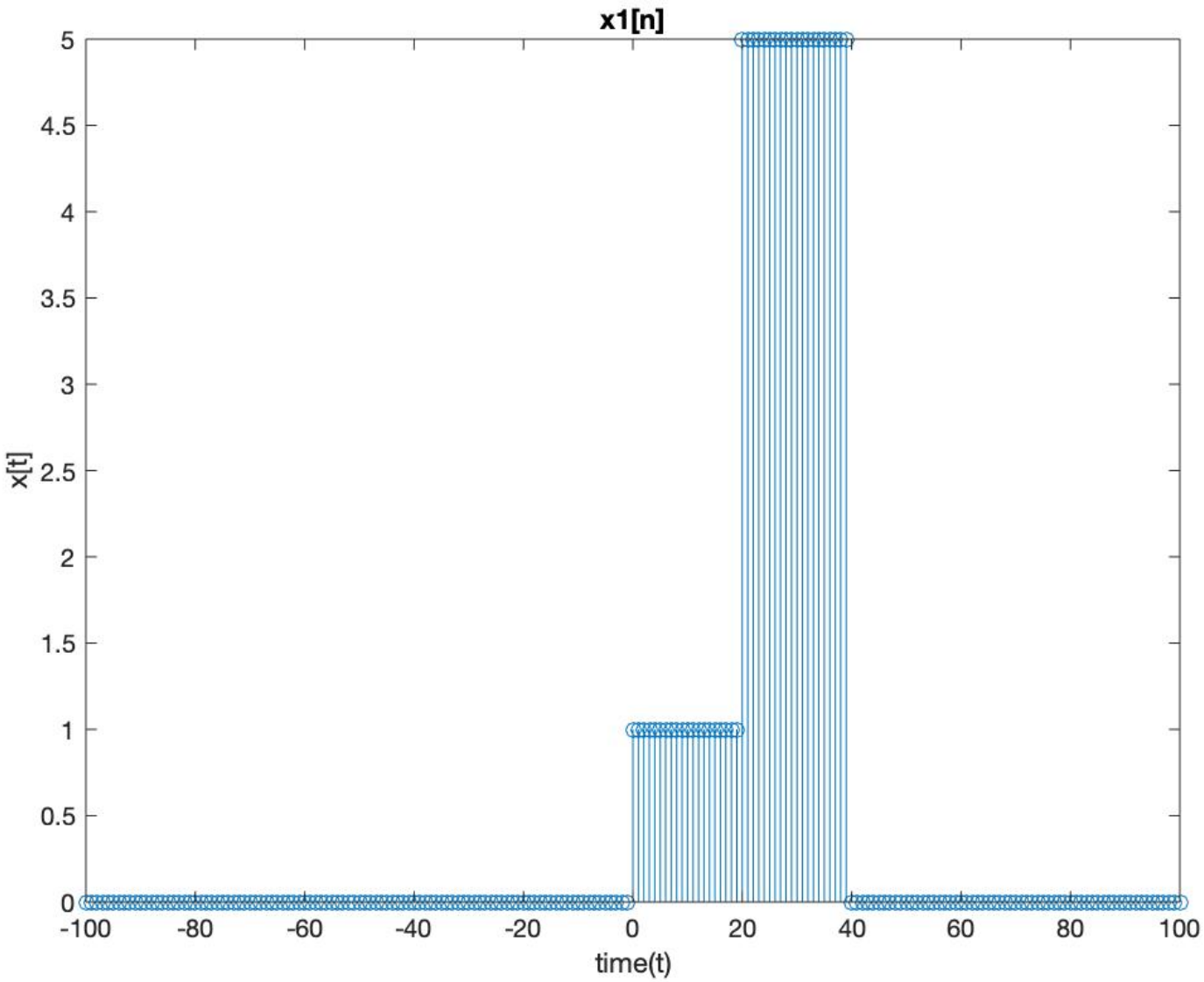
Arda Kıray
21302072
Section 2

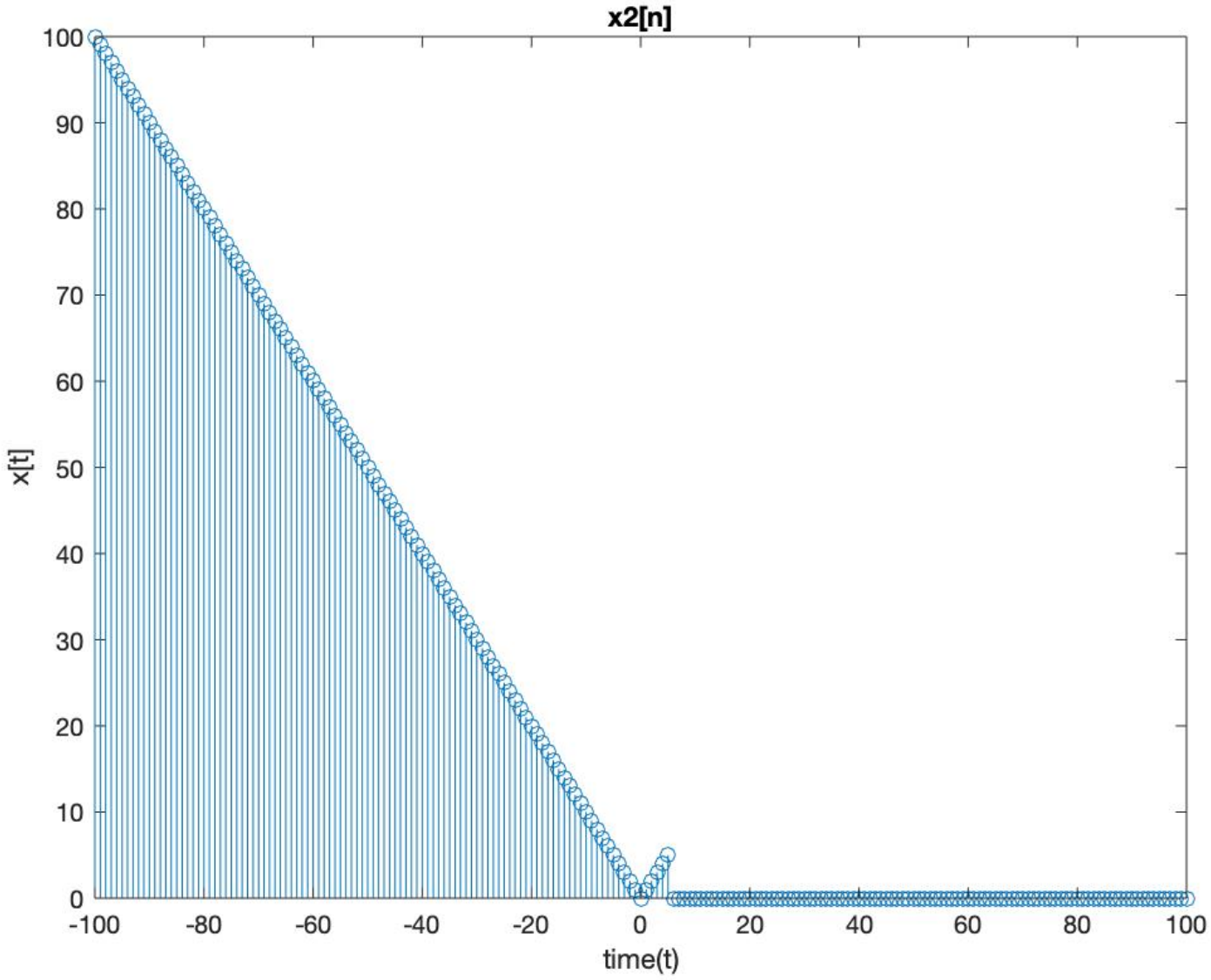
EEE 391
Basics of Signals and Systems
Computer Assignment 2

1) Plots of the 1st part :

