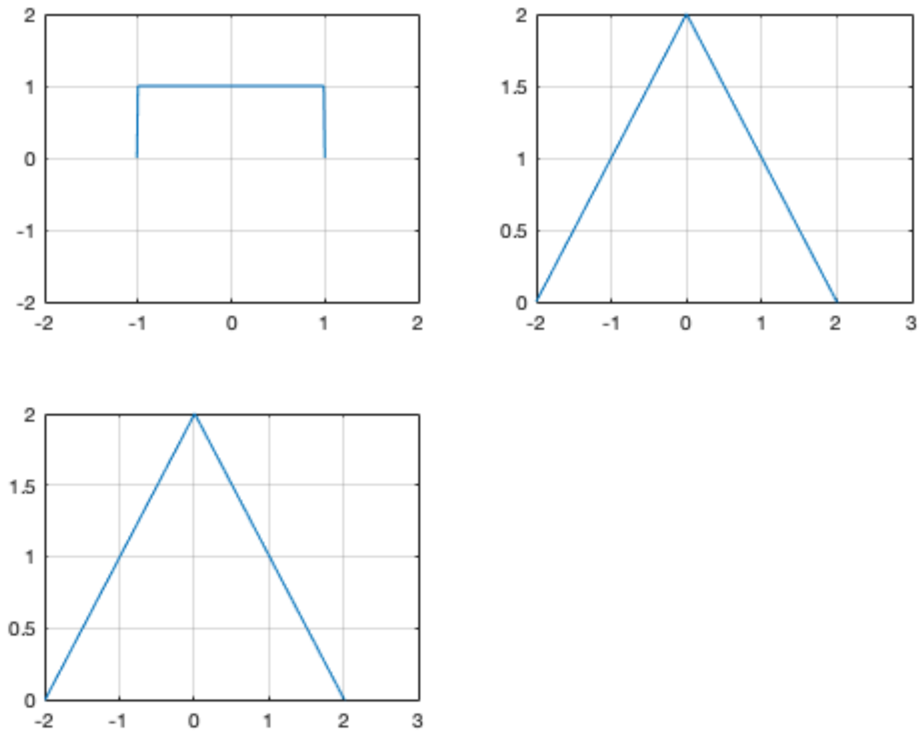
BME 306 Lab 3 - Convolution in Matlab

Table of Contents

Alexander Ross 10/14/19

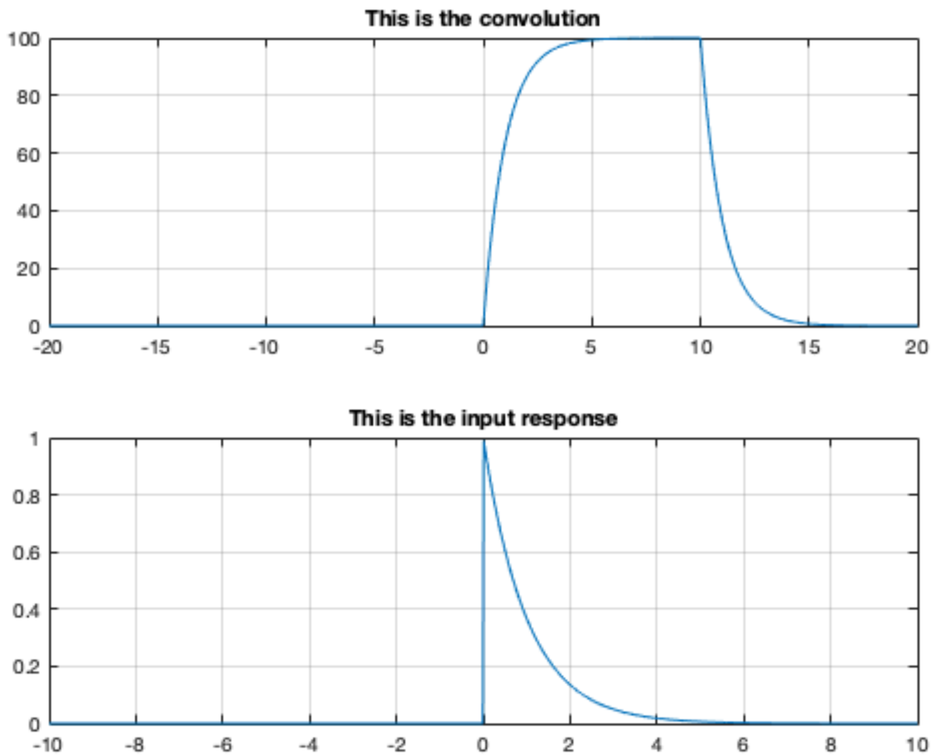
Question 1

```
% Analytical Convolution:
t2 = -2:0.01:2.02;
convolution = 2*tripuls(t2,4);

t = -1.01:0.01:1;
t2 = -2:0.01:2.02;
s = rectpuls(t,2);
subplot(2,2,1)
plot(t,s);
axis([-2 2 -2 2])
grid on
hold on
subplot(2,2,2)
plot(t2,convolution)
grid on
subplot(2,2,3)
plot(t2,conv(s,s)*0.01);
grid on;
```



