

## به نام خدا

نام: امیرمسعود

نام خانوادگی: شاکر

شماره دانشجویی: 97243081

تمرین دوم

(1) بخش تحقیق:

Syntax	Meaning	Description
1) $[h,t] = \text{impz}(b,a)$ 2) $[h,t] = \text{impz}(\text{sos})$ 3) $[h,t] = \text{impz}(d)$ 4) $[h,t] = \text{impz}(\_, n)$ 5) $[h,t] = \text{impz}(\_, n, fs)$ 6) $\text{Impz}(\_)$	Impulse response of digital filter	1) This syntax returns the impulse response of the digital filter with numerator coefficients $b$ and denominator coefficients $a$ . The function chooses the number of samples and returns the response coefficients in $h$ and the sample times in $t$ . 2) This syntax returns the impulse response of the filter specified by the second-order sections matrix $\text{sos}$ . 3) This syntax returns the impulse response of the digital filter $d$ . 4) This syntax specifies what impulse-response samples to compute.

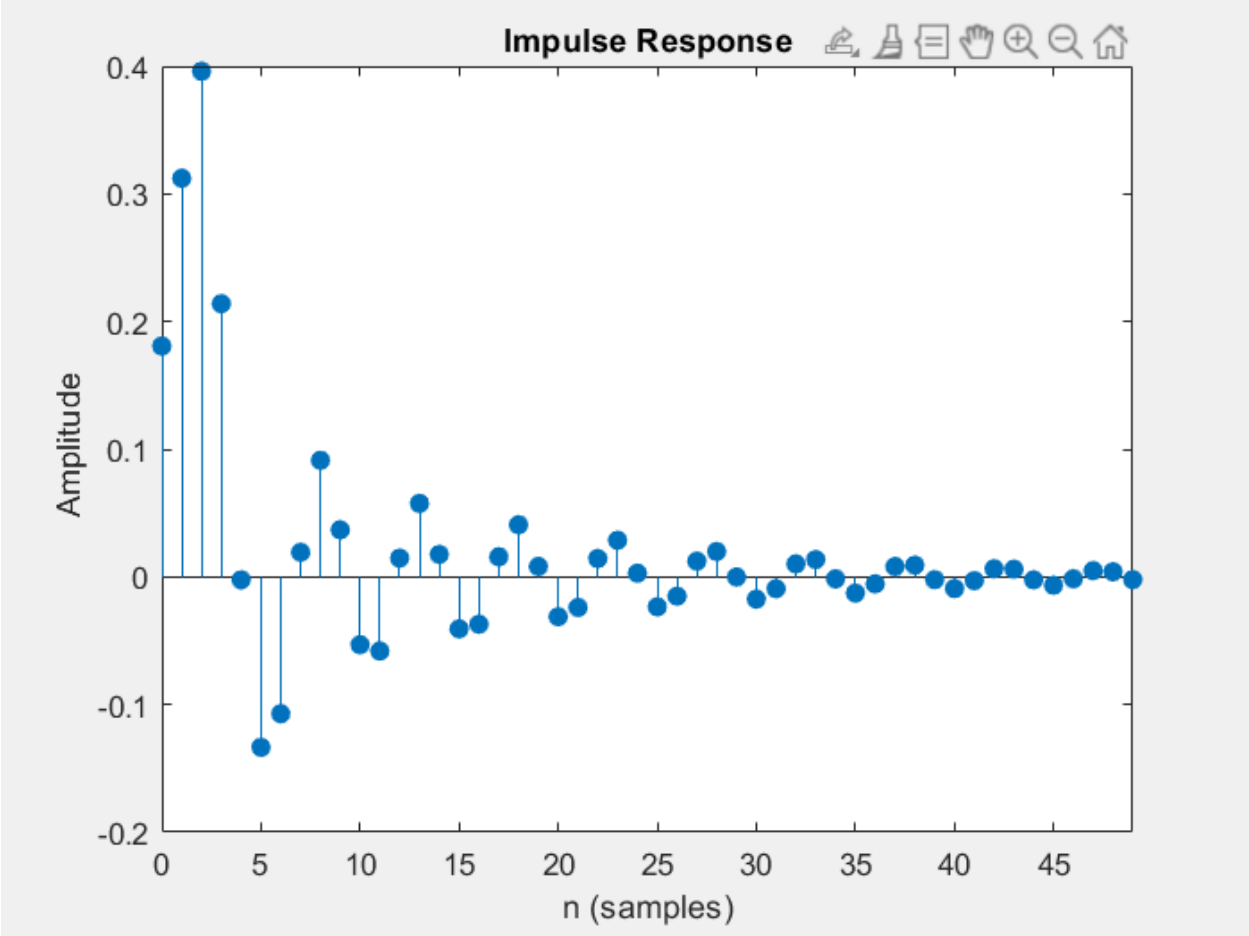
		<p>5) This syntax returns a vector <math>t</math> with consecutive samples spaced <math>1/f_s</math> units apart.</p> <p>6) This syntax with no output arguments plots the impulse response of the filter.</p>
--	--	--

