CA1 - Convolution

Tuesday, January 11, 2022 12:16 AM

Assignment

Write a report that addresses the following questions. This report includes Matlab plots. When making plots in Matlab or by hand, please keep the following points in mind.

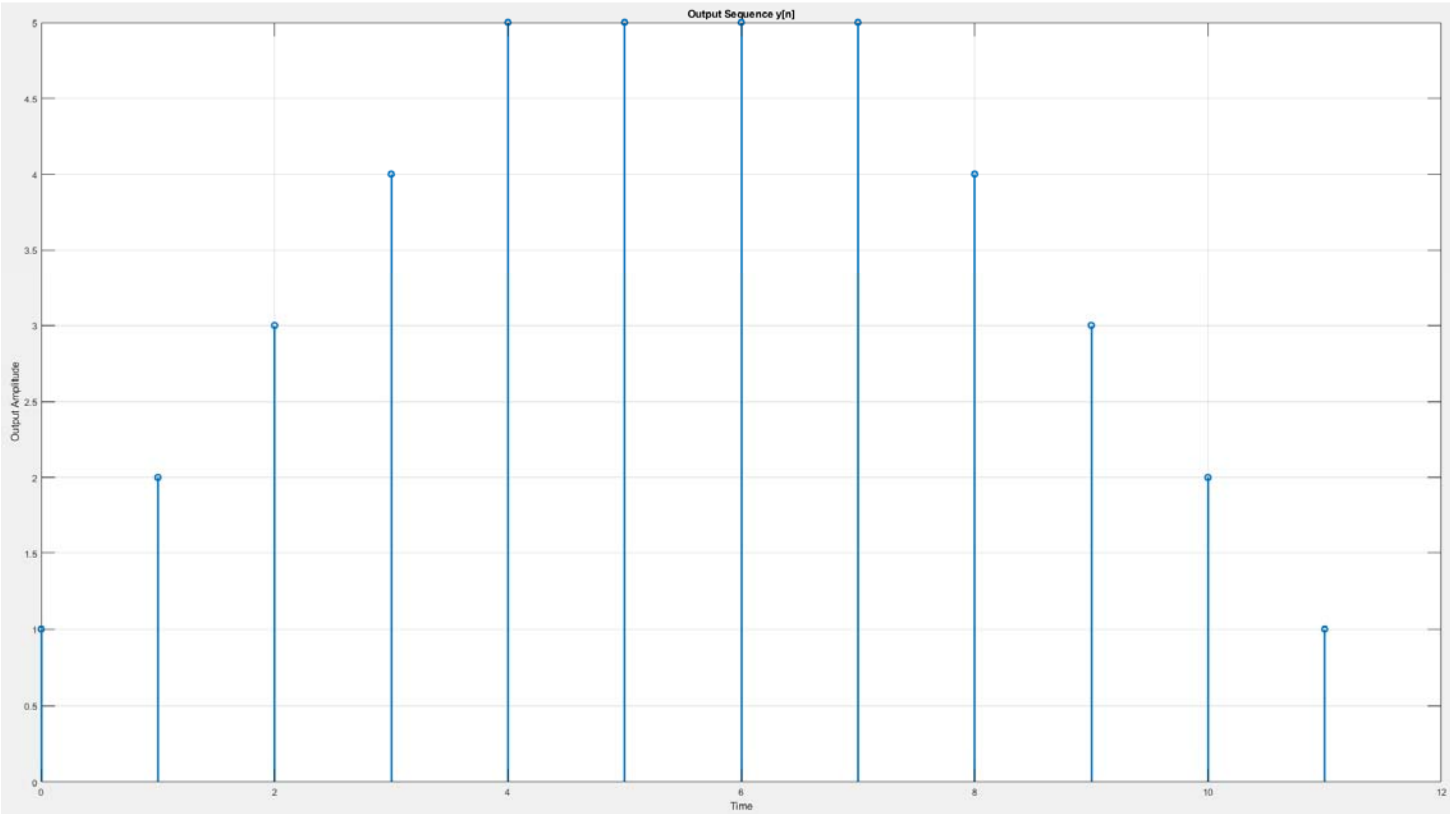
- Axes on plots should be labeled.
- Plots should have a title.
- If there are multiple lines on a plot, include a legend.
- The lines on plots should be easily visible (thick enough).
- Intensity images should include a colorbar.

