DSP2 Week 2 experiment Report

2019023436 Kim, Hyunsoo

EXERCISE 1

a) Make the function 'impseq', 'stepseq'

(Source Code)

```
function [x, n] = impseq(n0, lb, ub)
    n = [lb:ub];
    x = [(n-n0) == 0];
end
```

```
function [x, n] = stepseq(n0, lb, ub)
    n = [lb:ub];
    x = [(n - n0) >= 0];
end
```

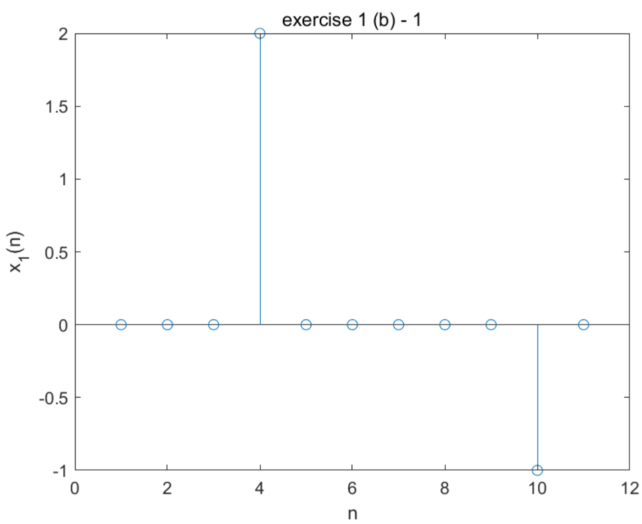
b) Plot (stem) each of the following sequences over the indicated interval.

(1) (Source Code)

exercise 1 (b) - 1

```
x_1 = 2 * impseq(-2, -5, 5) - impseq(4, -5, 5)
stem(x_1)
title('Problem(1)');
xlabel('n');
ylabel('x_1(n)');
```

(Result)

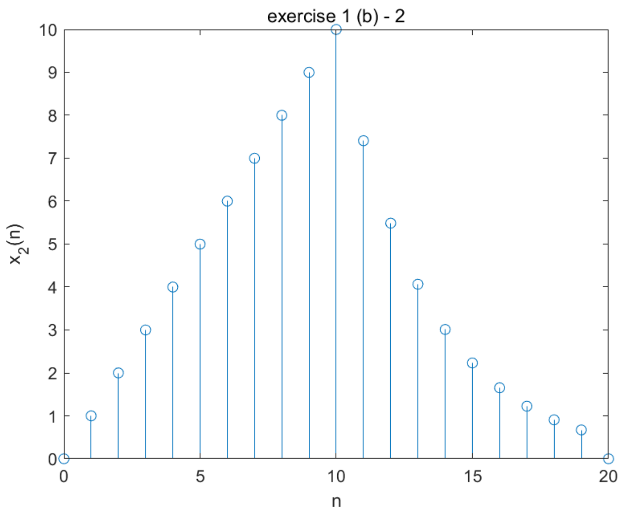


(2) (Source Code)

exercise 1 (b) - 2

```
n = 0:20;
x_2 = (n .* (stepseq(0, 0, 20) - stepseq(10, 0, 20))) + 10 * exp(-0.3 .* (n - 10)) .* (stepseq(10, 0, 20) - stepseq(20, 0, 20));
stem(n, x_2)
title('exercise 1 (b) - 2');
xlabel('n');
ylabel('x_2(n)');
```

(Result)



EXERCISE 2

a) Make the function 'conv_m'

(Source Code)

```
function [y, ny] = conv_m(x, nx, h, nh)
ny_start = nx(1) + nh(1);
ny_end = nx(length(x)) + nh(length(h));
ny = ny_start:ny_end;
y = conv(x, h);
end
```

b) Generate and plot below signals. Explain graphically whether it is time-variant or time-invariant.

