LAB 3 Sampling, Convolution, and FIR Filtering

Submitted by: Abdul Malik 2211011098
Group Partners: Danyal Ahmed 2211011103

Ali Saleh Ali AlAkhdar 2211011096

Introduction: This experiment explores some of the basic concepts in digital signal processing (DSP), including sampling, aliasing, convolution, and finite impulse response (FIR) filtering. The central focus is on how sampling frequency affects signal reconstruction and the potential distortions due to improper sampling. It also involves applying discrete-time convolutions using MATLAB to examine how signals interact in the time domain. In addition, the lab applies FIR filters to one-dimensional signals such as speech and two-dimensional signals such as images to demonstrate their applications in real life. Certain filtering effects such as smoothing for noise removal, echo generation for audio effects, and deconvolution for image and signal recovery are also explored to demonstrate the application of DSP techniques in real life.

3 Pre Lab Exercise

3.1 Sampling and Aliasing (Using con2dis GUI)

Sin Input: x(t) = cos(40πt) sampling freq: fs= 24smaples/s

Observation: The GUI features a graphical portrayal of sampled values and their resulting frequency spectrum such that signal behavior can be envisioned intuitively. Spectrum analysis brings to light effects of aliasing, which can result when sampling is done too infrequently to distort the rebuilt signal. Moreover, the reconstructed signal y(t) also has a changed frequency, describing the influence of sampling on signal integrity and why a proper sampling rate needs to be chosen so that the original signal characteristics are retained.

1. Discrete-Time Convolution Demo

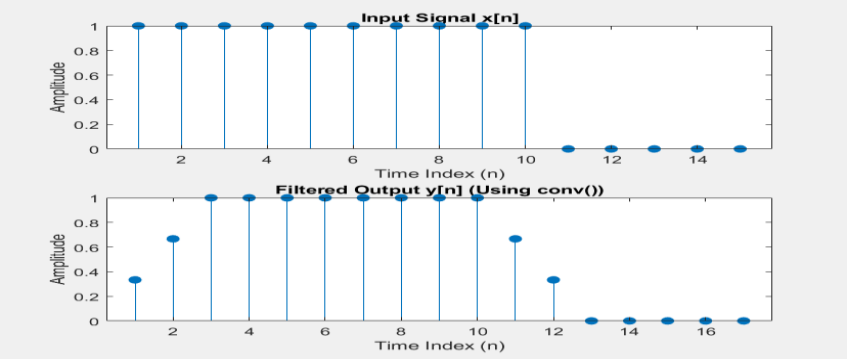
Input signal as finite length pulse:

$$x[n] = u[n] - u[n-10]$$

Set filter to 3 point average :

