

# SS - lab S7 - Zeic Benjamin

## Exercise 1

Compute the convolution of the following signals using the analytical method, the graphical method and the function conv.

$$x[n] = u[n] - u[n-2], y[n] = 0.2^n u[n]$$

$$\begin{aligned} x[n] * y[n] &= \sum_{k=-\infty}^{\infty} x[k] y[n-k] \\ &= \sum_{k=-\infty}^{\infty} (u[k] - u[k-2]) 0.2^{n-k} u[n-k] \\ &= \sum_{k=-\infty}^{\infty} u[k] 0.2^{n-k} u[n-k] - \sum_{k=-\infty}^{\infty} u[k-2] 0.2^{n-k} u[n-k] \\ &= \sum_{k=0}^n 0.2^{n-k} u[n-k] - \sum_{k=2}^n 0.2^{n-k} u[n-k] \\ &= 0.2^n \sum_{k=0}^n 5^k u[n-k] - 0.2^n \left( \sum_{k=0}^n 5^k - 6 \right) u[n-2] \\ &= 0.2^n \frac{(1-5^{n+1})}{1-5} u[n] - 0.2^n \frac{(25-5^{n+1})}{1-5} u[n-2] \\ &= \frac{(1-5^{n+1}) u[n] - (25-5^{n+1}) u[n-2]}{-4(5^n)} \end{aligned}$$

In the above obtained equation, we must re-center the two unit step functions ( $u[n]$  will become  $u[n+1]$  and  $u[n-2]$  will become  $u[n-1]$ ).

```
prec = 1;
x1 = @(n) ((n>=0)-(n>=2));
y1 = @(n) (0.2.^n.*(n>=0));
sx1 = -2; ex1 = 10; dtx1 = sx1:prec:ex1;
sy1 = -2; ey1 = 10; dty1 = sy1:prec:ey1;
dtw1 = sx1+sy1:prec:ex1;

w1 = conv(double(x1(dtx1)), y1(dty1));

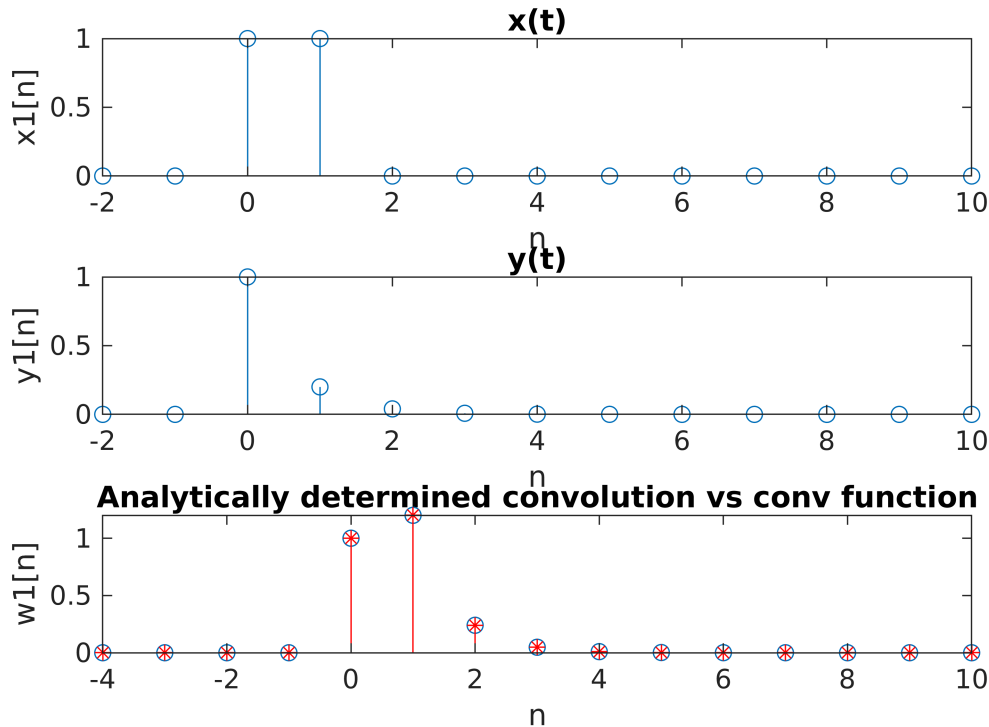
figure;
sgtitle("Exercise 1")
subplot(3,1,1); stem(dtx1,x1(dtx1)); xlabel('n'); ylabel('x1[n]');
title("x(t)")
subplot(3,1,2); stem(dty1,y1(dty1)); xlabel('n'); ylabel('y1[n]');
title("y(t)")
subplot(3,1,3); stem(dtw1,w1(1:length(dtw1))); xlabel('n'); ylabel('w1[n]'); hold on

title("Analytically determined convolution vs conv function")
```

```
% w_calculated = @(n) ((0.2.^n).*6.*(n>=0));
w_calculated = @(n) (((1-5.^(n+1)).*(n>-1) - (25-5.^(n+1)).*(n>1))./(-4*(5.^n)));

stem(dtw1, w_calculated(double(dtw1)), 'r');
hold off;
```