Question 2

```
t = -10:0.01:10;
s = heaviside(t);
s2 = exp(-t).*s;
t2 = -20:0.01:20;
subplot(2,1,1)
plot(t2,conv(s,s2));
title('This is the convolution');
grid on;
hold on
subplot(2,1,2)
plot(t,s2)
title('This is the input response');
grid on
```



Question 3

```
t = -100:0.001:100;
s = heaviside(t);
s2 = exp(-t).*s;
t2 = -200:0.001:200;
subplot(2,1,1)
plot(t2,conv(s,s2));
title('This is the convolution');
grid on;
hold on
subplot(2,1,2)
plot(t,s2)
title('This is the input response');
grid on

% Note: See Attached Plot

% Explanation: The result differs in that the MatLab code only
 provides an
% approximation to the result found in class.  The solution found in
 class
% mathematically is continuous, and spans all of time, while the
 MatLab
```

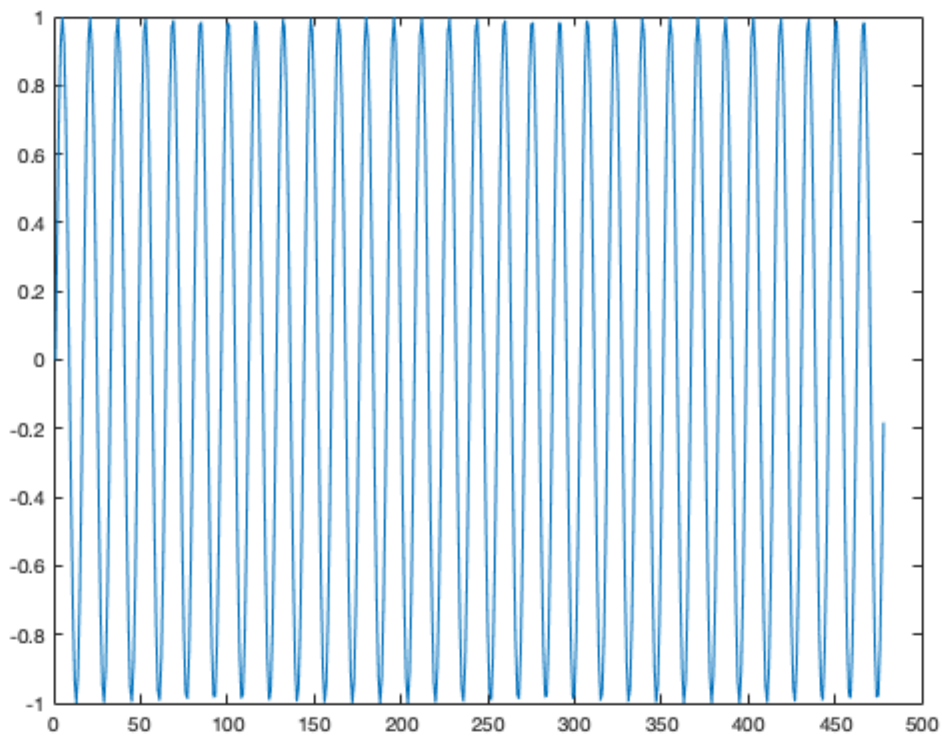
```
% solution is a descrete solution over a given time interval. Thus
 after a
% certain numerical value, the graph ceases to provide a accurate
% depictrion of the function.  It is possible to make the function
 look more
% like the class result by increasing the length of the time vector.
```