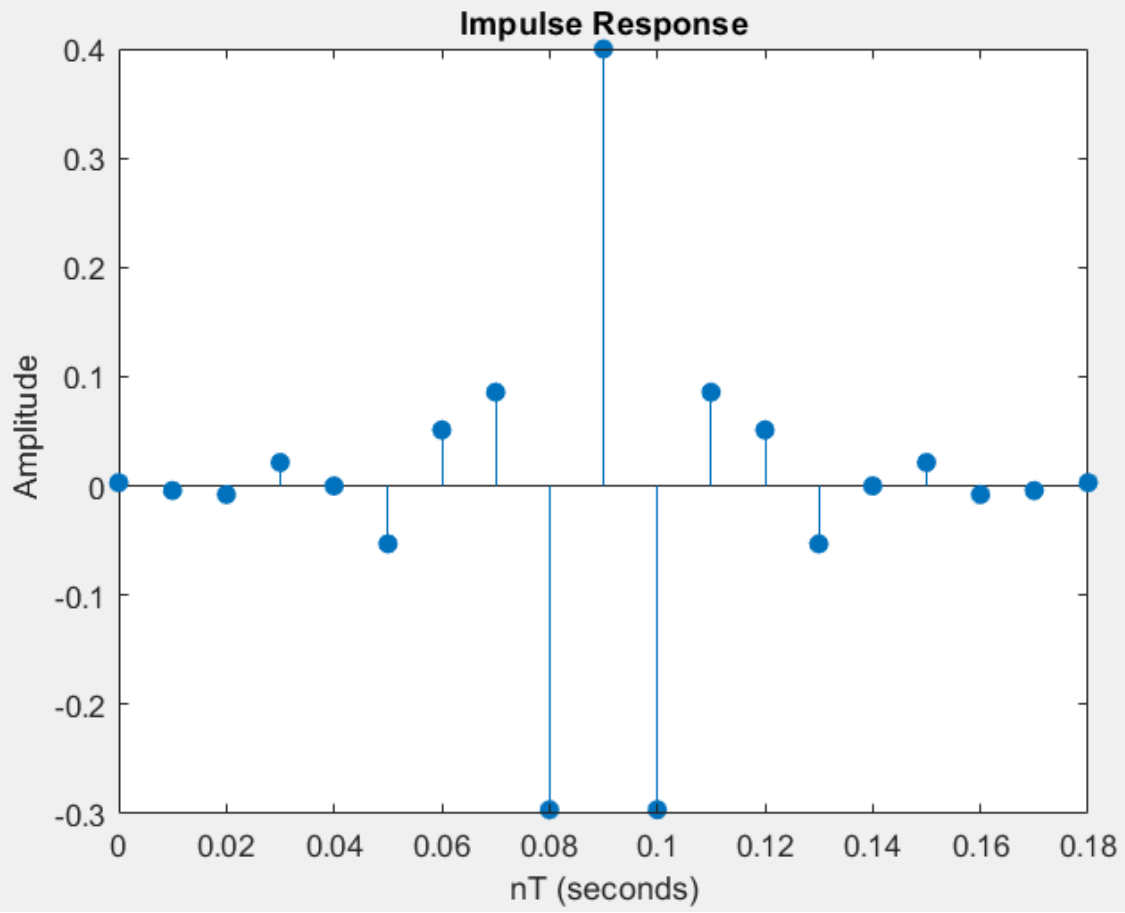
\* `impz` filters a length- $n$  impulse sequence using `filter(b,a,[1 zeros(1,n-1)])` and plots the result using `stem`.

### Examples:

```
[b,a] = ellip(4,0.5,20,0.4);
figure(1)
impz(b,a,50)
% Impulse Response of Elliptic Lowpass Filter

figure(2)
b = fir1(18,30/(100/2),'high',kaiser(19,4));
impz(b,1,[],100)
% Impulse Response of Highpass FIR Filter
```





Syntax	Meaning	Description
1) <code>y = filter(b,a,x)</code> 2) <code>y = filter(b,a,x,zi)</code> 3) <code>y = filter(b,a,x,zi,dim)</code> 4) <code>[y,zf] = filter( __ )</code>	1-D digital filter	1) This syntax filters the input data <code>x</code> using a rational transfer function defined by the numerator and denominator coefficients <code>b</code> and <code>a</code> . 2) This syntax uses initial conditions <code>zi</code> for the filter delays. The length of <code>zi</code> must equal $\max(\text{length}(a), \text{length}(b)) - 1$ . 3) This syntax acts along dimension <code>dim</code> . 4) This syntax returns the final conditions <code>zf</code> of the filter delays, using any of the previous syntaxes.

