

Digital Signal Processing

Final Project

Audio Equalizer

Student Name	ID
Mohannad Bashar	6656
Youssef Victor	6668
Mohamed Medhat Zaytoon	6672

"main.m" file

```
close all;
clear;

[file, path] = uigetfile('*.wav');
fullFileName = fullfile(path, file);

[data, original_fs] = audioread(fullFileName);

disp('File information:');
disp(['Path: ' fullFileName]);
disp(['Data dimensions: ' mat2str(size(data))]);
disp(['Frequency: ' num2str(original_fs)]);

bands = get_bands();

fs = original_fs;
if bands(end, end) >= fs / 2
    margin = 11;
    fs = 2 * (bands(end, end) + margin);
    resampled_data = resample(data, fs, original_fs);
else
    resampled_data = data;
end

gains = zeros(1, length(bands));
for i = 1:length(bands)
    gains(i) = get_number(['Enter gain for ' mat2str(bands(i,:)) 'Hz (between -20 dB and 20 dB): '], @(x) x >= -20 && x <= 20);
end

filter_type = get_string('Enter filter type ("fir" or "iir"):', @(x) strcmp('fir', x) || strcmp('iir', x));

output_fs = get_number('Enter a valid output sample rate: ', @(x) x > 340);

if strcmp('fir', filter_type)
    fir_order = 150;
    filters = fir_filters(fir_order, fs, bands);
else
    iir_order = 4;
    filters = iir_filters(iir_order, fs, bands);
end

acc_filtered = resampled_data .* 0;

freq_range_plt = [' 0-170 Hz'; '170-310 Hz'; '310-600 Hz'; '0.6-1 kHz'; '1-3 kHz'; '3-6 kHz'; '6-12 kHz'; '12-14 kHz'; '14-16 KHz'];
for i = 1:length(filters)
    plot_requirements(filters(i), fs, [mat2str(bands(i,:)) 'Hz']);
    filtered = filter(filters(i).Numerator, filters(i).Denominator, resampled_data);
    plot_time_frequency_domain(filtered, fs, ['Output in time-domain for filter ' freq_range_plt(i,:)'], ['Output in frequency-domain for filter ' freq_range_plt(i,:)]);
    acc_filtered = acc_filtered + filtered * db2mag(gains(i));
    [z, p, k] = tf2zpk(filters(i).Numerator, filters(i).Denominator);
    order = filtord(filters(i).Numerator, filters(i).Denominator);
    fprintf('The gain of %s filter : %s is %f , order is %d \n', filter_type, freq_range_plt(i,:), k, order);
end

acc_filtered = resample(acc_filtered, output_fs); % resample to the output fs
plot_time_frequency_domain(data, original_fs, 'Input in time', 'Input in freq.', acc_filtered, output_fs, 'Output in freq.', 'Output in time.');

fullFileName = fullfile(path, 'output_sample_run.wav');

audiowrite(fullFileName, acc_filtered, output_fs);
```

"plot_requirements" helper function

Plots the frequency response (magnitude and phase), impulse response, step response, and the pole-zero plot of a given filter.

```
function plot_requirements(filter, fs, name)
    x = fvtool(filter.Numerator, filter.Denominator);
    set_common_properties(x, fs);
    x.Analysis = 'freq';
    x.Name = ['Magnitude response (dB) and phase response of filter: ' name];

    x = fvtool(filter.Numerator, filter.Denominator);
    set_common_properties(x, fs);
    x.Analysis = 'impulse';
    x.Name = ['Impulse response of filter: ' name];

    x = fvtool(filter.Numerator, filter.Denominator);
    set_common_properties(x, fs);
    x.Analysis = 'step';
    x.Name = ['Step response of filter: ' name];

    x = fvtool(filter.Numerator, filter.Denominator);
    set_common_properties(x, fs);
    x.Analysis = 'polezero';
    x.Name = ['Pole-zero plot of filter: ' name];
end

function set_common_properties(x, fs)
    x.NormalizedFrequency = 'off';
    x.fs = fs;
    x.WindowStyle = 'normal';
    x.NumberTitle = 'off';
end
```

“plot_time_frequency_domain” helper function

Plots one or two signals in both time and frequency domains.

```
function plot_time_frequency_domain(data, fs, title_time, title_freq, varargin)
    sub_plots = 1;
    figure;
    if nargin > 5
        sub_plots = 2;
        length_output = length(varargin{1});
        df = varargin{2} / length_output;
        frequency_audio = -varargin{2}/2 : df : varargin{2}/2-df;

        not_shifted_output_fft = fft(varargin{1});
        fft_data = fftshift(not_shifted_output_fft) / length(not_shifted_output_fft);

        subplot(2, sub_plots , 3);
        plot(frequency_audio, abs(fft_data));
        title(varargin{3});
        xlabel('Frequency (Hz)');
        ylabel('Amplitude');

        subplot(2, sub_plots, 4);
        t = (0:length_output-1)/ varargin{2};
        plot(t, varargin{1});
        title(varargin{4});
        xlabel('Time (s)');
        ylabel('Amplitude');
    end

    length_data = length(data);
    df = fs / length_data;
    frequency_audio = -fs/2 : df : fs/2-df;
    not_shifted_fft = fft(data);
    fft_data = fftshift(not_shifted_fft) / length(not_shifted_fft);

    subplot(2, sub_plots, 1);
    plot(frequency_audio, abs(fft_data));
    title(title_freq);
    xlabel('Frequency (Hz)');
    ylabel('Amplitude');

    subplot(2, sub_plots, 2);
    t = (0:length_data-1)/ fs;
    plot(t, data)
    title(title_time);
    xlabel('Time (s)');
    ylabel('Amplitude');
end
```

“fir_filters” helper function

Returns a list of fir filters of a given order, fs, and bands.

```
function filters = fir_filters(order, fs, bands)
    filters = [];
    for i = 1:length(bands)
        band = bands(i,:) / (fs / 2);
        if band(1) == 0
            b = fir1(order, band(2));
        else
            b = fir1(order, band);
        end
        filter = Filter(b, 1);
        filters = [filters filter];
    end
end
```

“iir_filters” helper function

Returns a list of iir (butter) filters of a given order, fs, and bands.

```
function filters = iir_filters(order,fs,bands)
    filters = [];
    for i = 1:length(bands)
        band = bands(i,:) / (fs / 2);
        if band(1) == 0
            [b, a] = butter(order,band(2));

        else
            [b, a] = butter(order,band,'bandpass');
        end
        filters = [filters Filter(b, a)];
    end
end
```

“Filter” class

Represents a filter by its numerator and denominator. It makes it easier for us to have a vector of filters.

```
classdef Filter

    properties
        Numerator
        Denominator
    end

    methods
        function obj = Filter(numerator, denominator)
            obj.Numerator = numerator;
            obj.Denominator = denominator;
        end
    end
end
```

“get_bands” helper function

Returns a list of filter bands as required.

```
function bands = get_bands()
    bands = [0 170;
              170 310;
              310 600;
              600 1000;
              1000 3000;
              3000 6000;
              6000 12000;
              12000 14000;
              14000 16000];
end
```

“get_number” helper function

Takes a message to display to the user, and keep asking the user to enter a valid real number based on the validation predicate.

```
function f = get_number(prompt, predicate)
    while 1
        x = inputdlg(prompt);
        n = str2double(x{1});
        if length(n) == 1 && isreal(n) && predicate(n)
            f = n;
            break;
        end
    end
end
```

“get_string” helper function

Takes a message to display to the user, and keep asking the user to enter a valid string based on the validation predicate.

```
function f = get_string(prompt, predicate)
    while 1
        s = inputdlg(prompt);
        if predicate(s{1})
            f = s{1};
            break;
        end
    end
end
```

Sample run 1:

Type of filter: fir

Specified gains for each of the nine filters: 0, 0, 0, 0, 0, 0, 0, 0, 0 dB

Specified output frequency: 8000 Hz

Output:

File information:

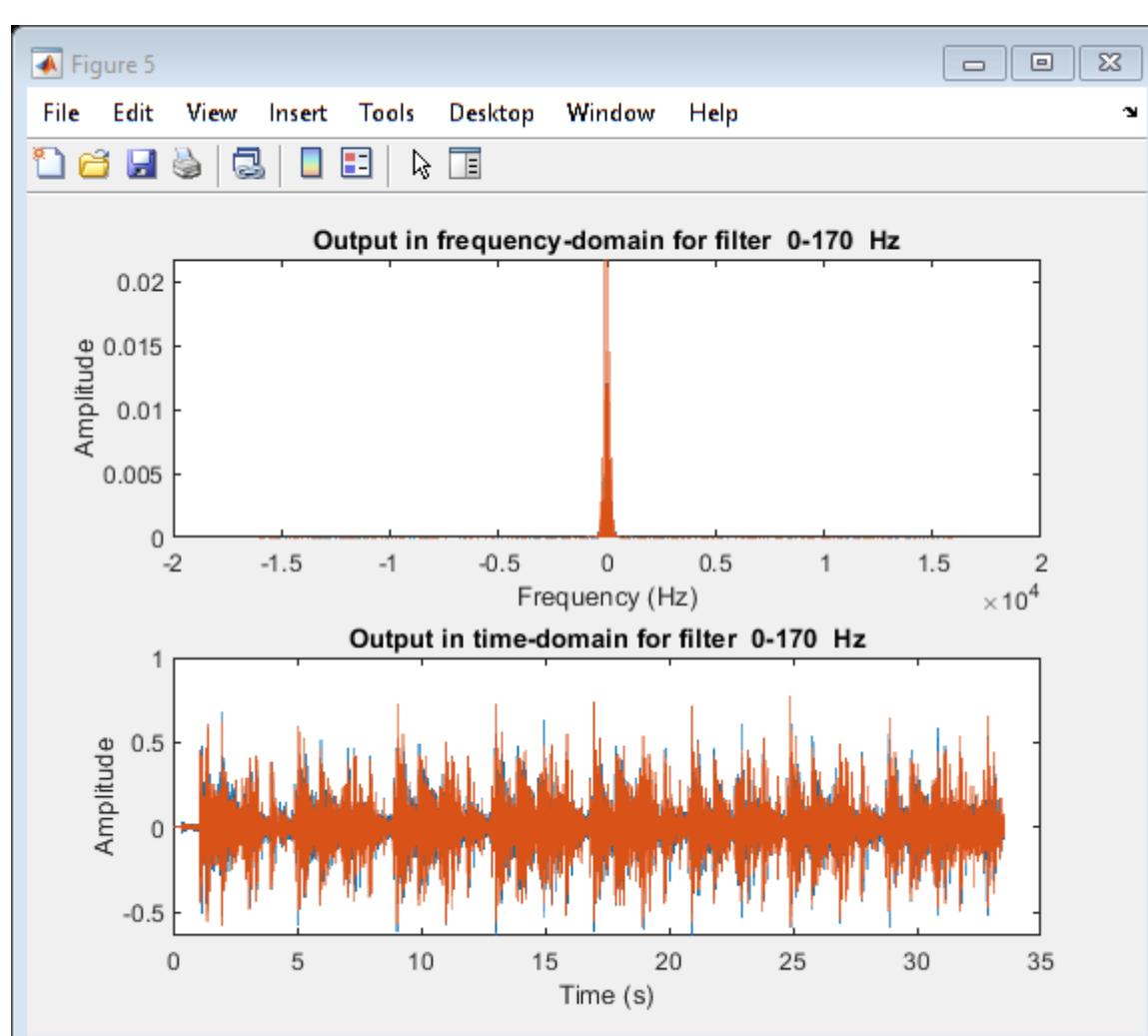
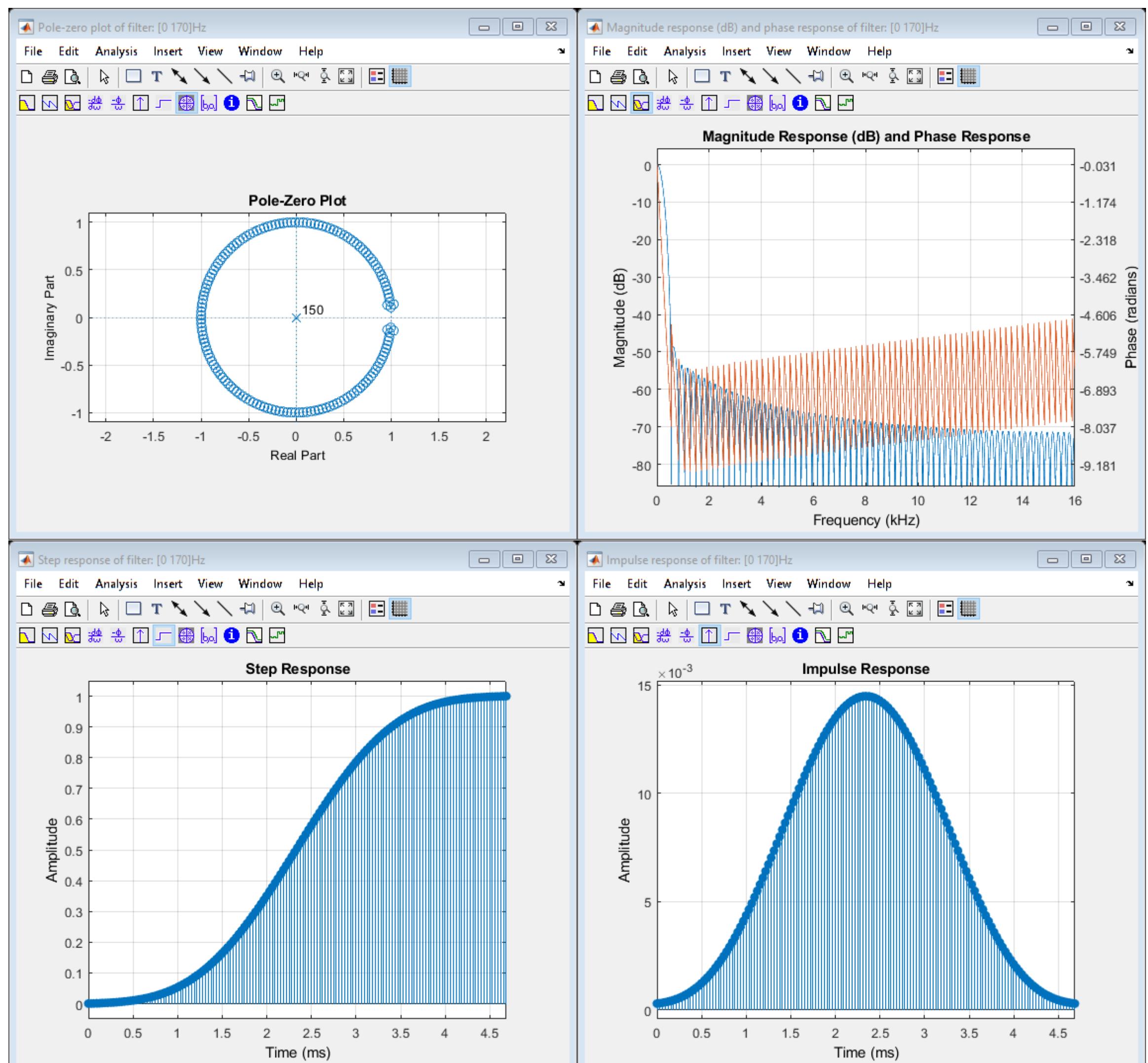
Path: C:\Users\PC\Desktop\FinalProject_DSP\voice.wav

Data dimensions: [268237 2]

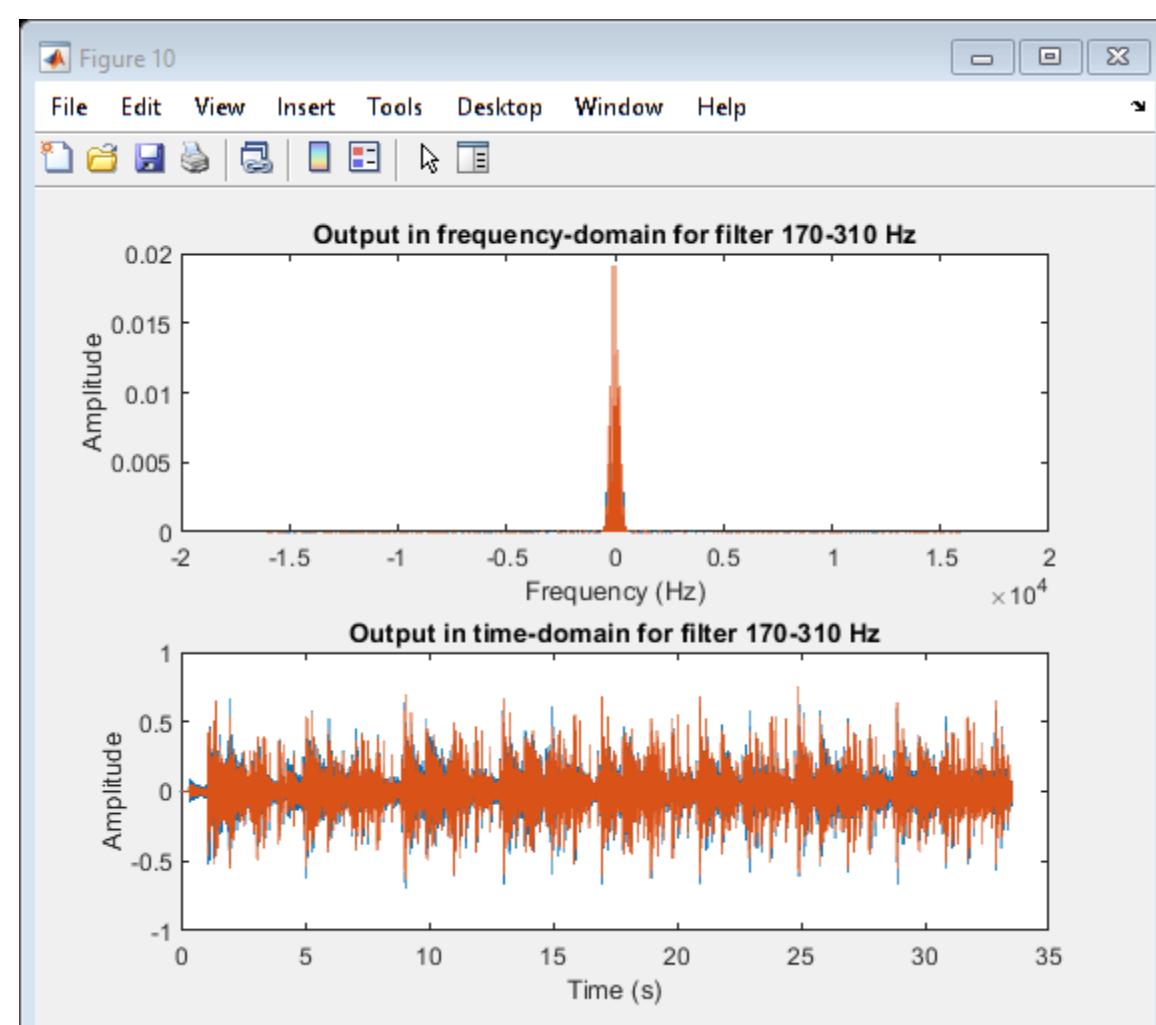
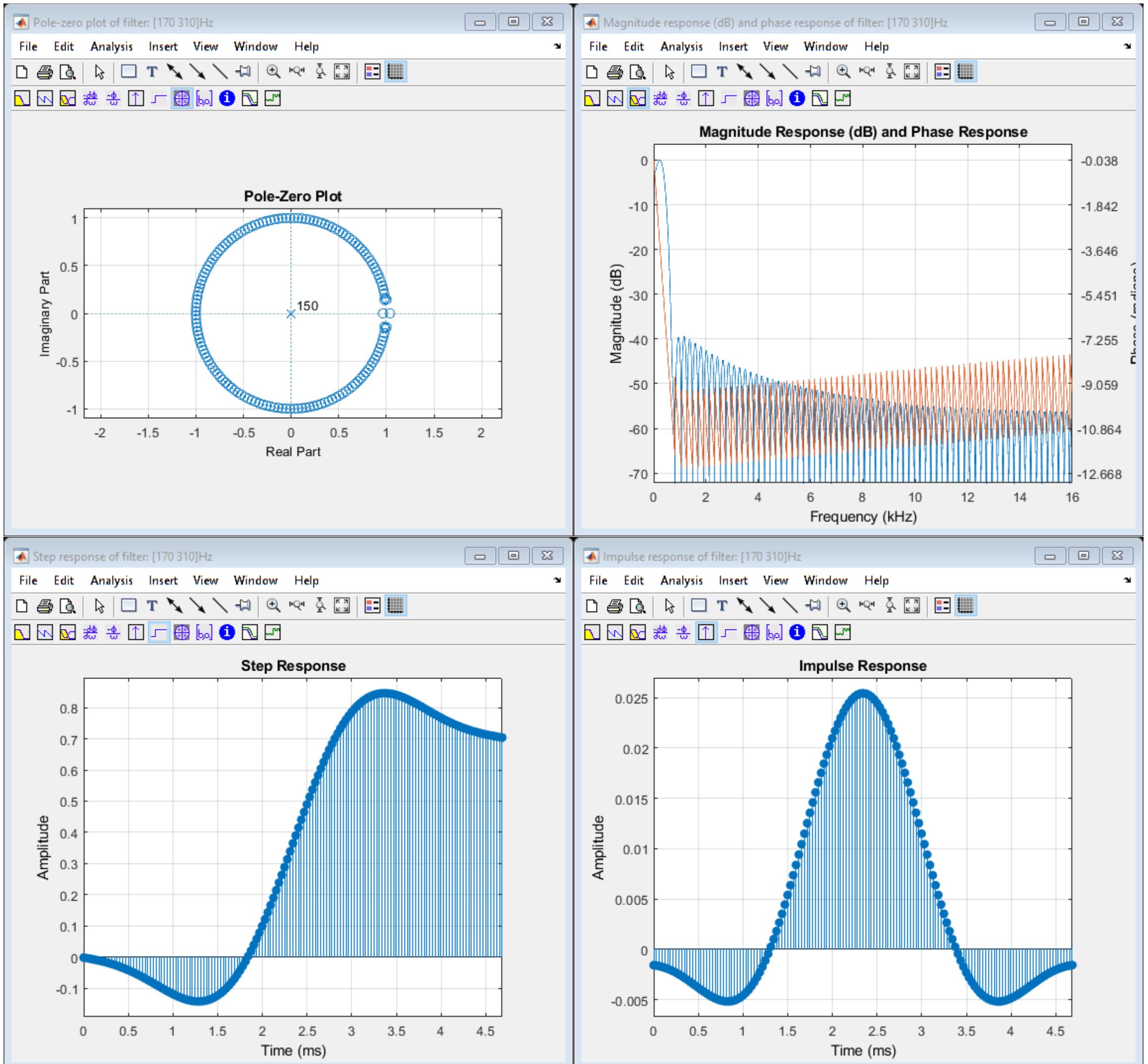
Frequency: 8000

```
The gain of fir filter : 0-170 Hz is 0.000277 , order is 150
The gain of fir filter : 170-310 Hz is -0.001567 , order is 150
The gain of fir filter : 310-600 Hz is 0.000804 , order is 150
The gain of fir filter : 0.6-1 kHz is 0.000115 , order is 150
The gain of fir filter : 1-3 kHz is -0.000228 , order is 150
The gain of fir filter : 3-6 kHz is 0.000054 , order is 150
The gain of fir filter : 6-12 kHz is 0.000099 , order is 150
The gain of fir filter : 12-14 kHz is -0.000537 , order is 150
The gain of fir filter : 14-16 KHz is 0.000384 , order is 150
```

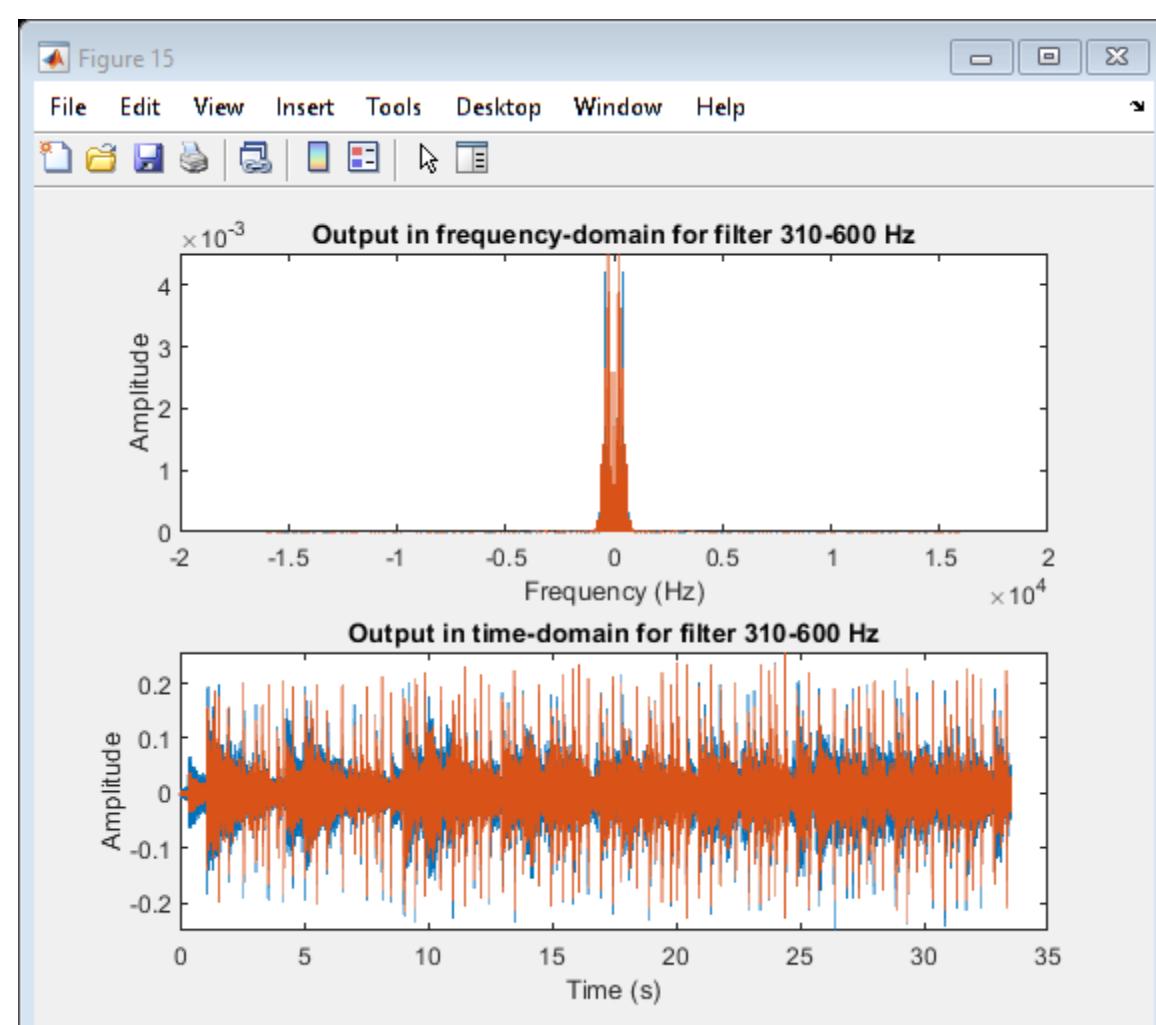
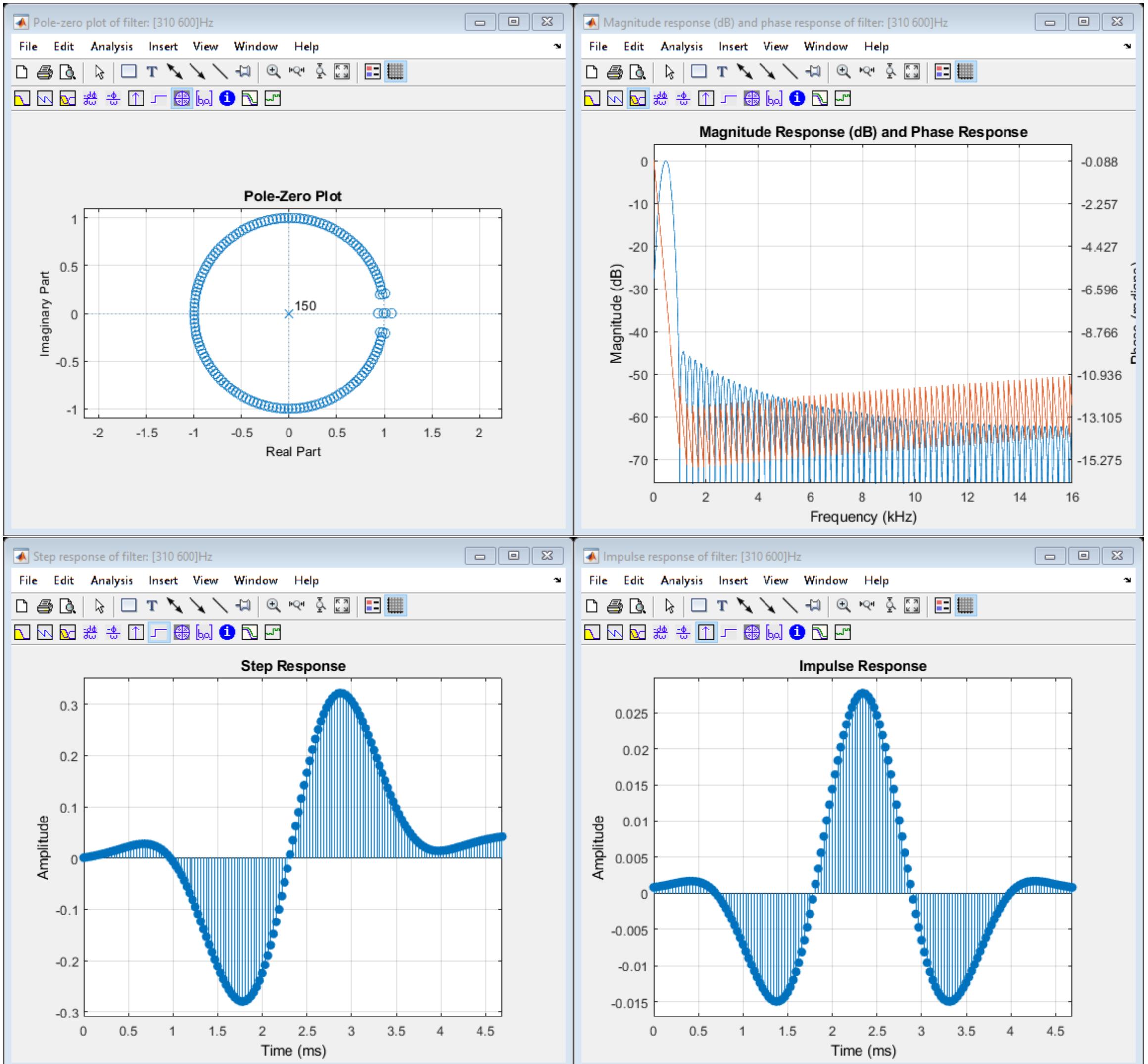
Sample run 1 (cont.):



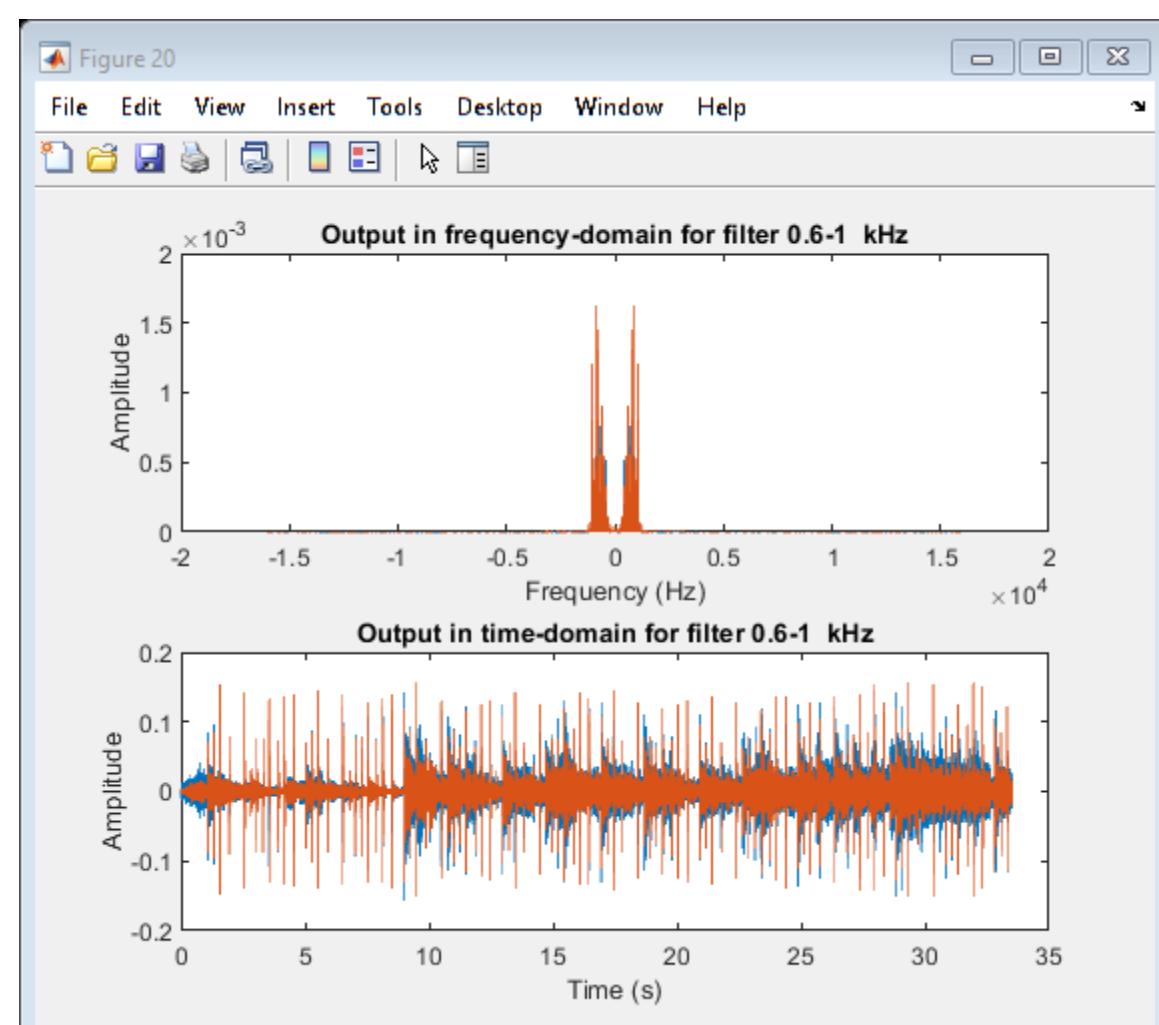
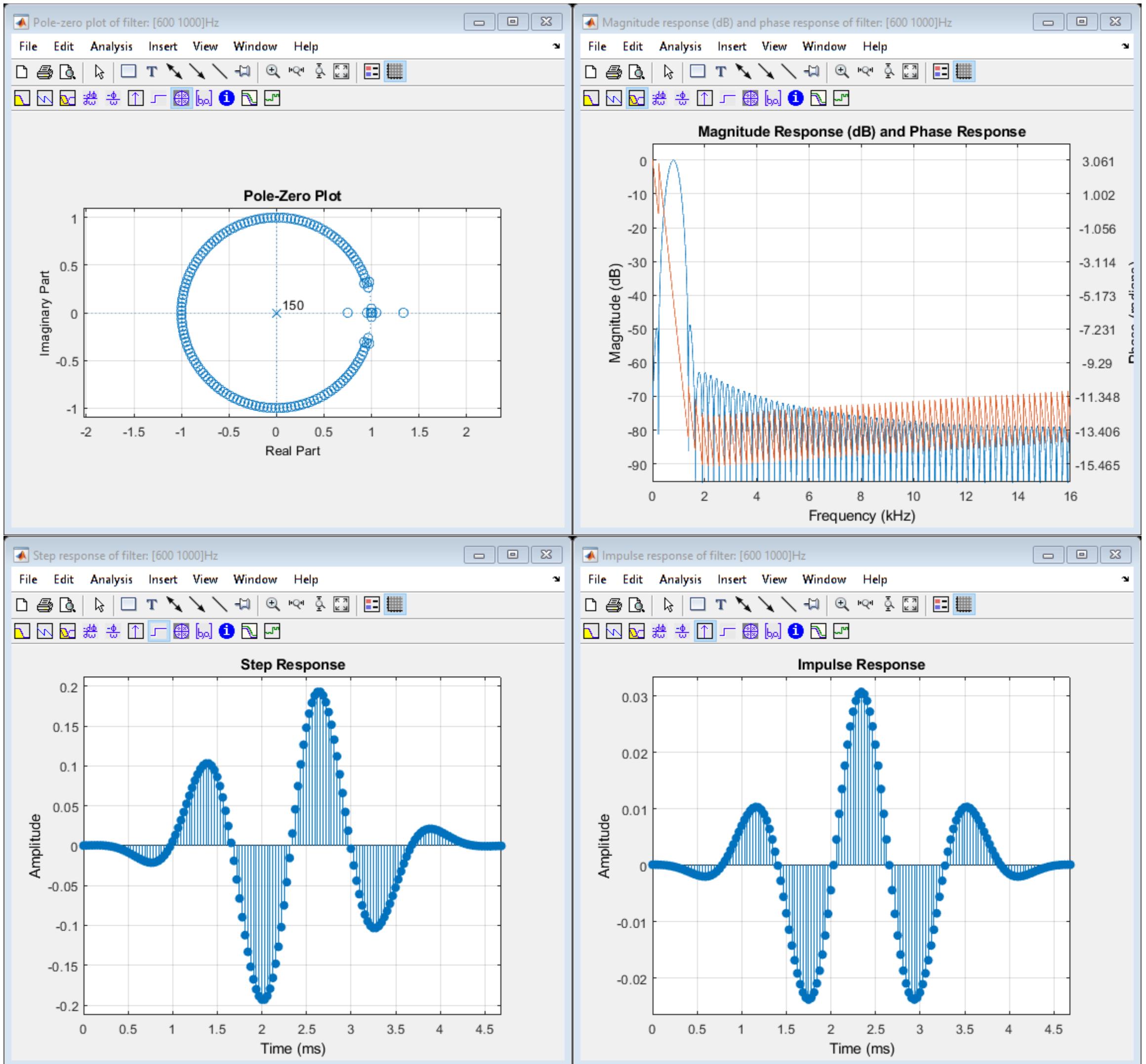
Sample run 1 (cont.):



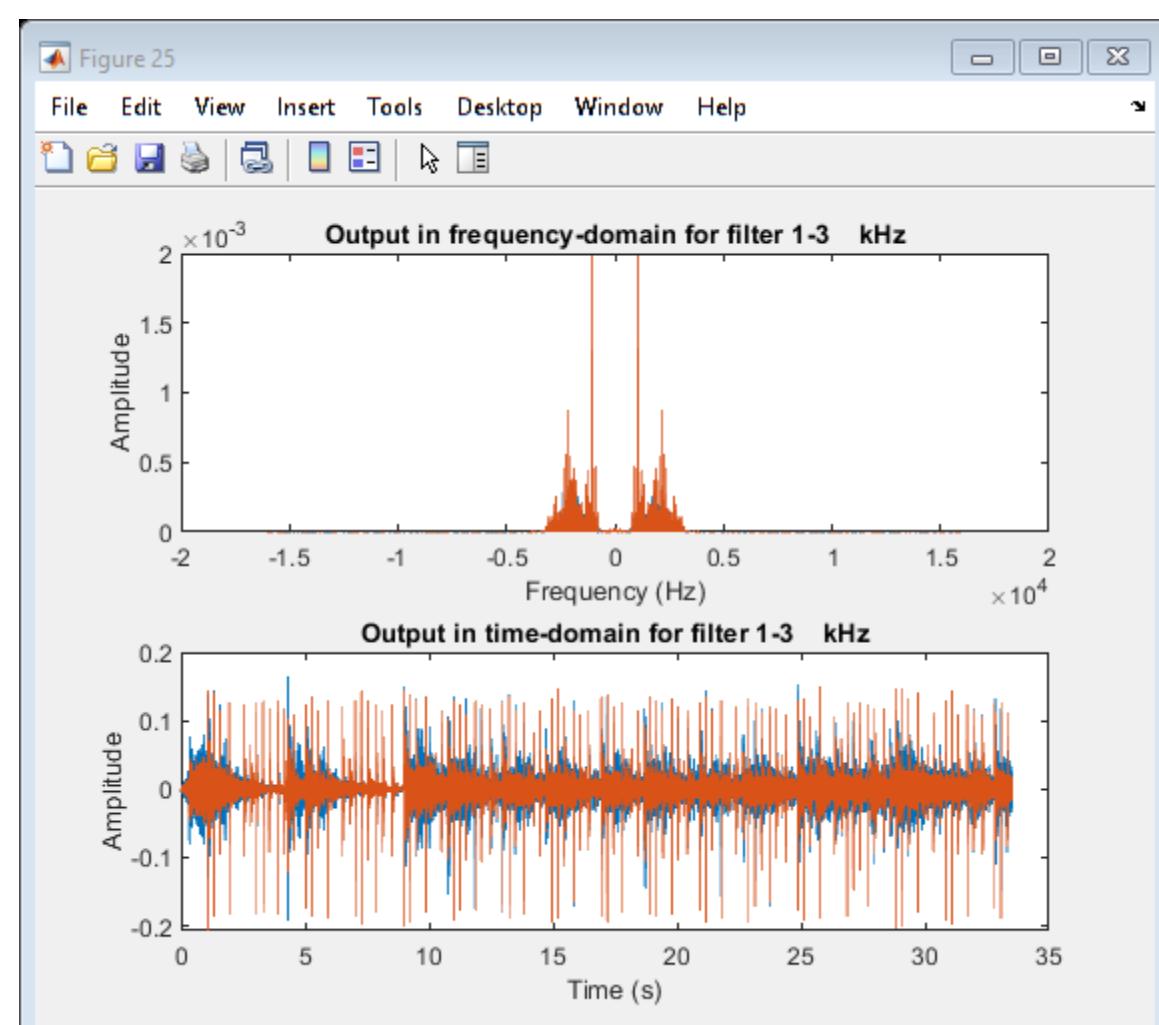
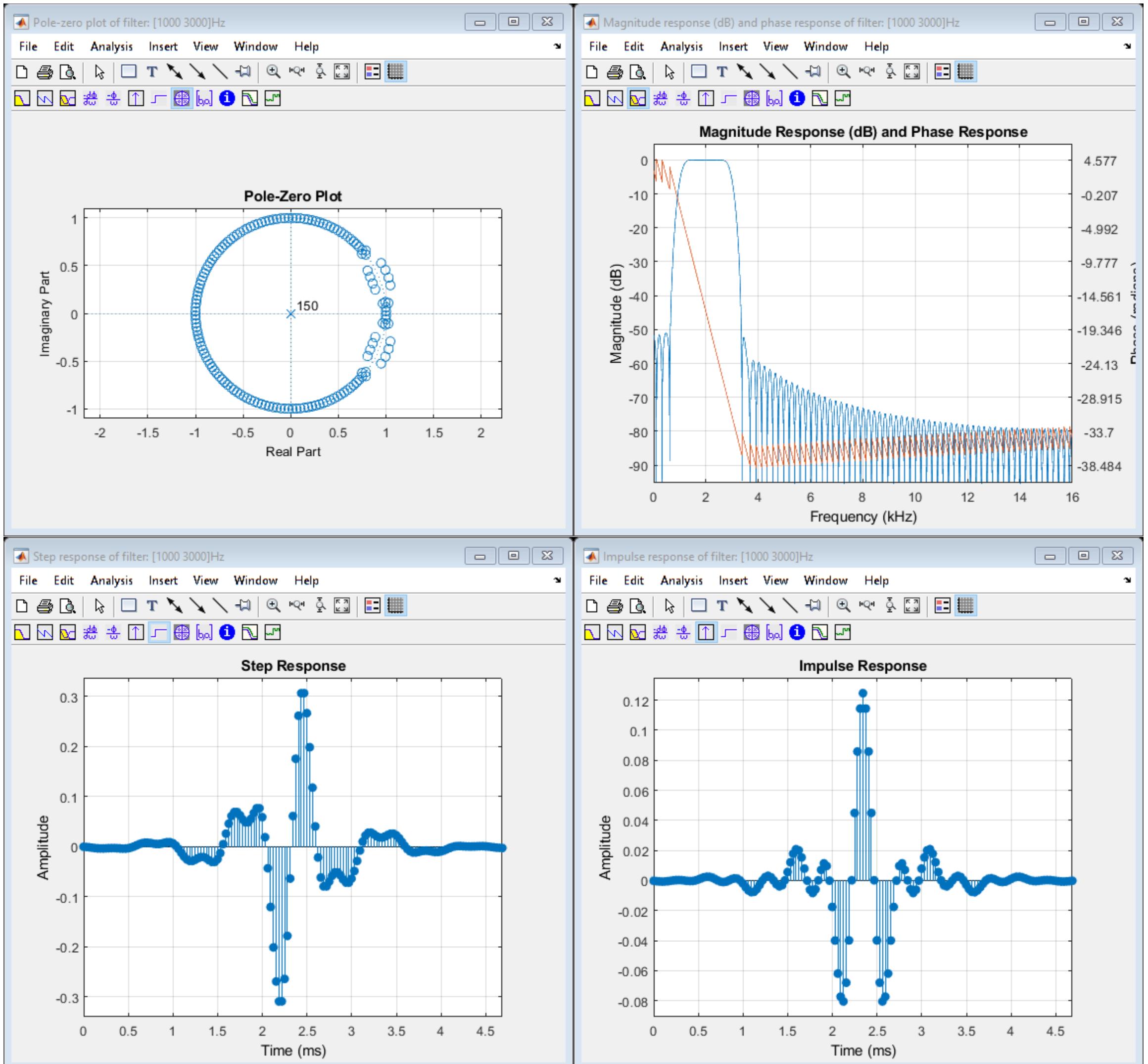
Sample run 1 (cont.):