Here are the report questions (five points for each question).

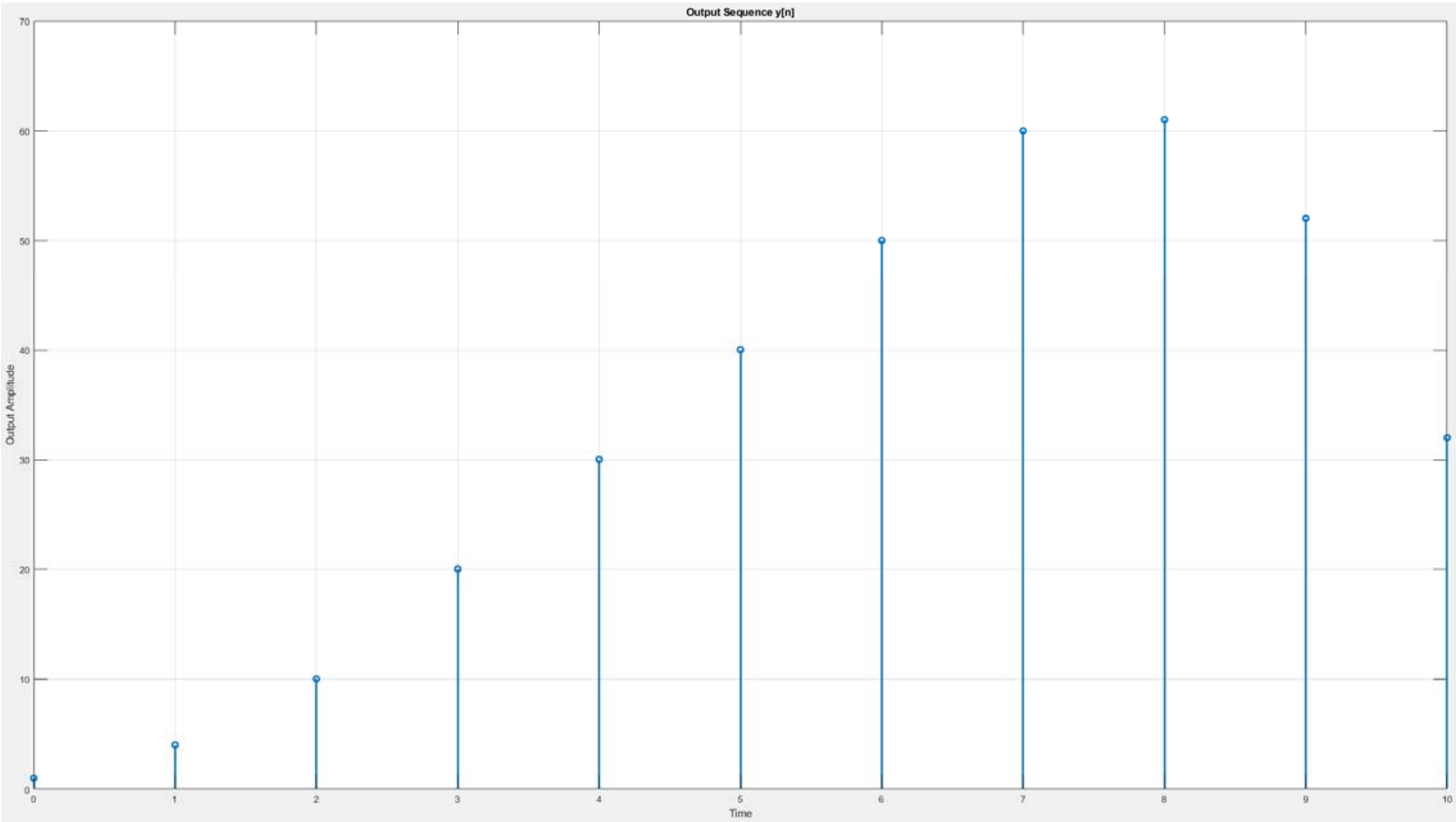
1. What are the array indexes you used for xzp(...) and h(...) in the `myconv` code?
2. Include a stem plot showing the convolution of the following sequences: x = [1, 2, 3, 4, 5, 6, 7, 8] and h = [1 2 3 4]. Also list the values $y[n]$ in the convolution result.
3. Plot the impulse response of the filter used in the audio example.
4. Plot the magnitude response of the filter used in the audio example.
5. Use the magnitude response plot to explain why the filter in the audio example is considered to be a low-pass filter. Describe the features of the passband and the stop band of this filter. What frequencies does this filter "pass" and what frequencies does this filter "stop"?
6. What would the response of a high pass filter look like? Make a sketch.
7. Plot spectrograms of the audio signals at the filter input and output (2 spectrogram plots).
8. Use the spectrograms to describe the action of this filter on a signal.
9. Plot the filter input and output signals on the same axis (time-domain plot).
10. There are times when the input signal is large and the output signal is essentially zero. See for example the intervals 2-3 seconds and 7-8 seconds in the time-domain plot. Use the filter magnitude response and the spectrograms to explain what is happening in these intervals. Why is the output signal zero there?
11. Attach your code.