## Examples:

```
% Moving-Average Filter
t = linspace(-pi,pi,100);
rng default %initialize random number generator
x = sin(t) + 0.25*rand(size(t));

windowSize = 5;
b = (1/windowSize)*ones(1,windowSize);
a = 1;

y = filter(b,a,x);

figure(1);
plot(t,x)
hold on
plot(t,y)
legend('Input Data','Filtered Data')

% Filter Matrix Rows
rng default %initialize random number generator
x = rand(2,15);

b = 1;
a = [1 -0.2];

y = filter(b,a,x,[],2);

t = 0:length(x)-1; %index vector

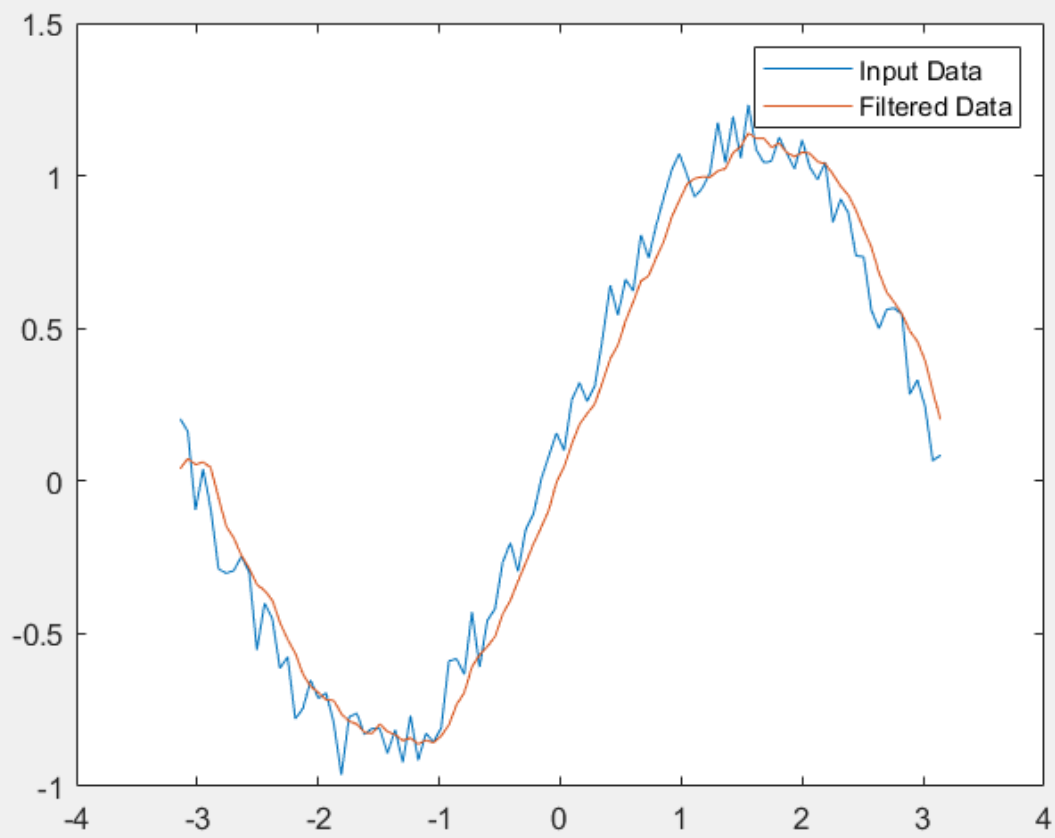
figure(2)
plot(t,x(1,:))