## Exercise 1



## Exercise 2

Let  $x[n] = \delta[n] + 2\delta[n-1] - \delta[n-3]$  and  $h[n] = 2\delta[n+1] + 2\delta[n-1]$

Compute the convolution of:

- i.  $y[n] = x[n] * h[n]$
- ii.  $y[n] = x[n+1] * h[n]$
- iii.  $y[n] = x[n] * h[n+2]$

Represent graphically in Matlab the signals and their convolutions.

Subpoint A

```
% Subpoint A
t = -4:1:4;
% Define x[n] and h[n] as an anonymous functions
% Use eq to simulate the Kronecker-delta function
x_func = @(n) (eq(n,0)+2*eq(n-1,0)-eq(n-3,0));
h_func = @(n) (2*eq(n+1,0)+2*eq(n-1,0));
```

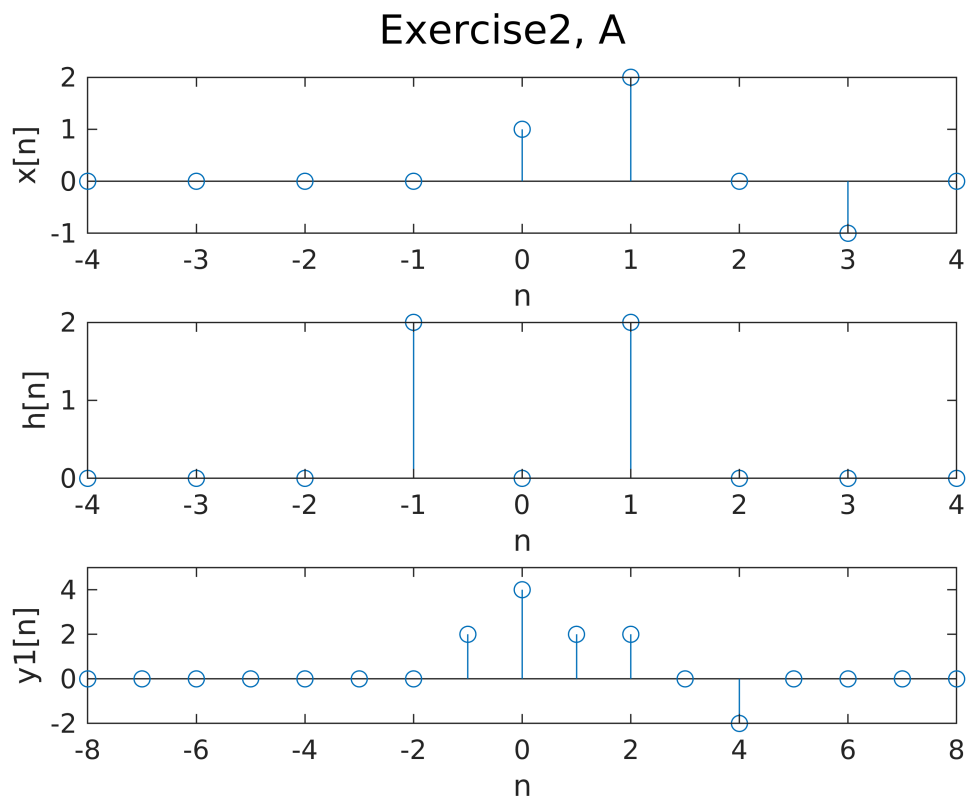
```

% Obtain the signal x[n] for the timespan defined in "t"
x = x_func(t);
h = h_func(t);

y1 = conv(x, h);

figure
sgtitle("Exercise2, A");
subplot(3,1,1); stem(t, x); xlabel('n'); ylabel('x[n]');
subplot(3,1,2); stem(t, h); xlabel('n'); ylabel('h[n]');
subplot(3,1,3); stem(-8:8, y1); xlabel('n'); ylabel('y1[n]');