Question 4

```
f = 10;
% Freqency in hertz = 10
% Frequency in rad/s = 62.83
% Time constant (in microseconds[uSec]) = 7958
```

Question 5

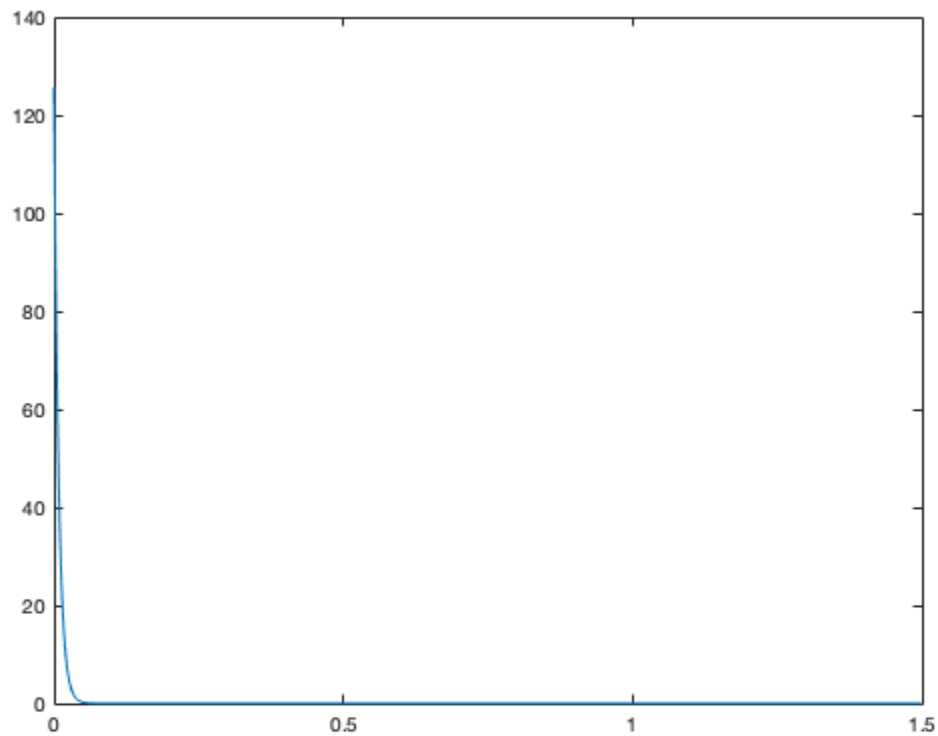
```
t = 0:pi/2000:0.75;
inputsignal = sin(2*pi*40*t);
plot(inputsignal);
```



