به نام خدا

دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر

تجزیه و تحلیل سیگنال‌ها و سیستم‌ها

تمرین کامپیوتری شماره 1

نام و نام خانوادگی: علیرضا جابری راد
شماره دانشجویی: 810196438

دی ماه 1398

فهرست

چکیده

In this computer assignment we are going to analyze and verify some of the theories which were tought about Fourier series and sampling. But the main purpose of this CA is to get our hands dirty with coding in Matlab environment.

سوال 1

MATLAB Drive > exercise1.m

```
1 function [A,D,r] = exercise1(n,m,q)
2     matrix=repmat([1:n],m-n+1,1);
3     plus=[0:m-n];
4     plus=plus';
5     plus=repmat(plus,1,n);
6     A=matrix+plus
7     r=sort( randperm(m,randi(m)) )
8     D=ismember(A,r);
9     D=D-0.5;
10    D=2*D
11 end
```

1 **Figure**

Line 2: “repmat” will repeat the row vector [1 to n] horizontally m-n+1 times.

Line 4: in this line “plus” will be a column vector which contains [0 to m-n].

Line 5: “repmat” will repeat the “plus” vector “n” times vertically.

Line 7: “randi” will first generate an integer number between [1 to m], then “randperm” will generate an array containing “randi(m)” number of non-repeating integers between [1 to m]. “sort” is not doing anything special. I just used it to make everything clean in the output.

Line 8: “ismember” will return a boolean matrix which indicates whether an element in “A” is in the set “r” or not.

Line 9,10: after running line 8, D is a boolean matrix that is made by zeros and ones. So, I subtracted it with 0.5 then multiplied it by 2 to get a matrix of -1 and 1s.

```
>> exercise1(5,20,4)

A =

     1     2     3     4     5
     2     3     4     5     6
     3     4     5     6     7
     4     5     6     7     8
     5     6     7     8     9
     6     7     8     9    10
     7     8     9    10    11
     8     9    10    11    12
     9    10    11    12    13
    10    11    12    13    14
    11    12    13    14    15
    12    13    14    15    16
    13    14    15    16    17
    14    15    16    17    18
    15    16    17    18    19
    16    17    18    19    20


r =

     1     3     4     5     6     7     8     9    10    13    14    15    18


D =

     1    -1     1     1     1
    -1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1    -1
     1     1     1    -1    -1
     1     1    -1    -1     1
     1    -1    -1     1     1
    -1    -1     1     1     1
    -1     1     1     1    -1
     1     1     1    -1    -1
     1     1    -1    -1     1
     1    -1    -1     1    -1
    -1    -1     1    -1    -1
```

Figure 2

Here you can see the final result after running the implemented function for exercise 1.

سوال 2

```
exercise1.m × | exercise2.m × | exercise3.m × | +
1   function exercise2()
2 -     x=[zeros(1,8) (1/2).^(0:2) zeros(1,6)];
3 -     n=-24:24;
4 -     h1=zeros(1,17);
5 -     h1(7)=1;
6 -     h1(8:9)=-1;
7 -     h1(10)=1;
8 -     a=zeros(1,8);
9 -     b=ones(1,2);
10 -    c=2*ones(1,7);
11 -    h2=[a b c].*[a 1*(1/3).^(0:8)];
12 -    h=conv(h1,h2);
13 -    firstOutput=conv(conv(x,h1),h2);
14 -    secondOutput=conv(conv(x,h2),h1);
15 -    thirdOutput=conv(x,h);
16 -    subplot(3,1,1);stem(n,firstOutput);
17 -    subplot(3,1,2);stem(n,secondOutput);
18 -    subplot(3,1,3);stem(n,thirdOutput);
19 - end
```

Figure 3

Line 2: “x” is a row vector. First 8 elements of it is set to be zero, 3 of the following ones are (0.5)^0 & (0.5)^1 & (0.5)^2 and the rest of the elements are zero. The resultant “x” is identical to the given x[n] signal limited by the interval [-8,8[.