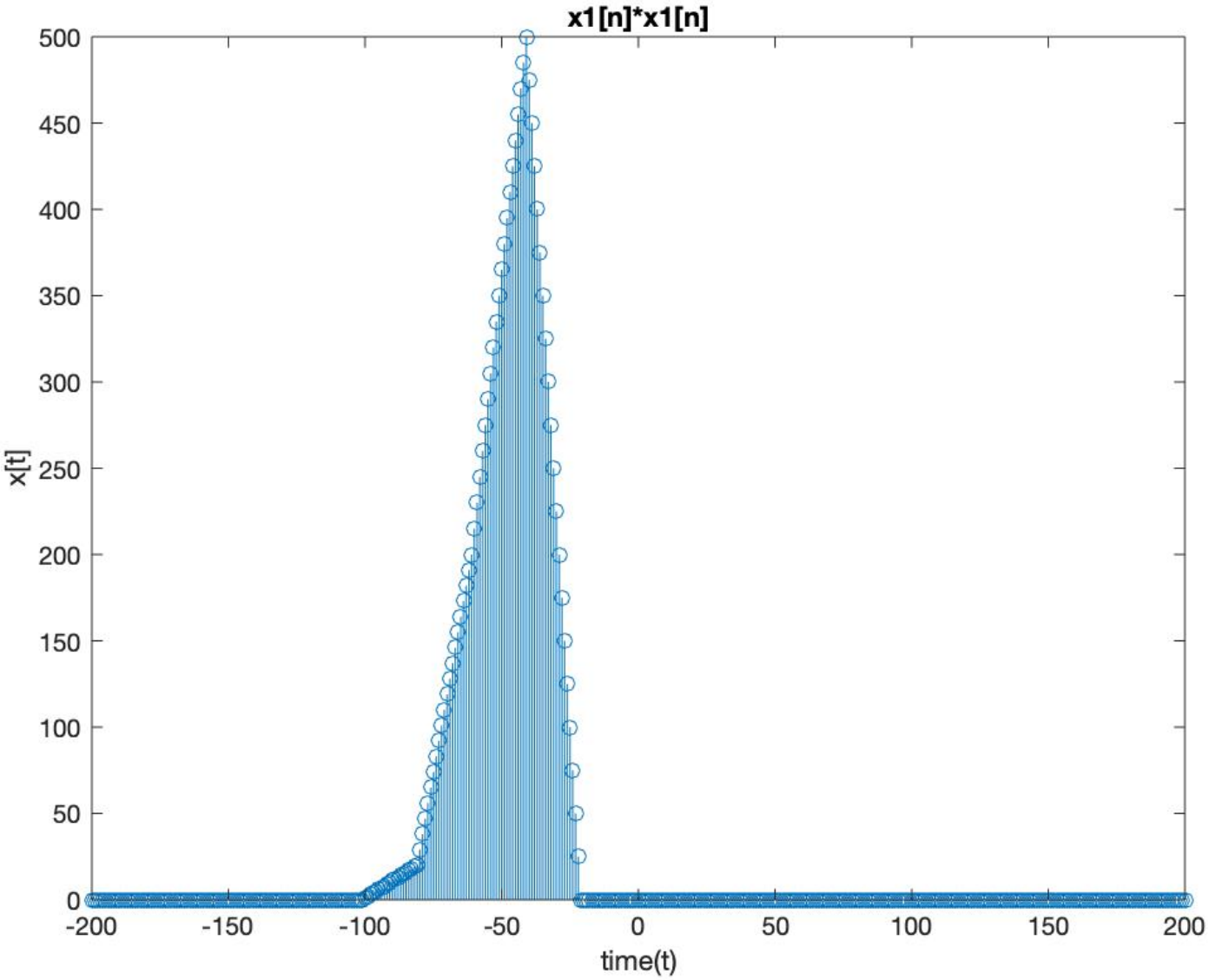
- Plot of x1[n] :



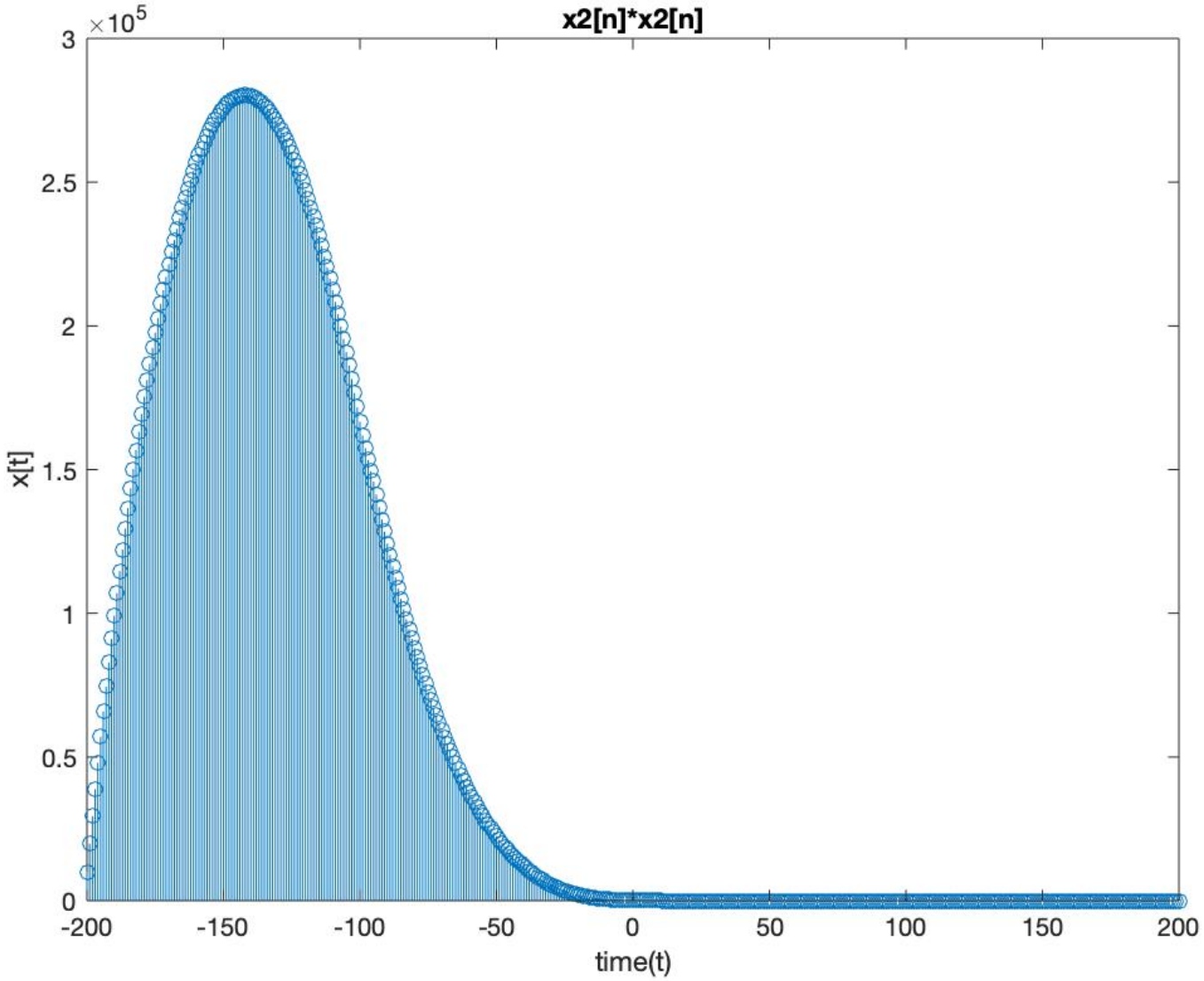
- Plot of x2[n] :


x2[n]
x[t]
time(t)
100
90
80
70
60
50
40
30
20
10
0
-100
-80
-60
-40
-20
0
20
40
60
80
100

