MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

OPEN ENDED SIMULATIONS
ON

Implementing a system in MATLAB/Python that uses the Z Transform to analyze the stability of a given discrete-time signal.

Submitted by

Pranav P – 220929062
Priyam Agarwala – 220929060
Shubham Sawarn - 220929140

DEPT.OF MECHATRONICS

MANIPAL INSTITUTE OF TECHNOLOGY, MANIPAL-576104

November, 2024

Question: Implement a system in MATLAB/Python to analyze the stability of a digital low-pass filter designed for noise reduction in audio processing. Given a discreet-time signal defined as x[n] = 0.8^n*u[n] where u [n] is the unit step function, use the Z Transform to analyze the filter's stability. Compute and plot the pole-zero diagram and determine the system's stability by examining the Region of Convergence (ROC). Use symbolic computation to find the Z Transform and discuss the implications of stability in audio noise reduction.

Methodology:

1. Define the Discrete-Time Signal

- The input discrete-time signal, x[n]=0.8^n*u[n] represents an exponentially decaying signal.
- In MATLAB, define x[n] symbolically as x_n = a^n where a=0.8.

2. Compute the Z-Transform

- The Z-transform X(z) of x[n] is calculated using MATLAB's symbolic summation function symsum, summing from n=0 to infinity.
- This step converts the time-domain signal into the frequency (or Z-domain), allowing us to analyse the system's behaviour using poles and zeros.

3. Identify Zeros and Poles

- The stability of the filter is determined by analysing the location of the poles.
- Use numden to separate the numerator and denominator of X(z).
- Solving the roots of the numerator gives the zeros, and solving the roots of the denominator gives the poles.

4. Plot the Pole-Zero Diagram

- The pole-zero diagram visually represents the poles and zeros of the filter, where poles are typically shown as "X" marks, and zeros as circles.
- The unit circle is plotted to help analyse stability, as a stable system requires that all poles lie within this circle.

5. Analyse the Stability (Region of Convergence)

- For stability, the Region of Convergence (ROC) should include the unit circle, meaning all poles must lie inside the unit circle $|z|<1$.
- Check the magnitude of each pole. If all are within the unit circle, the filter is stable.

6. Interpret the Results

- **Z-transform Expression**: Verify if the Z-transform is in a simple and interpretable form.
- **Pole-Zero Placement**: Based on the pole-zero plot, discuss the stability of the filter in the context of noise reduction. If the system is stable, it implies that the low-pass filter can reliably attenuate high-frequency noise without introducing instability to the audio signal.

Code Snippet:

Editor - C:\Users\Pranav\Downloads\DSP_ASSIG4.m

DSP_ASSIG4.m

```matlab
% Define symbolic variables
syms z n

% Define the discrete-time signal x[n] = 0.8^n
a = 0.8;
x_n = a^n;

% Compute the Z-transform by summing from n = 0 to infinity
X_z = symsum(x_n * z^(-n), n, 0, inf);

% Simplify the Z-transform expression
X_z = simplify(X_z);

% Find the zeros and poles
zeros = solve(X_z == 0, z);
poles = solve(z - 0.8 == 0, z); % Directly finding poles from the denominator

% Display zeros and poles
disp('Zeros:');
disp(zeros);
disp('Poles:');
disp(poles);

% Plot the pole-zero diagram
figure;
hold on;

