CMPE 323: Signals and Systems

Dr. LaBerge

Lab 04 Report:

More Convolution

Sabbir Ahmed

Table of Content

1. Introduction……3

2. Experiment……3

3. Procedure……3

3.1 A Simple Filter……3

3.2 A Less Simple Filter……3

3.3 A Mechanization of this Lab……3

3.4 Complex Eigenvalues……3

4. Results……4

5. Appendices……11

5.1 Appendix A……8

1. Introduction

In this lab, the convolution integral

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau = \int_{-\infty}^{\infty} x(t-\tau)h(\tau)d\tau$$

will be determined to be a filter operation.

2. Equipment

A computer with MATLAB installed.

3. Procedure

3.1 A Simple Filter

Using a sample rate of 1 kHz, and a time record from -1 to +10 seconds, create a simple rectangular pulse $p(t, 0.25)$ and use it as the system impulse response. Then create system inputs $x(t; f) = e^{j2\pi ft} \times u(t)$. Using the MATLAB $conv(x, h)$ function, compute and plot the magnitude of the output for frequencies $f = 0, 1, 4, 6.5, 8.5, 11.5, 19.5$. Comment on whether this is a high pass, low pass or band pass filter.

Replot the results, discarding the convolution tail at the end of the output array. Then, recompute the convolution and replot the results using only the non-zero element of h. Comment on any changes on the outputs.

3.2 A Less Simple Filter

Now compute a new $h_2(t) = h(t) \times e^{j15\pi t}$. Compute and plot the magnitude of the convolution output of the system to the same set of exponentials. Explain the results, including the identification of the type of filter.

3.3 A Mechanization of this Lab

Create a MATLAB function to return an array of complex outputs representing the steady state output at each of the M frequencies.

3.4 Complex Eigenvalues

Use the function to compute and plot the responses of $h(t)$ and $h_2(t)$ from parts 3.1 and 3.2 to complex exponentials with circular frequencies $f = [0: 0.1: 50]$ Hz. Plot the amplitude and phase of the output as a function of f.

4. Results

The pulse of duration 0.25 was created; $h = u(t) - u(t - 0.25)$. The system input $x(t; f) = e^{j2\pi ft} \times u(t)$ was convoluted with the pulse and the following plots were created corresponding to the different frequencies:

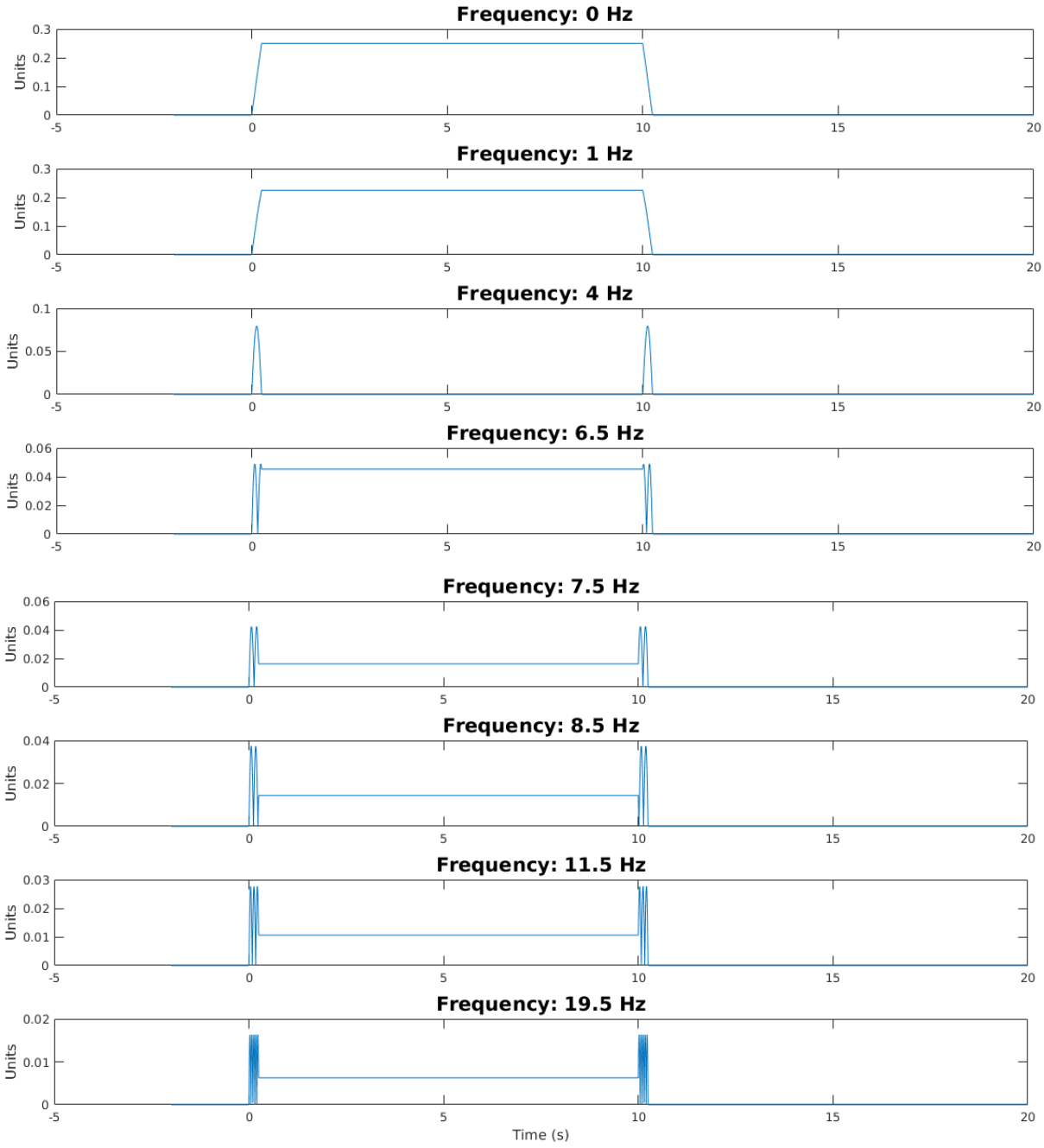

Figure 1: Convolution of the system input with the pulse computed by MATLAB's conv method

The convolution tail was trimmed off to create a window consisting of the responses' transient to steady state output.

