DSP Lab Solved problems

Last updated: 2025-10-20 23:26

LaTeX source here

Contents

1 FIR Filter Design for Signal Restoration

You are given a discrete-time signal $x(n)$ which is a combination of a desired low-frequency signal and an unwanted high-frequency noise component. The signal is defined as:

$$x(n) = \cos(0.1\pi n) + 0.5 sin(0.8\pi n) \quad \text{for} 0 \leq n \leq 100$$

Question is:

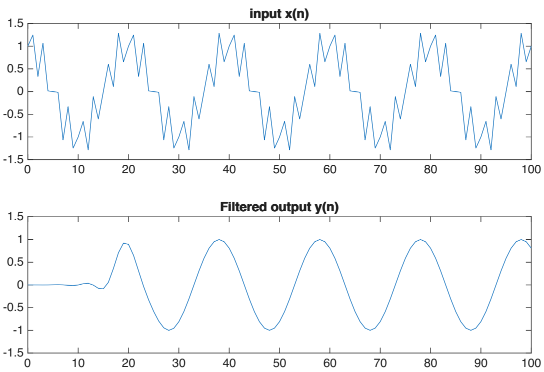
1. Design a digital FIR low-pass filter to remove the high-frequency noise component.
2. Filter the input signal using designed filter

Program

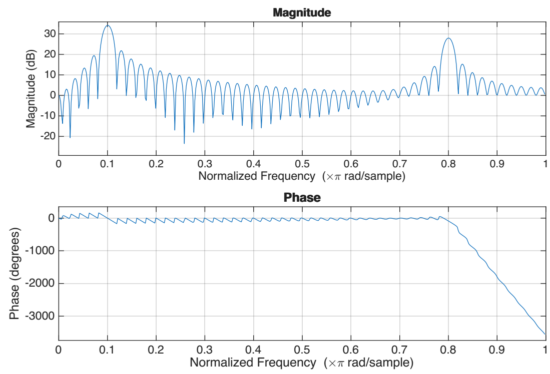
```
clearvars; close all;

% given signal
n = 0:100;
x = cos(0.1 * pi * n) + 0.5*sin(0.8 * pi * n);

%%% design LPF
wp = 0.2*pi;
ws = 0.5*pi;
TW = (ws - wp);
fc = (ws + wp)/2/pi;

% As = 60dB; so use blackman which has As = 74dB
M = ceil(11 * pi / TW);
m = 0:M-1;
m = m - ceil((M-1)/2);
w = blackman(M);

hd = fc * sinc(fc * m);
h = hd .* w'; % required filter
```

```matlab
%%% Filter the input signal
y = filter(h, 1, x);

tiledlayout(2, 1);
nexttile; plot(n, x); title("input x(n)");
nexttile; plot(n, y); title("Filtered output y(n)");
% plot frequency responses and spectrums
figure; freqz(h);
figure; freqz(x);
figure; freqz(y);
```

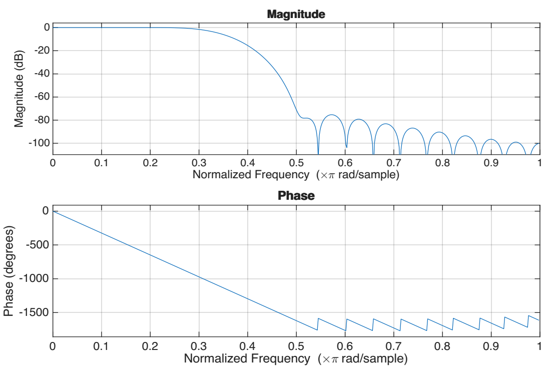
Plotted signal



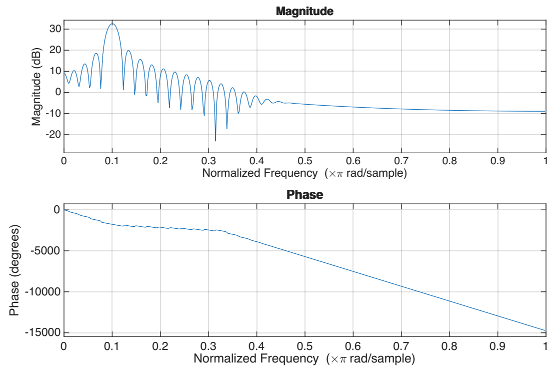
Input message spectrum



Filter's frequency response



Filtered Output spectrum



2 LTI System Stability Analysis and Modification

Consider the LTI system described by the following difference equation:

$$y(n) - 1.2y(n-1) + 0.8y(n-2) = x(n)$$

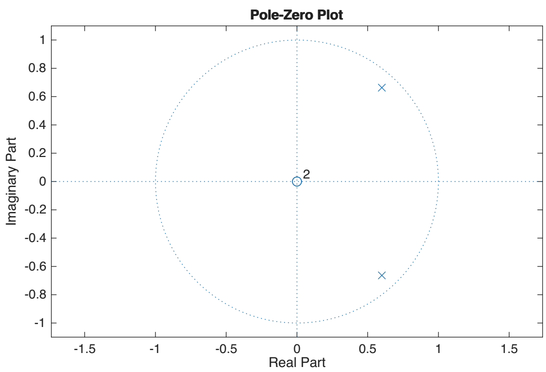
Question:

