ECE310 – Project 2

Jonathan Lam

October 22, 2020

Project description

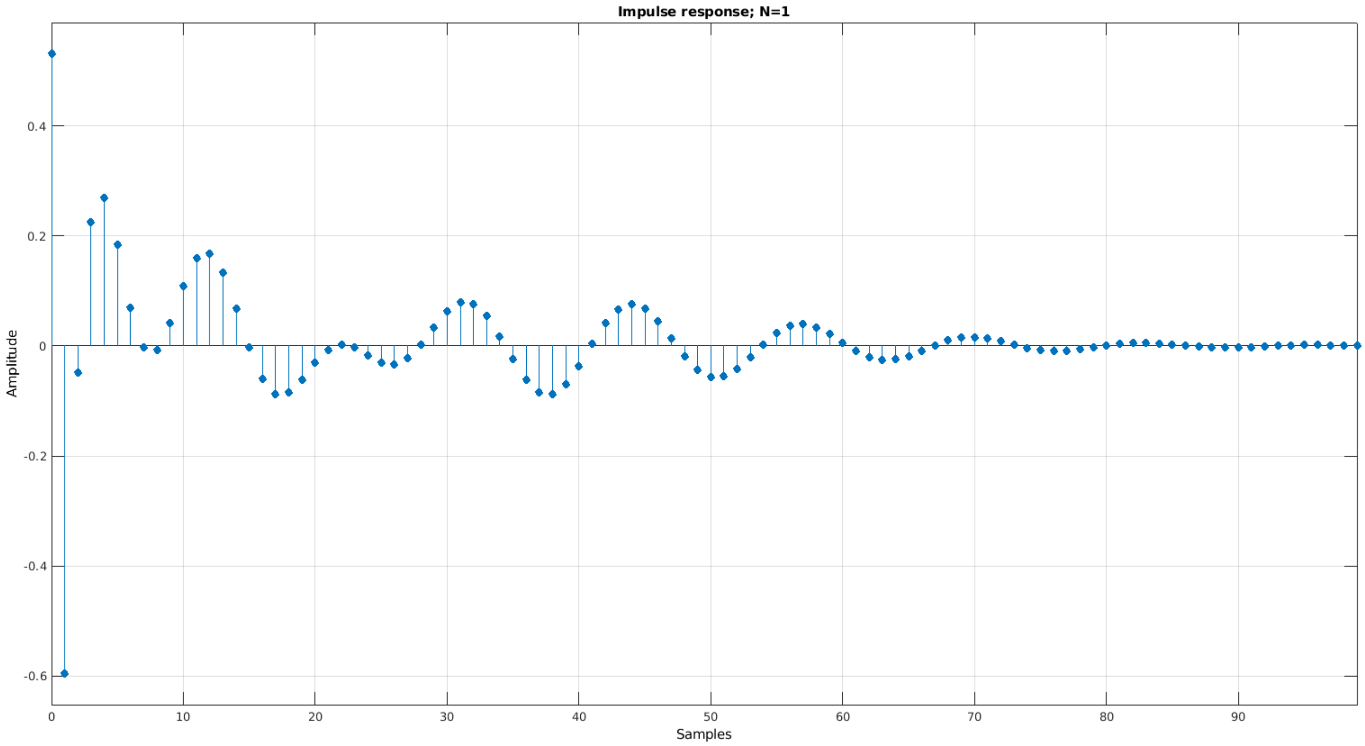
Explore MATLAB's `dfilt` package, cascading filters, different filter implementations (DF1, DF1SOS, DF2SOS, DF2TSOS).