1. (Source Code)

Exercise 2 (b) - 1

```
n = -10:10
x_1 = stepseq(0, -10, 10)
y_1 = (0.8.^n) .* x_1
plot(n, y_1, 'g')
```

2. (Source Code)

Exercise 2 (b) - 2

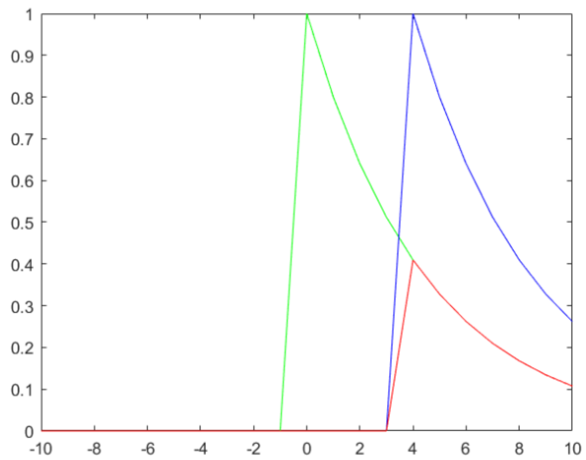
```
hold on
x_2 = stepseq(4, -10, 10)
y_2 = 0.8.^(n-4) .* x_2
plot(n, y_2, 'b')
```

3. (Source Code)

Exercise 2 (b) - 3

```
x_3 = stepseq(4, -10, 10)
y_3 = 0.8.^(n) .* x_3
plot(n, y_3, 'r')
hold off
```

(Result)



(Conclusion)

$y[n-4]$ and $y_4[n]$ is not the same and we can find it from the above graph: The red one ($y_4[n]$) and the blue one ($y[n-4]$). Thus, $y[n]$ is time-invariant.

c) LTI system

1. Show numerically the identity h1 * delta[n] = h1 is correct.

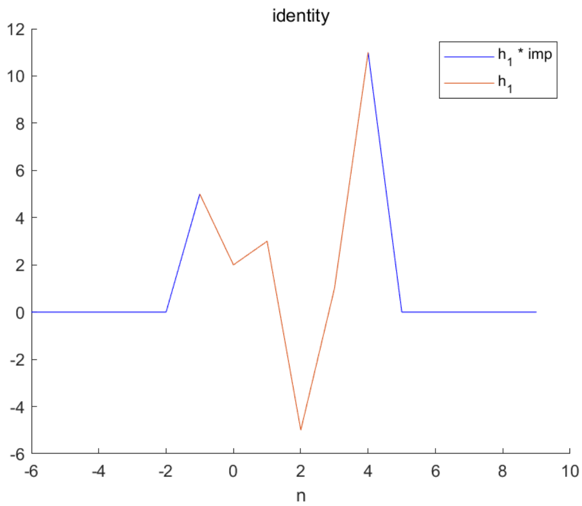
(Source Code)

Exercise 2 (c) - 1

```
[imp, impn] = impseq(0, -5, 5)
h_1 = [5 2 3 -5 1 11]
h_1_n = -1:4

[y, ny] = conv_m(h_1, h_1_n, imp, impn)

hold on
plot(ny, y, 'b')
plot(h_1_n, h_1)
title('identity');
xlabel('n');
legend('h_1 * imp', 'h_1')
hold off
```

(Result)

```
imp = 1×11 logical 배열
     0    0    0    0    0    1    0    0    0    0    0

impn = 1×11
    -5    -4    -3    -2    -1     0     1     2     3     4     5

h_1 = 1×6
     5     2     3    -5     1    11

h_1_n = 1×6
    -1     0     1     2     3     4

y = 1×16
     0     0     0     0     0     5     2     3    -5     1    11     0     0     0     0     0

ny = 1×16
    -6    -5    -4    -3    -2    -1     0     1     2     3     4     5     6     7     8     9
```



(Conclusion)

y is the result of $h_1 * \delta$ and h_1 is h_1, and they are the same. Above graph also shows that the graph of h_1 and the graph of $h_1 * \delta$ coinside.

2. By plotting both h1 * h2 * x and h2 * h1 * x, show numerically they are the same. Write down the error between the two by subtracting one from another.