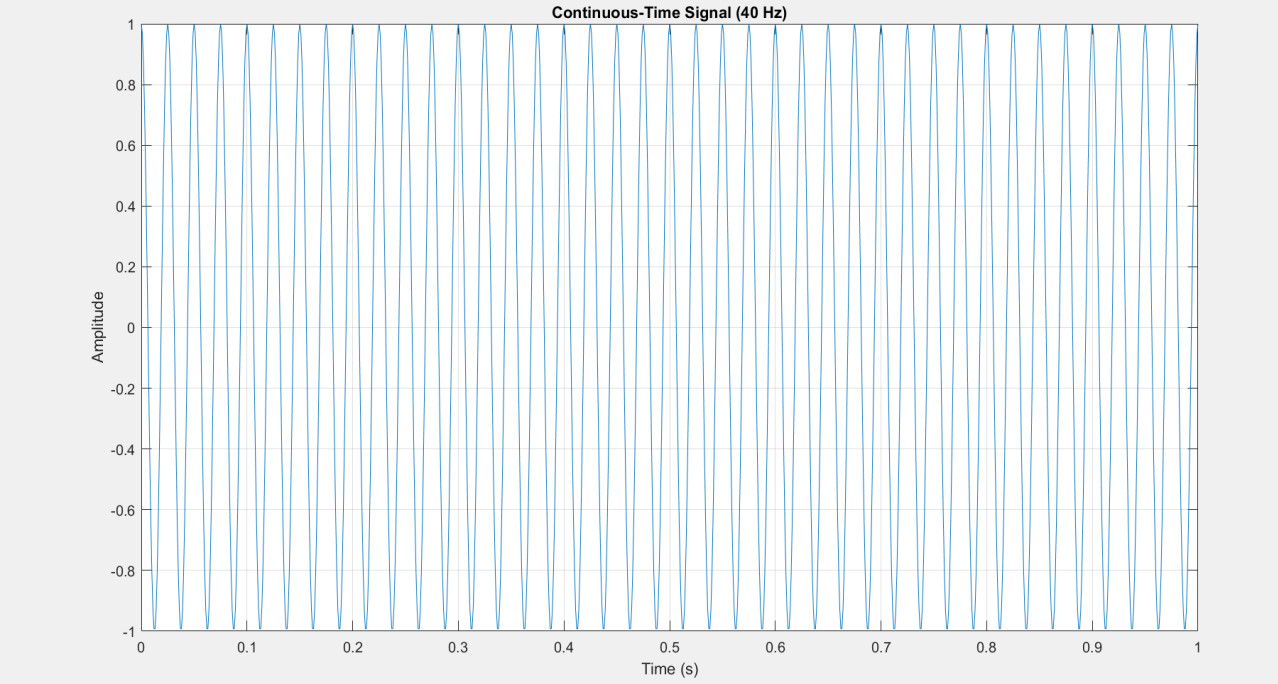
$$h[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2])$$

Result :