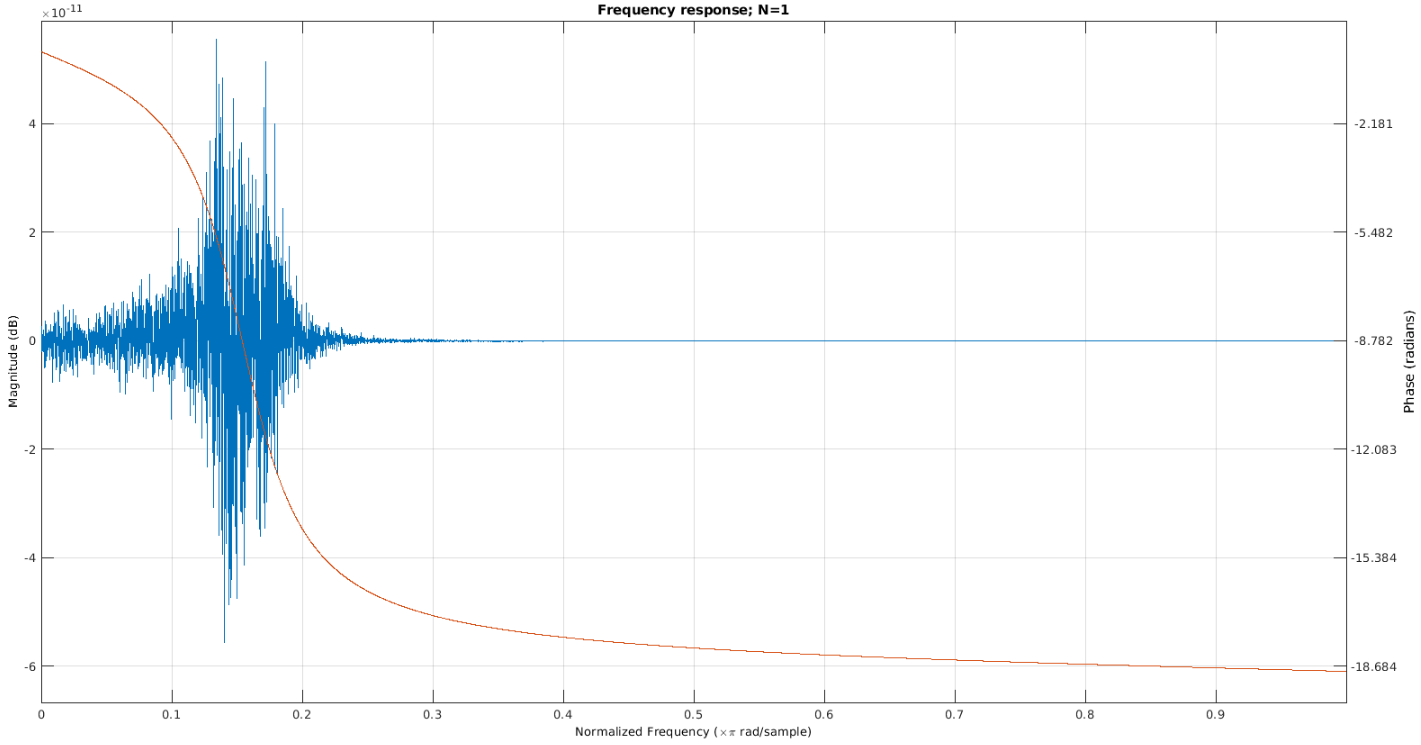
Notes/Answers to questions

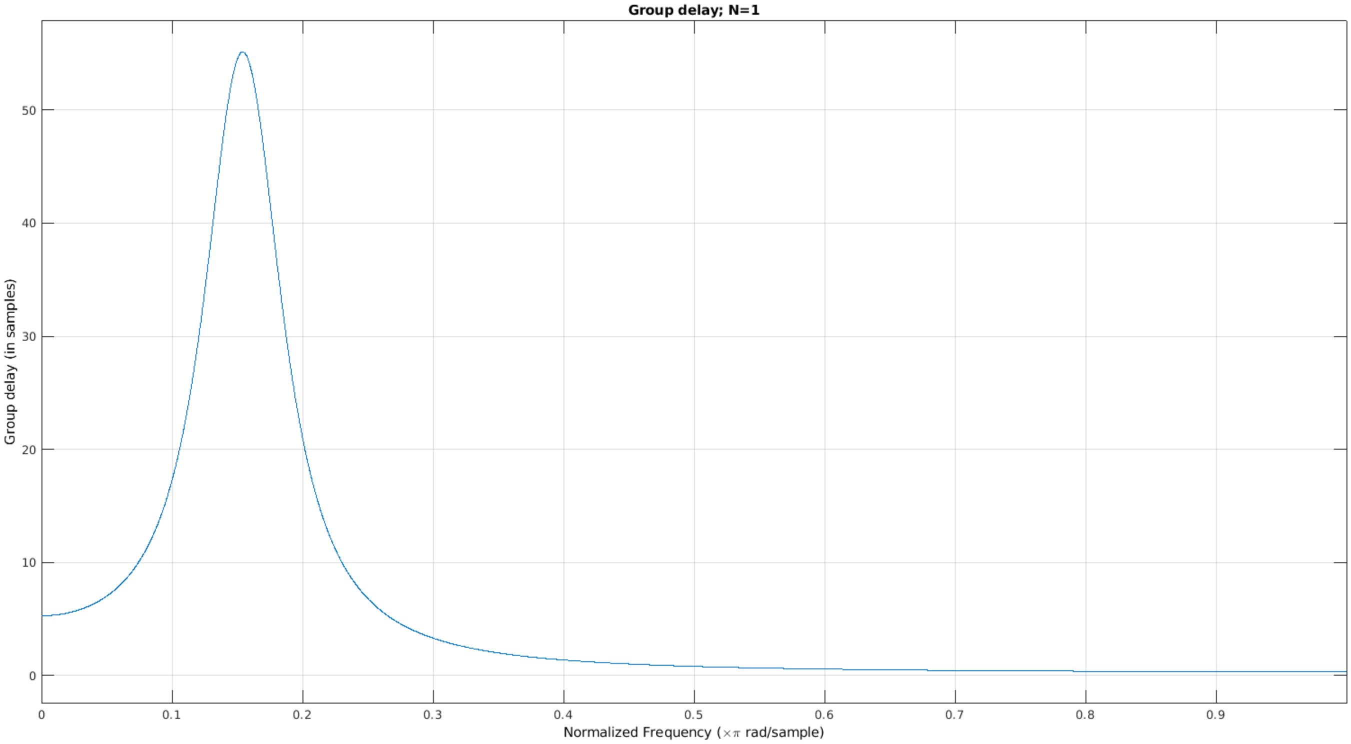
- With $N = 1$, the audio does sound roughly like the original, confirming that an all-pass filter with some phase distortion sounds like the original signal.
- When attempting generating the fiftieth-order transfer function polynomials using convolution of the first-order polynomials and plugging those into the dfilt objects, the resulting functions were very bad (as expected – bad numerical stability).
- The graphs and resulting audio produced by all of the implementations for $N = 50$ roughly all look and sound the same. The filtered audio sounds warbled/alienish, but roughly the same volume. Looking at the group delay plot, the audio at ≈ 0.15rad (263Hz) is now greatly delayed; since this falls within the range of human speech, it makes sense that the text sounds jumbled.

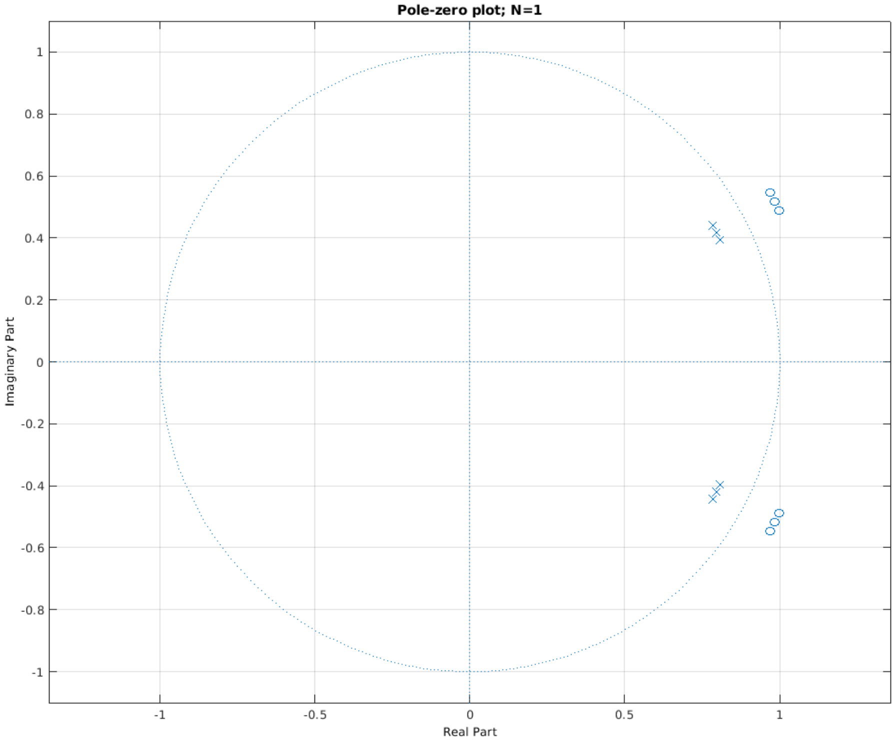
Figures

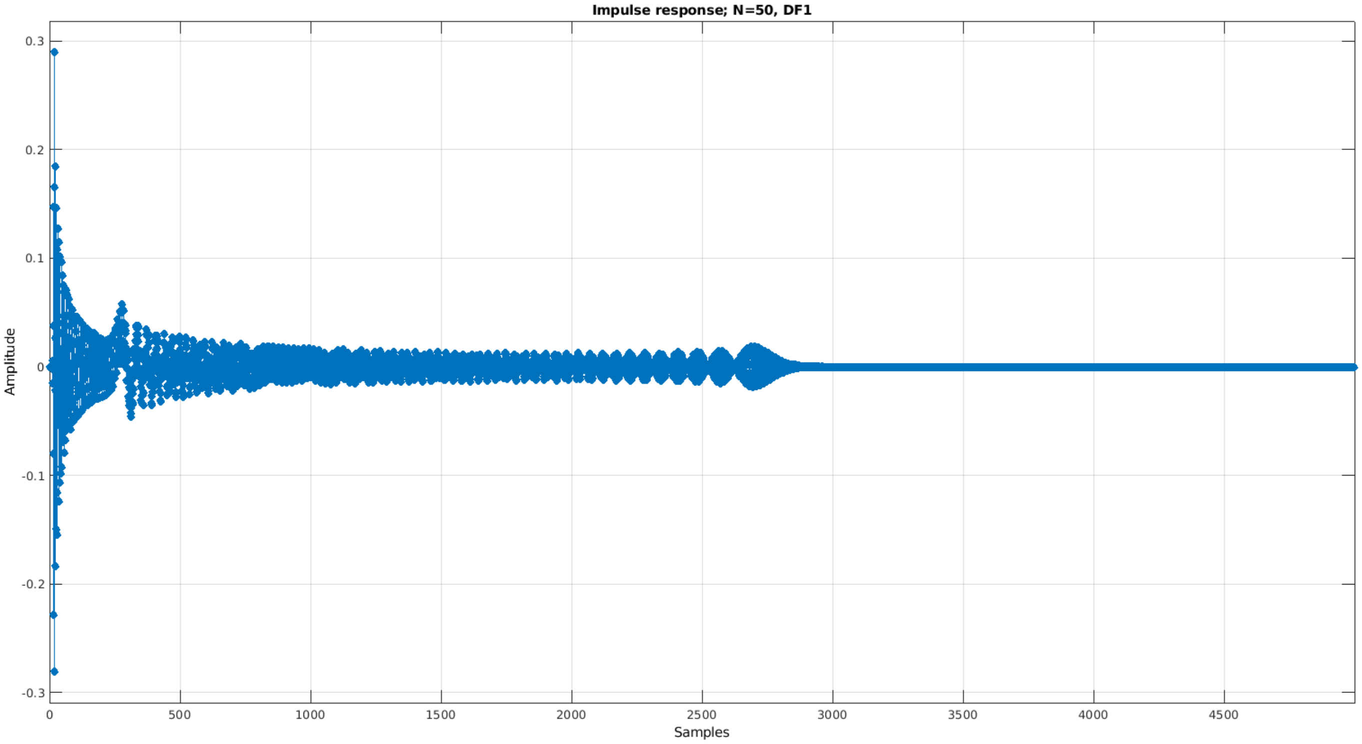
Figures are shown on the next page.