```



### Subpoint B

```

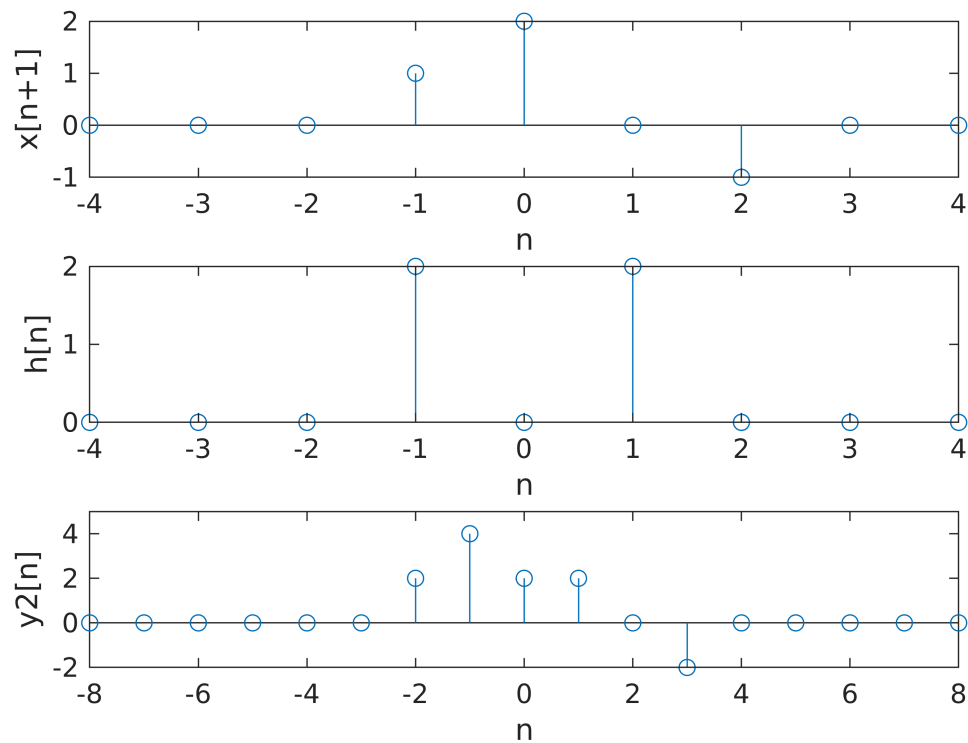
% Subpoint B
% Obtain the signal x[n] for the timespan defined in "t"
x_nplus1 = x_func(t+1);

y2 = conv(x_nplus1, h);

figure
sgtitle("Exercise2, B");
subplot(3,1,1); stem(t, x_nplus1); xlabel('n'); ylabel('x[n+1]');
subplot(3,1,2); stem(t, h); xlabel('n'); ylabel('h[n]');
subplot(3,1,3); stem(-8:8, y2); xlabel('n'); ylabel('y2[n]');

