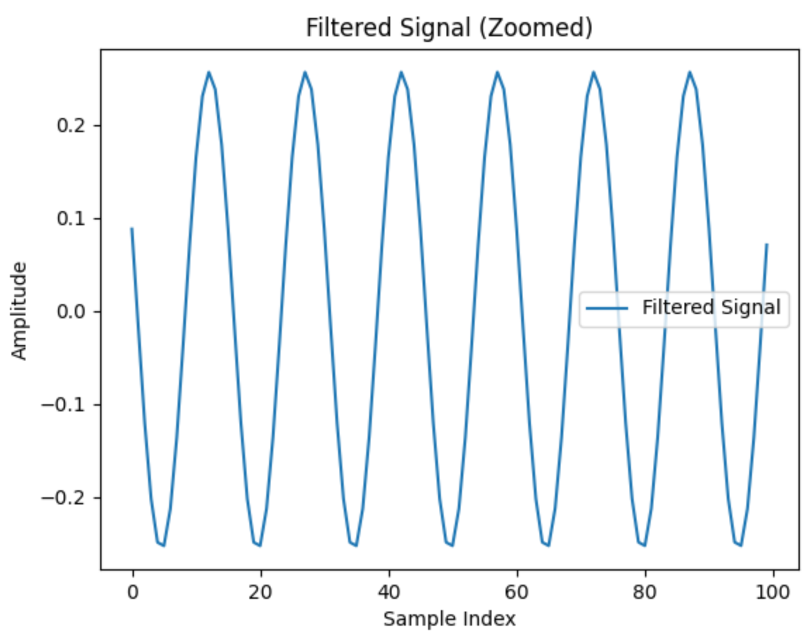
In the time domain, filtering with an FIR filter is **mathematically equivalent to convolving the input signal with the filter's impulse response**.

Q18, Q19

To pass the input signal through the designed filter, we will need to perform convolution between the input signal and the impulse response of the filter. In Python, this can be done using the `scipy.signal.lfilter()` when working with FIR filters.



It seems the filtered signal is very compressed in amplitude, making it hard to distinguish any detailed behavior. We will zoom in on a 100-sample range to visualize the finer details better.



Now after zooming the plot, it presents the filtered signal more clearly. We can observe the sinusoidal behavior, which indicates that the band-pass filter is effectively isolating a specific frequency component of the input signal. This result is consistent with a properly working filter that passes the desired frequency range while attenuating others.

The peaks of the sinusoidal signal might not be ideal due to several factors:

1. Filter Design Imperfections
2. Windowing Effects
3. Finite Impulse Response (FIR) Approximation
4. Signal Leakage
5. Numerical Precision and Quantization
6. Edge Effects from Convolution
7. Sampling Rate and Aliasing

To improve the result, we could try:

- Increasing the filter order to sharpen the roll-off and reduce leakage.
- Exploring different windowing techniques with a trade-off between side lobes and transition bands.

Here are some commonly used windowing techniques:

Window	Main Lobe Width	Side Lobe Level	Best For
Rectangular	Narrow	High	High resolution, but high spectral leakage
Hamming	Moderate	-42 dB	Good balance between leakage and resolution
Hann	Moderate	-31 dB	Smoother, less leakage than rectangular
Blackman	Wide	-74 dB	Excellent suppression of side lobes
Kaiser	Adjustable	Adjustable	Flexibility in shaping the response
Tukey	Adjustable	Moderate	Blending rectangular and Hann characteristics
Bartlett	Moderate	-26 dB	Simple, moderate leakage
Chebyshev	Adjustable	Adjustable	Minimizing maximum side lobe
Gaussian	Adjustable	Very low	Smooth, low-leakage responses

Which to Choose?