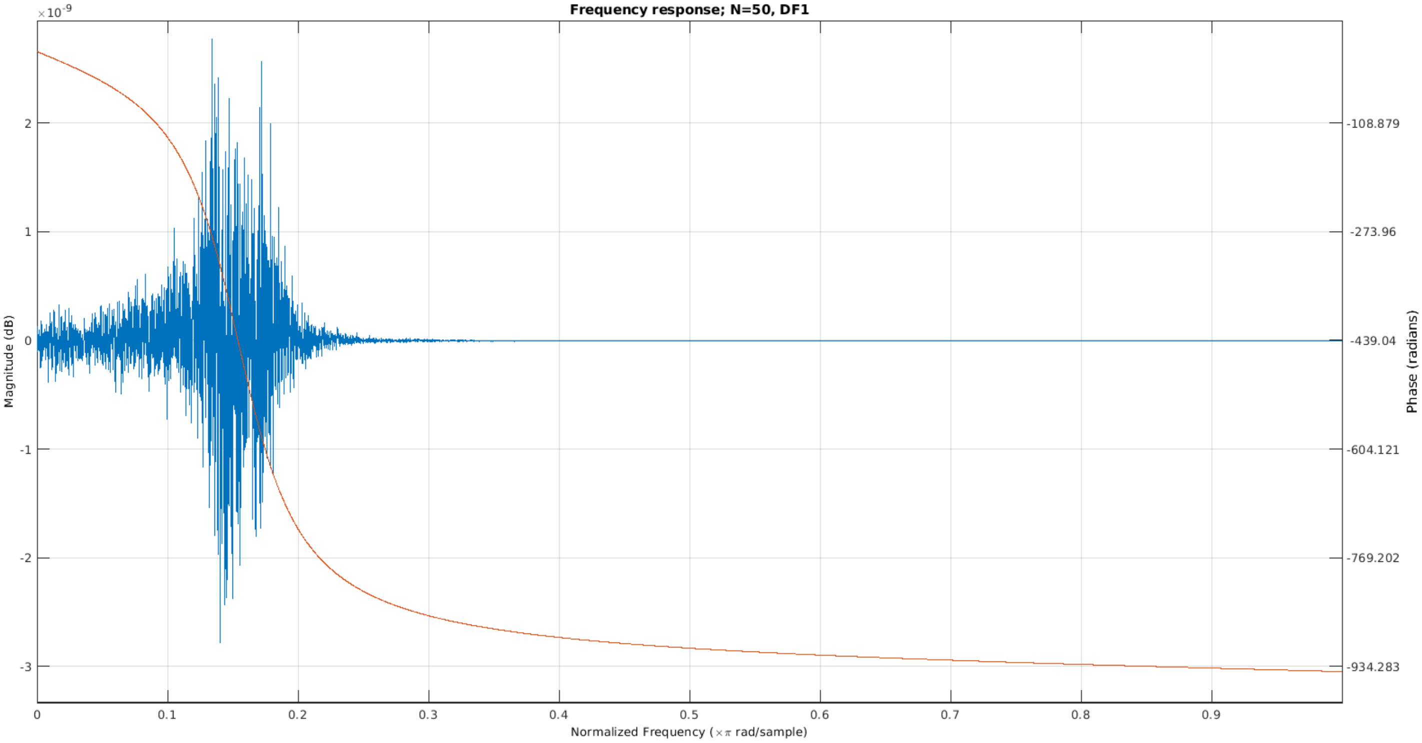

Impulse response; N=1
Amplitude
Samples
0.4
0.2
0
-0.2
-0.4
-0.6
0
10
20
30
40
50
60
70
80
90

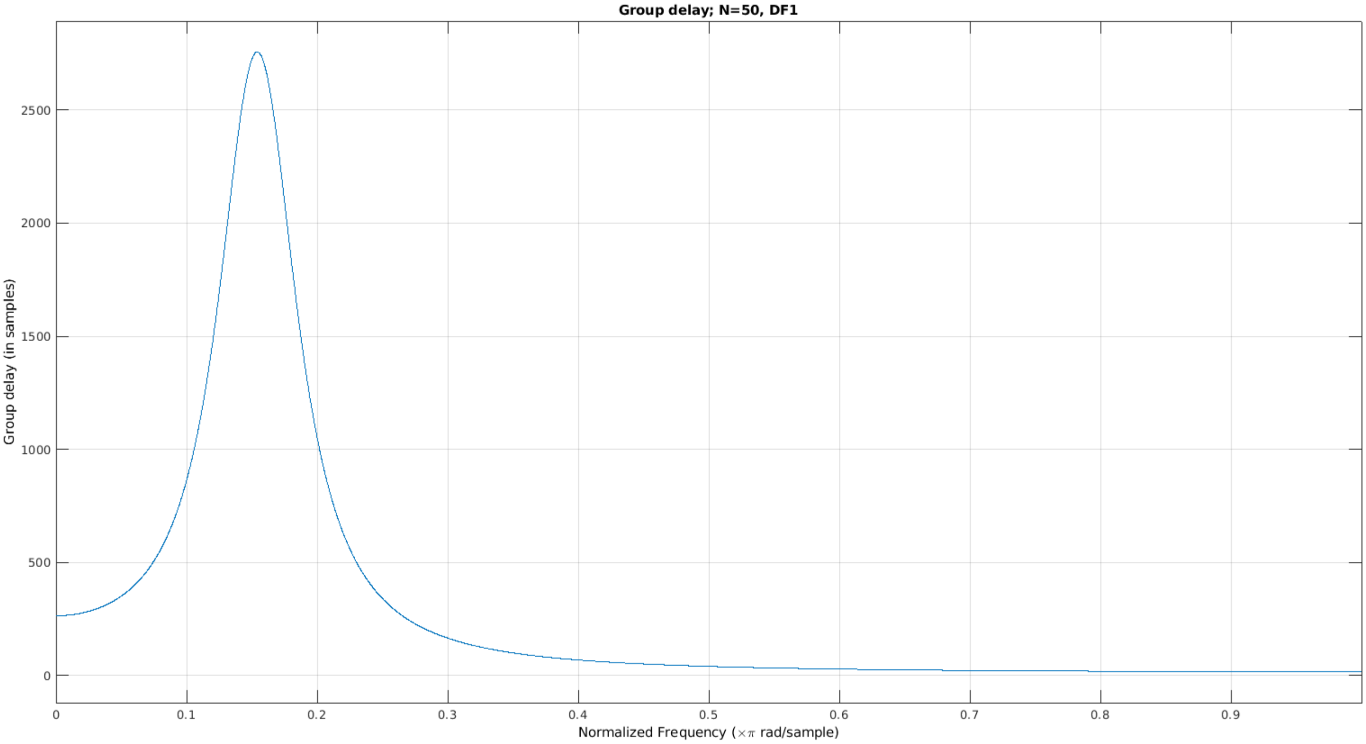

Frequency response; N=1
$\times 10^{-11}$
Magnitude (dB)
Phase (radians)
Normalized Frequency (×π rad/sample)
4
2
0
-2
-4
-6
-2.181
-5.482
-8.782
-12.083
-15.384
-18.684
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

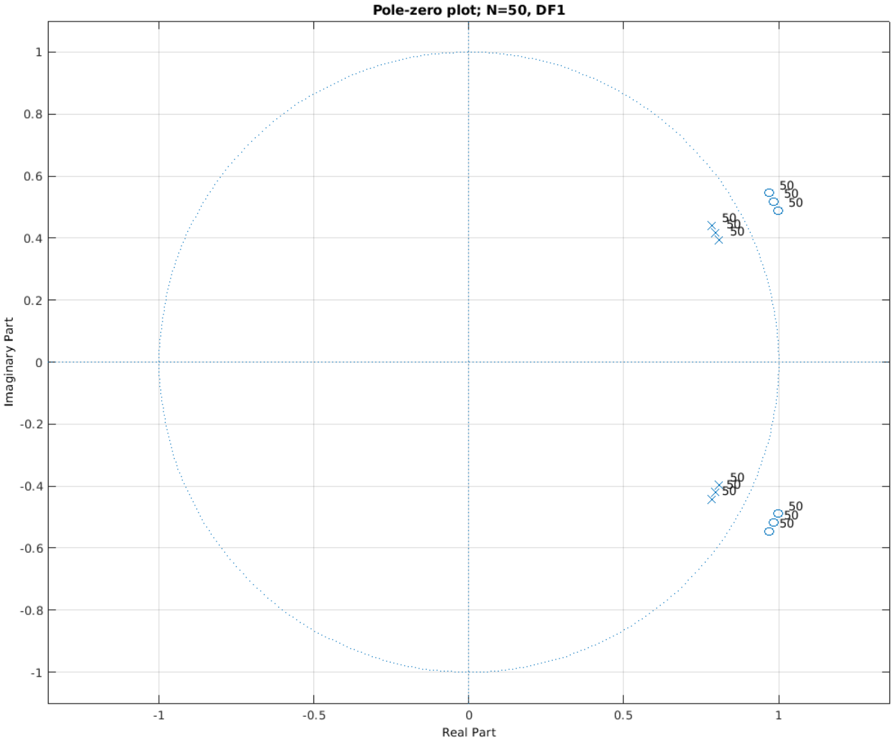

Group delay; N=1
Group delay (in samples)
0
10
20
30
40
50
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Normalized Frequency (×π rad/sample)

