Digital Signal Processing HW3 MATLAB Part

Name: Jingyang Zhang
Net ID: jz2807

1) Load the signal in DataEOG.txt from Resources->Sample Data (Also available from http://eeweb.poly.edu/iselesni/EL6113/misc/DataEOG.txt).
Suppose your measured signal r(n) is a blurred version of the true signal x(n) where the blurring can be modeled as b(n)= x(n)+4 x(n-1) +2 x(n-2). Note that r(n) can be considered as b(n)=x(n)*h(n), where h(n)=[1, 4, 2]. As shown in the example given in the class (also in the note "inverse system example.pdf"), there are two possible inverse filters, one, g1(n), is causal but unstable, another, g2(n), is non-causal but stable. Apply the blurring filter to the signal in the file 'DataEoG.txt', to generate a blurred signal b(n). Then apply each of the two inverse filters to b(n) to obtain the deblurred signal y(n). For implementation using convolution, please truncate the inverse filter to only keep those filter coefficients in the range of -10<=n<=10. Plot x(n), b(n) and deblurred signals by the two inverse filters. Compare the results obtained using these two different inverse filters. Are the results as expected?

Note that you must specify the range of n properly, so that your plot shows the correct "n" values. Suppose the length of the data signal x(n) is N, then the range of n for x is 0 to N-1. If your filter ranges from –K to K, then the filtered signal ranges from –K to N+K-1. Also, because x(n) is very long, you should use "plot" instead of "stem" when plotting these signals.

Note that it may be hard to see the difference between x(n), r(n) and y(n) if you plot the entire signal. In addition to ploting the entire signals, you can plot a subsegment of these signals so that you can see the difference.

Solution:

.m file(s): jyz_HW3.m

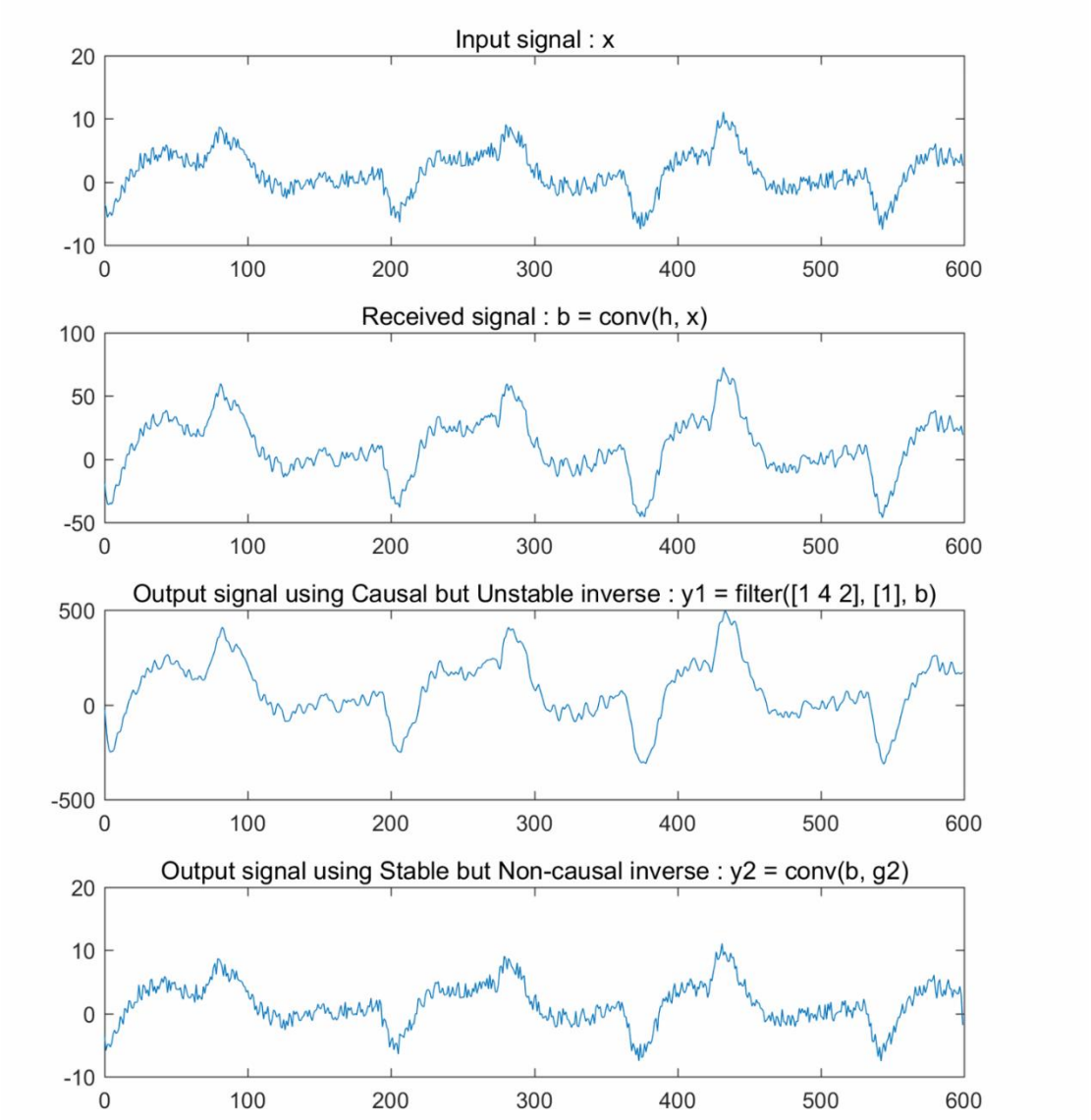
Code:

```
clear
close all

u = @(n) (n >= 0); %Function
del = @(n) (n == 0); % Function

n1 = 0 : 2;
h = del(n1) + 4 * del(n1 - 1) + 2 * del(n1 - 2);
n2 = -10 : 10;
[r, p, k] = residue([1 0],[1 4 2]);
g1 = r(1) * p(1).^n2 .* u(n2) + r(2) * p(2).^n2 .* u(n2);
%Causal but Unstable, ROC: z > 3.4142
g2 = -r(1) * p(1).^n2 .* u(-n2-1) + r(2) * p(2).^n2 .* u(n2);
%Stable but Non-causal, ROC: 0.5858< z <3.4142
```