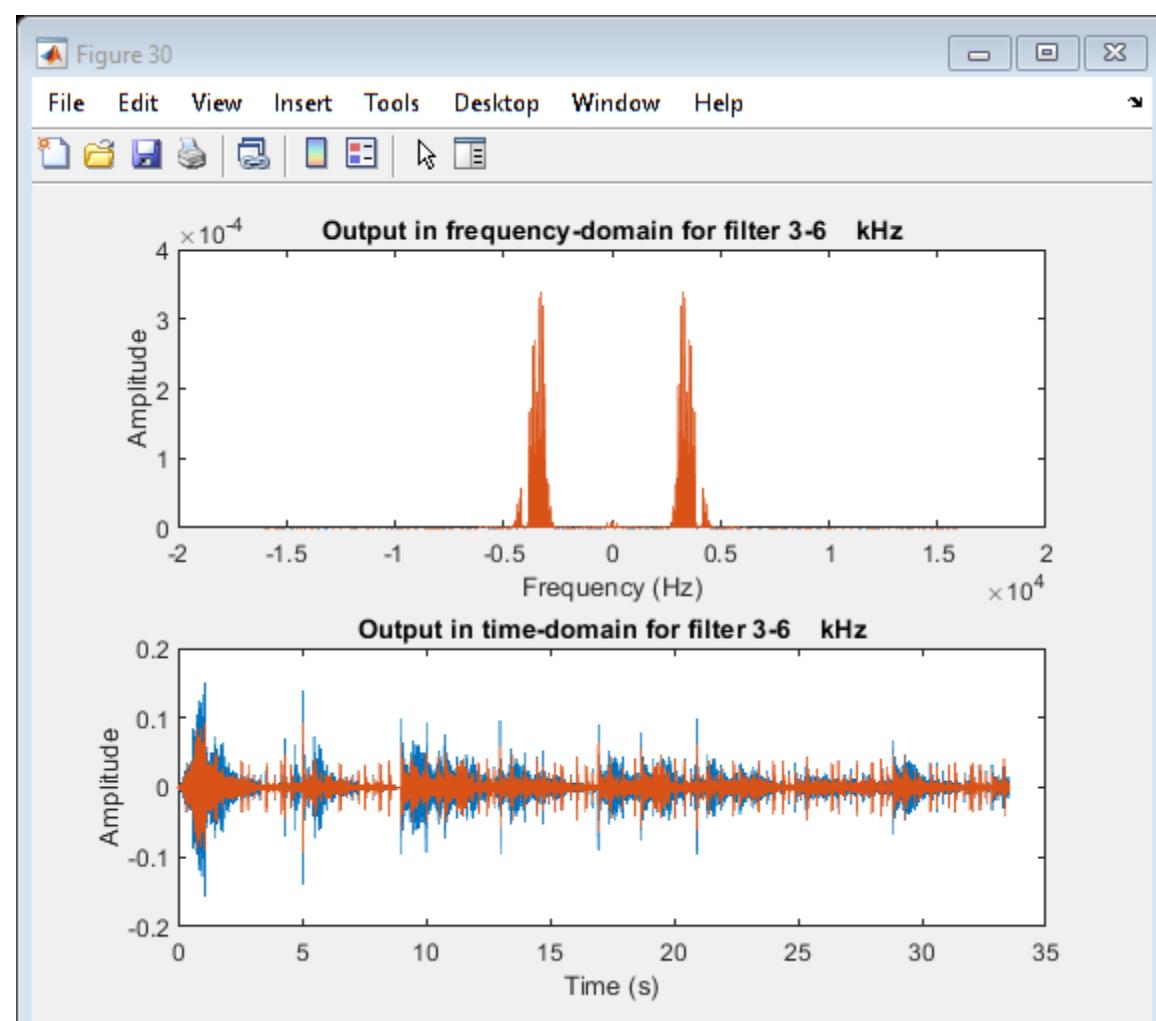
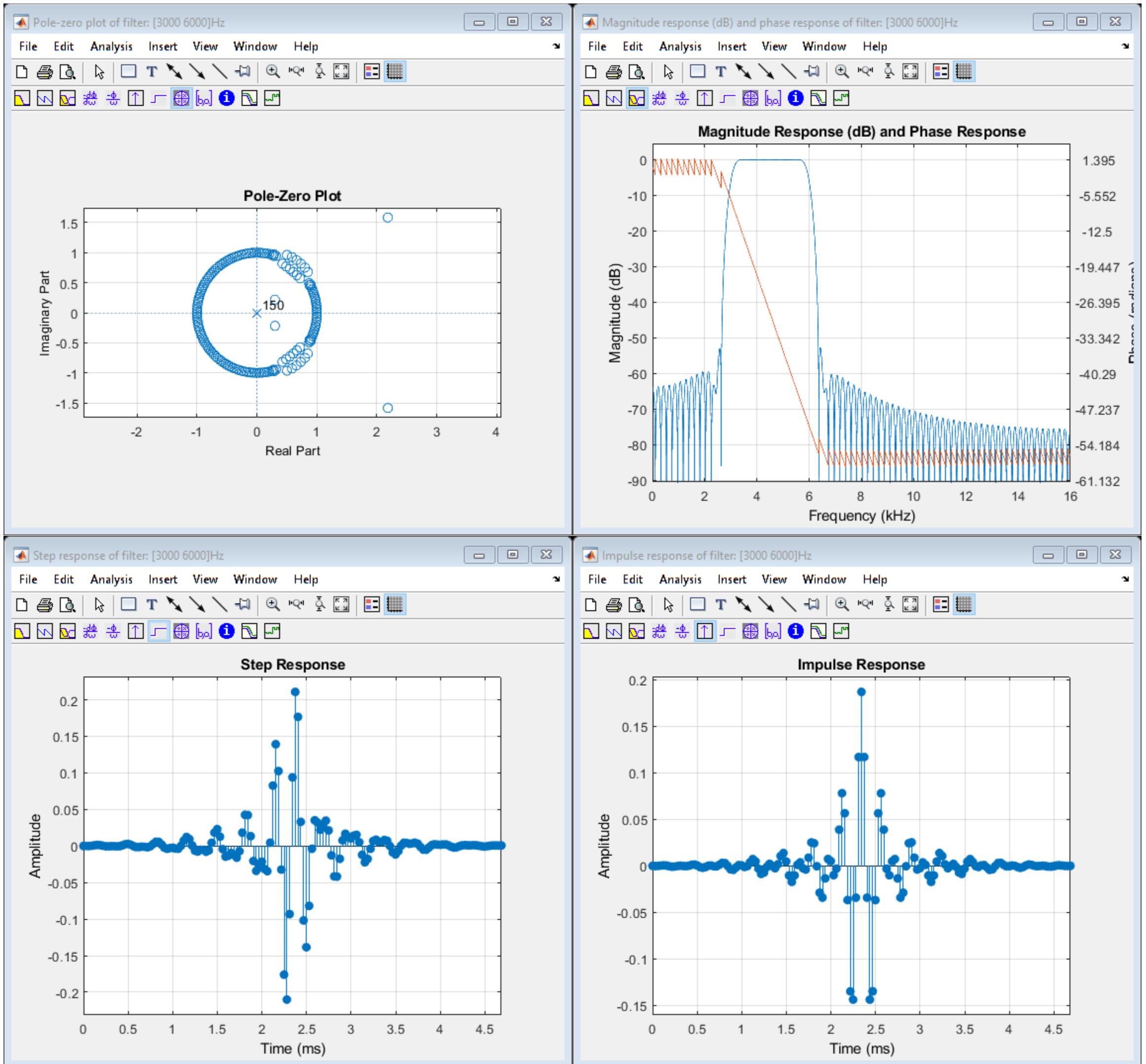
Sample run 1 (cont.):



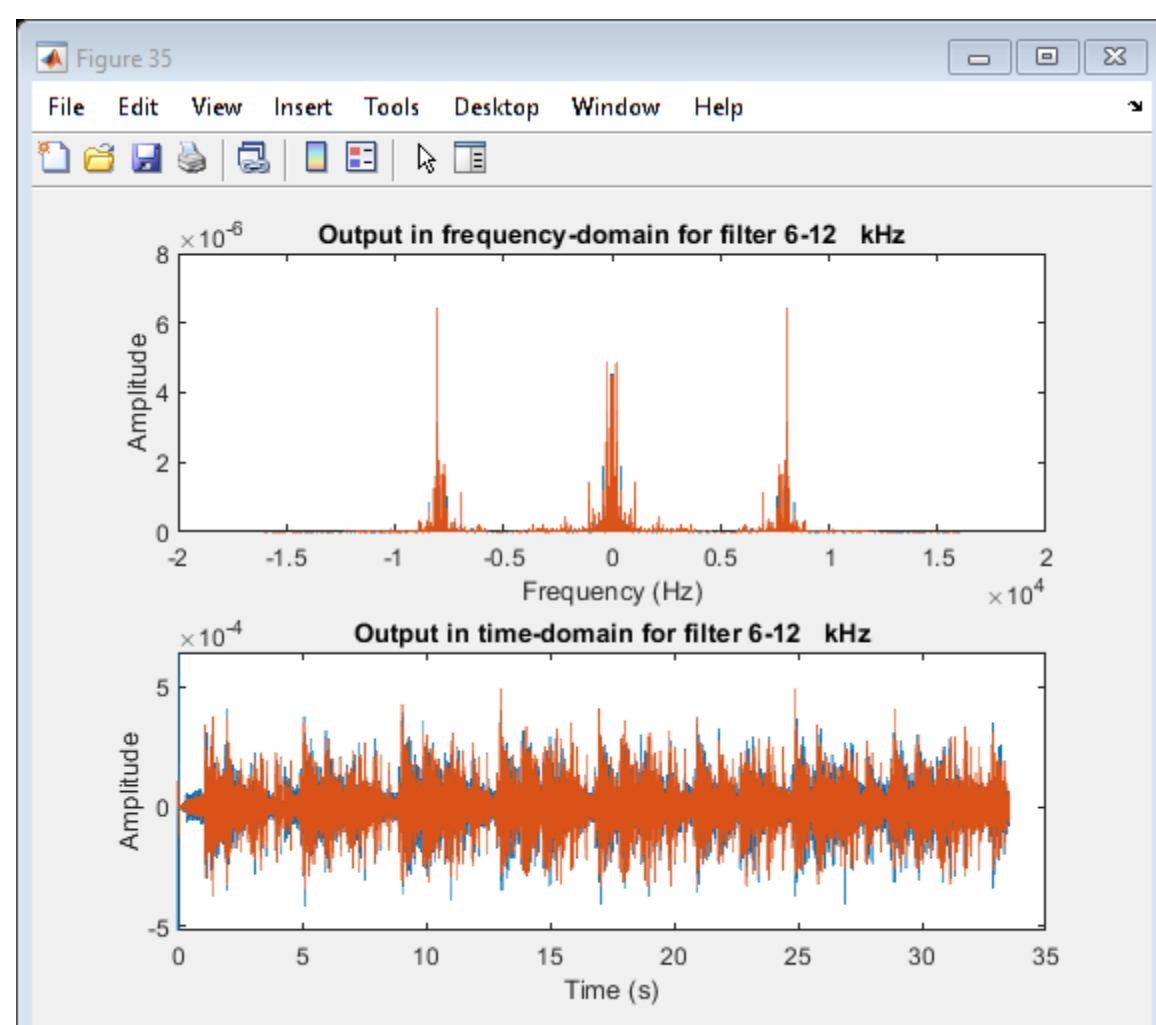
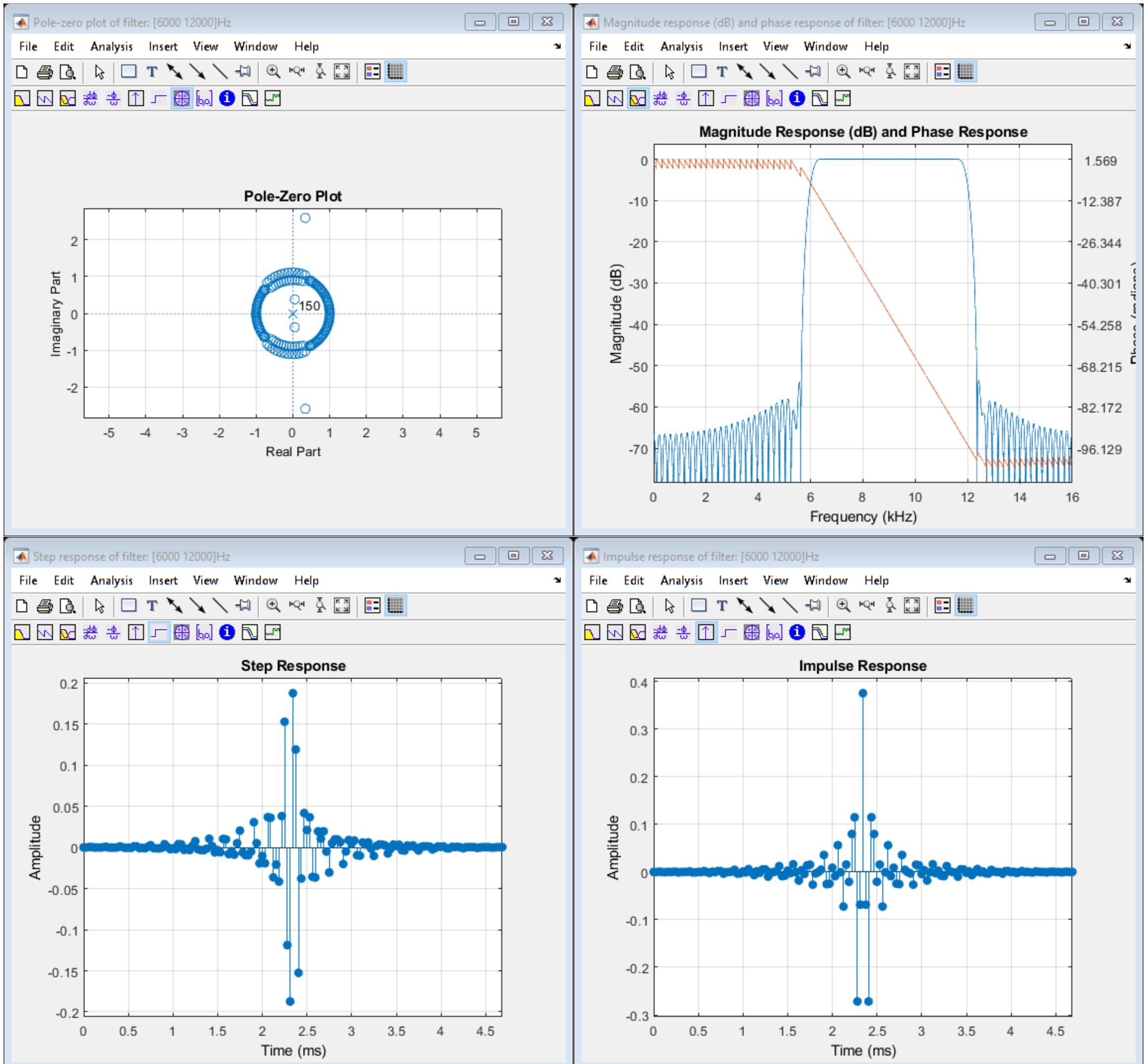
Sample run 1 (cont.):



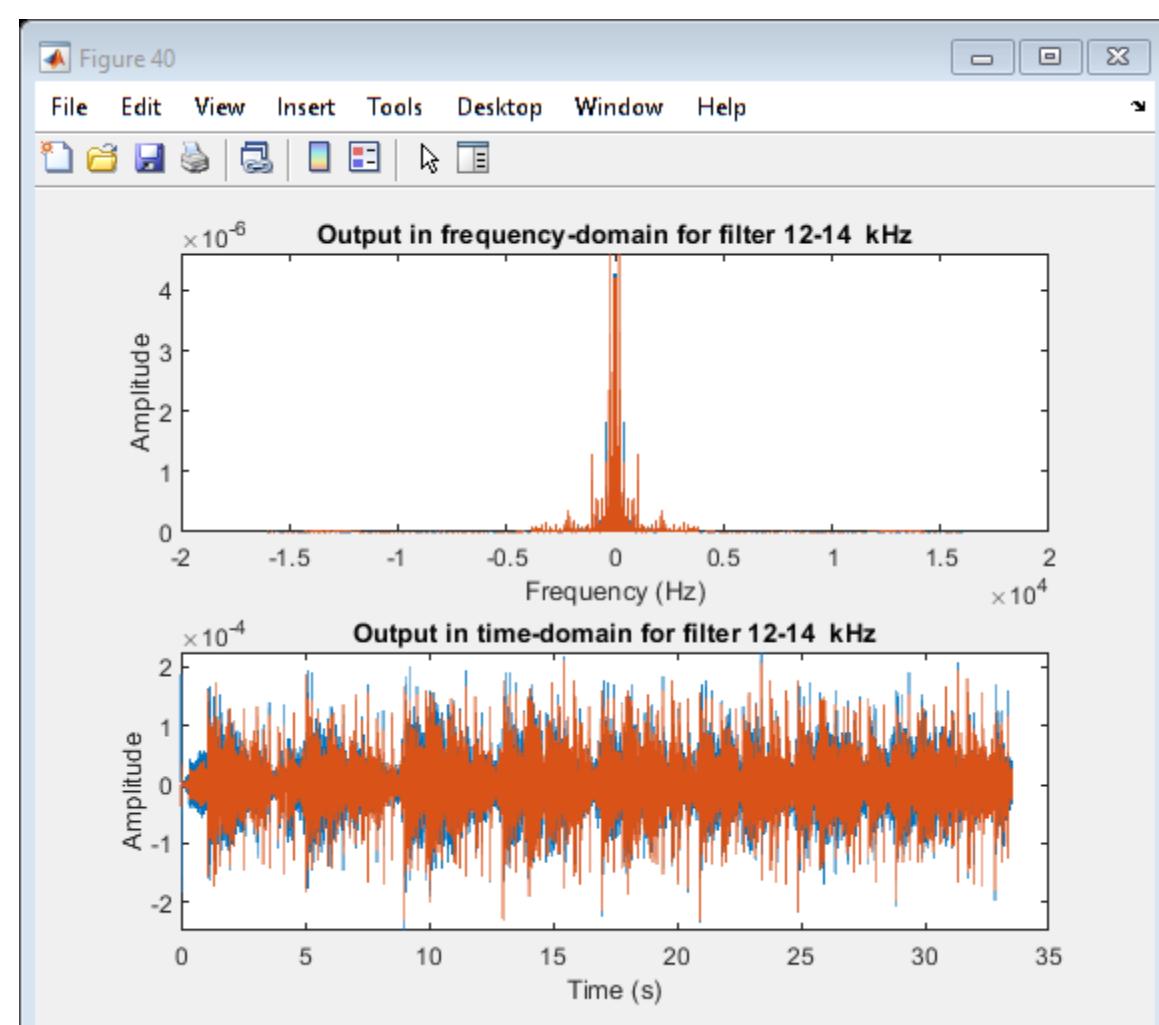
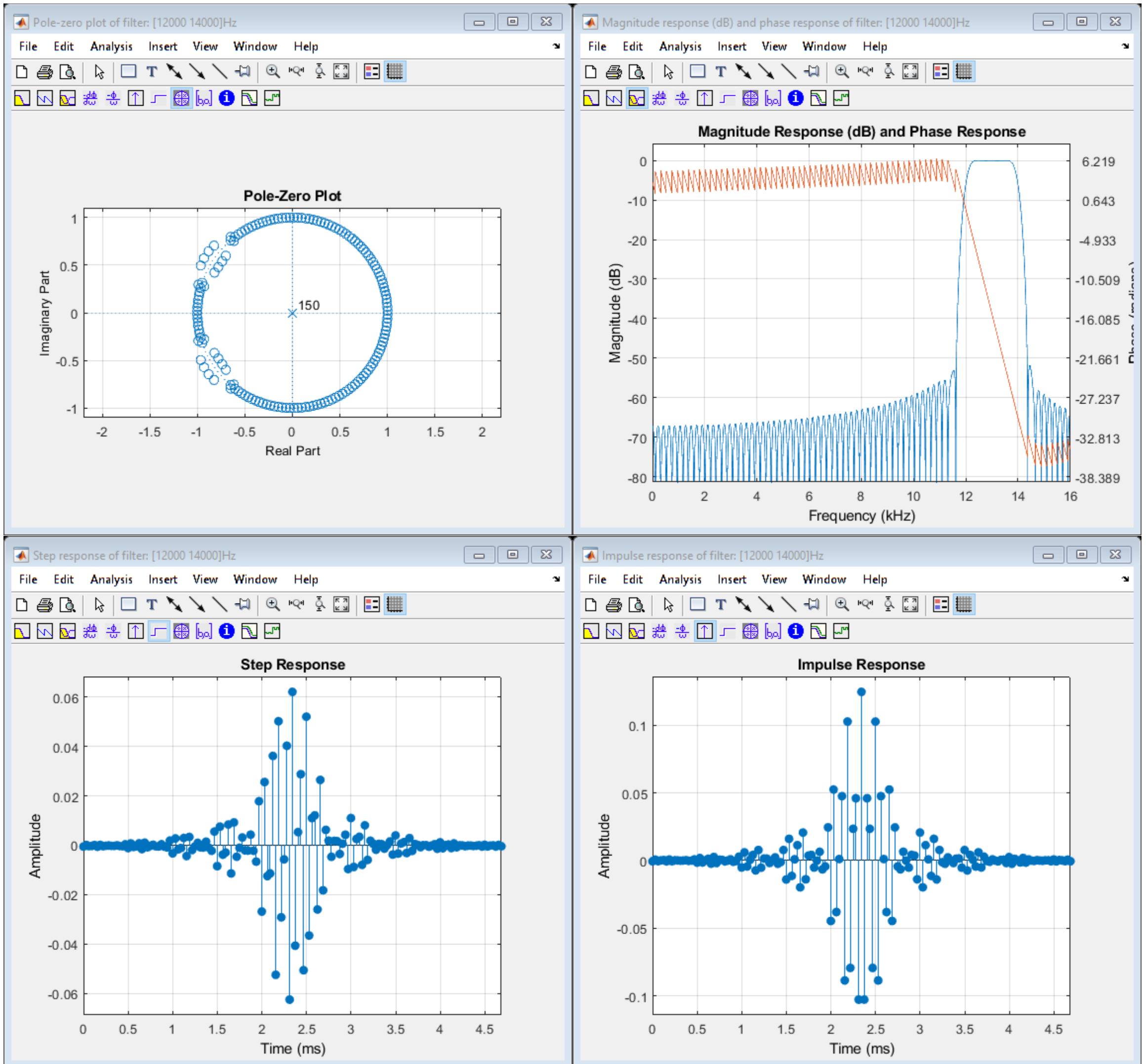
Sample run 1 (cont.):



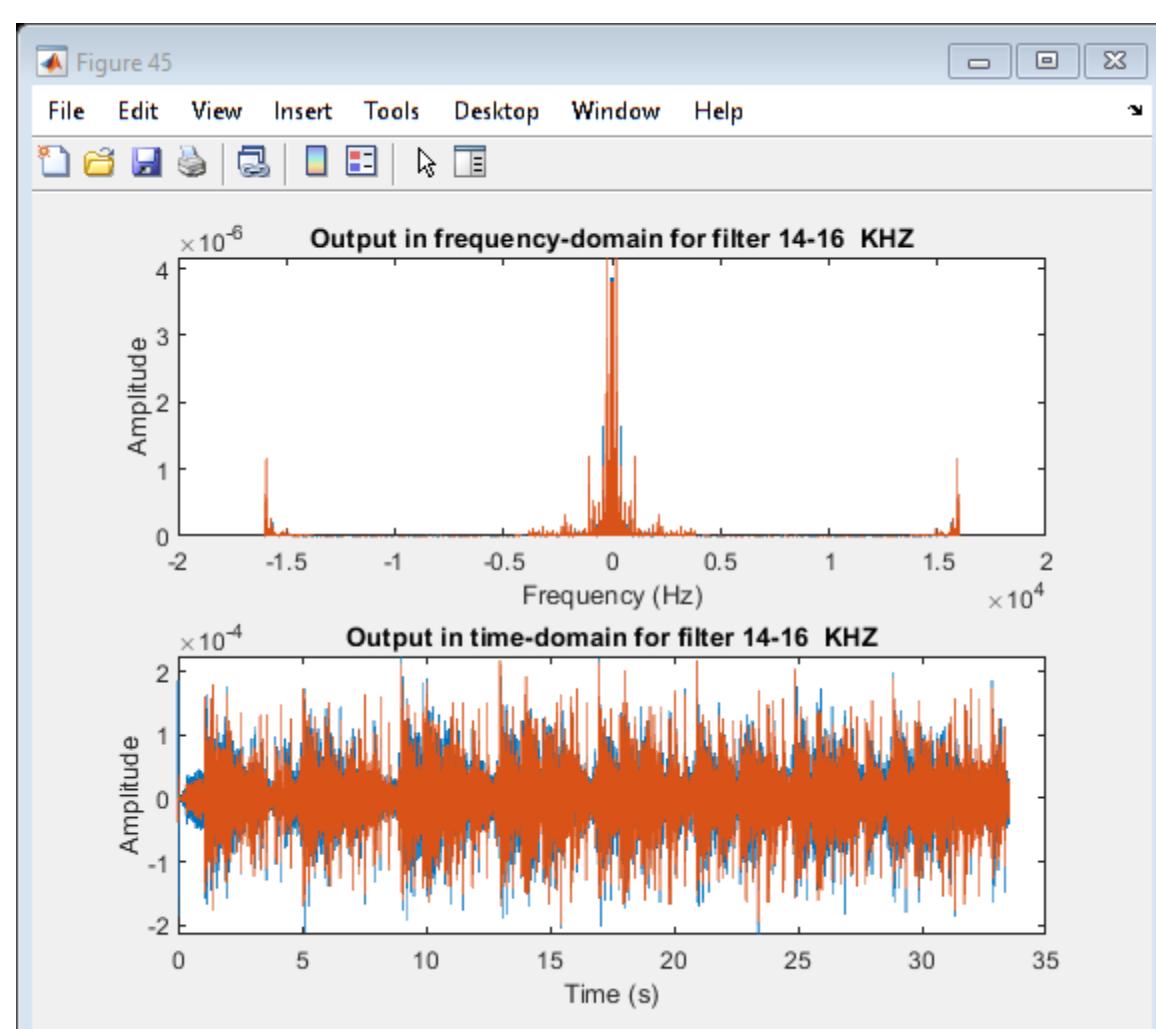
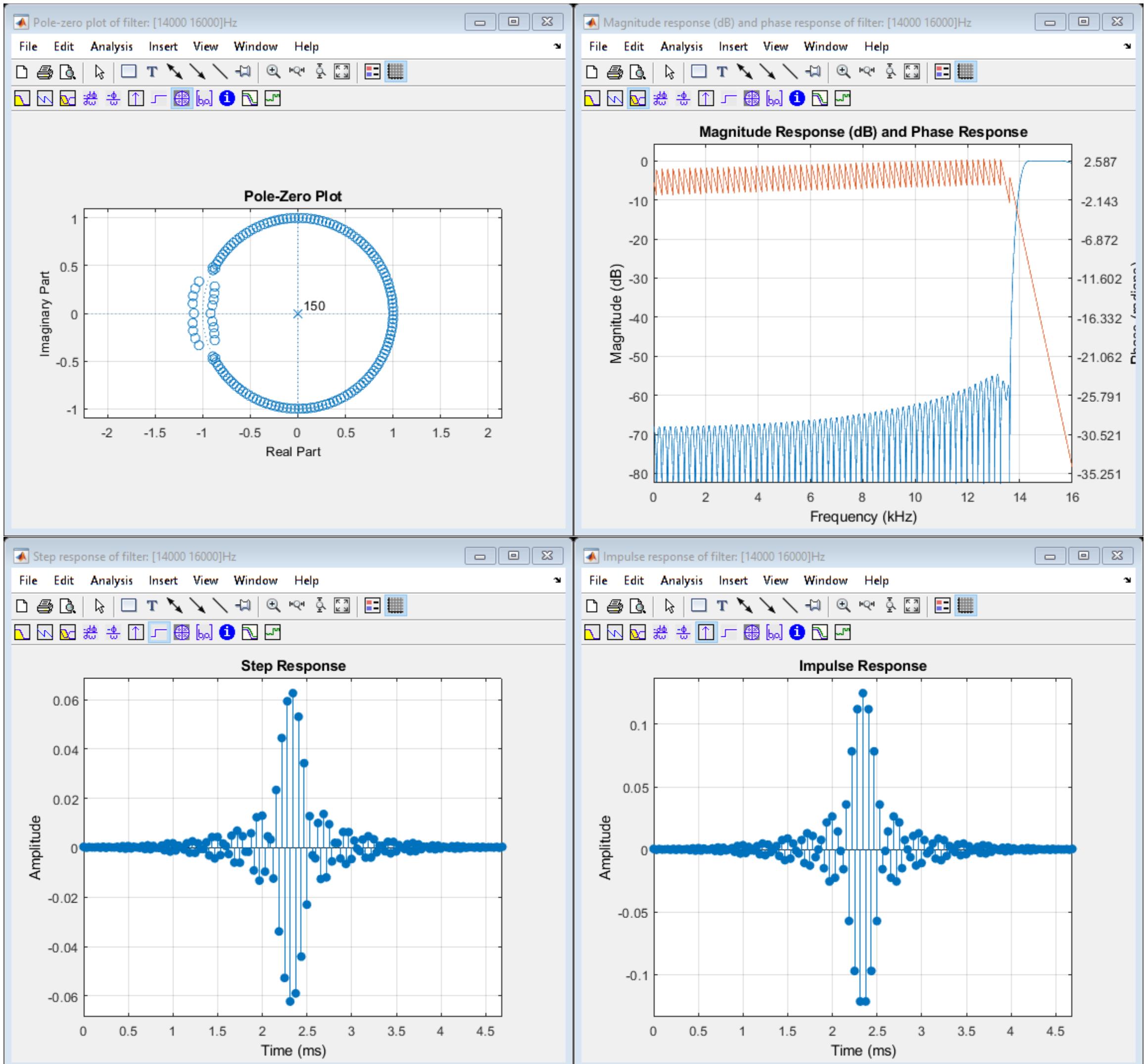
Sample run 1 (cont.):



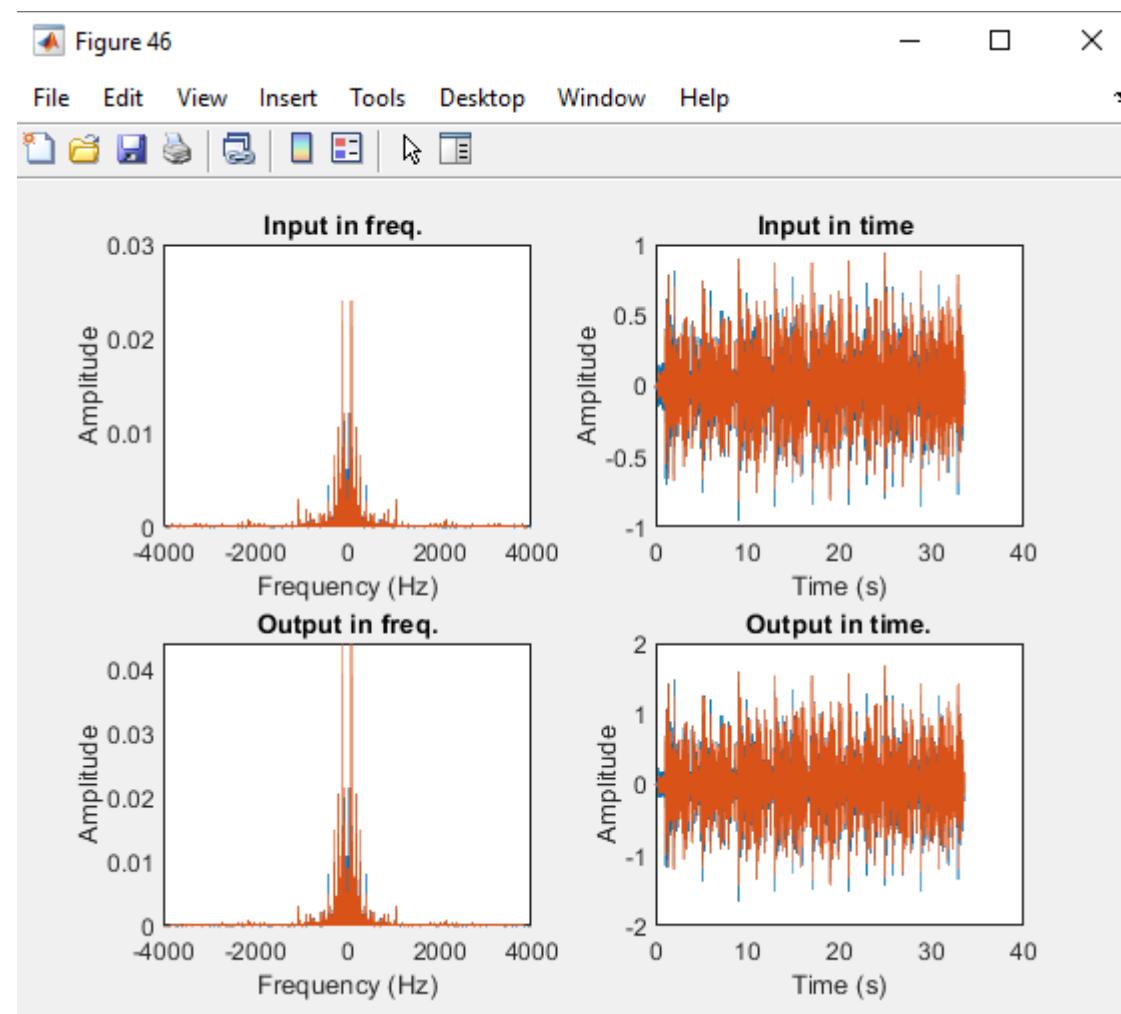
Sample run 1 (cont.):



Sample run 1 (cont.):



Sample run 1 (cont.):



Sample run 2 (Half sample rate):

Type of filter: iir

Specified gains for each of the nine filters: -15, -15, -15, 0, 0, 0, 15, 15, 15 dB

Specified output frequency: 11025 Hz

Output:

File information:

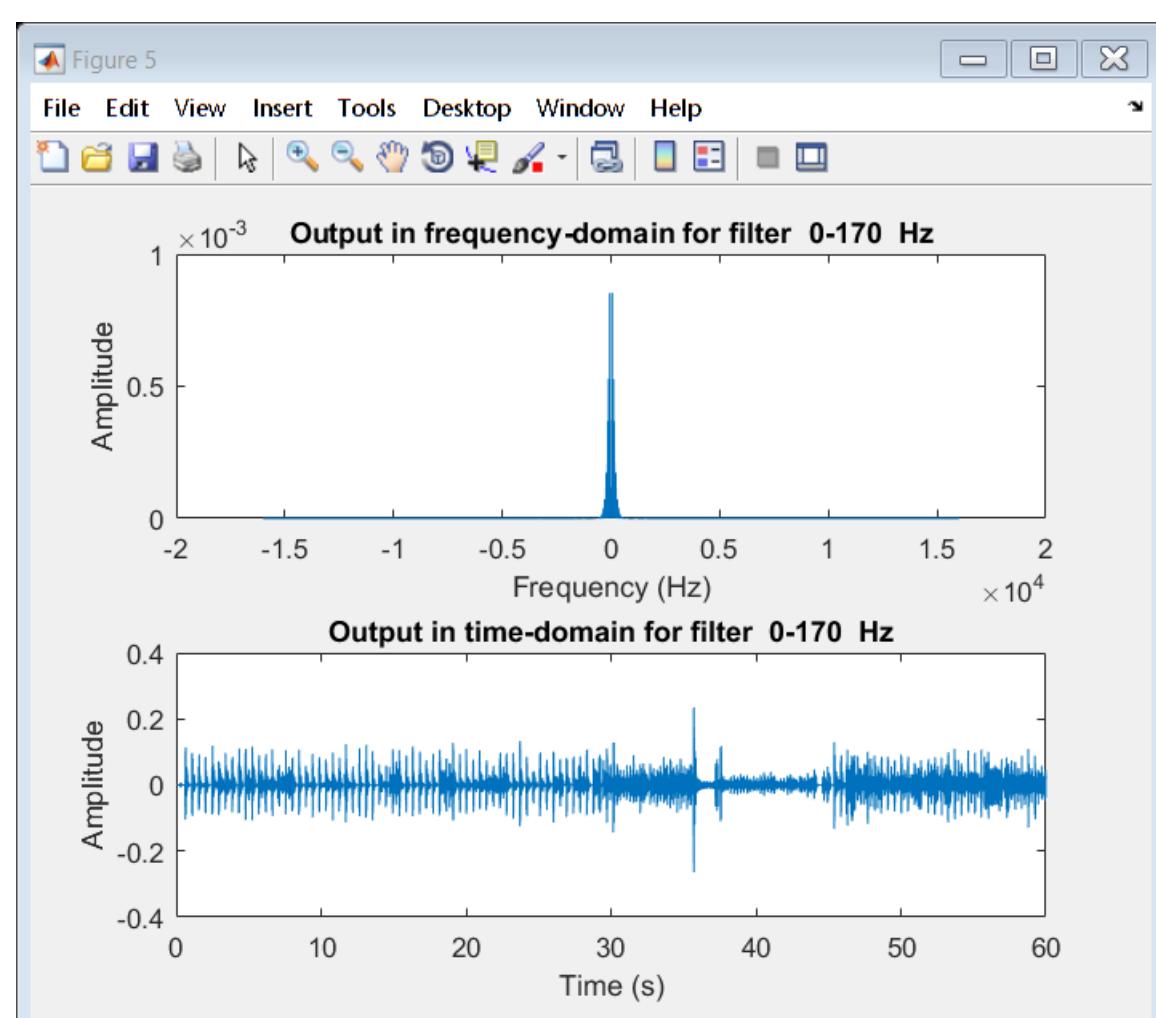
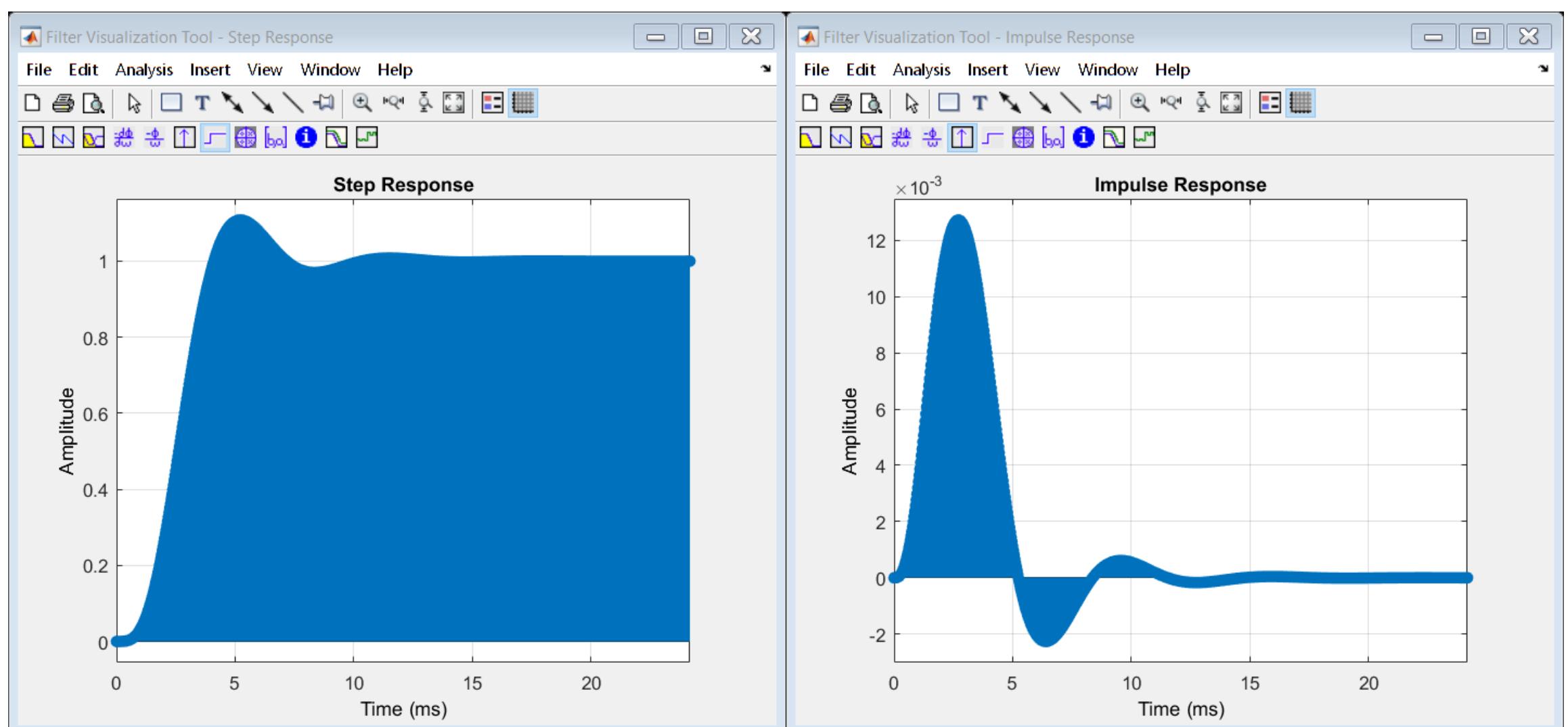
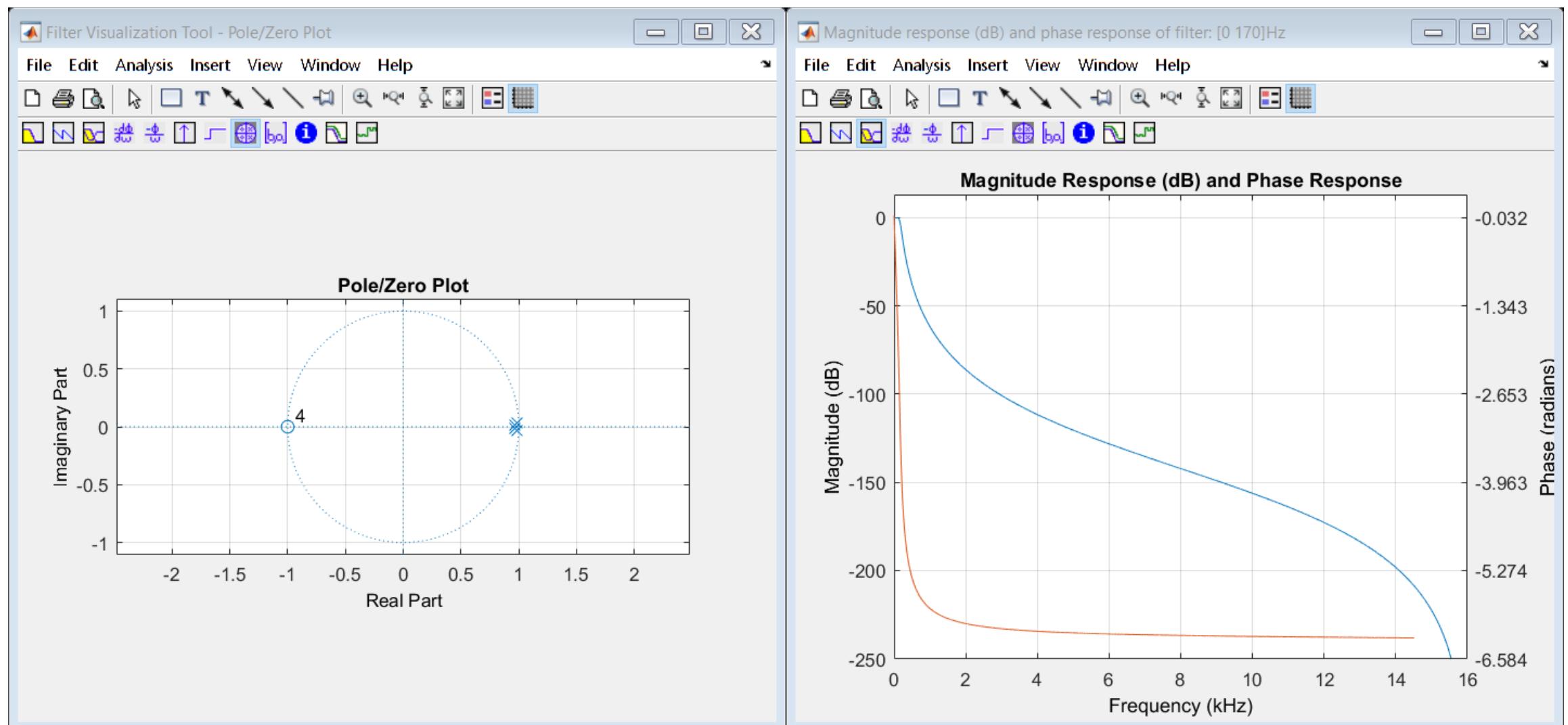
Path: D:\SSP\term 6\DSP\CantinaBand60.wav

Data dimensions: [1323000 1]

