**In-lab Tasks**

**Task 01: Plot the signal  for , where is a rectangular pulse of duration T, denoted by .**

|  |
| --- |
| t=[-2:0.1:5];  u=(t>=0);  u1=(t-2>=0);  p=u-u1;  x=t.^3.\*cos(10\*pi.\*t).\*p;  plot(t,x) |

**Task 02: Suppose that  Plot the signals**

1. ****
2. 
3. 
4. 
5. 

|  |
| --- |
| **Part a:**  t=[0:0.1:5];  x=t.\*cos(2\*pi.\*t);  plot(t,x)    **Part b:**  t=[0:0.1:5];  x=t.\*cos(2\*pi.\*t);  plot(-t,x)    **Part c:**  t=[0:0.1:5];  x=t.\*cos(2\*pi.\*t);  plot(5.\*t,x)    **Part d:**  t=[0:0.1:5];  x=t.\*cos(2\*pi.\*t);  t=t-1;  t=t./3;  plot(t,x)    **Part e:**  t=[0:0.1:5];  x=t.\*cos(2\*pi.\*t);  t=t+1;  t=t./3;  plot(-t,x) |

**Task 03: Suppose that . Plot the signals**

1. 
2. 
3. 
4. 
5. 

|  |
| --- |
| **Part f:**  t=[0:0.01:2];  t1=[2.01:0.01:4];  x=t;  x1=4-t1;  x2=[x x1];  t=[t t1];  plot(t,x2)    **Part g:**  t=[0:0.01:2];  t1=[2.01:0.01:4];  x=t;  x1=4-t1;  x2=[x x1];  t=[t t1];  plot(-t,x2)    **Part h:**  t=[0:0.01:2];  t1=[2.01:0.01:4];  x=t;  x1=4-t1;  x2=[x x1];  t=[t t1];  t=2.\*t;  plot(t,x2)    **Part i:**  t=[0:0.01:2];  t1=[2.01:0.01:4];  x=t;  x1=4-t1;  x2=[x x1];  t=[t t1];  t=t-2;  t=t./4;  plot(t,x2)    **Part J:**  t=[0:0.01:2];  t1=[2.01:0.01:4];  x=t;  x1=4-t1;  x2=[x x1];  t=[t t1];  t=t-2;  t=t./4;  plot(-t,x2) |

**Task 04: Write a function that accepts a sequence, the discrete time n and a number n0, a, b as input arguments, and returns the signals and. Whererepresents the time compressed version of  andis the time expanded version of **.

|  |
| --- |
| function [ X,n ] = lab4task4( x,N,n0,a,b )  %LAB4TASK4 Summary of this function goes here  % Detailed explanation goes here  n=[0:1:N];  %code starts from here  L=length(x);  if n0==0 && a==0 && b==0  X=fliplr(x);  stem(n,X);  title('x[-n]');  elseif n0~=0 && a==0 && b==0  n0=-n0;  z=zeros(1,abs(n0));  if n0>0  X=[x z];  X(1:abs(n0))=[];  stem(n,X);  title('x[n-n0]')  else  X=[z x];  L1=length(X);  X(L+1:L1)=[];  stem(n,X);  title('x[n-n0]');  end    elseif n0==0 && a==0 && b~=0  X=upsample(x,b);  L1=length(X);  L3=L1-L;  X(1:L3/2)=[];  L4=length(X);  X(L+1:L4)=[];  stem(n,X);  title('x[ba]');  elseif n0==0 && a~=0 && b==0  X=downsample(x,2);  L1=length(X);  L3=L-L1;  Z1=zeros(1,L3/2);  X=[Z1 X Z1];  stem(n,X);  title('x[ab]');  end  end  lab4task4([1 1 0 1 0 1],5,0,0,0)    lab4task4([1 1 0 1 0 1],5,2,0,0)  lab4task4([1 1 0 1],3,0,2,0)    lab4task4([1 1 0 1],3,0,0,2) |

**Post-lab Task**

**Critical Analysis / Conclusion**

|  |
| --- |
| In this lab we performed signal shifting, scaling and reversal. If a signal has both shifting, scaling and reversal properties we have to perform them in a specific order otherwise our output signal will be wrong. The order is as follow   1. Shifting 2. Scaling 3. Reversal   If we perform reversal first then we must perform scaling in reverse order. |