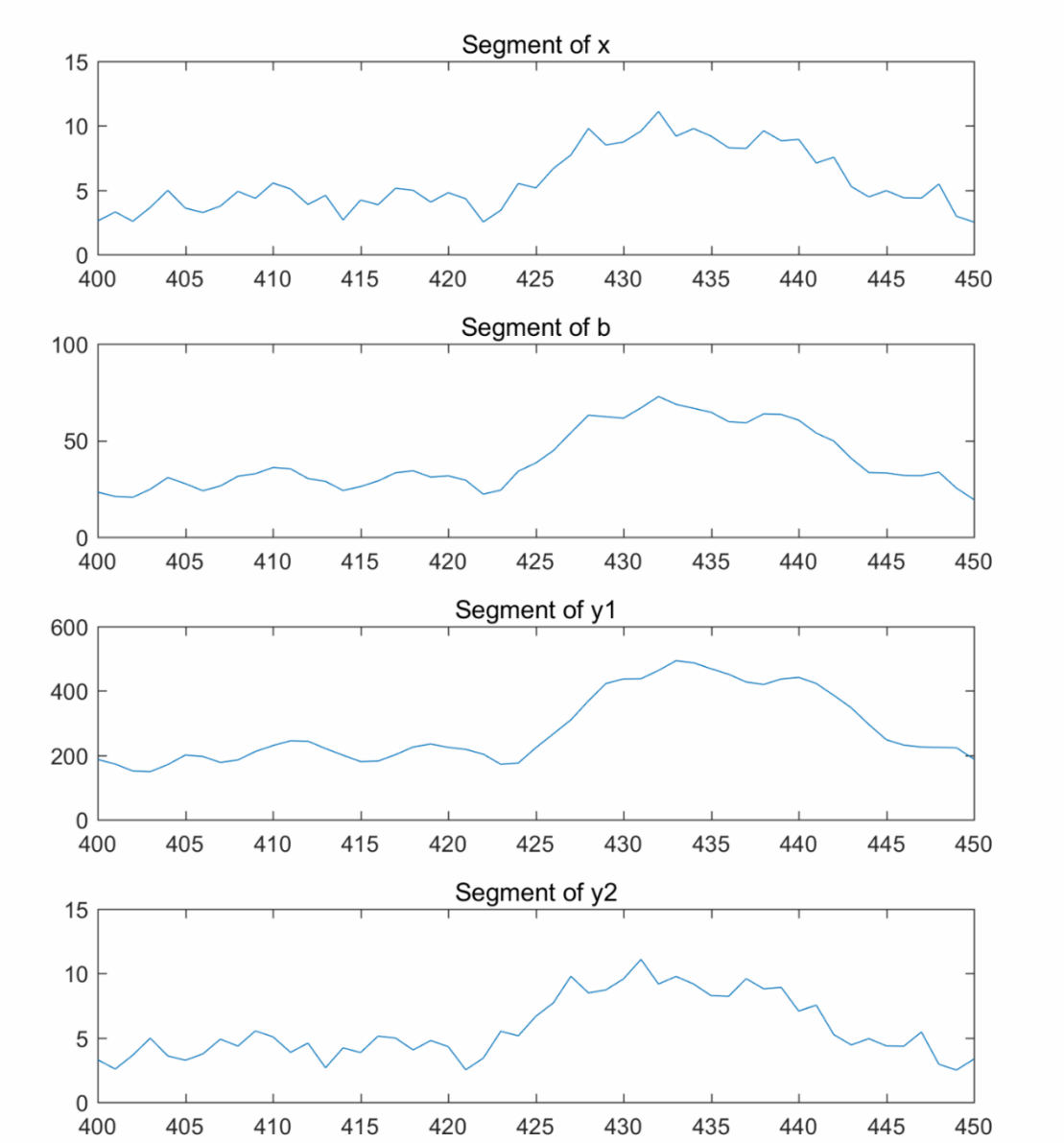
```
load DataEOG.txt
x = DataEOG;
b = conv(x, h);
lenb = length(b);
b(1 : (lenb - length(x)) / 2) = [];
b(end - ((lenb - length(x)) / 2) + 1: end) = [];
%Cutting

y1 = filter([1 4 2], [1], b);

y2 = conv(b, g2);
leny2 = length(y2);
y2(1 : (leny2 - length(x)) / 2) = [];
y2(end - (leny2 - length(x)) / 2 + 1: end) = [];
%Cutting

subplot(4, 2, 1)
plot(0 : length(x) - 1, x)
title('Input signal : x')

subplot(4, 2, 3)
plot(0 : length(b) - 1, b)
title('Received signal : b = conv(h, x)')

subplot(4, 2, 5)
plot(0 : length(y1) - 1, y1)
title('Output signal using Causal but Unstable inverse : y1 = filter([1 4 2], [1], b)')

subplot(4, 2, 7)
plot(0 : length(y2) - 1, y2)
title('Output signal using Stable but Non-causal inverse : y2 = conv(b, g2)')

subplot(4, 2, 2)
plot(0 : length(x) - 1, x)
xlim([400 450])%Set range of axe of n
title('Segment of x')

subplot(4, 2, 4)
plot(0 : length(b) - 1, b)
xlim([400 450])%Set range of axe of n
title('Segment of b')

subplot(4, 2, 6)
```

```
plot(0 : length(y1) - 1, y1)
xlim([400 450])%Set range of axe of n
title('Segment of y1')

subplot(4, 2, 8)
plot(0 : length(y2) - 1, y2)
xlim([400 450])%Set range of axe of n
title('Segment of y2')
```

