
```

%ECG680 - DSP Assignment 1
%Author - Cho Minsung

% code 1
nx = 0:11
x = [0 1 2 3 2 1 0 -1 -2 -3 -2 -1]

% 1_a_i
ns1 = (0):(11)
s1 = x

figure(1)
subplot(2,2,1)
stem(nx,x)
axis([-15,15, -4, 4])
grid on
title('x[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

subplot(2,2,2)
stem(ns1,s1)
axis([-15,15, -4, 4])
grid on
title('s1[n] = x[n]', 'fontsize', 12)
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 1_a_ii
ns2 = 3:14
s2 = x

subplot(2,2,3)
stem(ns2,s2)
axis([-15,15, -4, 4])
grid on
title('s2[n] = x[n-3]', 'fontsize', 12)
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 1_a_iii
ns3 = -5:6
s3 = x

subplot(2,2,4)
stem(ns3,s3)
axis([-15,15, -4, 4])
grid on
title('s3[n] = x[n+5]', 'fontsize', 12)
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

```

```

% 1_a_shift
[s1, n1] = shift(x,0,nx)
[s2, n2] = shift(x,3,nx)
[s3, n3] = shift(x,-5,nx)

figure(2)
plot(n1,s1,'*',n2,s2,'x',n3,s3,'o')
axis([-15,15, -4, 4])
grid on
title('s1[n], s2[n], & s3[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 1_b_i
figure(3)
subplot(3,2,1)
stem(nx,x)
axis([-15,15, -4, 4])
grid on
title('x[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

subplot(3,2,2)
stem(-nx,x)
axis([-15,15, -4, 4])
grid on
title('x[-n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 1_b_ii
nv = 4:15
v = x

subplot(3,2,3)
stem(nv,v)
axis([-15,15, -4, 4])
grid on
title('v[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

subplot(3,2,4)
stem(-nv,v)
axis([-15,15, -4, 4])
grid on
title('v[-n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 1_b_iii

```

```

nr = -3:8
r = x

subplot(3,2,5)
stem(nr,r)
axis([-15,15, -4, 4])
grid on
title('r[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

subplot(3,2,6)
stem(-nr,r)
axis([-15,15, -4, 4])
grid on
title('r[-n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

%1_b_reverse_function
[r1, k1] = reverse(x,nx)
[r2, k2] = reverse(v,nv)
[r3, k3] = reverse(r,nr)

figure(4)
plot(k1,r1,'*',k2,r2,'x',k3,r3,'o')
axis([-15,15, -4, 4])
grid on
title('x[-n], v[-n], & r[-n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

clear

% 2_a
nx = -100:100;
us1 = zeros;
us2 = zeros;
x = zeros;
us1(nx>=2) = 1;
us2(nx>=12) = 1;
x = us1 - us2;

figure(5)
subplot(6,1,1);
stem(nx,x)
axis([-20,20, -2, 2])
grid on
title('x[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 2_b
us1 = zeros;

```

```

us2 = zeros;
v = zeros;
us1(nx>=-16) = 1;
us2(nx>=7) = 1;
v = us1 - us2;

subplot(6,1,2);
stem(nx,v)
axis([-20,20, -2, 2])
grid on
title('v[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 2_c
y = zeros;
y = x.*v;

subplot(6,1,3);
stem(nx,y)
axis([-20,20, -2, 2])
grid on
title('y[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 2_d
r = zeros;
r = x+v;

subplot(6,1,4);
stem(nx,r)
axis([-20,20, -2, 2])
grid on
title('r[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 2_e
x1 = zeros;
nt1 = -98:102;
us1 = zeros;
us2 = zeros;
us1(nt1>=2) = 1;
us2(nt1>=12) = 1;
x1 = us1 - us2;
nt2 = -102:98;
us1 = zeros;
us2 = zeros;
v1 = zeros;
us1(nt2>=-16) = 1;
us2(nt2>=7) = 1;
v1 = us1 - us2;
s = x1.*v1;