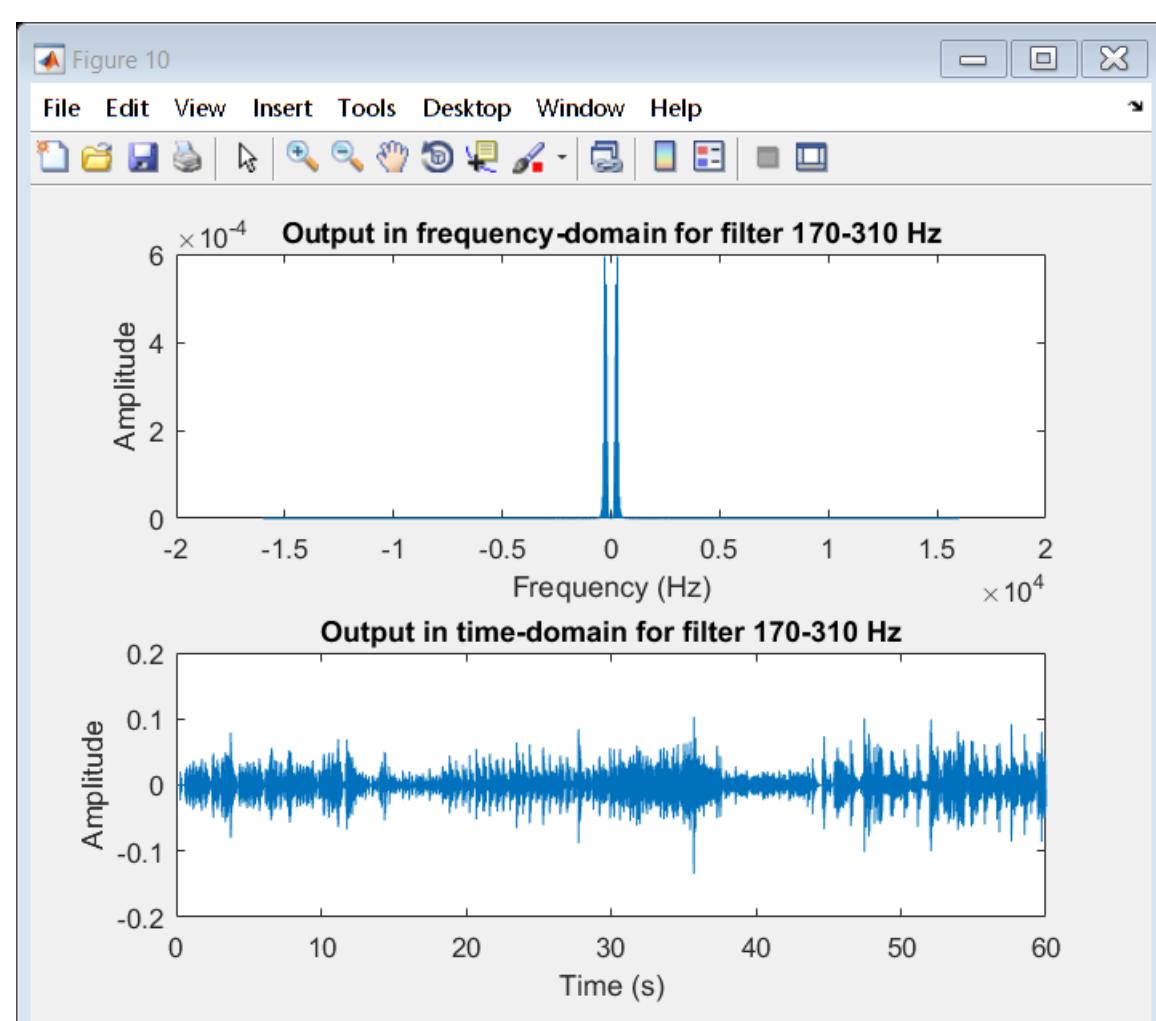
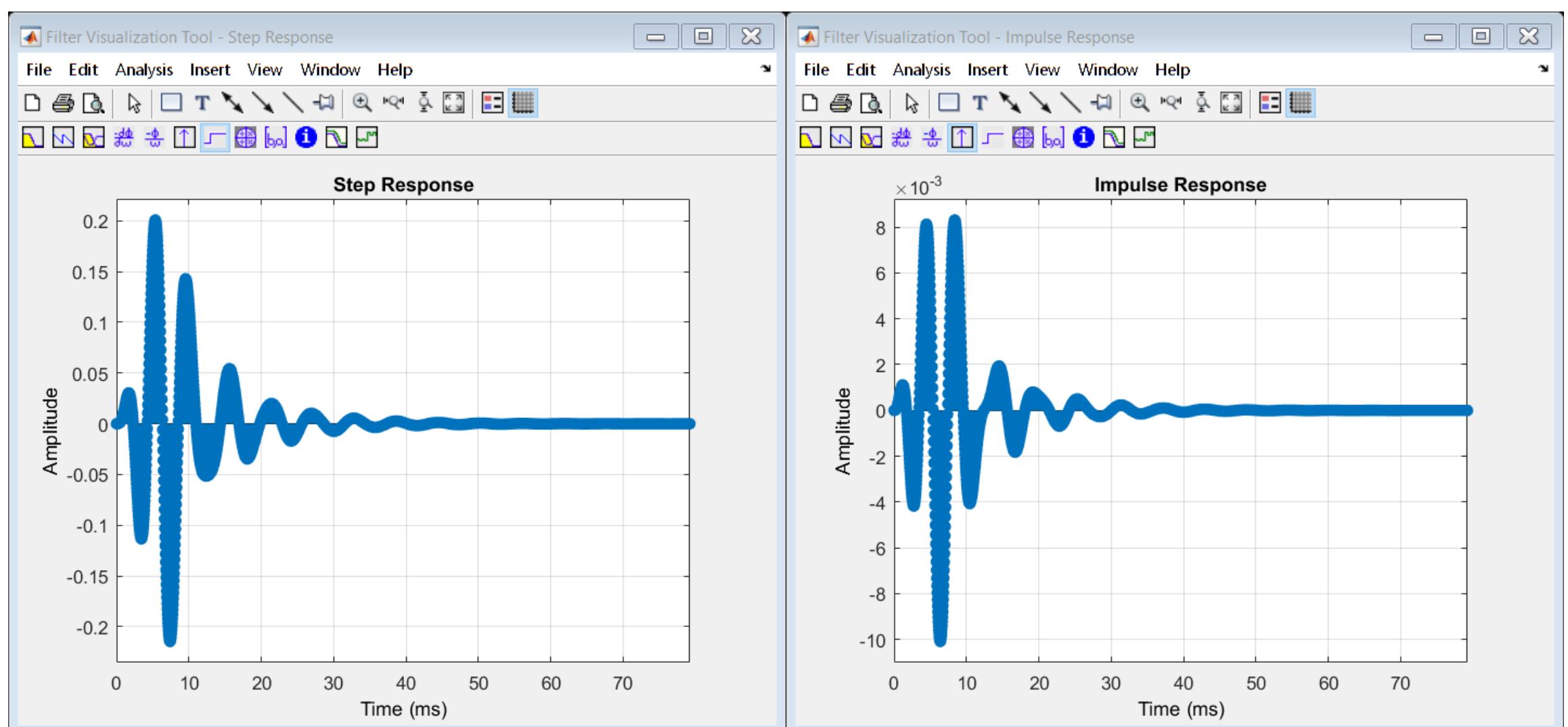
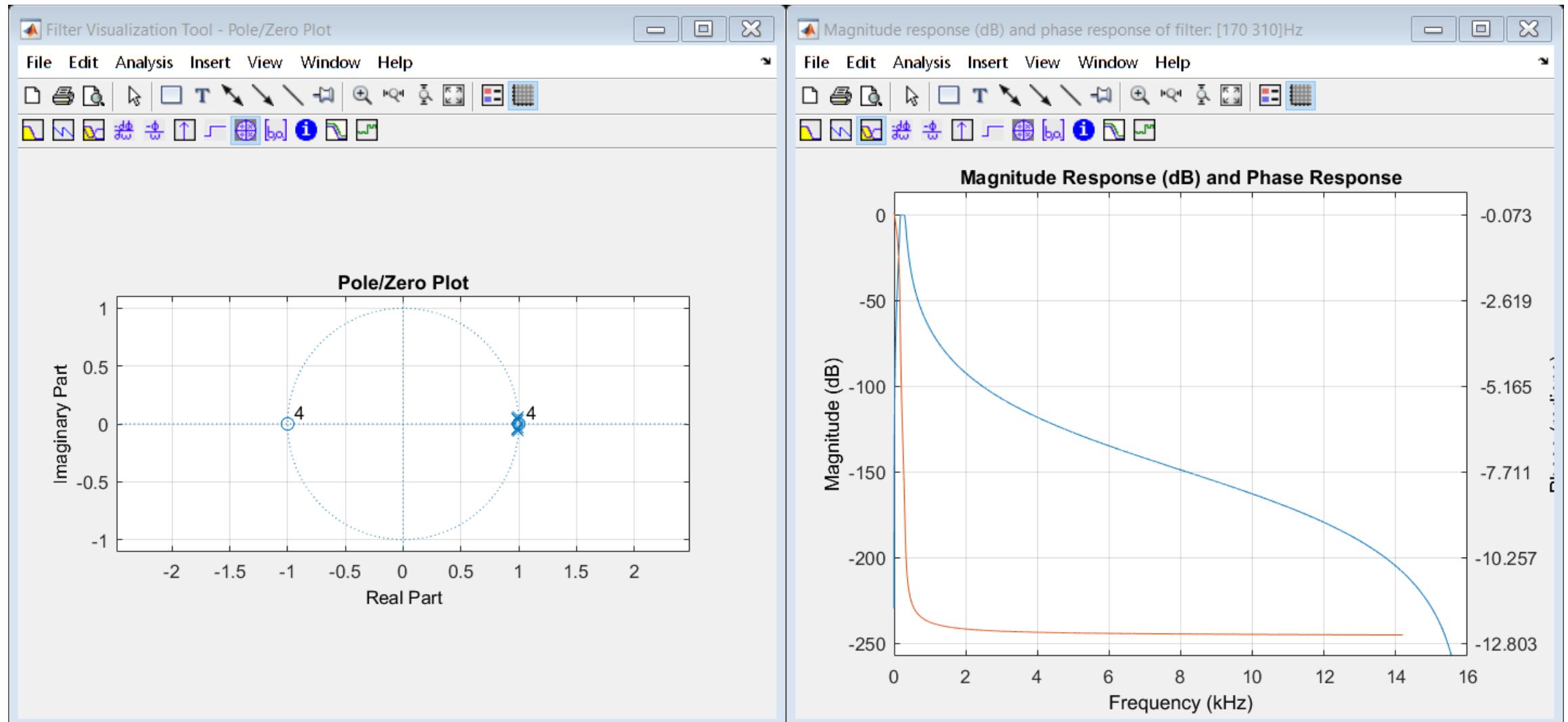
Frequency: 22050

The gain of iir filter : 0-170 Hz is 0.000000 , Order is 4
The gain of iir filter : 170-310 Hz is 0.000000 , Order is 8
The gain of iir filter : 310-600 Hz is 0.000001 , Order is 8
The gain of iir filter : 0.6-1 kHz is 0.000002 , Order is 8
The gain of iir filter : 1-3 kHz is 0.000931 , Order is 8
The gain of iir filter : 3-6 kHz is 0.003860 , Order is 8
The gain of iir filter : 6-12 kHz is 0.037890 , Order is 8
The gain of iir filter : 12-14 kHz is 0.000931 , Order is 8
The gain of iir filter : 14-16 kHz is 0.000931 , Order is 8

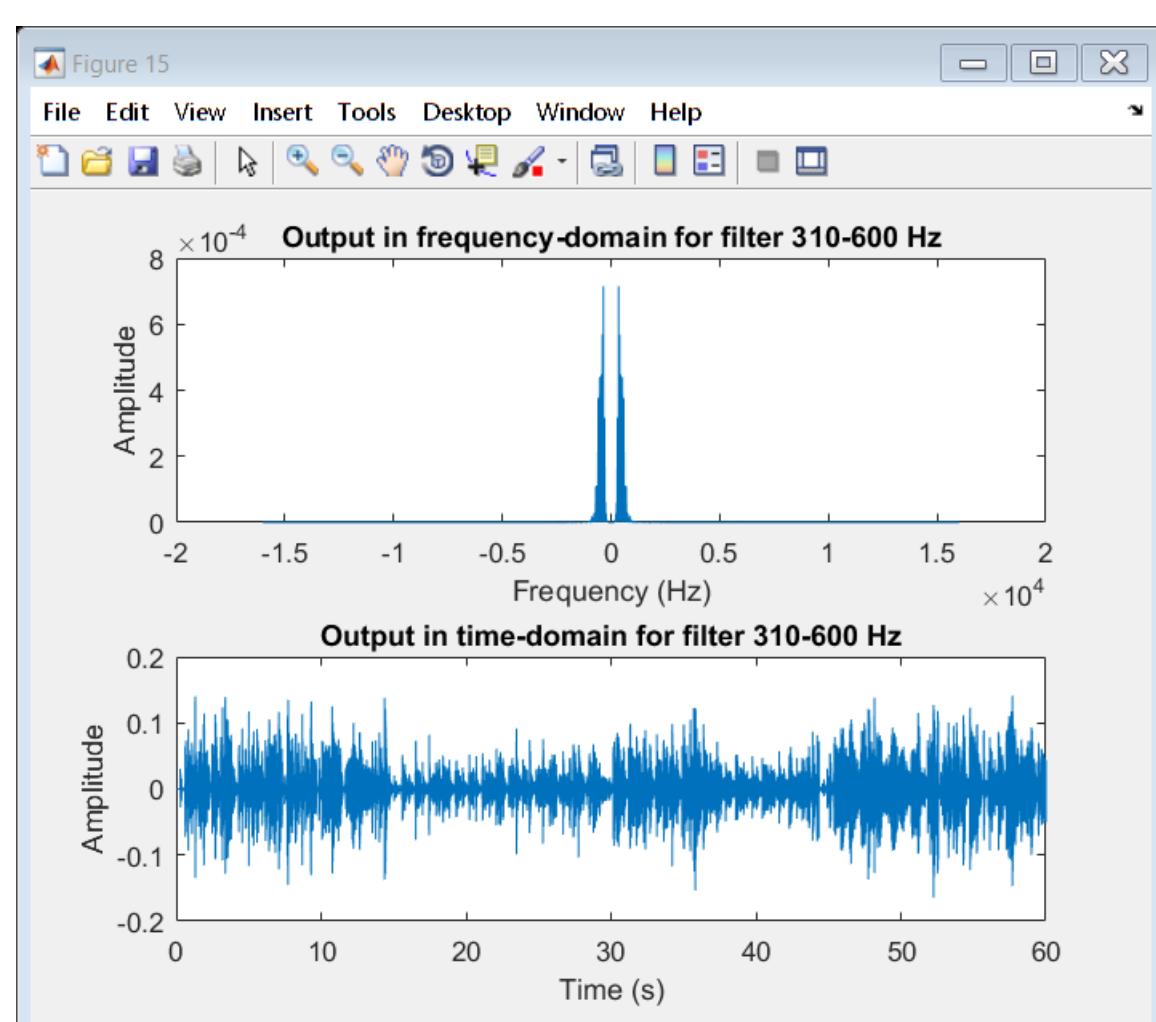
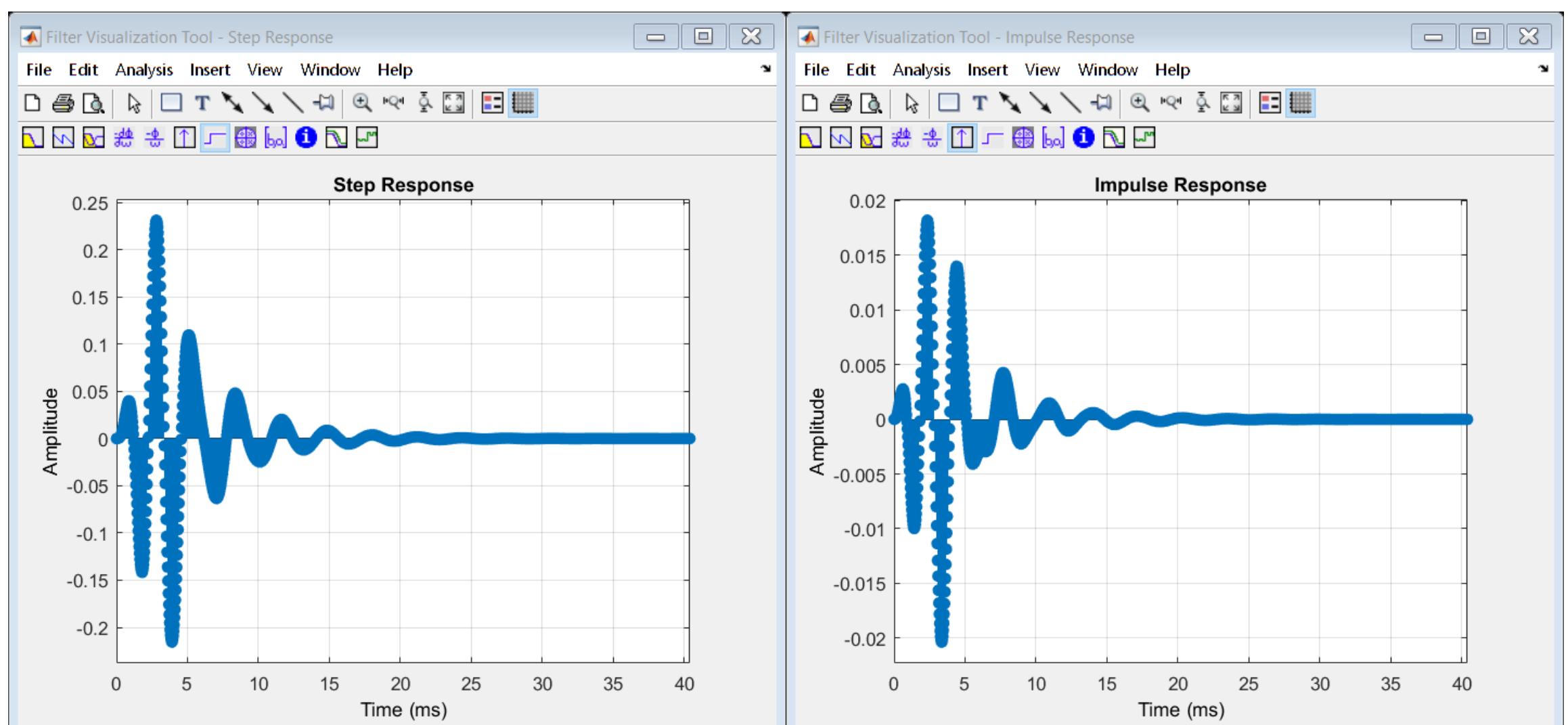
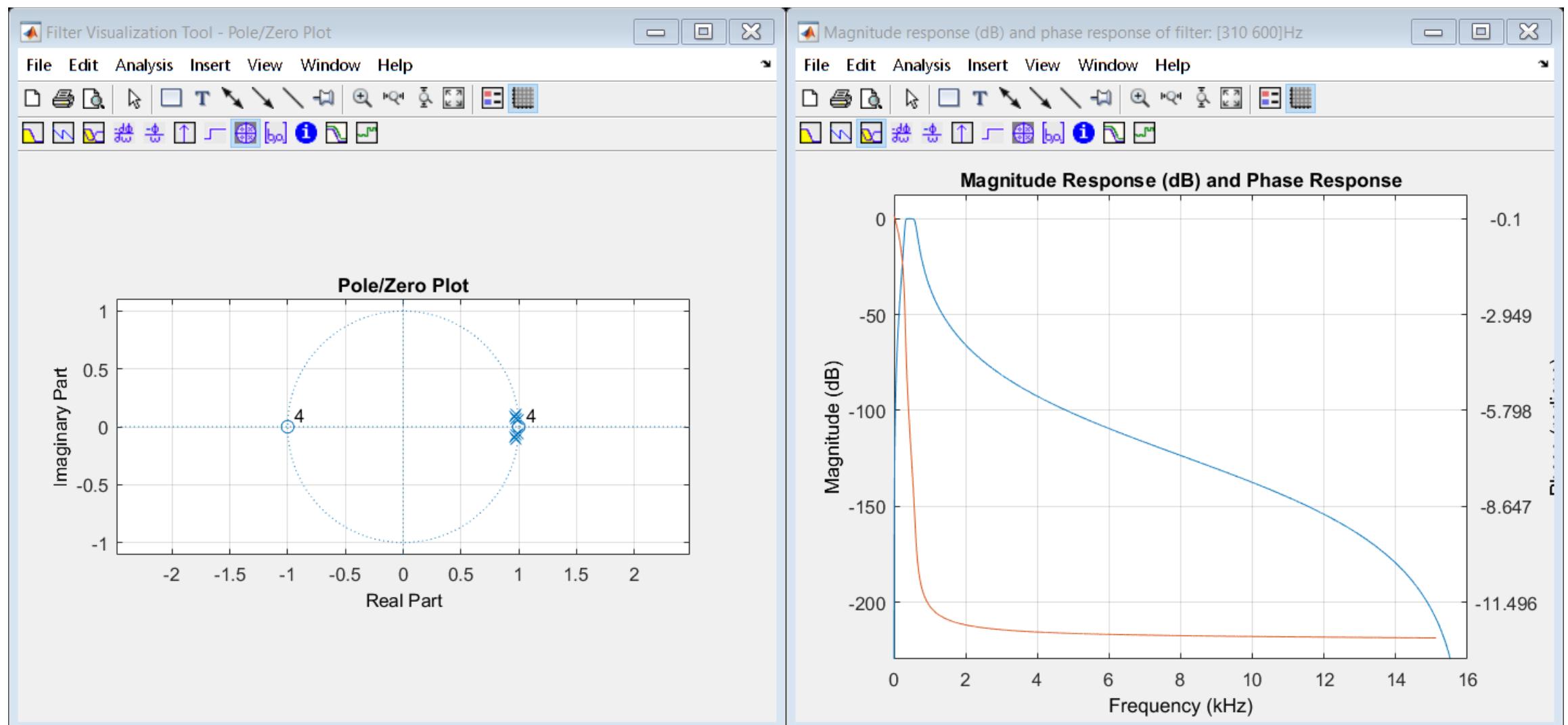
Sample run 2 (cont.):



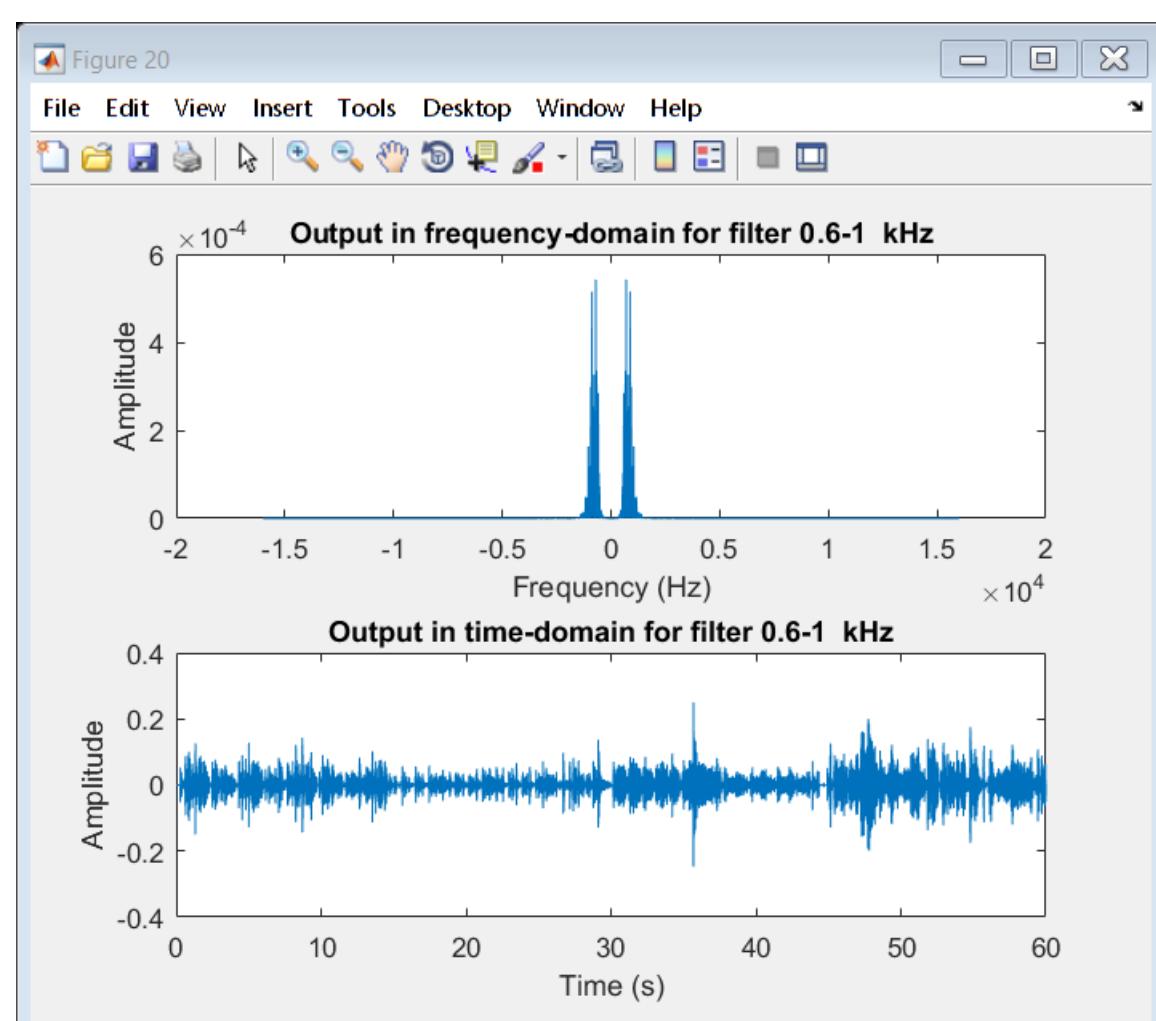
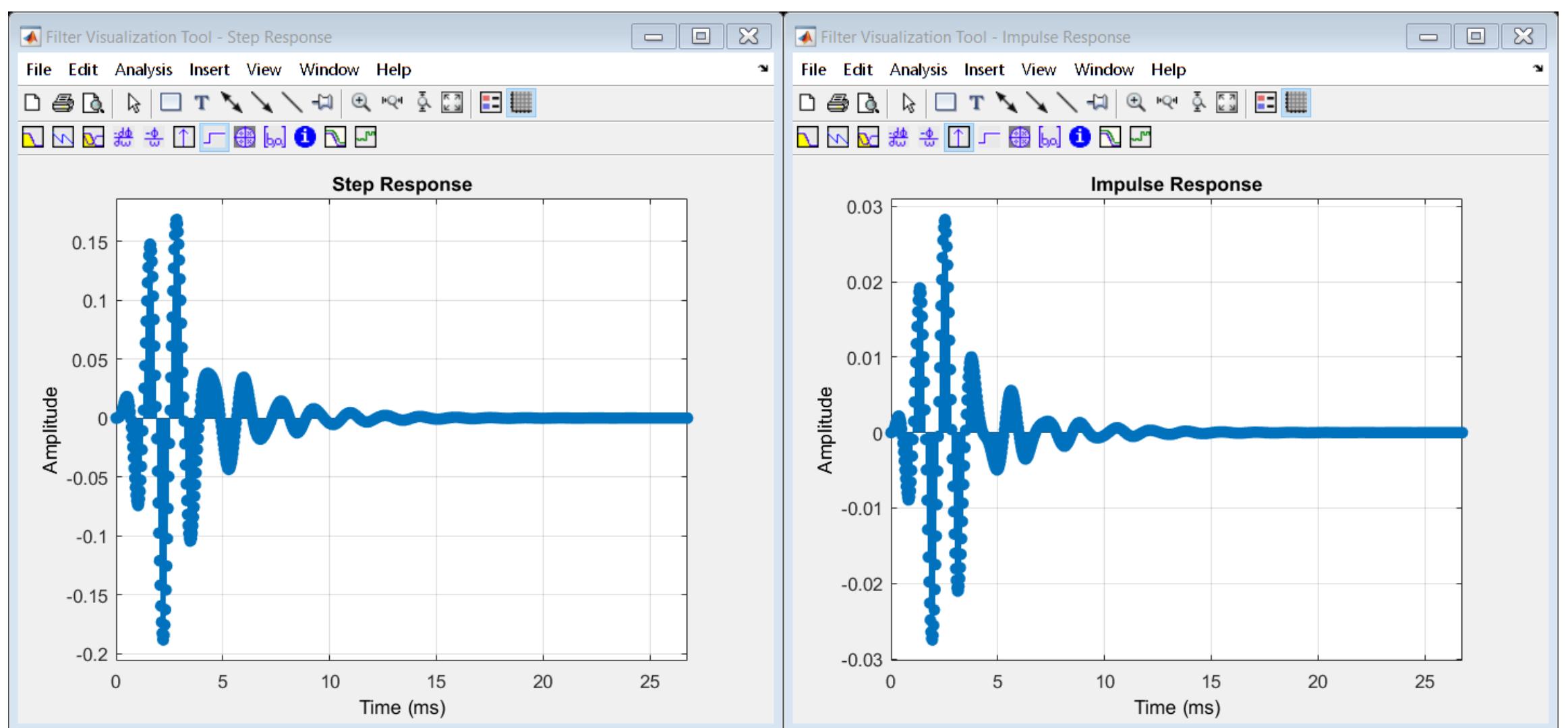
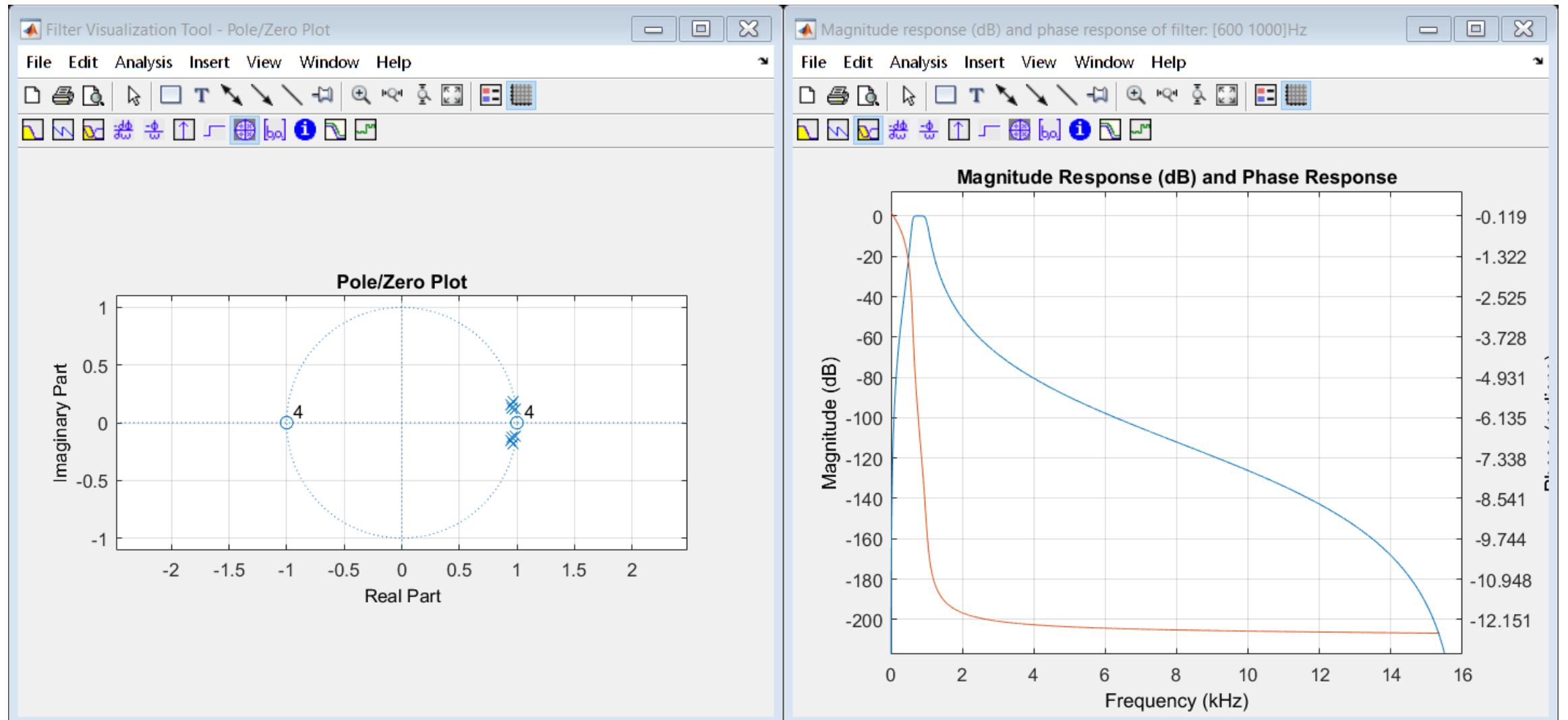
Sample run 2 (cont.):



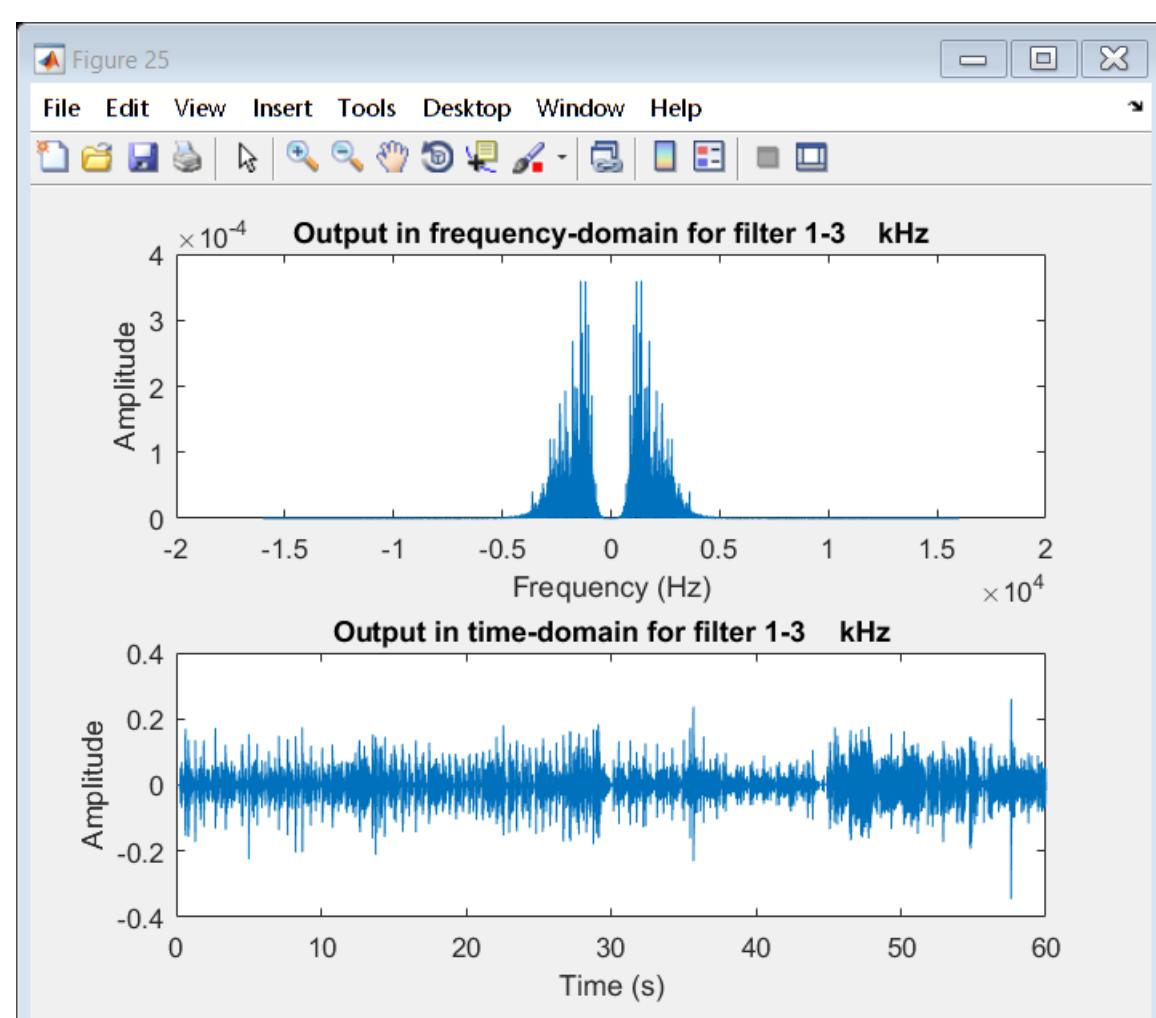
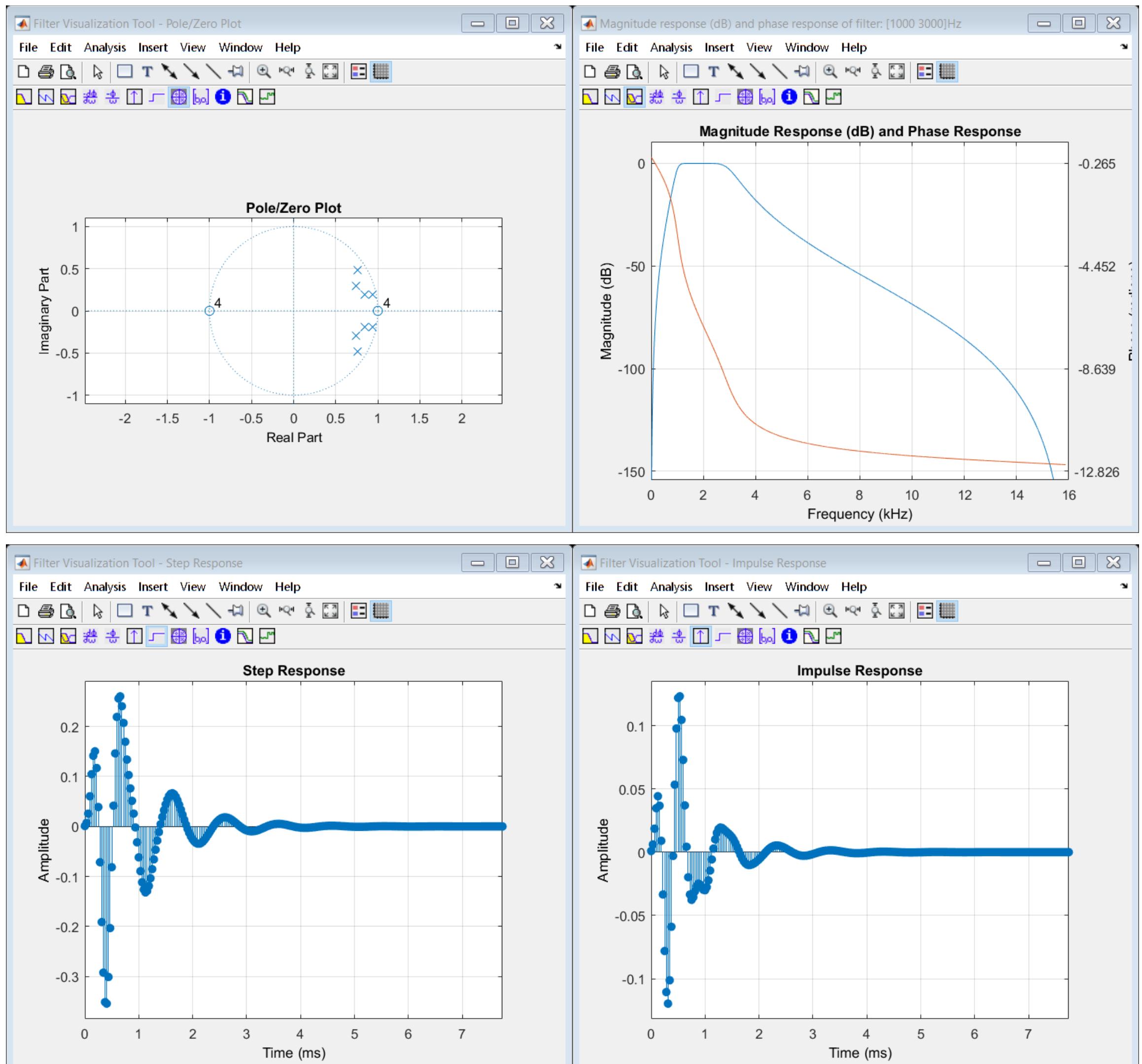
Sample run 2 (cont.):



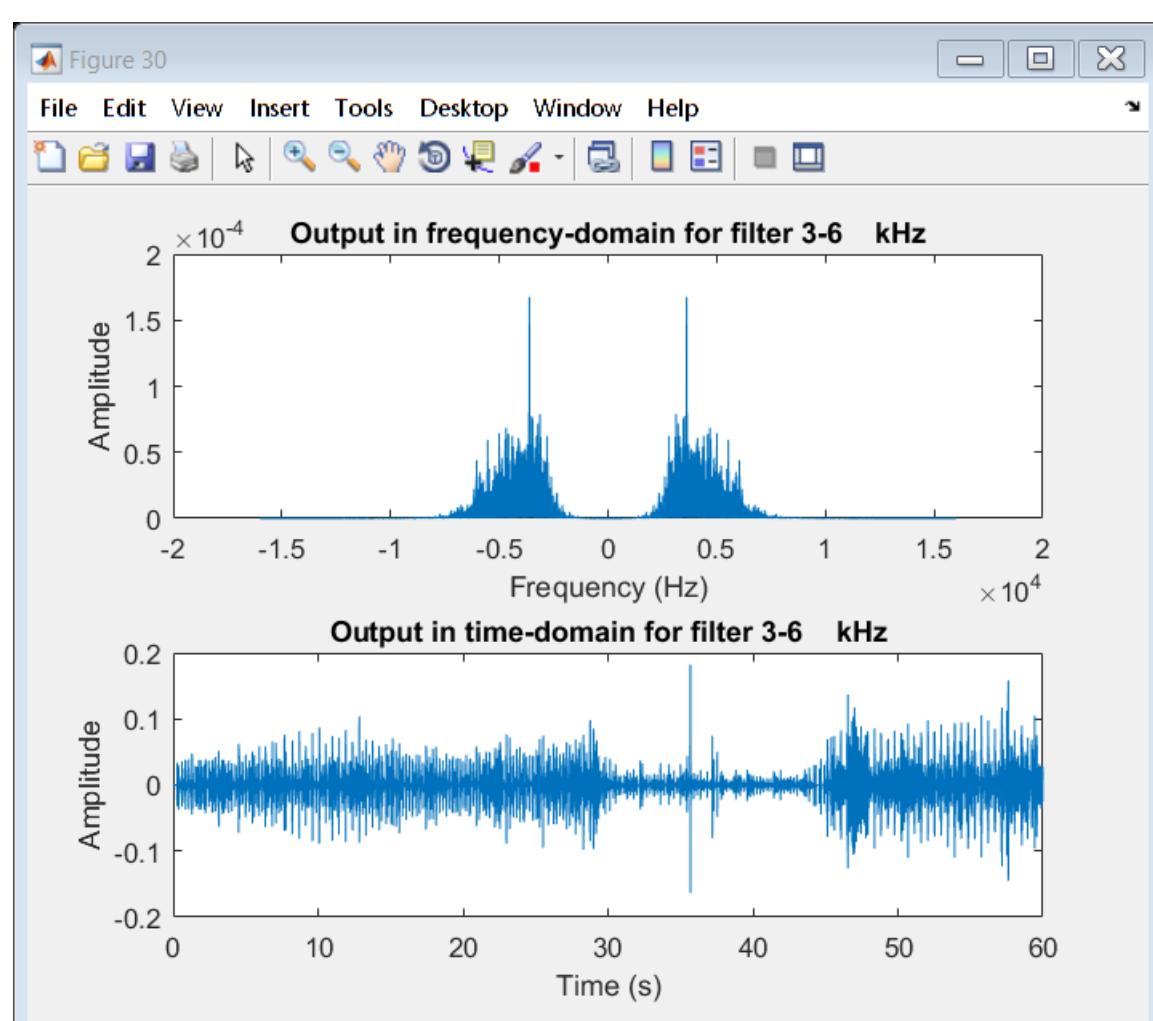
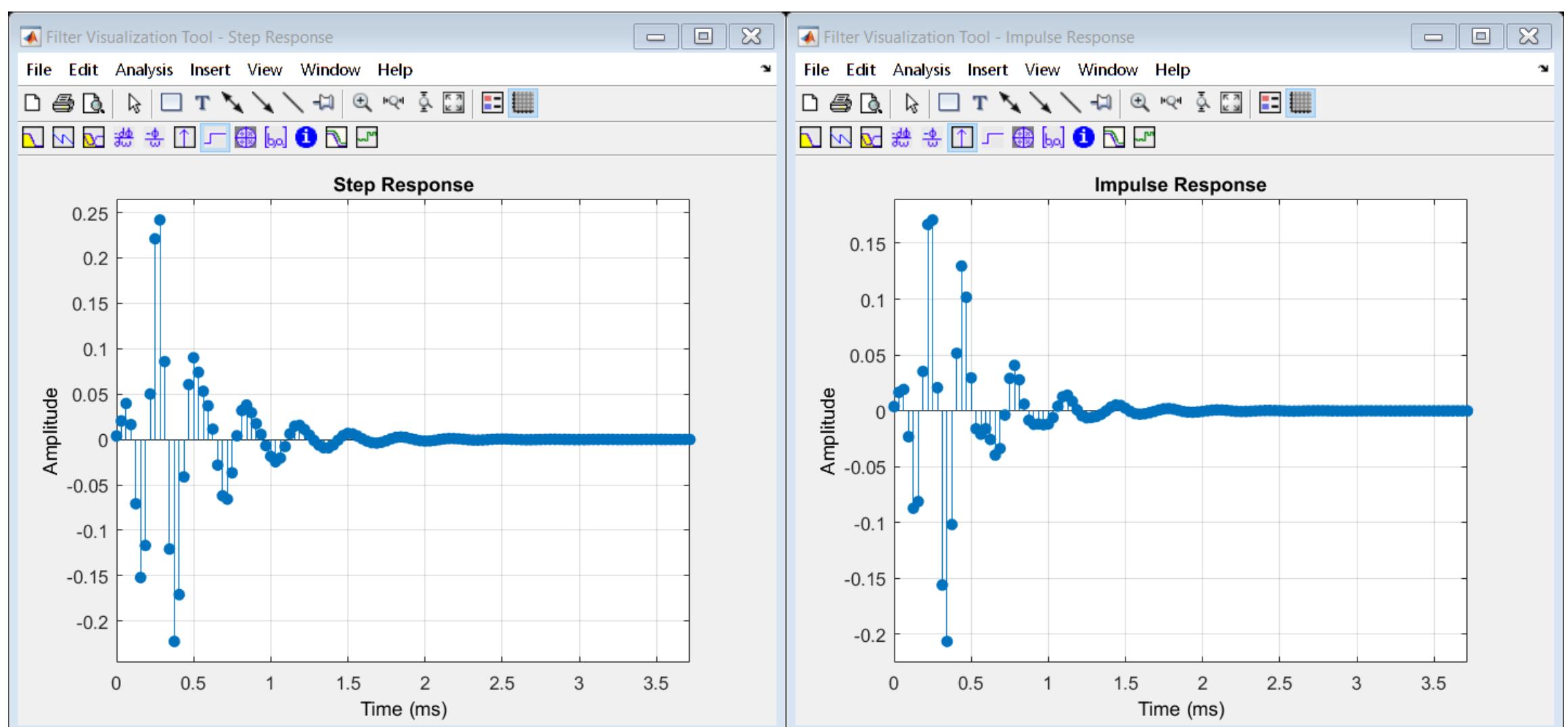
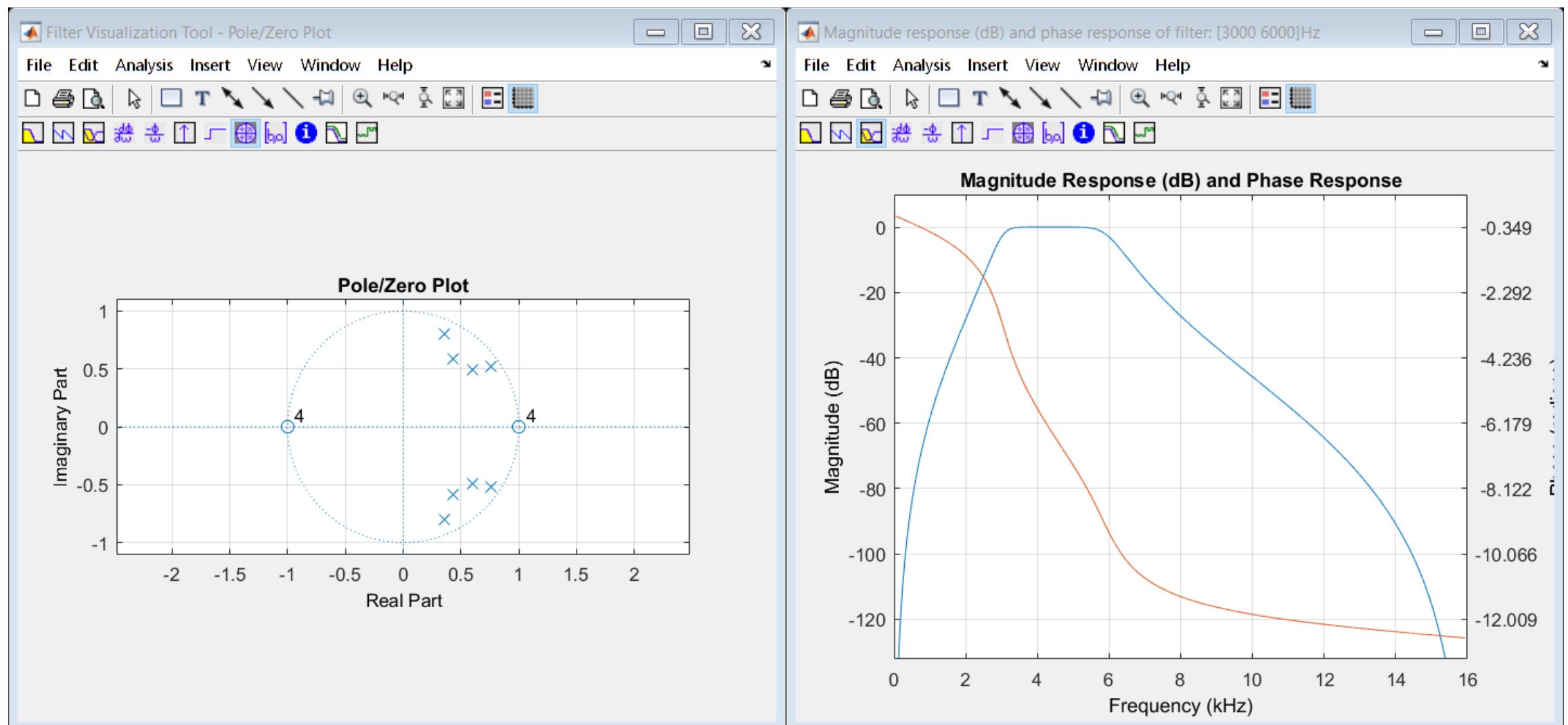
Sample run 2 (cont.):