hold on
plot(t,y(1,:))
legend('Input Data','Filtered Data')
title('First Row')

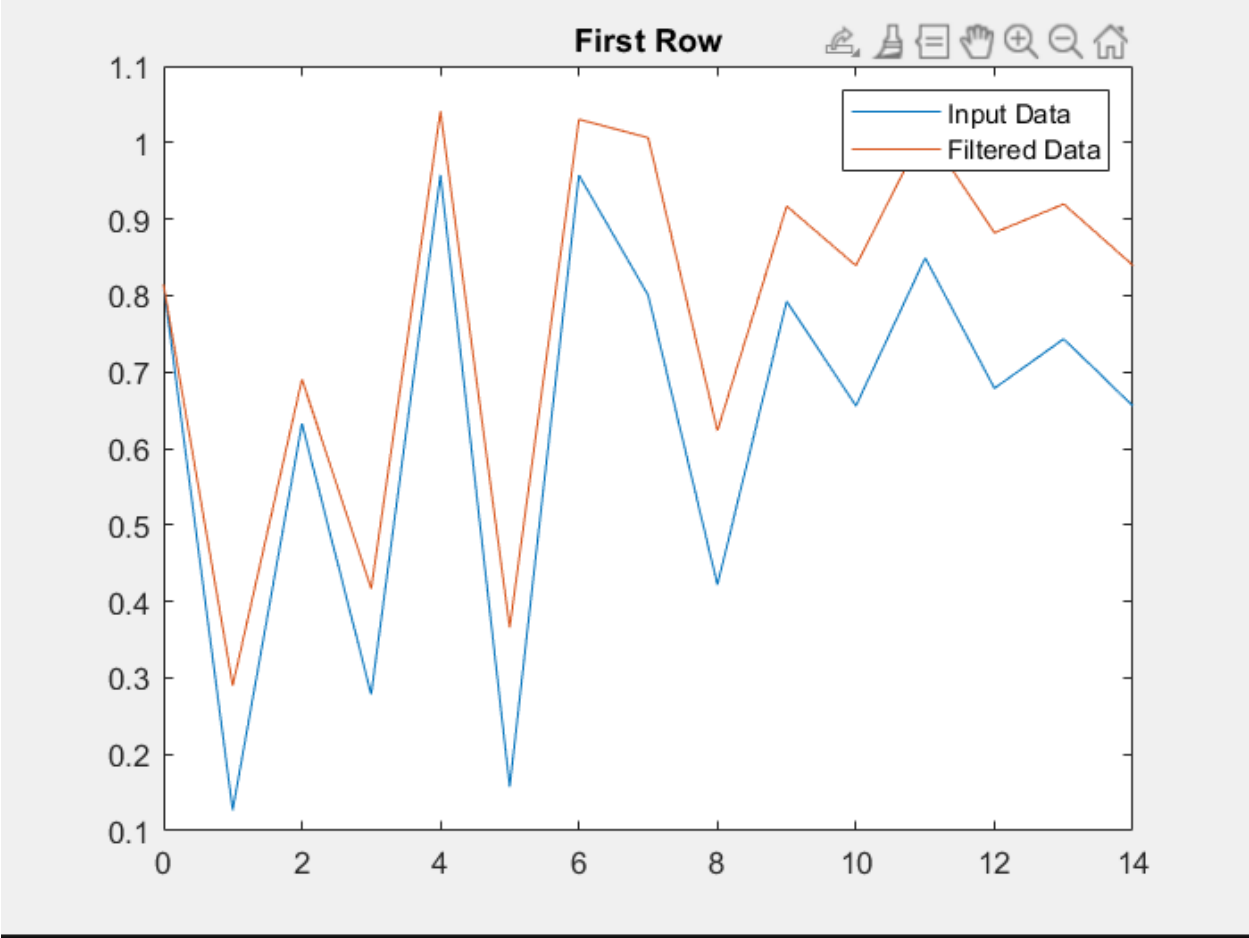
figure(3)
plot(t,x(2,:))
hold on
plot(t,y(2,:))
legend('Input Data','Filtered Data')
title('Second Row')

% Filter Data in Sections
x = randn(10000,1);

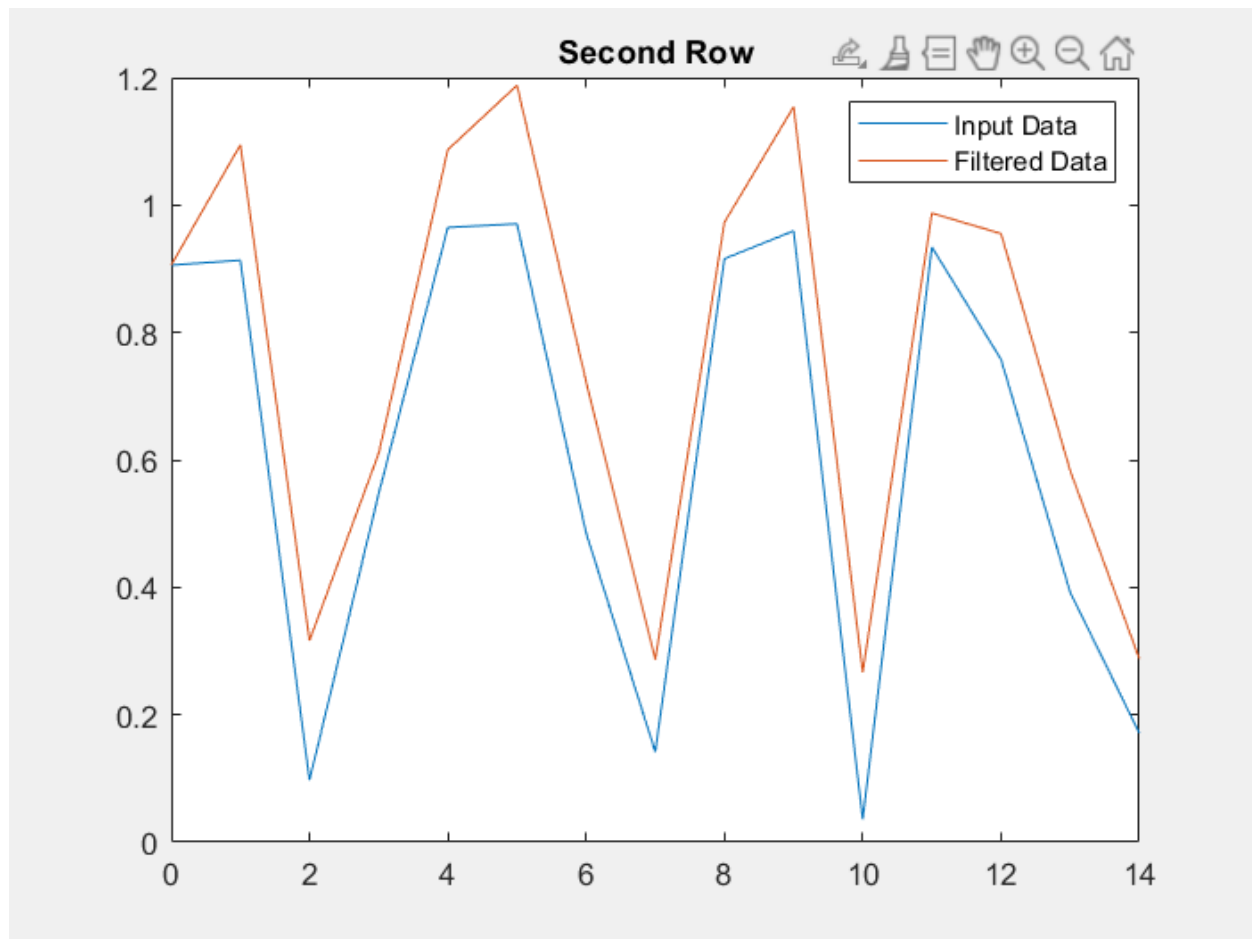
x1 = x(1:5000);
x2 = x(5001:end);
b = [2,3];
a = [1,0.2];
[y1,zf] = filter(b,a,x1);
y2 = filter(b,a,x2,zf);
y = filter(b,a,x);

isequal(y,[y1;y2])
```









```
Command Window
>> filter_ex
ans =
    logical
    1
```