```

```

subplot(6,1,5);
stem(nx,s)
axis([-20,20, -2, 2])
grid on
title('s[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 2_f
y1 = zeros;
ny1 = -101:99;
us1 = zeros;
us2 = zeros;
us1(ny1>=2) = 1;
us2(ny1>=12) = 1;
x2 = us1 - us2;
us1 = zeros;
us2 = zeros;
v2 = zeros;
us1(ny1>=-16) = 1;
us2(ny1>=7) = 1;
v2 = us1 - us2;
y1 = x2.*v2;;

ny2 = -99:101;
y2 = zeros;
x3 = zeros;
v3 = zeros;
us1 = zeros;
us2 = zeros;
us1(ny2>=2) = 1;
us2(ny2>=12) = 1;
x3 = us1 - us2;
us1 = zeros;
us2 = zeros;
v3 = zeros;
us1(ny2>=-16) = 1;
us2(ny2>=7) = 1;
v3 = us1 - us2;
y2 = x3.*v3;

t = y1 + y2;
subplot(6,1,6);
stem(nx,t)
axis([-20,20, -2, 2])
grid on
title('t[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

clear

% 3_i

```

```

n = 0:30;
impulse = zeros(size(n));
impulse(1) = 1;

% 3_ii
step = zeros;
step(n>=0) = 1;
ramp = n.*step;

% 3_iii
exp = (5/6).^n.*step;

% 3_a
[r, nr] = shift (ramp, 6, n);
v = r.*step;

figure(6)
subplot(5,1,1);
stem(nr,v)
axis([-7,23, -1, 20])
grid on
title('v[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 3_b
[sigma, nsigma] = shift (impulse, 15, n);
[sigma1, v1, nb] = compsig(sigma, nsigma, v, nr);
x = sigma1.*v1;

subplot(5,1,2);
stem(nb,x)
axis([0,30, -1, 20])
grid on
title('x[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 3_c
[e, ne] = shift(exp, -10, n);
[e1, step1, nc] = compsig(e, ne, step, n);
y = e1.*step1;

subplot(5,1,3);
stem(nc,y)
axis([-30,59, -0.1, 0.2])
grid on
title('y[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 3_d
[u1, nul] = shift(step, 10, n);
[u1_1, step1_1, nvar] = compsig(u1, nul, step, n);

```

```

var = step1_1 - ul_1;
[var1, e2, nd] = compsig(var, nvar, exp, n);
z = e2.*var1;

subplot(5,1,4);
stem(nd,z)
axis([-30,59, -0.1, 1])
grid on
title('y[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

% 3_e
ne = 0:2:60;
expl = (5/6).^ne.*step;
[exp2, nc] = shift (expl, 5, ne);
c = exp2;

subplot(5,1,5);
stem(nc,c)
axis([-5,61, -0.1, 1])
grid on
title('y[n]')
xlabel('n', 'fontsize', 12)
ylabel('s[n]', 'fontsize', 12)

%shift function
function [s, n] = shift (x, N, nx)
s = x;
n = nx + N;
end

%reverse function
function [r, k] = reverse (x, nx)
r = x;
k = -nx;
end

% compsig function
function [s1, s2, n] = compsig(x1, n1, x2, n2)
    nmin = min([min(n1), min(n2)]); %determines the minimum time
    index.
    nmax = max([max(n1), max(n2)]); %determines the maximum time
    index.
    n = nmin:nmax; %time indices are set.

    nsize = size(n, 2); %size to be
    [s1, s2] = deal(zeros(1, nsize)); %s1 and s2 are initialized
    with size n.

    x1size = size(x1,2); %determines the size of x1.
    x2size = size(x2,2); %determines the size of x2.
    x1first = find(n == n1(1)); %finds the
    x2first = find(n == n2(1));

```

```

    switch (n1(1) < n1(2))           %as long as the index is
incremental                          %the right index for x1 is
    case true                          %transferred.
        s1(x1first : x1first + x1size - 1) = x1;
    otherwise                          %this is when the order is
flipped.
        s1(x1first - x1size + 1 : x1first) = fliplr(x1);
    end

    switch (n2(1) < n2(2))
    case true
        s2(x2first : x2first + x2size - 1) = x2;
    otherwise
        s2(x2first - x2size + 1 : x2first) = fliplr(x2);
    end
end