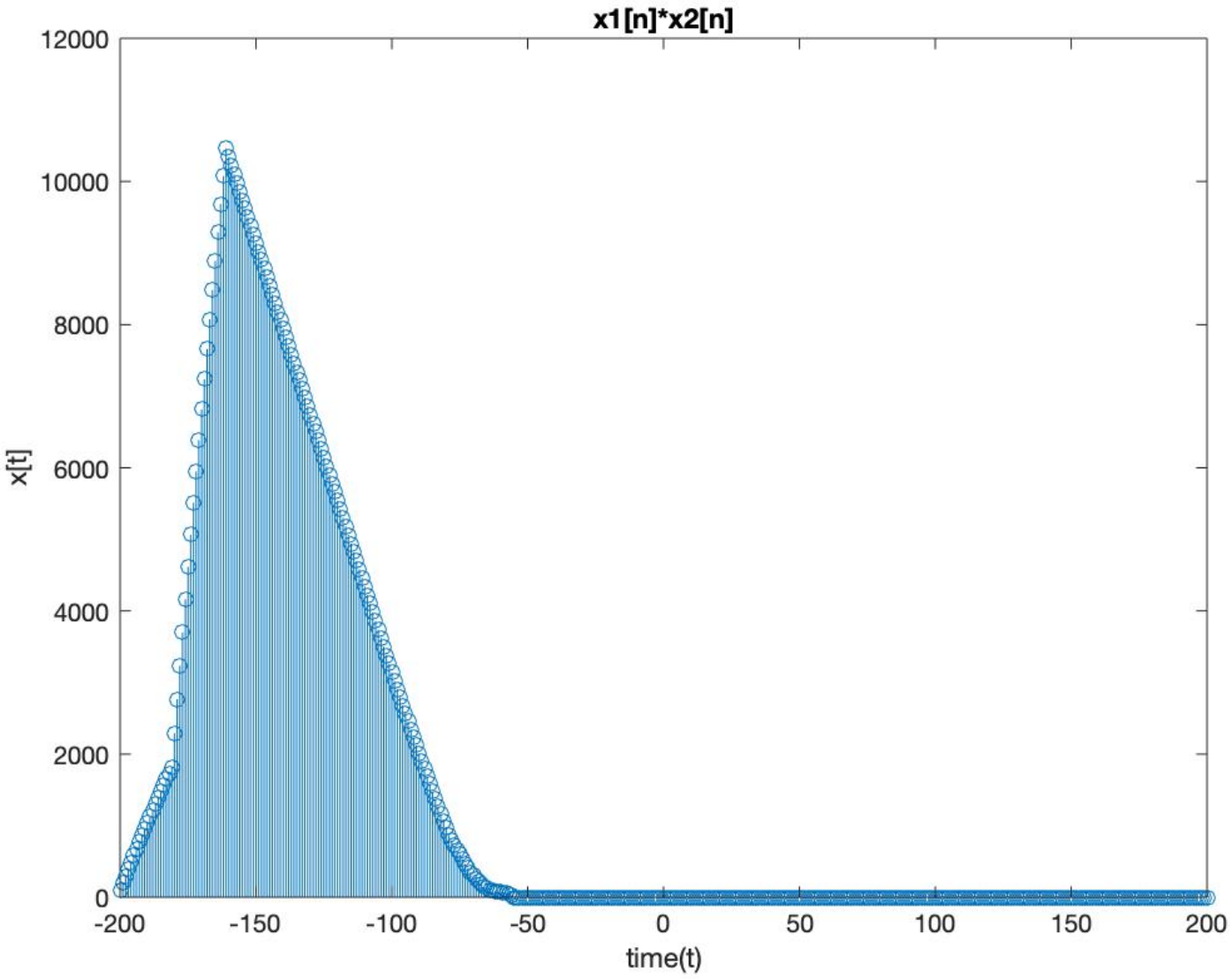
- Plot of x1[n] * x1[n] :



- Plot of x2[n] * x2[n] :

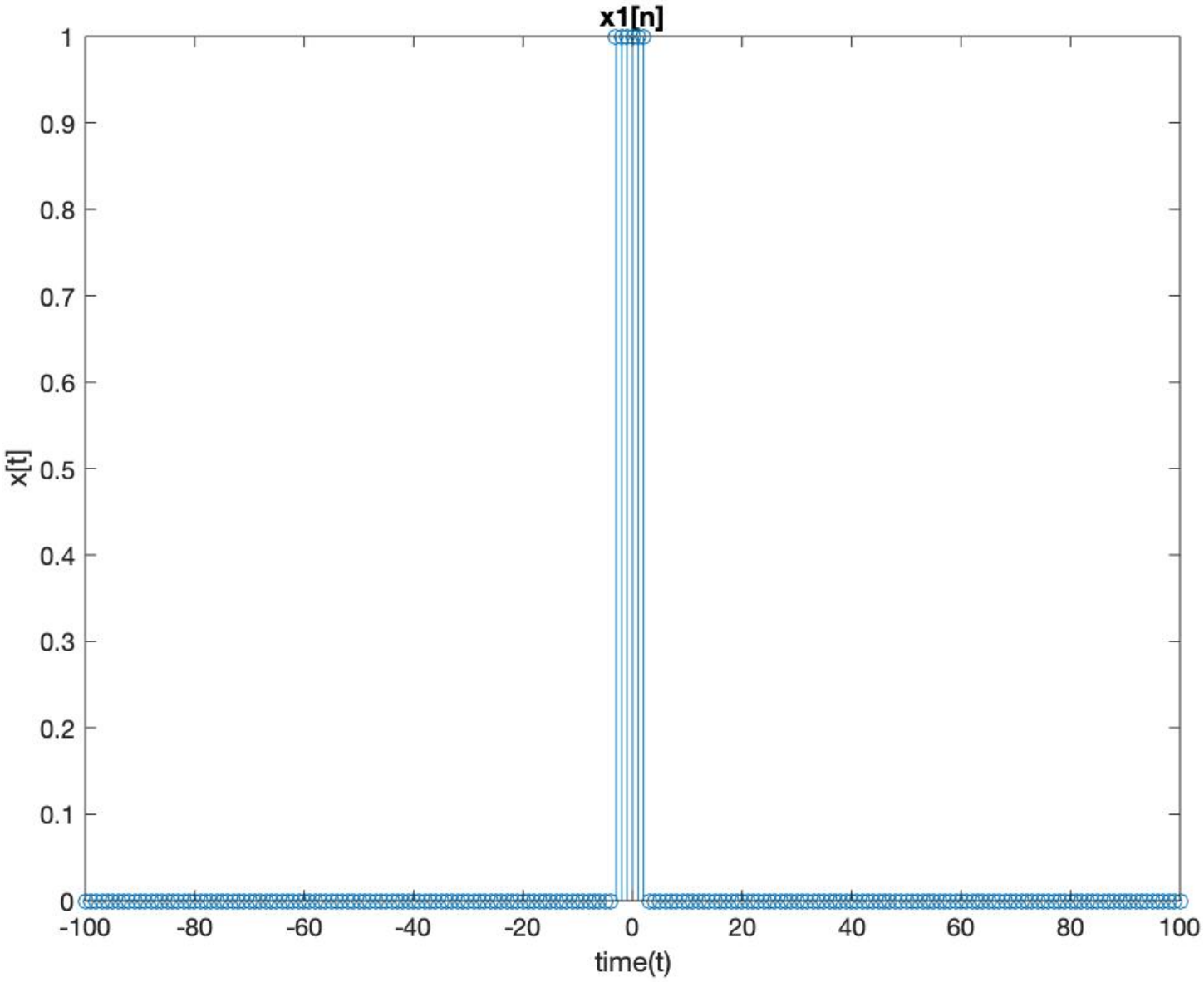


- Plot of x1[n] * x2[n] :