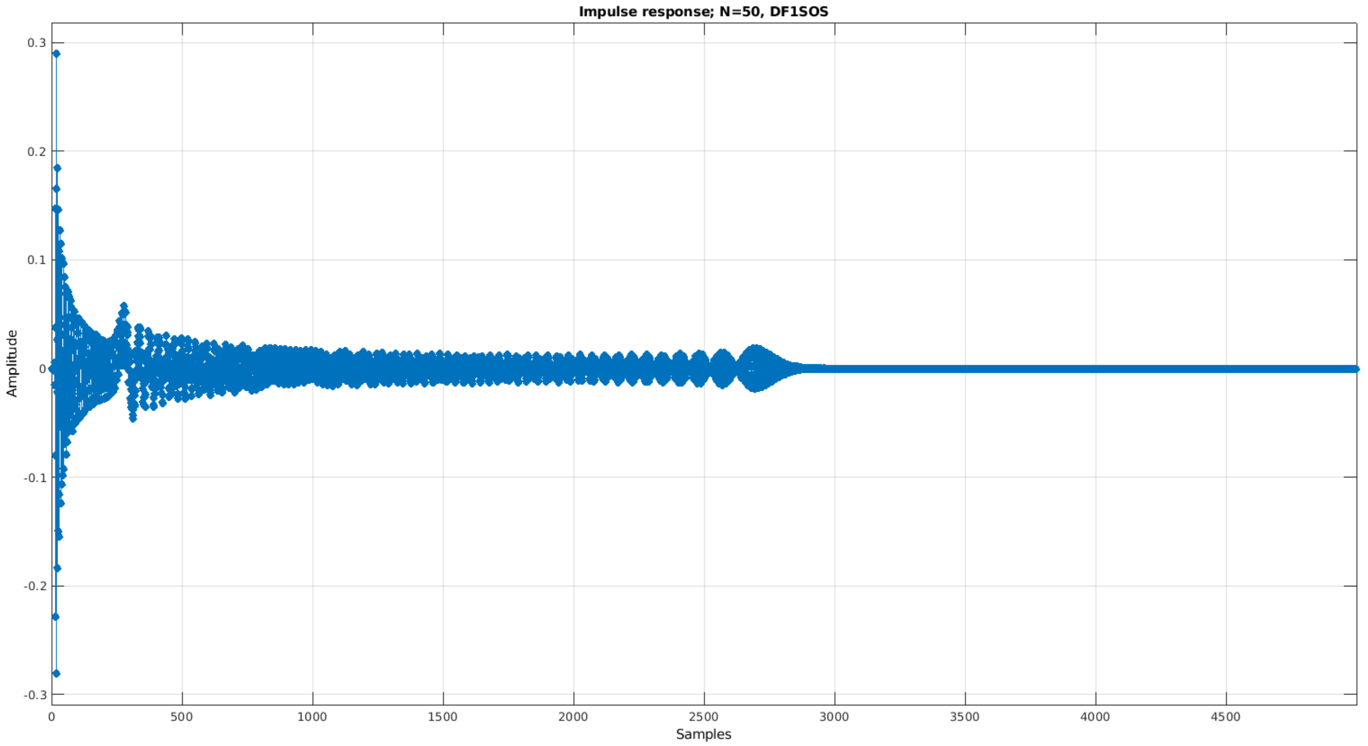

Pole-zero plot; N=1
Imaginary Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1
Real Part

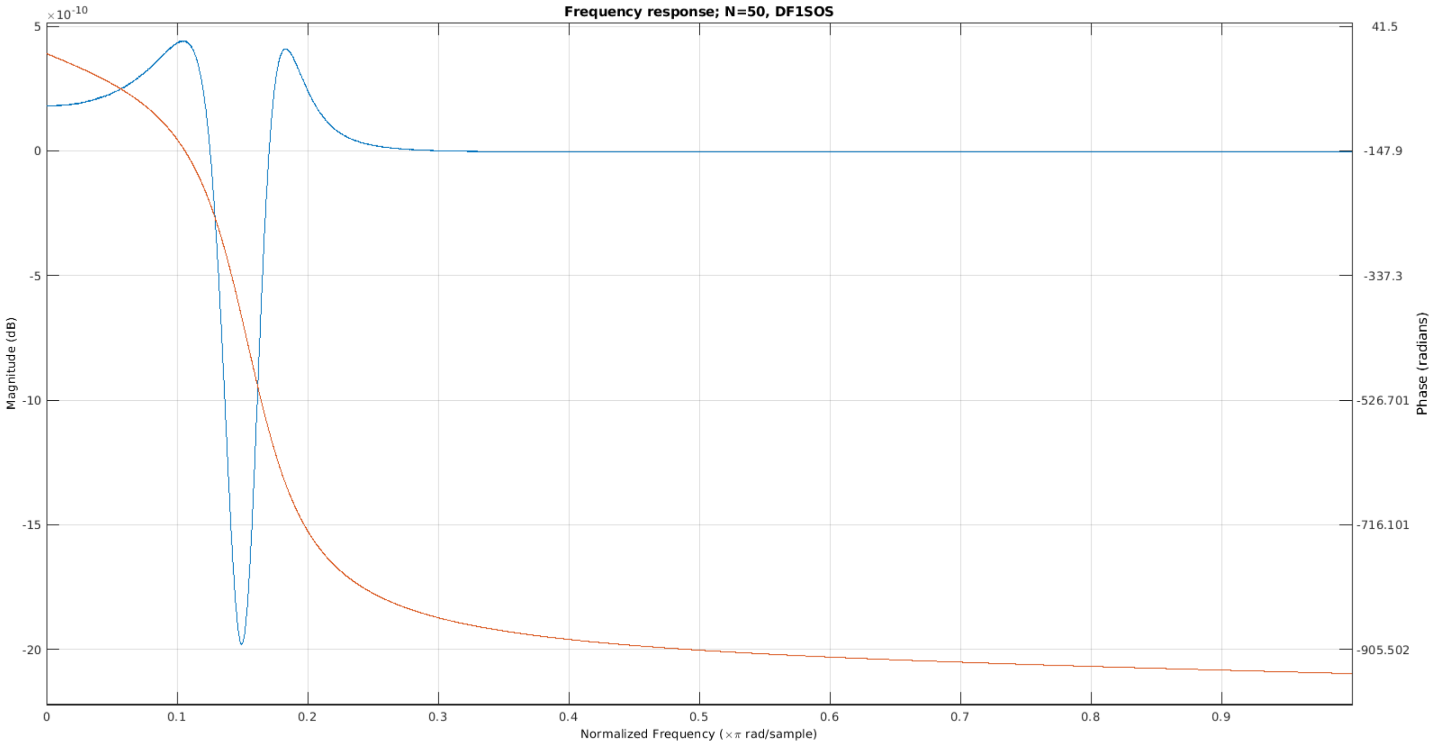

Impulse response; N=50, DF1
Amplitude
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Samples