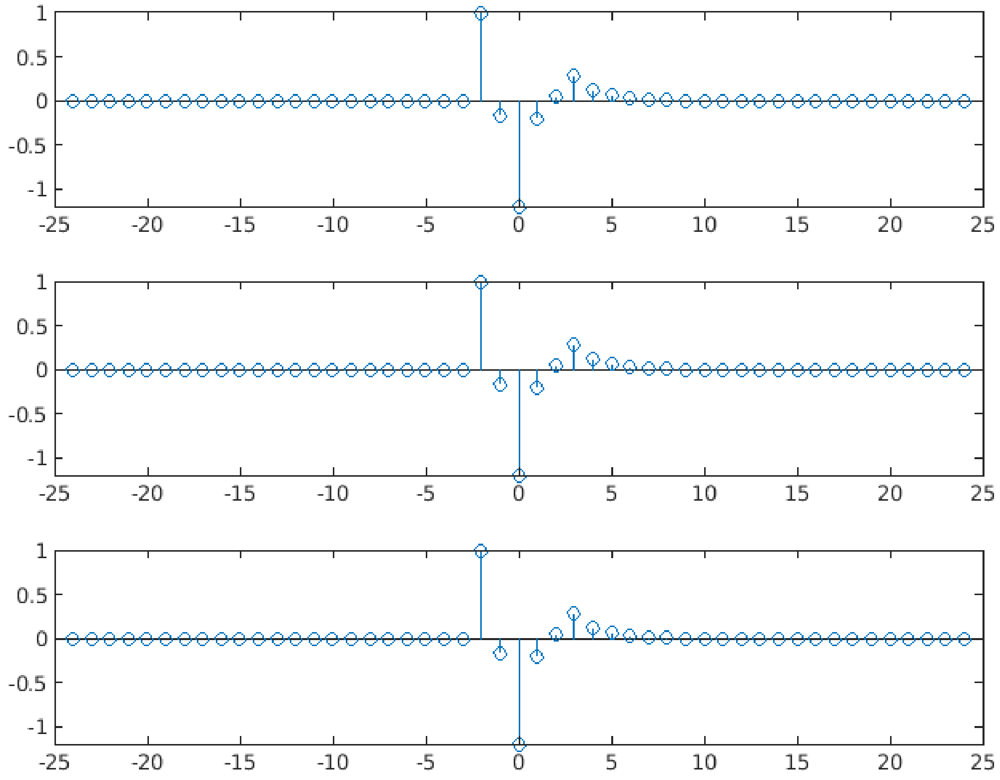
Line 4-7: I implemented the h1 signal which its domain is limited by [-8,8], by inserting 1 or -1 to the points that the delta function has a nonzero magnitude. “h1” is identical to the “h1” which is given in the exercise definition.

Line 11: [a b c] is a vector which constructs the “(u[n]+u[n-2])”. So I multiplied it elementwise by the vector [8 times zero & (1/3)^n ; n is from 0 to 8] to get h2 as the result.

Line 13: x is first convolved with h1 then the resultant output is convolved with h2.

Line 14: x is first convolved with h2 then the resultant output is convolved with h1.

Line 15: x is directly convolved with the cascaded system h1*h2



4 **Figure**

The plot that is on the top of the figure refers to the line 13 of the code.

Second one refers to the line 14 of the code.

Third one refers to the line 15 of the code.

As we expected the result of all of them are the same because of the properties that we know about convolution function.

P.S: We know h1*h2 is the same as h2*h1.

سوال 3

```
1   function [] = exercise3( )
2   t=-2:0.1:2;
3   subplot(2,1,1);plot(t,sin(2*pi*t));
4   subplot(2,1,2);stem(t,sin(2*pi*t));
5   figure;
6   subplot(2,1,1);plot(t,heaviside(t)-heaviside(t-1));
7   subplot(2,1,2);stem(t,heaviside(t)-heaviside(t-1));
8   figure;
9   subplot(2,1,1);plot(t,2*t.*heaviside(t)-(t-1).*heaviside(t-1)+heaviside(t+1));
10  subplot(2,1,2);stem(t,2*t.*heaviside(t)-(t-1).*heaviside(t-1)+heaviside(t+1));
```

Figure 5

General definition: “plot” connects two consecutive points of the given signal by a straight line. But “stem” shows the discrete signal (sampled signal) as the usual way we do with sketching discrete time signals on paper.

Line 3-4: will print the sampled sinusoid function as wanted.

Line 6-7: will print the sampled function of u(t) - u(t-1) as wanted.

Line 9-10: will print the sampled function of 2r(t) - r(t-1) + u(t+1) as wanted.

```
11  syms x t k T K X;
12  fun1=@(x)(sin(2*pi*x))^2;
13  energy=limit(int(fun1,x,-X,X),X,inf);
14  power=limit((int(fun1,x,-X/2,X/2))/X,X,inf);
15  disp("a.energy:");
16  disp(energy);
17  disp(" power:");
18  disp(power);
19  fun2=@(t)(heaviside(t)-heaviside(t-1))^2;
20  energy=limit(int(fun2,t,-T,T),T,inf);
21  power=limit((int(fun2,t,-T/2,T/2))/T,T,inf);
22  disp("b.energy:");
23  disp(energy);
24  disp(" power:");
25  disp(power);
26  fun3=@(k)(2*k.*heaviside(k)-(k-1).*heaviside(k-1)+heaviside(k+1))^2;
27  energy=limit(int(fun3,k,-K,K),K,inf);
28  power=limit((int(fun3,k,-K/2,K/2))/K,K,inf);
29  disp("c.energy:");
30  disp(energy);
31  disp(" power:");
32  disp(power);
33
34  end
```