Question 6

```
t2 = 0:pi/2000:1.5;
tc = 0.007958;
```

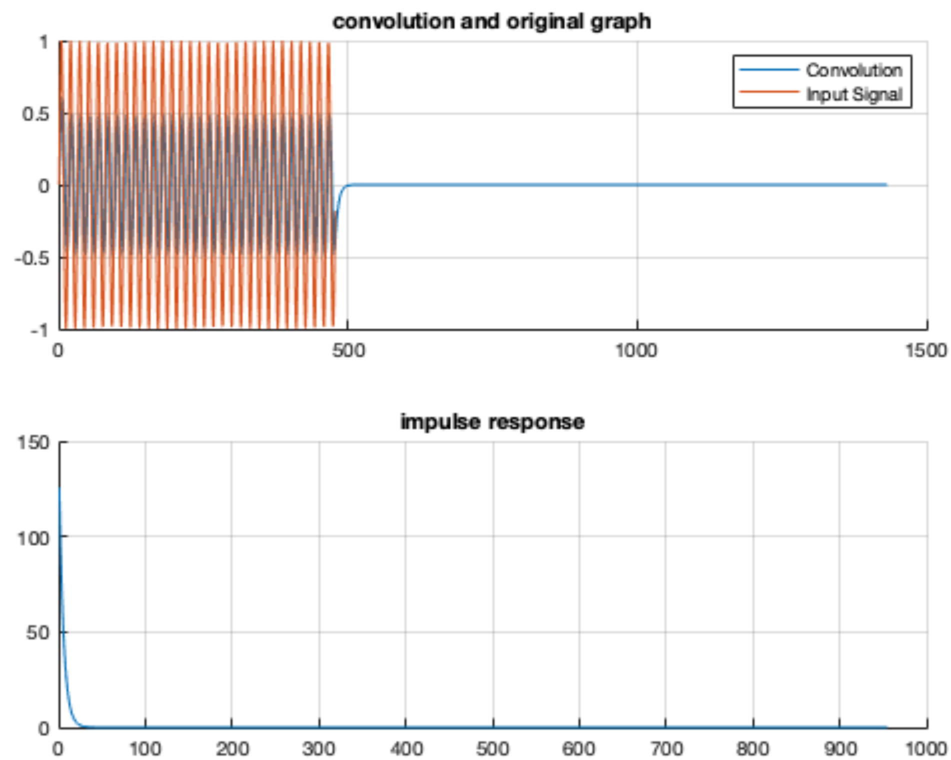
```
impulseresponse = 1/tc*exp(-t2/tc);
plot(t2,impulseresponse);

% Previous Code:
% transfer = tf([1],[1/(40*2*pi) 1]);
% impulseresponse = lsim(transfer,inputsignal,t);
% plot(impulseresponse);
```



Question 7

```
t = 0:pi/2000:0.75;
impulseresponse = 1/tc*exp(-t2/tc);
inputsignal = sin(2*pi*40*t);
convolution = (pi/2000).*conv(impulseresponse,inputsignal);

% Previous Code:
% convolution = conv(inputsignal,impulseresponse,'same');

subplot(2,1,1)
hold on
legend();
plot(convolution,'DisplayName','Convolution');
hold on
legend();
plot(inputsignal,'DisplayName','Input Signal');
title('convolution and original graph');
```

```
grid on
hold on
subplot(2,1,2)
title('impulse response');
hold on
plot(impulseresponse);
grid on

% Note: See attached plot
```



Question 8

```
% The output signal looks like an approximation of the real signal.
 It is
% not exact to the continous-time solution, but it is a good
 approximation
% of the real signal. However, it is only valid for a given time
 interval,
% and does not span all of time.
```

Question 9

```
% Ratio of the magnitude: 0.707 (or simply 1/sqrt(2))

% Explanation: The output to input ratio on the graph is 0.44.  Using
```

```
% phasors -- Y(jw)/X(jw) = 1/(251*(0.003988j+1)) = -0.5j+0.5.
 Therefor,
% the phasor magnitude is (0.5^2 + 0.5^2)^1/2 = 0.707 = 1/sqrt(2)
```

Question 10

```
% See attached EvelopeDetector.m
```

Question 11

```
% Recommended Cutoff Frequency: 200Hz
```

Question 12

```
% See attached EvelopeDetector.m
```

Question 13

```
% See attached plot
```

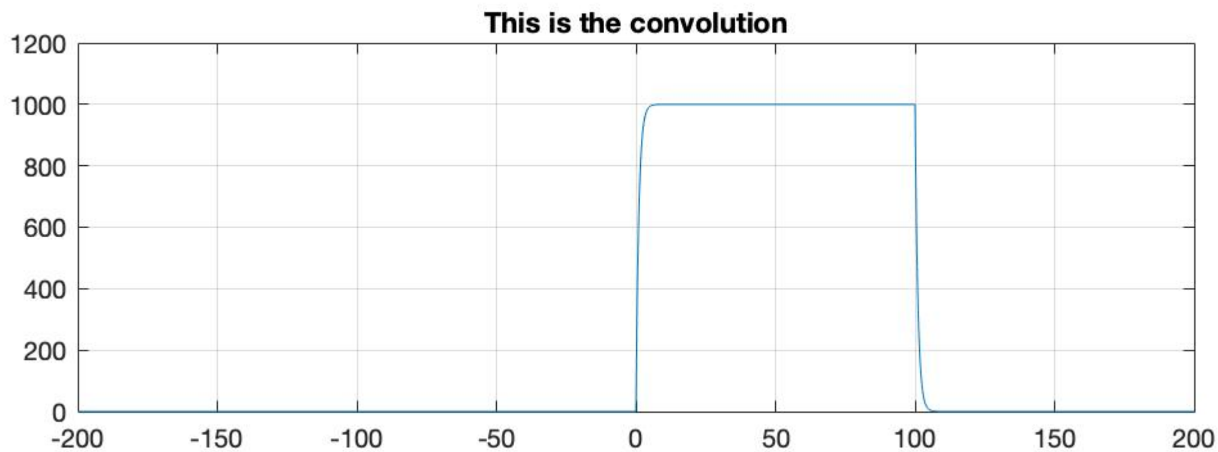
Question 14

```
% See attached plot
```