## بخش سوال:

ابتدا ضرایب سیگنال های  $y, x$  تعیین شده است. سپس با استفاده از تابع `impz`، پاسخ ضربه ی سیستم علی LTI تولید شده است. تابع `impz` ضرایب دو سیگنال و محدوده را در ورودی دریافت کرده و پاسخ ضربه ی سیستم را به ما میدهد.

در نهایت با استفاده از `plot`، 40 سمپل نخست پاسخ ضربه ی سیستم علی LTI رسم شده است.

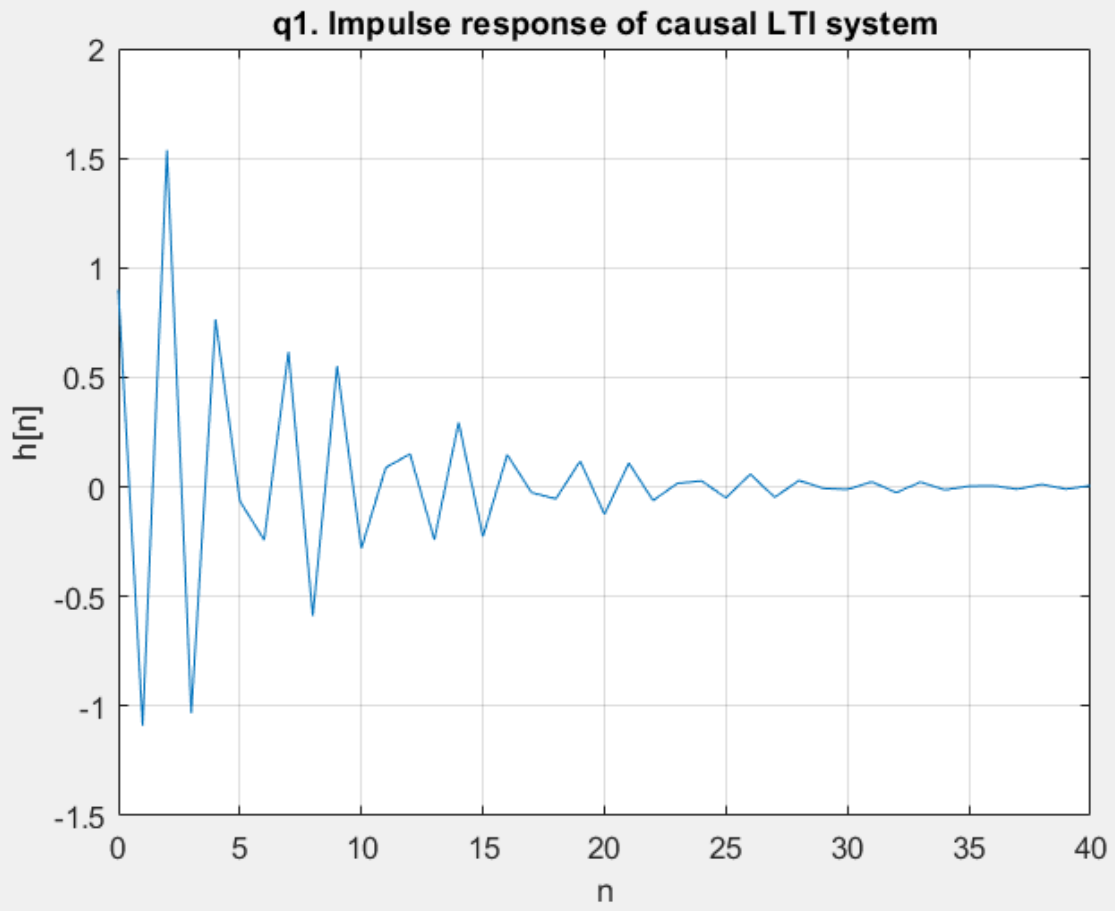
```
clc;
clear;

n = 0:40;

y = [1, 0.71, -0.46, -0.62];
% create the coefficients of signal y
x = [0.9, -0.45, 0.35, 0.002];
% create the coefficients of signal x

h = impz(x, y, n);
% create the impulse response using signals x, y and the range n

plot(n, h)
% plot the impulse response
grid on
xlabel('n');
ylabel('h[n]');
title('q1. Impulse response of causal LTI system');
```