```

```
nx =
```

```

    0    1    2    3    4    5    6    7    8    9   10
11

```

```
x =
```

```

    0    1    2    3    2    1    0   -1   -2   -3   -2
-1

```

```
ns1 =
```

```

    0    1    2    3    4    5    6    7    8    9   10
11

```

```
s1 =
```

```

    0    1    2    3    2    1    0   -1   -2   -3   -2
-1

```

```
ns2 =
```

```

    3    4    5    6    7    8    9   10   11   12   13
14

```

```
s2 =
```

```

    0    1    2    3    2    1    0   -1   -2   -3   -2
-1

```

```

ns3 =
  -5  -4  -3  -2  -1  0  1  2  3  4  5
  6

s3 =
  0  1  2  3  2  1  0  -1  -2  -3  -2
 -1

s1 =
  0  1  2  3  2  1  0  -1  -2  -3  -2
 -1

n1 =
  0  1  2  3  4  5  6  7  8  9  10
 11

s2 =
  0  1  2  3  2  1  0  -1  -2  -3  -2
 -1

n2 =
  3  4  5  6  7  8  9  10  11  12  13
 14

s3 =
  0  1  2  3  2  1  0  -1  -2  -3  -2
 -1

n3 =
  -5  -4  -3  -2  -1  0  1  2  3  4  5
  6

nv =
  4  5  6  7  8  9  10  11  12  13  14
 15