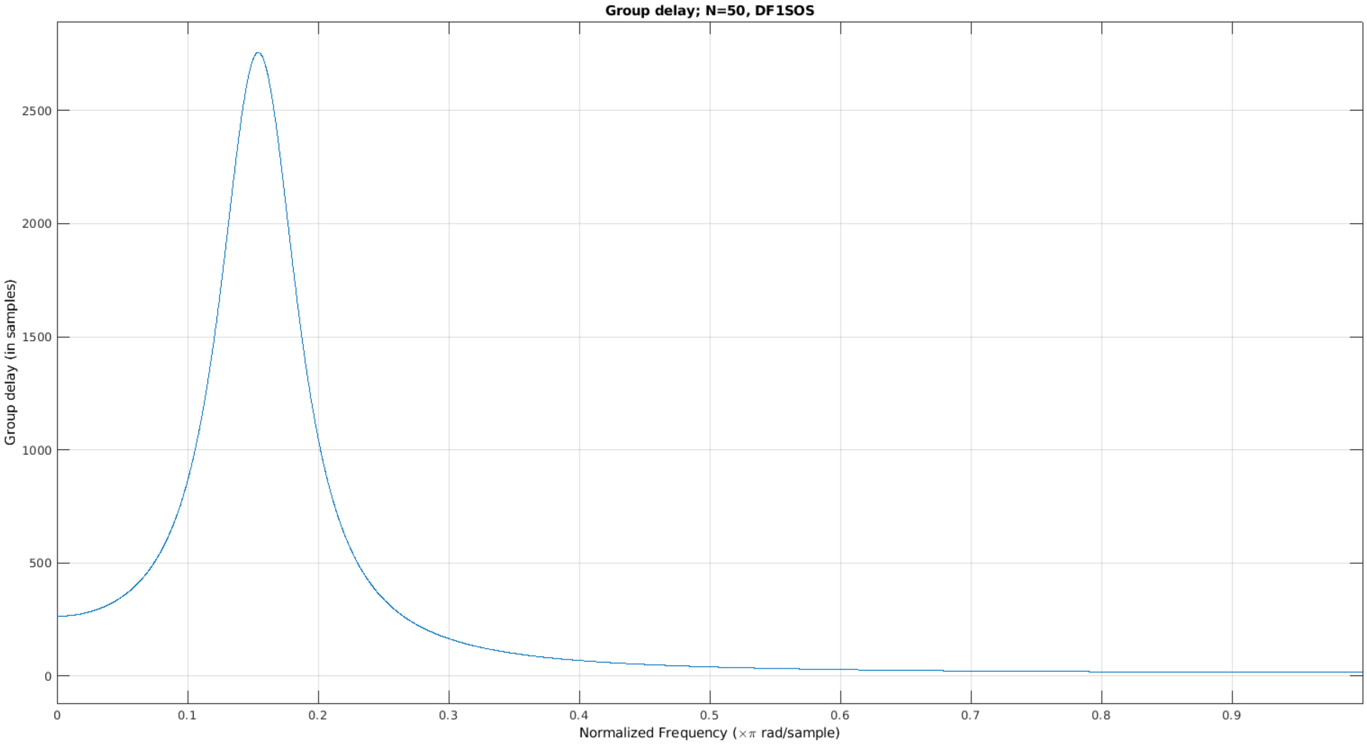

Frequency response; N=50, DF1
×10^-9
Magnitude (dB)
2
1
0
-1
-2
-3
Phase (radians)
-108.879
-273.96
-439.04
-604.121
-769.202
-934.283
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Normalized Frequency (×π rad/sample)

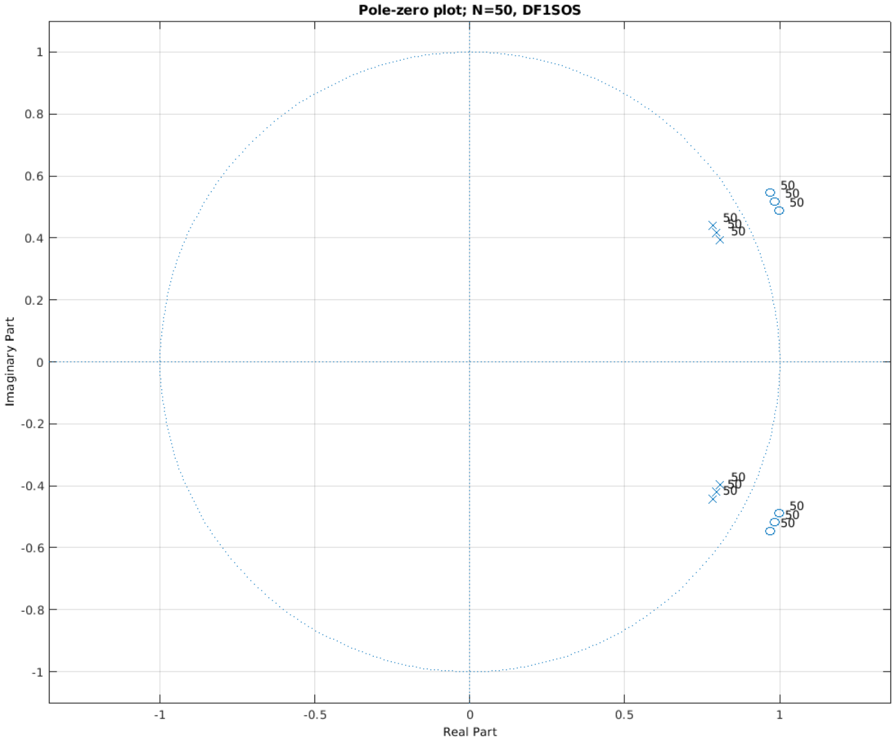

Group delay; N=50, DF1
Group delay (in samples)
2500
2000
1500
1000
500
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Normalized Frequency (×π rad/sample)

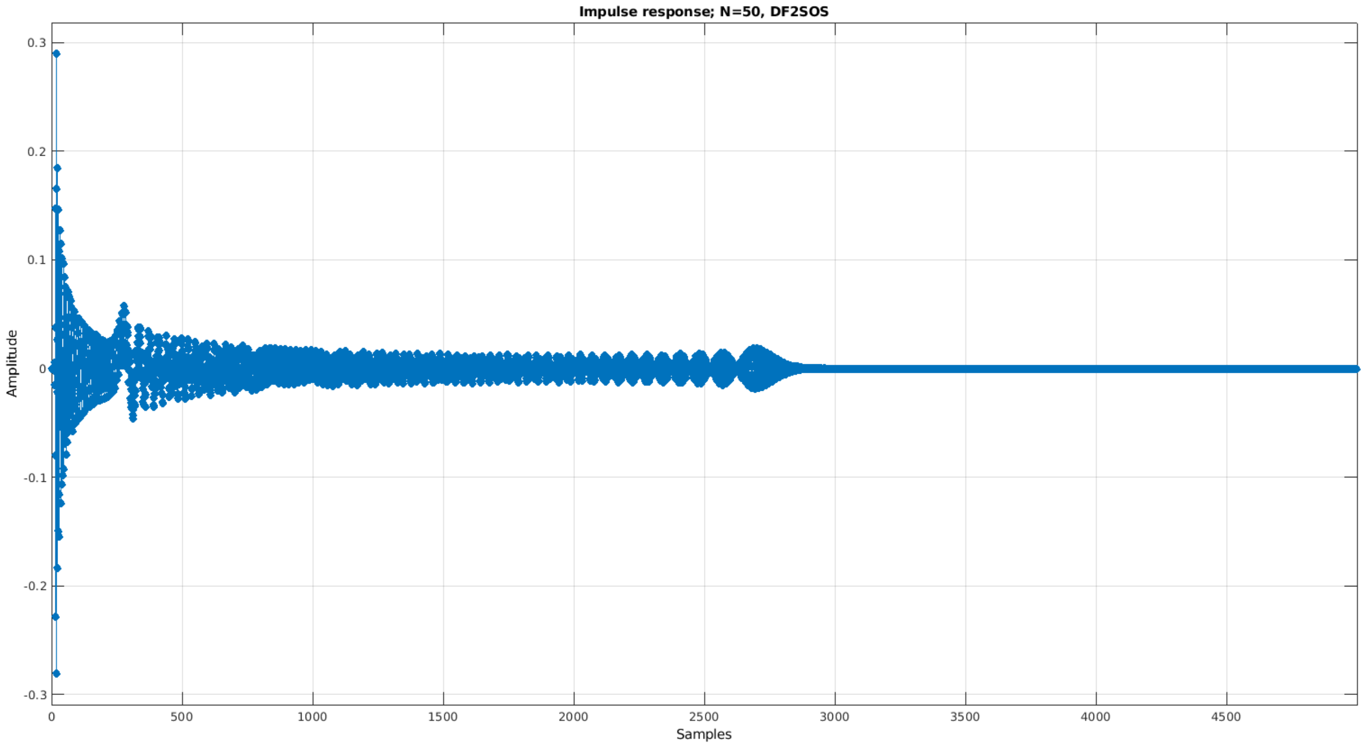

Pole-zero plot; N=50, DF1
Imaginary Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1
Real Part
50
50
50
50
50
50
50
50
50
50
50
50