(2)

الف) در ابتدا ضرایب سیگنال های  $x, y$  تعیین شده است. سپس سیگنال پالس مستطیلی با استفاده از تابع `stepseq` تولید شده است.

در ادامه با استفاده از تابع `filter`، تفاضل گیر روی سیگنال پالس مستطیلی پیاده سازی شده است و در نهایت نتیجه رسم شده است.

ب) سیگنال پالس سینوسی با استفاده از توابع `sin` و `stepseq` تولید شده است.

در ادامه با استفاده از تابع `filter`، تفاضل گیر روی سیگنال پالس سینوسی پیاده سازی شده است و در نهایت نتیجه رسم شده است.

\* از آنجا که بازه  $n$  در این سوال مشخص نشده بود، بازه را بین 0 تا 150 در نظر گرفتم تا برای سیگنال پالس سینوسی، مقدار صفر برای  $n$  های بزرگتر از 100 مشخص باشد.

```

clc;
clear;

n = 0:150;

y = [1];
% create the coefficients of signal y
x = [1, -1];
% create the coefficients of signal x

% first part of the question
rect_pulse = 5 .* (stepseq(0, 0, 150) - stepseq(20, 0, 150));
% create the rectangle pulse signal
x1_sig = filter(x, y, rect_pulse);
% filter the rectangle pulse signal

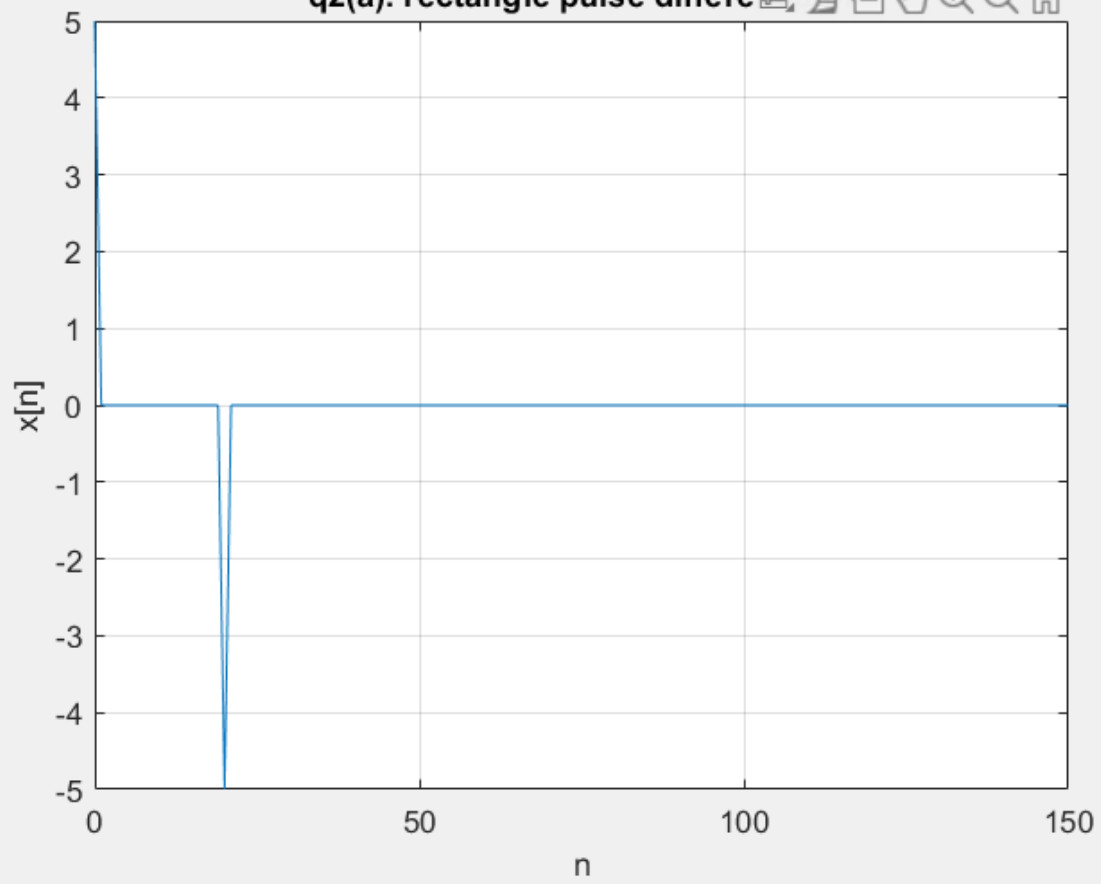
figure(1);
% let us have more than one figure
plot(n, x1_sig);
grid on
xlabel('n');
ylabel('x[n]');
title('q2(a). rectangle pulse differentiator');

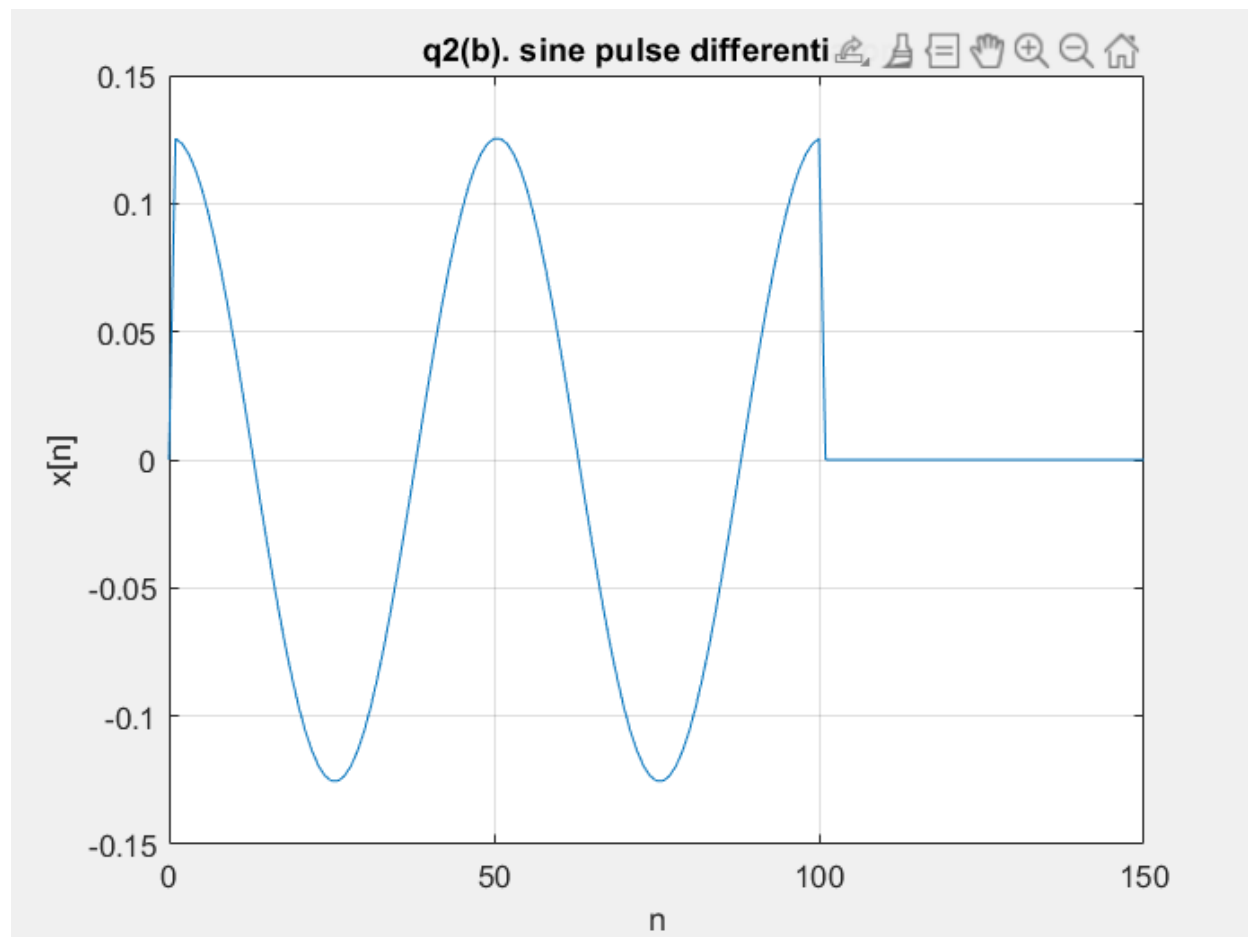
% second part of the question
sin_pulse = sin((n*pi)/25) .* (stepseq(0, 0, 150) - stepseq(100, 0, 150));
% create the sine pulse signal
x2_sig = filter(x, y, sin_pulse);
% filter the sine pulse signal

figure(2);
% let us have more than one figure
plot(n, x2_sig);
grid on
xlabel('n');
ylabel('x[n]');
title('q2(b). sine pulse differentiator');

```

q2(a). rectangle pulse differe





(3)

الف) در ابتدا بازه ی  $n$  و ضرایب سیگنال های  $x, y$  تعیین شده است.  
سپس با استفاده از تابع `impz`، پاسخ ضربه ی  $h[n]$  محاسبه و رسم شده است.