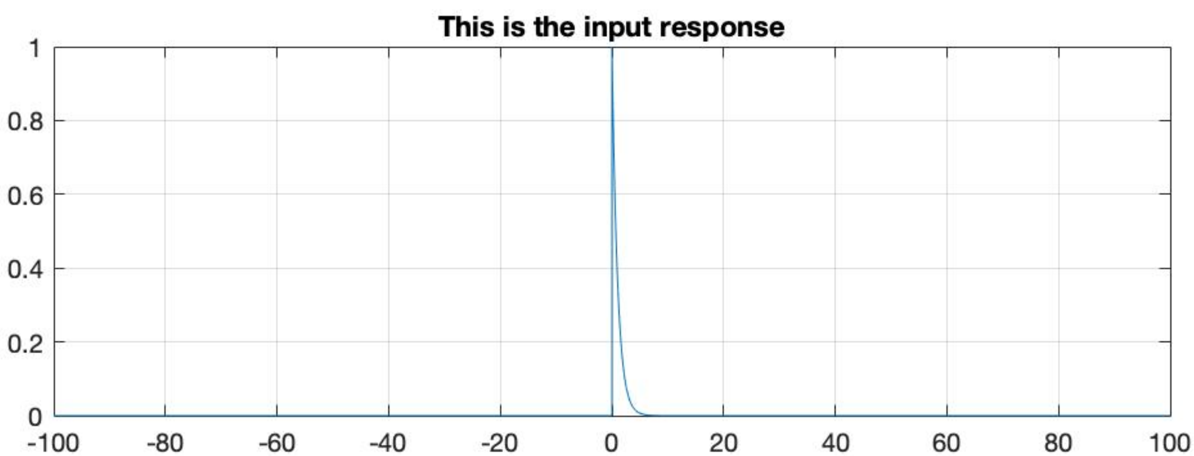
Published with MATLAB® R2019a

```matlab
function [output] = EnvelopeDetector(samplingfrequency)

myVoice = audiorecorder;

pause(2);

% Define callbacks to show when
% recording starts and completes.
myVoice.StartFcn = 'disp(''Start speaking.'')';
recordblocking(myVoice,5);
myVoice.StopFcn = 'disp(''End of recording.'')';

doubleArray = getaudiodata(myVoice);
subplot(2,1,2);
plot(abs(doubleArray));
title('Audio Signal (rectified)');
subplot(2,1,1)
plot(doubleArray)
title('Audio Signal (double)');

impulseresponsefinder(8000,doubleArray);

    function [impulseresponse] = impulseresponsefinder(sampf,doubleArray)

fc = 200;
t = 0:(1/(sampf)):5;
inputsignal = sin(2*pi*80*t);
inputsignal2 = abs(doubleArray);
t2 = 0:(1/(sampf)):2.5;
tc = 0.007958;
impulseresponse = 1/tc*exp(-t2/tc);
convolution1 = (1/sampf).*conv(impulseresponse,inputsignal);
convolution2 = (1/sampf).*conv(impulseresponse,inputsignal2);

figure();
hold on
plot(inputsignal);
hold on
plot(convolution1);
title('Orange = Envelope; Blue = Original');
hold off

figure();
subplot(2,1,1)
hold on
plot(convolution2);
title('Envelope')
hold on
subplot(2,1,2)
title('Original');
hold on
plot(inputsignal2);

    end

end
```