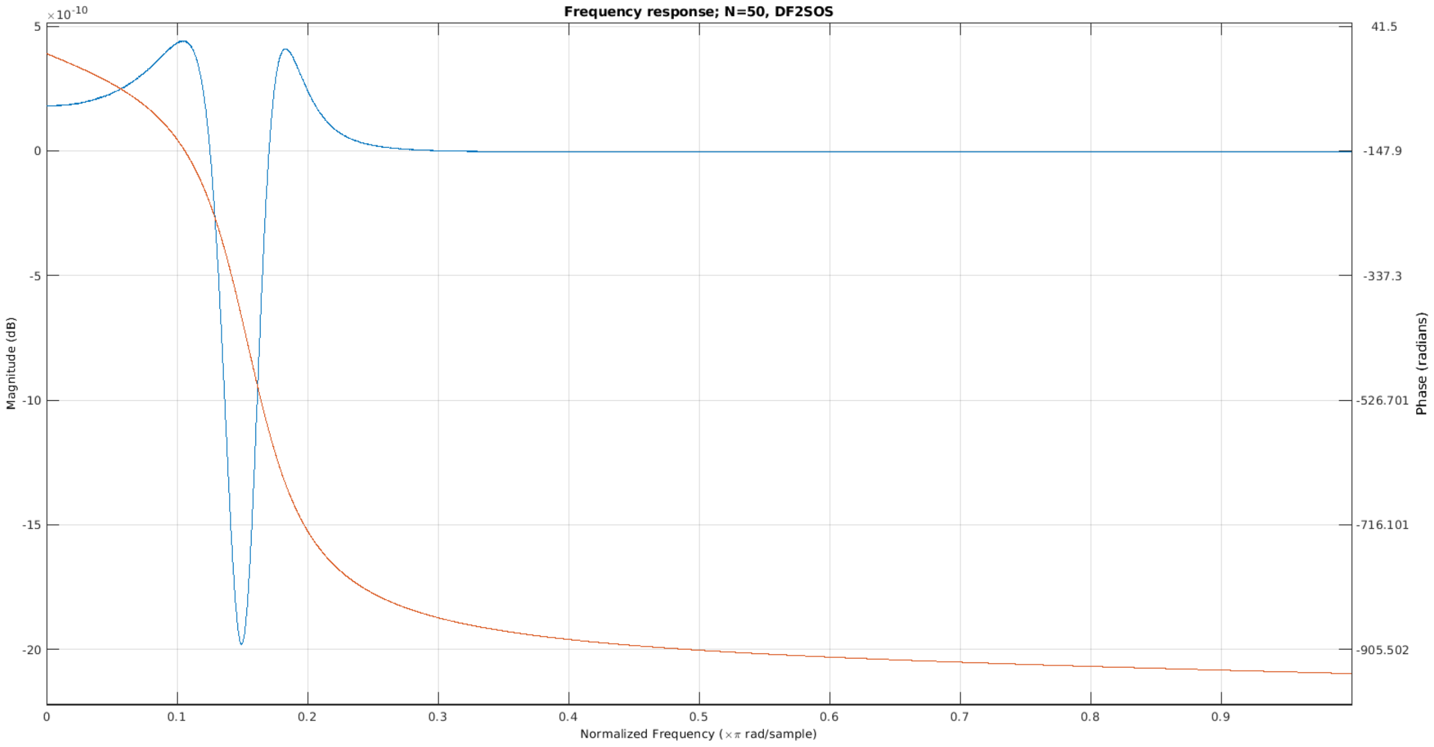

Impulse response; N=50, DF1SOS
Amplitude
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Samples

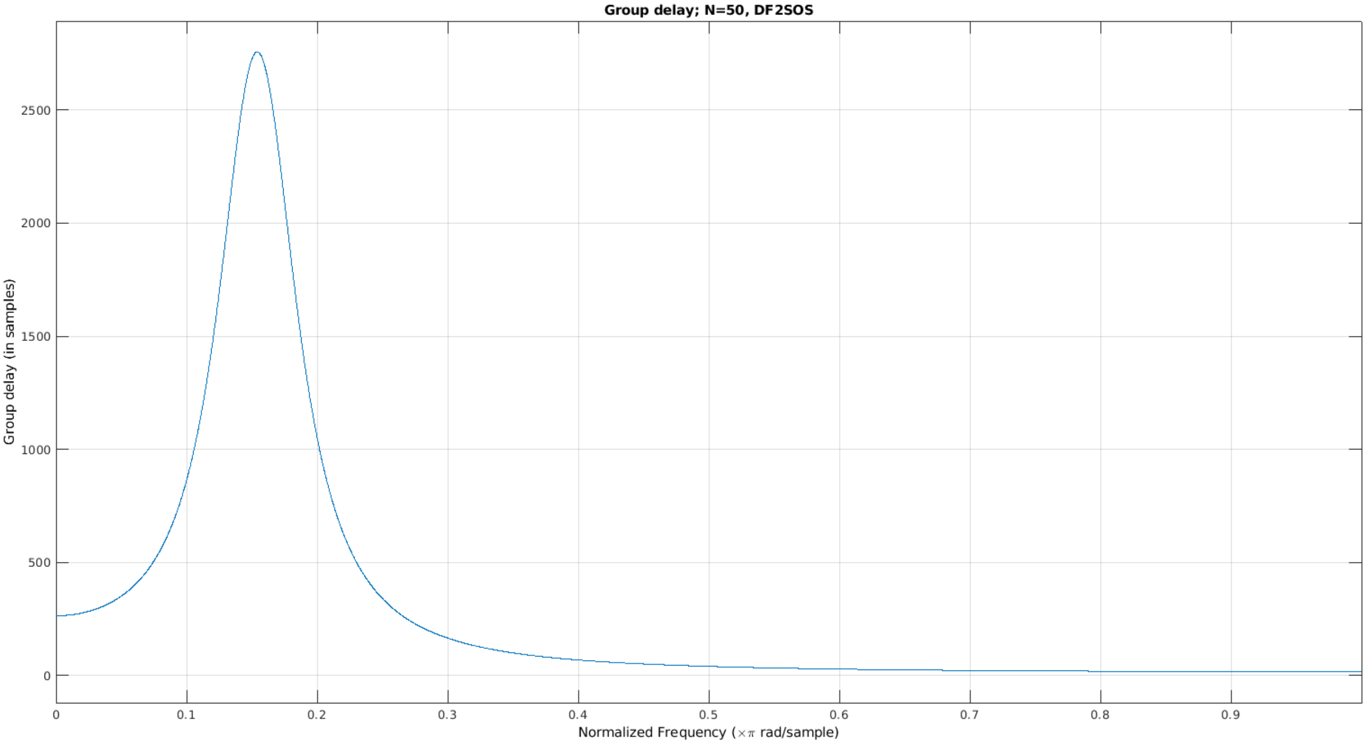

Frequency response; N=50, DF1SOS
$\times 10^{-10}$
Magnitude (dB)
5
0
-5
-10
-15
-20
Phase (radians)
41.5
-147.9
-337.3
-526.701
-716.101
-905.502
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Normalized Frequency ($\times\pi$ rad/sample)