The indexes used were 'L + (n - k)' for the zero padded x function, and 'k' for the h function. This was done to effectively 'flip' the x function by subtracting the increasing counter k from the overall time frame to iterate from the back of x toward the front while iterating from the front of h toward the back.

Convolution of the 1's vectors



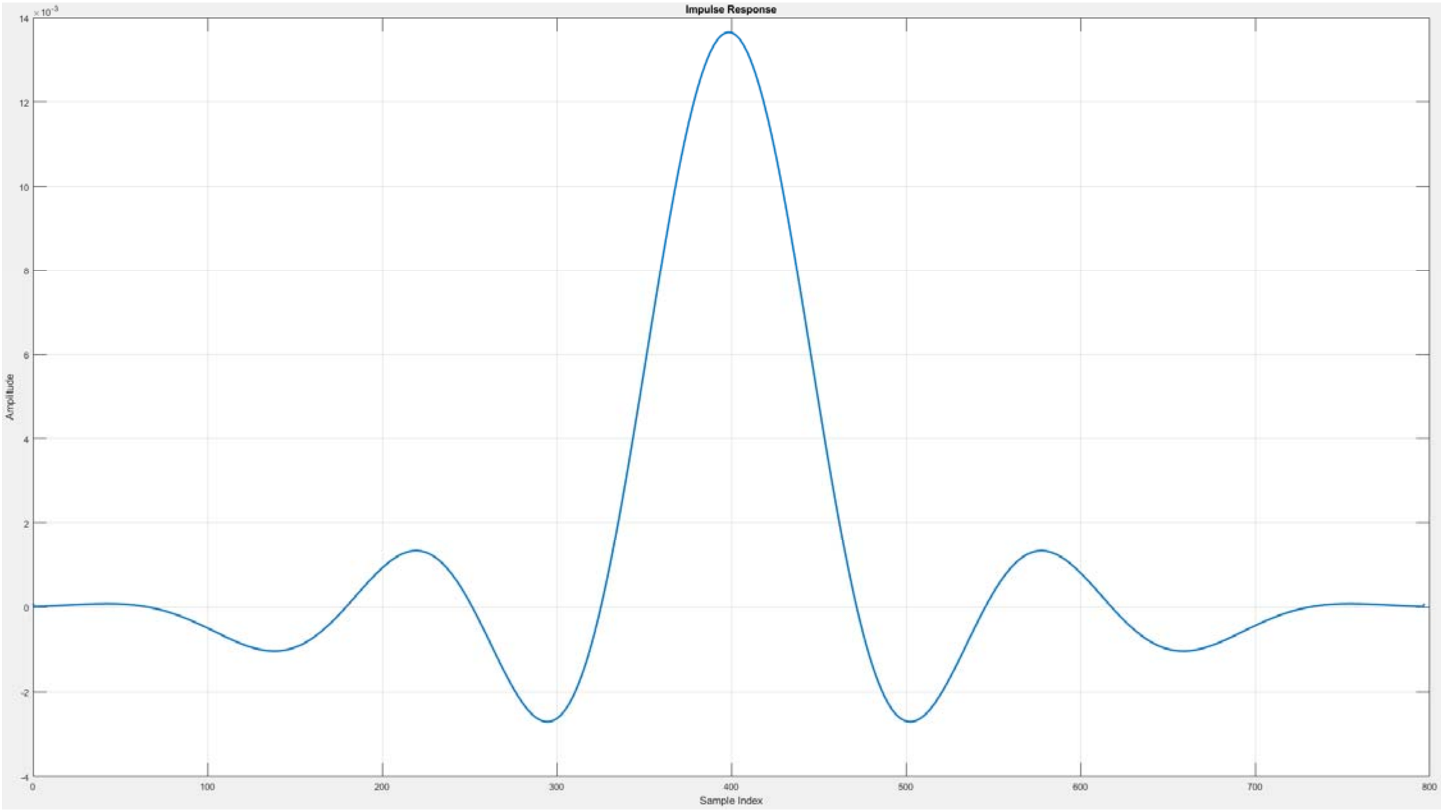
Question 2 stem plot.