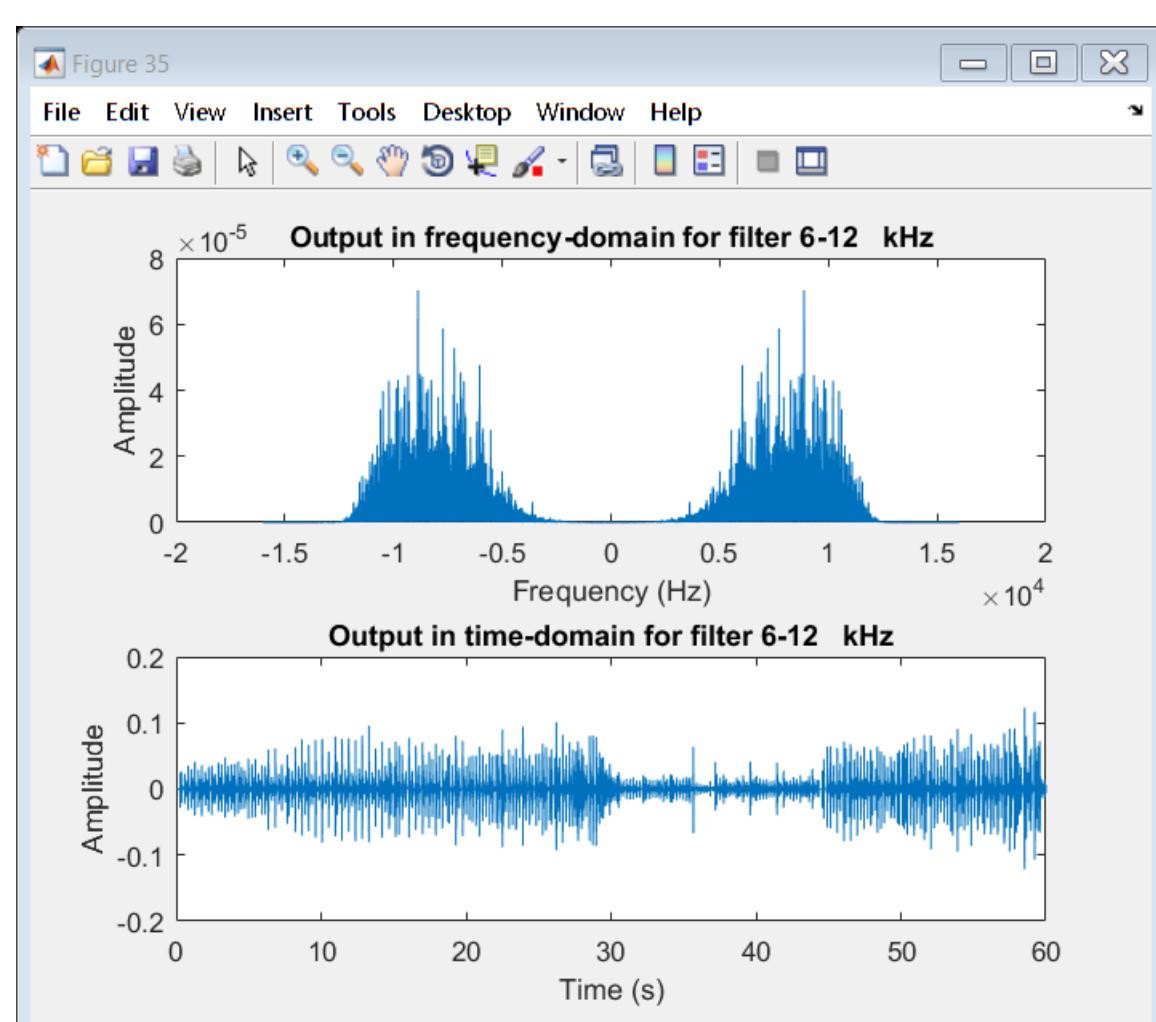
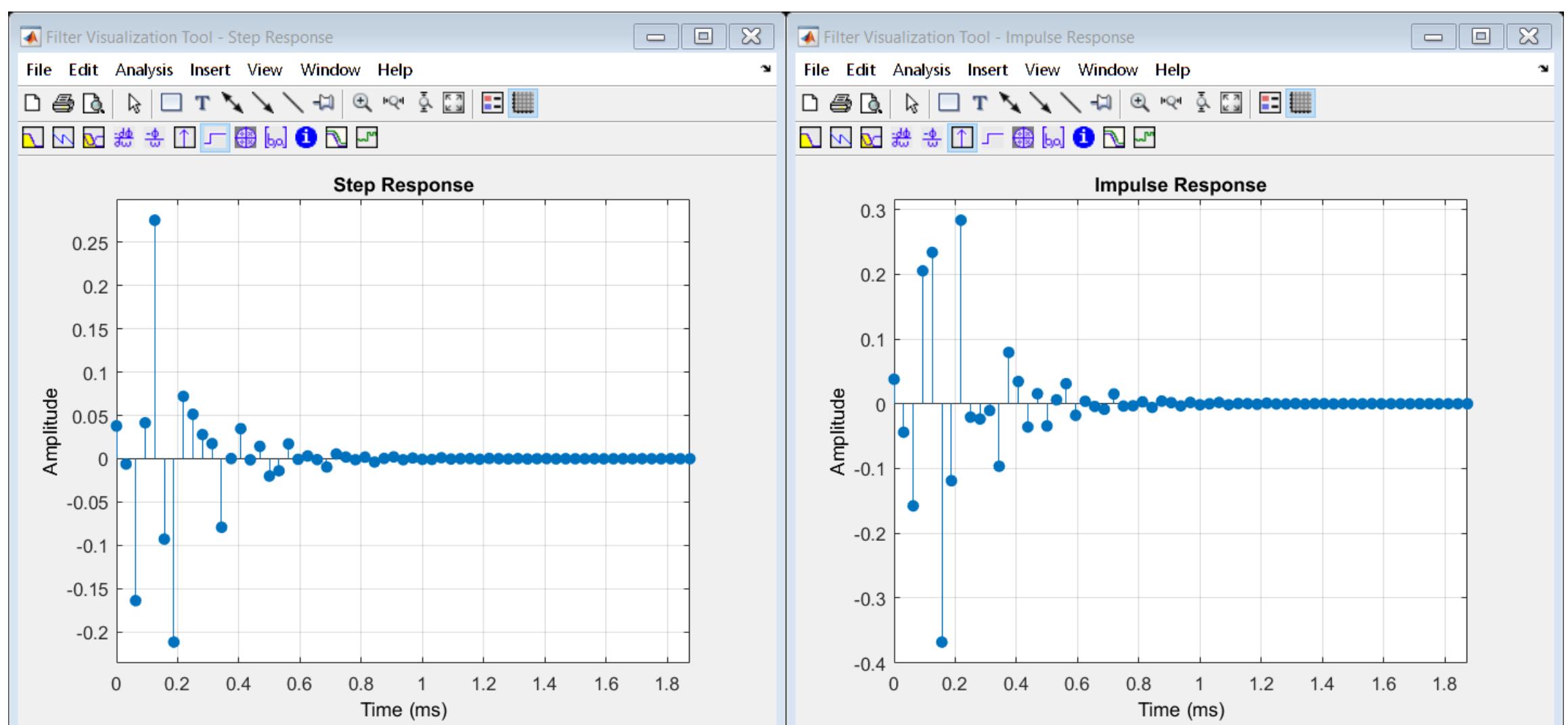
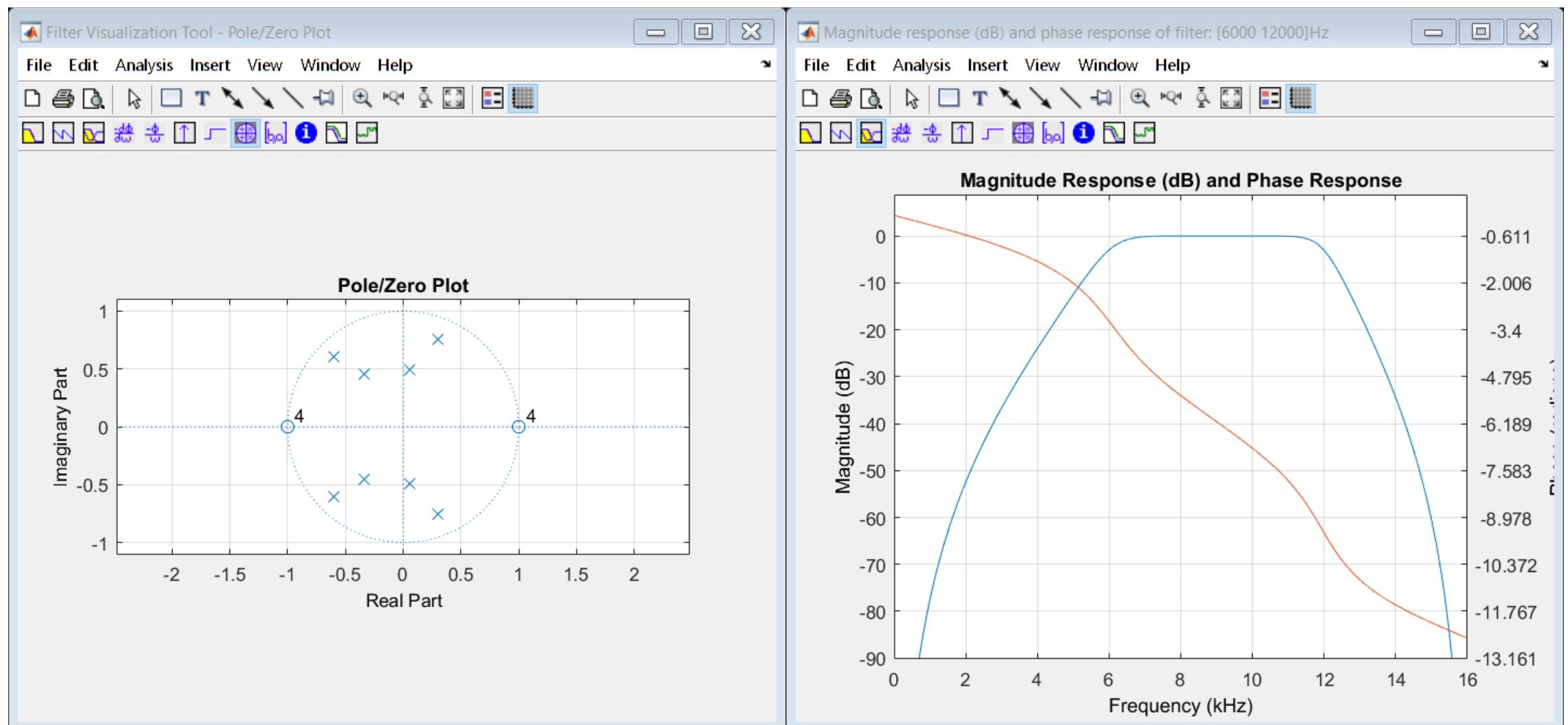
Sample run 2 (cont.):



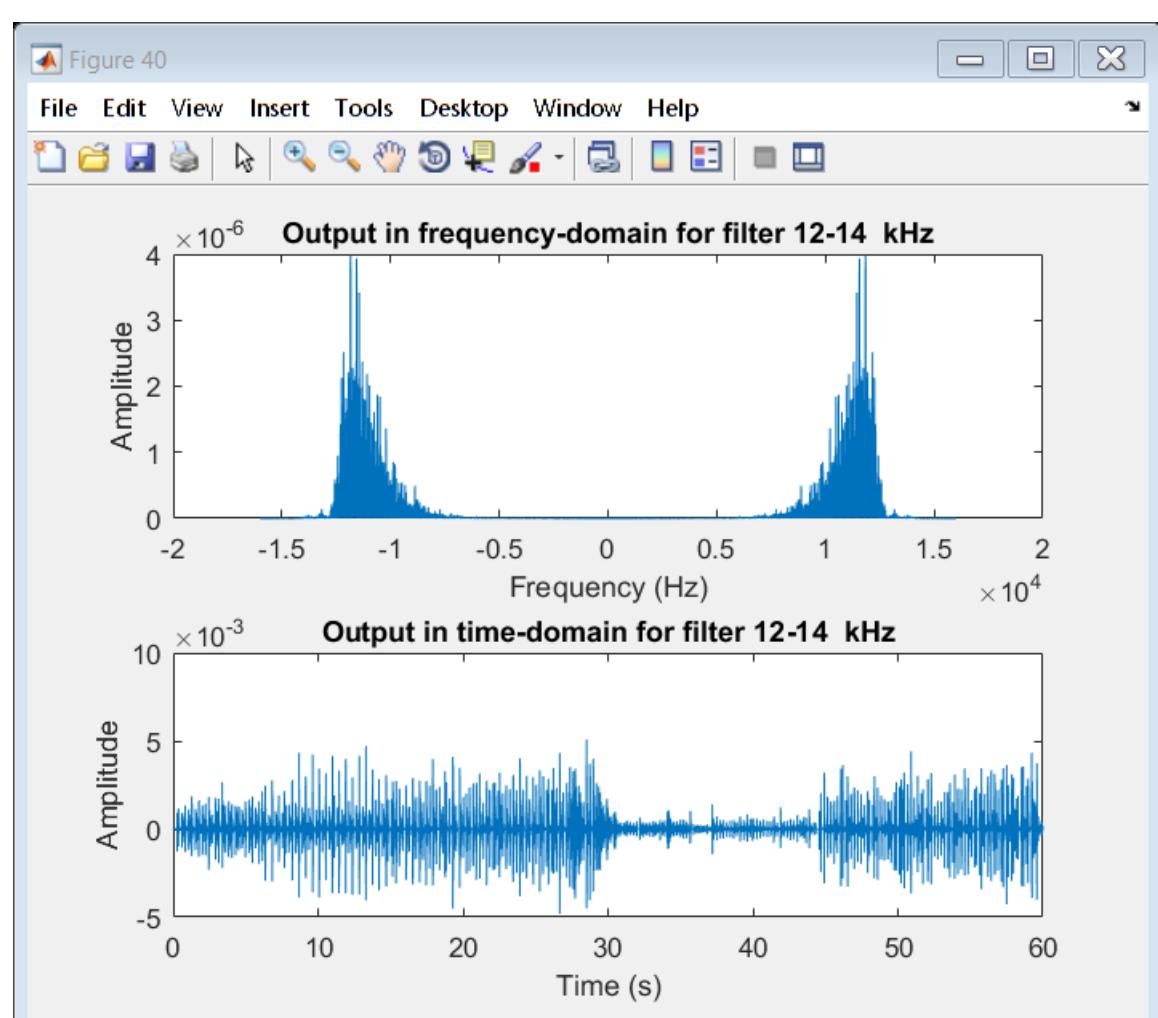
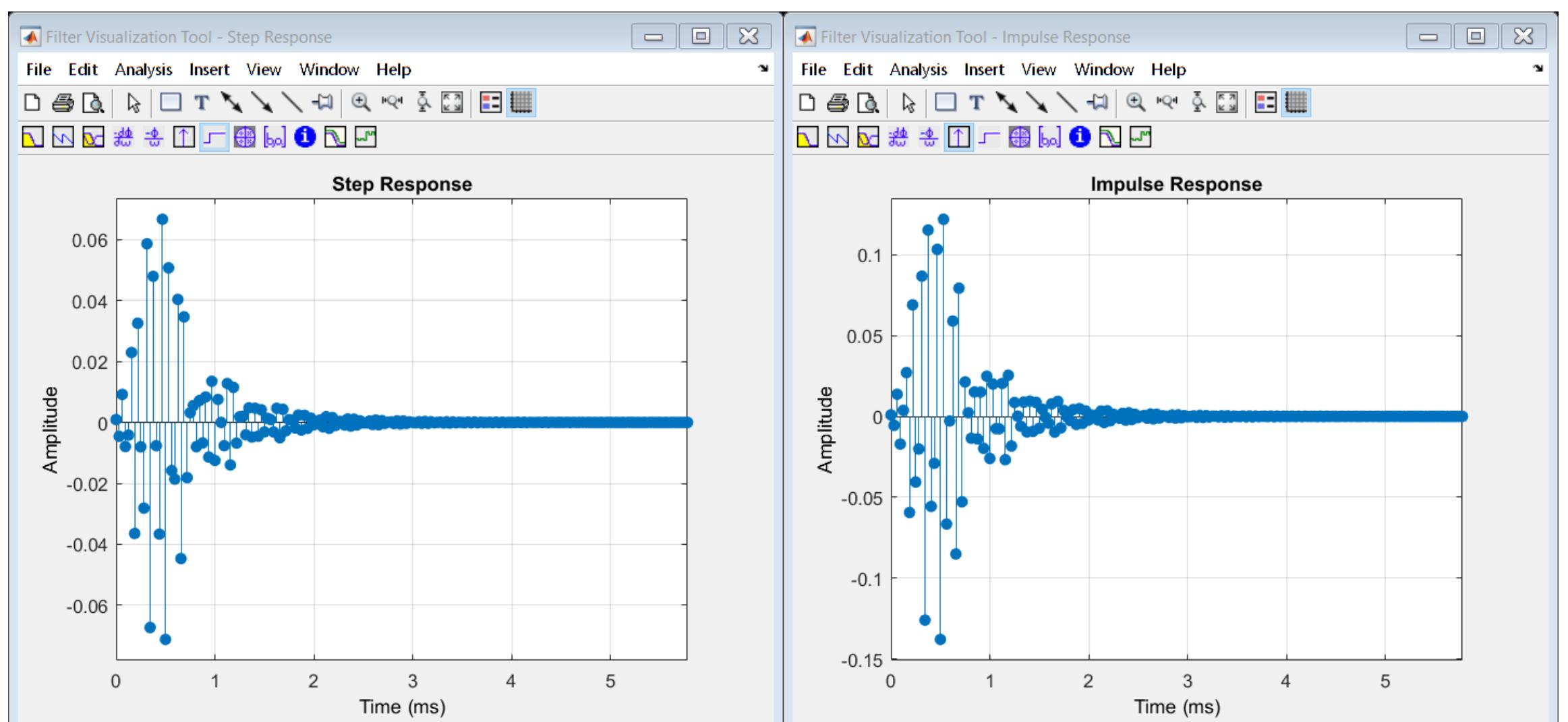
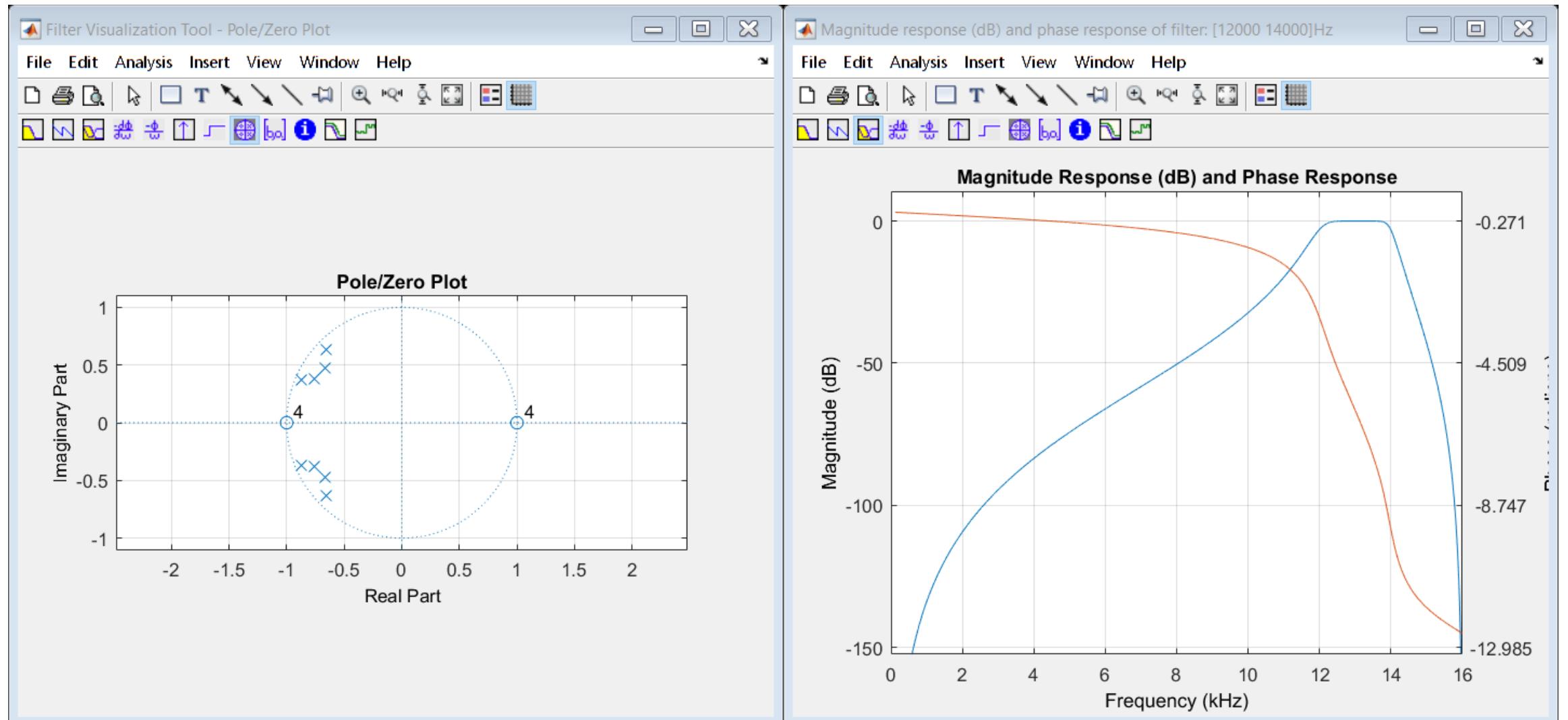
Sample run 2 (cont.):



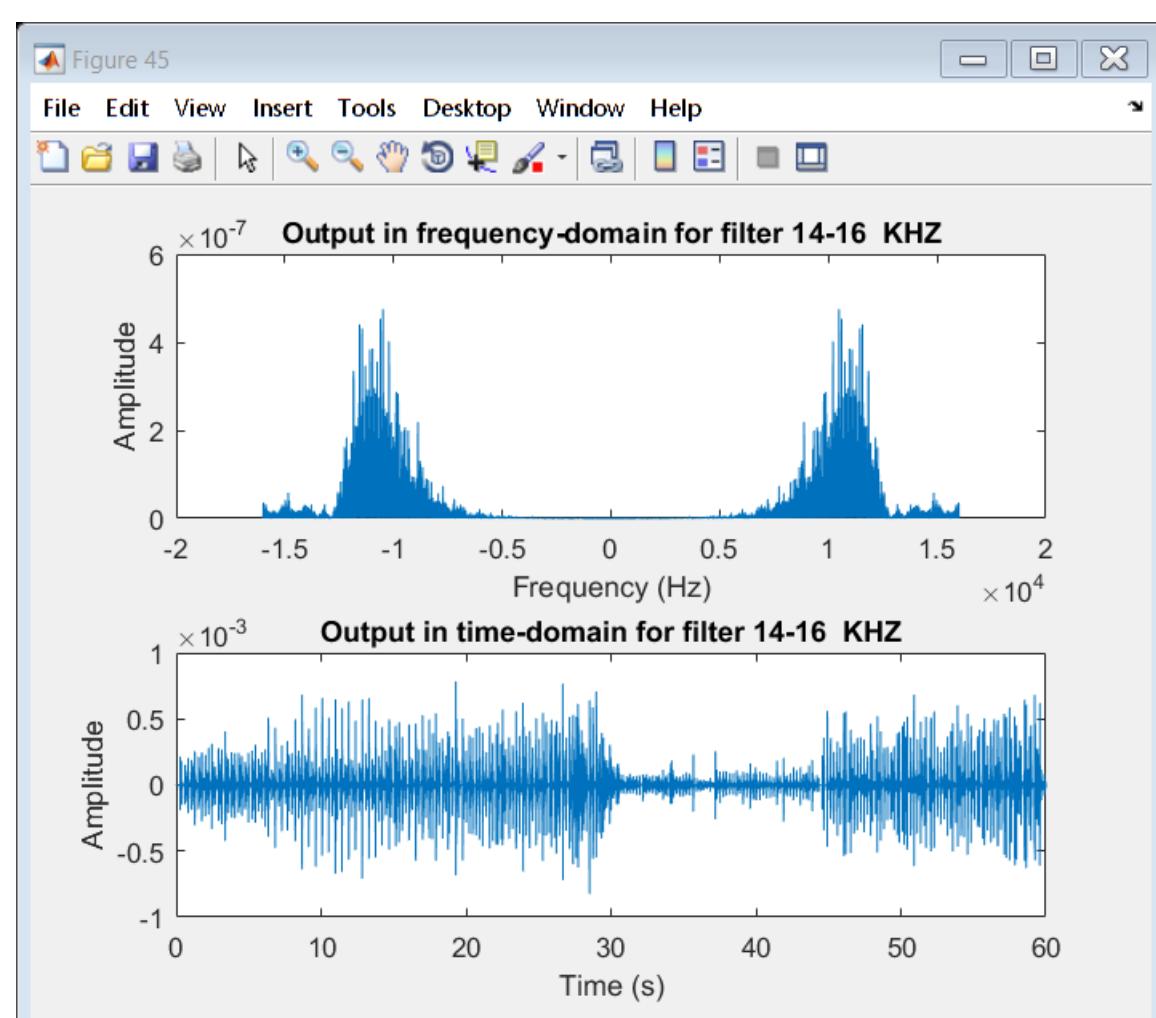
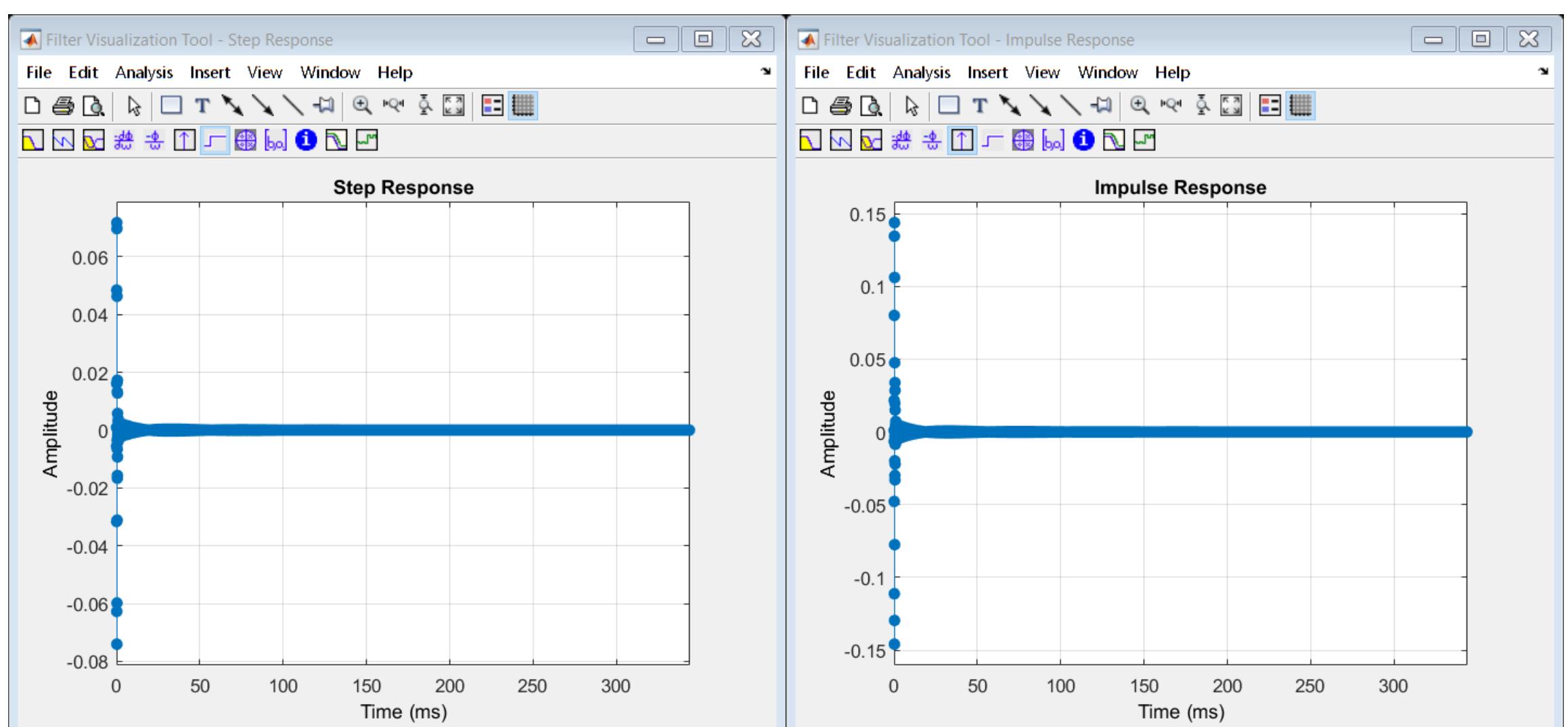
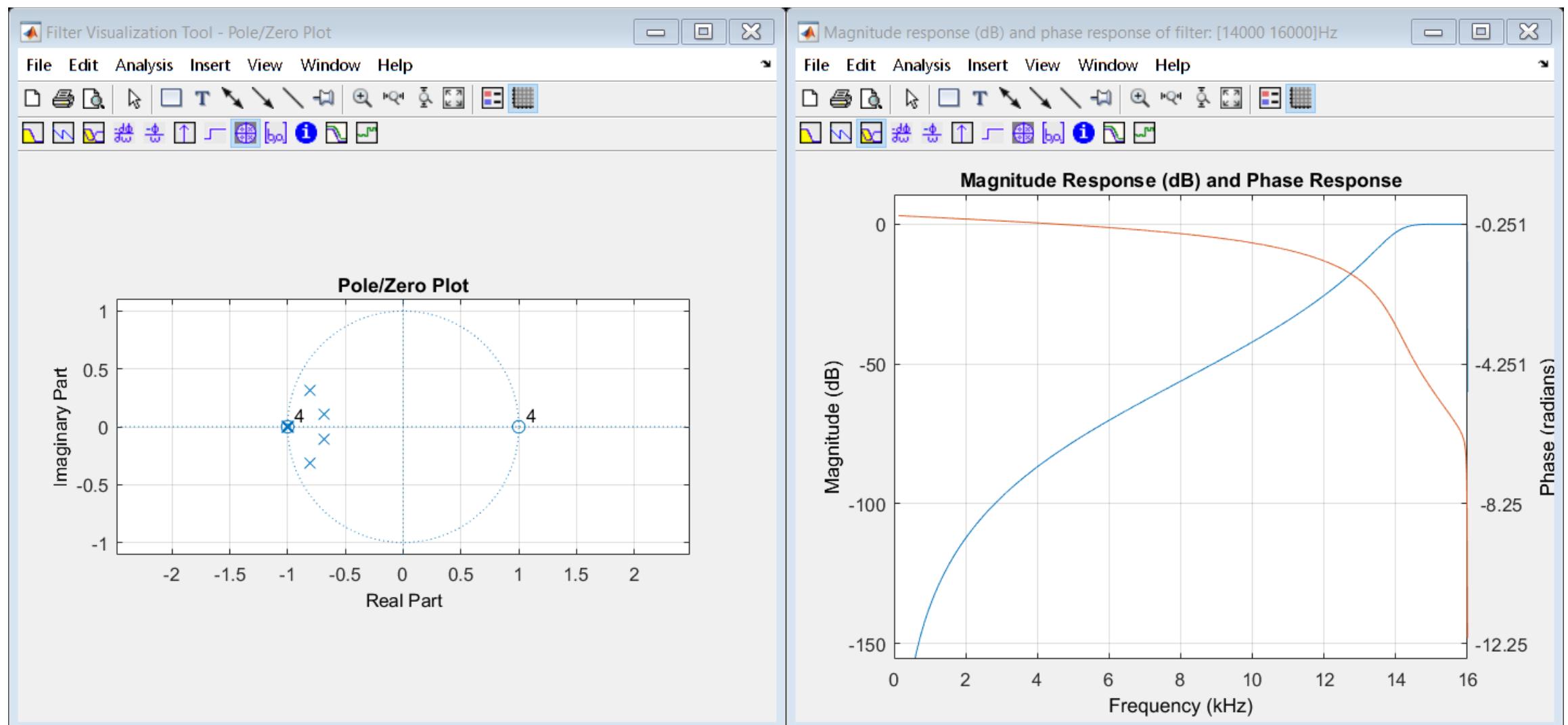
Sample run 2 (cont.):



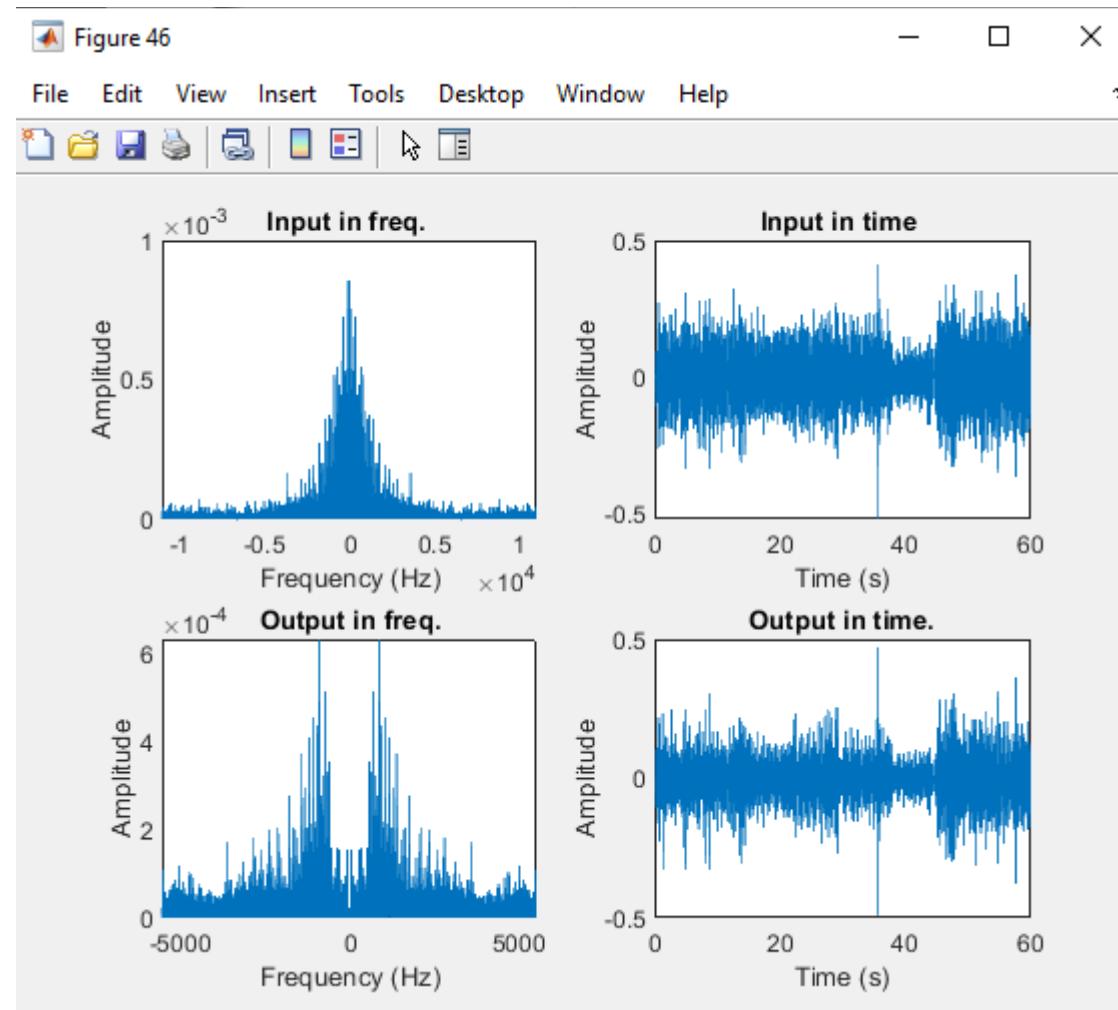
Sample run 2 (cont.):



Sample run 2 (cont.):



Sample run 2 (cont.):



Sample run 3 (Double sample rate):

Type of filter: fir

Specified gains for each of the nine filters: -20, -15, -10, -5, 0, 5, 10, 15, 20 dB

Specified output frequency: 44100 Hz

Output:

File information:

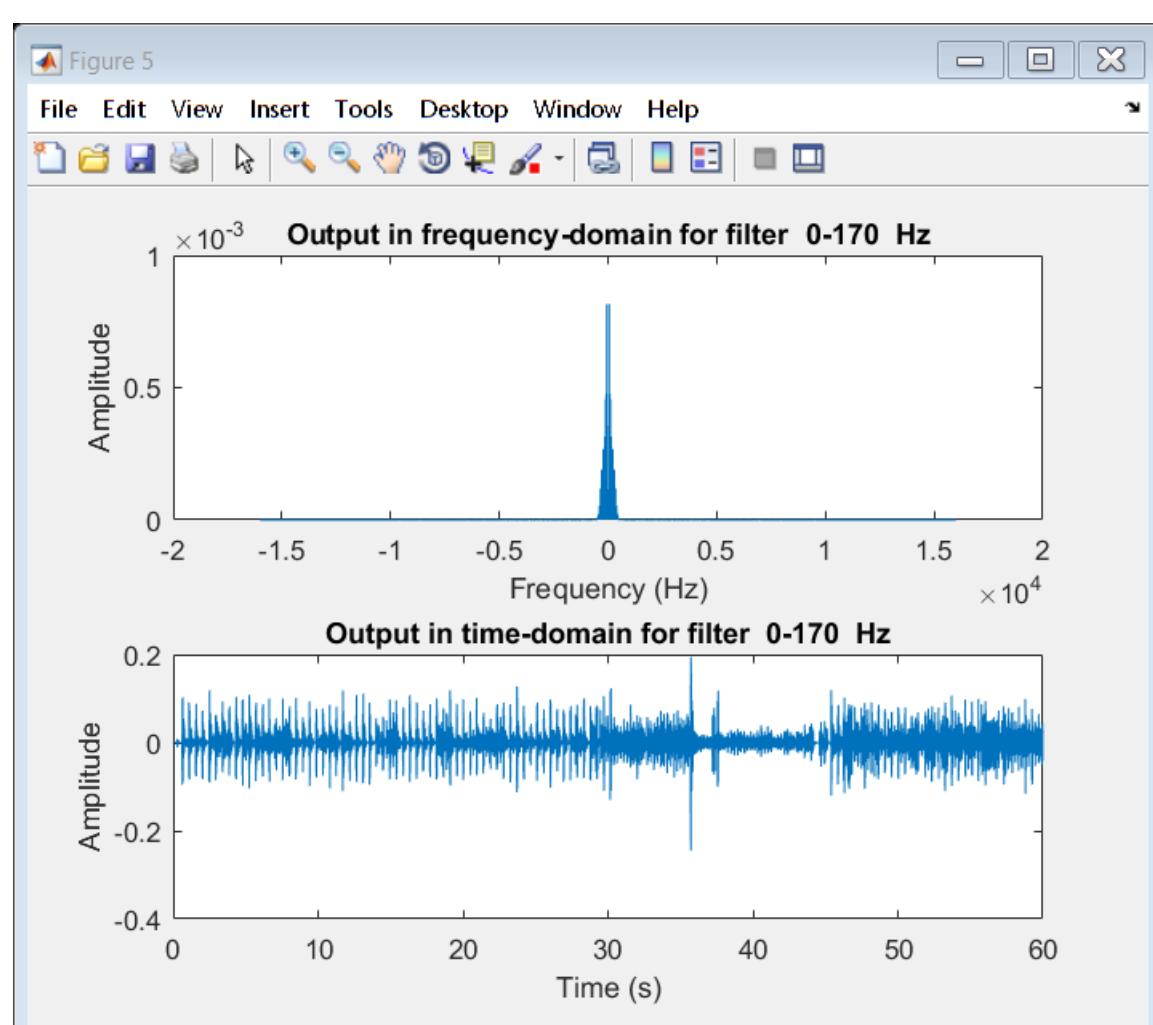
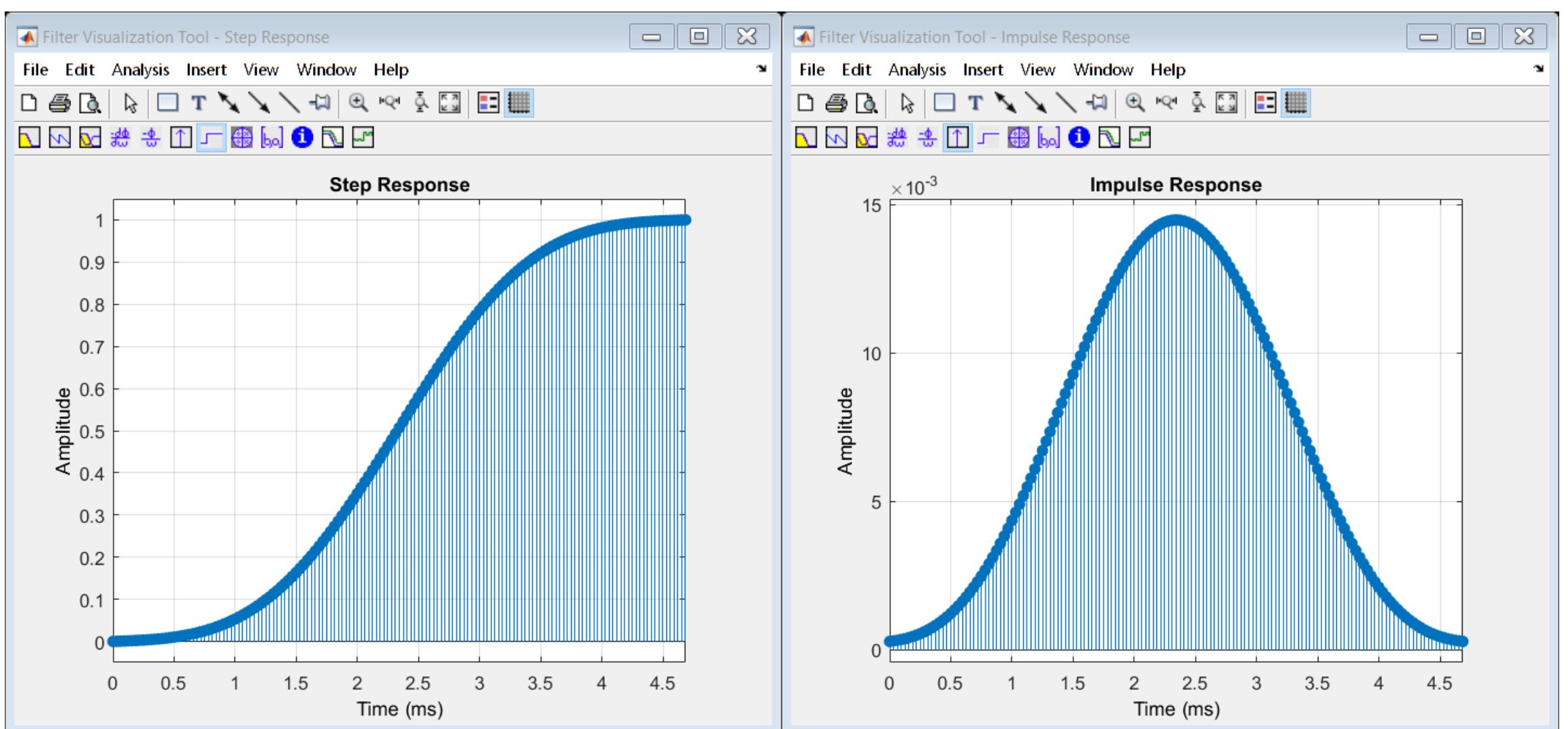
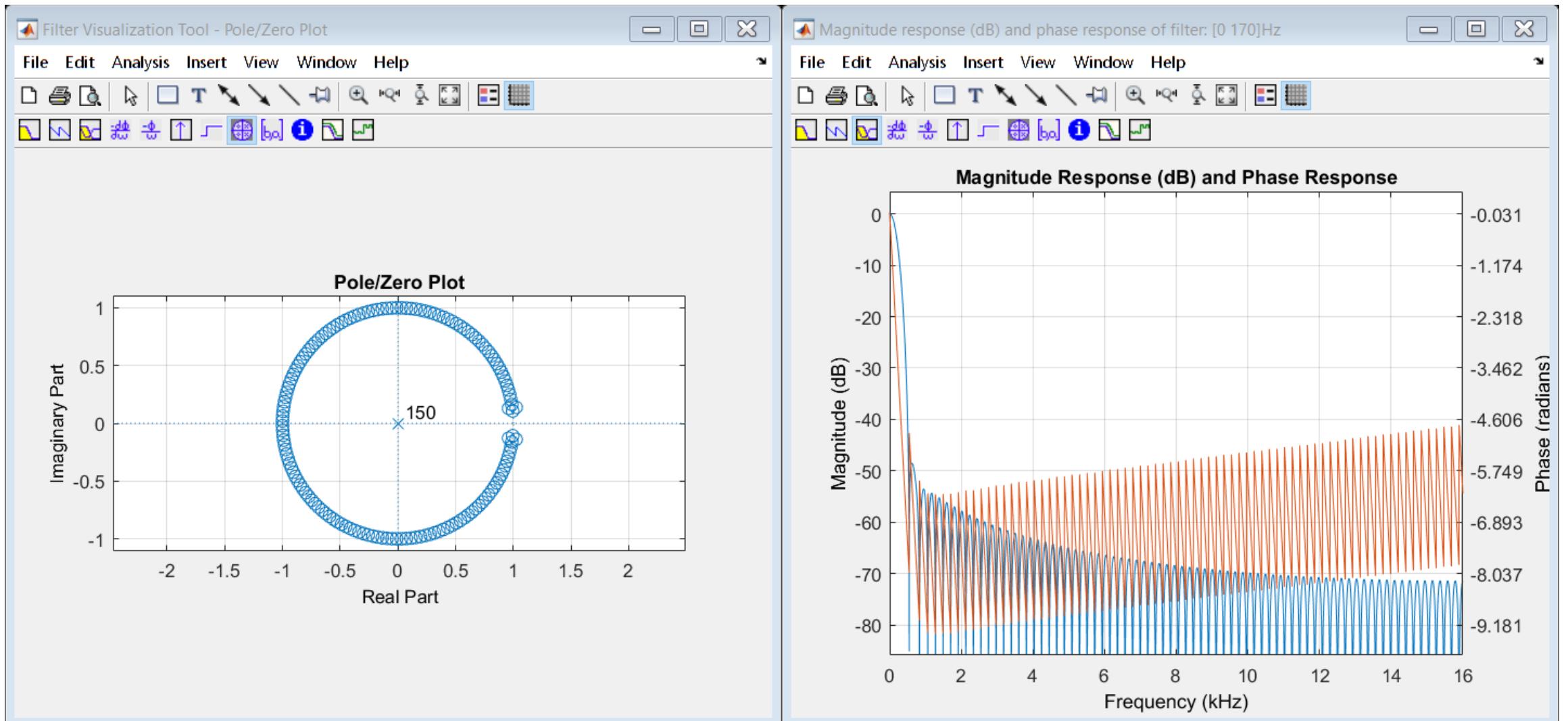
Path: D:\SSP\term 6\DSP\CantinaBand60.wav

