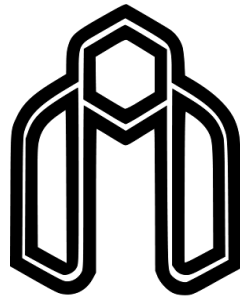


بسم الله الرحمن الرحيم



دانشگاه صنعتی شاهرود

آزمایش ۴

نام استاد : جناب دکتر مقیمی

نام دانشجو : محمد توزنده جانی

۹۷۲۰۷۸۳

(۱) بدون استفاده از توابع آماده متلب، مطابق رابطه زیر تابعی برای ایجاد تابع ضربه بنویسید. (ورودی تابع:

n_0, n_1, n_2 خروجی تابع x, n)

ضربه‌ی واحد در بازه‌ی $[n_1, n_2]$ که ضربه در نقطه‌ی $n=n_0$ رخ داده باشد،

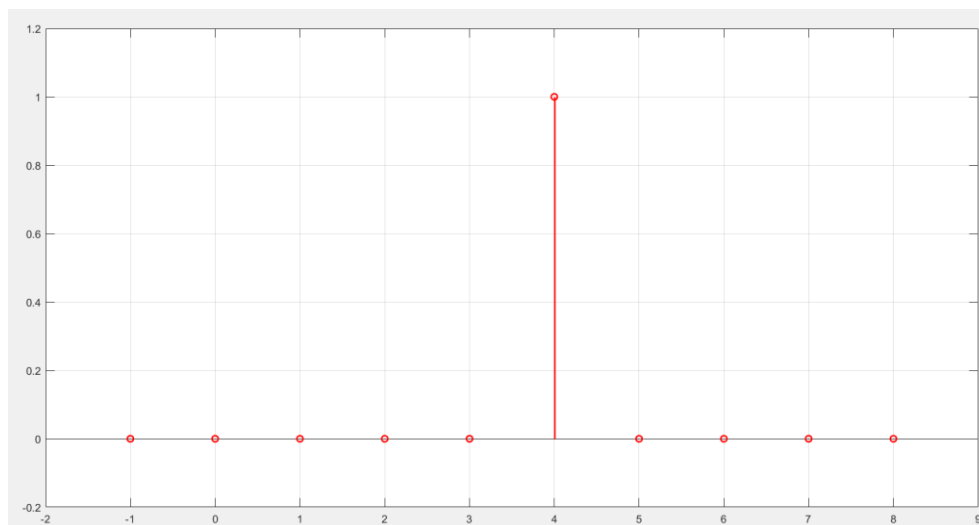
$$\delta(n - n_0) = \begin{cases} 1, & n = n_0 \\ 0, & n \neq n_0 \end{cases}$$

- کدهای متلب به صورت زیر می باشد:

```
- function [x,n]=impseq(n0,n1,n2)
-     n=n1:n2;
-     x=(n-n0)==0;           %Output(x) is Logical
-     stem(n,x,'r','linewidth',1.5)
- End
-
```

- خط $x=[(n-n_0)==0]$ ماتریس منطقی از به نام x می سازد که در تمام اعداد صحیح بین بازه ی n_1 و n_2 مقدار صفر دارد جز در $n=n_0$ که مقدار این ماتریس ۱ می باشد که برابر دامنه تابع ضربه است.

- مثال: نمایش تابع ضربه $\delta(n - 4)$ در بازه ی $[-1, 8]$



```
Command Window
>> [x,n]=impseq(4,-1,8)

x =

1x10 logical array

0 0 0 0 0 1 0 0 0 0

n =

-1 0 1 2 3 4 5 6 7 8
```