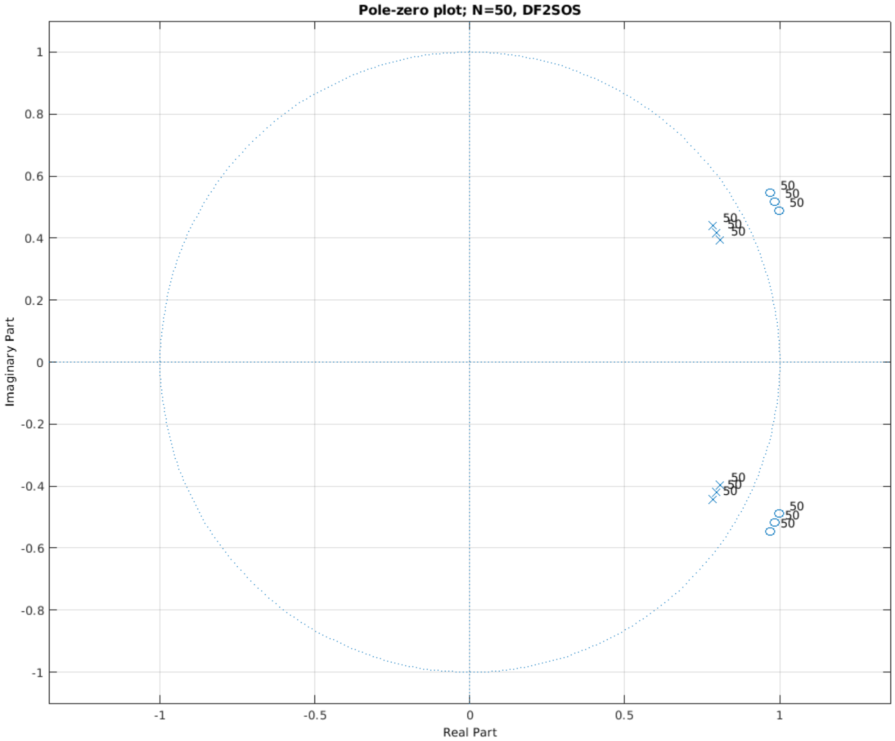

Group delay; N=50, DF1SOS

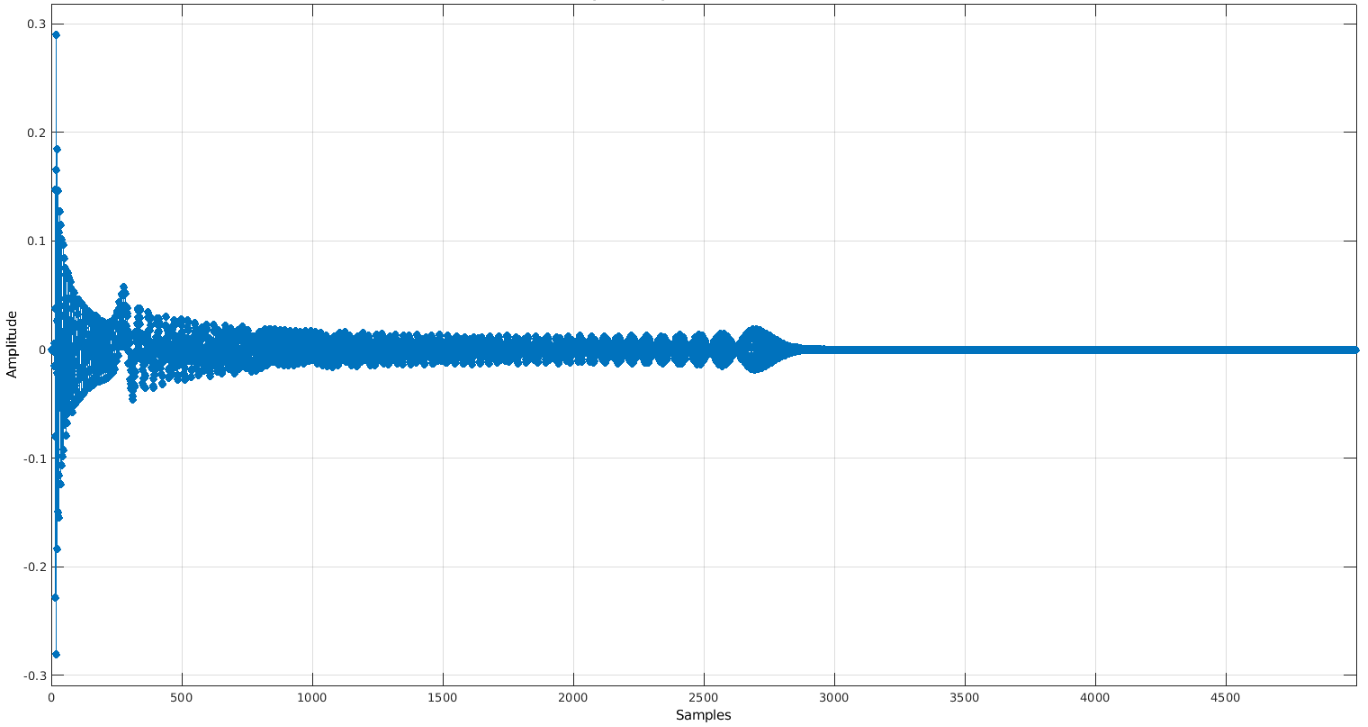

Pole-zero plot; N=50, DF1SOS

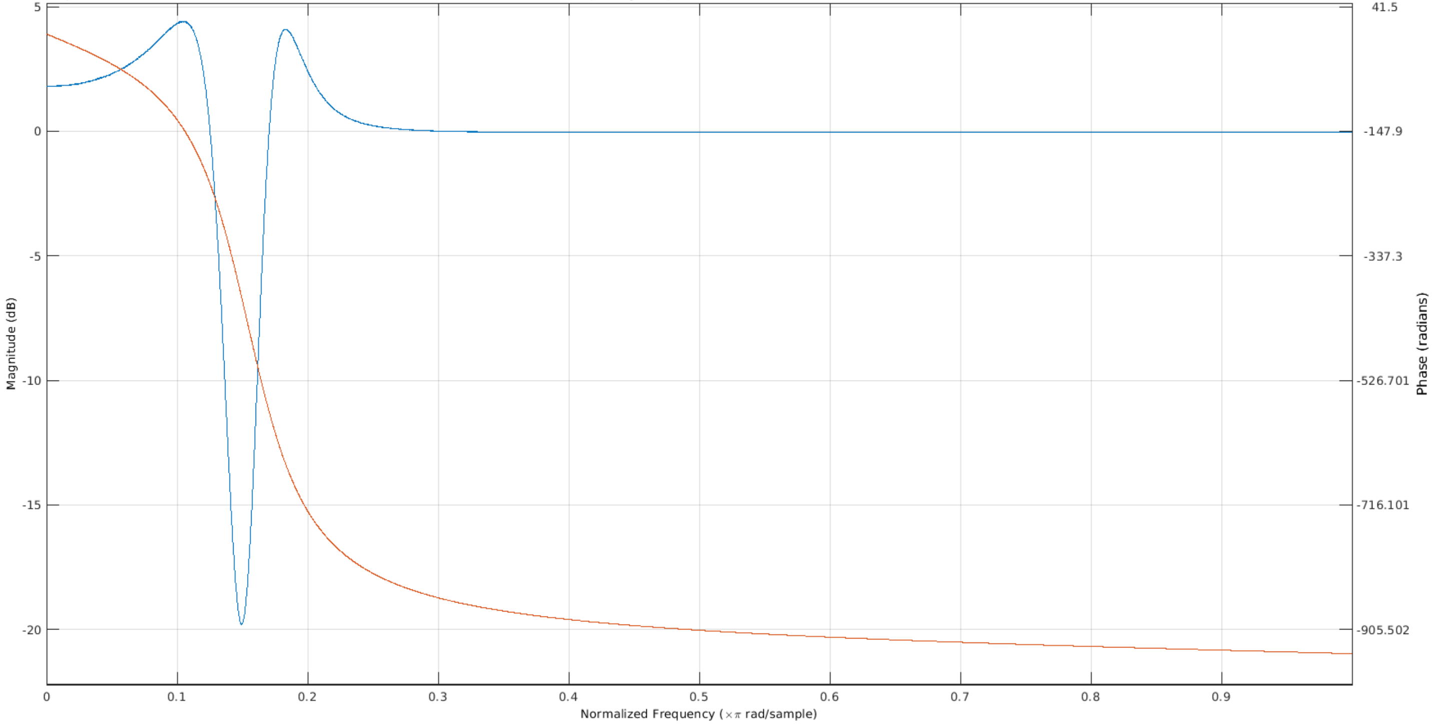

Impulse response; N=50, DF2SOS
Amplitude
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Samples