```

## Exercise2, B



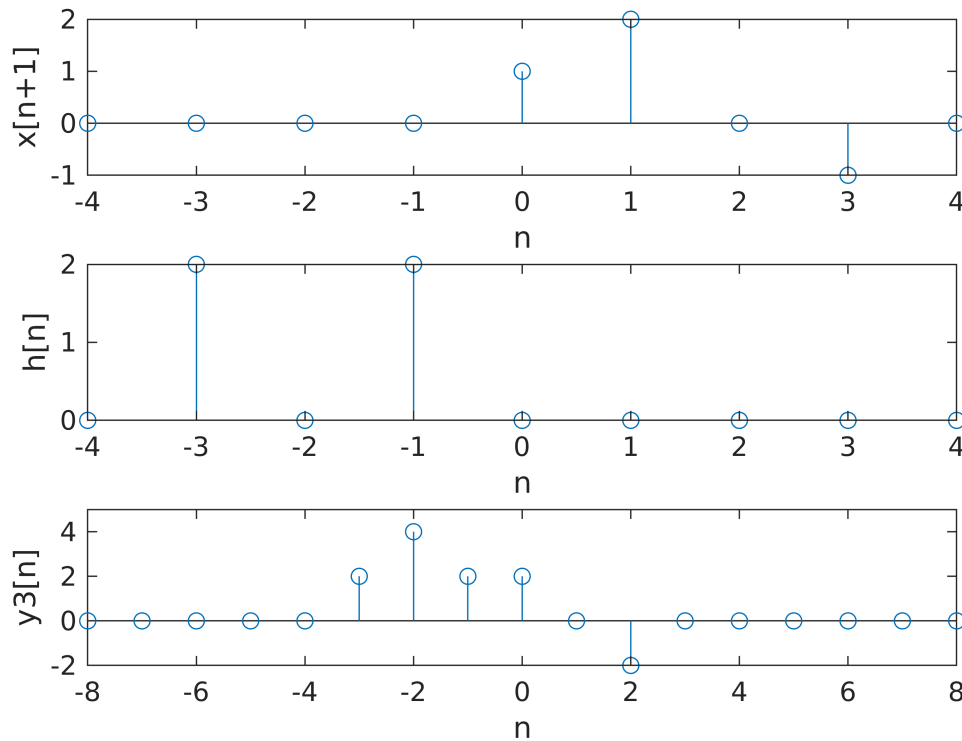
### Subpoint C

```
% Subpoint C
% Obtain the signal x[n] for the timespan defined in "t"
h_nplus2 = h_func(t+2);

y3 = conv(x, h_nplus2);

figure
sgtitle("Exercise2, C");
subplot(3,1,1); stem(t, x); xlabel('n'); ylabel('x[n+1]');
subplot(3,1,2); stem(t, h_nplus2); xlabel('n'); ylabel('h[n]');
subplot(3,1,3); stem(-8:8, y3); xlabel('n'); ylabel('y3[n]');
```

## Exercise2, C



### Exercise 3

Consider two discrete LTI systems S1 and S2, connected in series. The impulse response of S1 is  $h_1[n] = \alpha u[n]$ ,  $|\alpha| < 1$  and of S2 is  $h_2[n] = \sin(8n)$ . Find the output of the system S for the input  $x[n] = \delta[n] - \alpha\delta[n-1]$ .

Represent graphically in Matlab all the signals and the system response.

The two systems, S1 and S2 are connected in series, so the output of system S1 will be considered to be the input for system S2.

The response of each of the systems to the input signal will be calculated as the convolution of the impulse response and the input signal.

```
% Define the time interval
t_init = -4;
t_fin = 4;
t = t_init:1:t_fin;
% Define the unit step function
u = @(n) (n>=0);
alpha = 1;
% Define h1 and h2, the impulse response of the two systems
h1 = @(n) (min(max(alpha,-1),1).*u(t));
h2 = @(n) (sin(8*n));
```