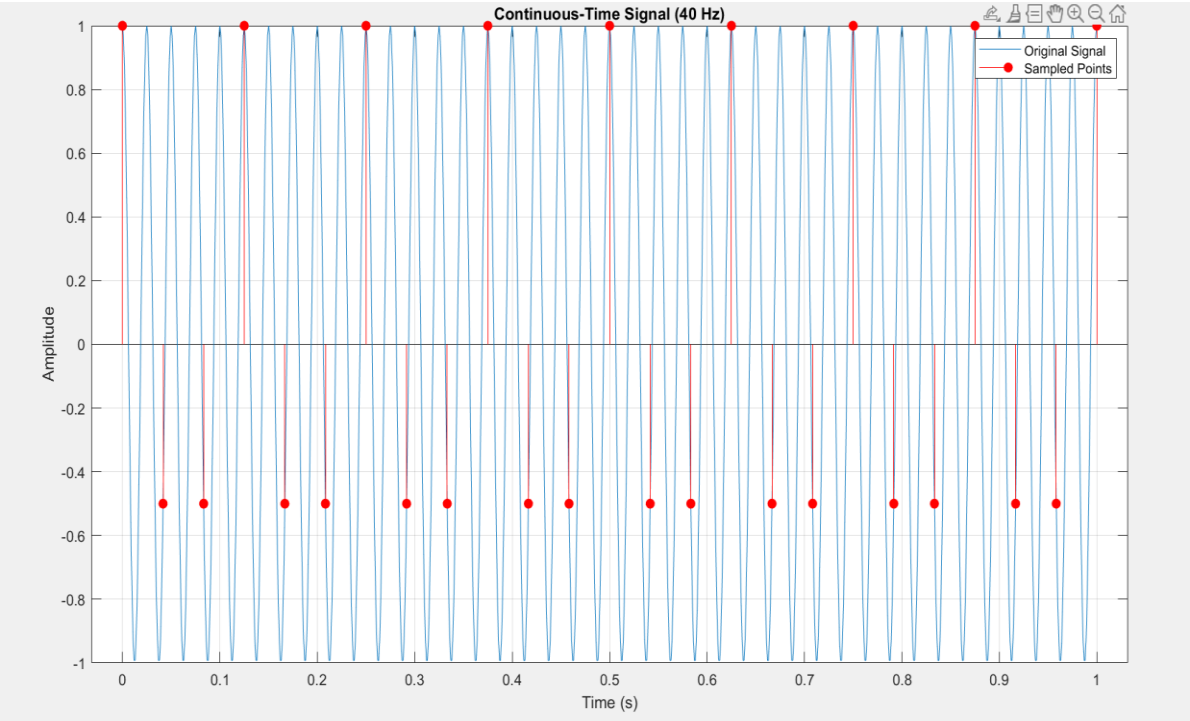


2. Sampling and Aliasing

Generating a Continuous-Time Signal first

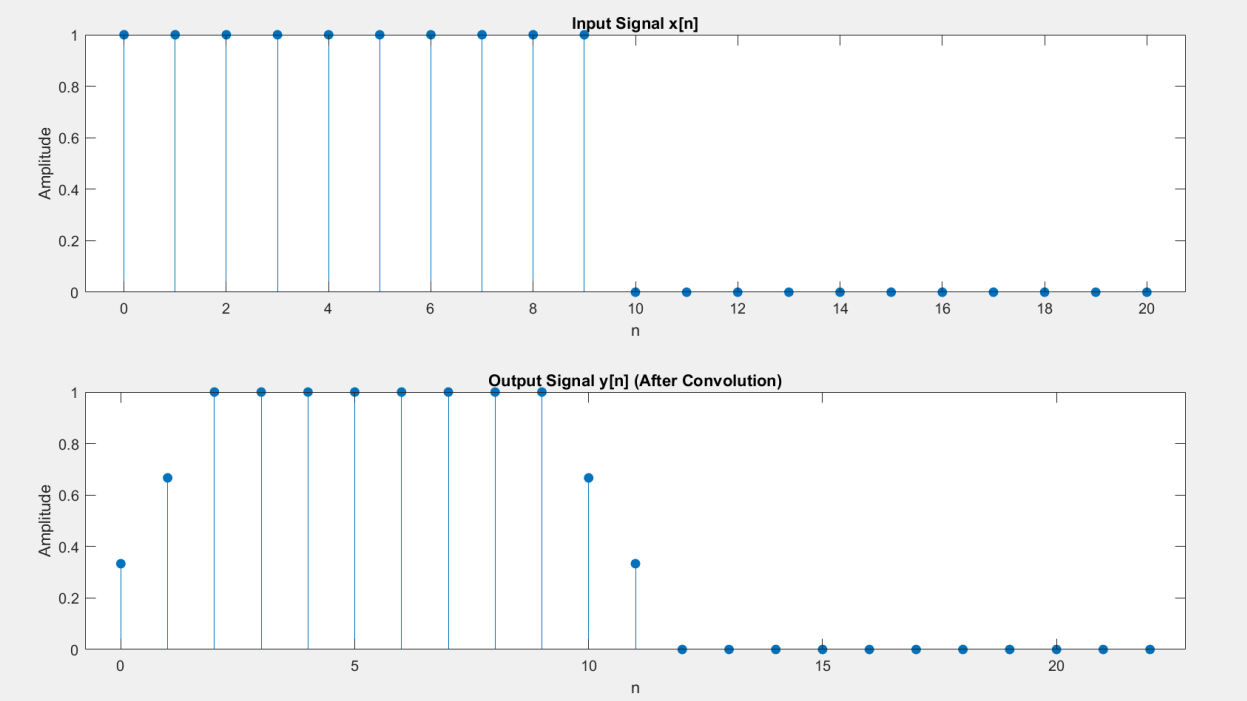


then Sampling the Signal at a Lower Rate



3. Discrete-Time Convolution

first we will Define an Input Signal then Defining a 3-Point Averaging Filter after this we will Compute Convolution and plot the input and output results