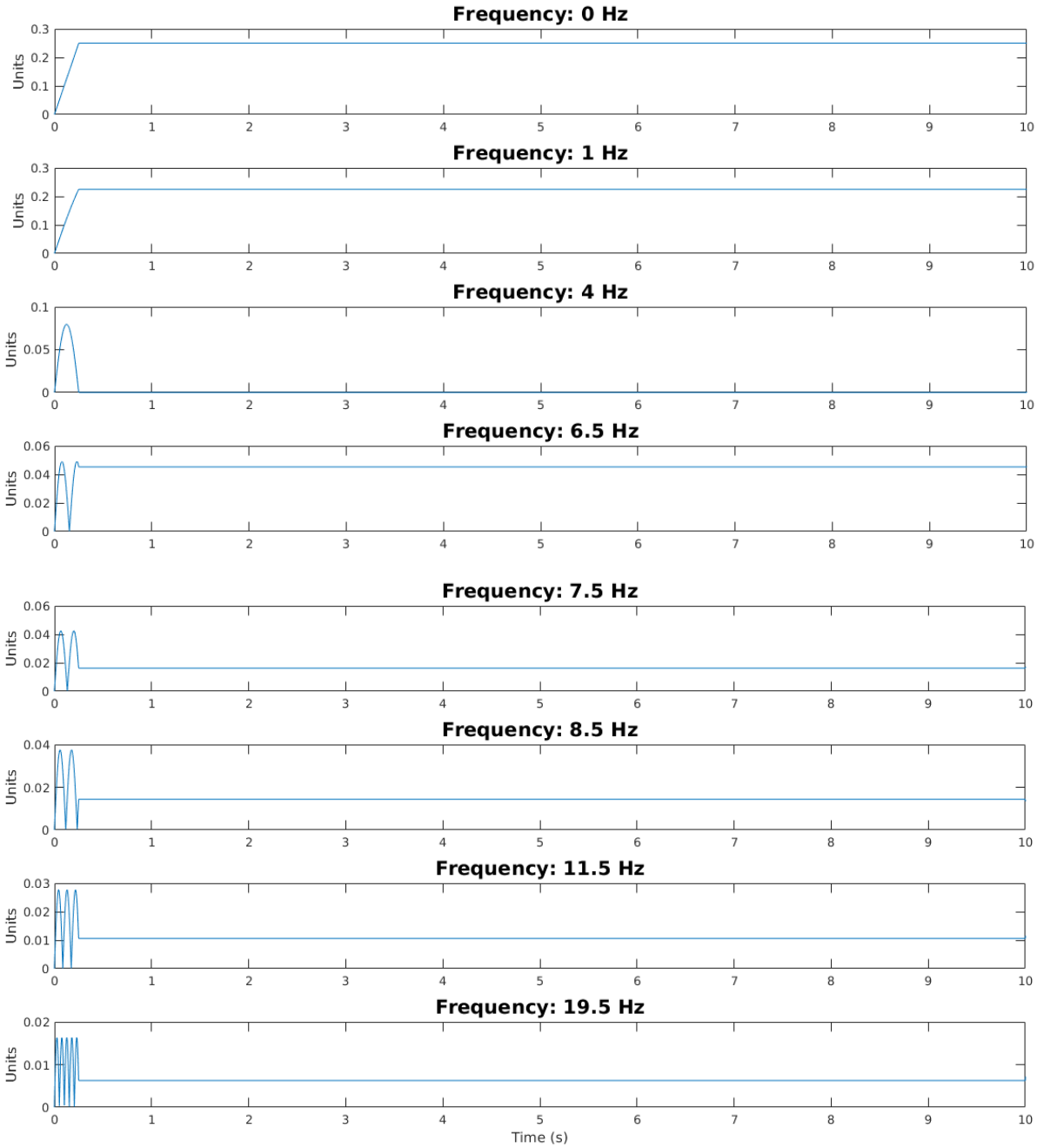

Figure 2: Convolution of the system input with the pulse with the "convolution tail" cut off

The output appears to be a low pass filter, as the amplitude begins to decay and get closer to 0 as the frequency increases.

Only the non-zero values of the pulse array, $h(h \sim = 0)$, were considered and convoluted with the system input to create the following outputs.