Question 3

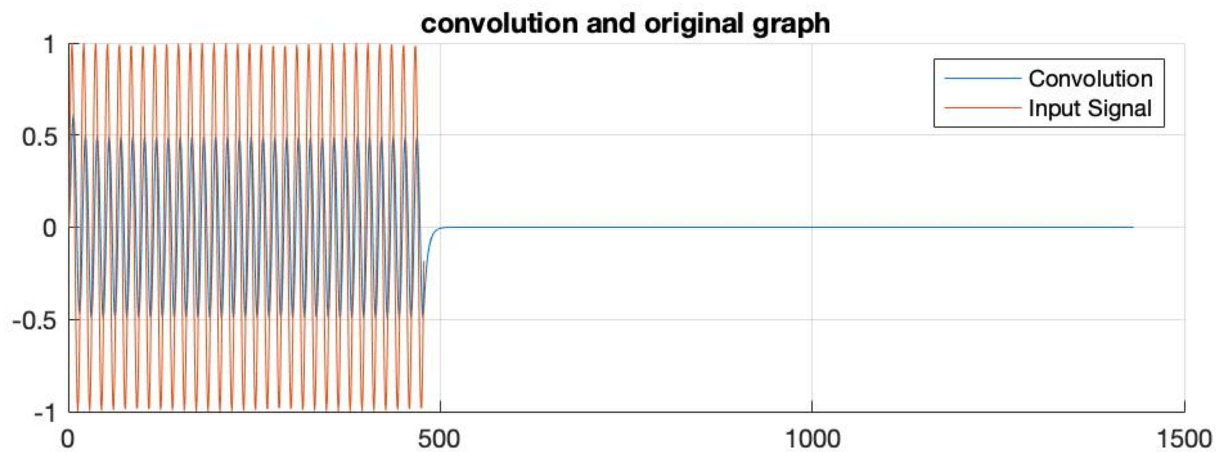


This is the convolution
1200
1000
800
600
400
200
0
-200
-150
-100
-50
0
50
100
150
200

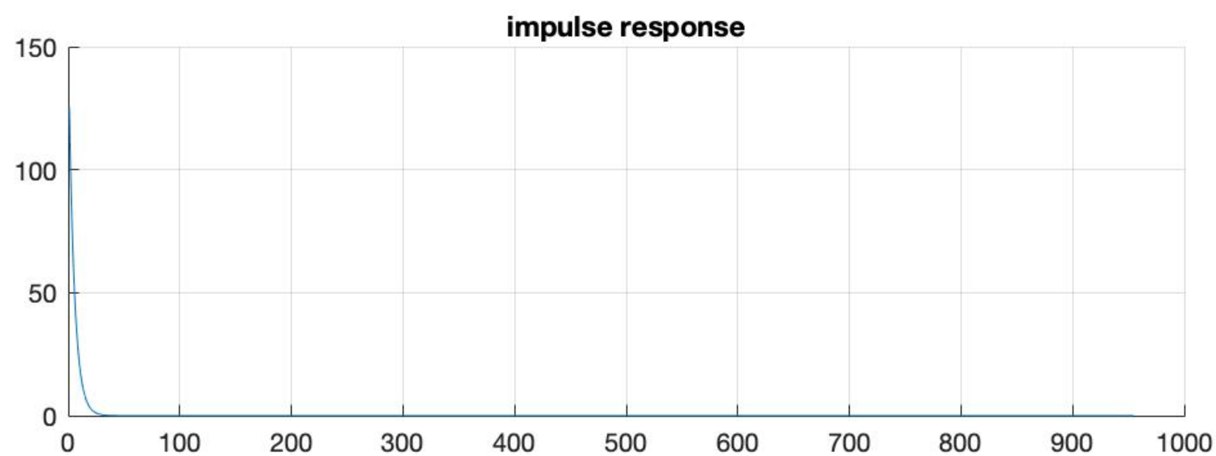


This is the input response
1
0.8
0.6
0.4
0.2
0
-100
-80
-60
-40
-20
0
20
40
60
80
100