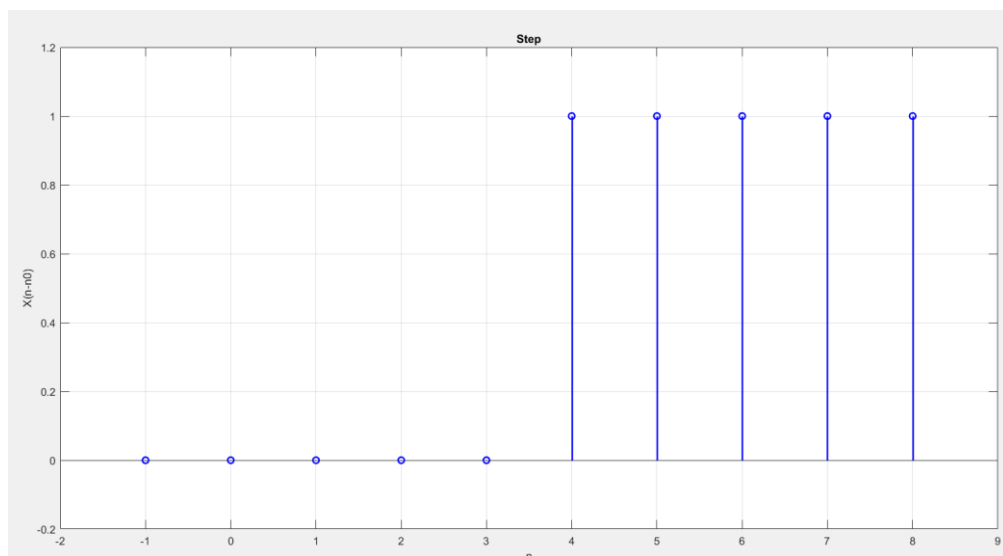
۲) مانند تمرین قبل، تابعی برای ایجاد تابع پله واحد بنویسید. (ورودی تابع: n_0, n_1, n_2 خروجی تابع x, n)

$$\Pi(n - n_0) = \begin{cases} 0 & n < n_0 \\ 1 & n \geq n_0 \end{cases}$$

- کدهای متلب به صورت زیر است:

```
- function [x,n]=step(n0,n1,n2)
-     n=n1:n2;
-     x=(n-n0)>=0;      %Output(x) is Logical
-     stem(n,x,'b','linewidth',1.5)
-     title('Step')
-     ylabel('X(n-n0)')
-     xlabel('n')
-     xlim([n1-1 n2+1])
-     ylim([-0.2 1.2])
-     grid on
- end
```

- مثال: نمایش تابع پله $\Pi(n - 4)$ در بازه $[-1, 8]$:



```
Command Window
>> [x,n]=step(4,-1,8)

x =

1×10 logical array

0  0  0  0  0  1  1  1  1  1

n =

-1  0  1  2  3  4  5  6  7  8

fx >> |
```

۳) می‌دانیم که هر سیگنالی را میتوان به صورت مجموع یک سیگنال زوج و یک سیگنال فرد توصیف نمود،

برنامه‌ای بنویسید که برای هر دنباله سیگنال ورودی، آن را به بخش‌های زوج و فرد تجزیه کند.

- هر سیگنال از یک بخش فرد و یک بخش زوج تشکیل شده است که برای بدست آوردن این بخش‌ها از دو رابطه‌ی زیر به صورت مجزا استفاده می‌شود:

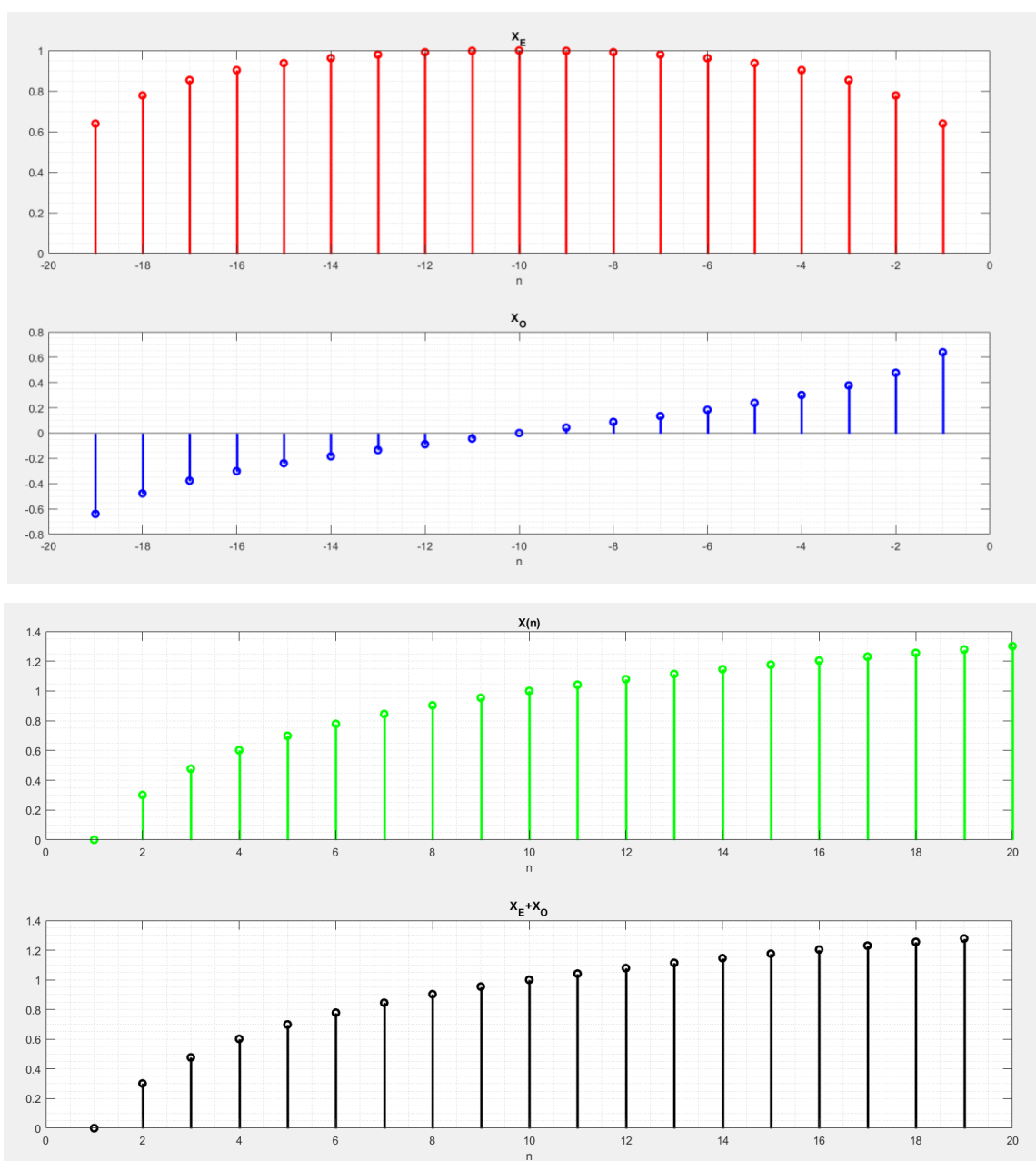
$$X_{Even}(n) = \frac{X(n) + X(-n)}{2}$$

$$X_{Odd}(n) = \frac{X(n) - X(-n)}{2}$$

- کدهای متلب به صورت زیر می‌باشد :

```
function [Xe,Xo,Sum,n1]=EvenOdd(x,n)
%-----
x1=flipplr(x);           %Create x1=x(-n)
n1=flipplr(-n);          %Create new axis
Xe=0.5*(x+x1);
Xo=0.5*(x-x1);
Sum=Xe+Xo;
%-----
figure(1)
subplot(2,1,1)
stem(n1,Xe,'r','linewidth',2.2)
title('X_E')
xlabel('n')
grid minor
subplot(2,1,2)
stem(n1,Xo,'b','linewidth',2.2)
title('X_O')
xlabel('n')
grid minor
%-----
figure(2)
subplot(2,1,1)
stem(n,x,'g','linewidth',2.2)
title('X(n)')
xlabel('n')
grid minor
subplot(2,1,2)
stem(n,Sum,'k','linewidth',2.2)
title('X_E+X_O')
xlabel('n')
grid minor
%-----
end
```

- مثال: نمایش قسمت فرد و زوج سیگنال $X(n) = \text{Log}_{10}^{(n)}$ در بازه $[0, 20]$



