
DIGITAL SIGNAL PROCESSING

TASK 4



Submitted by:

Anupam Kshetri
33471

Submitted to:

Prof. Vivien Glönkler

TASK 4

The task was done to calculate the $G(s)$ and state $G(s)$ including the computation of the bilinear transformation of a first-order filter using RC-high pass as a filter.

QUESTIONS

In a book publication office, there is a fire alarm systems that detect the occurrence of an event that may result to a fire. It receives a signal from a fire sensor which has a RC High Pass Filter. It detects the room temperature and only sends the signal of the frequency higher than the certain temperature that is likely to cause fire and the system then activates the alarms and sprinklers in the office.

1. Use the appropriate values for resistance and capacitance of RC High pass filter to calculate $G(s)$ and state $G(s)$ in the task.
2. Plot all the response including magnitude, phase and analog frequency.
3. Plot the 3D graphics of poles and zeroes of the RC high pass filter without using fvtool.

HINTS

1. Use the value of Resistance (R) = 100 and Capacitance (C) = 0.000008 to find out the value of Time Constant (τ).
2. Use the formula to find the analog frequency response.
3. Find the bilinear transformation without using prewarping.

SAMPLE CODE

```
R = 100
C = 0.000008
Tau = R * C % Time Constant

fga = 1 / (2 * pi * Tau); % fga = 1/2*pi*R*C analog
cut off frequency
fs = 600; %Sampling frequency
Ts = 1/fs;
fa = 0: .01: (fs/2);

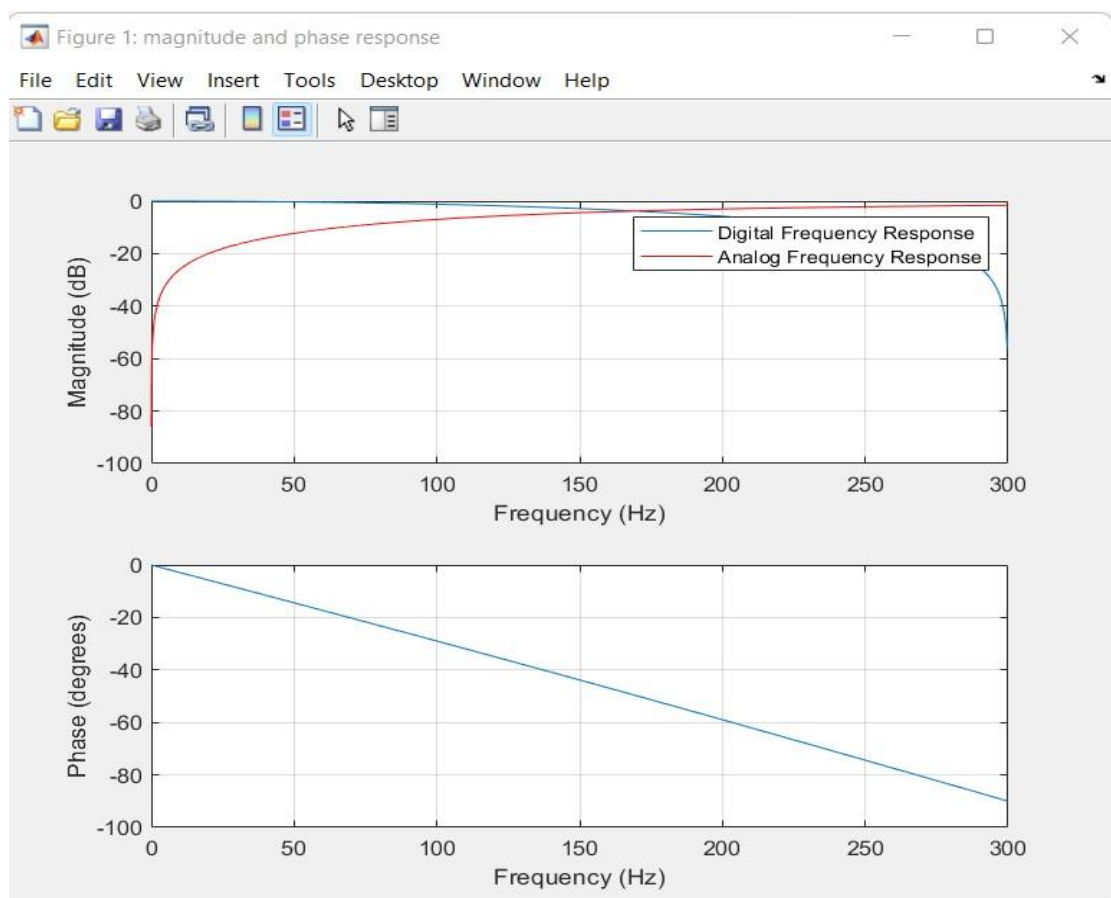
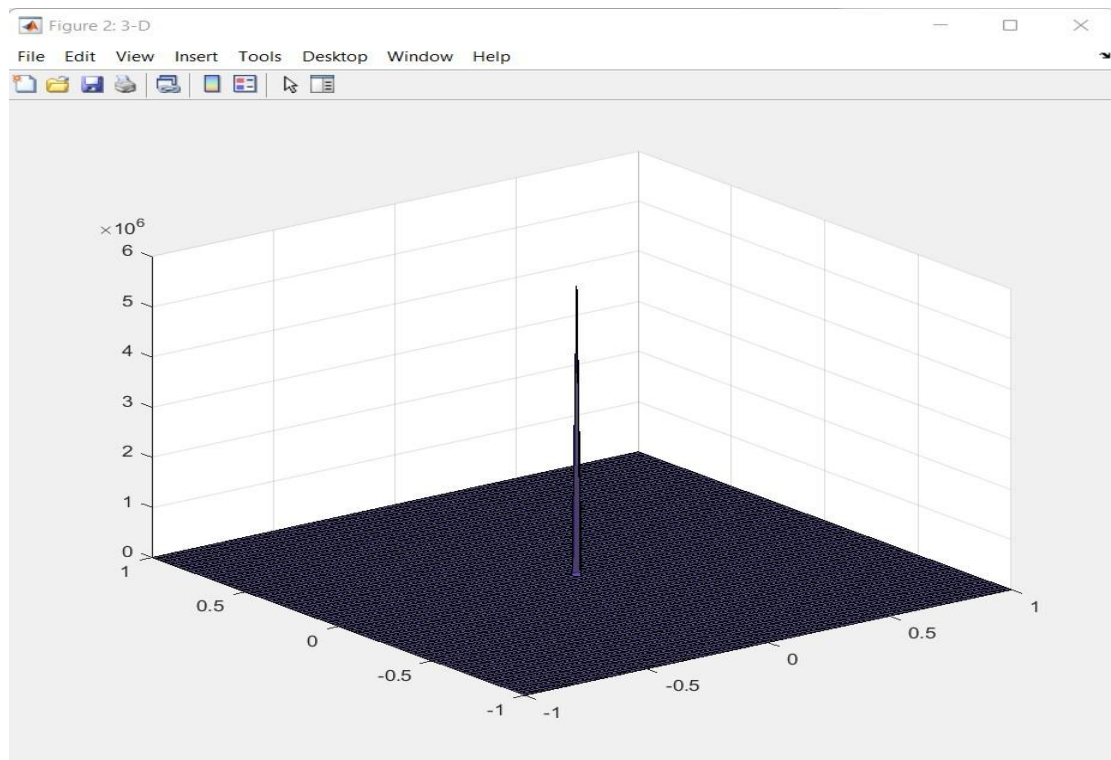
z1 = 1i * 2 * pi * fa;
Gs = Tau * z1 ./ (1 + z1 * Tau)
Gs_db = 20 * log10(abs(Gs)); %analog frequency
response in db

[b, a] = bilinear([1], [Tau 1], fs); % bilinear
transform

[z, p, k] = tf2zpk(b, a);
%%
figure("Name", "magnitude and phase response")
subplot(2, 1, 1);
hold on;
freqz(b, a, 1024, fs); %digital frequency response
plot(fa, Gs_db, 'r'); %analog frequency response
legend('Digital Frequency Response', 'Analog
Frequency Response');
hold off;

%%
%3d plotting
x = -1:0.02:1 ; % x axis
y = -1:0.02:1 ; % y axis
[X, Y] = meshgrid(x, y) ;
s1 = X + j.*Y ;
H1 = (s1 - z) .* (s1 - z) ./ ((s1 - p) .* (s1 - p)) ; %
function
figure("Name", "3-D") ;
s3 = surf(X, Y, abs(H1), 'FaceAlpha', .75); % 3D graph
zeros and poles
camlight('headlight') ;
```

GRAPHS



The graph from the sample solution.

Reference:

- [1][Implement first-order filter - Simulink - MathWorks Deutschland](#)
- [2][Bilinear transformation method for analog-to-digital filter conversion - MATLAB bilinear - MathWorks Deutschland](#)