ب) با استفاده از تابع `filter`، پاسخ پله ی  $s[n]$  تولید شده است.  
در اینجا تابع `filter` ضرایب سیگنال های  $x, y$  و سیگنال پله ( $n \geq 0$ ) را در ورودی دریافت میکند و پاسخ پله را به ما میدهد.  
در نهایت پاسخ پله ی تولید شده رسم شده است.

```
clc;
clear;

n = -20:100;

y = [1, -1, 0.9];
% create the coefficients of signal y
x = [1, 0, 0];
% create the coefficients of signal x

% first part of the question
h = impz(x, y, n);
% create the impulse response using signals x , y and the range n

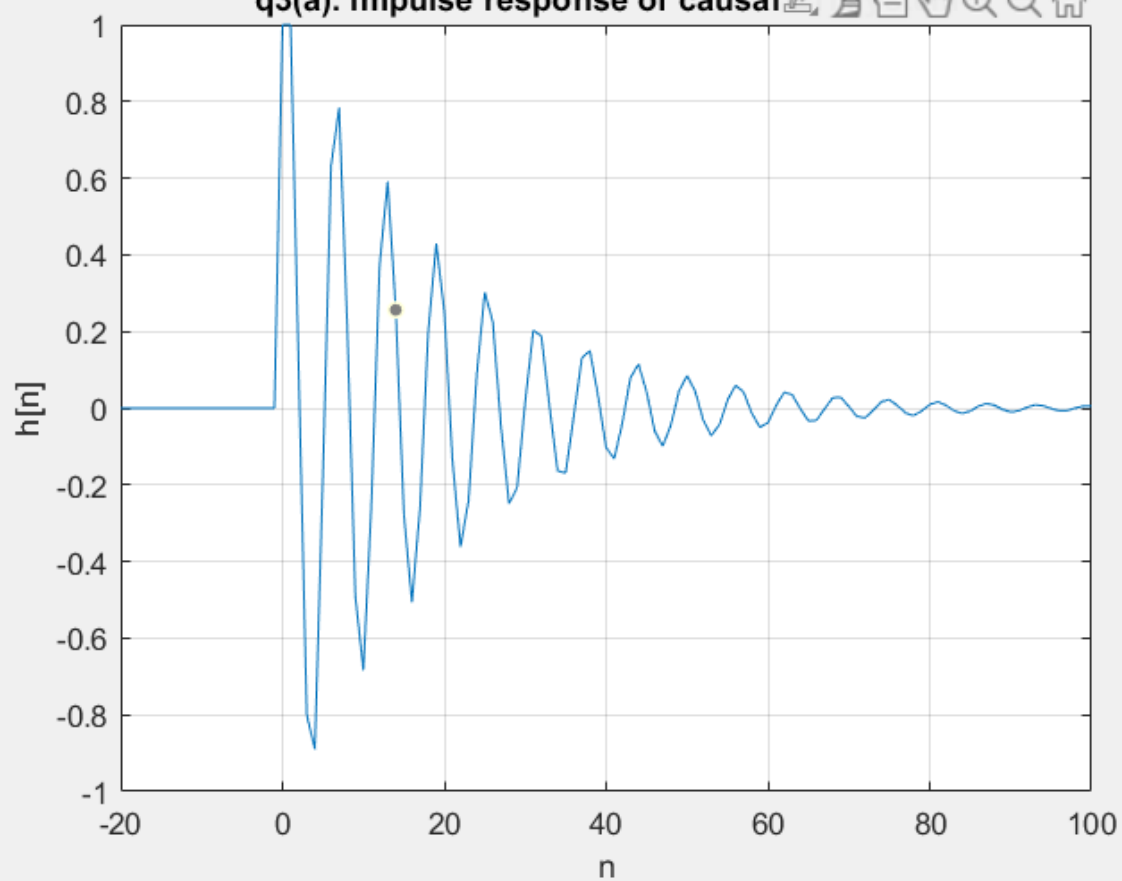
figure(1);
% let us have more than one figure
plot(n, h)
% plot the impulse response
grid on
xlabel('n');
ylabel('h[n]');
title('q3(a). Impulse response of causal LTI system');

% second part of the question
s = filter(x, y, n >= 0);
% create the step response using signals x , y and the range n

figure(2);
% let us have more than one figure
plot(n, s)
% plot the step response
grid on
xlabel('n');
ylabel('s[n]');
title('q3(b). Step response of causal LTI system');
```



q3(a). Impulse response of causal



**q3(b). Step response of causal LTI system**