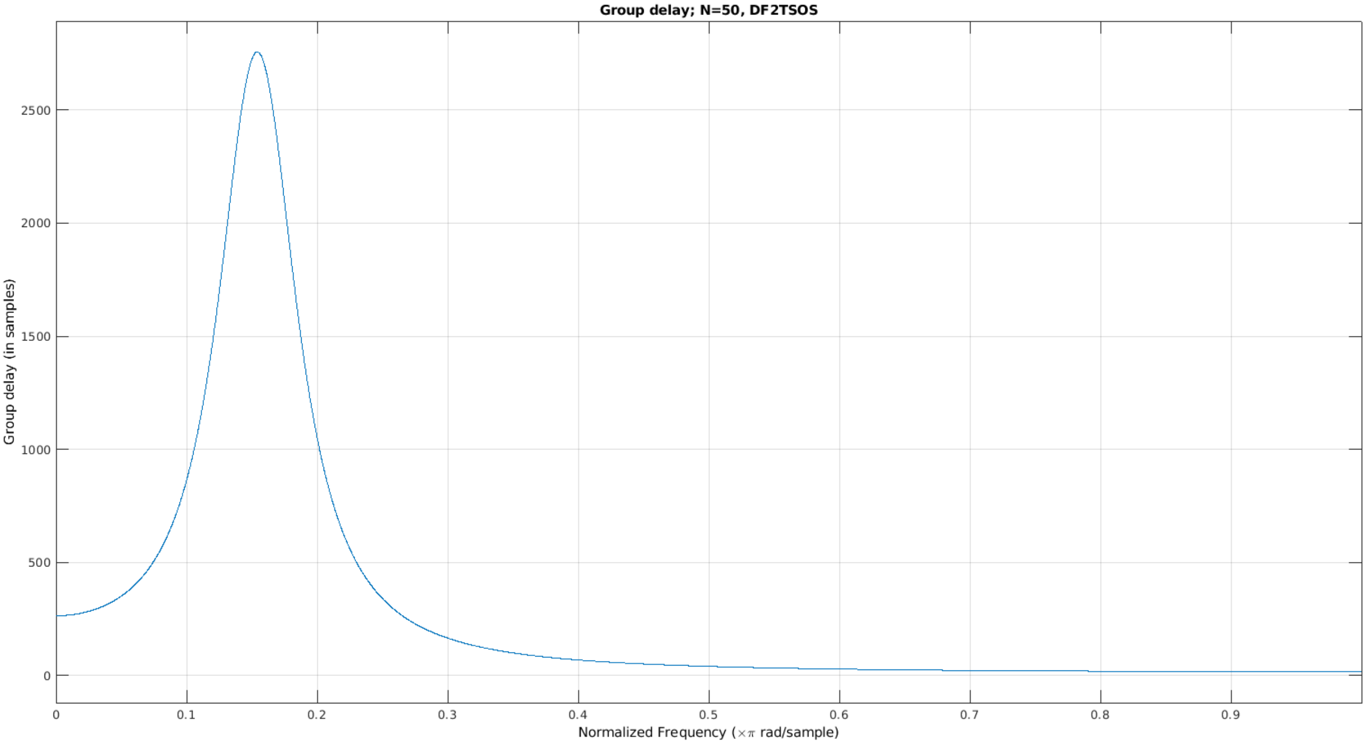

Frequency response; N=50, DF2SOS
×10-10
5
0
-5
-10
-15
-20
Magnitude (dB)
41.5
-147.9
-337.3
-526.701
-716.101
-905.502
Phase (radians)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Normalized Frequency (×π rad/sample)

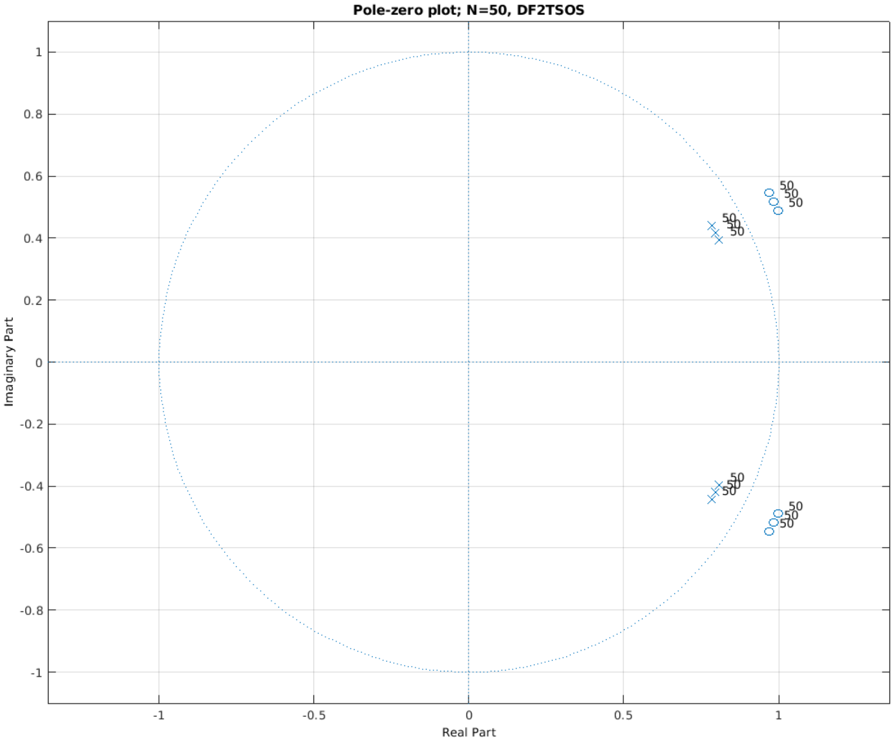

Group delay; N=50, DF2SOS
Group delay (in samples)
0
500
1000
1500
2000
2500
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Normalized Frequency (×π rad/sample)


Pole-zero plot; N=50, DF2SOS
Imaginary Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1
Real Part
50
50
50
50
50
50
50
50
50
50
50
50


Impulse response; N=50, DF2TSOS


Frequency response; N=50, DF2TSOS


Group delay; N=50, DF2TSOS
Group delay (in samples)
2500
2000
1500
1000
500
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Normalized Frequency (×π rad/sample)


Pole-zero plot; N=50, DF2TSOS
Imaginary Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1
Real Part
50
50
50
50
50
50
50
50
50
50
50
50

Source code

```matlab
clc; close all; clear;
load('projIA.mat');

%% create direct form 1 filter
Hd = dfilt.df1(b, a);
plot_filter(Hd, 100, 'N=1');

%% save audio to file
audiowrite('n1.wav', filter(b, a, speech), fs);

%% attempted convolution
b50 = b;
a50 = a;
for i = 1:49
    b50 = conv(b, b50);
    a50 = conv(a, a50);
end % doesn't work; bad numerical stability

% used for second-order sections filters
[s, g] = tf2sos(b, a);

%% direct form 1
Hd_df1 = cascade_filter(dfilt.df1(b, a));
plot_filter(Hd_df1, 5000, 'N=50, DF1');
audiowrite('df1.wav', filter(Hd_df1, speech), fs);

%% direct form 1 (second-order sections)
Hd_df1sos = cascade_filter(dfilt.df1sos(s, g));
plot_filter(Hd_df1sos, 5000, 'N=50, DF1SOS');
audiowrite('df1sos.wav', filter(Hd_df1sos, speech), fs);

%% direct form 2 (second-order sections)
Hd_df2sos = cascade_filter(dfilt.df2sos(s, g));
plot_filter(Hd_df2sos, 5000, 'N=50, DF2SOS');
audiowrite('df2sos.wav', filter(Hd_df2sos, speech), fs);

%% direct form 2 transposed (second-order sections)
Hd_df2tsos = cascade_filter(dfilt.df2tsos(s, g));
plot_filter(Hd_df2tsos, 5000, 'N=50, DF2TSOS');
audiowrite('df2tsos.wav', filter(Hd_df2tsos, speech), fs);

%% cascade filter 50 times
function Hd = cascade_filter(Hd)
    Hd = dfilt.cascade(repmat(Hd, 1, 50));
end

%% plot impulse response, frequency response, group delay, and
%  pole-zero plot of filter
function plot_filter(Hd, N, figname)
    impz(Hd, N);
    title(sprintf('Impulse response; %s', figname));
    freqz(Hd);
    title(sprintf('Frequency response; %s', figname));
    grpdelay(Hd);
    title(sprintf('Group delay; %s', figname));
    zplane(Hd);
    title(sprintf('Pole-zero plot; %s', figname));
end
```