Data dimensions: [1323000 1]

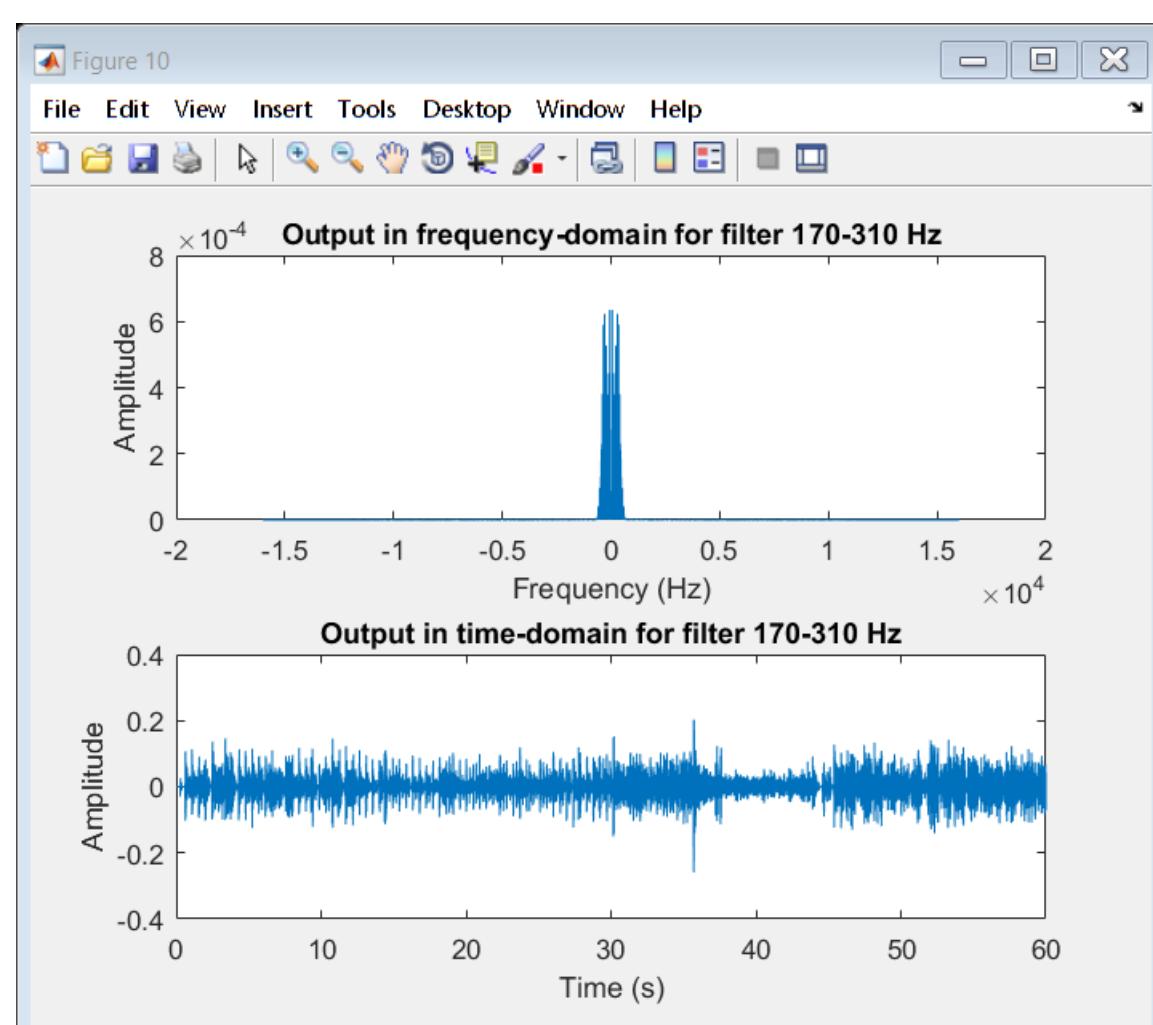
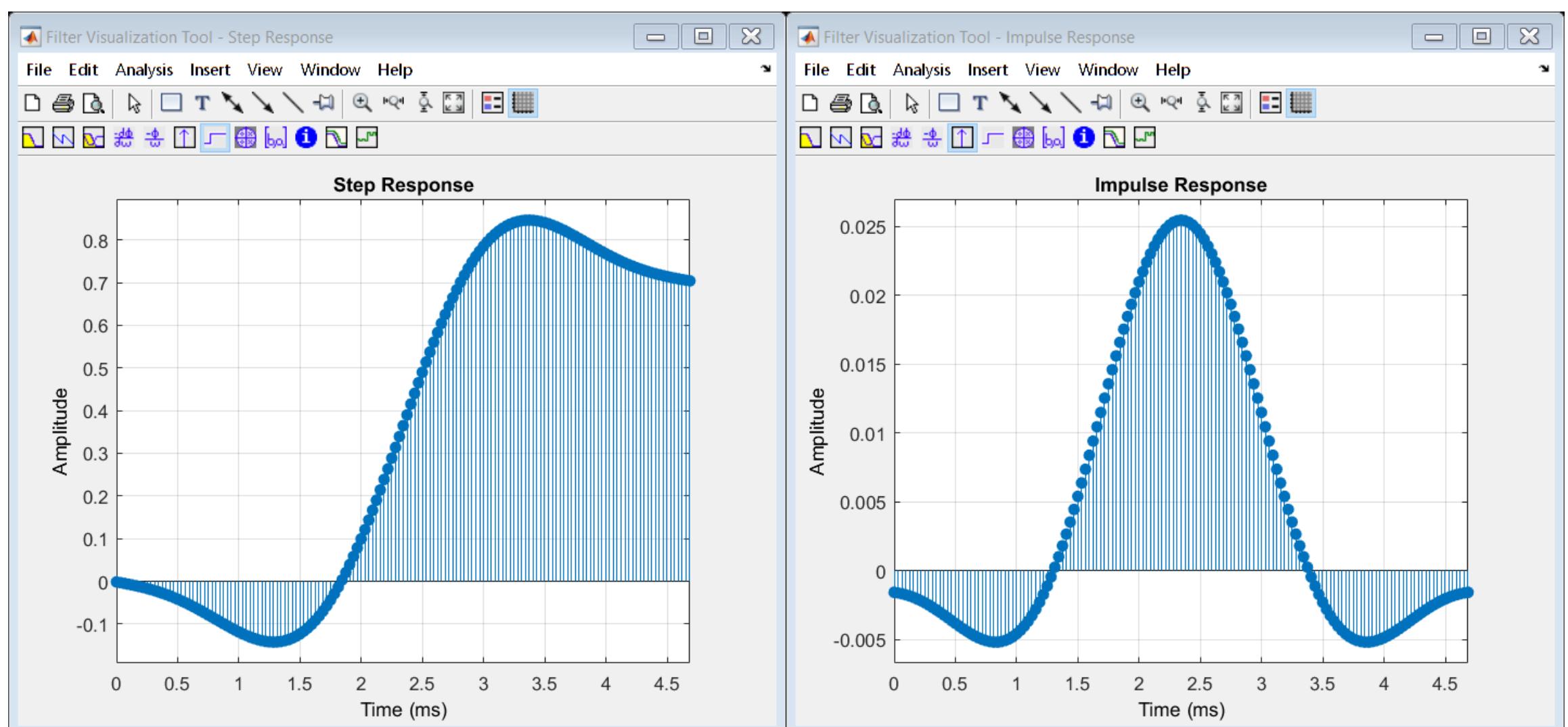
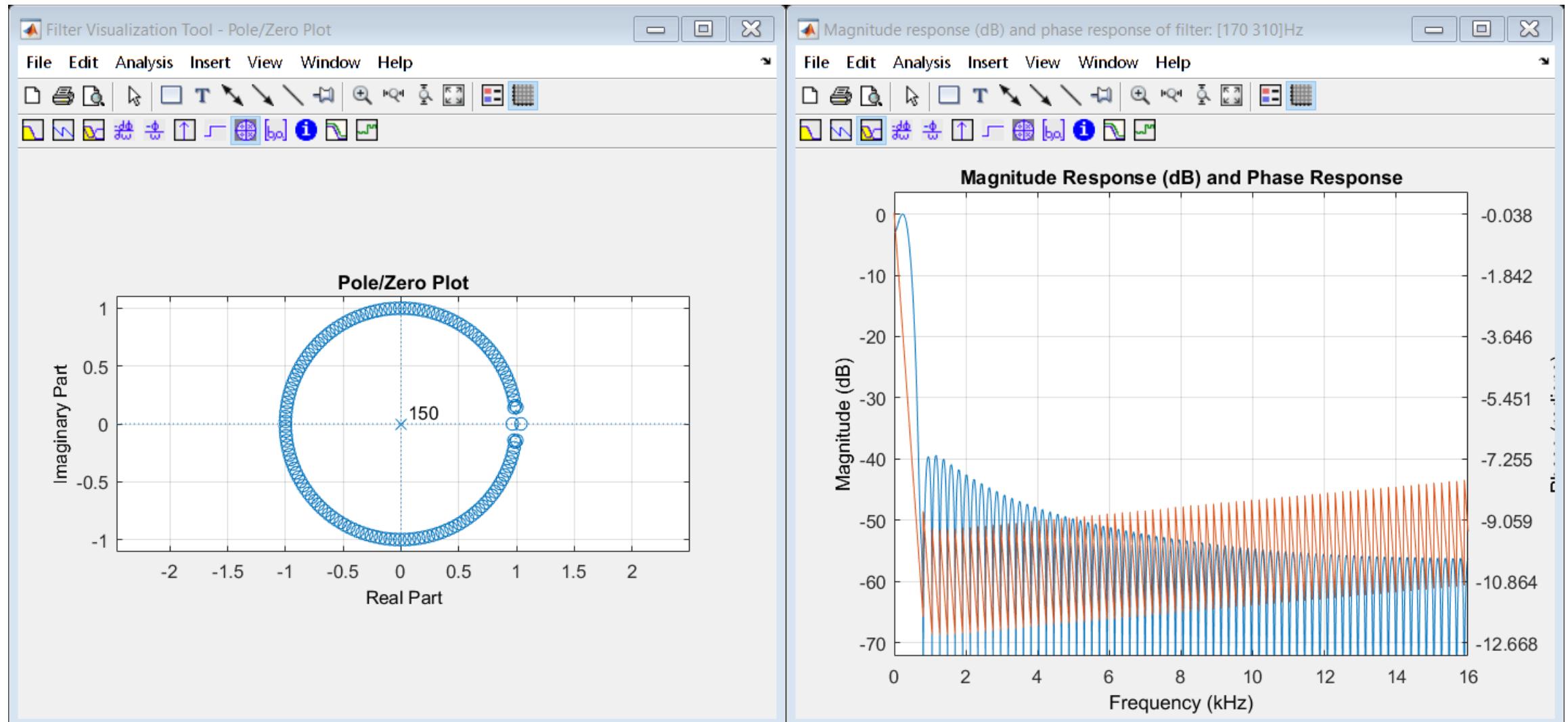
Frequency: 22050

The gain of fir filter : 0-170 Hz is 0.000277 , Order is 150
The gain of fir filter : 170-310 Hz is -0.001567 , Order is 150
The gain of fir filter : 310-600 Hz is 0.000804 , Order is 150
The gain of fir filter : 0.6-1 kHz is 0.000115 , Order is 150
The gain of fir filter : 1-3 kHz is -0.000228 , Order is 150
The gain of fir filter : 3-6 kHz is 0.000054 , Order is 150
The gain of fir filter : 6-12 kHz is 0.000099 , Order is 150
The gain of fir filter : 12-14 kHz is -0.000537 , Order is 150
The gain of fir filter : 14-16 kHz is 0.000384 , Order is 150

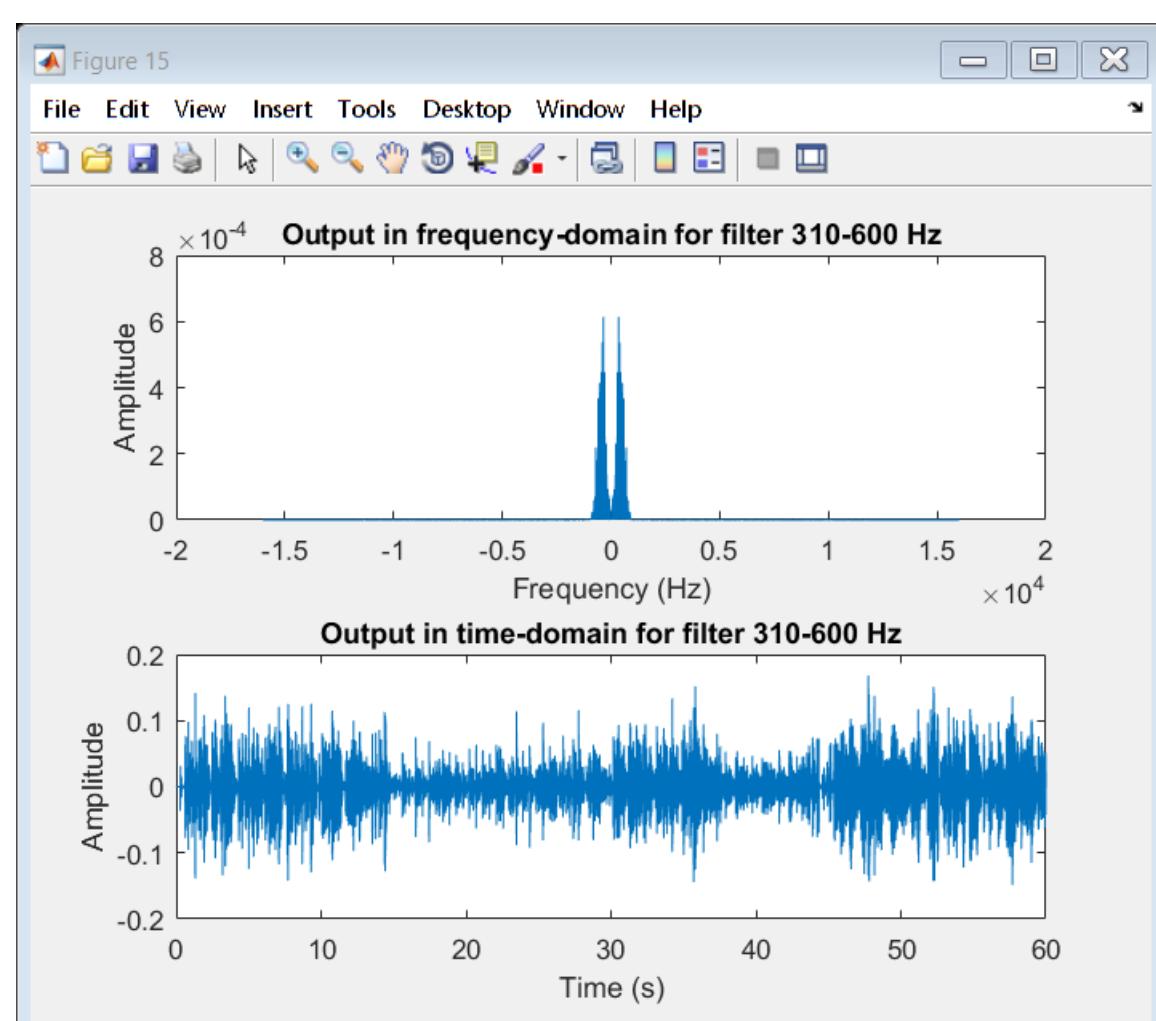
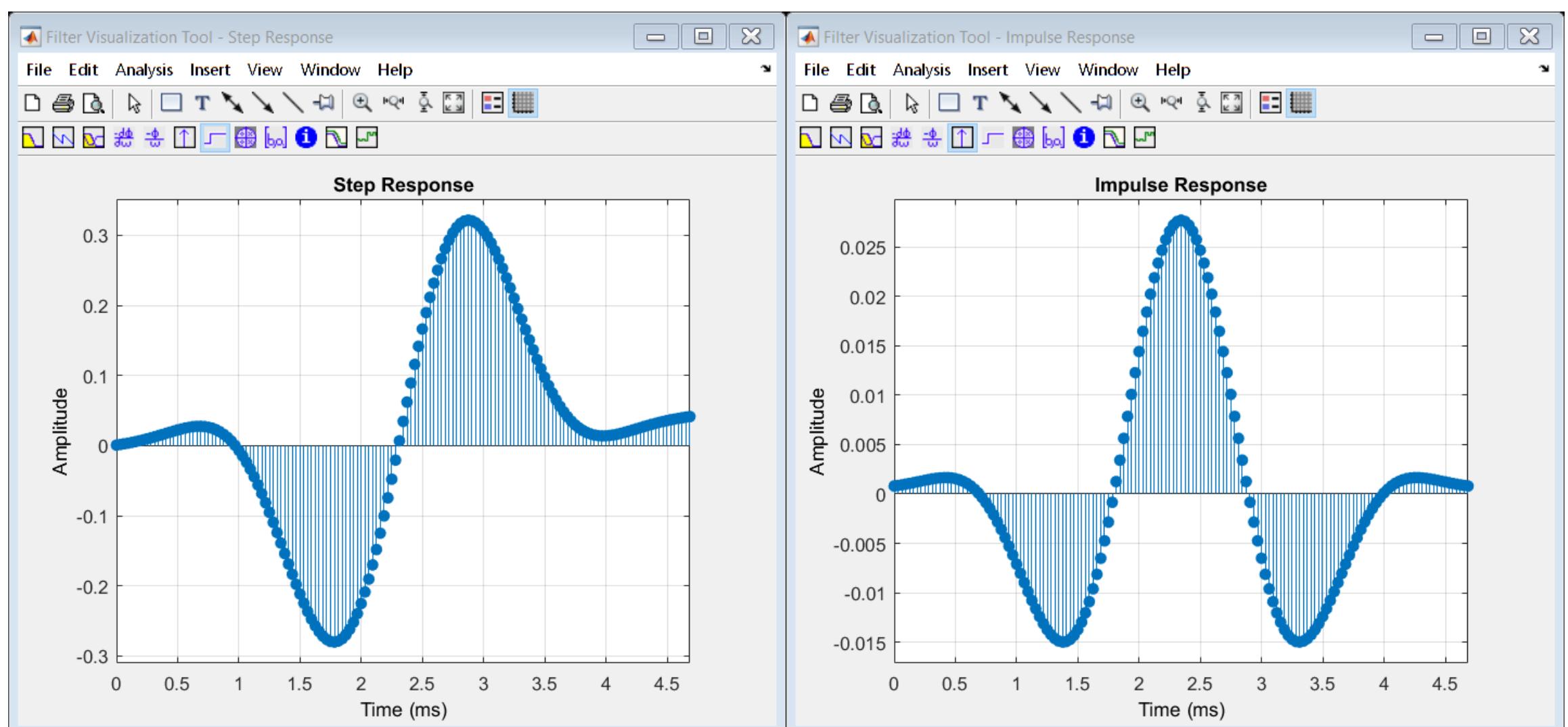
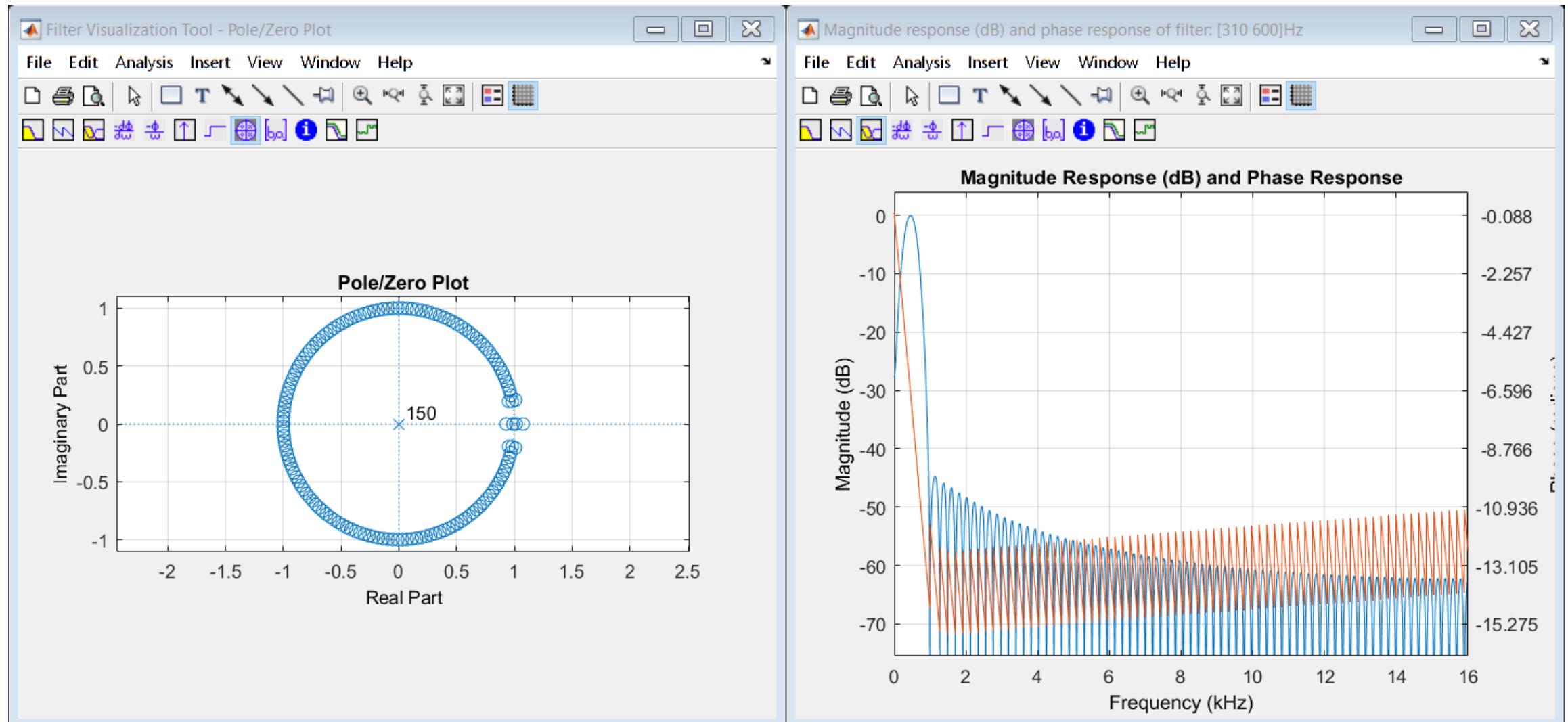
Sample run 3 (cont.):



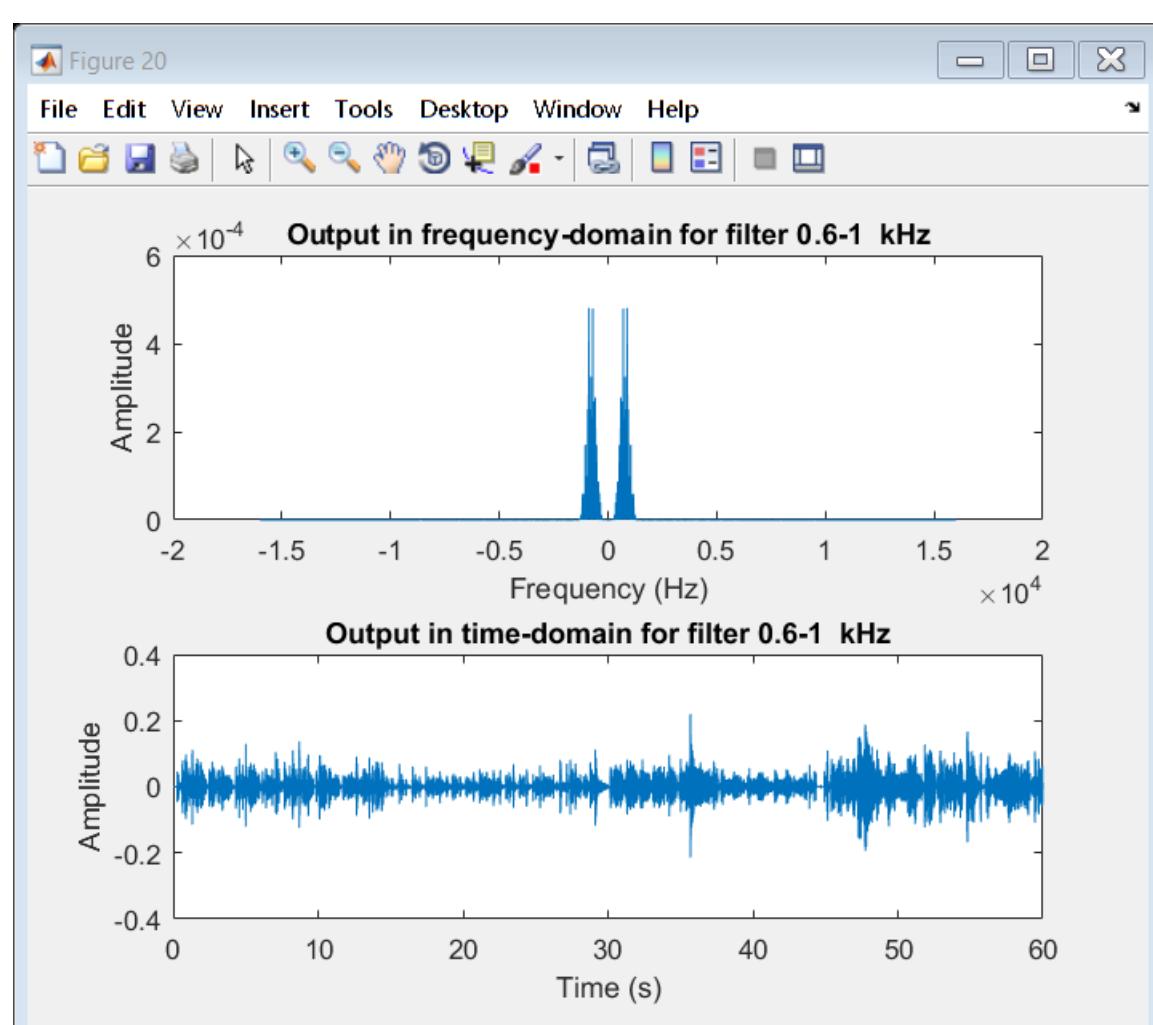
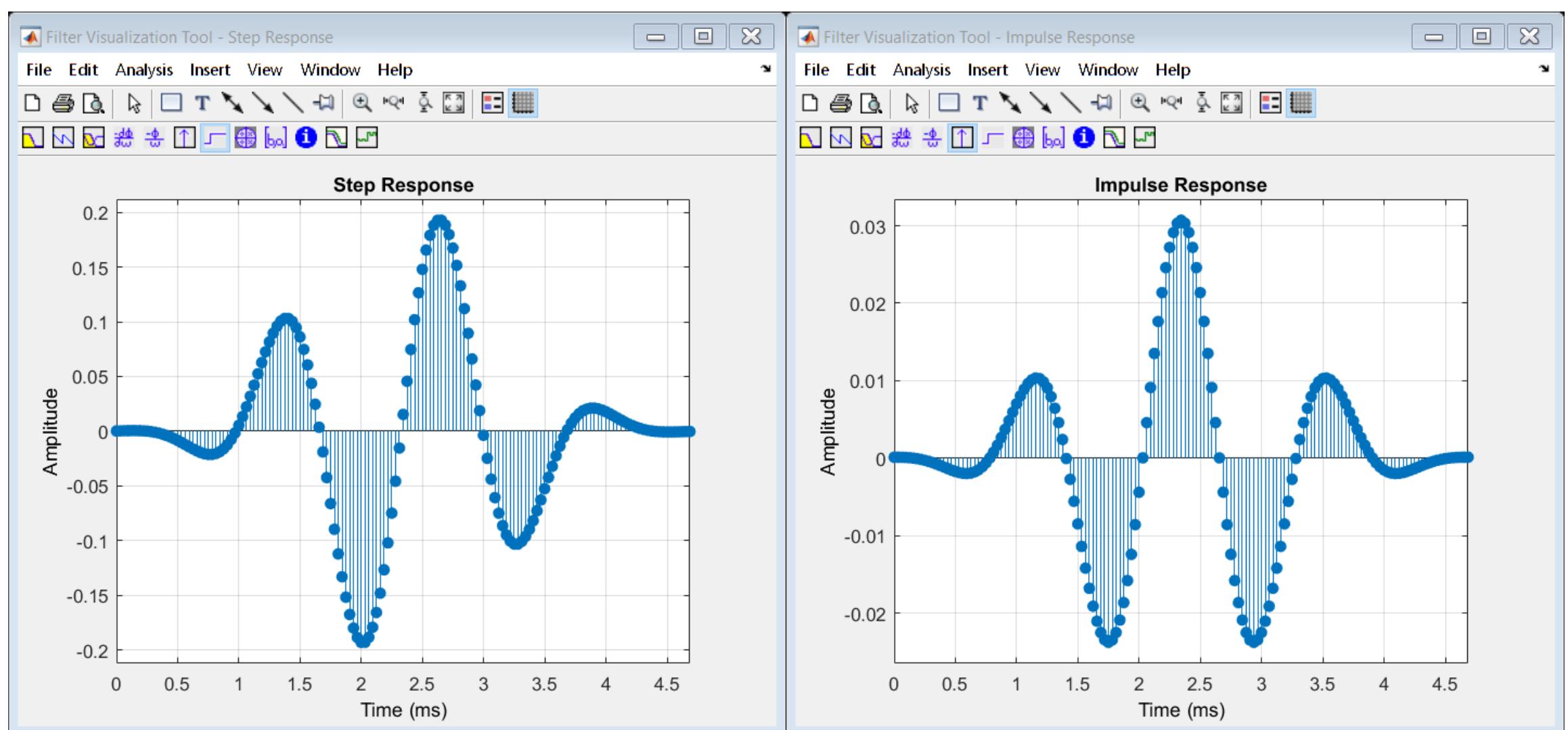
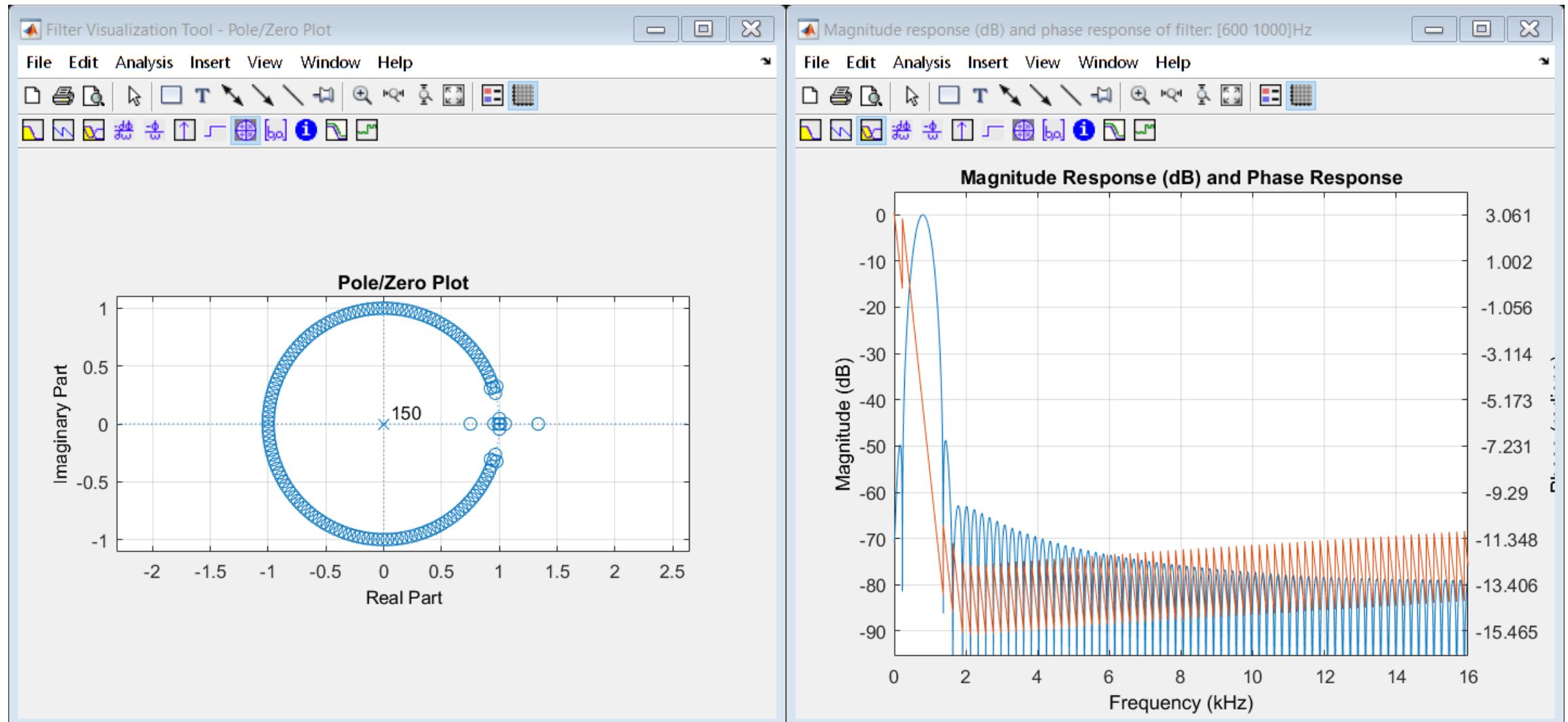
Sample run 3 (cont.):



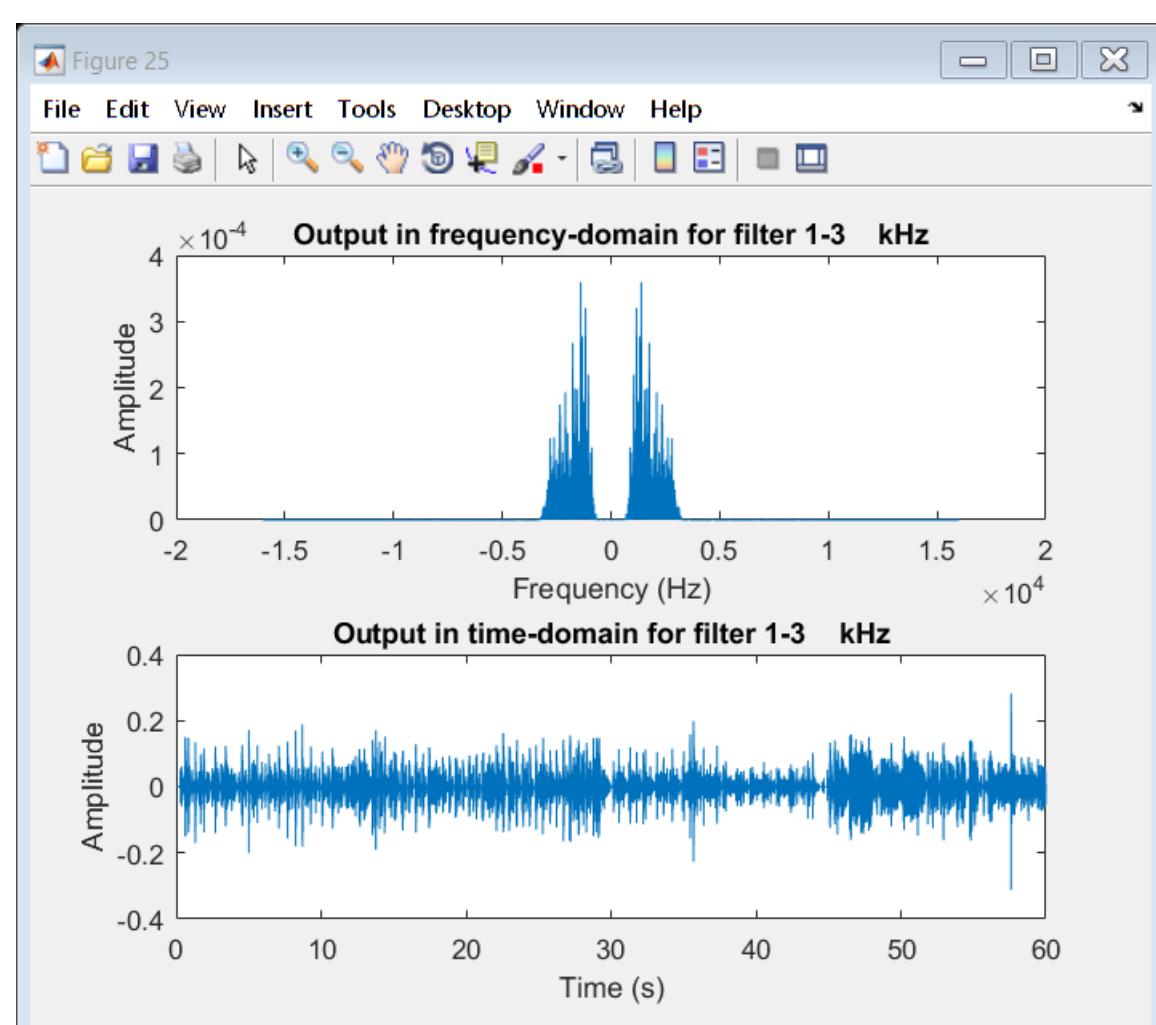
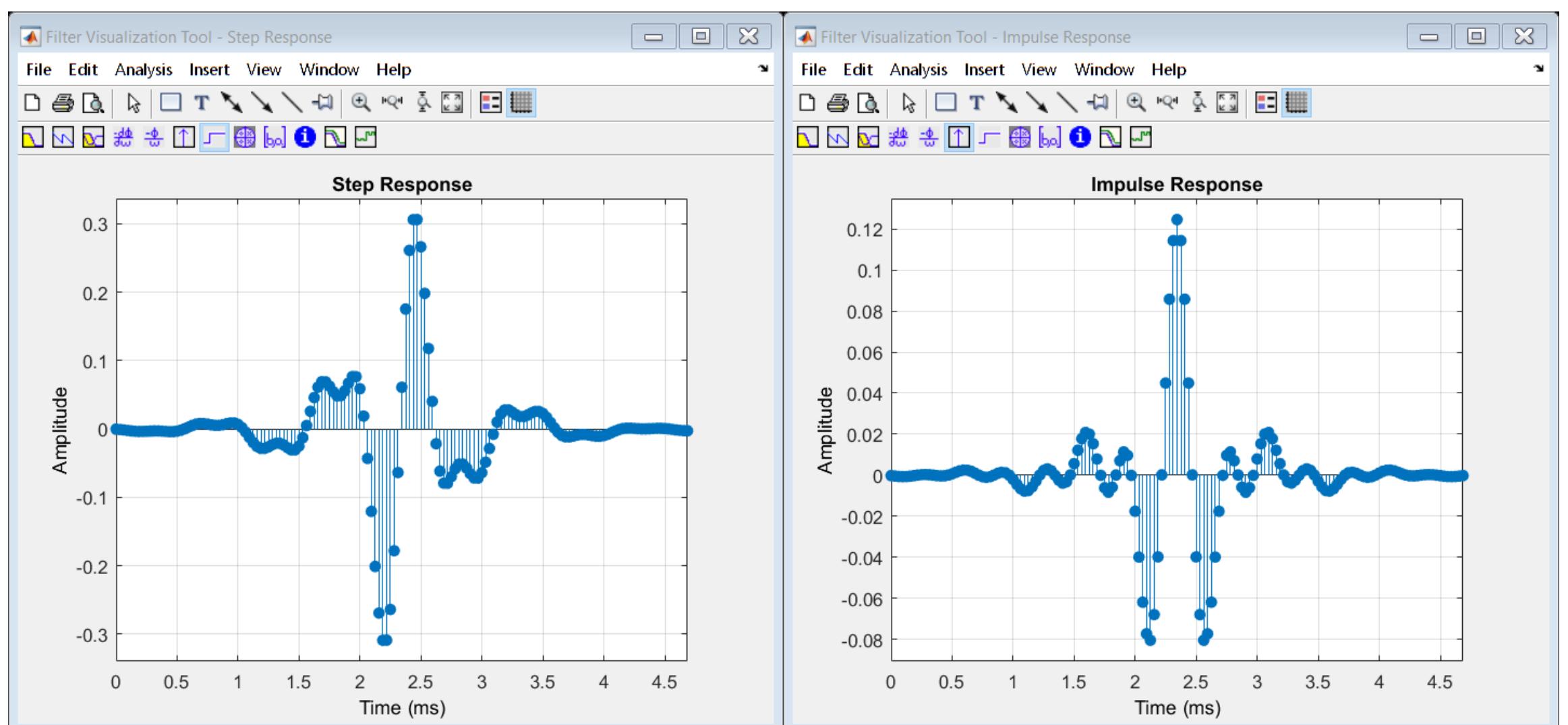
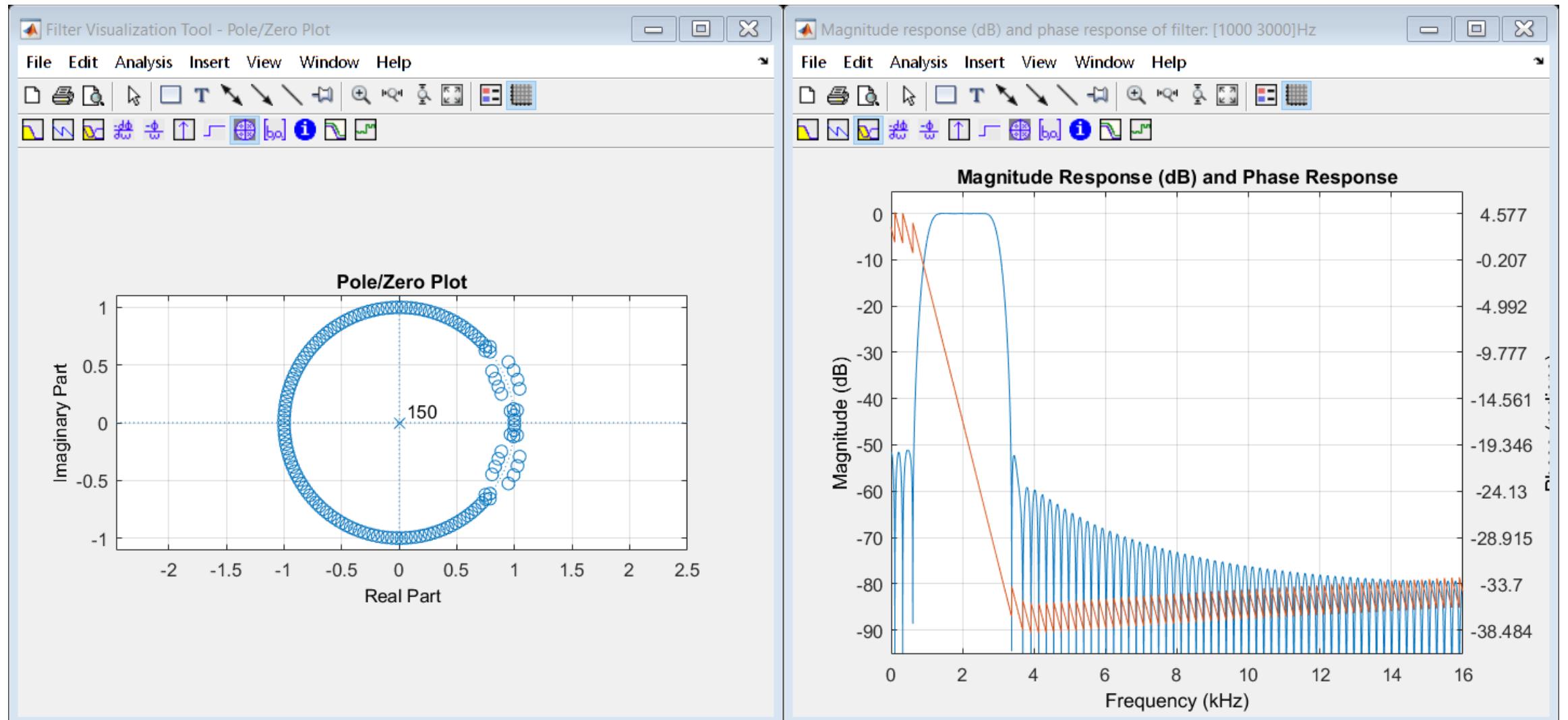
Sample run 3 (cont.):