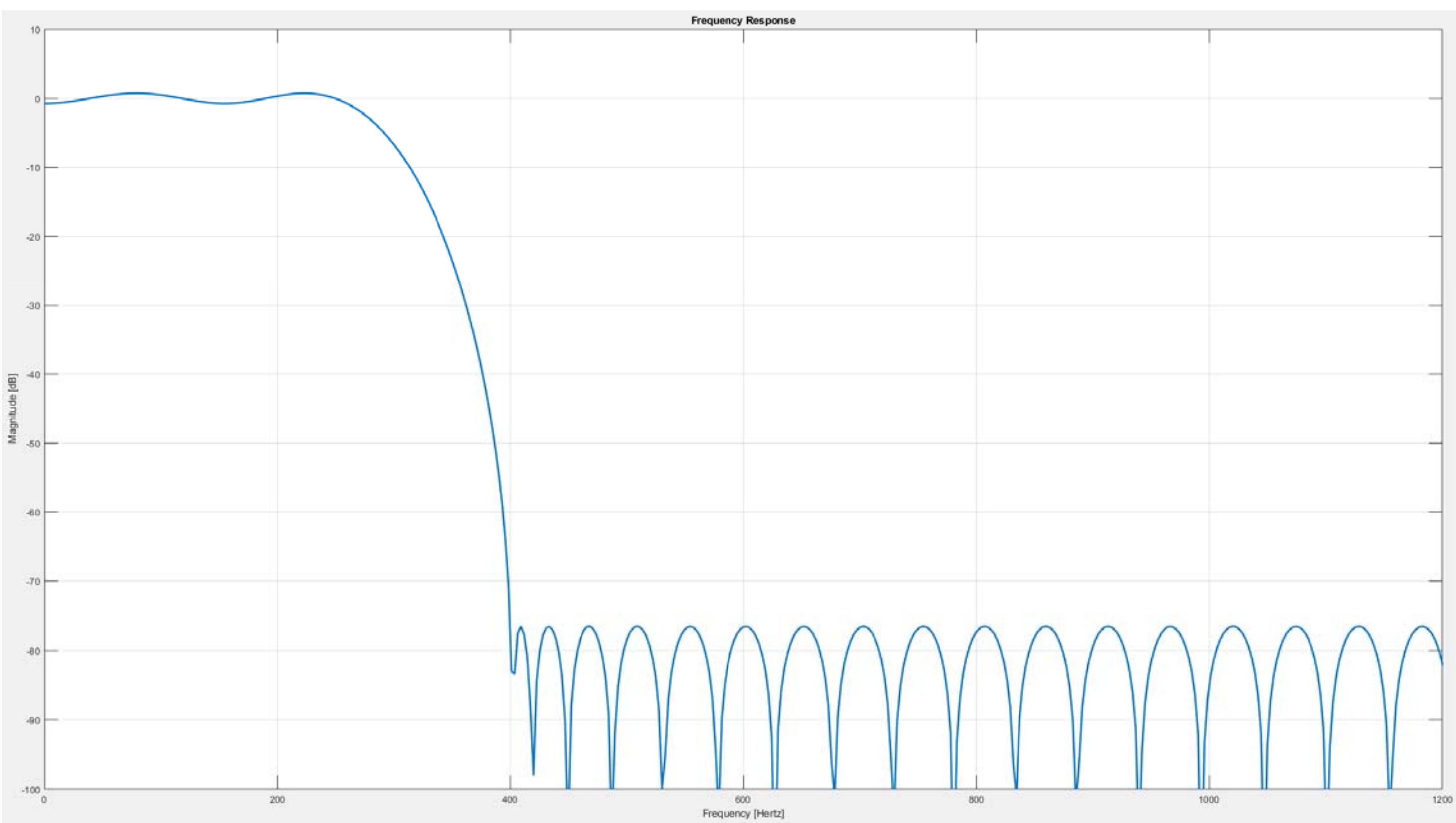
Values:

1
4
10
20
30
40
50
60
61
52
32

Impulse response:

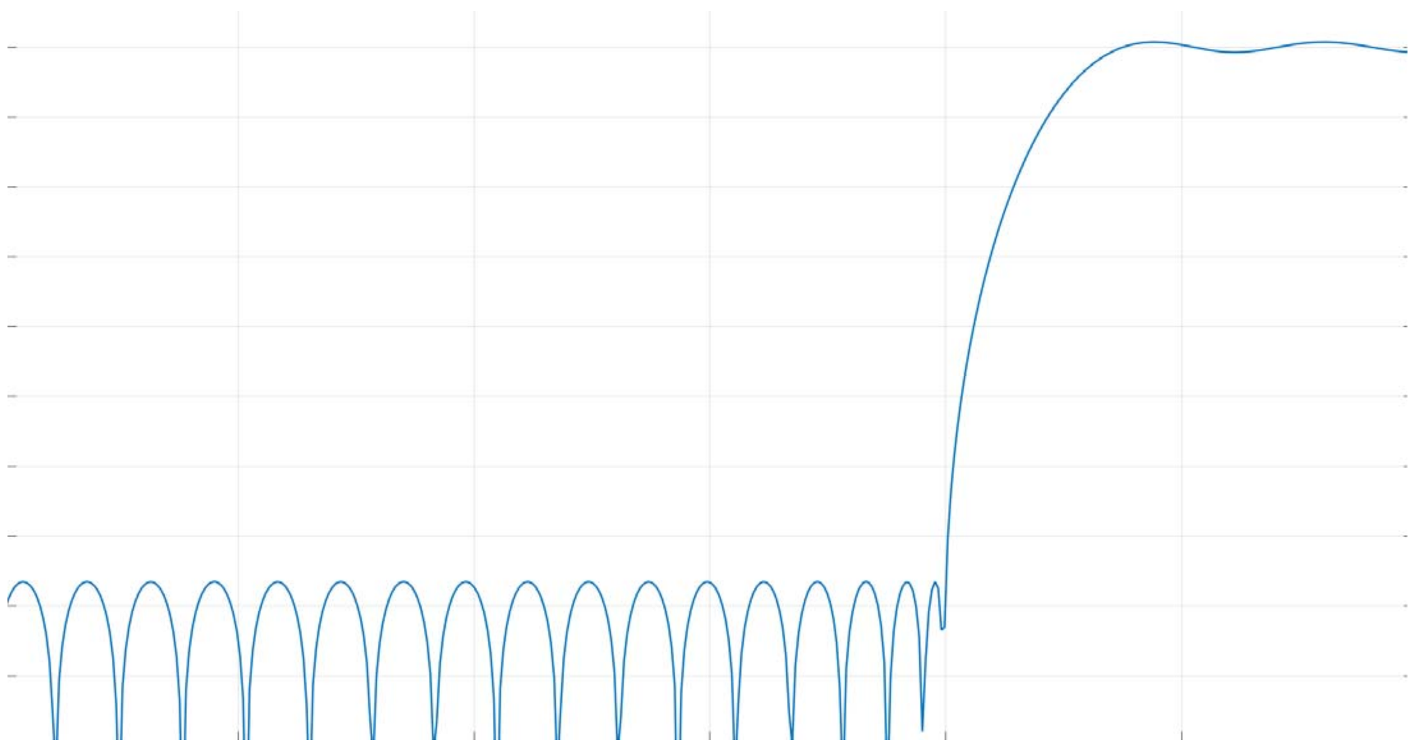


Magnitude Response:

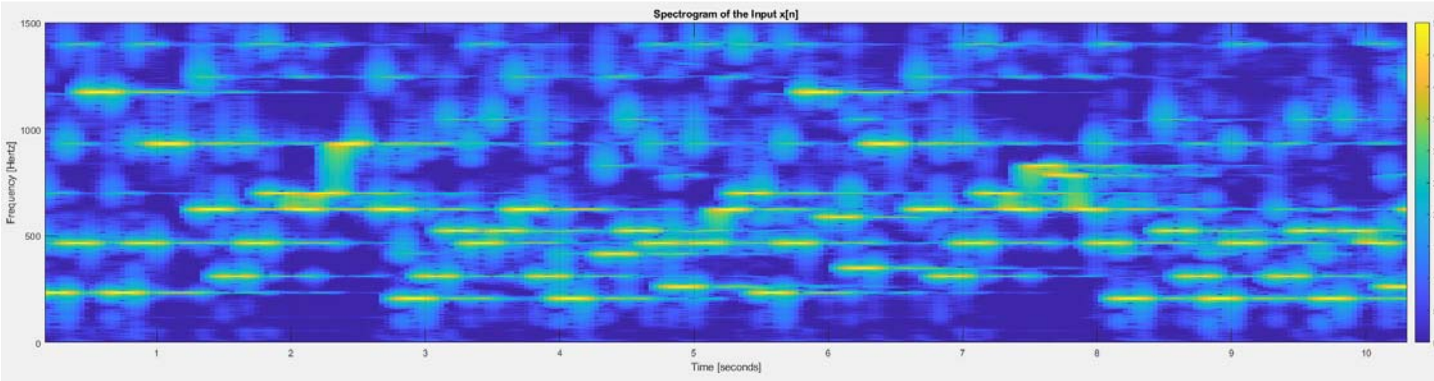


This can be considered a low-pass filter due to the lower frequencies having little to no attenuation, and the higher frequencies having large attenuation. The passband is in the very low frequencies, around 300 Hz or 400 Hz and below and has very little attenuation other than the slight ripples observed toward the top left of the graph. The stop band is anywhere from around 300 Hz or 400 Hz and above, and has very strong attenuation, decreasing the levels to almost -80 dB. If you consider -3dB to be the cutoff point, then frequencies higher than around 280 Hz are in the stop band.