```

```

v =
  0   1   2   3   2   1   0  -1  -2  -3  -2
-1

nr =
 -3  -2  -1   0   1   2   3   4   5   6   7
  8

r =
  0   1   2   3   2   1   0  -1  -2  -3  -2
-1

r1 =
  0   1   2   3   2   1   0  -1  -2  -3  -2
-1

k1 =
  0  -1  -2  -3  -4  -5  -6  -7  -8  -9 -10
-11

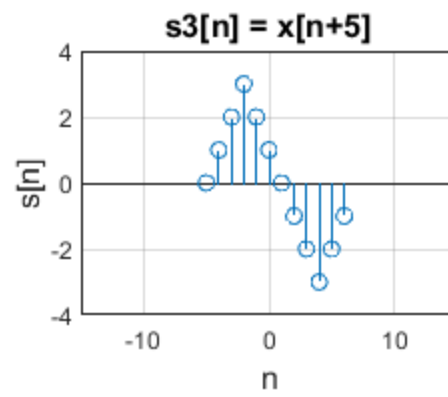
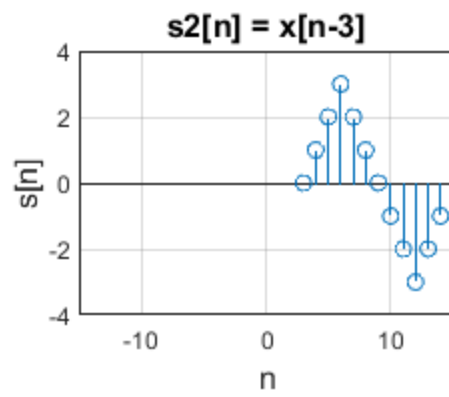
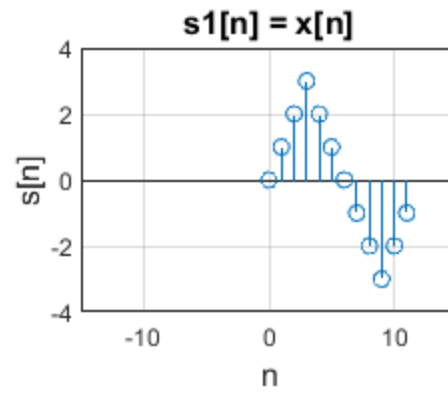
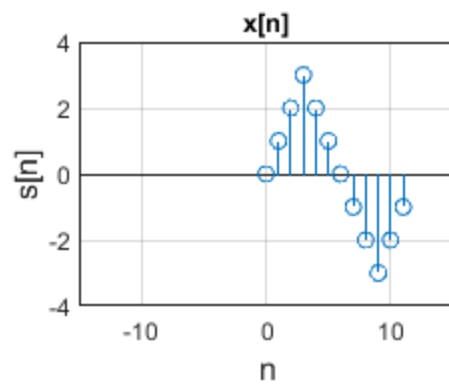
r2 =
  0   1   2   3   2   1   0  -1  -2  -3  -2
-1

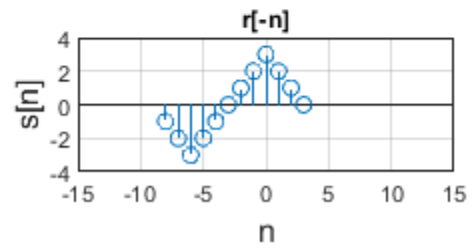