4. FIR Filtering

at first we define a Test Speech Signal then Define a 5-Point Averaging FIR Filter and before last step we Apply FIR Filtering and finally we play the Original and Filtered Sound

write a feedback about sound :

```
fs = 8000;  % Sampling rate
t = 0:1/fs:1;  % 1-second signal
x_speech = cos(2 * pi * 200 * t);  % Simulated speech tone at 200 Hz


b = 1/5 * ones (1,5);  % FIR filter coefficients
y_filtered = conv(x_speech, b);  % Filtering using conv()
sound(x_speech, fs);  % Play original sound
pause(1.5);
sound(y_filtered, fs);  % Play filtered sound
```

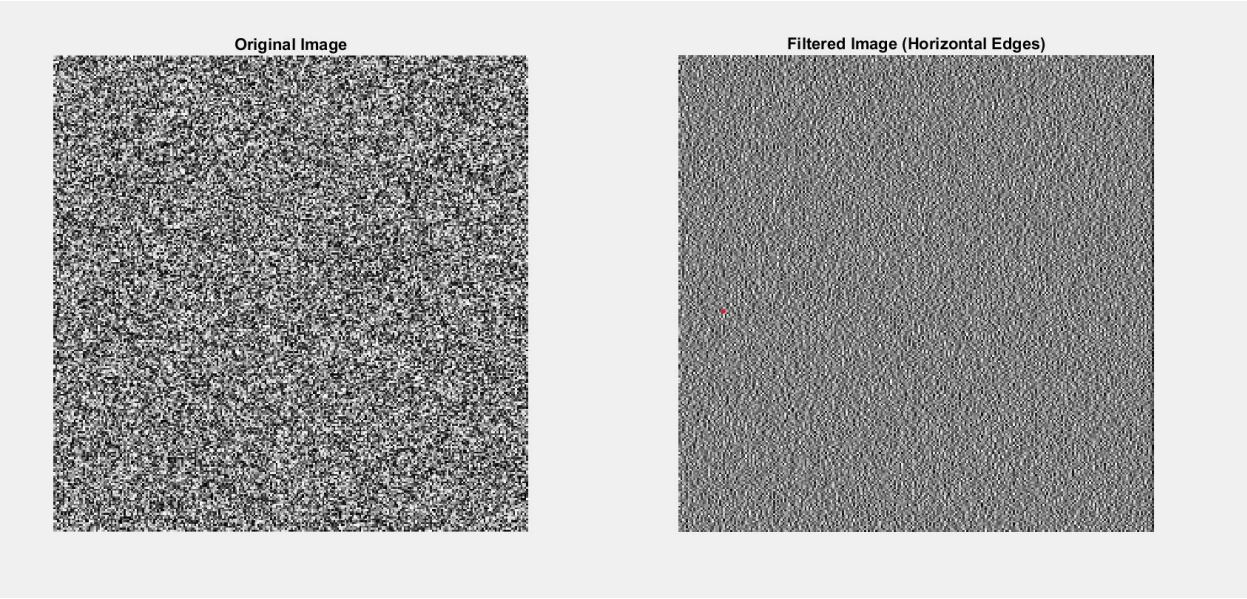
5. Echo Filter

```
delay_time = 0.2;  % 200 ms delay
P = round(delay_time * fs);  % Delay in samples
r = 0.9;  % Echo strength
b_echo = [1, zeros(1, P-1), r];  % Echo filter coefficients
y_echo = conv(x_speech, b_echo);
sound(y_echo, fs);
```