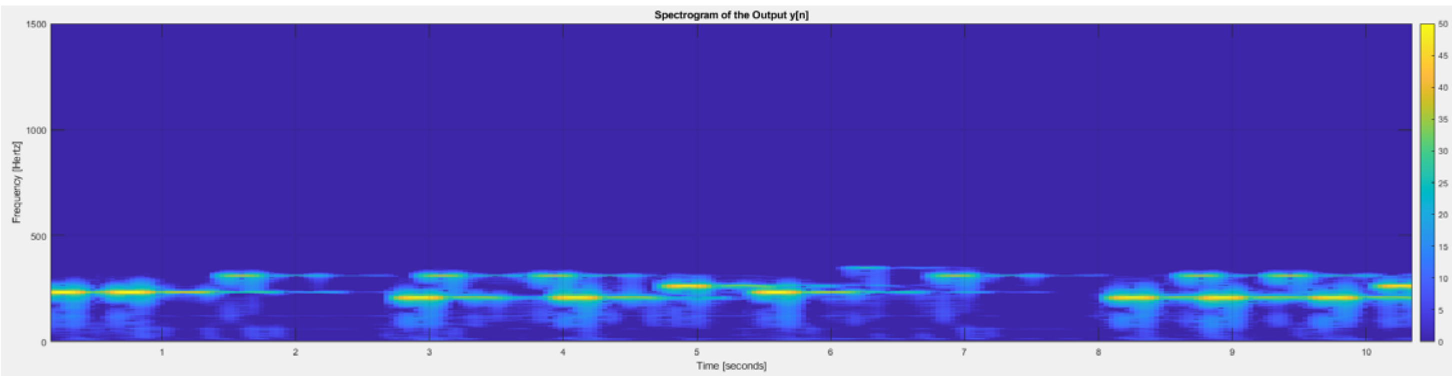
The response of a high pass filter would look like a mirror image of the low pass filter's response. (see below).