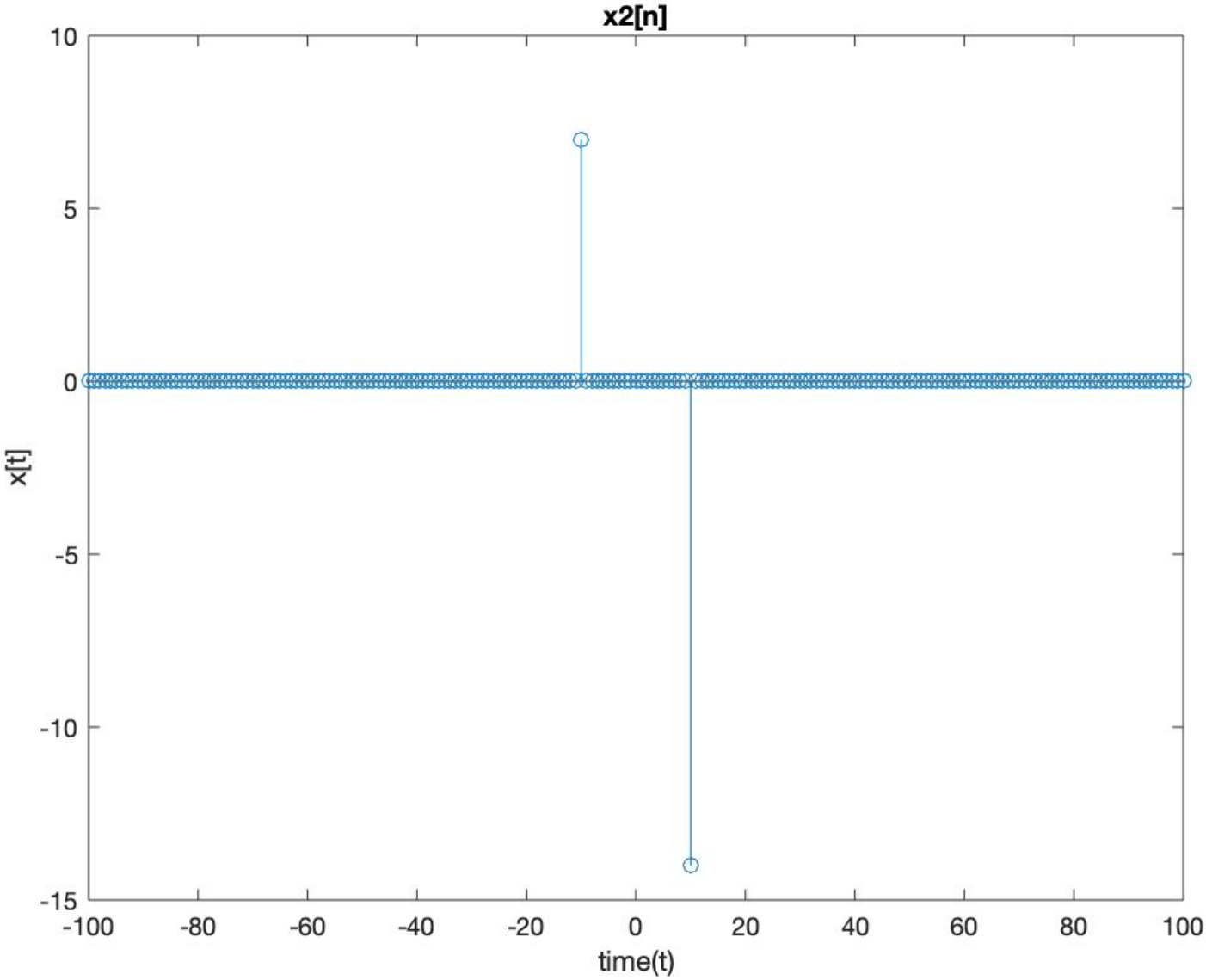


2) Plots of the 2nd Part :

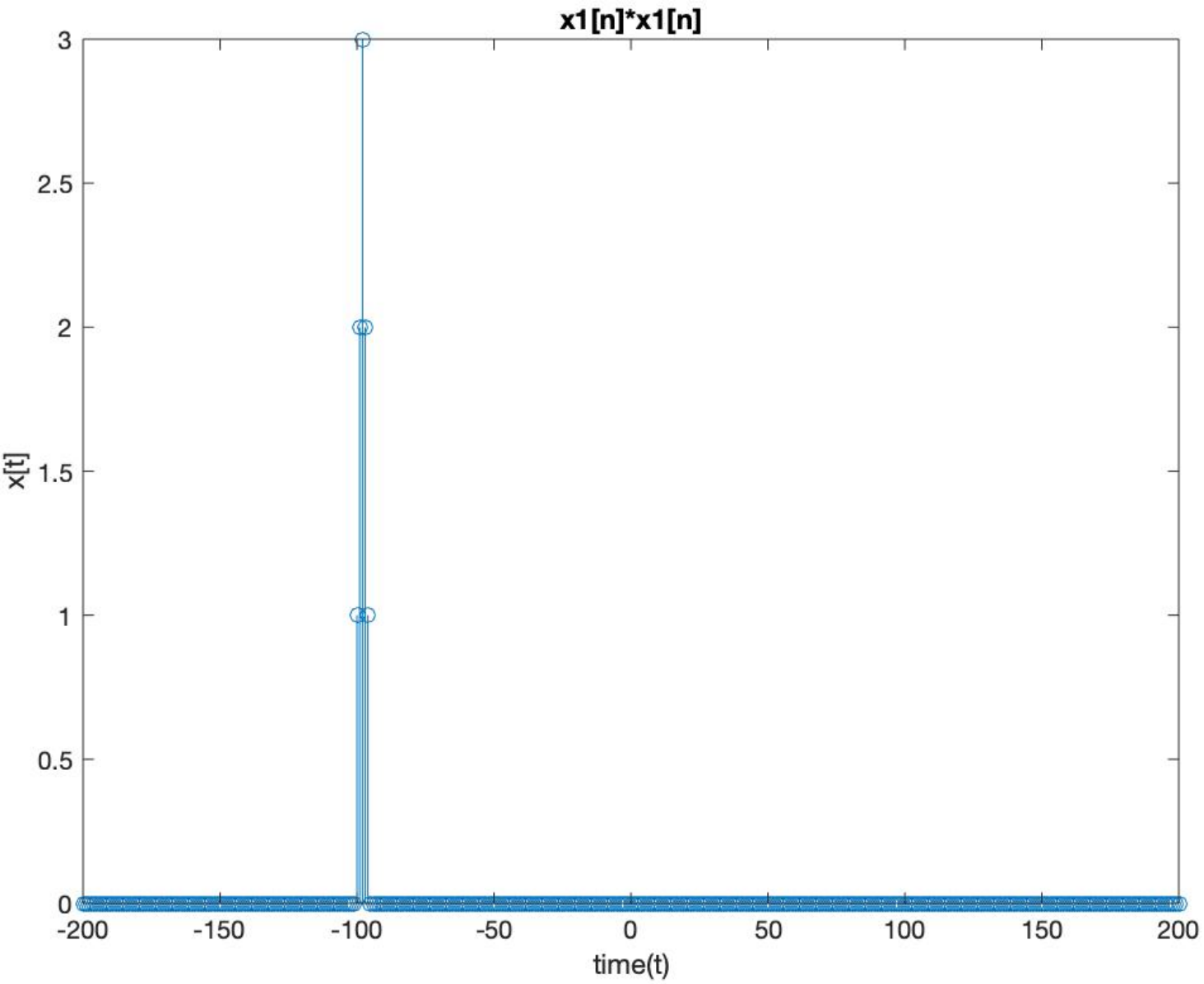
- Plot of x1[n] :