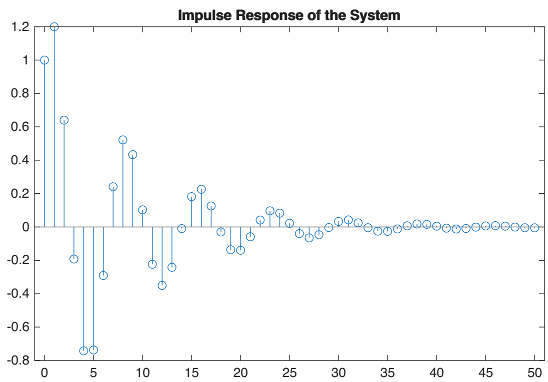
1. Determine if this system is stable or not
2. Plot poles and zeros of the system

3. Compute and plot the impulse response of the system for $0 \leq n \leq 50$
4. Modify the system by changing only the coefficient of y(n-1) (the -1.2 term) so that the poles are located at $p = 0.5 \pm 0.5j$. Plot the pole-zero diagram for new system

Program for Question 1-3:

```
close all; clearvars;
% given system: y(n)  1.2y(n1) + 0.8y(n2) = x(n)
num = 1; den = [1, -1.2, 0.8];

%%% plot zeros and poles
zplane(num, den);

%%% check if system is stable or not
% by checking if all poles lie inside unit circle
system = filt(num, den);
if (abs(pole(system)) < 1)
    disp("stable");
else
    disp("unstable");
end

%%% compute impulse response
n = 0:50;
impulse = n == 0;
response = filter(num, den, impulse);
figure;
stem(n, response);
title('Impulse Response of the System');
```





Program For Question 4

Given system has 2 zeros at $p = 0$ and 2 poles. To change the poles location to $p = 0.5 \pm 0.5j$, First write it in system function form, change denominator expressions and work backwards:

(here z^2 implies double zero at $p = 0$)

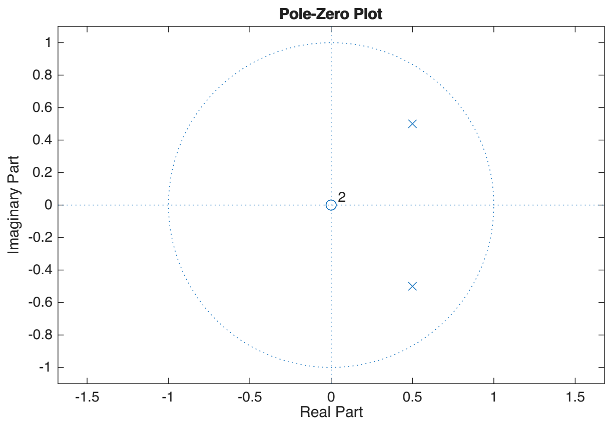
$$
\begin{aligned}
H'(z) &= \frac{z^2}{(z-p_1)(z-p_2)} \\
H'(z) &= \frac{z^2}{[z-(0.5+0.5i)]\,[z-(0.5-0.5i)]} \\
&= \frac{z^2}{z^2 - 1z + 0.5} \\
\frac{Y(z)}{X(z)} &= \frac{1}{1 - 1z^{-1} + 0.5z^{-2}} \\
\Rightarrow X(z) &= Y(z)\left[1 - 1z^{-1} + 0.5z^{-2}\right]
\end{aligned}
$$

Applying inverse Z-transform gives the required system:

$$x(n) = y(n) - y(n-1) + 0.5y(n-2)$$

Hence the implementation:

```
num = 1; den = [1, -1, 0.5];
zplane(num, den);
```



new system function

3 Gibbs oscillation

Simulate Gibbs oscillation on a given signal $s(t)$.

Background

The Fourier series of a complex-valued periodic function $s(t)$, integrable over the interval $[a, b]$on the real line, is defined as:

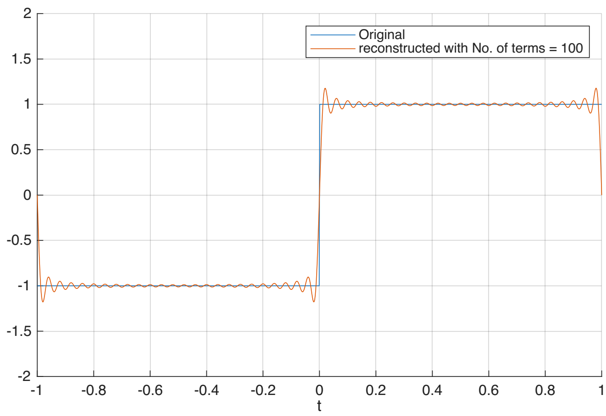
$$s(t) = \sum_{n=-\infty}^{\infty} c_n e^{i2\pi \frac{n}{T} t}$$

Where $T = b - a$ is the period of function. Fourier coefficients c_n are:

$$c_n = \frac{1}{T} \int_a^b s(t)\ e^{-i2\pi \frac{n}{T} t} dt$$

Program

```
clearvars; close all;

a = -1;
b = 1;
t = linspace(a, b, 1000);
T = b - a;

% input signal
s = @(t) sign(t);

% Reconstruct signal with N harmonics on each side
N = 50;
f_series = zeros(size(t));

% coefficients
c = @(k) 1 / T * integral(@(t) s(t) .* exp(-2j * pi * t * k / T), a, b);

% Fourier series summation
for k = -N:N
    f_series = f_series + c(k) * exp(2j * pi * k * t / T);
end

% Plot the actual and reconstructed signal
hold on;
plot(t, s(t));
plot(t, real(f_series));
xlabel('t');
legend(["Original", "reconstructed with No. of terms = " + N*2]);
grid on;
ylim([-2, 2]);
```



Notice the wide oscillation near discontinuities