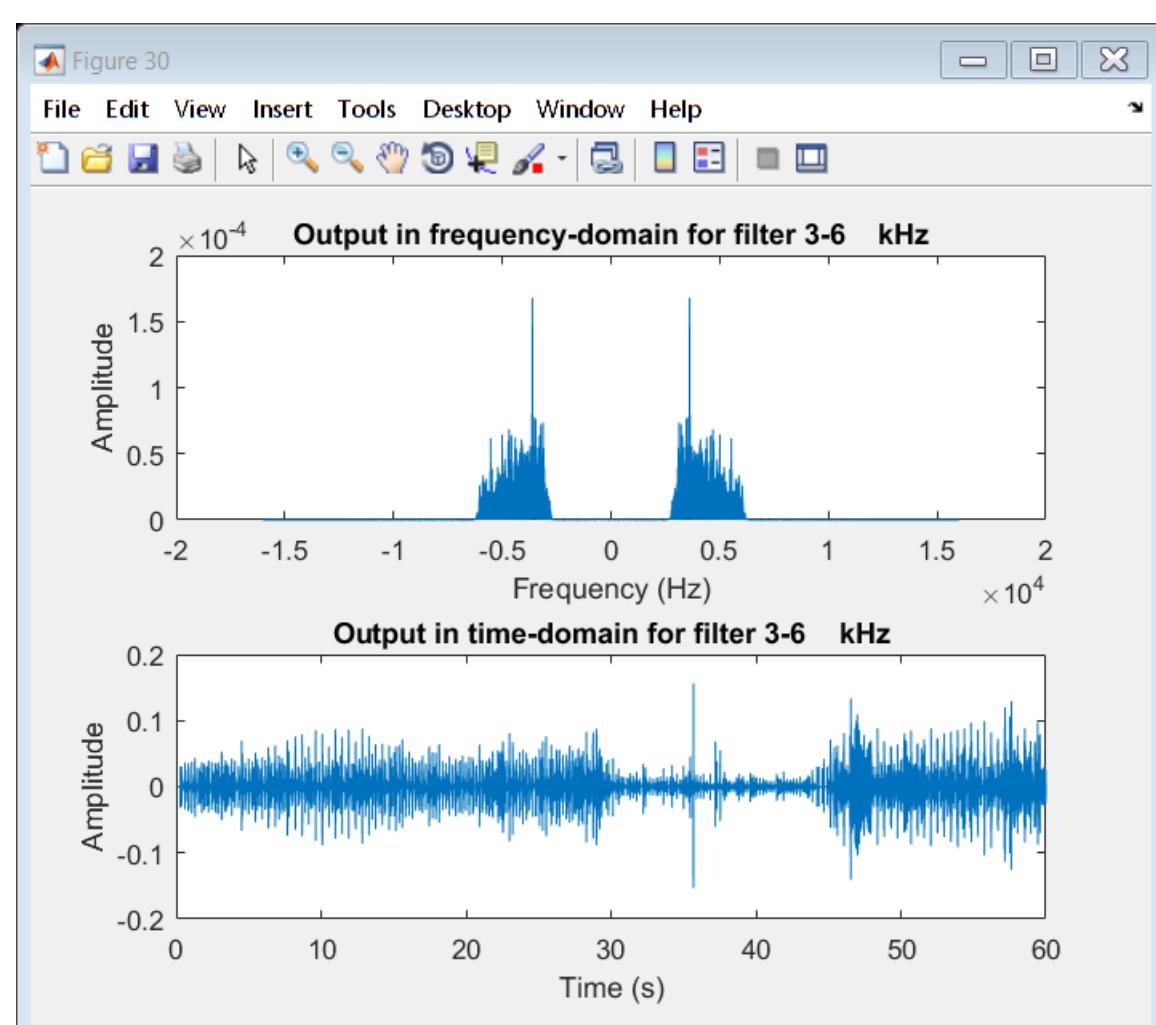
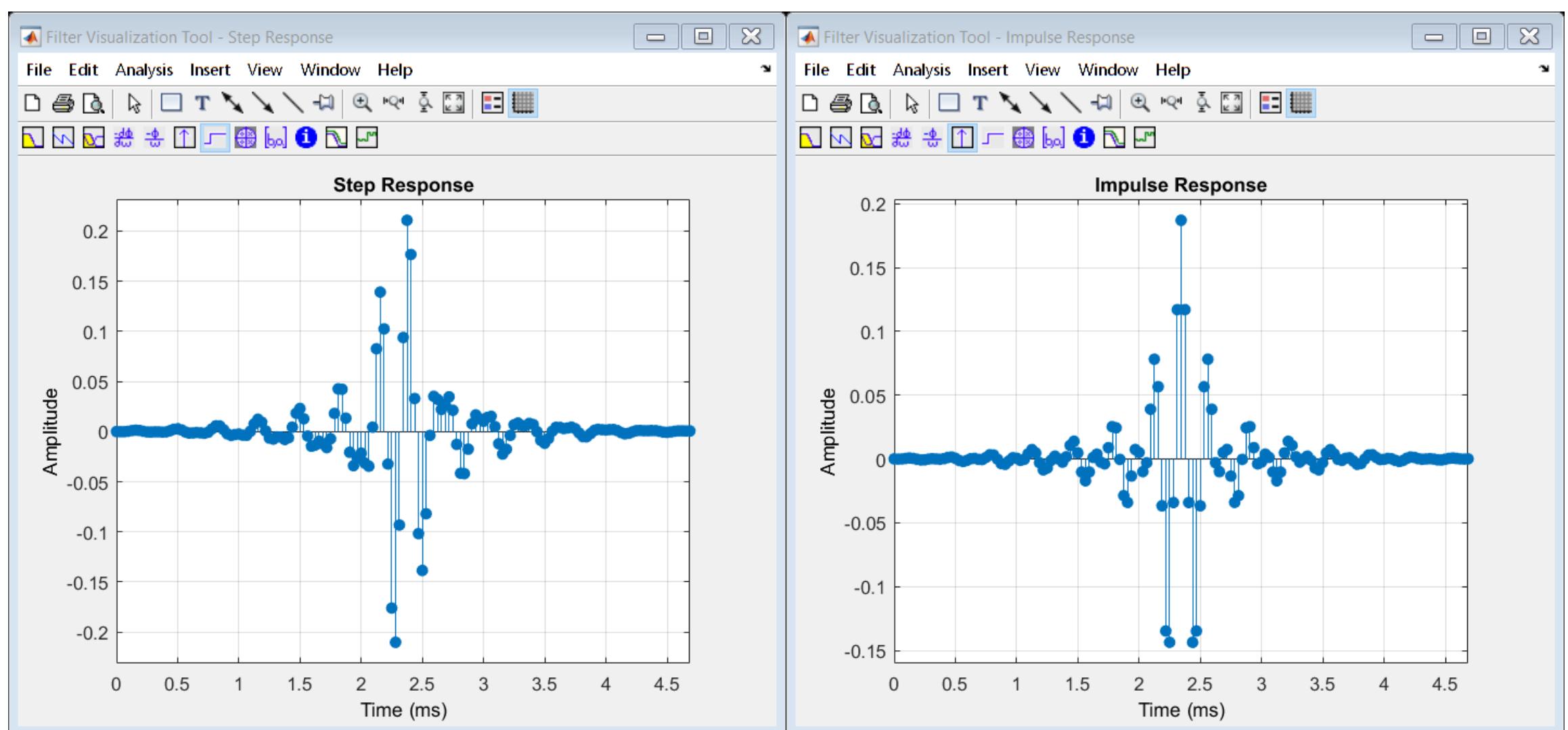
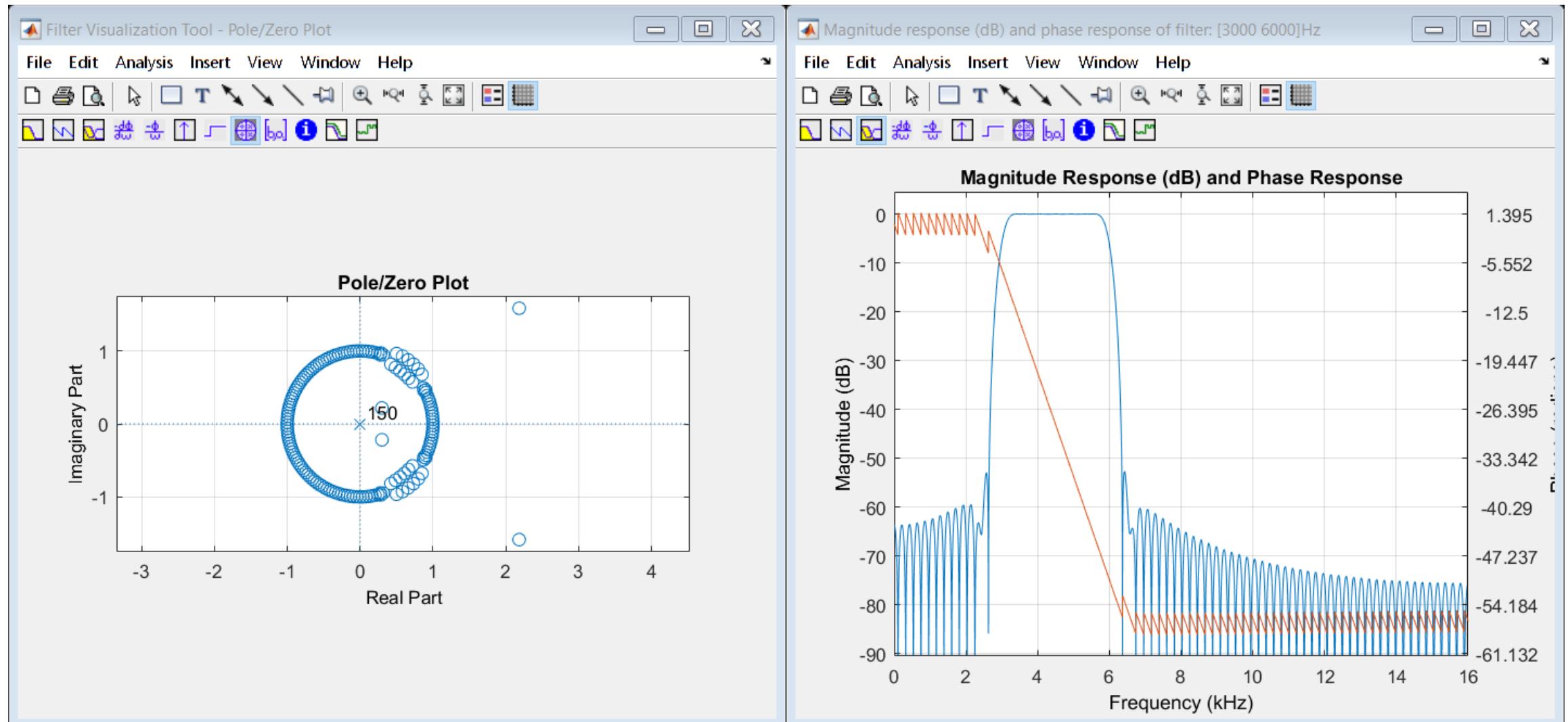
Sample run 3 (cont.):



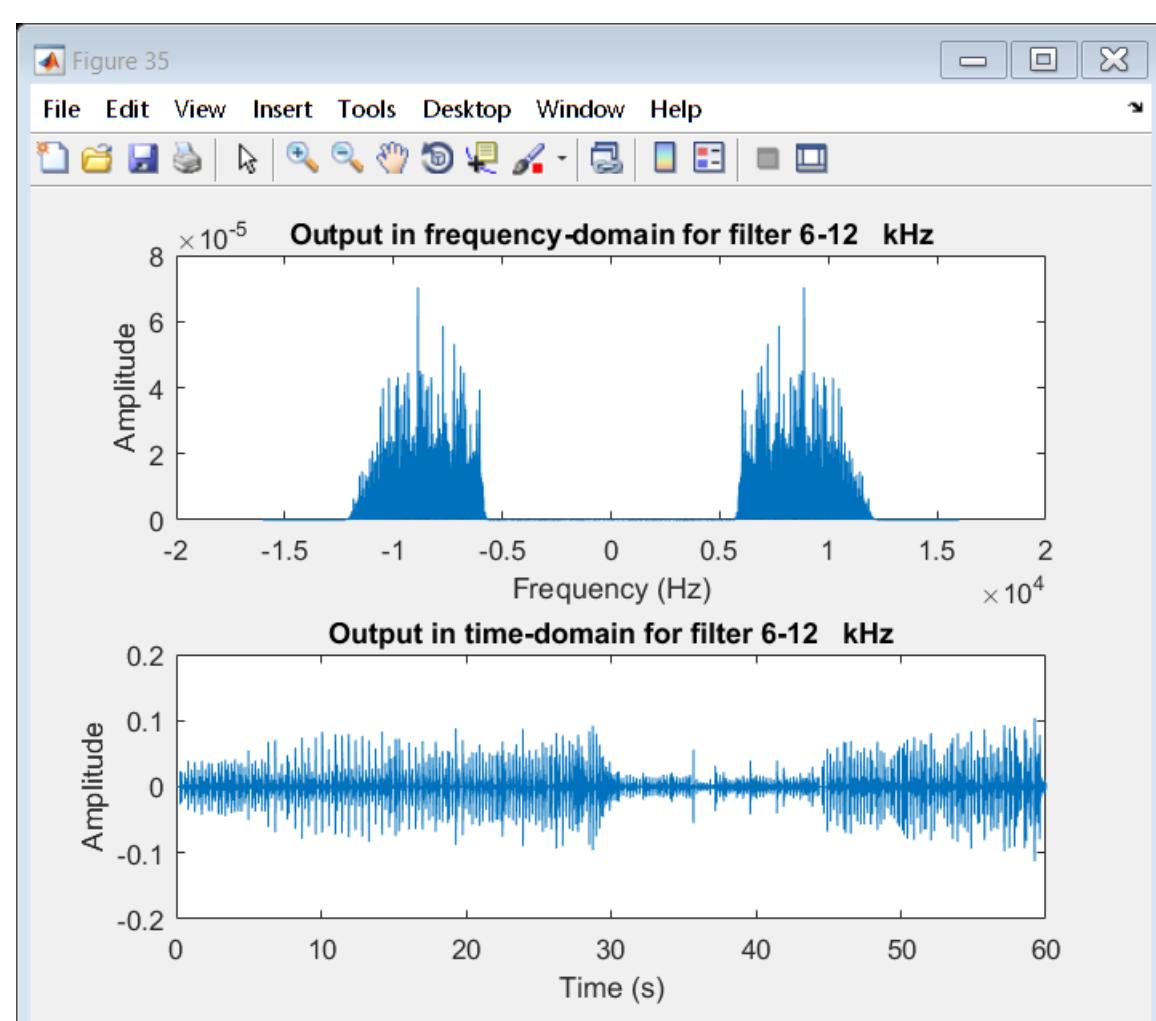
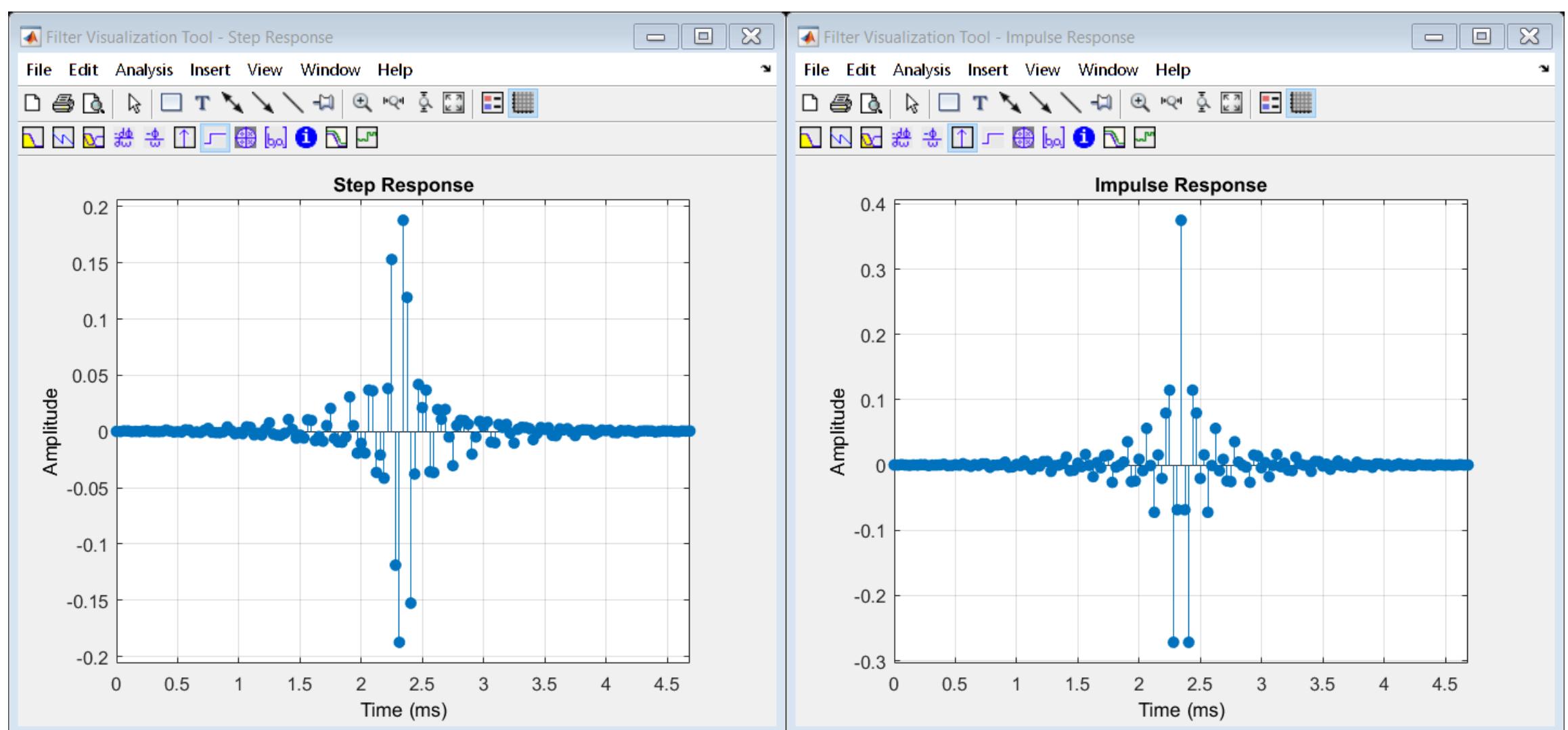
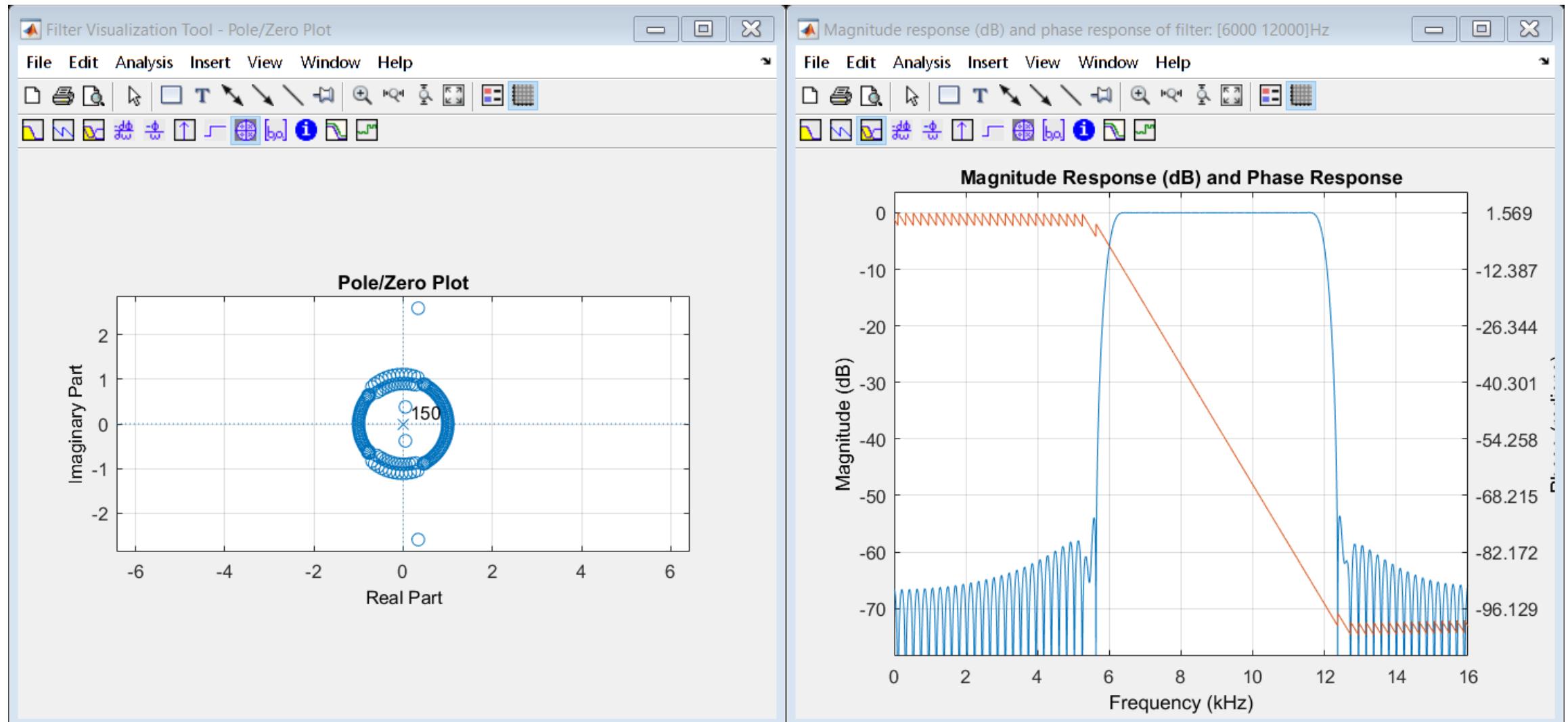
Sample run 3 (cont.):



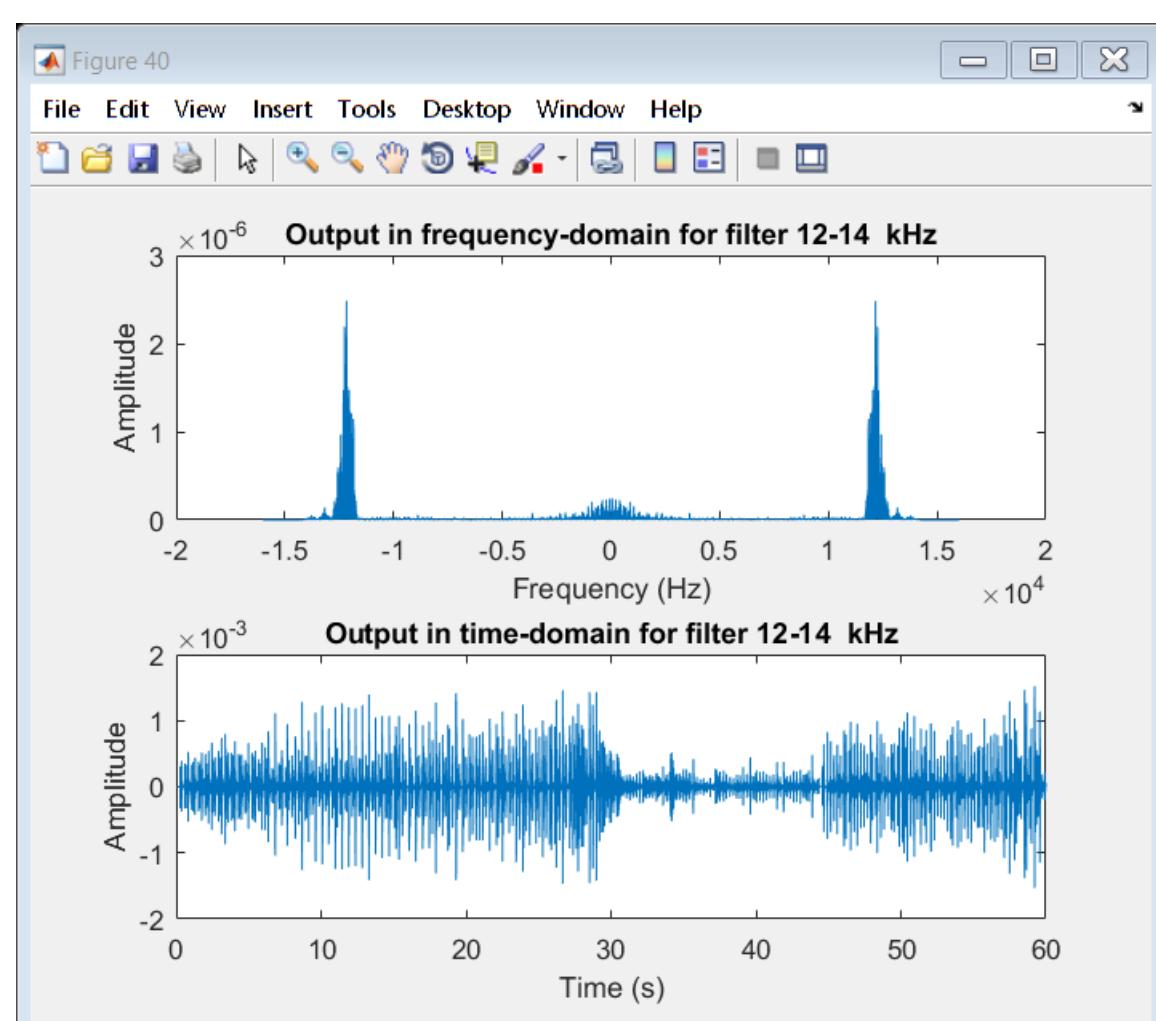
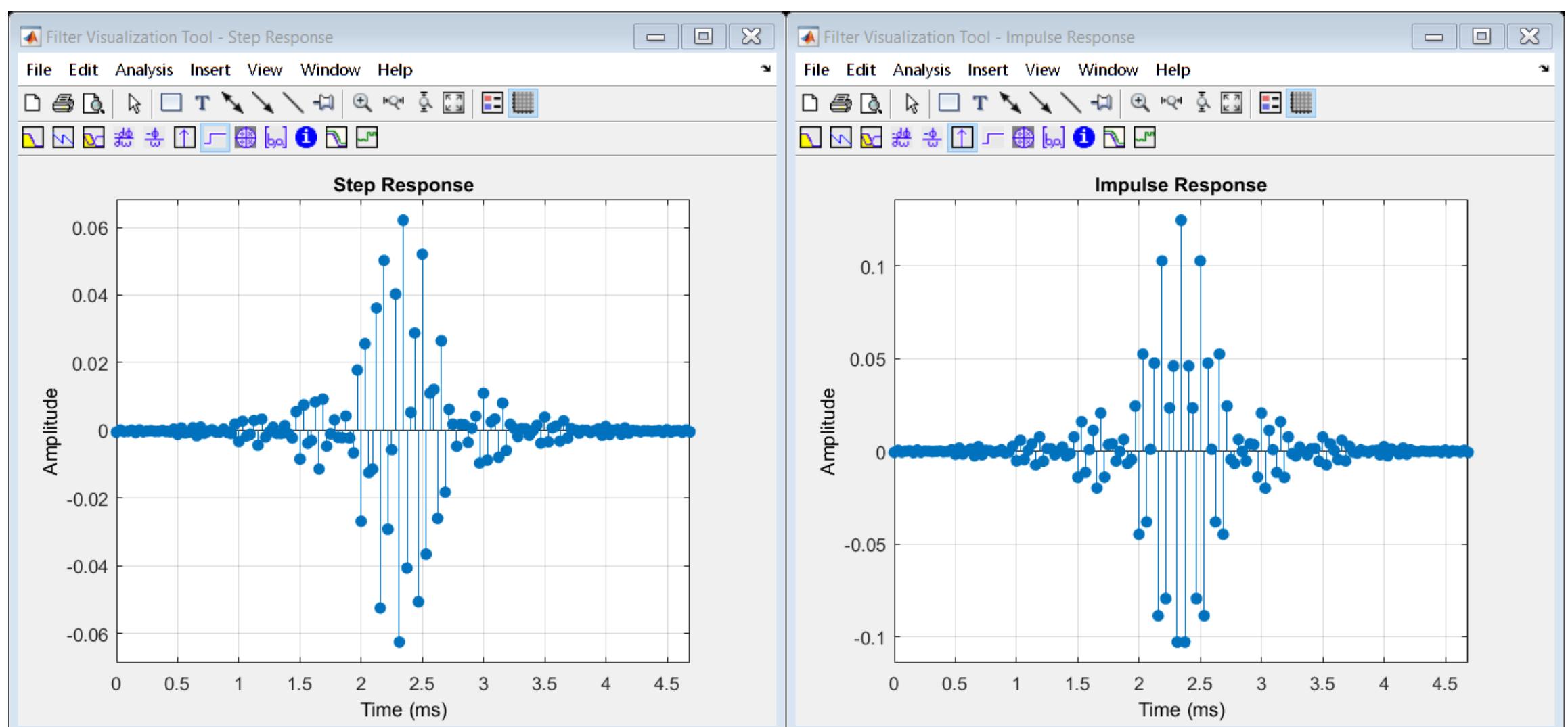
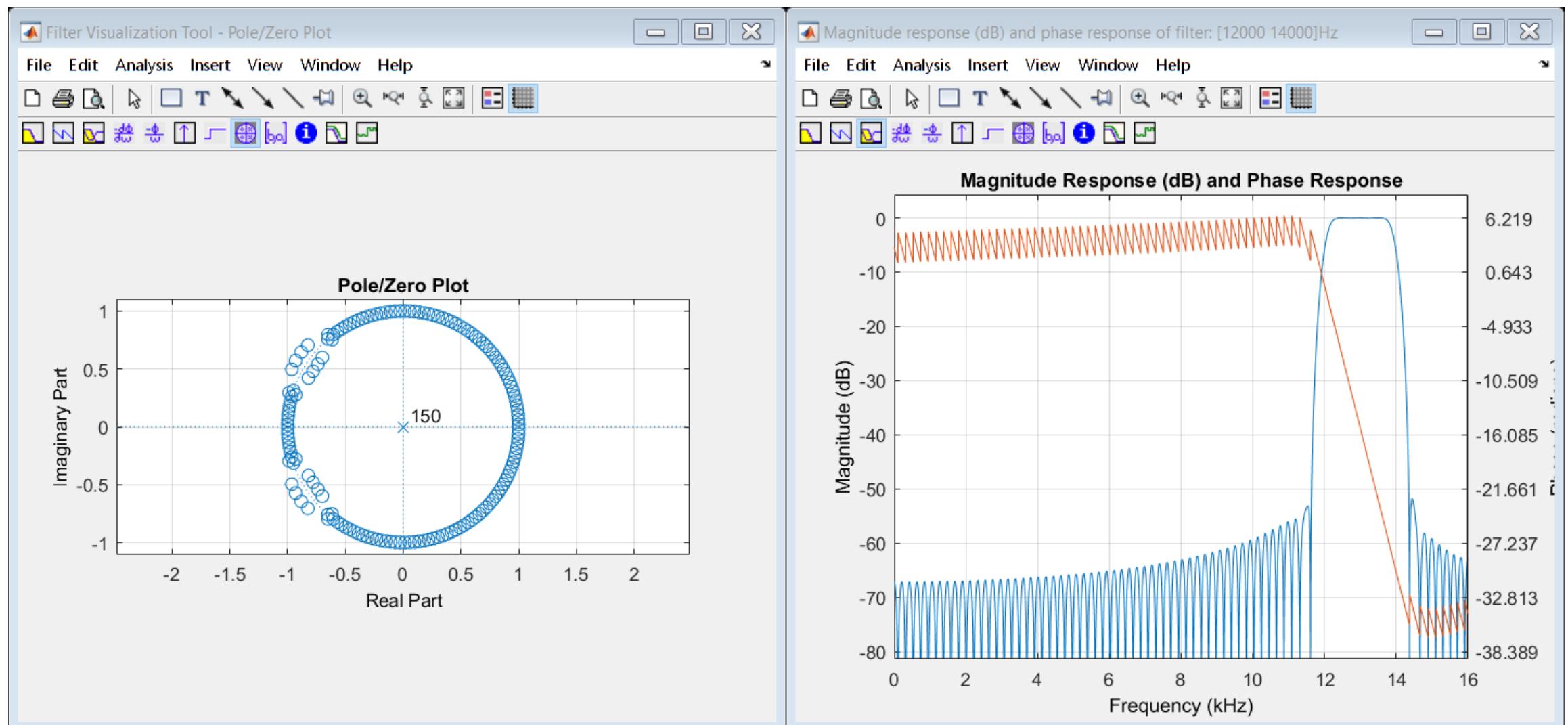
Sample run 3 (cont.):



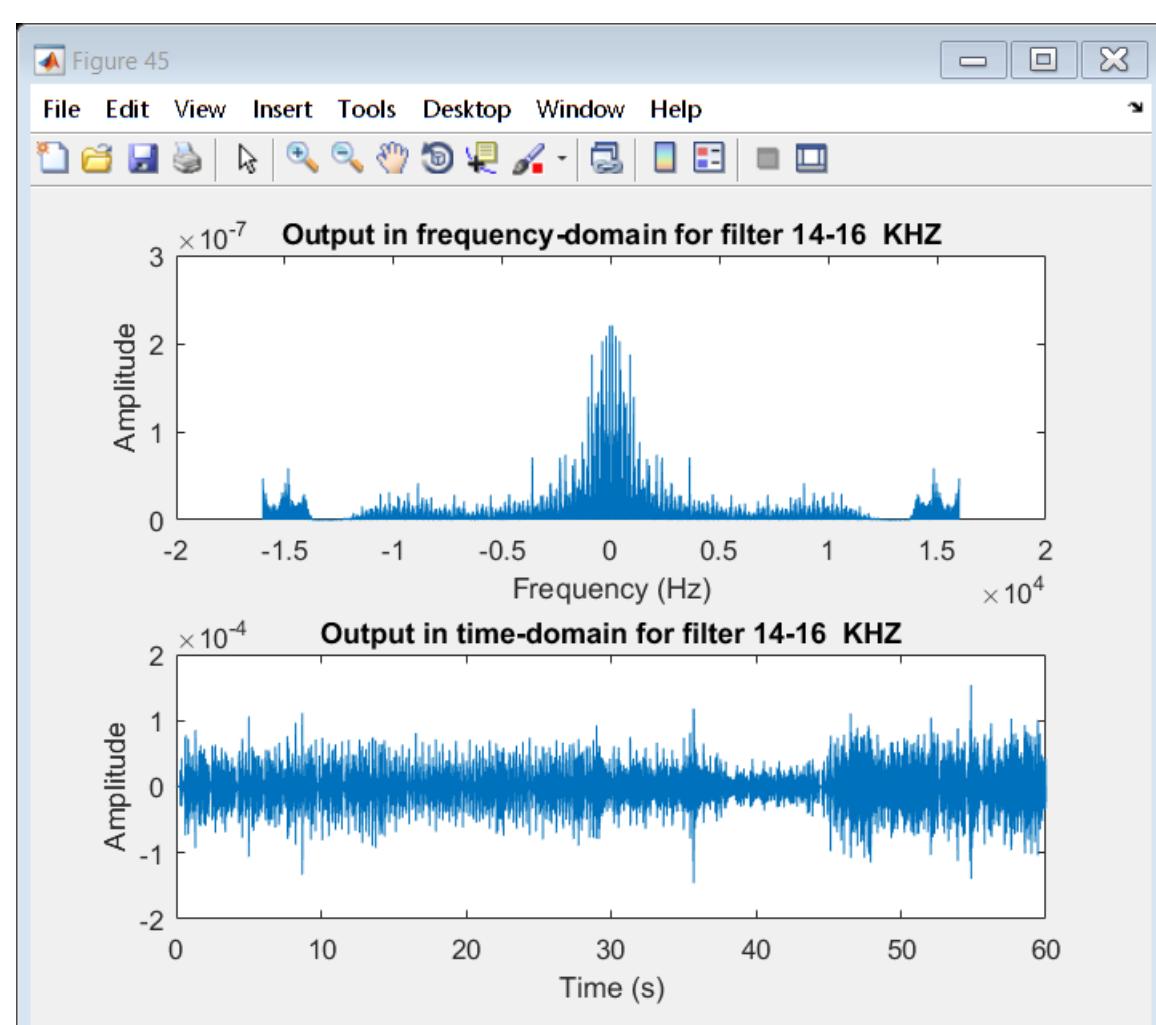
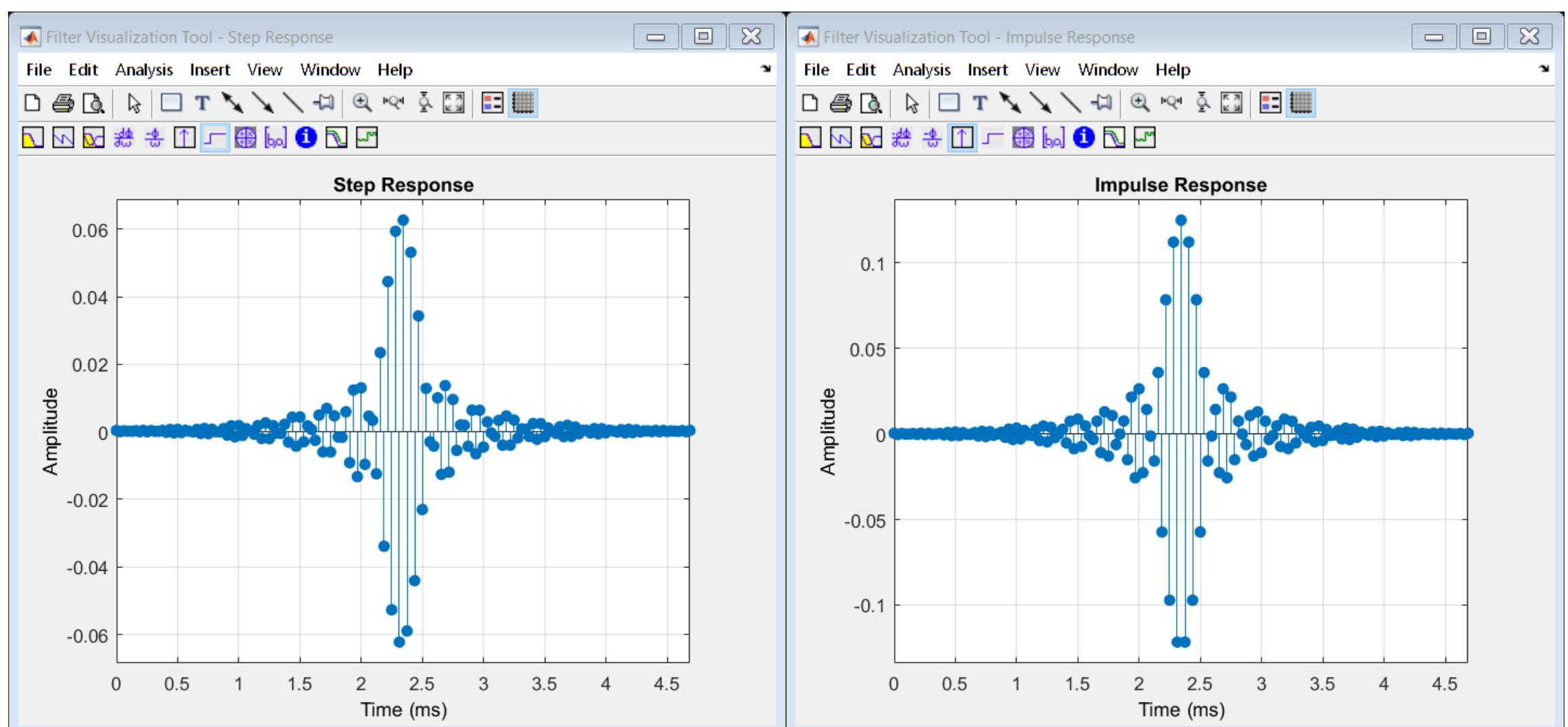
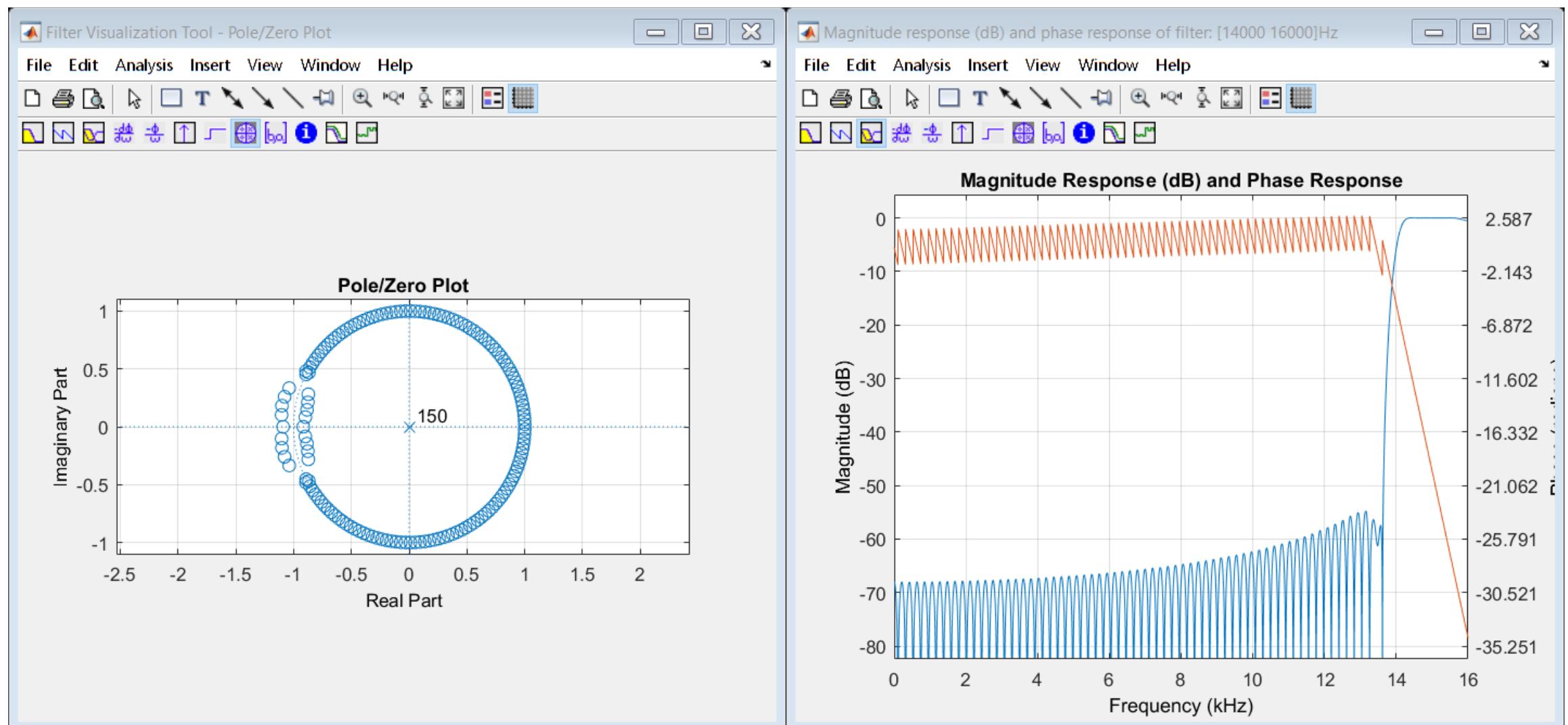
Sample run 3 (cont.):