(Source Code)

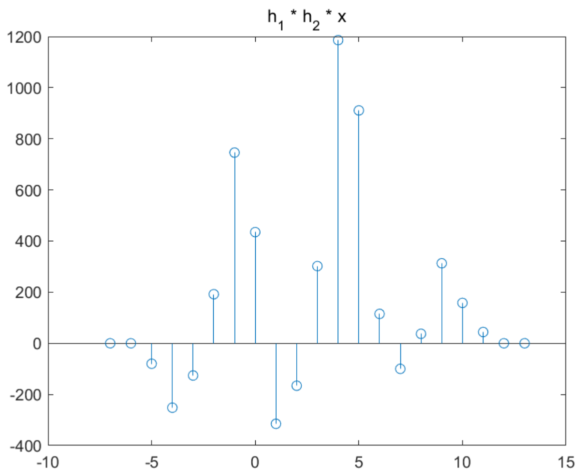
Exercise 2 (c) - 2

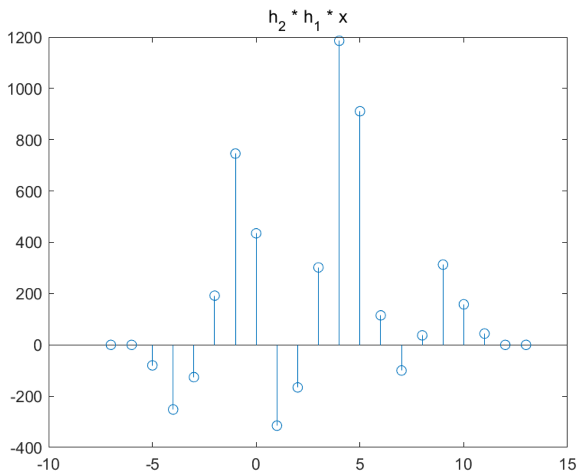
```
x_n = -5:5
x = [0 0 4 13 6 -3 -1 2 4 0 0]
h_2_n = -1:4
h_2 = [-4 2 0 6 3 1]
```

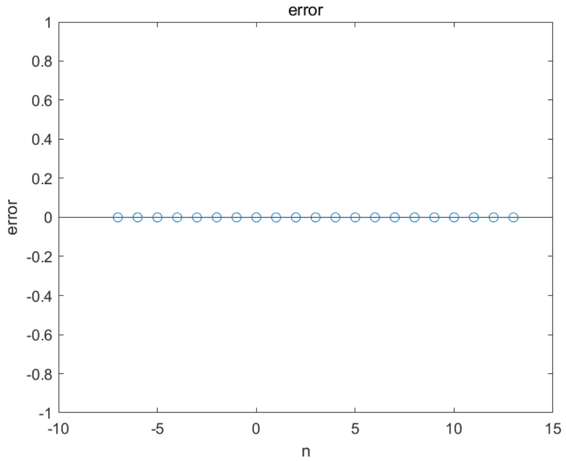
```
[tmp, tmpn] = conv_m(h_1, h_1_n, h_2, h_2_n);
[y1, ny] = conv_m(tmp, tmpn, x, x_n);
stem(ny, y1)
title('h_1 * h_2 * x');

[tmp, tmpn] = conv_m(h_2, h_2_n, h_1, h_1_n);
[y2, ny] = conv_m(tmp, tmpn, x, x_n)
stem(ny, y2)
title('h_2 * h_1 * x');
```

```
error = y1 - y2
stem(ny, error)
title('error');
xlabel('n');
ylabel('error');
```

(Result)







(Conclusion)

$h_1 * h_2 * x$ and $h_2 * h_1 * x$ have exactly the same graph and the error between the tow by subtracting one from another is 0 for all n.

EXERCISE 3

a) Create a vector from $x[0]$ to $x[10]$ where $x[n] = 0.5^n\, u[n]$ by using vectorization.