Figure 6

Line 12-18: will execute and show the energy and power signal of sinusoid. (part a)

Line 19-25: will execute and show the energy and power signal of u(t) - u(t-1). (part b)

Line 26-32: will execute and show the energy and power signal of 2r(t) - r(t-1) + u(t+1). (part c)

Part a:

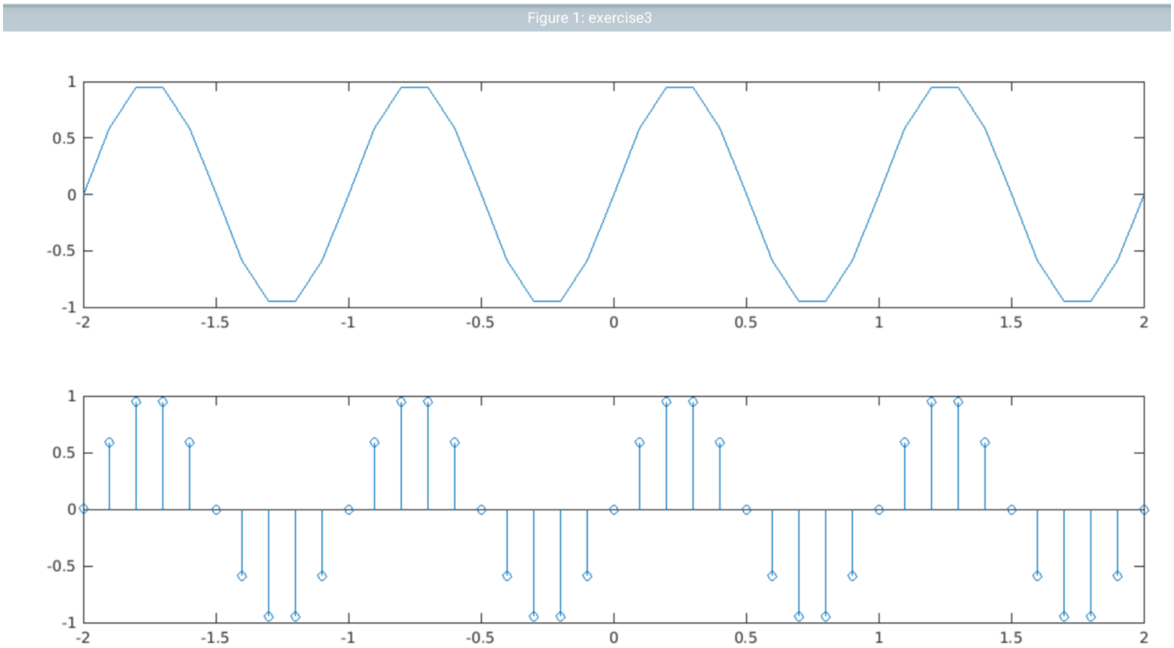


Figure 7:Sin(t)