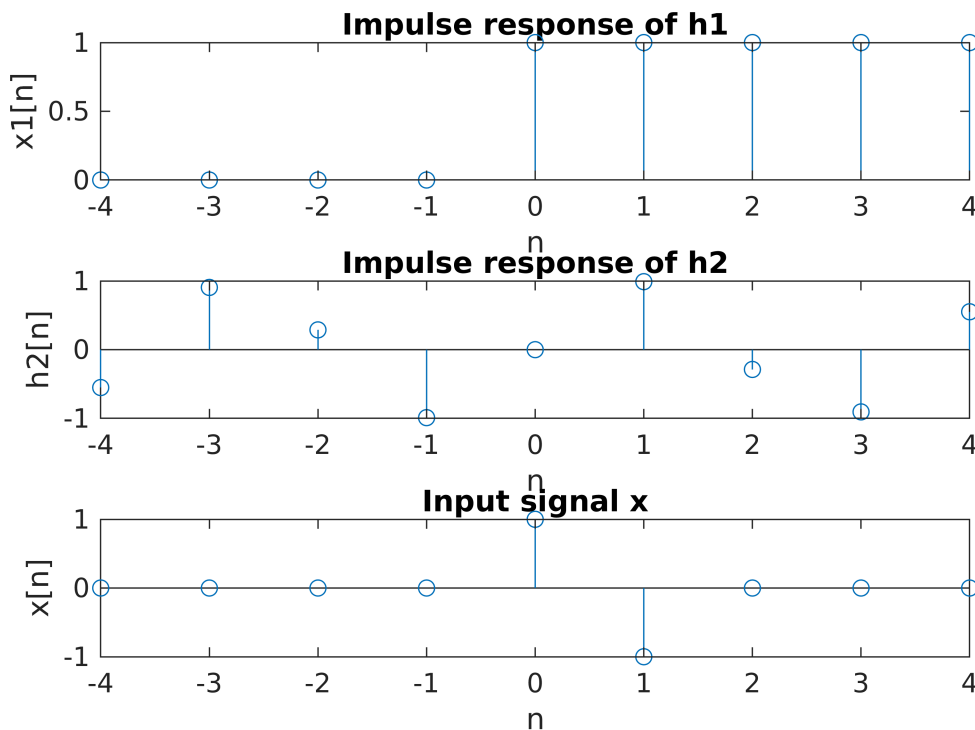
```

% Define the input signal, x
x = @(n) (eq(n,0) - alpha*eq(n-1,0));

figure
sgtitle("Exercise3");
subplot(3,1,1); stem(t, h1(t)); xlabel('n'); ylabel('x1[n]');
title("Impulse response of h1");
subplot(3,1,2); stem(t, h2(t)); xlabel('n'); ylabel('h2[n]');
title("Impulse response of h2");
subplot(3,1,3); stem(t, x(t)); xlabel('n'); ylabel('x[n]');
title("Input signal x");

```

### Exercise3



```

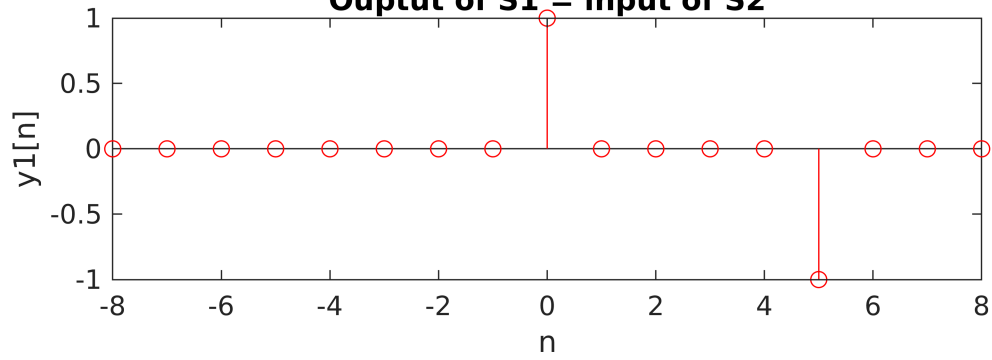
% Output of S1 is the input x, convolved with the impulse response of S1
% (i.e. h1[n]).
y1 = conv(x(t), h1(t));
figure
sgtitle("Output signals");
subplot(2,1,1); stem(2*t_init:2*t_fin, y1, "r"); xlabel('n'); ylabel('y1[n]');
title("Ouptut of S1 = input of S2");

y2 = conv(y1, h2(2*t_init:2*t_fin));
subplot(2,1,2); stem(4*t_init:4*t_fin, y2, "g"); xlabel('n'); ylabel('y2[n]');
title("Ouptut of S2 = ouput of S");

```

## Output signals

**Ouput of S1 = input of S2**



**Ouput of S2 = ouput of S**