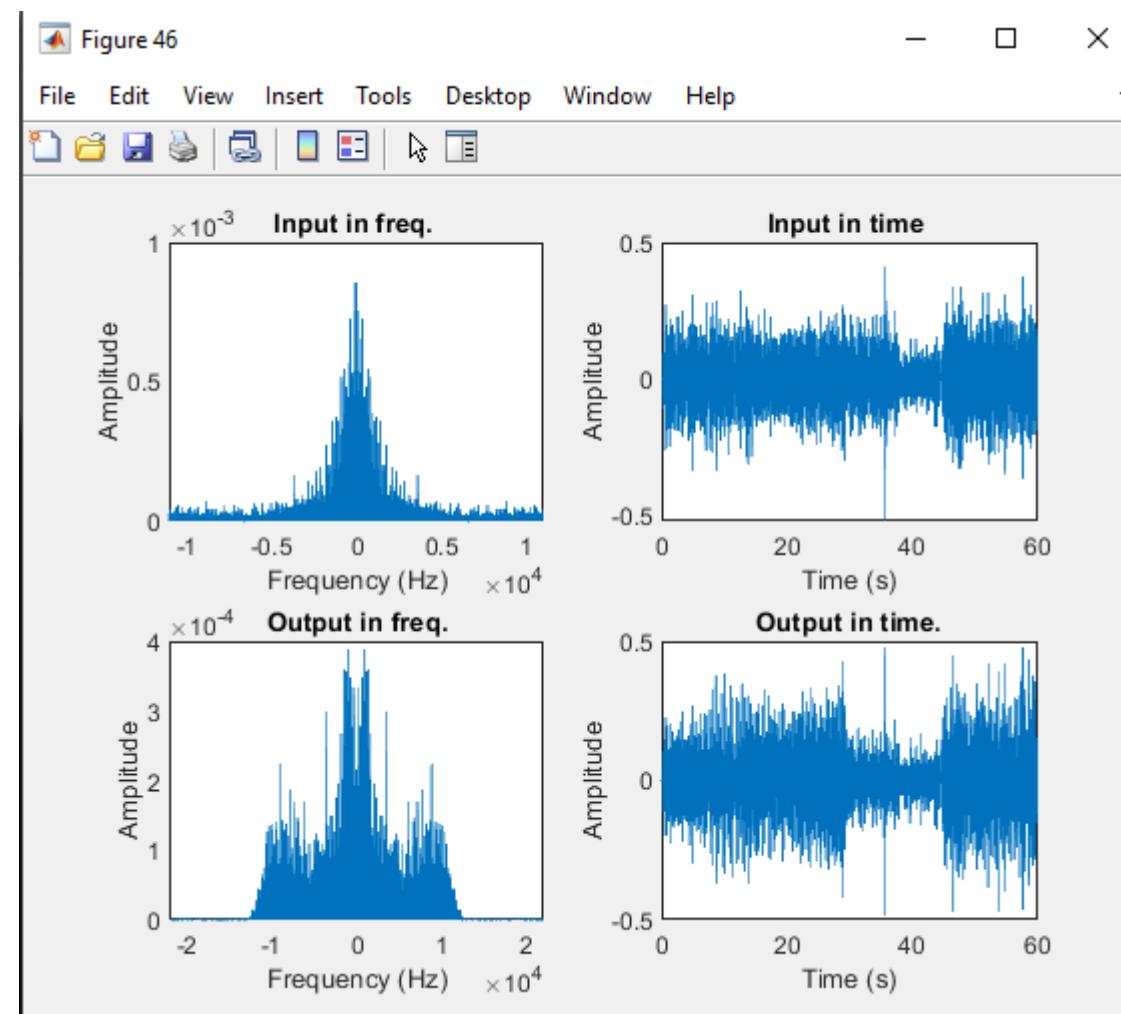
Sample run 3 (cont.):



Sample run 3 (cont.):



Sample run 3 (cont.):



Sample run 4 (double sample rate):

Type of filter: iir

Specified gains for each of the nine filters: 20, 15, 10, 5, 0, -5, -10, -15, -20 dB

Specified output frequency: 44100 Hz

Output:

File information:

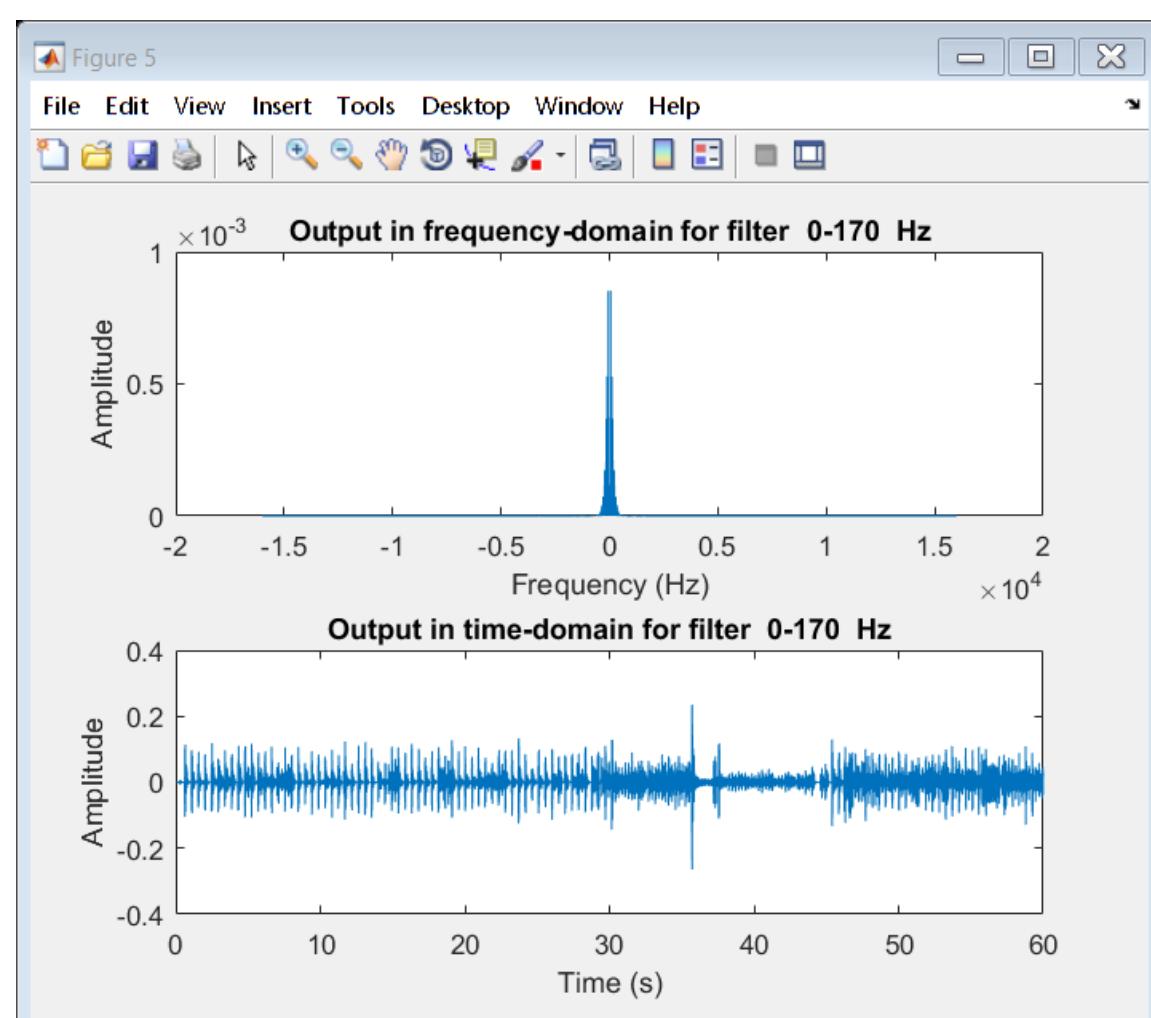
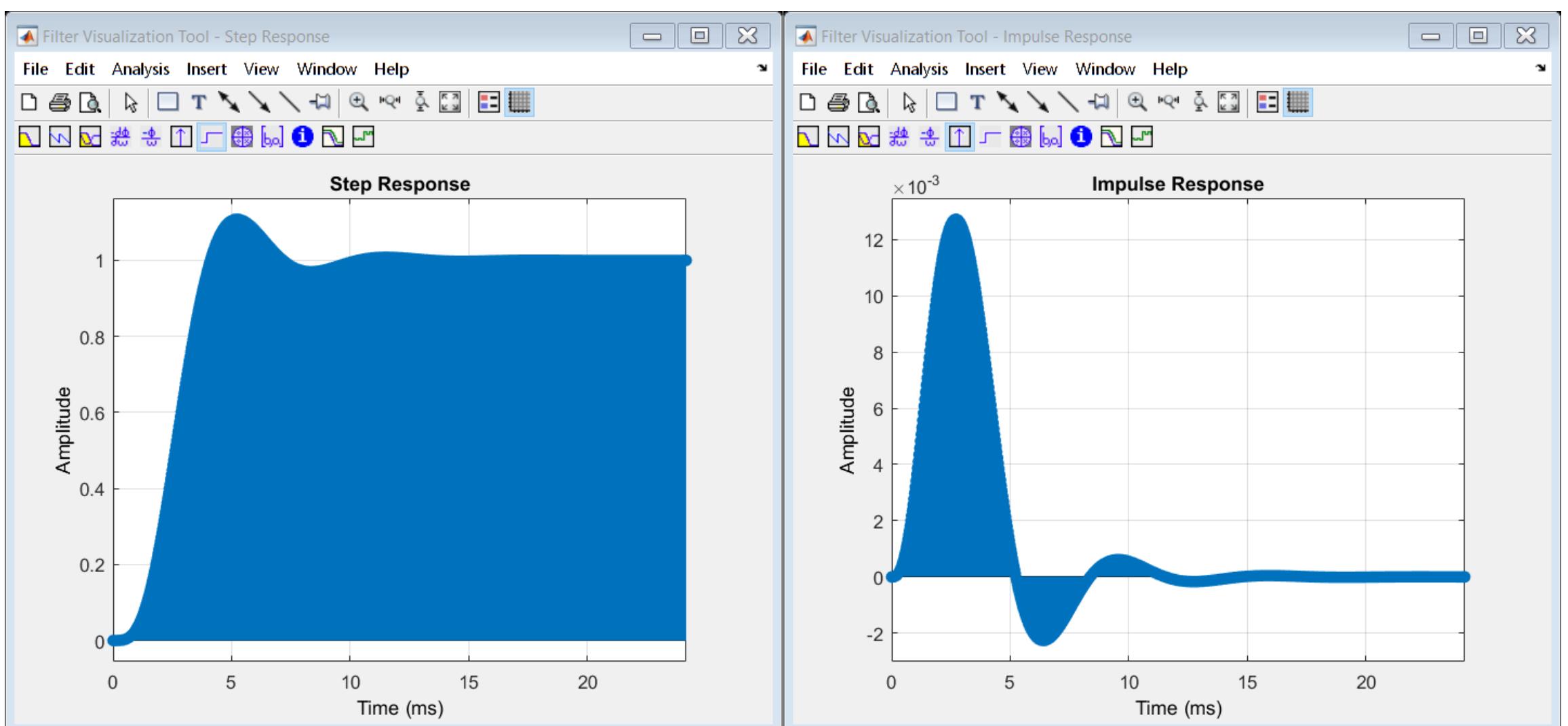
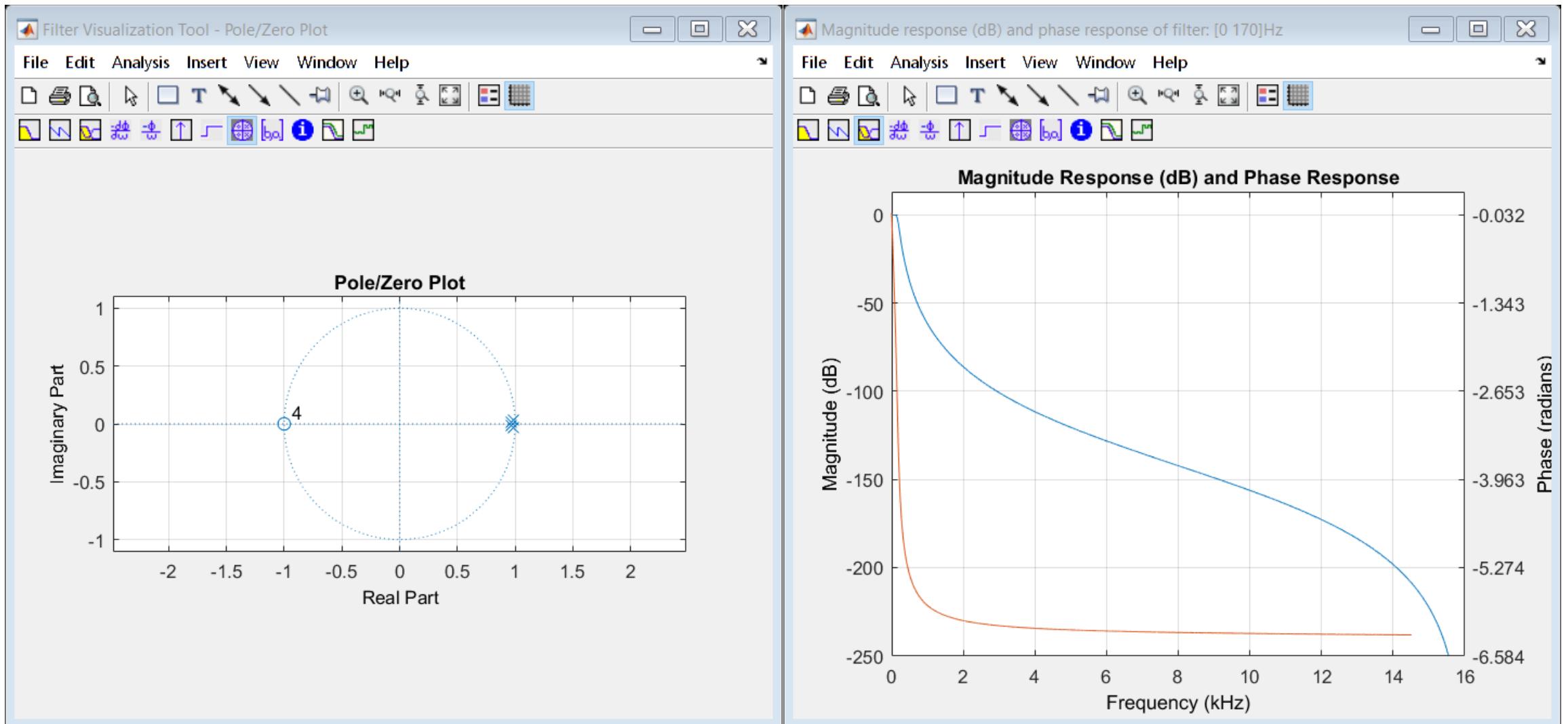
Path: D:\SSP\term 6\DSP\CantinaBand60.wav

Data dimensions: [1323000 1]

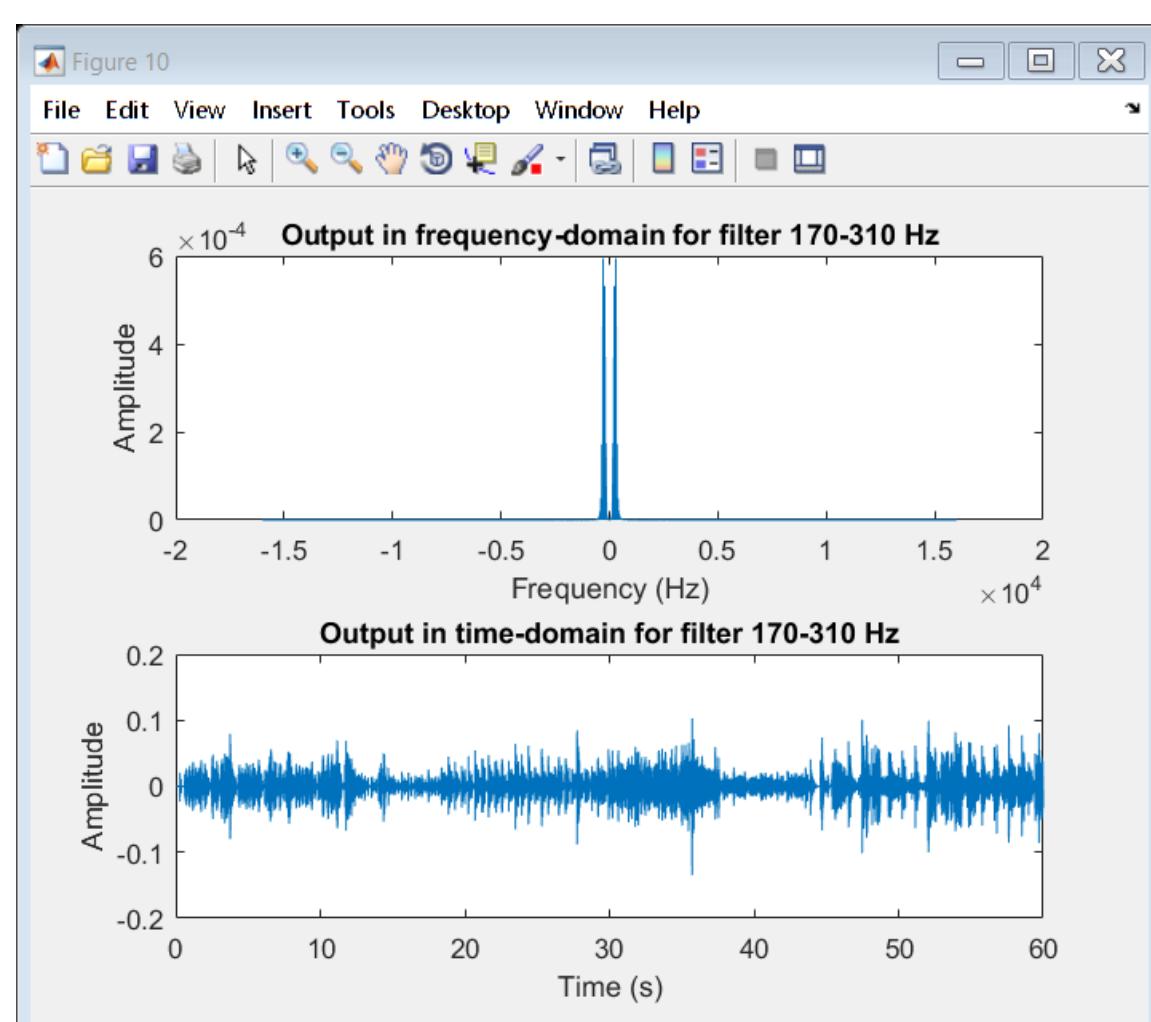
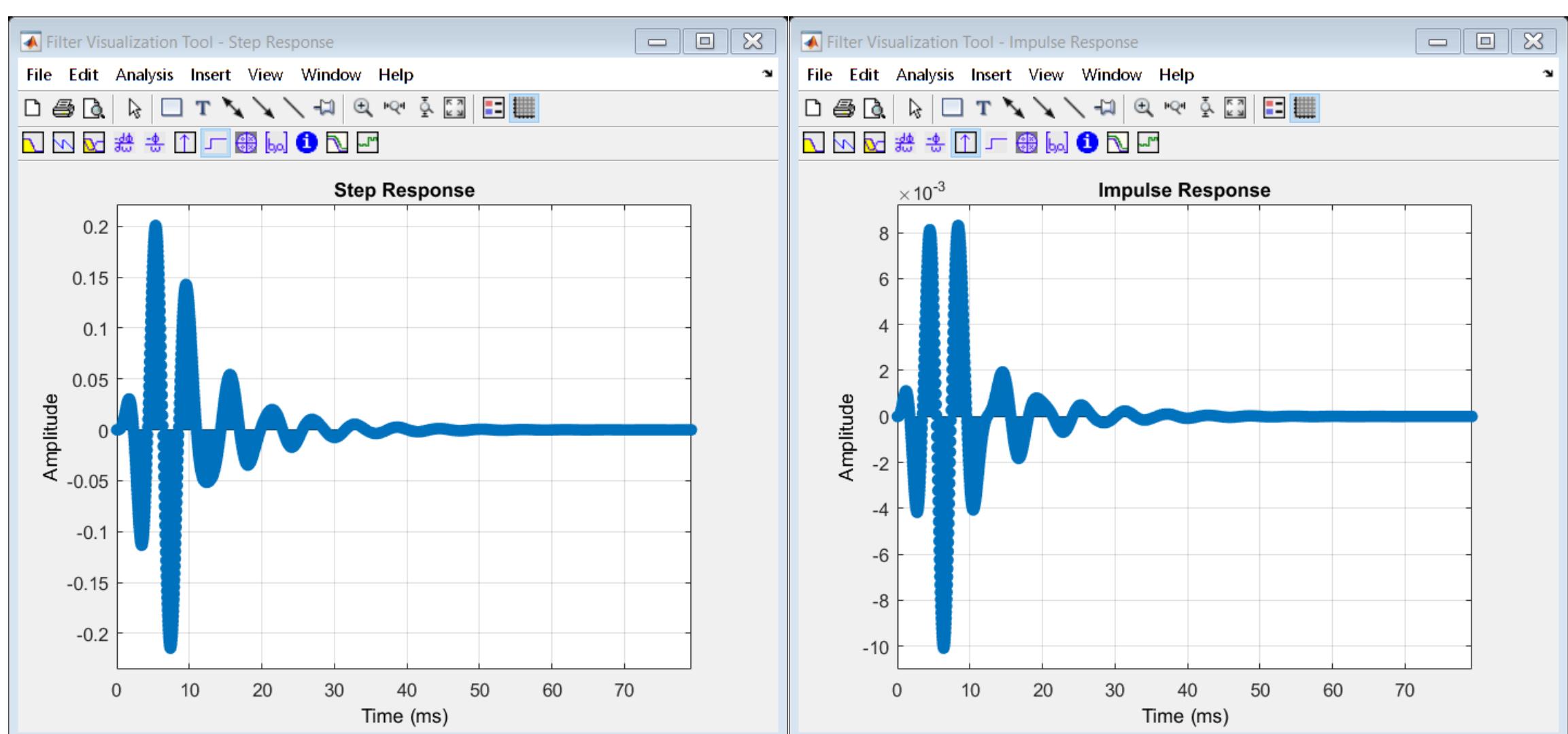
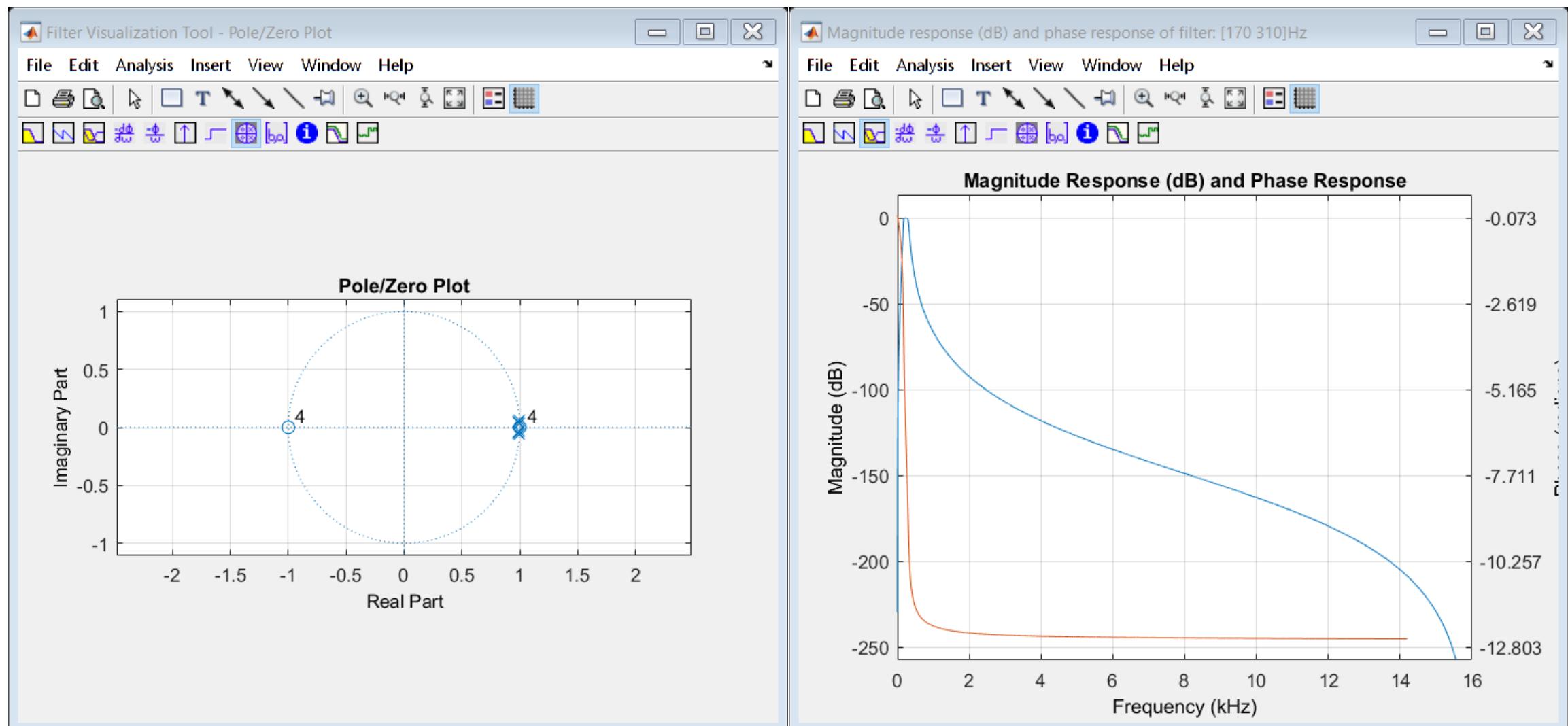
Frequency: 22050

The gain of iir filter : 0-170 Hz is 0.000000 , order is 4
The gain of iir filter : 170-310 Hz is 0.000000 , order is 8
The gain of iir filter : 310-600 Hz is 0.000001 , Order is 8
The gain of iir filter : 0.6-1 kHz is 0.000002 , order is 8
The gain of iir filter : 1-3 kHz is 0.000931 , Order is 8
The gain of iir filter : 3-6 kHz is 0.003860 , order is 8
The gain of iir filter : 6-12 kHz is 0.037890 , Order is 8
The gain of iir filter : 12-14 kHz is 0.000931 , order is 8
The gain of iir filter : 14-16 KHz is 0.000931 , order is 8
Warning: Data clipped when writing file.

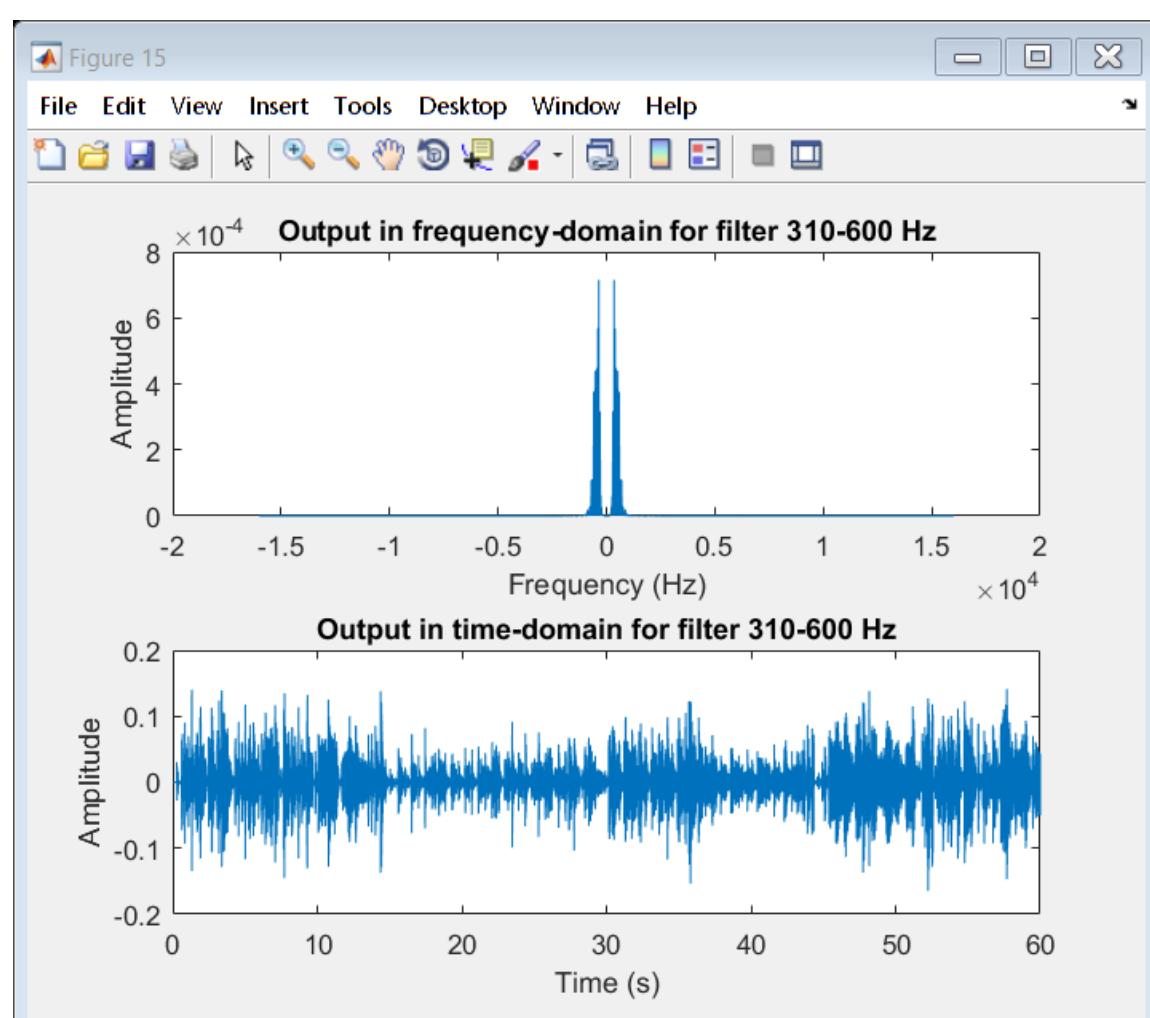
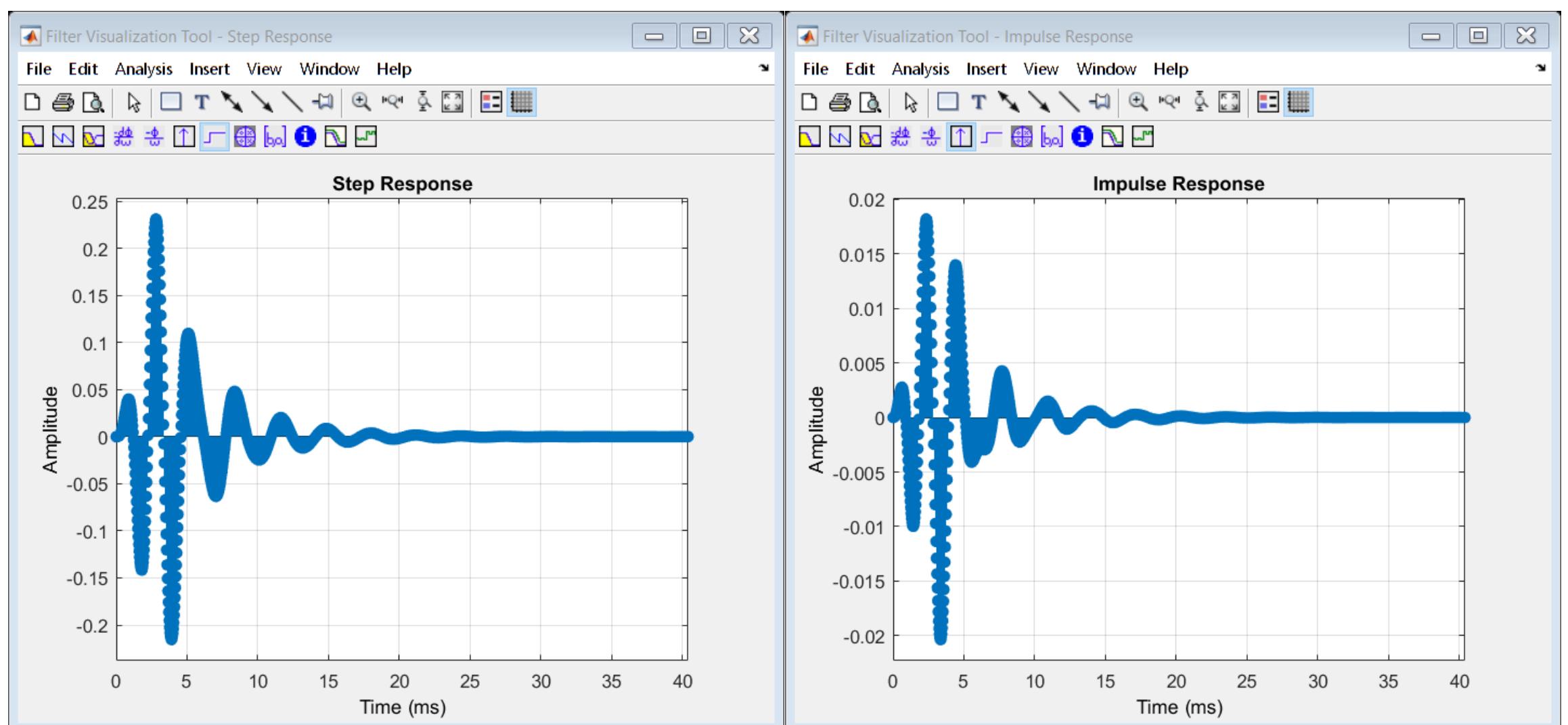
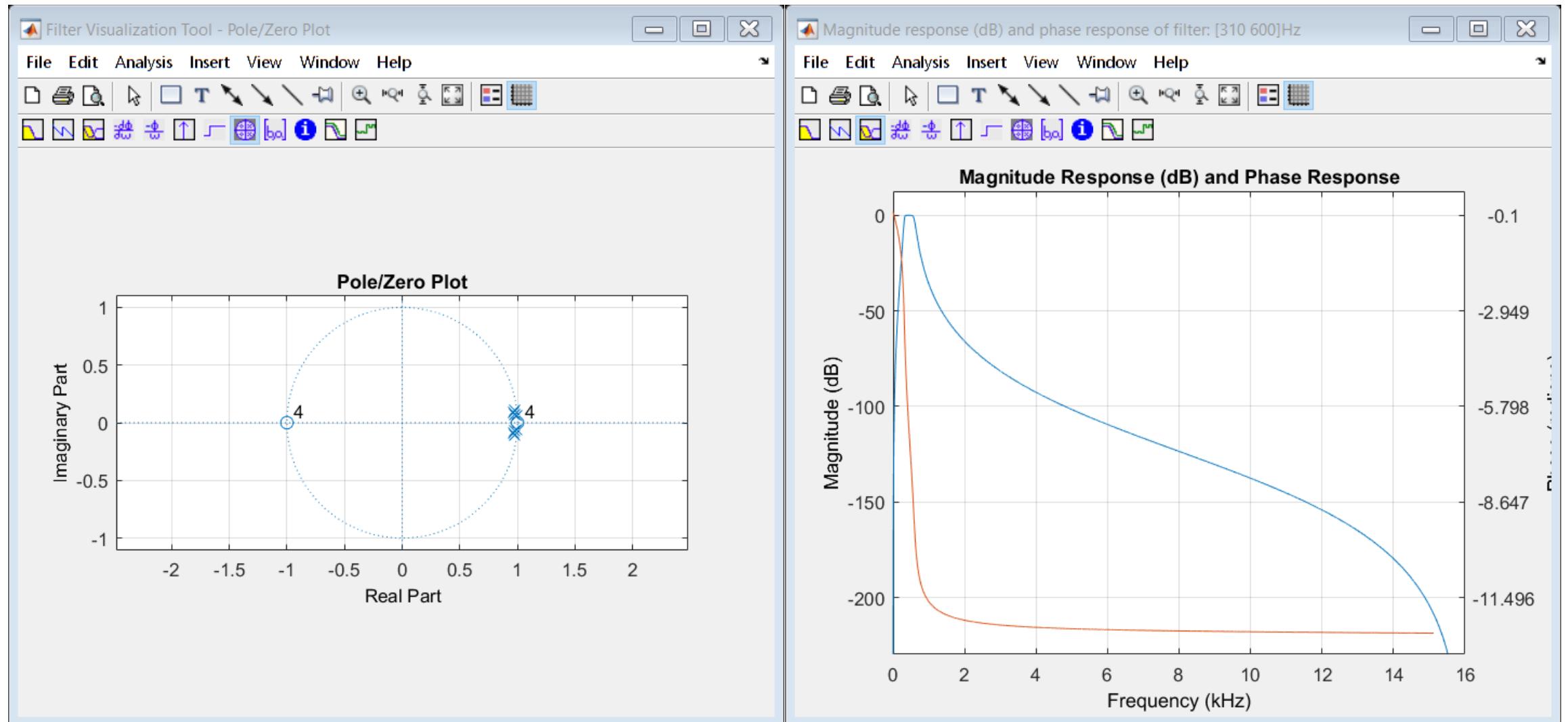
Sample run 4 (cont.):



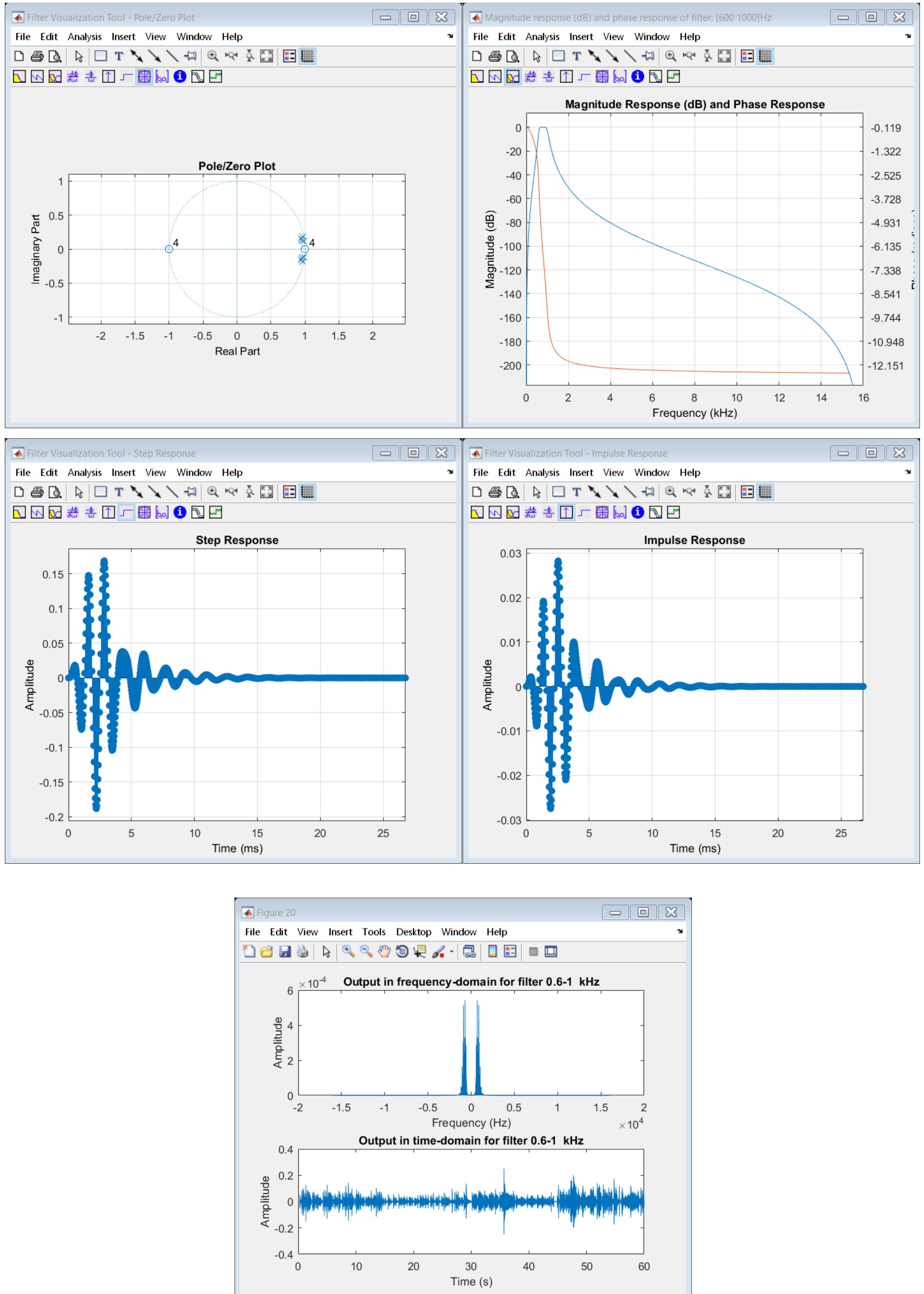
Sample run 4 (cont.):



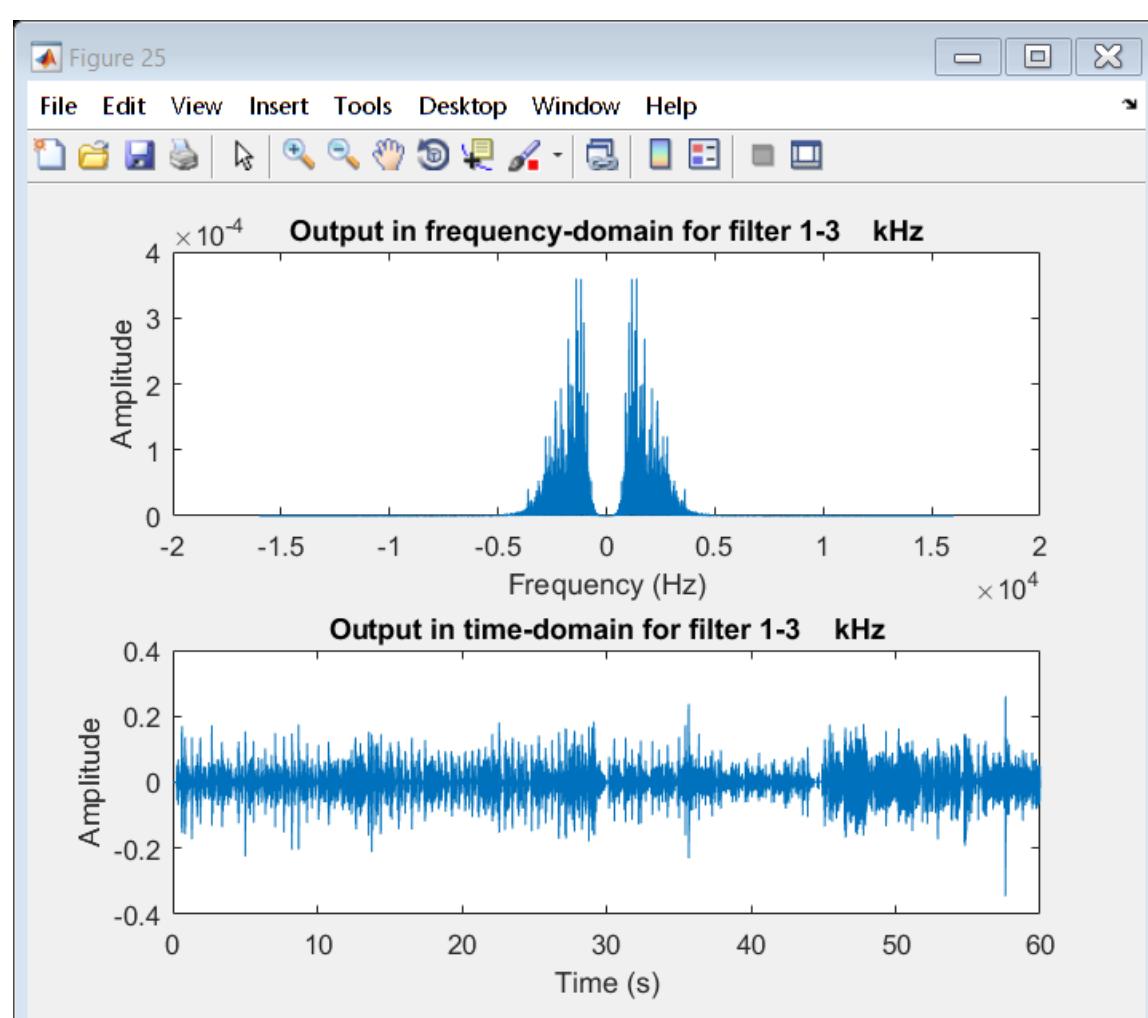
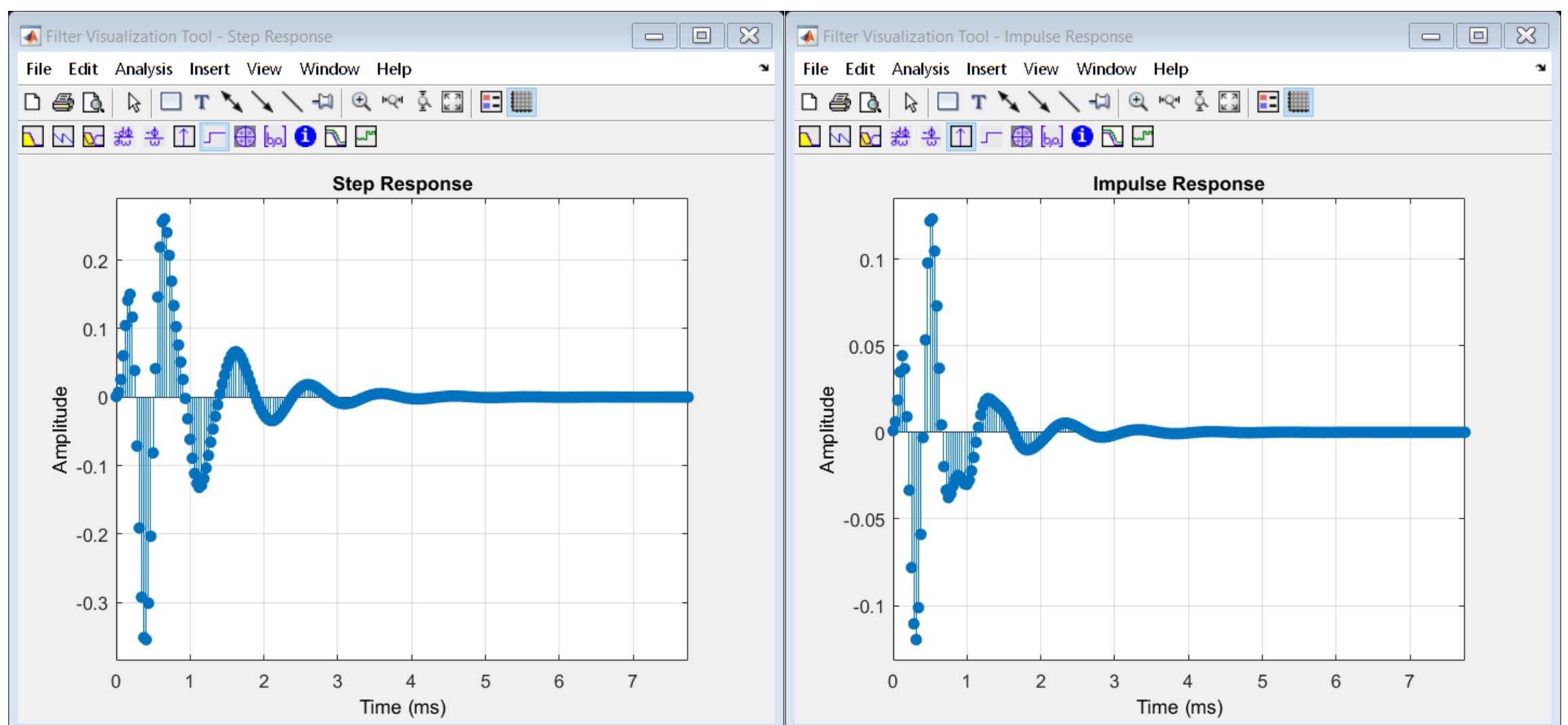
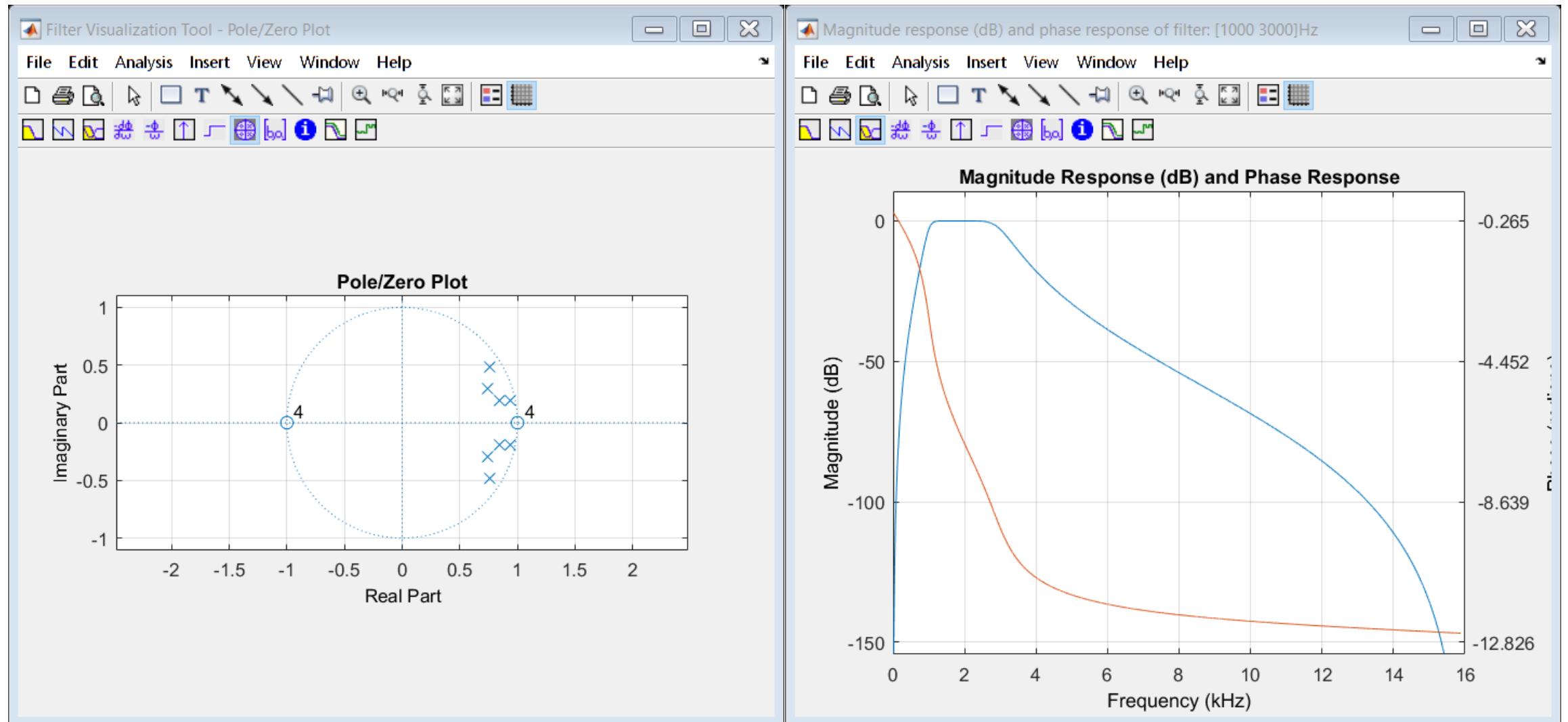
Sample run 4 (cont.):