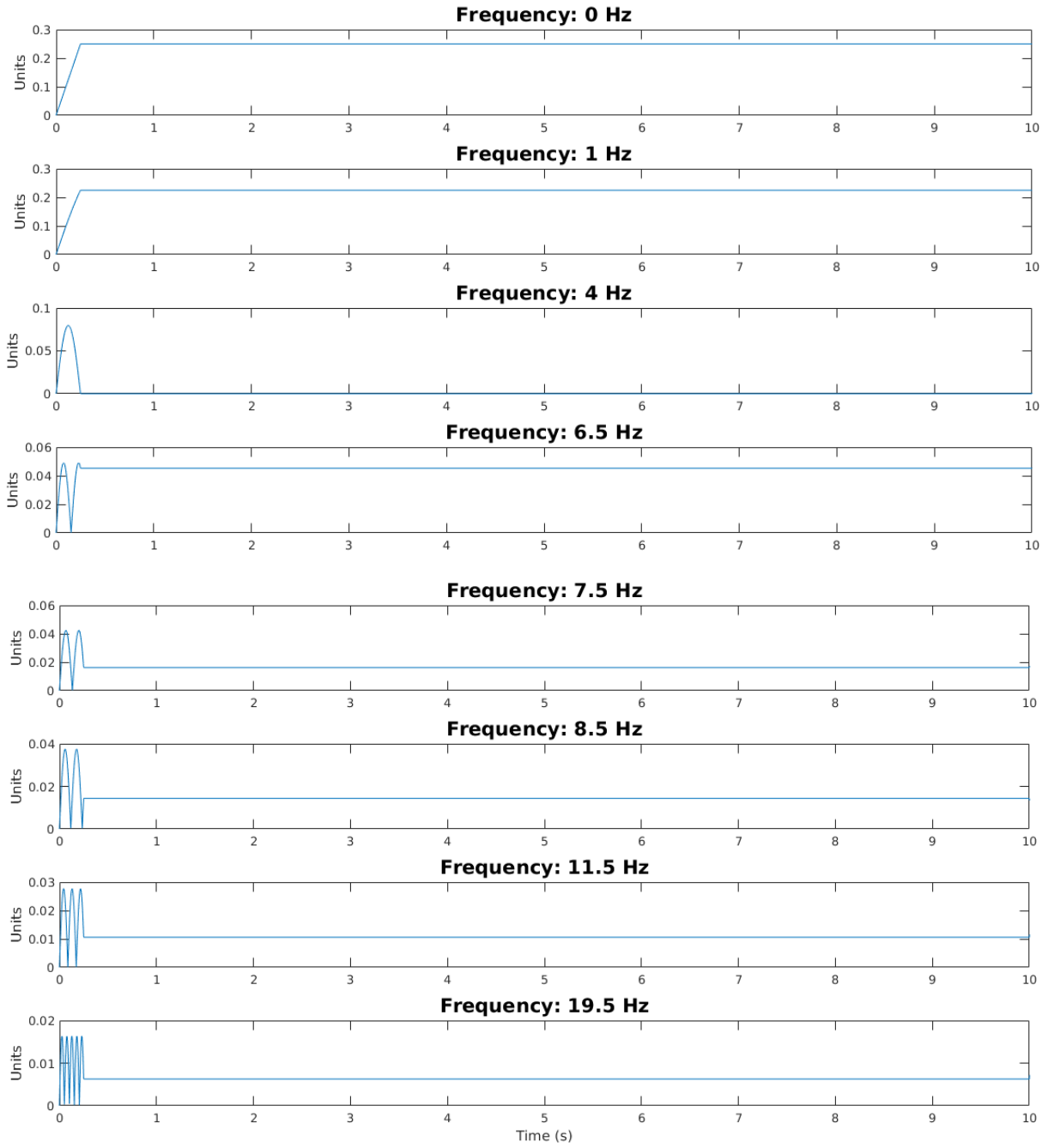

Figure 3: Convolution of the system input with the non-zero values of the pulse

The stripped out input array yielded identical plots because the amplitude and duration of the pulse remained constant.

The new pulse input $h_2(t) = h(t) \times e^{j15\pi t}$ was created and convoluted with the system input $x(t)$ to create the following outputs:

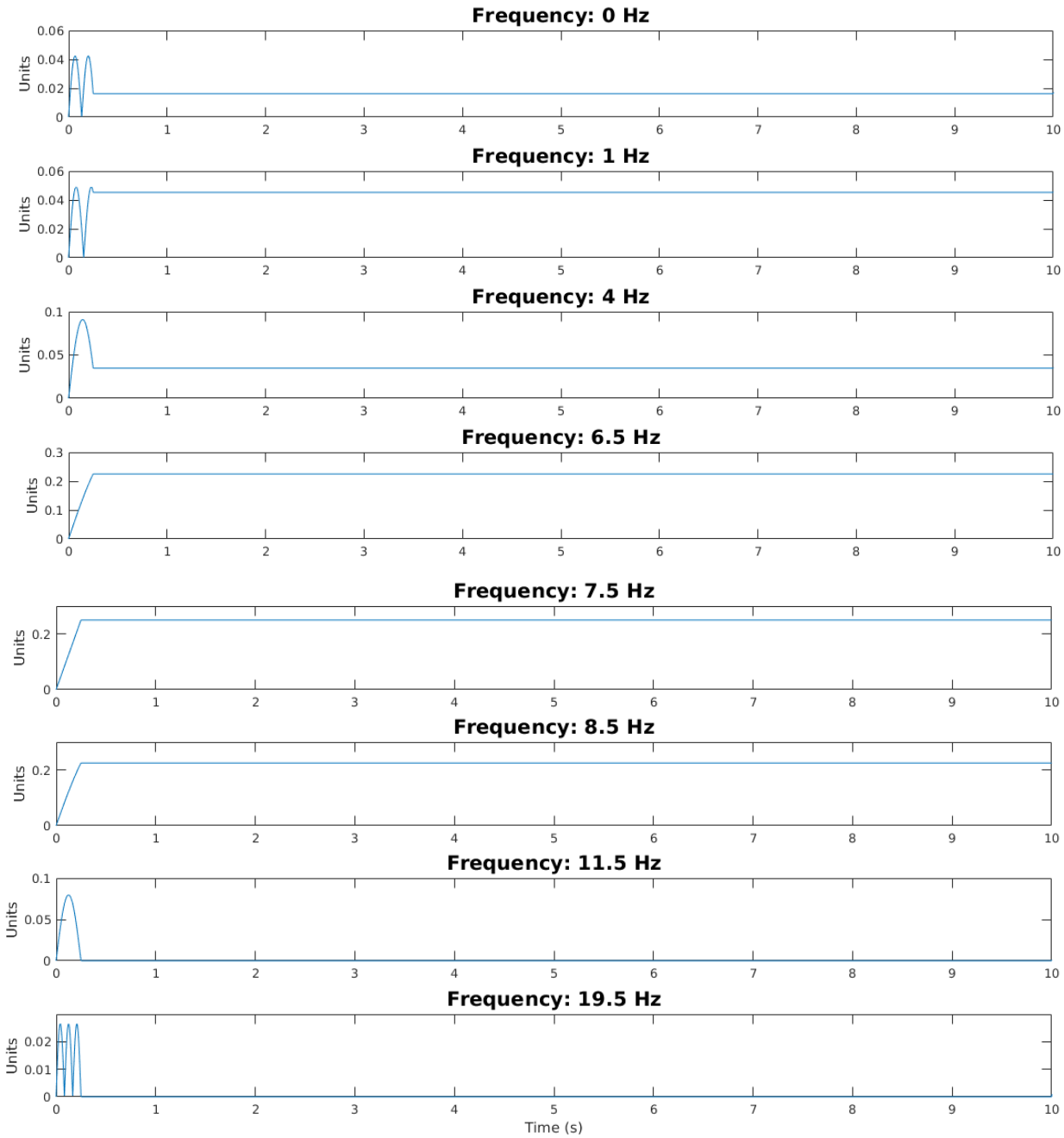

Figure 4: Convolution of the system input with the new pulse

The new pulse creates a low pass filter when convoluted with the complex exponential.

The MATLAB function cexp_response() was created to return a 1 by M array of complex outputs representing the steady state output at each of the M frequencies given. The function (attached as Appendix A) took in the pulse, h, the time array, t, the array of frequencies to iterate over, f, and an option to plot the response. The inputs were convoluted and assigned to a temporary array y. Since y consists of the complex output, the complex magnitude of the convolution was used to pinpoint the steady state value of the response. The transient response ends after the duration of the input pulse, but instead of using multiple array index mapping, all the zeros of the convolution output were dropped and the mode of the array was used as the steady state value. The index of that value was then taken to find the phase angle of the convolution. cexp_response() was tested with $h(t)$ and $h_2(t)$ to generate the following plots:

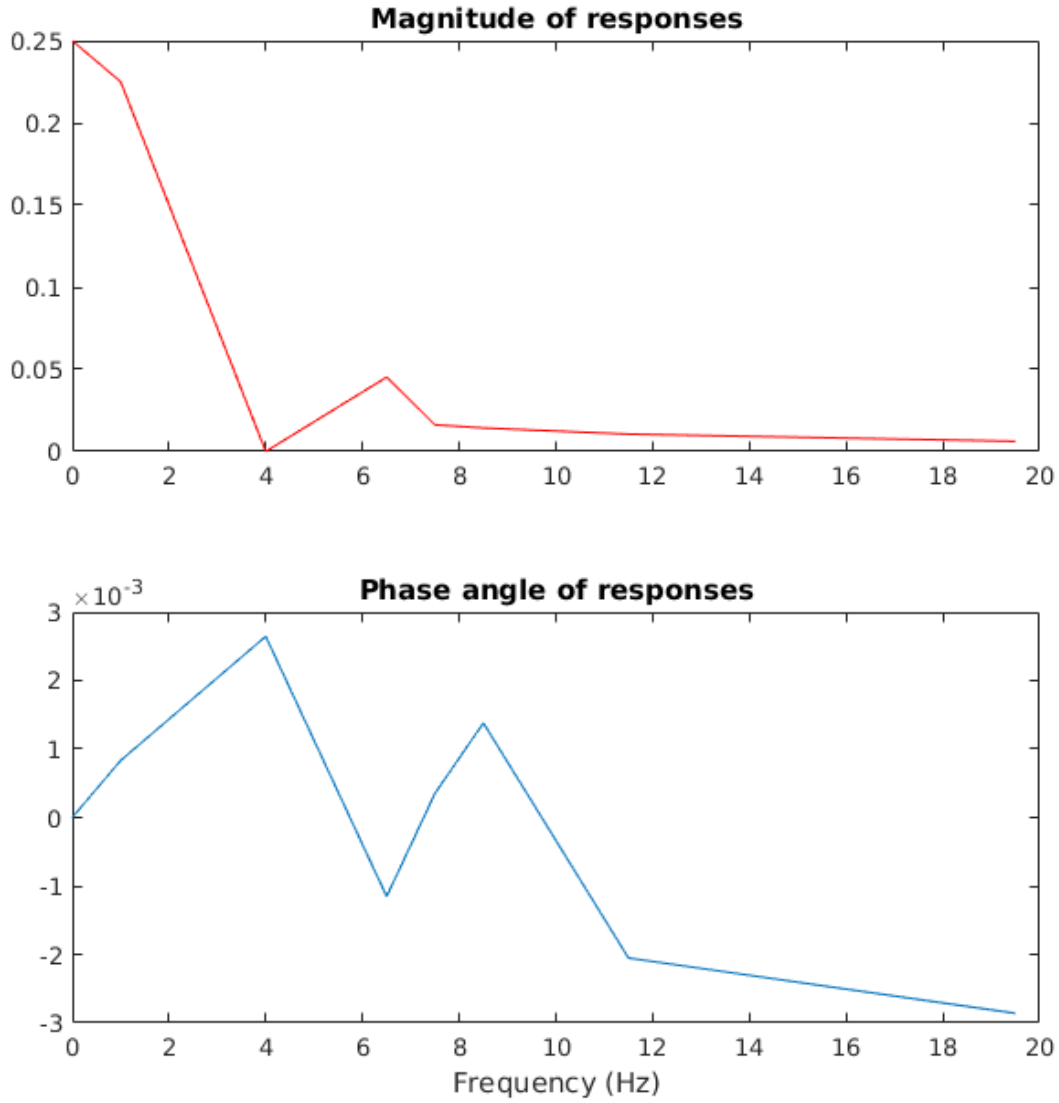

Figure 5: cexp_response() with $h(t)$ and the previous frequencies as the pulse input