(Source Code)

Exercise 3 (a)

```
n = 0:10;
x = (0.5 .^ n) .* stepseq(0, 0, 10)
```

(Result)

```
x = 1×11
    1.0000    0.5000    0.2500    0.1250    0.0625    0.0312    0.0156    0.0078    0.0039    0.0020    0.0010
```

b) Write a MATLAB function for the accumulator which yields a vector of $\{y[n]\}$ where $y[n] = y[n-1] + x[n]$

(Source Code)

function accumulator

```
function y = accumulator(x)
    y = 1:length(x);
    for i = 1:length(x)
        if i == 1
           y(i) = x(i);
        else
           y(i) = y(i-1) + x(i);
        end
    end
end
```

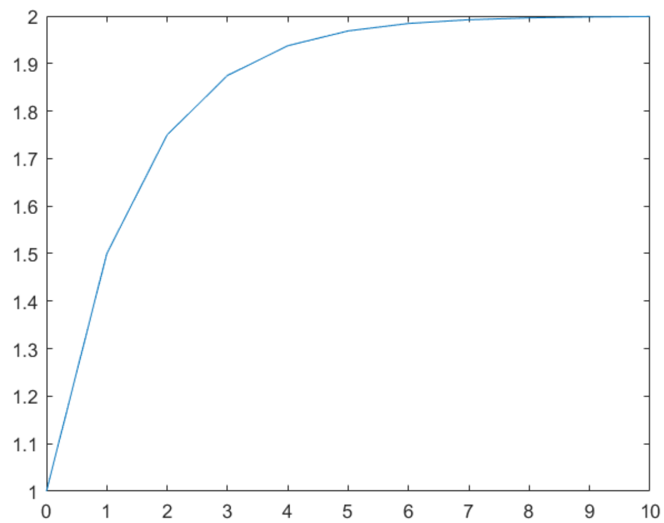
c) Plot $y[n]$ and compute the value of convergence when $n \to \infty$.

(Source Code)

```
y = accumulator(x)
plot(n, y);
```

(Result)

```
y = 1×11
    1.0000    1.5000    1.7500    1.8750    1.9375    1.9688    1.9844    1.9922    1.9961    1.9980    1.9990
```



(Conclusion)

accumulator is an example of LCCDE system.

0. Given equation:

   A. $x[n] = 0.5^n u[n]$

   B. $y[n] = y[n-1] + x[n]$

C. $y[n] - y[n-1] = x[n]$

1. Complementary solution

   A. $y_g[n] - y_g[n-1] = 0$

   B. Guess: $y_g[n] = c\,\lambda^n$

   C. $y_g[n] - y_g[n-1] = c\,\lambda^n - c\,\lambda^{n-1} = 0$

   D. $\lambda - 1 = 0$

   E. $\lambda = 1$

   F. $y_g[n] = c_1(1)^n = c_1$

2. Particular solution

   A. $y[n] - y[n-1] = x[n] = 0.5^n\,u[n]$

   B. $y_p[n] = C\,0.5^n\,u[n]$

   C. $C\,0.5^n\,u[n] - C\,0.5^{n-1}\,u[n-1] = 0.5^n\,u[n]$

   D. set $n \geq 1$ (order of LCCDE)

   E. at $n = 1$,

   F. $C\,0.5^1\,u[1] - C\,0.5^0\,u[0] = 0.5^1\,u[1]$

   G. $C\,0.5 - C = 0.5$

   H. $C = -1$

   I. $y_p[n] = -0.5^n\,u[n]$

3. Boundary Conditions

   A. $n_0 = 0$: time at which the signal arrives

   B. $N = 1$: order of this system

   C. This system is initially-at-rest.

   D. For $n < 0$: $x[n] = 0,\ y[n] = 0$

   E. For $n \geq 1$:

i. The complementary solution adds up to $y_g[n] - y_g[n-1] = 0$

ii. The particular solution adds up to $y_p[n] - y_p[n-1] = x[n]$

F. For $0 \le n < 1\ (N\ data\ points)$:

i. The complementary and particular solutions contribute jointly to form $(y_g[n] + y_p[n]) - (y_g[n-1] + y_p[n-1]) = x[n]$

ii. Explicitly set the complementary solution coefficients to get $x[n]$.

iii. $Set\ n = 0$

1. $y_g[0] + y_p[0] - (0 + 0) = x[0]$
2. $C_1 - 1 = 1$
3. $C_1 = 2$

iv. $Thus,\ y[n] = y_g[n] + y_p[n] = 2 - 0.5^n u[n]$

4. Computing the value of convergence when $n \to \infty$,

A. $\lim_{n\to\infty} (2 - 0.5^n\, u[n]) = 2 - 0 = 2$