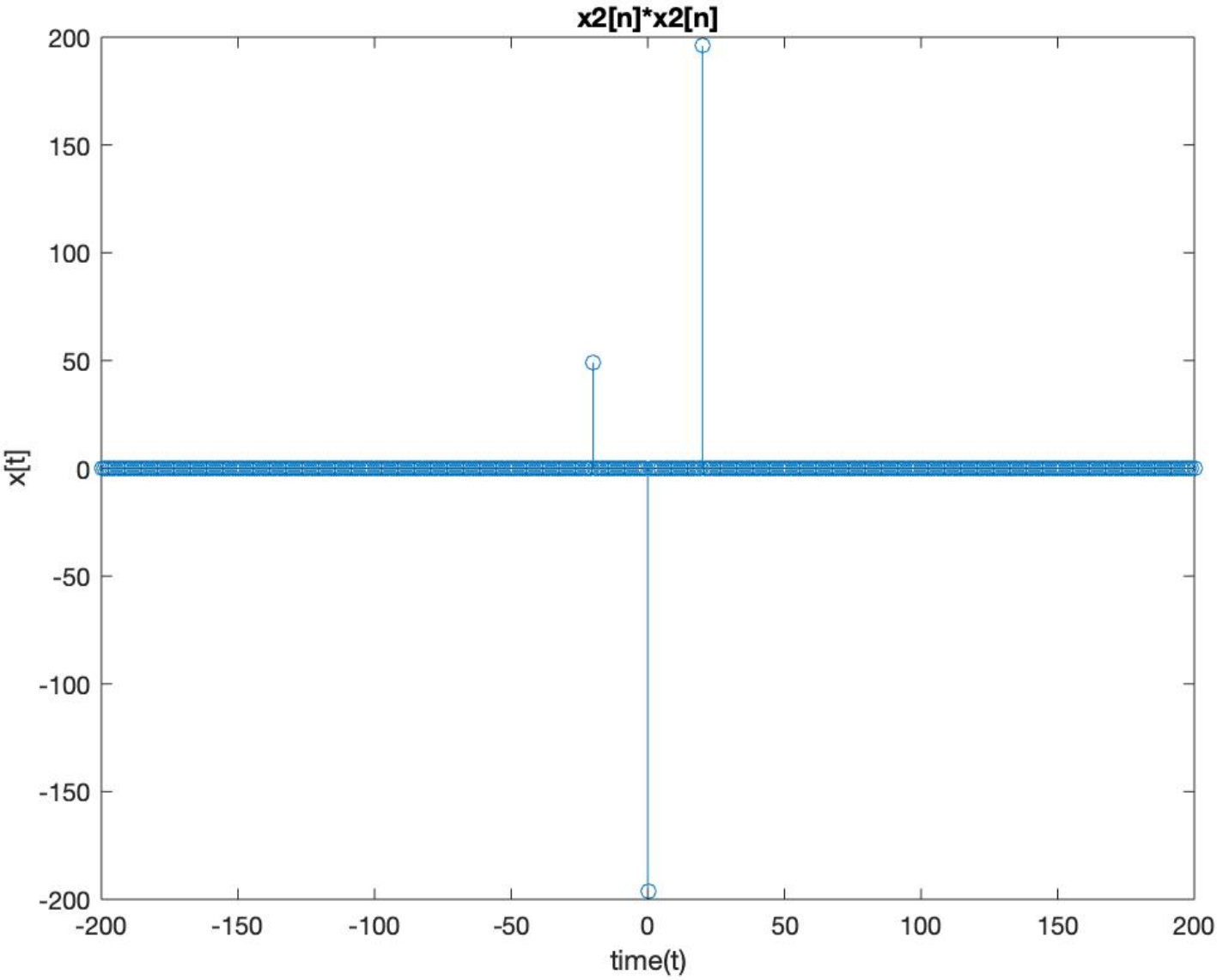
- Plot of x2[n] :



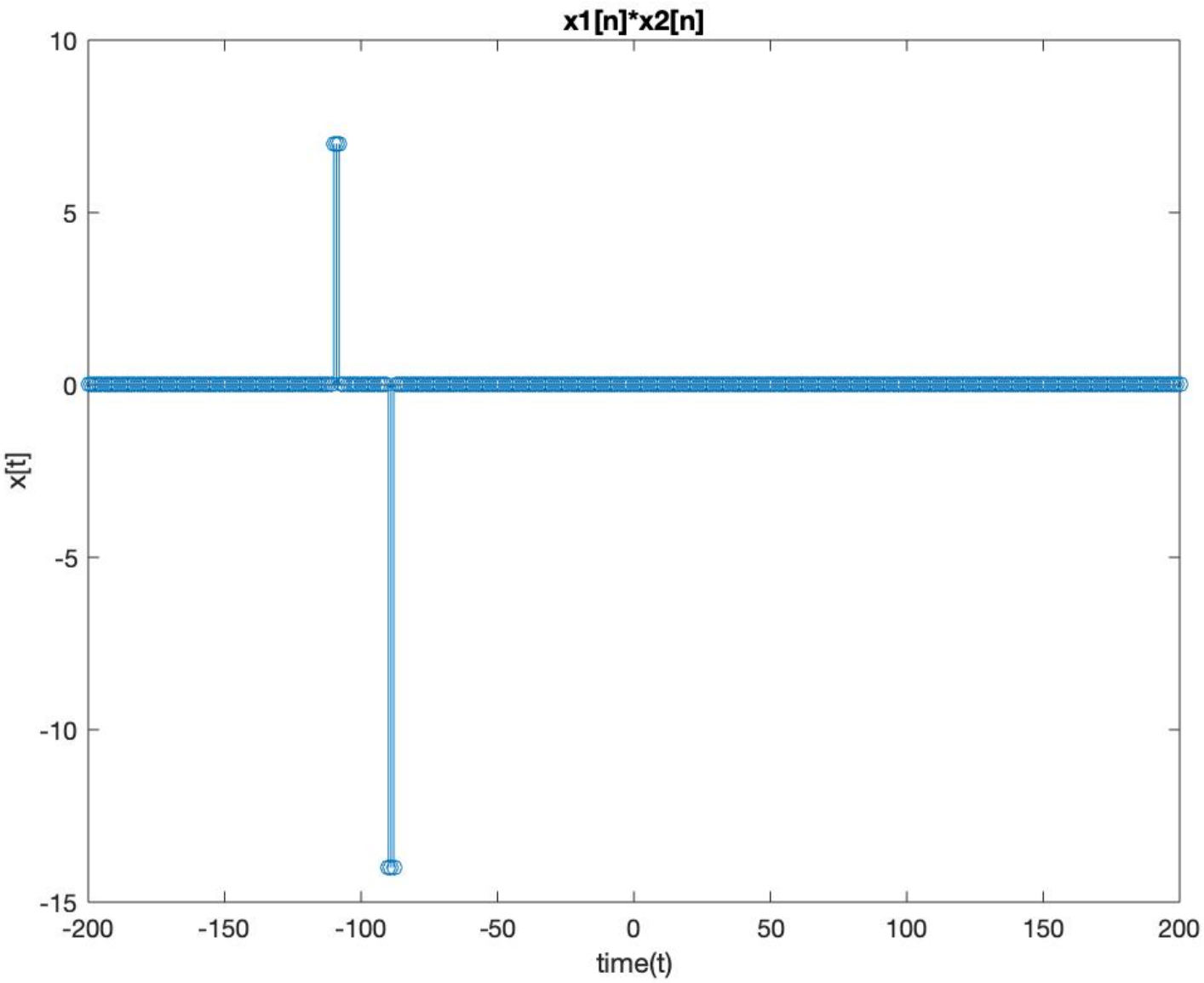
- Plot of x1[n] * x1[n] :



- Plot of x2[n] * x2[n] :