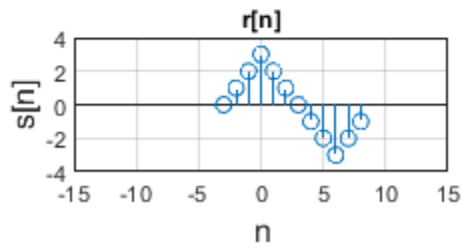
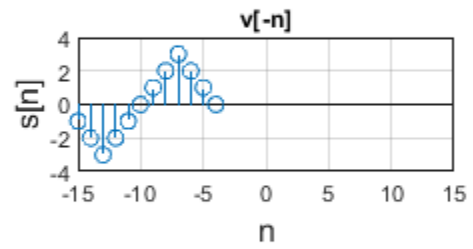
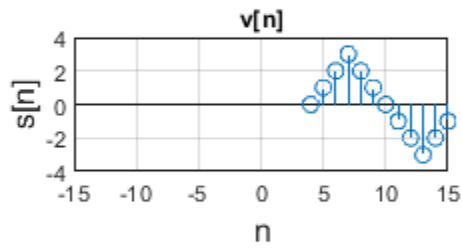
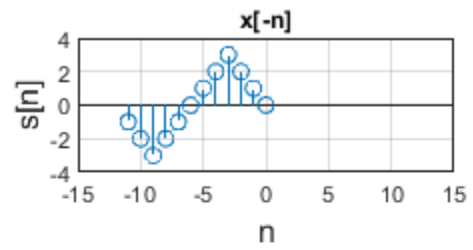
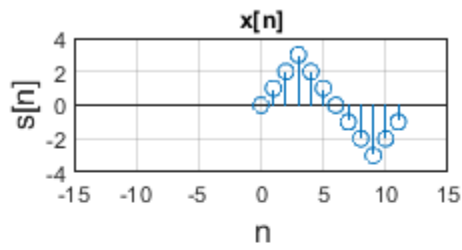
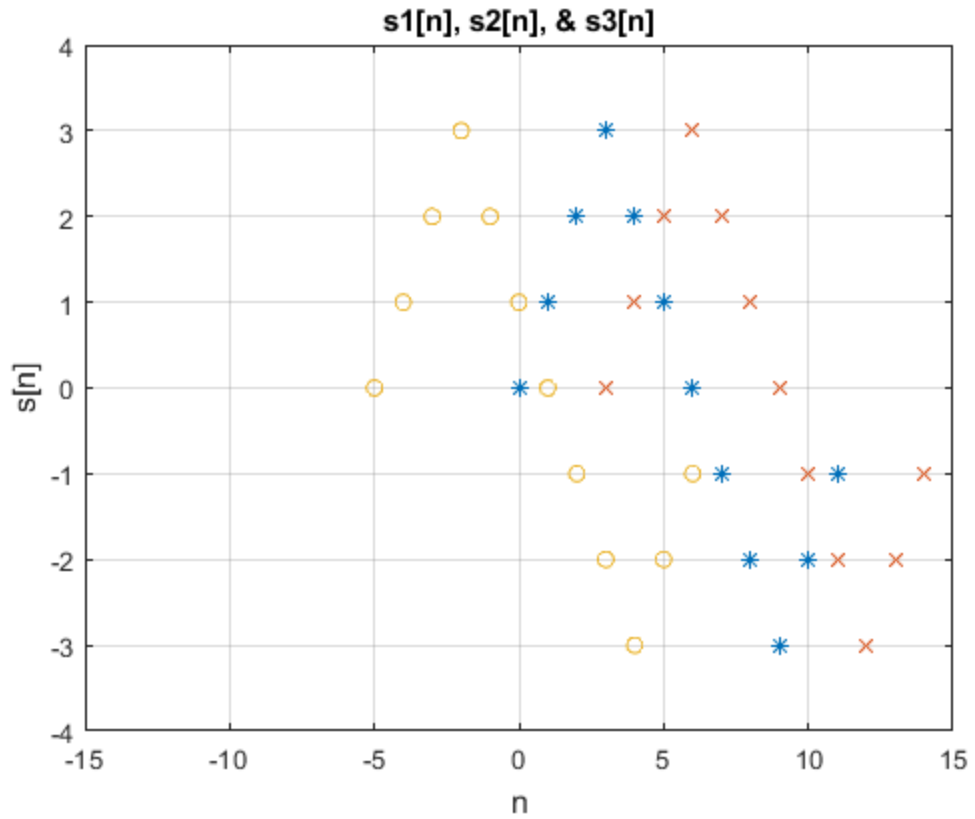
k2 =
 -4  -5  -6  -7  -8  -9 -10 -11 -12 -13 -14