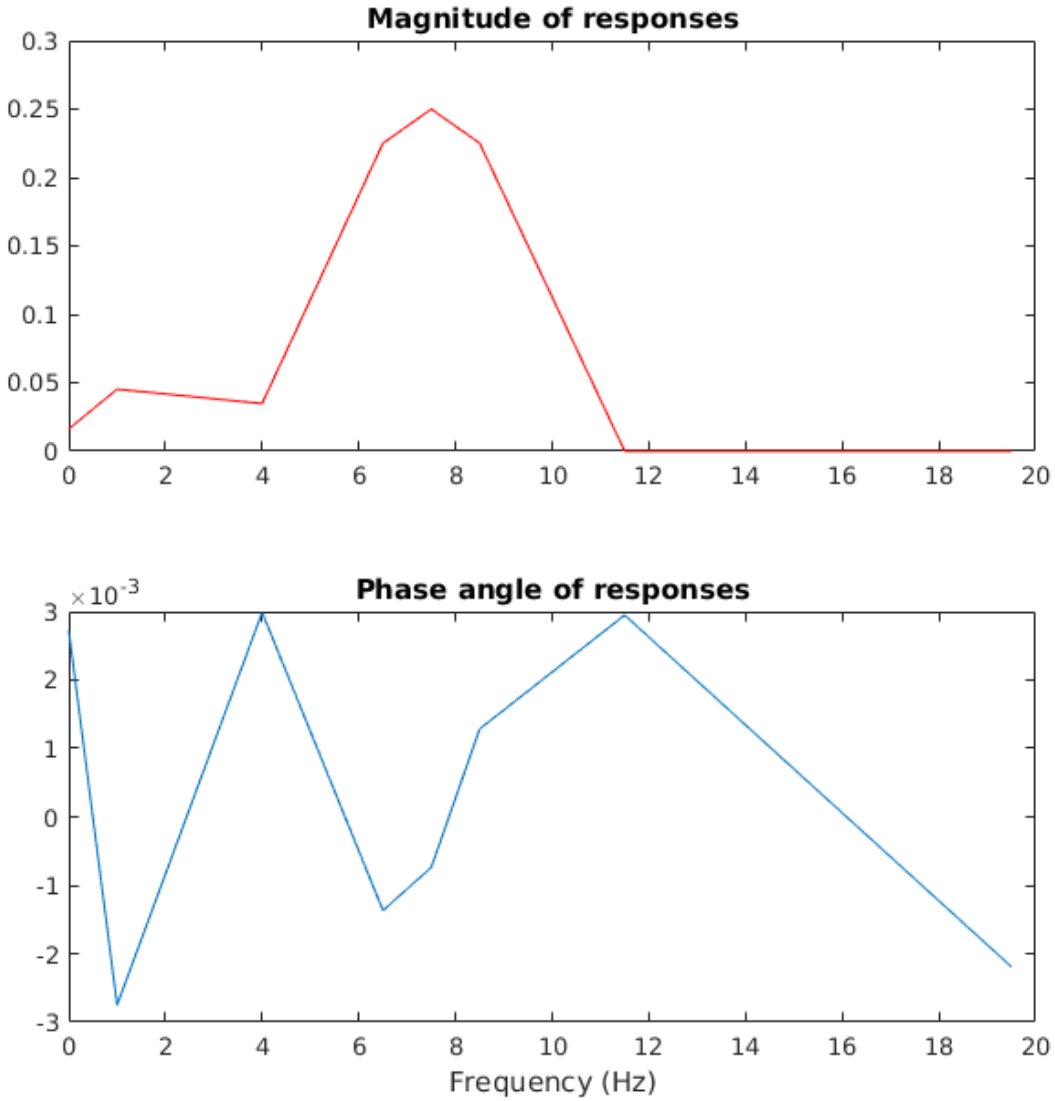

Figure 6: cexp_response() with $h_2(t)$ and the previous frequencies as the pulse input

The input pulses were passed into cexp_response() with the circular frequency array $f = 0{:}\,0.1{:}\,50\ Hz$ and the following plots were generated:

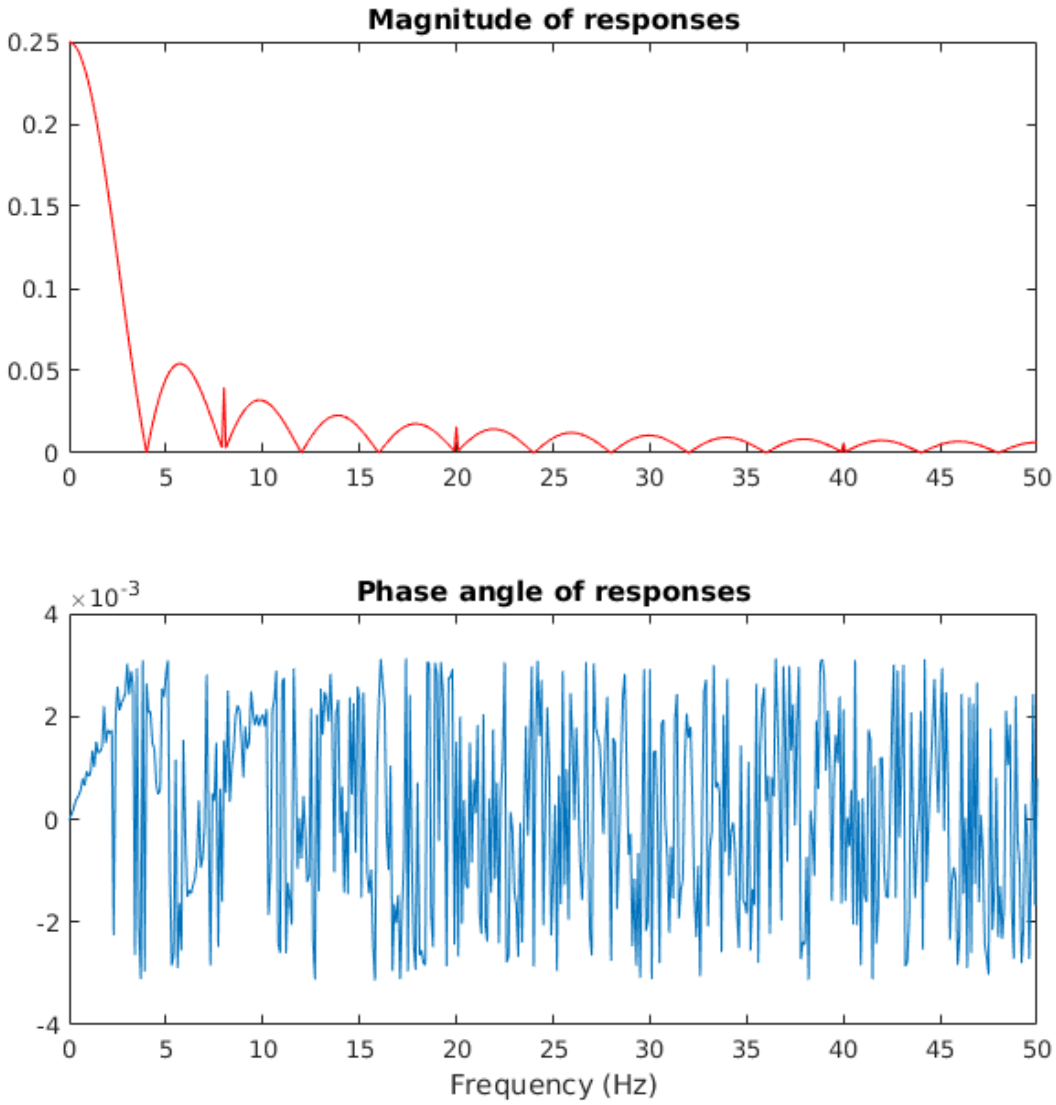

Figure 7: cexp_response() with $h(t)$ and the circular frequency as the pulse input