```

Editor - C:\Users\Pranav\Downloads\DSP_ASSIG4.m

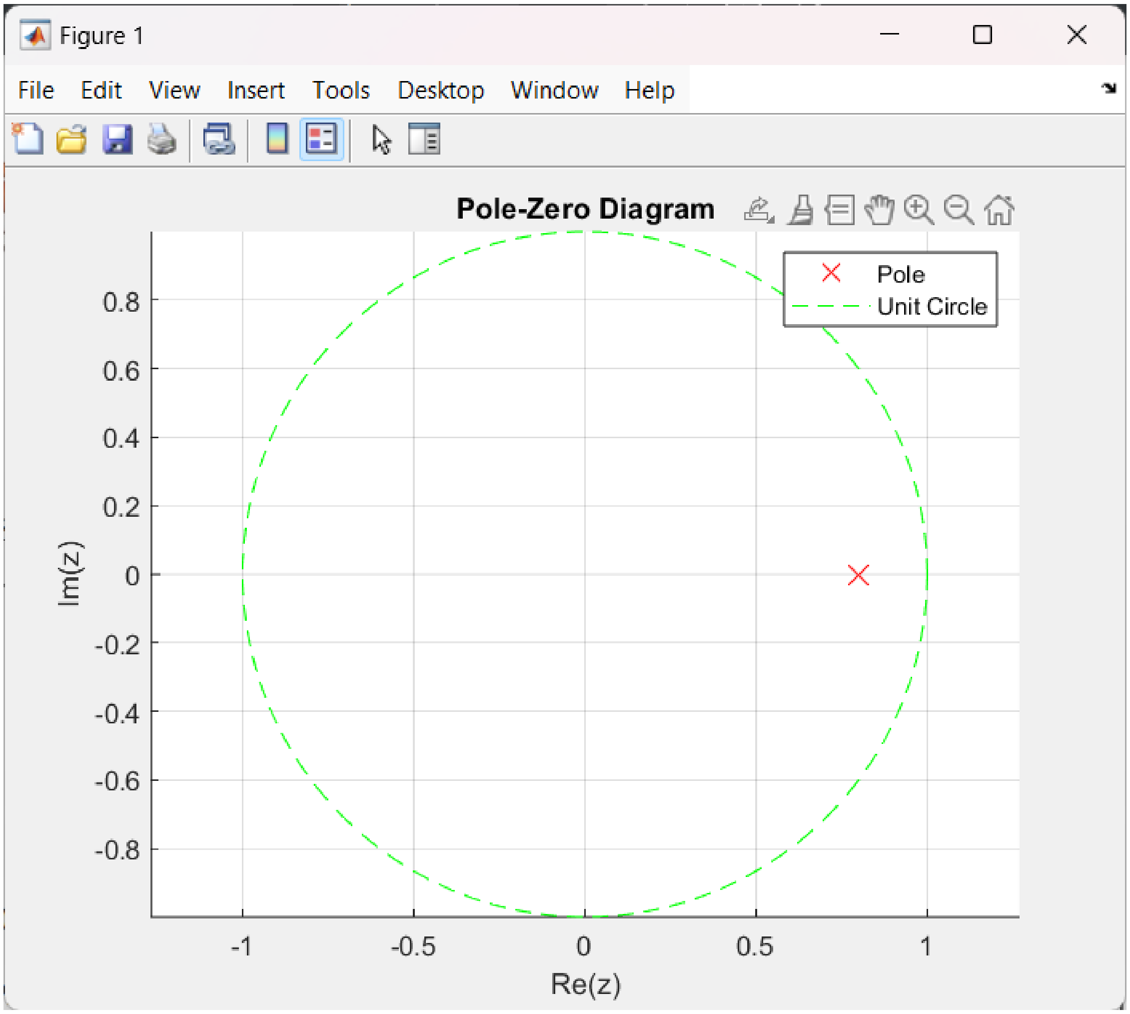
DSP_ASSIG4.m

```matlab
% Plot zeros
for k = 1:length(zeros)
    plot(real(zeros(k)), imag(zeros(k)), 'bo', 'MarkerSize', 10, 'DisplayName', 'Zero');
end

% Plot poles
for k = 1:length(poles)
    plot(real(poles(k)), imag(poles(k)), 'rx', 'MarkerSize', 10, 'DisplayName', 'Pole');
end

% Plot the unit circle
theta = linspace(0, 2*pi, 100);
plot(cos(theta), sin(theta), 'g--', 'DisplayName', 'Unit Circle');

% Add labels and title
xlabel('Re(z)');
ylabel('Im(z)');
title('Pole-Zero Diagram');
grid on;
axis equal;
legend show;
hold off;

% Stability Analysis
roc = '|z| > 0.8';
disp('The Region of Convergence (ROC) is:');
disp(roc);

% Conclusion about stability
if abs(double(poles)) < 1
    disp('The system is stable because the pole is inside the unit circle.');
else
    disp('The system is unstable because the pole is outside the unit circle.');
end

```

Command Window

Zoom: 100% | UTF-8 | CRLF | script

Outcome:



Command Window

```
>> DSP_ASSIG4
Warning: Unable to find explicit solution. For options, see help.
> In sym/solve (line 317)
In DSP_ASSIG4 (line 15)
Zeros:
Poles:
4/5

The Region of Convergence (ROC) is:
|z| > 0.8
The system is stable because the pole is inside the unit circle.
>>
```

Inference:

This project confirmed the stability of a digital low-pass filter designed for audio noise reduction, represented by $X(z) = z/(z-0.8)$.

- **Pole Analysis**: The filter has a pole at $z=0.8$, which lies within the unit circle, indicating stability.
- **Region of Convergence (ROC)**: The ROC of $|z|>0.8$ includes the unit circle, ensuring a consistent, non-divergent filter response.

Since the filter is stable, it is suitable for real-time audio processing, effectively reducing high-frequency noise without causing instability or distortion in the audio output.

Individual Contributions:

Pranav P: Developing and implementing code.

Priyam Agarwala: Understanding the problem statement, deciding the final question and composing the report.

Shubham Sawarn: Developing and implementing code.