Digital Signal Processing HW4

2. Impulse Response

Problem 1: Impulse response of audio echo system

Let the sampling frequency be 20 kHz.

(a) Start with a 2-second $\delta[n]$. How many samples should it contain?

To determine the number of samples in a 2-second $\delta[n]$:

$$N = f_s \cdot T = 20000 \cdot 2 = 40000 \text{ samples}$$

The impulse response should contain 40000 samples with a single 1 at the beginning.

Octave code:

```
fs = 20000;
duration = 2;
N = fs * duration;

delta = zeros(1, N);
delta(1) = 1;
```

(b) Modify the $\delta[n]$ to create an echo every 0.2 seconds, and give the locations of the non-zero samples.

Since the sampling rate is 20000 samples/sec, an echo every 0.2 seconds corresponds to every 0.2 * 20000 = 4000 samples.

Octave code:

```
echo_interval_sec = 0.2;
echo_samples = round(echo_interval_sec * fs);

echo_positions = 1:echo_samples:N;
disp('Echo positions (samples):');
disp(echo_positions);
```

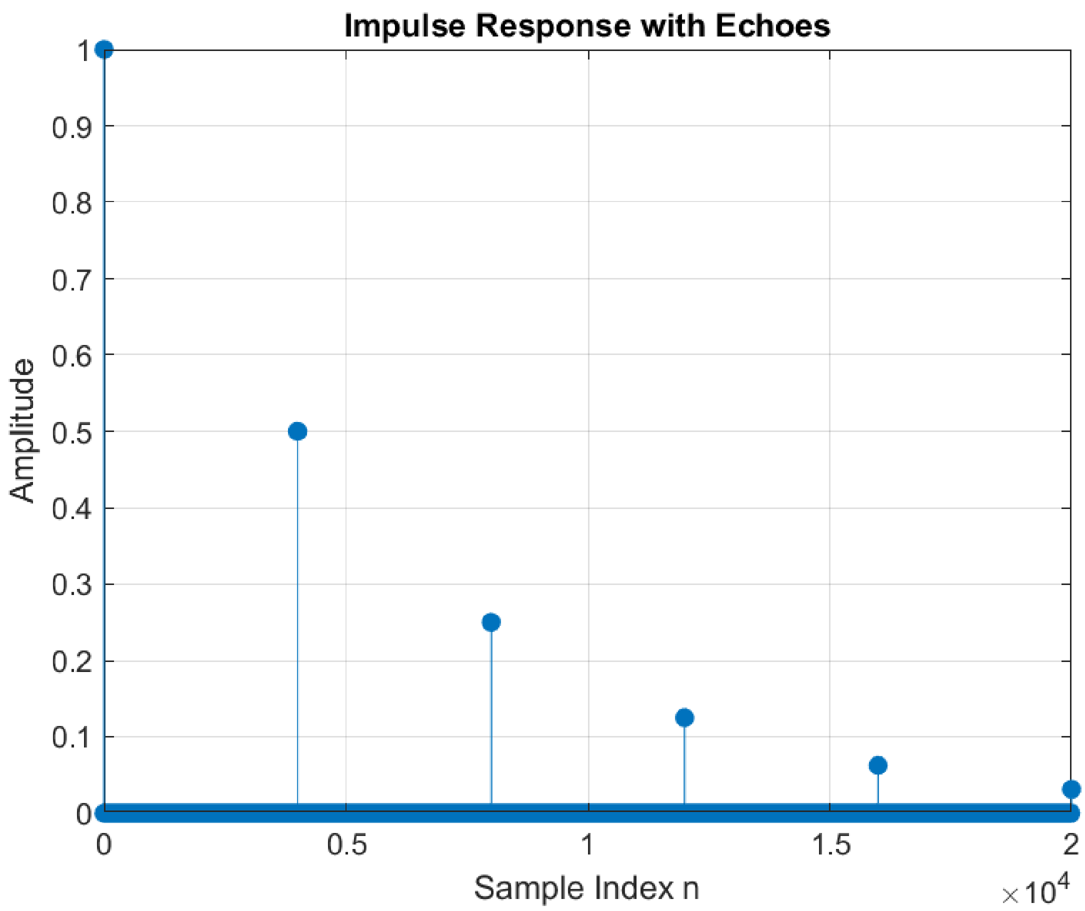
Echo positions (samples): 1, 4001, 8001, 12001, 16001, 20001, 24001, 28001, 32001, 36001

(c) Modify the response function to make each echo half the amplitude of the previous echo.

Octave code:

```
h = zeros(1, N);
for k = 0:floor(N / echo_samples) - 1
    index = 1 + k * echo_samples;
    h(index) = (0.5)^k;
end

figure;
stem(0:N-1, h, 'filled');
xlabel('Sample Index n');
ylabel('Amplitude');
title('Impulse Response with Echoes');
grid on;
xlim([0 20000]);
print('impulse_response_echoes.png', '-dpng');
```



(d) Test your DSP echo on a sine-tone that is 0.1 seconds long.

We generate a 440 Hz sine tone for 0.1 seconds and convolve it with the impulse response to apply the echo.