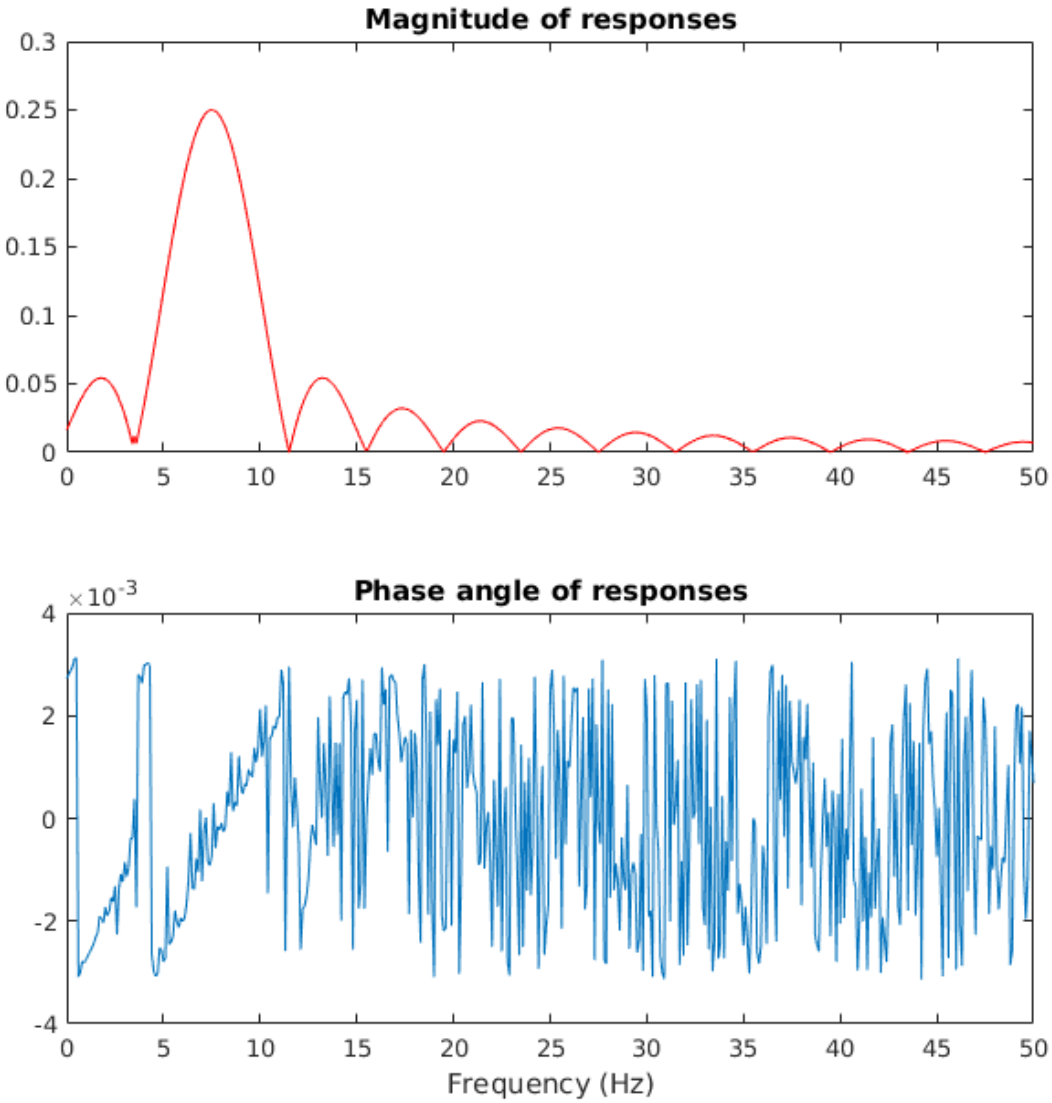

Figure 8: cexp_response() with $h_2(t)$ and the circular frequency as the pulse input

5. Appendices

5.1 Appendix A

Please refer to the attached script as Appendix A. The definition and documentation of the function cexp_response() that was used for parts 3.3 and 3.4 is located there.