Result(Overview plots):



Result(Plots of signal segments):



Discussion:

a) Convolution makes a difference between length of the input and output signal. In this program, in order to keep the length of 4 signals(x, b, y1, y2) the same, I cut the first and last several data from b(n), y1(n) and y2(n), the length of these 4 signals are all 600.

b) b(n) = (x * h)(n), as a result, the plot of b(n) is a smoother than that of the input signal, x(n), in other words, b(n) is a blurred signal of x(n).

c) The MATLAB function *residue()* is used to find the parameter of 2 inverse filter. As for the property of these filters.

   The ROC of G1(z) is z>3.4142, g1(n) = r1·p1^n·u(n) + r2·p2^n·u(n), so g1 is causal but unstable.

   The ROC of G2(z) is 0.5858<z<3.4142, g2(n)=-r1·p1^n·u(-n-1) + r2·p2^n·u(n),

so g2 is Stable but Non-causal.

r1=1.2071, r2=-0.2071, p1 =-3.4142, p2=-0.5858.

d) The inverse filter g1 does not deblur, or, recover, the input signal in terms of magnification. As we can see in the plot, the output signal is amplified by about 25 times. But it generally deblur the signal in terms of shape. When looking at a segment ($400 \leq n \leq 450$)of the 4 signals, we can notice y1(n) is smother than the input signal x(n).
e) The inverse filter g2 deblur the signal in a considerably high level, no matter in terms of shape, magnification and details. Besides, there is hardly any delay.