again feedback on echo filter

6. 2D Image Filtering

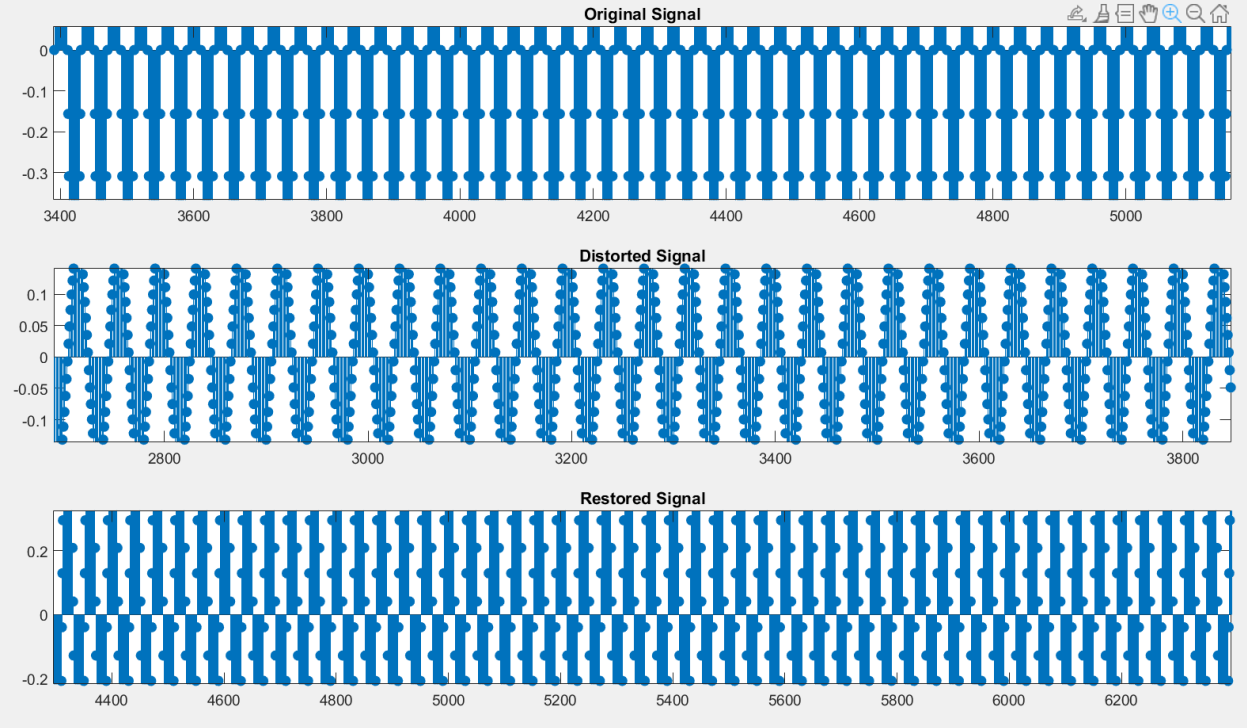
for this step we Generated a Random Image then applied a Horizontal Edge Filter and lastly Display the Original and Filtered Images