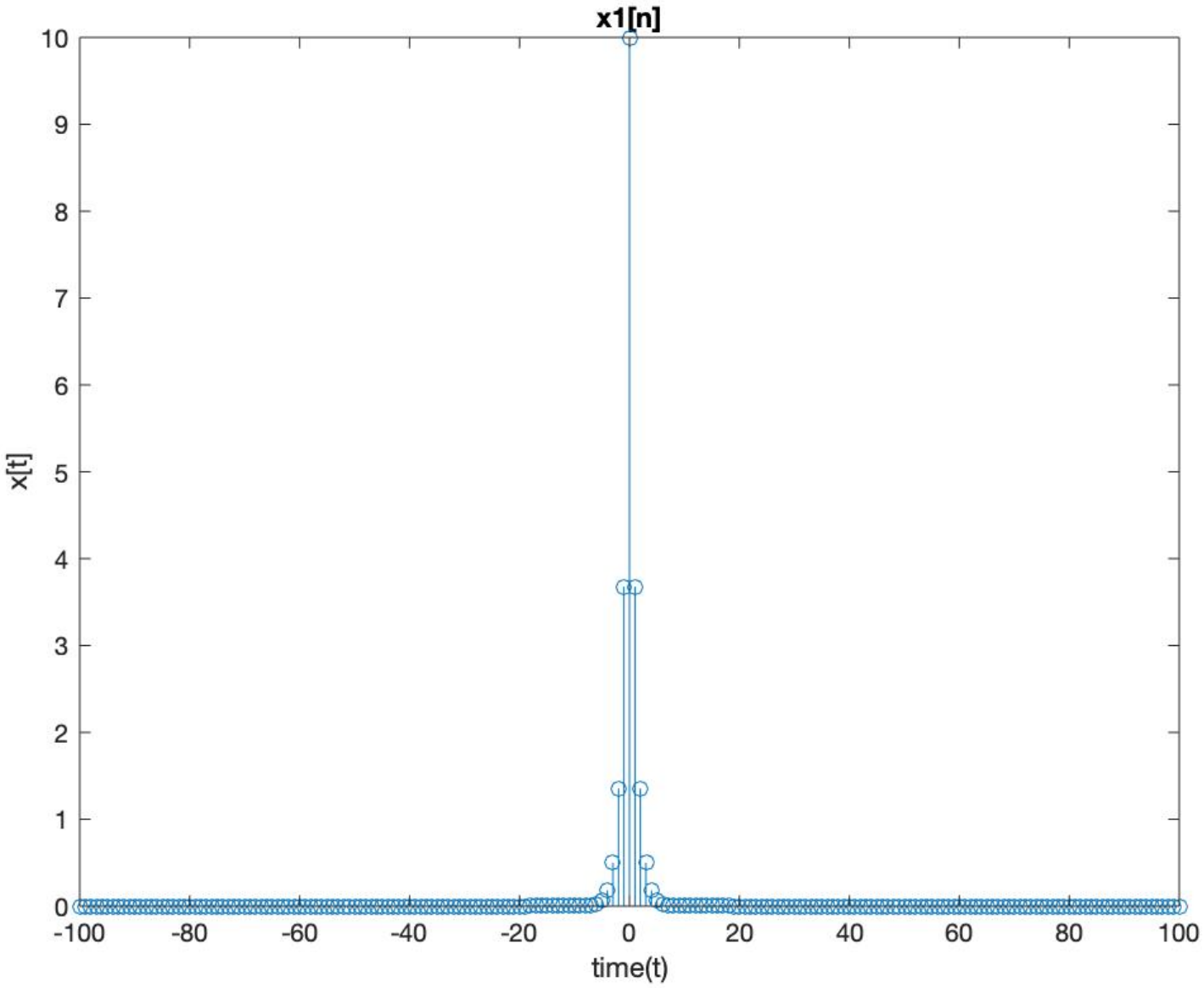


- Plot of x1[n] * x2[n] :

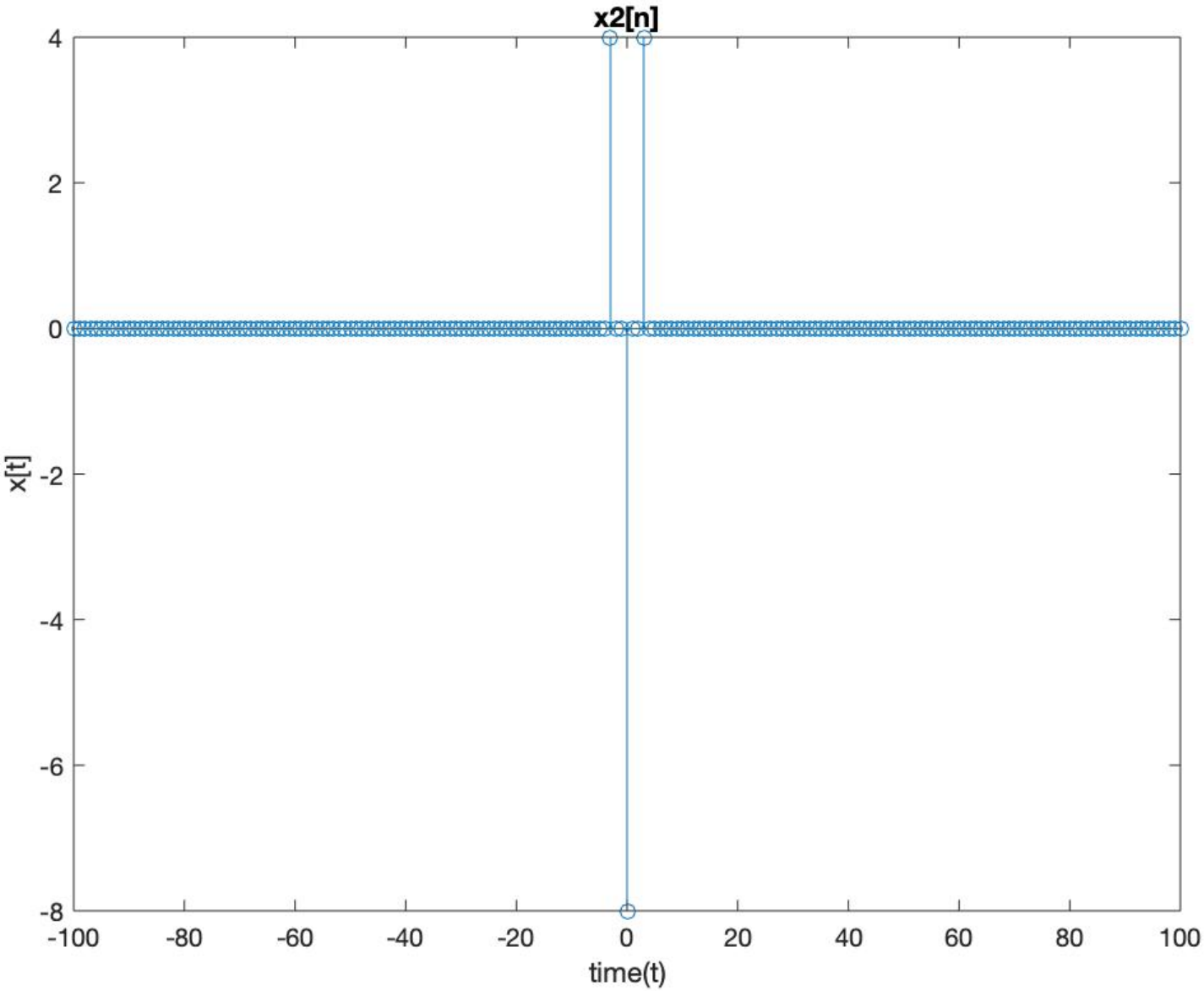


3) Plots of the 3rd part :