- همانطور که در شکل فوق مشخص است حاصل جمع دو سیگنال فرد و زوج برابر سیگنال اصلی است بنابراین تابع نوشته شده درست عمل می نماید.

```
Command Window
Editor - EvenOdd.m

>> n=[0:20];
>> x=log10(n);
>> [Xe,Xo,Sum,n1]=EvenOdd(x,n)

Xe =

Columns 1 through 9

    -Inf    0.6394    0.7782    0.8538    0.9031    0.9375    0.9621    0.9795    0.9911

Columns 10 through 18

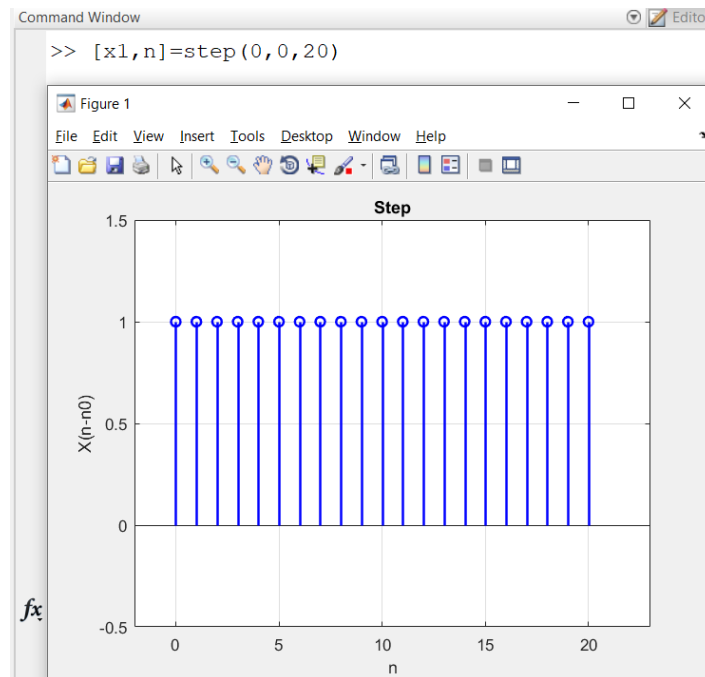
    0.9978    1.0000    0.9978    0.9911    0.9795    0.9621    0.9375    0.9031    0.8538

Columns 19 through 21

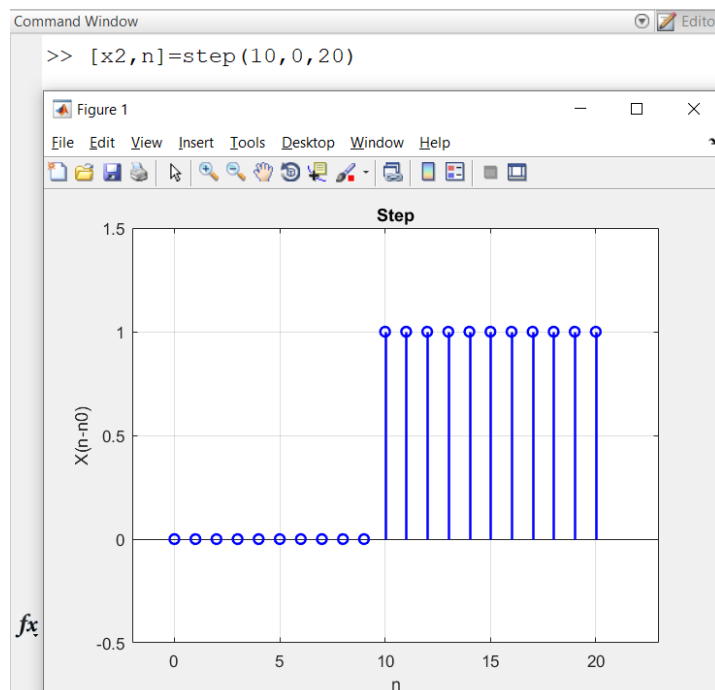
    0.7782    0.6394    -Inf
```

۴) با استفاده از توابع مراحل قبل سیگنال $X[n] = U[n] - U[n-10]$ را به بخش‌های زوج و فرد تجزیه کنید.

- سیگنال $U[n]$ را از بازه ۰ تا ۲۰ ایجاد می‌کنیم :