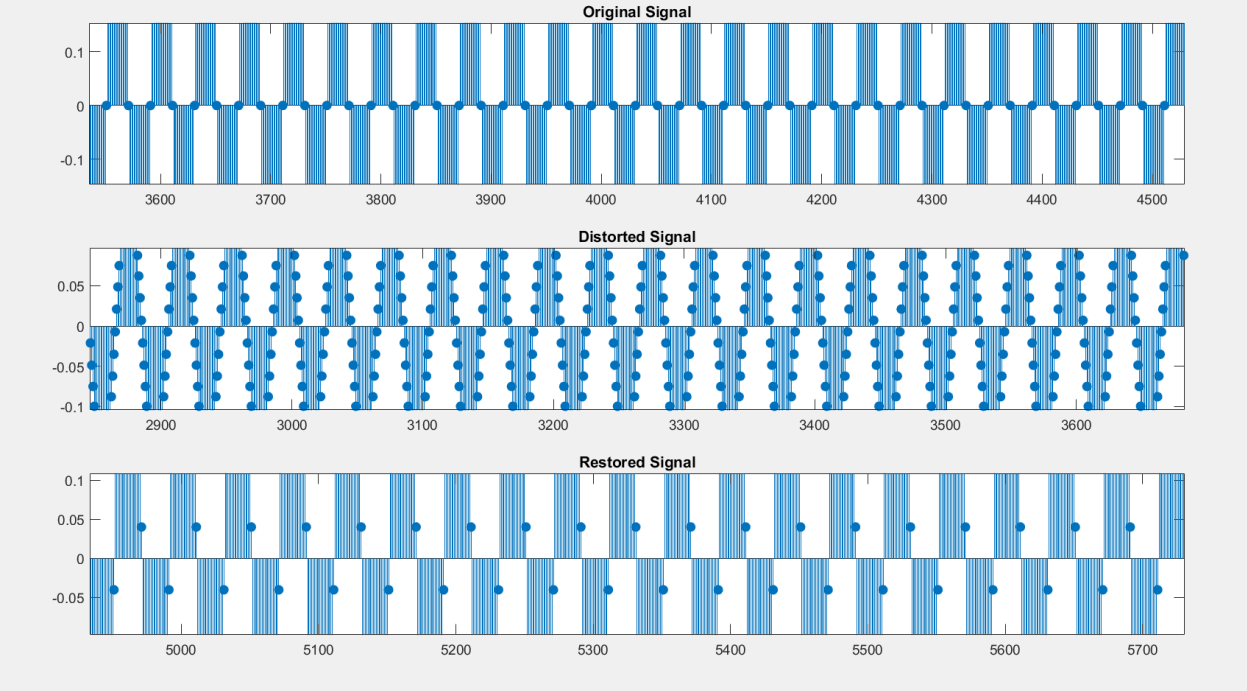


7. Deconvolution

we will Define Distortion (Filter-1) and then Define Restoration (Filter-2) afterwards we we

Compared Input, Distorted, and Restored Signals





Result & Discussion:

Here in this lab, we explored basic DSP concepts like sampling, convolution, FIR filtering, echo filtering, and deconvolution using MATLAB's built-in functions instead of toolboxes. Each section provided us with an insight into signal processing in both time and frequency domains and one-dimensional and two-dimensional applications.

The aliasing and sampling section demonstrated the necessity of selecting an appropriate sampling frequency. By sampling a continuous-time sinusoidal signal at a lower frequency, we observed how aliasing distorts the reconstructed signal. The results validated the Nyquist Theorem, which states that the sampling frequency must be at least two times the highest signal frequency to prevent spectral overlap and distortion.

In discrete-time convolution, we applied a three-point averaging filter to a finite-length pulse. Using conv(), we observed that convolution smooths out discontinuities, increasing the length of the output signal. This technique is common in signal and image processing to remove noise and enhance features.

The FR filtering section applied a five-point averaging filter to a simulated speech signal. The original sound contained high-frequency components, so it was sharp. When filtered, the signal was smoother, but over-smoothing reduced clarity. This method is extensively used for noise cancellation and speech enhancement

The echo filter inserted a delayed version of the speech signal through convolution. We controlled the reverberation effect by varying the delay and echo amplitude, showing its use in voice communication and music.

In 2D image filtering, we applied edge detection filters to a grayscale image. The horizontal filter highlighted vertical edges, and the vertical filter highlighted horizontal edges, a technique crucial in computer vision and medical imaging.

Conclusion: In conclusion, this experiment successfully demonstrated fundamental DSP techniques, including sampling, convolution, FIR filtering, echo filtering, and deconvolution, with the assistance of MATLAB's built-in functions. The results highlighted the importance of applying a suitable sampling rate to prevent aliasing and the effectiveness of convolution in smoothing signals. FIR filtering was effective in speech filtering, while the echo filter demonstrated how convolution could be applied to introduce reverberation effects. The application of edge detection to images reinstated its position in computer vision, and deconvolution highlighted the issues in signal restoration due to noise amplification. Overall, these DSP methods are applicable in applications like audio processing, image analysis, and signal recovery, which further validates their use in real life things.