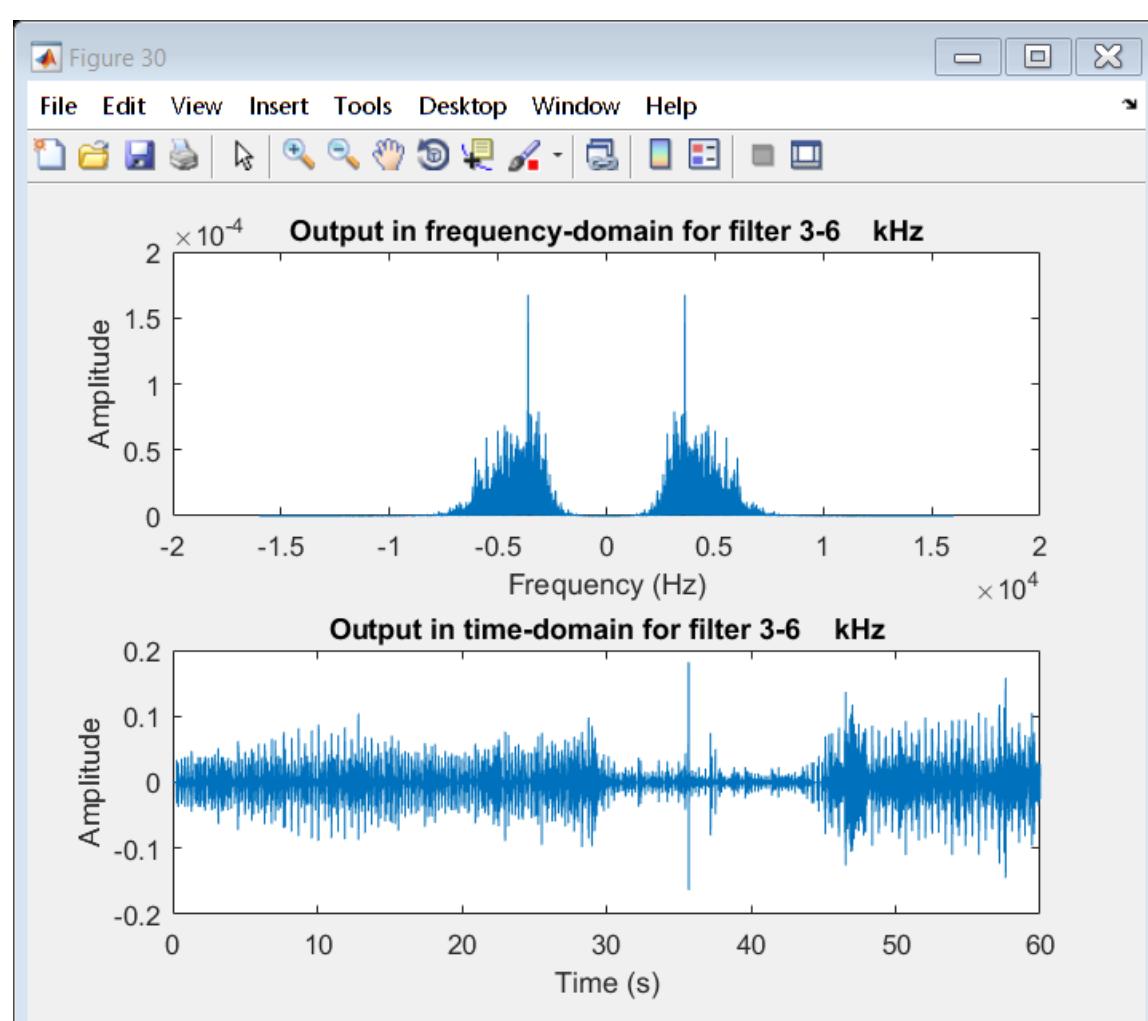
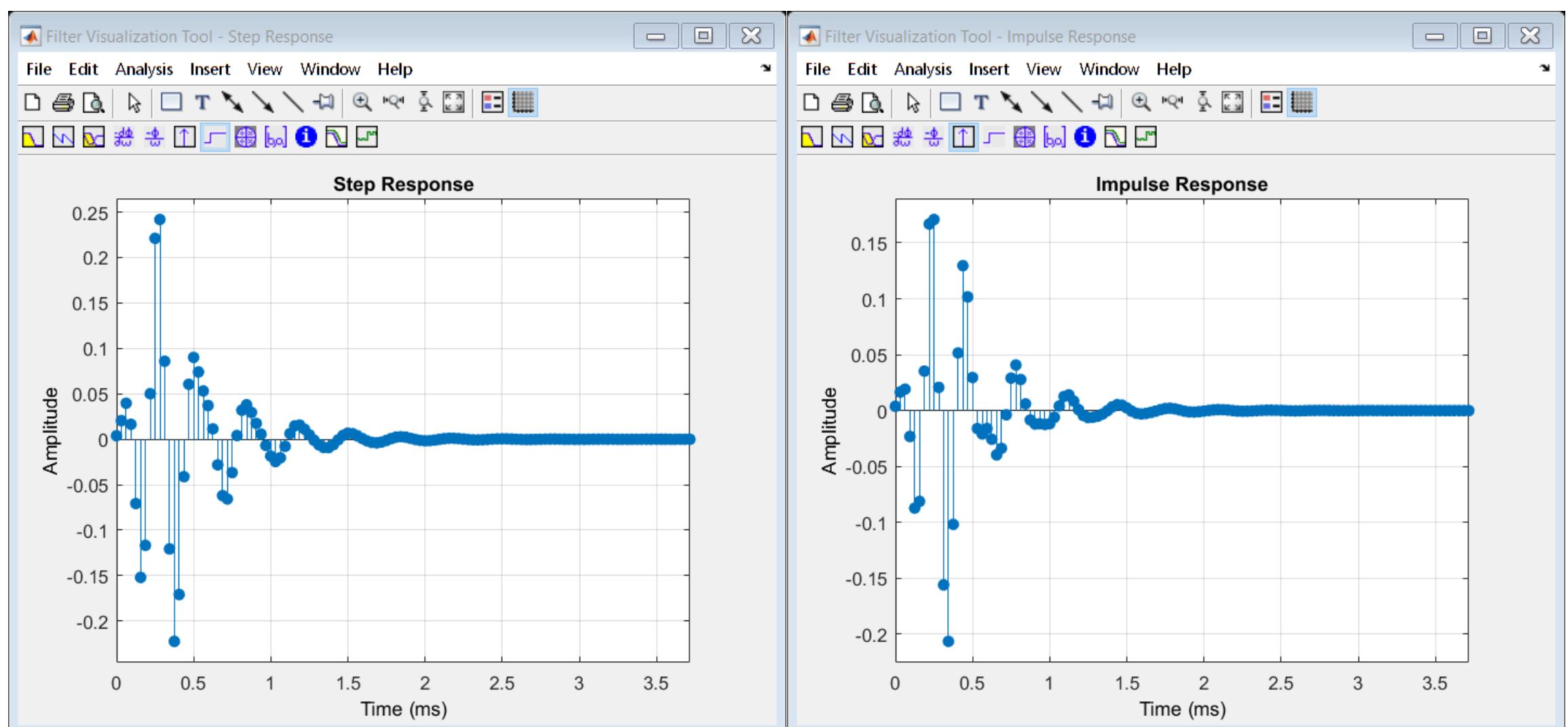
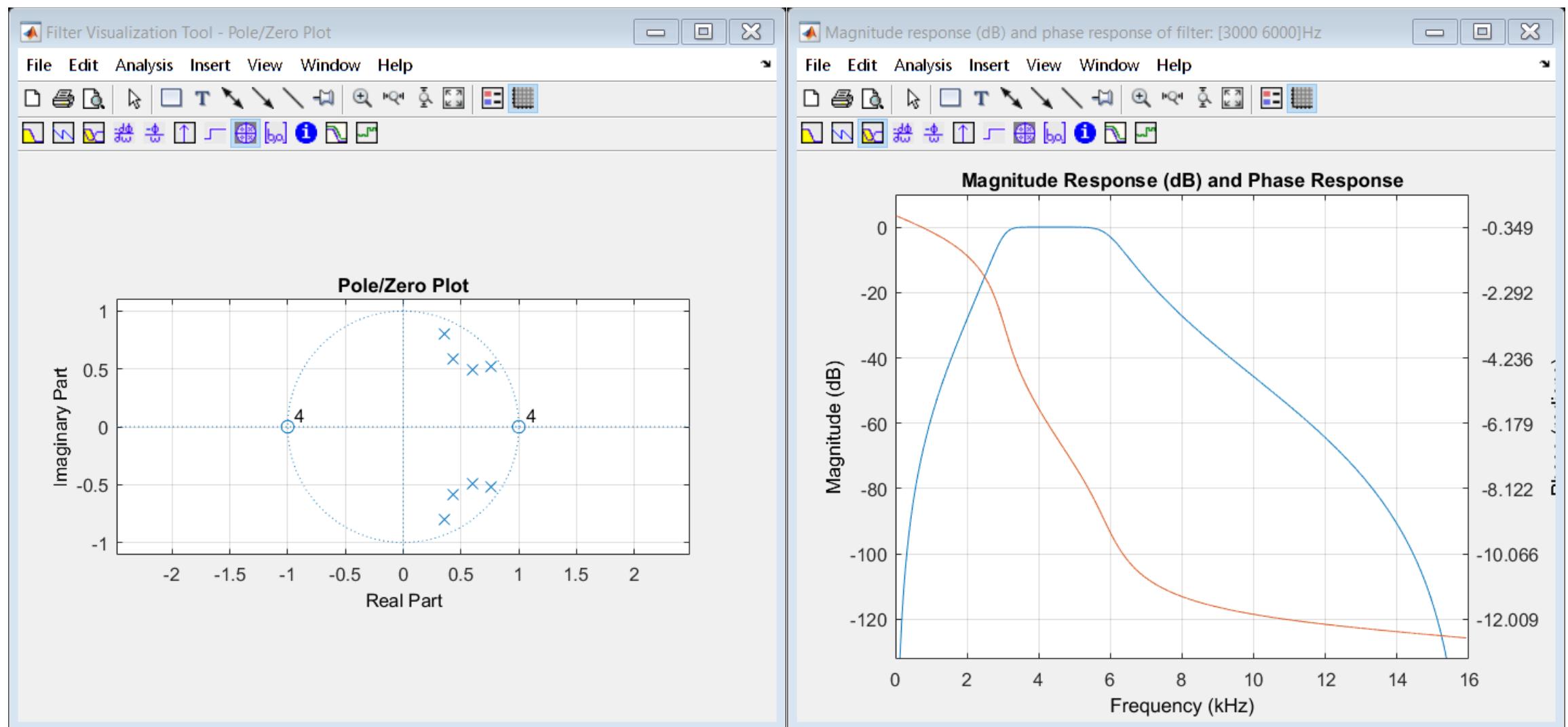
Sample run 4 (cont.):



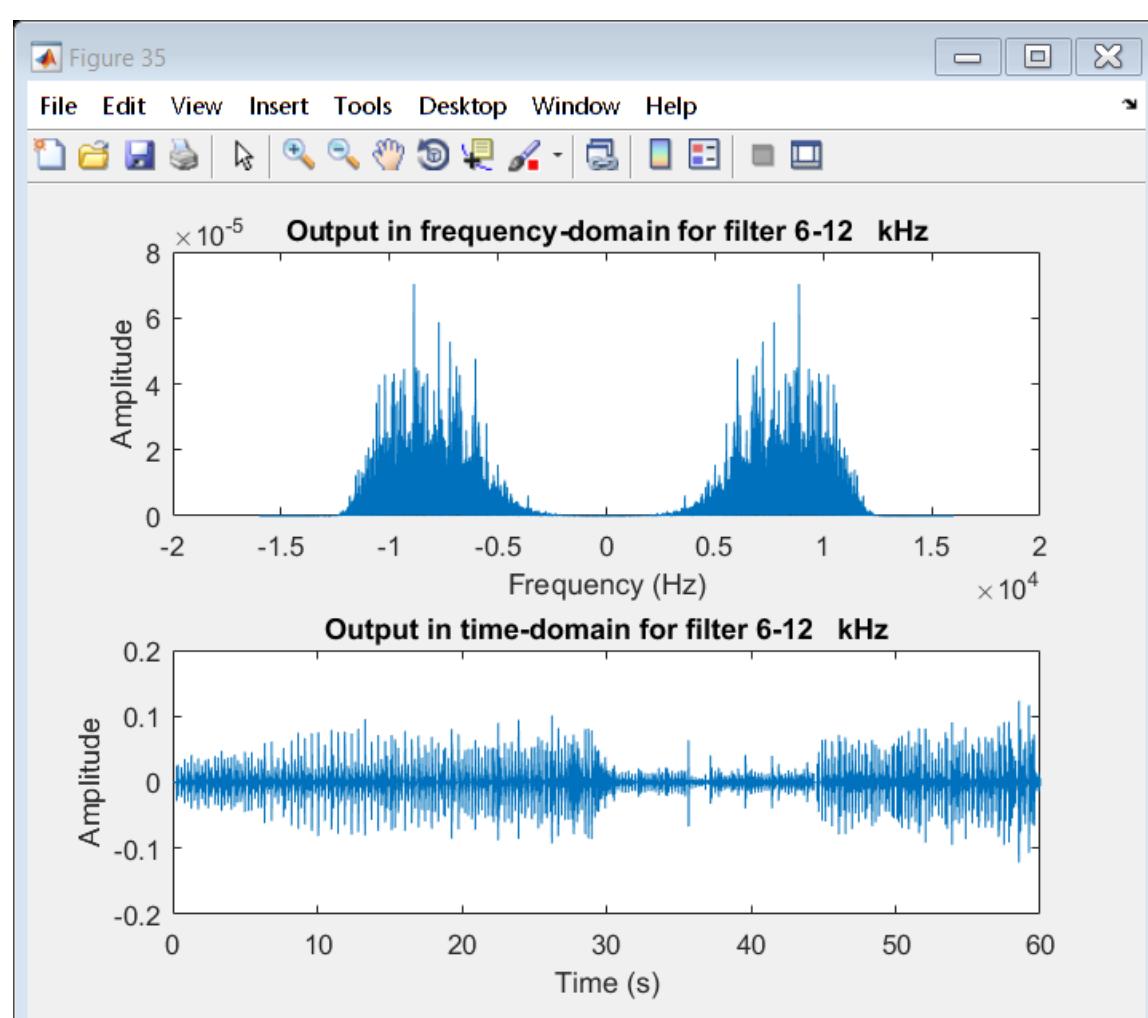
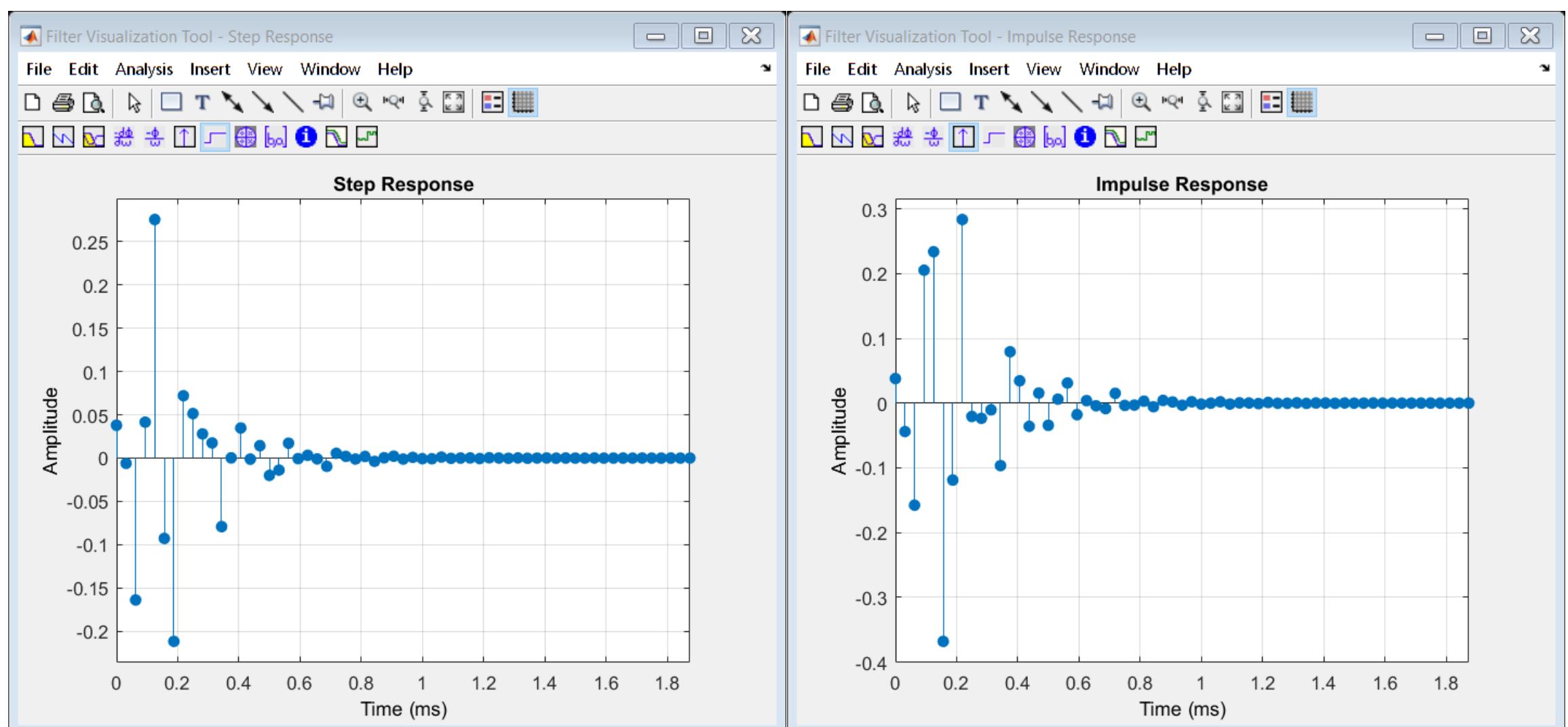
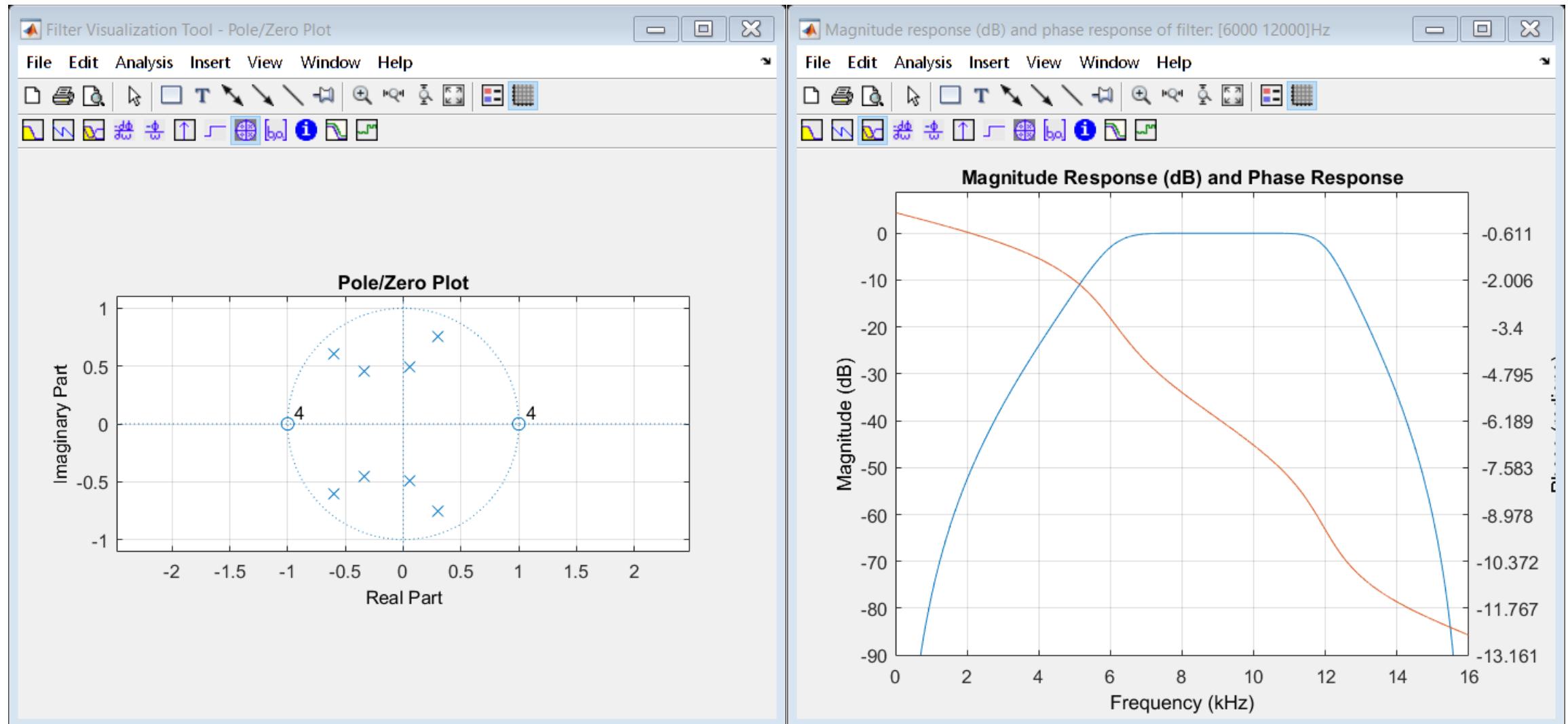
Sample run 4 (cont.):



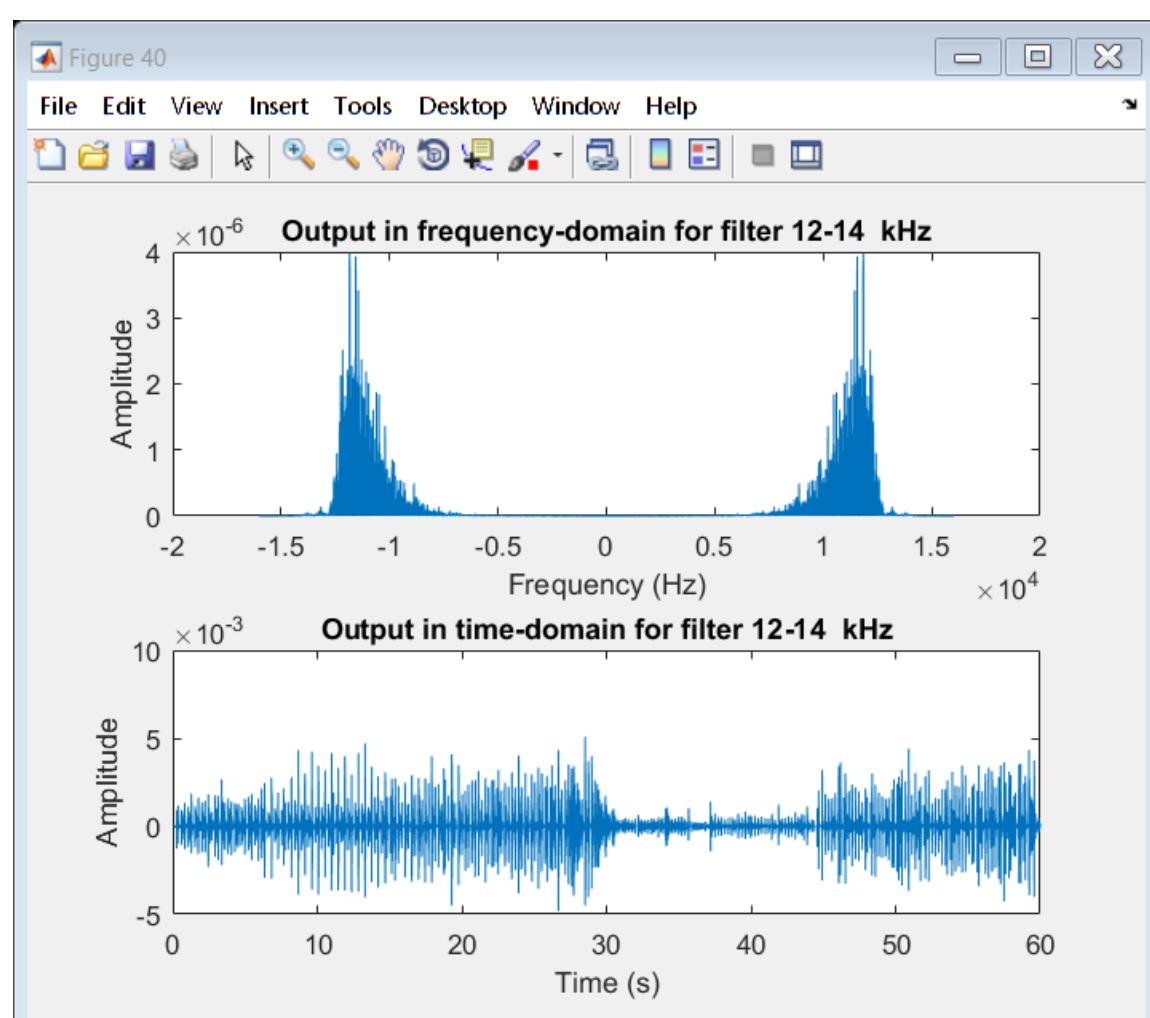
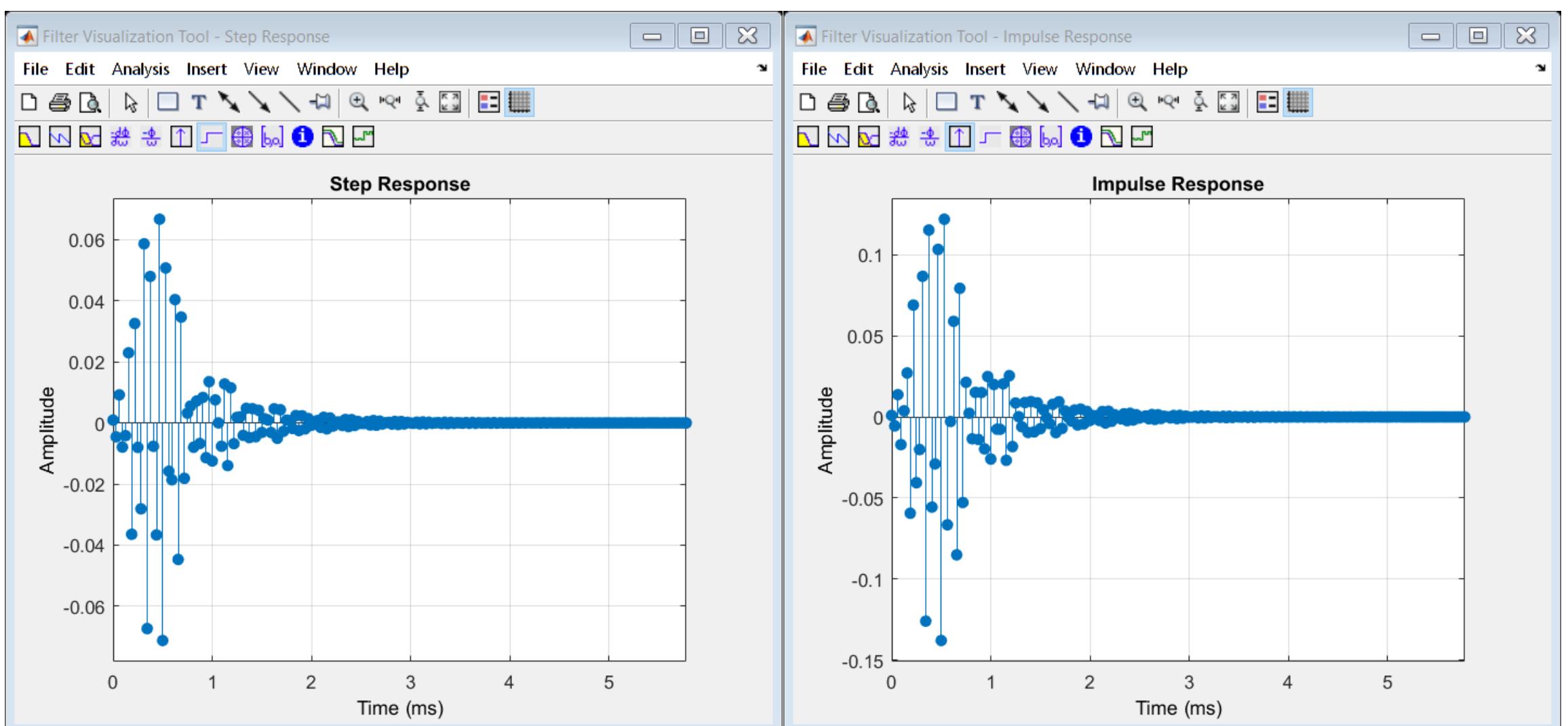
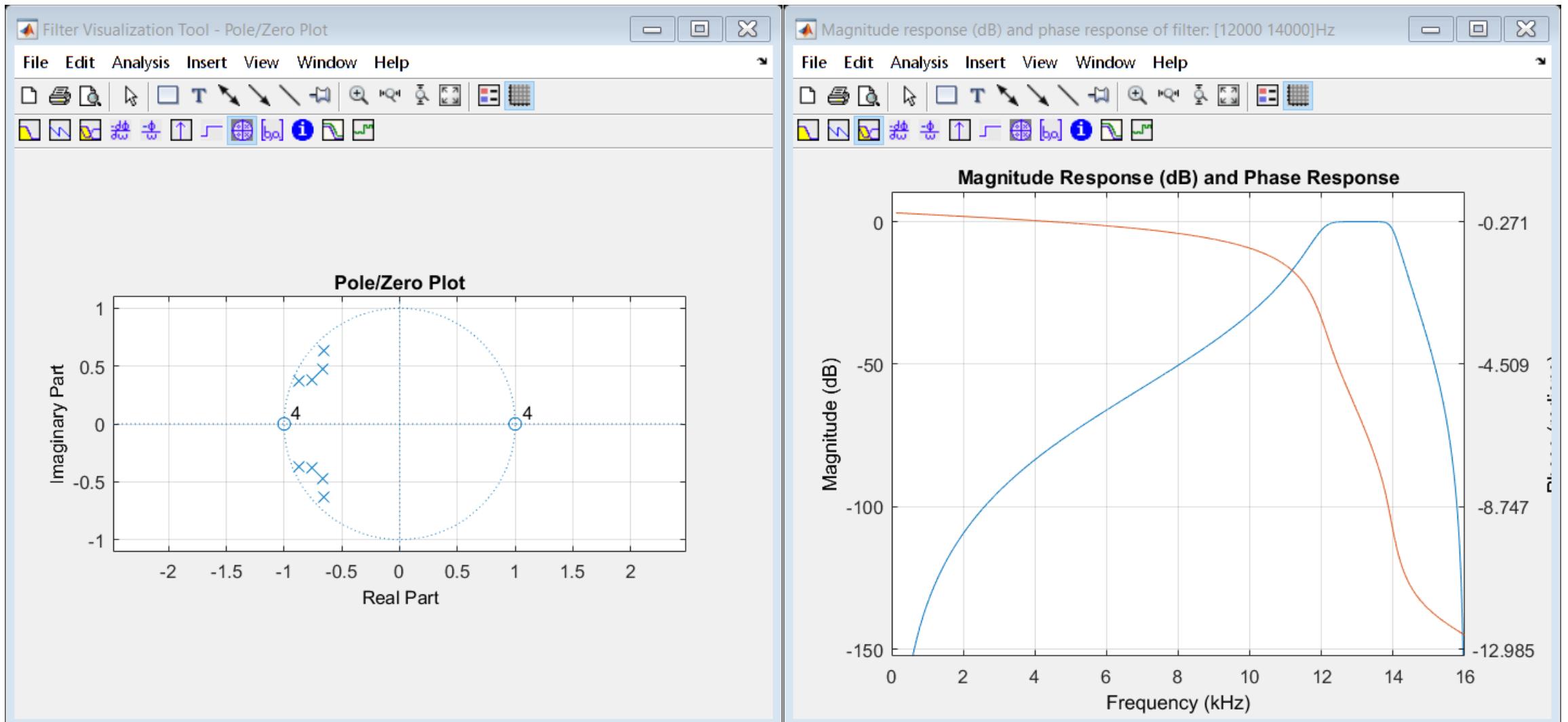
Sample run 4 (cont.):



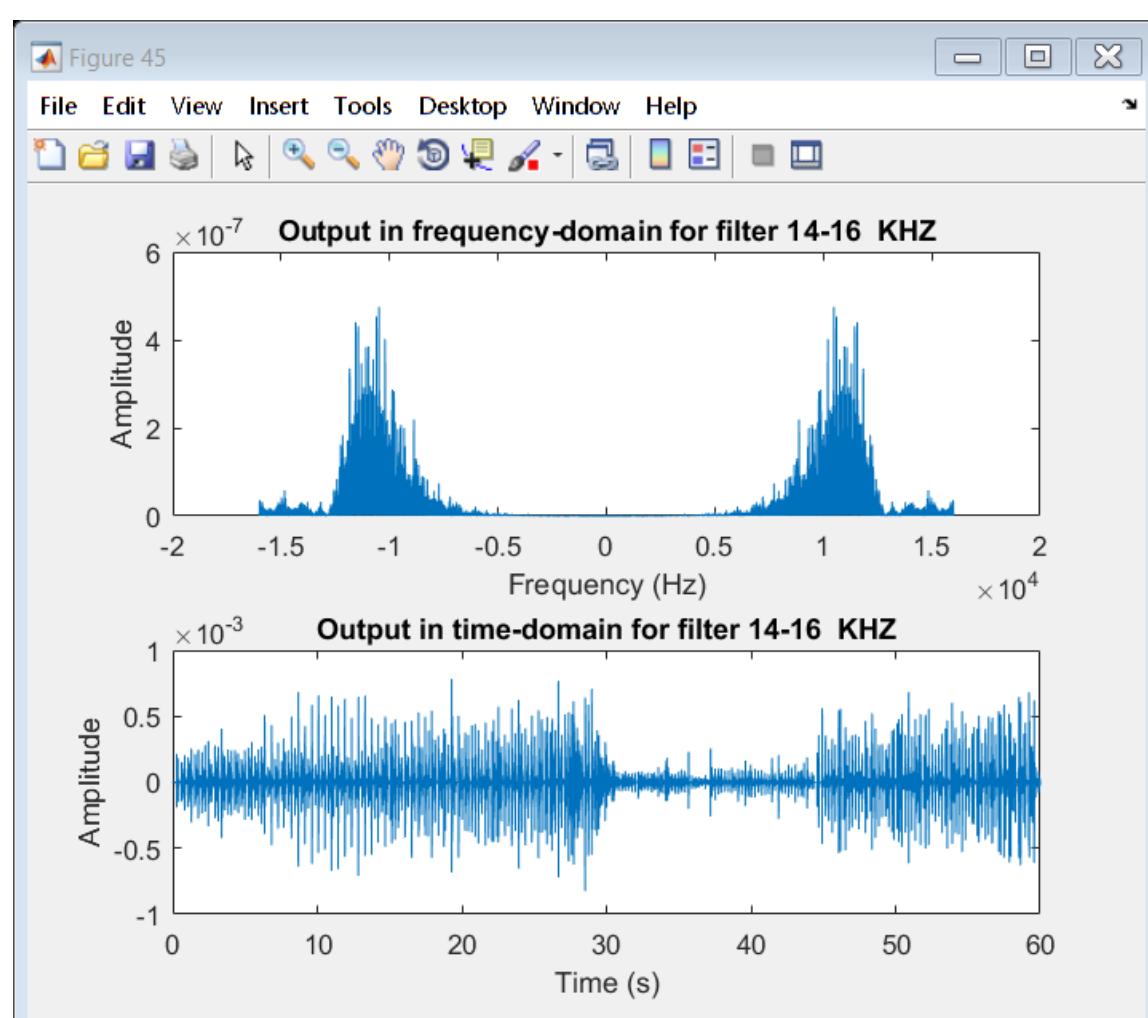
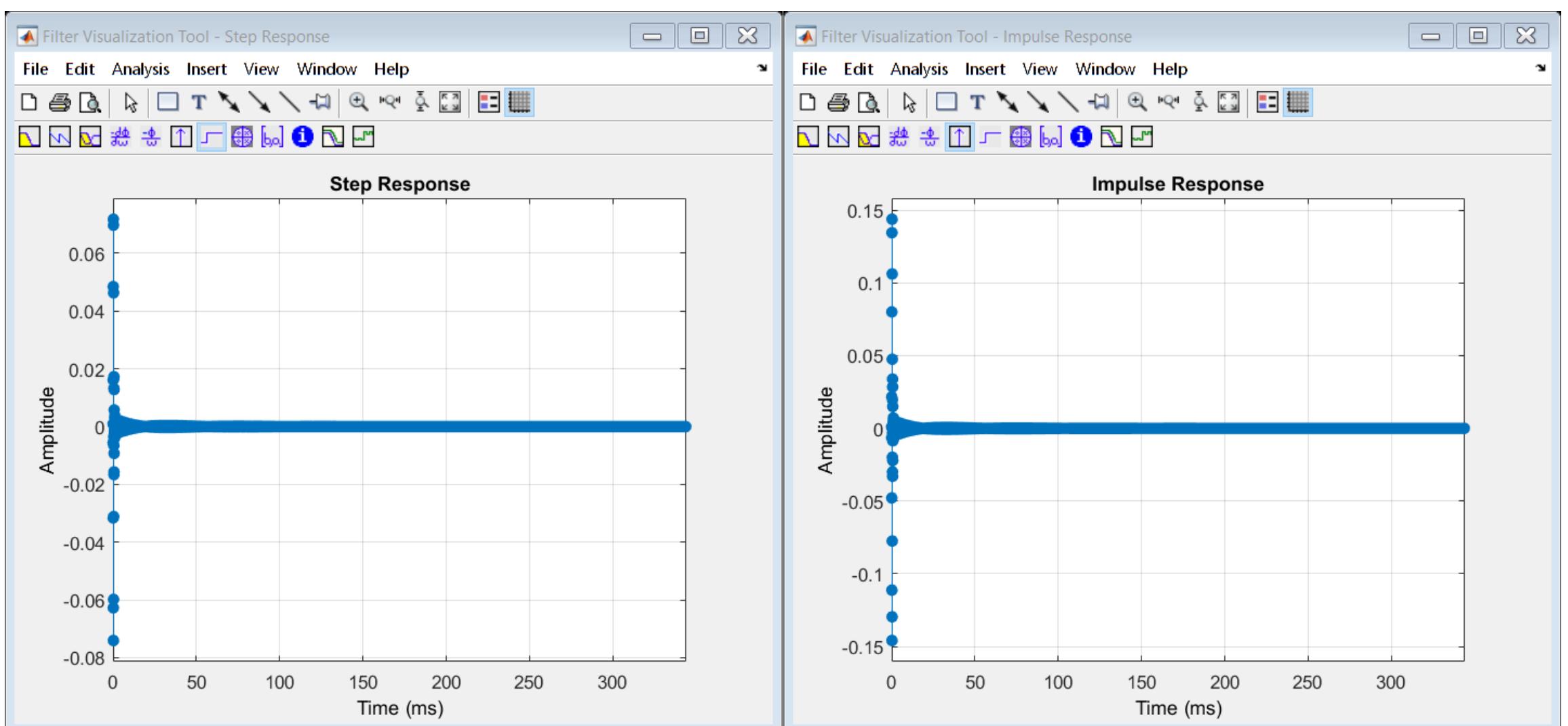
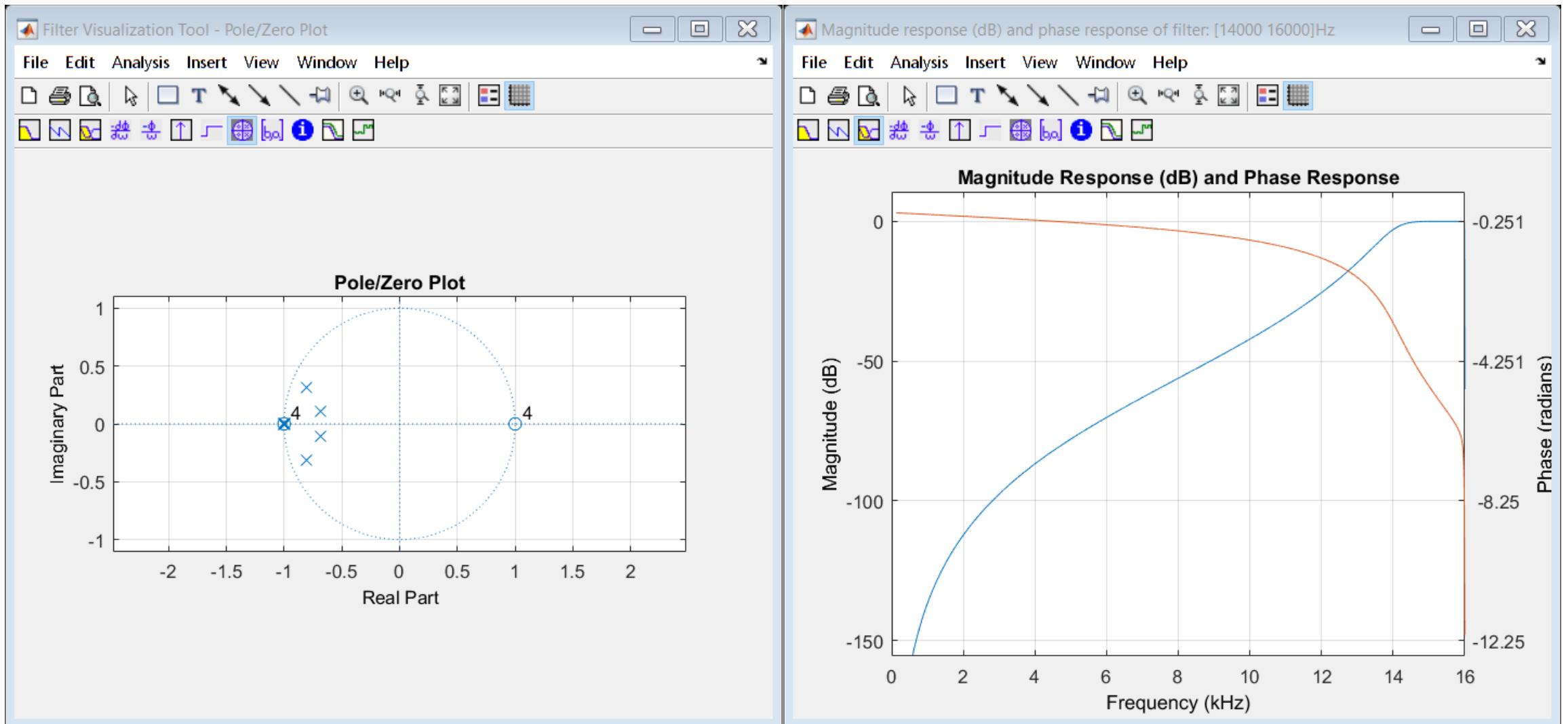
Sample run 4 (cont.):



Sample run 4 (cont.):



Sample run 4 (cont.):



Sample run 4 (cont.):