Spectrogram of the input

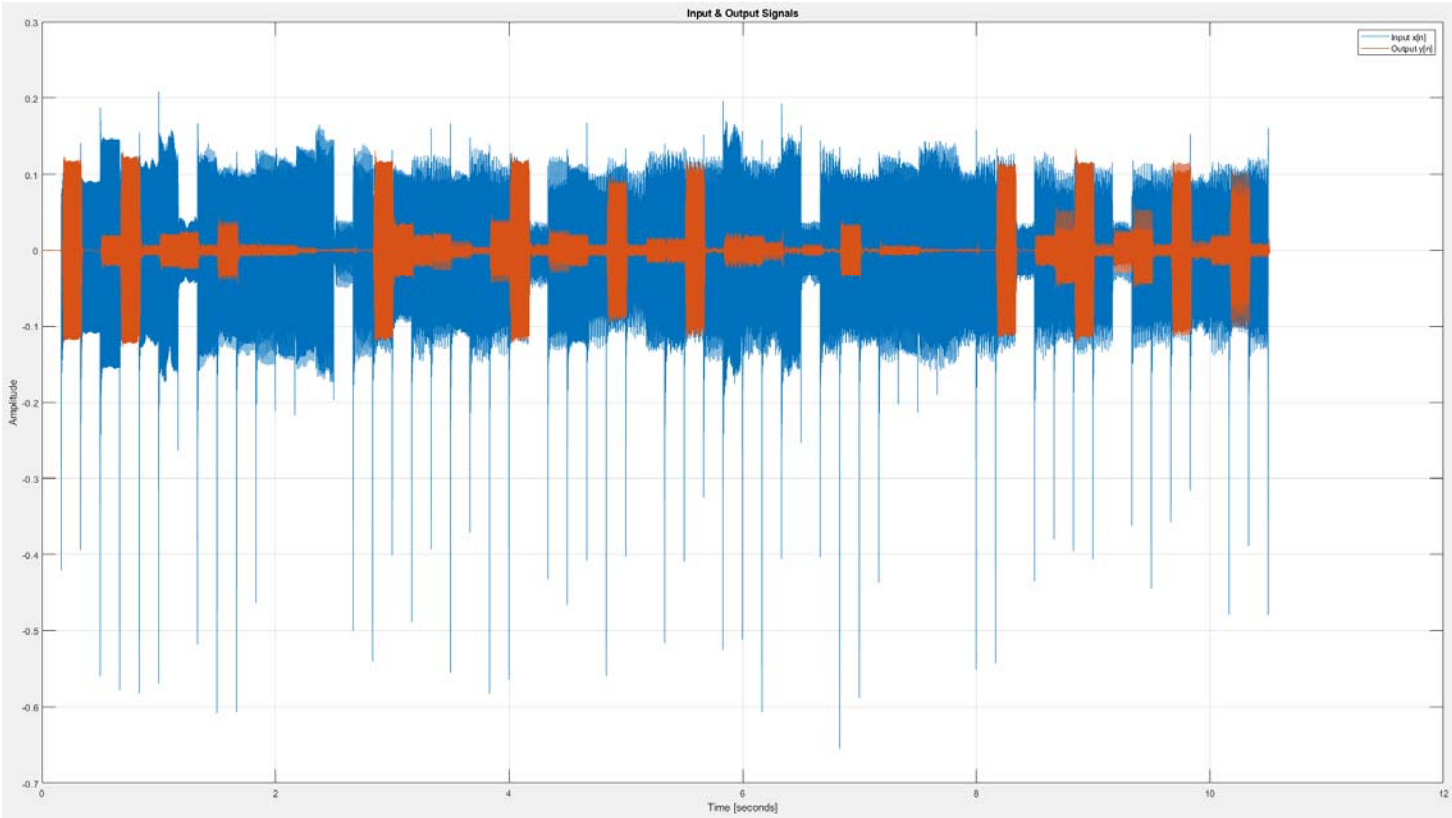


Spectrogram of the output



As can be seen in the spectrograms above, the filter effectively blocked all signals above around 400 Hz from being output.

Input and Output Signals



In some portions of the output, it can be seen that there is a lot of input information, and essentially no output data. This can be explained by looking at the previously discussed magnitude and spectrogram plots that show all high frequency signals are blocked. Since all high frequency signals were blocked from being output, all parts of the input that had mainly high frequency data were blocked, leaving only low frequency output signals. In a song like the one used, there are some portions that had almost exclusively high notes being played, which were then blocked by the filter from being output, ultimately resulting in little total output.

Code used:

```
% Input signal
N = 8;
x = [1:1:8];

% Impulse response
L = 4;
h = [1:1:4];

% Convolve - call custom convolution function
y = myconv(x,h);


% Visualize
M = N+L-1; % Compute length of output
n = [0:M-1]; % Construct a time sequence
stem(n,y,'LineWidth',2); % Plot
xlabel('Time');
ylabel('Output Amplitude');
title('Output Sequence y[n]');
grid on;
shg; % Show handle graphics


function y = myconv(x,h)
N = length(x); % Length of x[n]
L = length(h); % Length of h[n]

K = N+2*(L-1); % Length of zero-added x[n]
xzp = zeros(K,1); % Allocate memory
xzp(L:L+N-1) = x; % Zero-pad x[n] on both sides

M = N+L-1; % Length of output sequence
y = zeros(M,1); % Allocate memory
for n=1:M % Loop over output times
```

```
    for k=1:L % Multiply-accumulate loop
        y(n) = y(n) + xzp(L+(n-k))*h(k); % <= Array indexes?
    end
end
end
```