Question 7

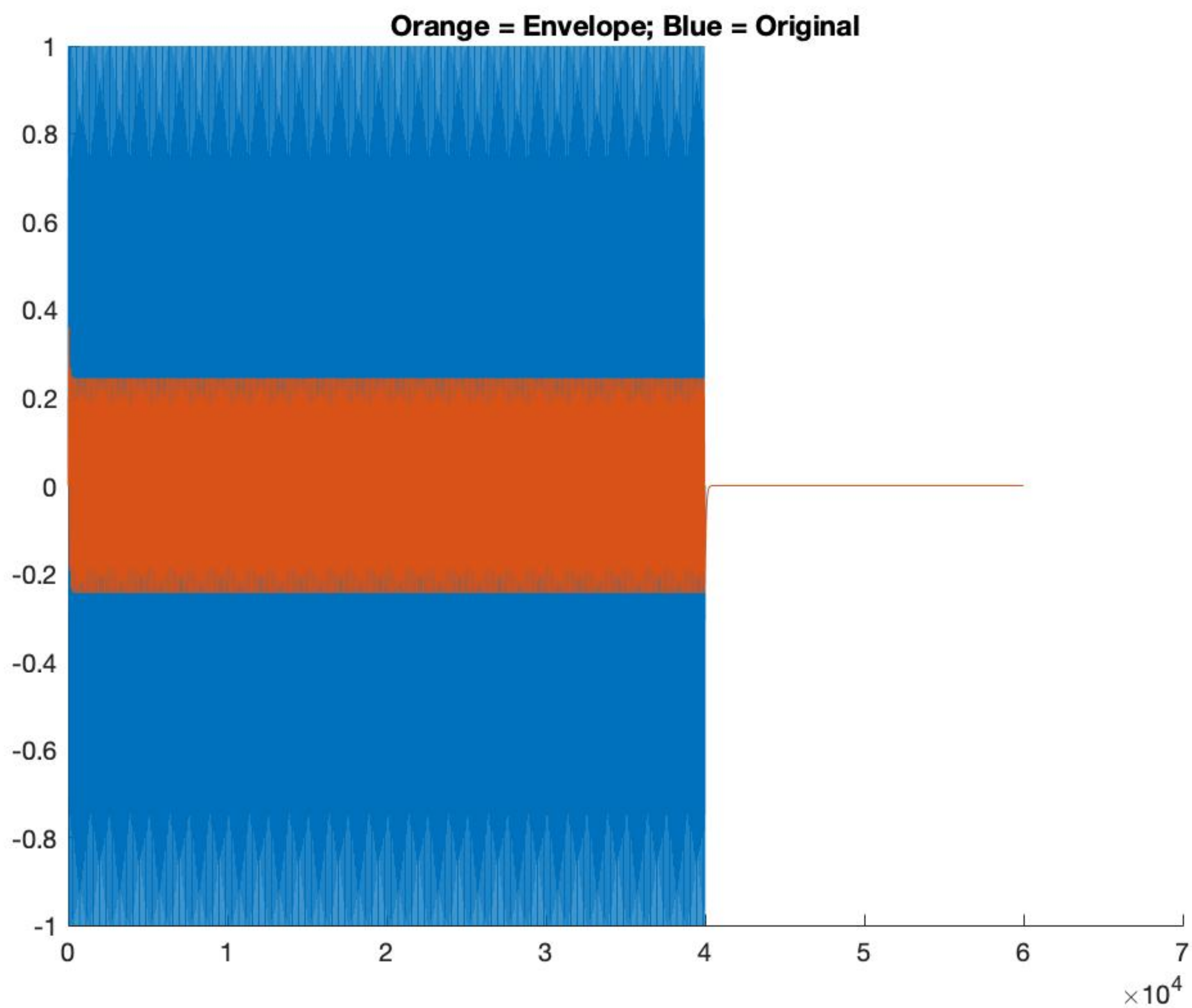


convolution and original graph
Convolution
Input Signal
1
0.5
0
-0.5
-1
0
500
1000
1500