-15

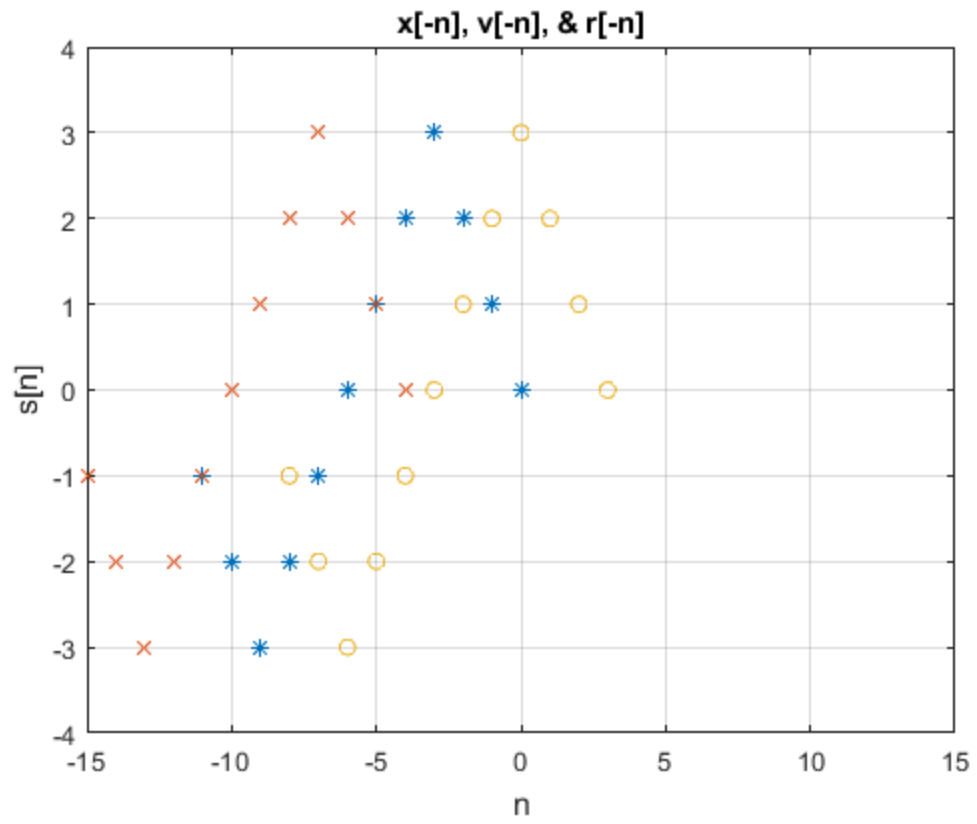
r3 =
  0   1   2   3   2   1   0  -1  -2  -3  -2
-1

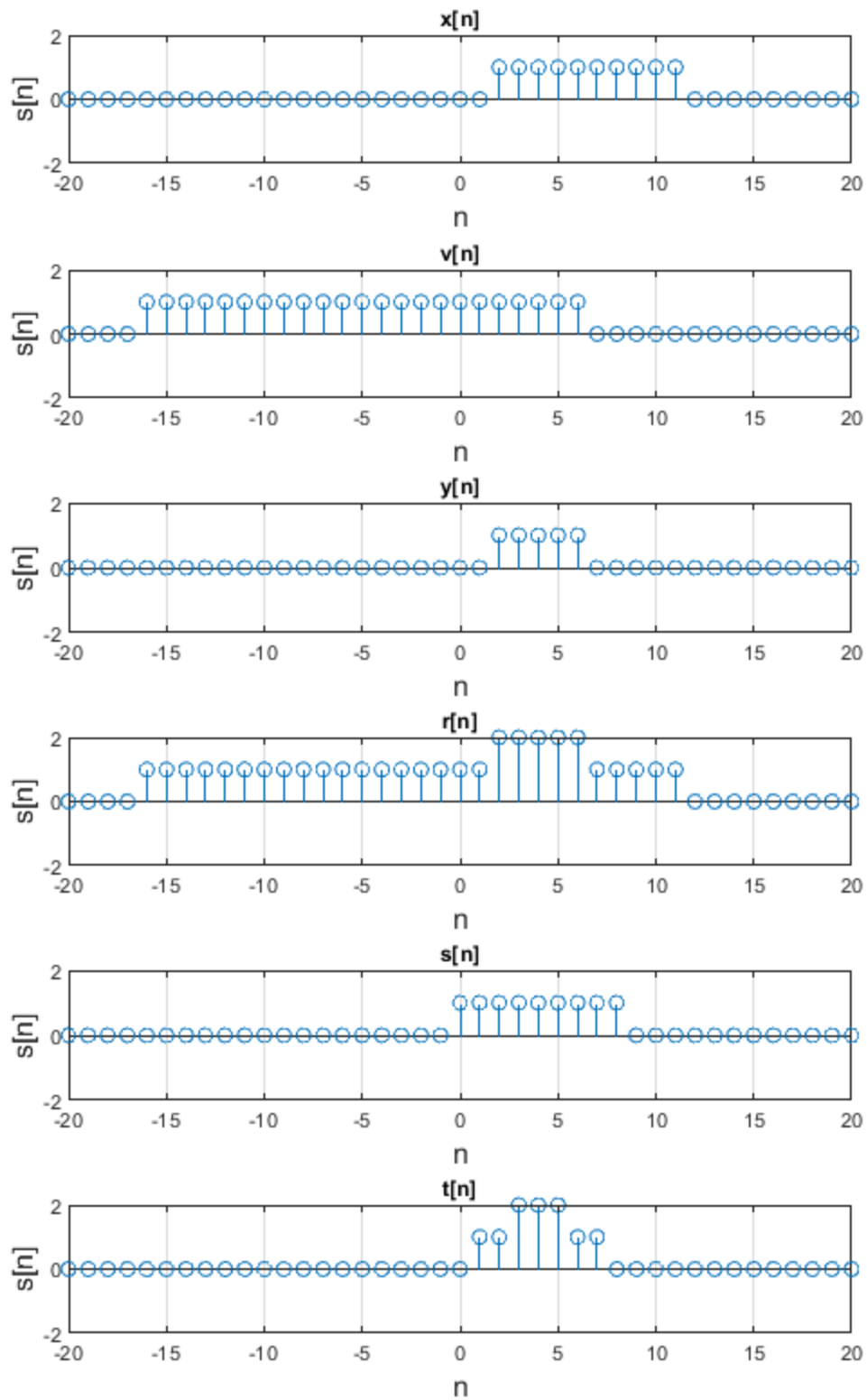
k3 =
  3   2   1   0  -1  -2  -3  -4  -5  -6  -7
-8

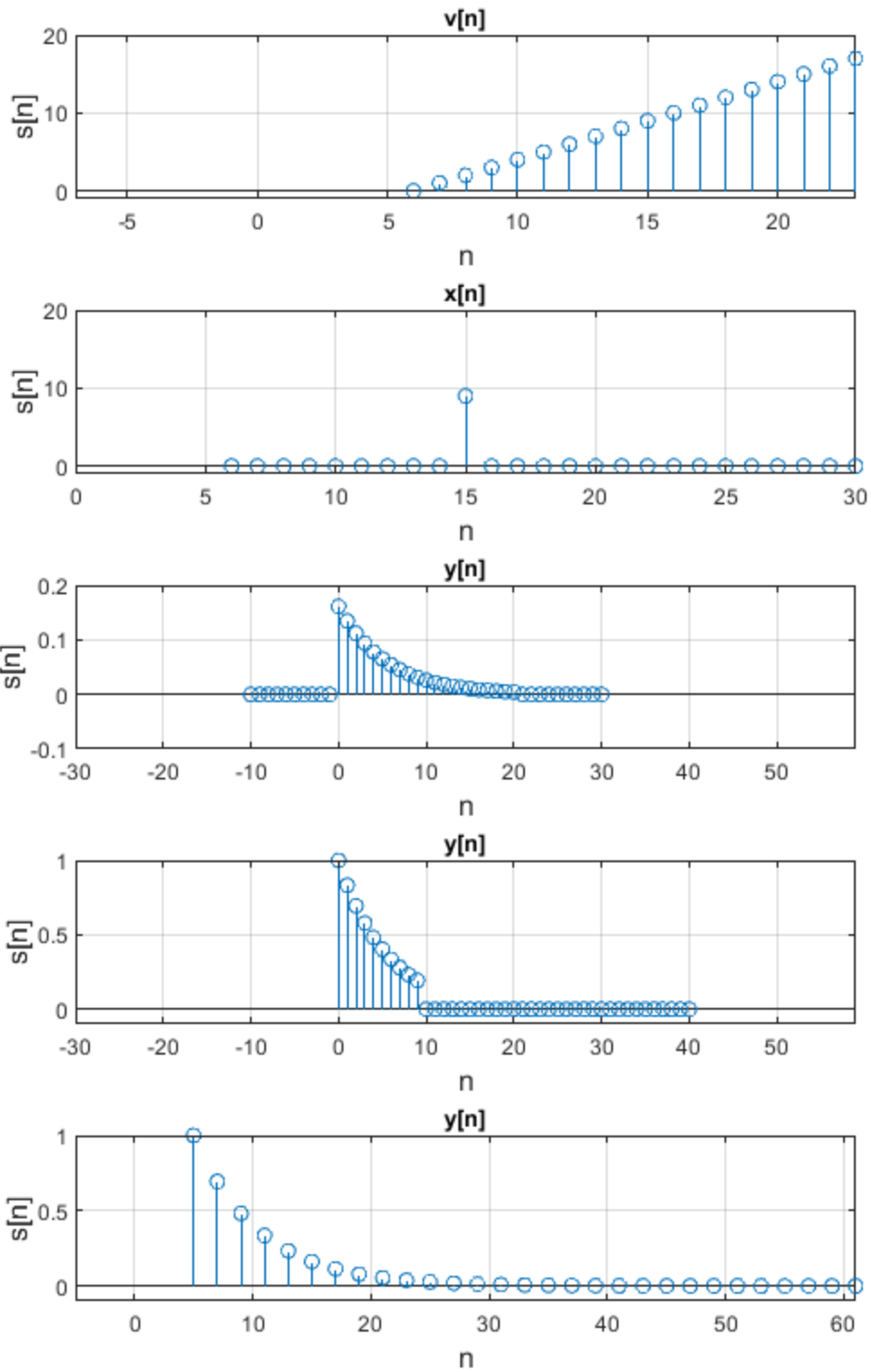
```











Published with MATLAB® R2020a