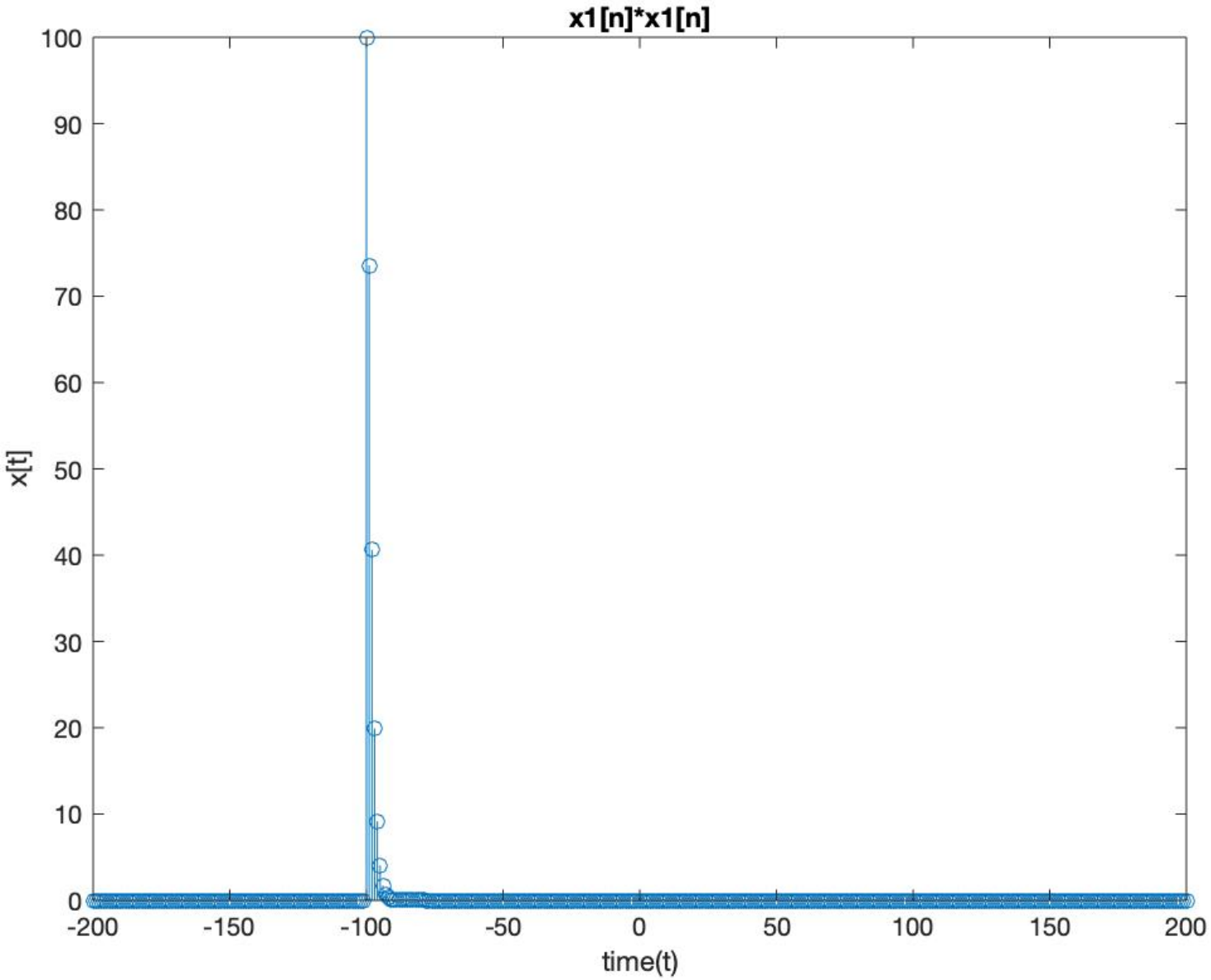
- Plot of x1[n] :



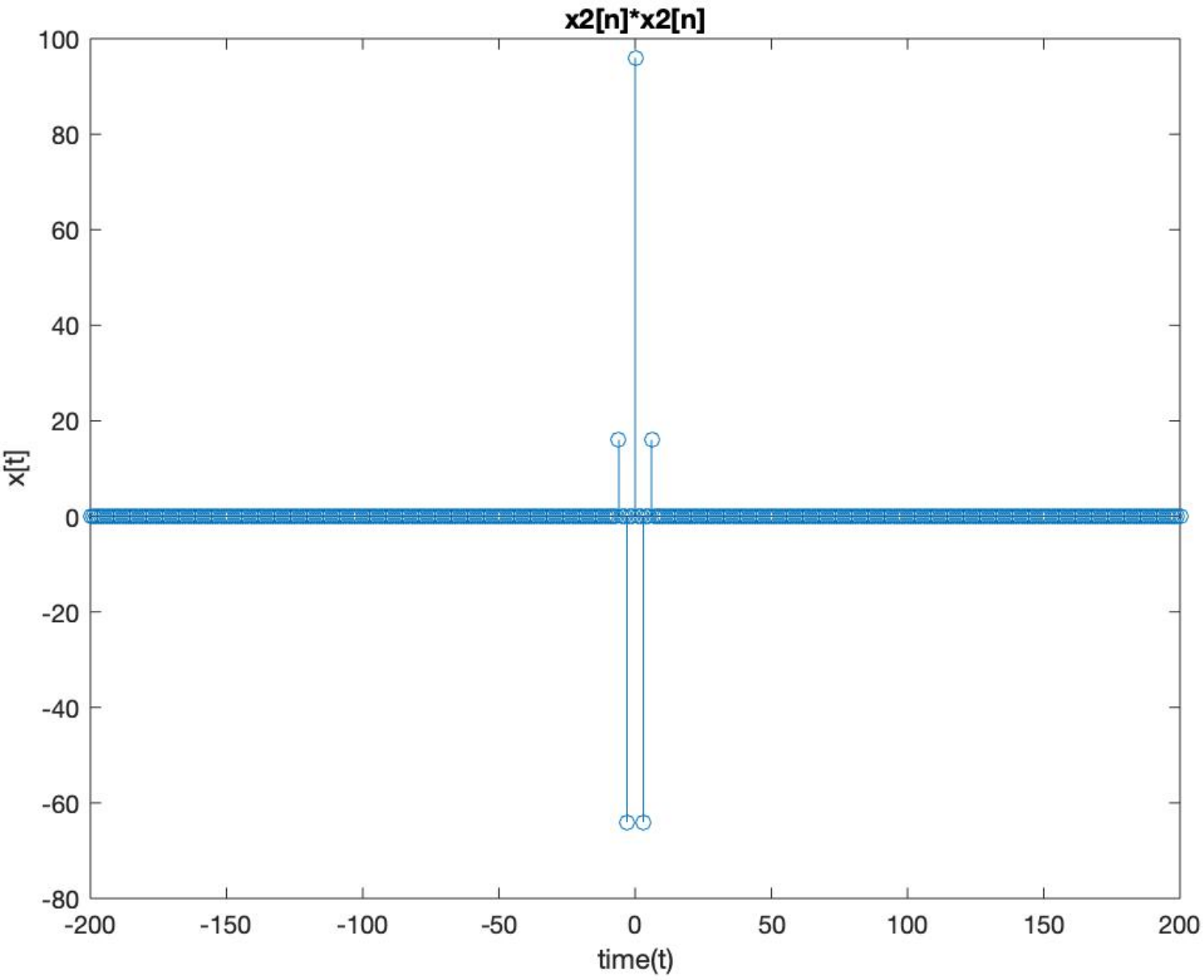
- Plot of x2[n] :



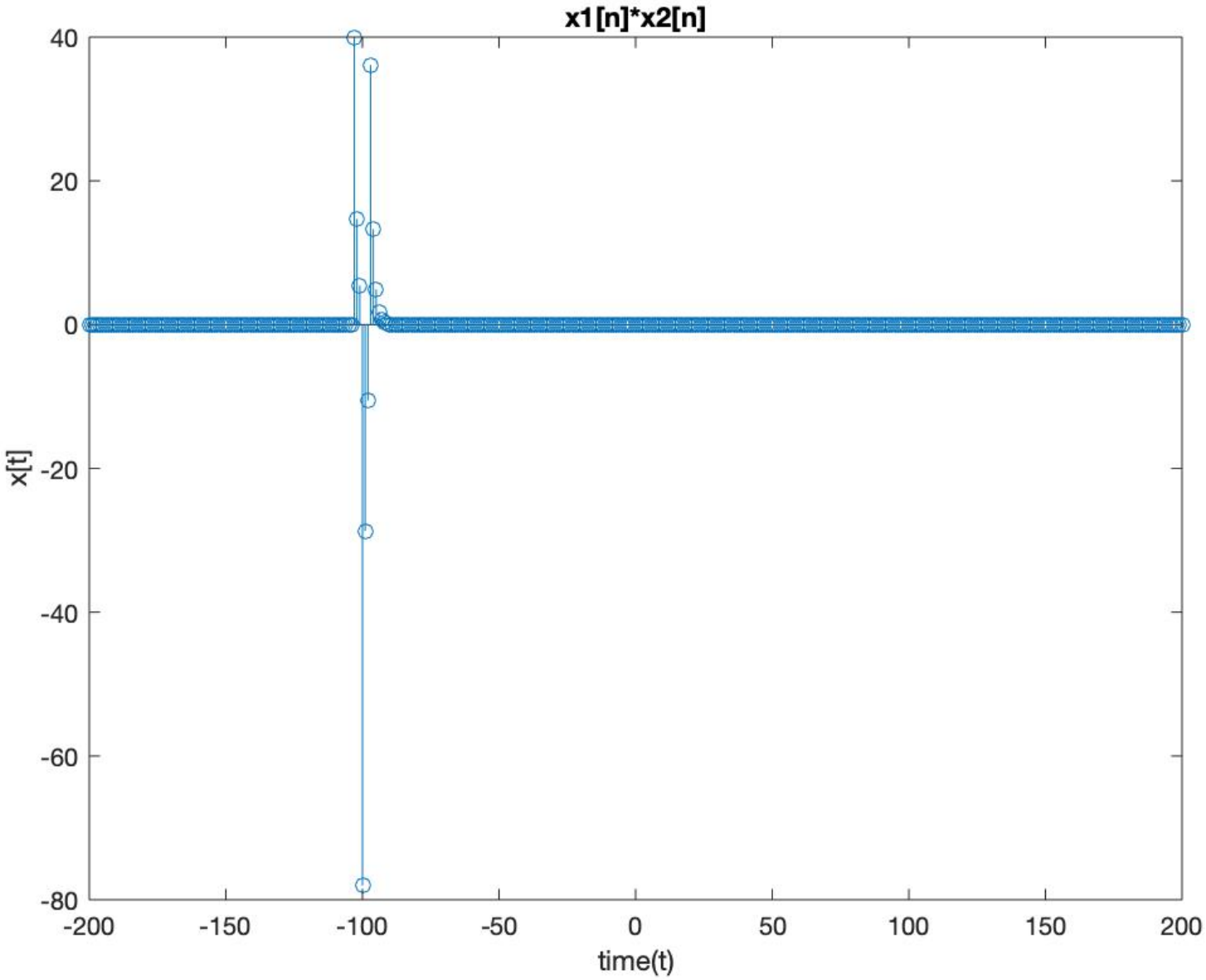
- Plot of x1[n] * x1[n] :