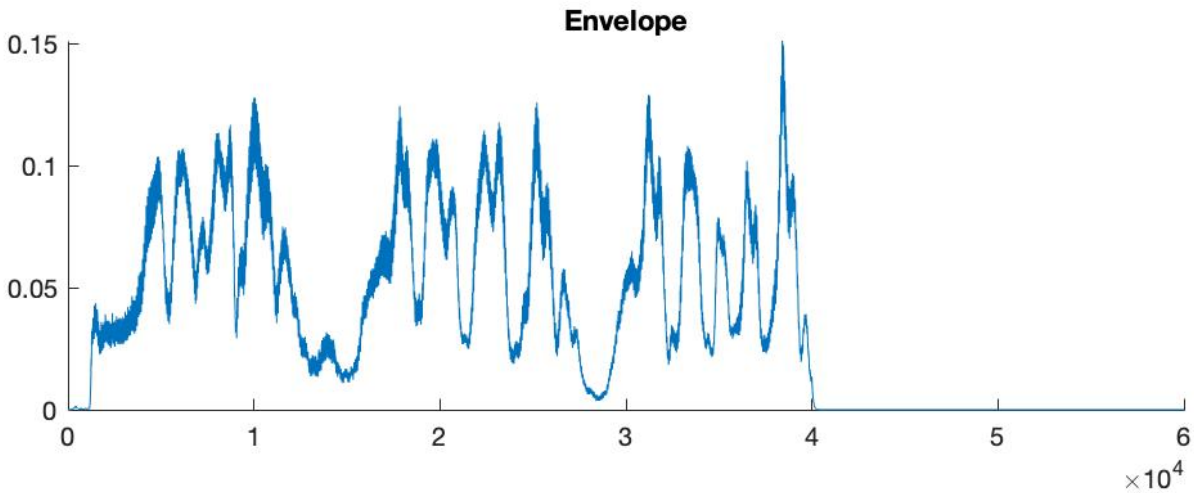


impulse response
150
100
50
0
0
100
200
300
400
500
600
700
800
900
1000

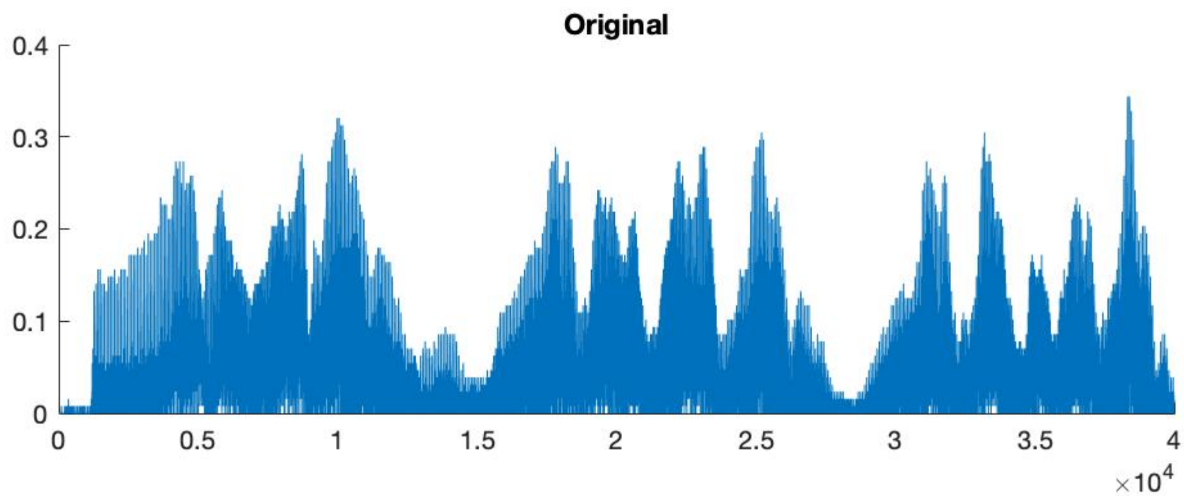
Question 13


Orange = Envelope; Blue = Original
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6
7
$\times 10^4$

Question 14



Envelope
0.15
0.1
0.05
0
0
1
2
3
4
5
6
×10^4



Original
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
3
3.5
4
×10^4