Octave code:

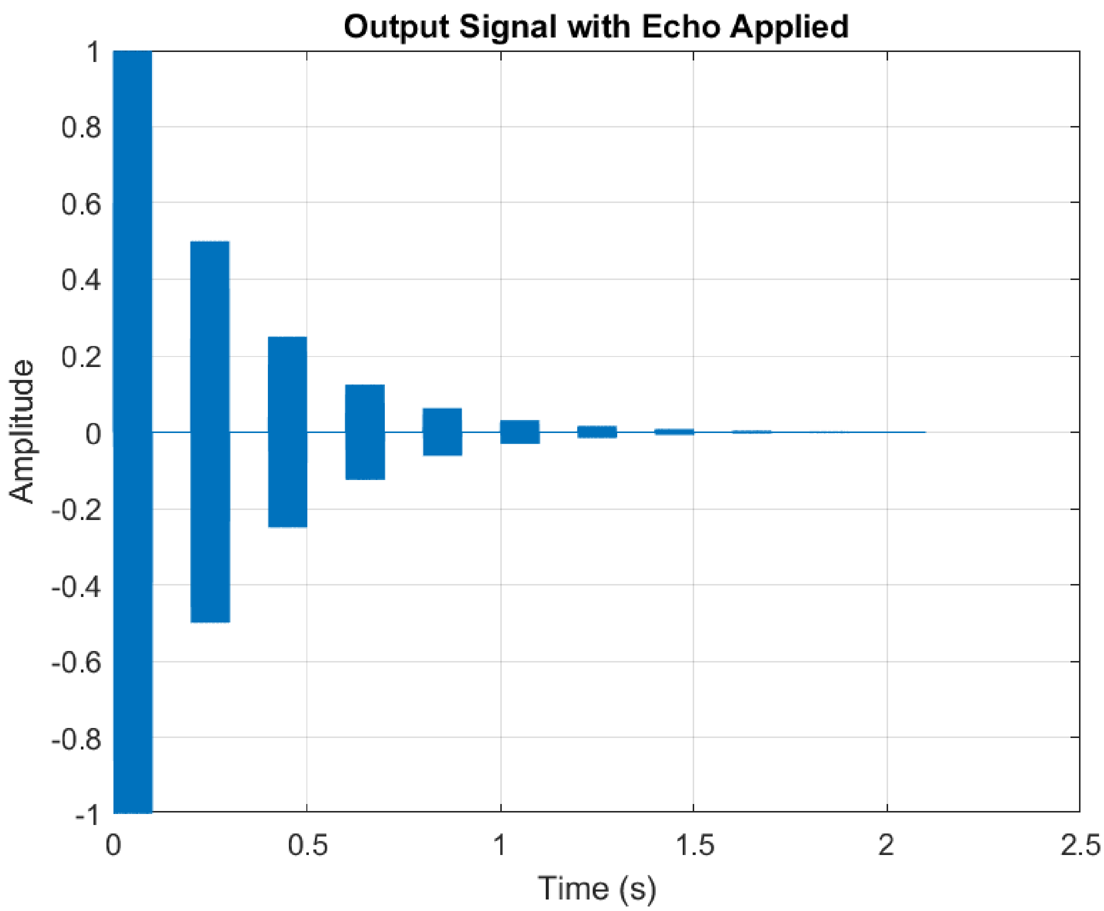
```
tone_duration = 0.1;
```

```
tone_N = round(fs * tone_duration);
f0 = 440;

t = (0:tone_N-1)/fs;
x = sin(2 * pi * f0 * t);

y = conv(x, h);

figure;
plot((0:length(y)-1)/fs, y);
xlabel('Time (s)');
ylabel('Amplitude');
title('Output Signal with Echo Applied');
grid on;
print('echoed_sinetone.png', '-dpng');
```



Problem 2: Impulse response of a band-pass filter

Let $l[n]$ and $h[n]$ be the impulse responses of single-pole low-pass and high-pass filters with the same cutoff frequency f_c, respectively.

(a) Show that when an input signal $s[n]$ is split into two copies and sent to $l[n]$ and $h[n]$ in parallel, the sum of the outputs is still $s[n]$.

Let the outputs of each path be:

$$y_l[n] = l[n] * s[n], \quad y_h[n] = h[n] * s[n]$$

If the impulse responses satisfy:

$$l[n] + h[n] = \delta[n]$$

Then their outputs will sum to:

$$y_l[n] + y_h[n] = (l[n] + h[n]) * s[n] = \delta[n] * s[n] = s[n]$$

Thus, the combined output is equal to the input $s[n]$, proving the result.

(b) Show that the result in (a) implies $h[n] = \delta[n] - l[n]$.

From part (a), we know:

$$l[n] + h[n] = \delta[n]$$

Rearranging the equation:

$$h[n] = \delta[n] - l[n]$$

This expresses the high-pass filter as the difference between an identity system and the low-pass filter.