- Plot of x2[n] * x2[n] :



- Plot of x1[n] * x2[n] :



Implementation of the convolution function :

I implemented the convolution function by taking 2 [-100,100] array parameters, their first non-zero indices and their length. I took one array and iterate over its non-zero entries by multiplying them one by one with the non-zero elements of the second array and adding it to the according position of the output array. Each time the algorithm shifts one index from the first array, it also shifts the index at the resulting array. The logic is the same with the calculation of the convolution by using a tabular method. Each results are cross-checked with the Matlab's built-in conv( ) function for reassuring the correctness. The code for convolution function and the creation of the each respected discrete sample arrays are below.

MATLAB CODE :

Convolve.m :

```
function result = convolve(A,lengthA,indexA, B, lengthB, indexB)
    result = zeros(1,401);
    calcIndex = 0;
    for i = indexA:lengthA
        for j = indexB:lengthB
            k = A(i)*B(j);
            result(j + calcIndex) = result(j + calcIndex) + k;
        end
    calcIndex = calcIndex + 1;
    end
```

Main.m :

```
%% ################### 1st Part: ###################
%% 1st x1[n] sample array:
x1 = zeros(1,201);
for i = 101:120
    x1(i) = 1;
end
for i = 121:140
    x1(i) = 5;
end

%% 1st x2[n] sample array:
x2 = zeros(1,201);
for i = 1:106
    if i < 101
        x2(i) = 101 - i;
    else
        x2(i) = i - 101;
    end
end

%% X axis time values
xData = zeros(1,201);
xData2 = zeros(1,401);
for i = 1:201
    xData(i) = i - 101;
end
for i = 1:401
```

```
    xData2(i) = i - 201;
end

%% Convolve and plot the results:
result1 = convolve(x1, 201, 101, x1, 201, 101);
result2 = convolve(x2, 201, 1, x2, 201, 1);
result3 = convolve(x1, 201, 101, x2, 201, 1);

%% 1st - x1[n] plot
figure
figure1 = stem(x1);
figure1.XData = xData;
xlabel('time(t)')
ylabel('x[t]')
title(‘x1[n]')

%% 1st - x2[n] plot
figure
figure2 = stem(x2);
figure2.XData = xData;
xlabel('time(t)')
ylabel('x[t]')
title(‘x2[n]')

%% 1st - x1[n]*x1[n] plot
figure
figure3 = stem(result1);
figure3.XData = xData2;
xlabel('time(t)')
ylabel('x[t]')
title(‘x1[n]*x1[n]')

%% 1st - x2[n]*x2[n] plot
figure
figure4 = stem(result2);
figure4.XData = xData2;
xlabel('time(t)')
ylabel('x[t]')
title(‘x2[n]*x2[n]')

%% 1st - x1[n]*x2[n] plot
figure
figure5 = stem(result3);
figure5.XData = xData2;
xlabel('time(t)')
```

```
ylabel('x[t]')
title('x1[n]*x2[n]')

%% ################### 2nd Part: ###################
%% 2nd x1[n] sample array :
x1 = zeros(1,201);
for i = 98:103
    x1(i) = 1;
end

%% 2nd x2[n] sample array:
x2 = zeros(1,201);
x2(111) = -14;
x2(91) = 7;

%% Convolve and plot the results:
result1 = convolve(x1, 201, 101, x1, 201, 101);
result2 = convolve(x2, 201, 1, x2, 201, 1);
result3 = convolve(x1, 201, 101, x2, 201, 1);

%% 2nd - x1[n] plot
figure
figure1 = stem(x1);
figure1.XData = xData;
xlabel('time(t)')
ylabel('x[t]')
title(‘x1[n]')

%% 2nd - x2[n] plot
figure
figure2 = stem(x2);
figure2.XData = xData;
xlabel('time(t)')
ylabel('x[t]')
title(‘x2[n]')

%% 2nd - x1[n]*x1[n] plot
figure
figure3 = stem(result1);
figure3.XData = xData2;
xlabel('time(t)')
ylabel('x[t]')
title(‘x1[n]*x1[n]')
```

```
%% 2nd - x2[n]*x2[n] plot
figure
figure4 = stem(result2);
figure4.XData = xData2;
xlabel('time(t)')
ylabel('x[t]')
title(‘x2[n]*x2[n]')

%% 2nd - x1[n]*x2[n] plot
figure
figure5 = stem(result3);
figure5.XData = xData2;
xlabel('time(t)')
ylabel('x[t]')
title('x1[n]*x2[n]')

%%  ################### 3rd Part: ###################
%% 3rd x1[n] sample array :
x1 = zeros(1,201);
counter = 25;
for i = 76:101
    x1(i) = 10*exp(-counter);
    counter = counter - 1;
end
counter = 1;
for i = 102:126
    x1(i) = 10*exp(-counter);
    counter = counter + 1;
end

%% 3rd x2[n] sample array:
x2 = zeros(1,201);
x2(98) = 4;
x2(101) = -8;
x2(104) = 4;

%% Convolve and plot the results:
result1 = convolve(x1, 201, 101, x1, 201, 101);
result2 = convolve(x2, 201, 1, x2, 201, 1);
result3 = convolve(x1, 201, 101, x2, 201, 1);

%% 3rd - x1[n] plot
figure
figure1 = stem(x1);
```

```
figure1.XData = xData;
xlabel('time(t)')
ylabel('x[t]')
title(‘x1[n]')

%% 3rd - x2[n] plot
figure
figure2 = stem(x2);
figure2.XData = xData;
xlabel('time(t)')
ylabel('x[t]')
title(‘x2[n]')

%% 3rd - x1[n]*x1[n] plot
figure
figure3 = stem(result1);
figure3.XData = xData2;
xlabel('time(t)')
ylabel('x[t]')
title(‘x1[n]*x1[n]')

%% 3rd - x2[n]*x2[n] plot
figure
figure4 = stem(result2);
figure4.XData = xData2;
xlabel('time(t)')
ylabel('x[t]')
title(‘x2[n]*x2[n]')
20
%% 3rd - x1[n]*x2[n] plot
figure
figure5 = stem(result3);
figure5.XData = xData2;
xlabel('time(t)')
ylabel('x[t]')
title('x1[n]*x2[n]')
```