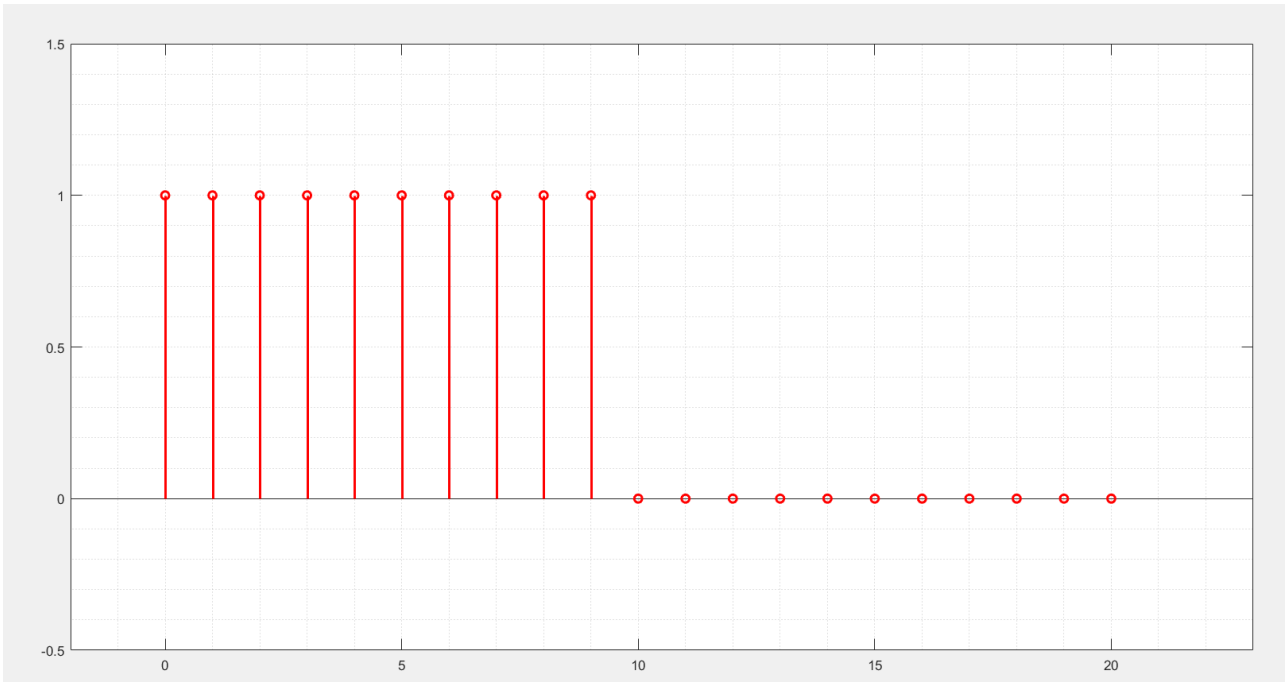


- سیگنال $U[n-10]$ را از بازه ۰ تا ۲۰ ایجاد می‌کنیم :



- حال سیگنال $X_3(n) = U[n] - U[n - 10]$ را با تفریق دو سیگنال فوق ایجاد کرده و با دستور stem رسم می کنیم :

```
Command Window Editor - step.m
>> x3=x1-x2;
>> n=[0:20];
>> stem(n,x3,'r','linewidth',1.8);grid minor;ylim([-0.5 1.5]);xlim([-2 23])
fx >>
```



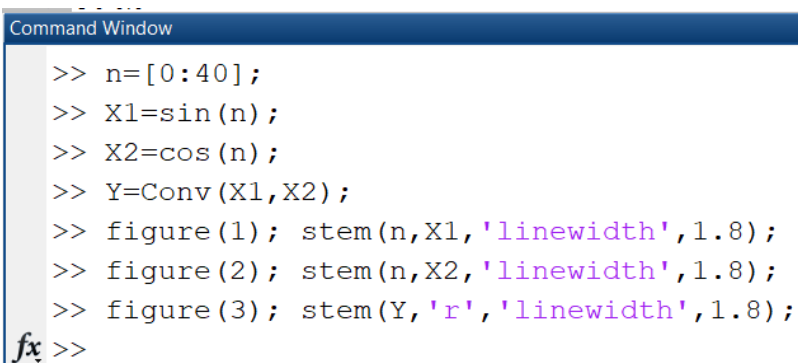
۵) در متلب برای عملیات کانولوشن می‌توانید از دستور conv استفاده کنید. بدون استفاده از این تابع، تابعی بنویسید که مشابه آن عمل کند و عملکرد تابع خود را با آن با دادن ورودی و رسم خروجی مقایسه کنید.

```
function Y = Conv(X,H)

z = [];
for i = 1:length(X)
    g = H.*X(i);
    z = [z;g];
end

[r c]=size(z);
k=r+c;
t=2;
Y=[];
cd=0;

while (t<=k)
    for i=1:r
        for j=1:c
            if ((i+j)==t)
                cd = cd + z(i,j);
            end
        end
    end
    t=t+1;
    Y=[Y cd];
    cd=0;
end
end
```



```
>> n=[0:40];
>> X1=sin(n);
>> X2=cos(n);
>> Y=Conv(X1,X2);
>> figure(1); stem(n,X1,'linewidth',1.8);
>> figure(2); stem(n,X2,'linewidth',1.8);
>> figure(3); stem(Y,'r','linewidth',1.8);
fx >>
```