(c) Now assume the cutoff frequencies are different for $h[n]$ and $l[n]$. If the filters act in series, the result is a band-pass filter if:

- A: the cutoff frequency of $l[n]$ is lower than that of $h[n]$
- B: the cutoff frequency of $h[n]$ is lower than that of $l[n]$
- C: the cutoff frequency of $l[n]$ is equal to that of $h[n]$
- D: both cutoff frequencies are equal to one-half the sampling frequency

To create a band-pass filter, the low-pass filter must eliminate frequencies above a certain value, and the high-pass filter must eliminate frequencies below a certain value. Therefore, the signal that passes through both filters lies in between their cutoff frequencies.

This means:

$$f_c^{(l)} < f < f_c^{(h)}$$

So the correct answer is:

A: the cutoff frequency of $l[n]$ is lower than that of $h[n]$

3. Discrete Fourier Transform, Filtering, and Noise

Problem 1: Discrete Fourier Transform properties

(a) Knowing that the DFT is a complex sum:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j\frac{2\pi}{N}kn}$$

we want to prove that the DFT is additive and homogeneous, which together show that it is a linear operator.

Let x_n and z_n be two time-domain sequences, and a and b be two scalars. Define a new sequence:

$$y_n = ax_n + bz_n$$

Now apply the DFT to y_n:

$$\begin{aligned} Y_k = \sum_{n=0}^{N-1} y_n \cdot e^{-j\frac{2\pi}{N}kn} &= \sum_{n=0}^{N-1} (ax_n + bz_n) \cdot e^{-j\frac{2\pi}{N}kn} \\ &= a\sum_{n=0}^{N-1} x_n \cdot e^{-j\frac{2\pi}{N}kn} + b\sum_{n=0}^{N-1} z_n \cdot e^{-j\frac{2\pi}{N}kn} \\ &= aX_k + bZ_k \end{aligned}$$

Therefore, the DFT satisfies both:
- Additivity: $\text{DFT}(x_n + z_n) = X_k + Z_k$ - Homogeneity: $\text{DFT}(ax_n) = aX_k$

Conclusion: The DFT is a linear operator because it satisfies both properties of linearity: additivity and homogeneity.

(b) Let $X_k = \delta[k - k_0] \cdot C$, meaning that in the frequency domain, the signal is zero everywhere except at index k_0, where it is equal to a constant C.

Now consider the inverse DFT, given by:

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k \cdot e^{j\frac{2\pi}{N}kn}$$

Substitute the definition of X_k:

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} \delta[k - k_0] \cdot C \cdot e^{j\frac{2\pi}{N}kn}$$

Since the delta function is zero everywhere except at $k = k_0$, the summation reduces to a single term:

$$x_n = \frac{1}{N} \cdot C \cdot e^{j\frac{2\pi}{N}k_0 n}$$

This is a complex sinusoidal signal of frequency k_0, scaled by $\frac{C}{N}$. It has the general form:

$$x_n = A \cdot e^{j\omega_0 n} \quad \text{where} \quad \omega_0 = \frac{2\pi k_0}{N}$$

Conclusion: A single non-zero frequency component in the DFT corresponds to a complex sinusoid in the time domain. This illustrates *sinusoidal fidelity*, the frequency in the time-domain signal is preserved exactly as represented in the frequency domain.

Problem 2: Spectrum of a Square Pulse

(a) Run the code `compare_spectra.m` and explain why the magnitude of the Fourier spectrum widens as the pulse width narrows.

This effect is explained by the *uncertainty principle* in signal processing. The principle states that the more a signal is localized in one domain (time or frequency), the more it is spread out in the other domain.

A square pulse with a shorter duration in the time domain contains sharper edges and more rapid transitions. These transitions introduce high-frequency components into the signal, which widens its spectrum in the frequency domain. Conversely, a wider pulse in time has smoother transitions and a more compact frequency spectrum.

In the figure generated by the code, each row corresponds to a different pulse width:

- The right column shows square pulses in the time domain.
- The left column shows the corresponding magnitude of the Fourier spectra.

From top to bottom, the time-domain pulse becomes wider, and the corresponding spectrum becomes narrower. This visually confirms that the spectrum *widens as the pulse narrows.*

(b) Measure the width of the time-domain signals and the Fourier spectra in a consistent fashion, and show that the product of the time-domain width and Fourier-domain width is approximately constant.

The code uses pulse widths of:

$$T = 0.01,\ 0.02,\ 0.04,\ 0.08 \text{ seconds}$$

Approximate corresponding spectral widths (as observed visually from the figure):

$$\Delta f = 100,\ 50,\ 25,\ 12.5 \text{ Hz}$$

Now compute the product of time-domain width and frequency-domain width:

$$\begin{aligned} 0.01 \times 100 &= 1.00 \\ 0.02 \times 50 &= 1.00 \\ 0.04 \times 25 &= 1.00 \\ 0.08 \times 12.5 &= 1.00 \end{aligned}$$

This product remains constant across the examples, which confirms the uncertainty principle. Specifically, the width of the signal in one domain is inversely proportional to its width in the other domain:

$$\text{time width} \times \text{frequency width} = \text{constant}$$

This principle demonstrates that we cannot simultaneously achieve high precision in both time and frequency representation of a signal.