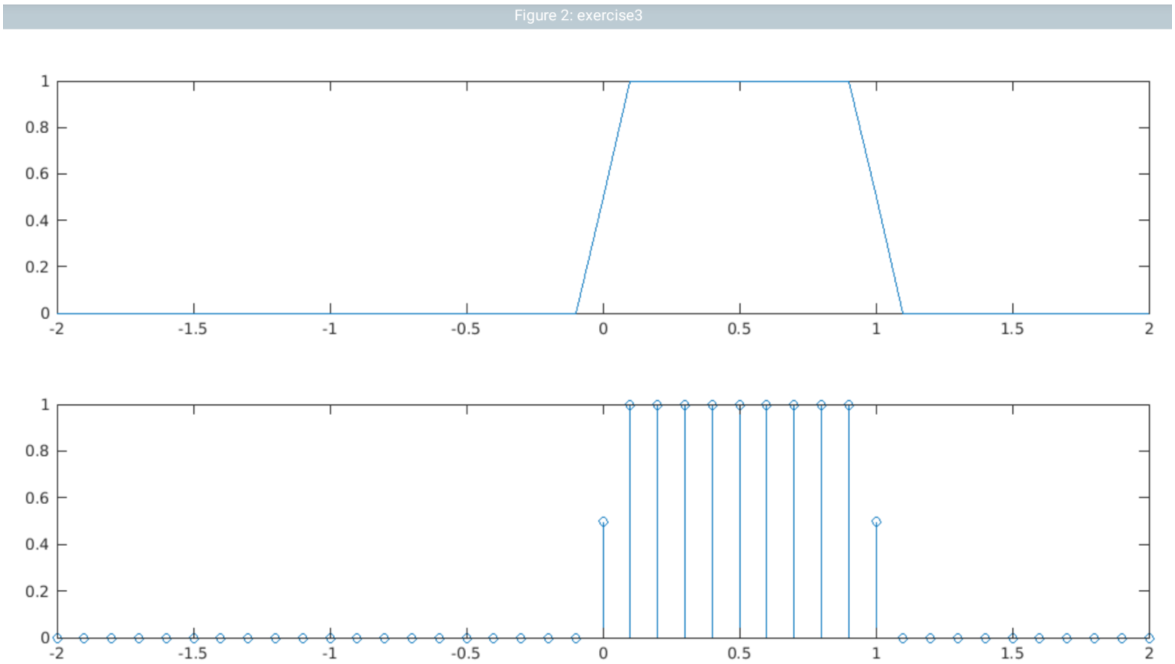


Figure 8: u(t) - u(t-1)

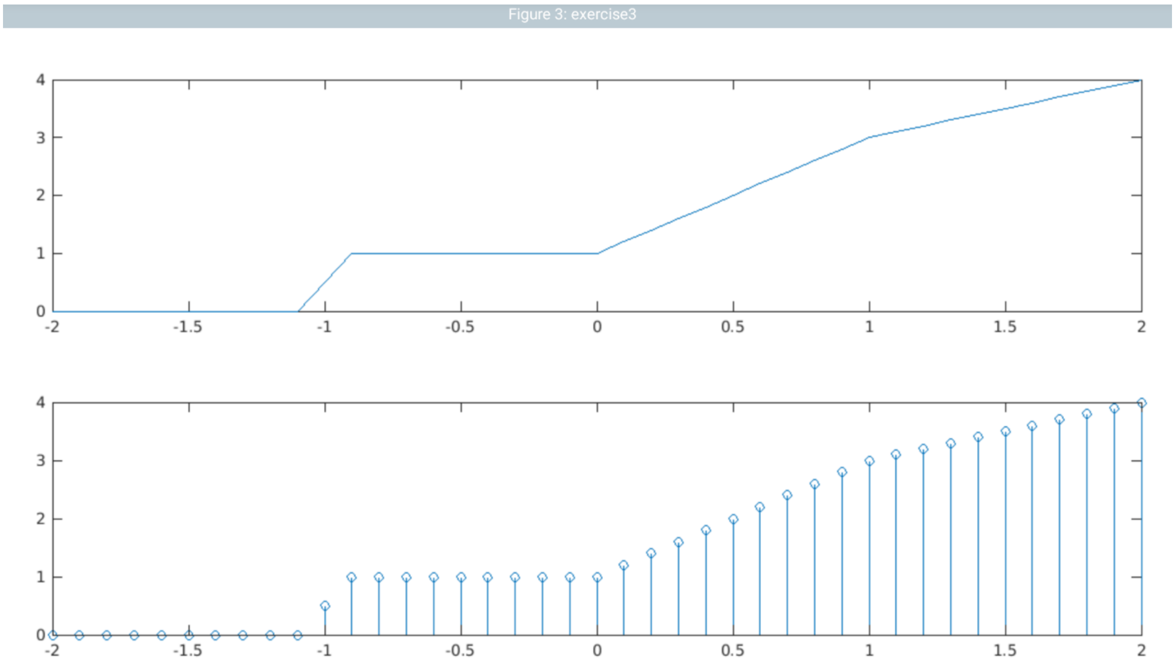


Figure 9: 2r(t) - r(t-1) + u(t+1)

Part b:

```
>> exercise3
a.energy:
Inf

 power:
1/2

b.energy:
1

 power:
0

c.energy:
Inf

 power:
Inf
```

Figure 10

We know if the energy signal of a signal is finite then the power signal of it will be zero. For $x_2(t)$ we have this case and the output regarding to this case is verifying this fact(b).

Also we know sinusoid is a repetitive function and the power in one period is equal to the total power signal of it and is equal to 1/2.(all of the repetitive functions have infinite energy signal. As you can see this fact is valid for sinusoid).

We can easily calculate and see that the power signal of a linear function is infinite. Also we can verify this fact for t.u(t)=r(t). you can see the matlab computation for this case in part c.

پیوست 1: روند اجرای برنامه

All of the codes have been implemented as function.

Exercise 1: you can just call it by 3 input values mentioned in the definition of the exercise 1.

Exercise 2: doesn’t need any input value and it’s enough to call it in command window.

Exercise 3: doesn’t need any input value and it’s enough to call it in command window.