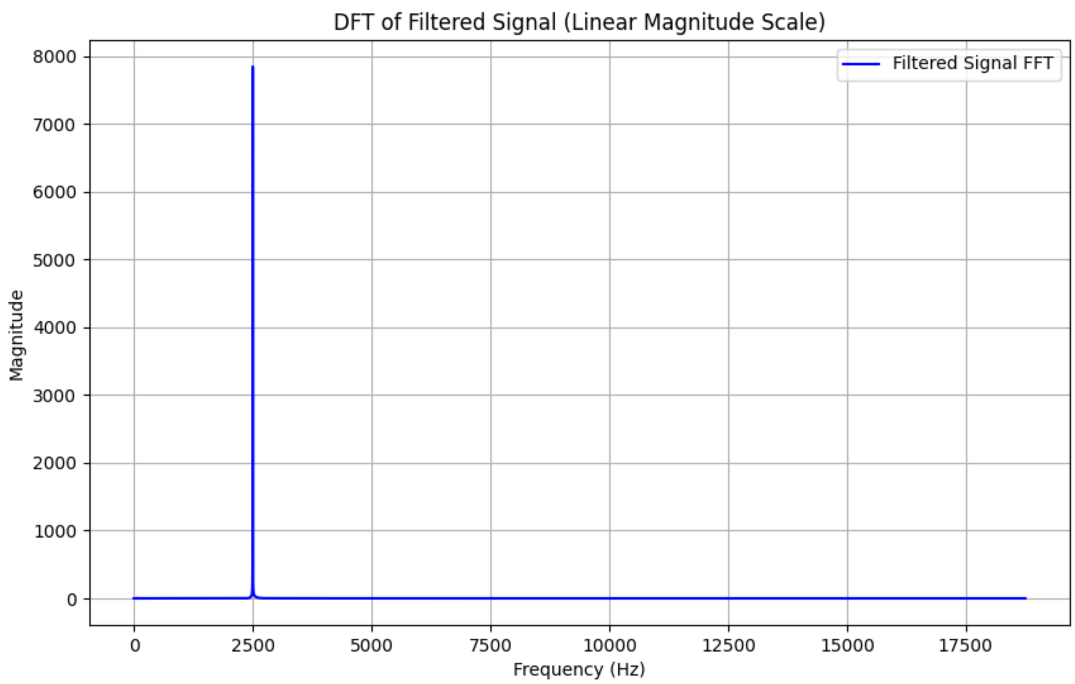
- If **side lobe suppression** is a main concern (i.e., to reduce leakage), we could try **Blackman**, **Kaiser**, or **Chebyshev** windows.
- If **frequency resolution** is more important (i.e., a narrow main lobe), we could try **Rectangular**, **Hamming**, or **Hann** windows.
- If we want **flexibility** to adjust between these properties, the **Kaiser** or **Chebyshev** windows are the best options.

Q20, Q21

For plotting the DFT of the filter's output signal, we can use the following steps:

Step-by-Step Approach:

1. **Perform FFT on the filtered signal** to obtain the frequency domain representation.
2. **Calculate the frequency axis** based on the sampling frequency.
3. **Plot the magnitude spectrum** in linear scale.



In the plot of the DFT of the filtered signal, we observe a significant peak at approximately 2500 Hz, which corresponds to the frequency passed by the band-pass filter. The filter was designed to isolate this frequency, and as a result, the other frequency components present in the original signal are heavily attenuated, leaving only the 2500 Hz component prominent in the spectrum.

The results demonstrate that the filter has successfully performed its function, allowing only the targeted frequency to pass through while attenuating all other components. This outcome validates the correct design and application of the band-pass filter.

Summary

In this project, we performed extensive analysis and processing of a given signal using various digital signal processing techniques. The primary objectives were to **analyze the frequency content of the signal**, **correct distortions using a correction system**, **design and implement a band-pass filter**, and **evaluate the performance of the correction and filtering processes** through both time and frequency domain analysis.

We began by examining the signal in both the time and frequency domains. The Discrete Fourier Transform (DFT) was used to reveal the signal's frequency components, highlighting dominant frequencies in the spectrum. We also implemented a correction system that addressed signal distortions by applying a difference equation, helping to restore the signal's original characteristics.

Next, we **designed a band-pass filter using the windowing technique** with a **Hamming window**. The filter was designed to pass only the second frequency component in the signal, with a bandwidth of 10% around that frequency. The impulse response of the filter was calculated using the sinc function, and the frequency response was analyzed using both linear and dB magnitude scales.

Finally, the corrected and filtered signals were compared in terms of their time-domain representation and frequency spectrum. The performance of the filter was validated through the DFT, demonstrating that the desired frequency was retained, and other frequency components were attenuated as expected.

Conclusions

1. **Signal Distortion and Correction**: The application of the difference equation effectively corrected the distorted signal. **The transient response at the start of the correction process was successfully removed** by ignoring the first few samples, as shown in the analysis of the signal after transient removal.

2. **Frequency Analysis**: The DFT provided clear insights into the dominant frequencies in the signal. Peaks at specific frequencies indicated the presence of multiple sinusoidal components in the original signal, which were detected and analysed. This analysis laid the foundation for the subsequent filtering steps.

3. **Filter Design**: The band-pass filter was designed using the windowing technique with a Hamming window. **The filter successfully isolated the second frequency component while attenuating other frequencies**. This result was **confirmed by the DFT of the filtered signal**, which showed a dominant peak at the desired frequency and minimal energy elsewhere in the spectrum.

4. **Windowing Effects**: The choice of the Hamming window helped reduce spectral leakage, allowing for a smoother transition at the filter's passband edges. Other windowing techniques, such as Hanning or Blackman, could have been explored to evaluate their impact on the frequency response and sidelobe attenuation.

5. **Overall Performance**: The project **successfully demonstrated the complete process of signal analysis, correction, and filtering**. The designed filter performed as expected, isolating the target frequency component while maintaining signal integrity.

In conclusion, the combination of signal correction, window-based filter design, and frequency analysis techniques allowed for effective signal restoration and component isolation. This project highlights the importance of digital signal processing methods in practical applications where signal distortion, noise, or interference needs to be managed or filtered out.