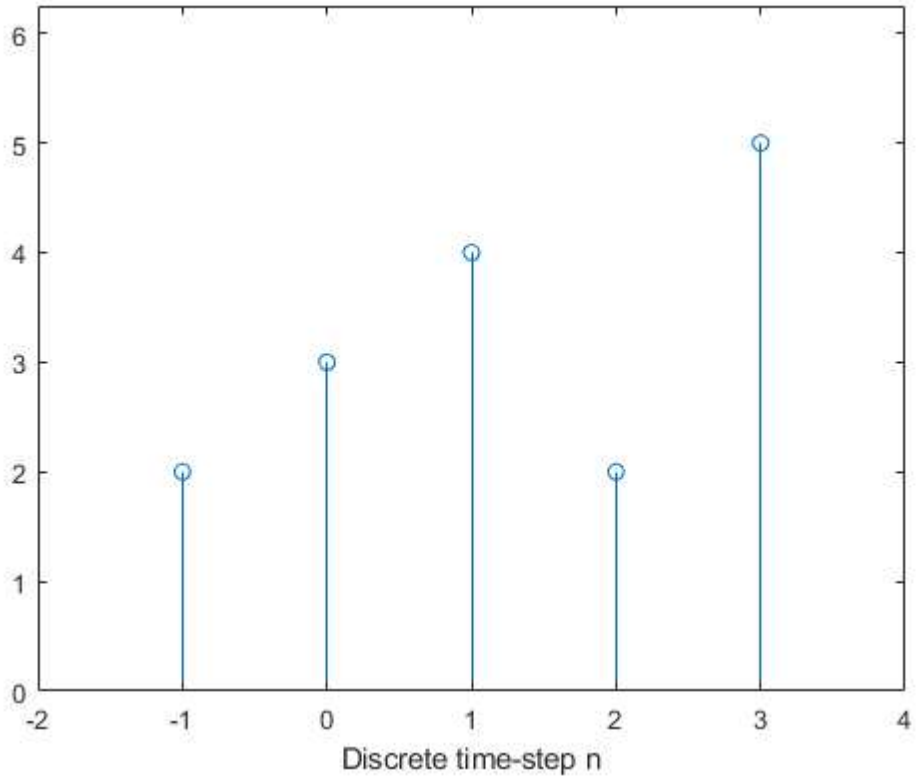
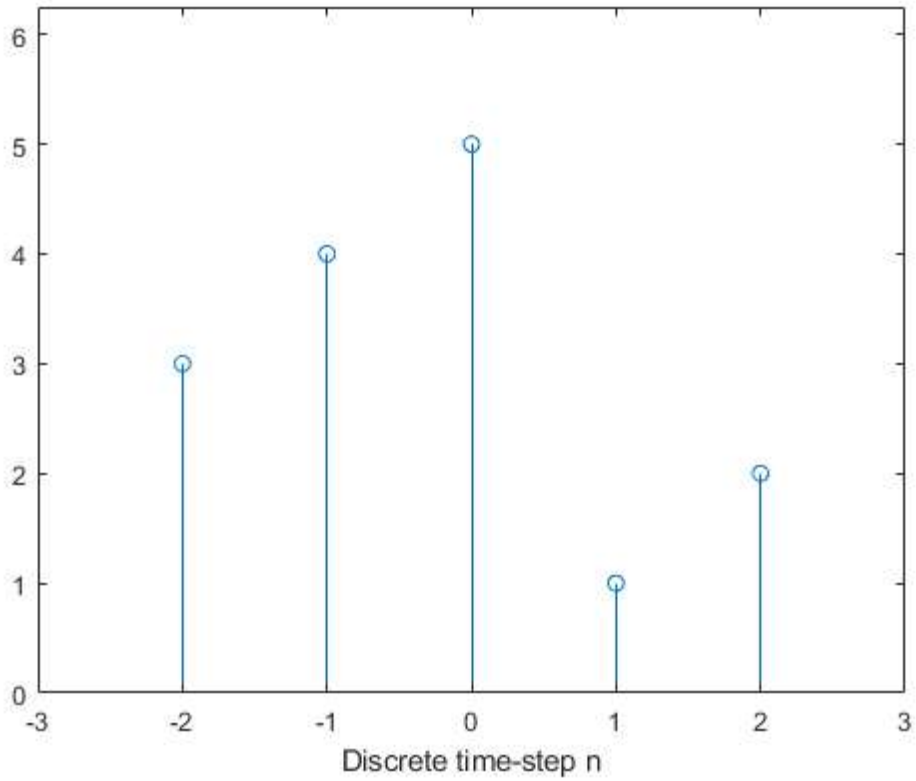
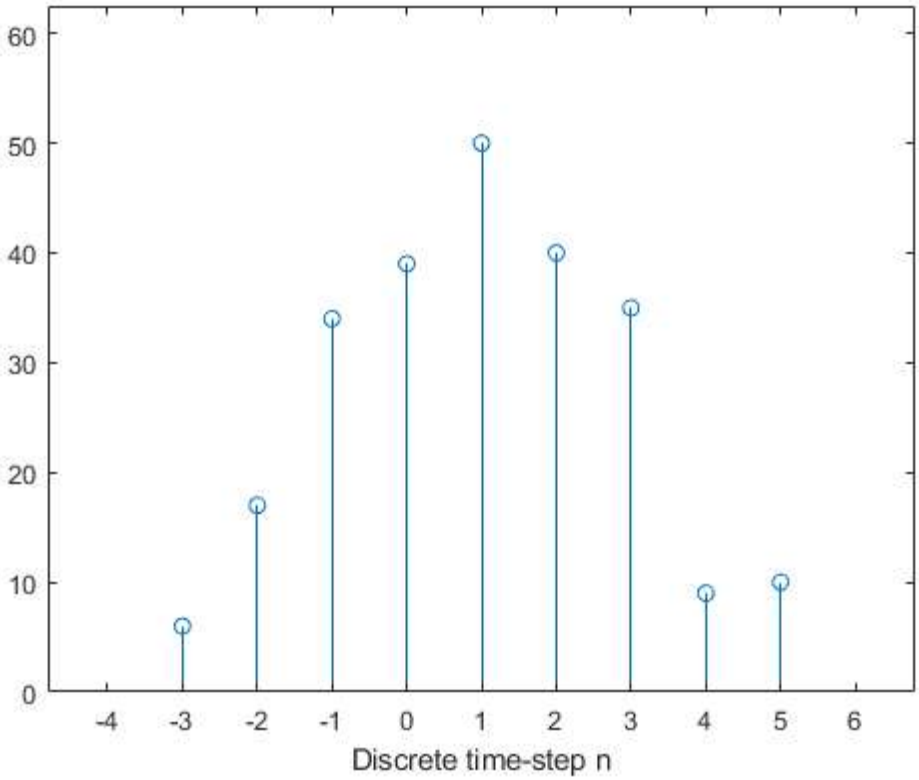
```
% SPDX-License-Identifier: GPL-3.0-or-later
%
% ECE211 Signal Processing - Problem Set 3
% Copyright (C) 2024 Aidan Cusa <aidancusa@gmail.com>

clc;    % clear command window
clear;  % clear all variables from current worwkspace
close all;

% data values vectors
h = [2, 3, 4, 2, 5];
x = [3, 4, 5, 1, 2];

% time index vectors corresponding to data value vector indices
nh = [-1, 0, 1, 2, 3];
nx = [-2, -1, 0, 1, 2];

figure;
h_plot = s_plot(nh, h);

figure;
x_plot = s_plot(nx, x);

% find value vector and time index vector for convolution y = h * x
[yn, yd] = c_vals(h, nh, x, nx);

figure;
y_plot = s_plot(yn, yd);

function [yn, yd] = c_vals(h, nh, x, nx)
    % calculate convolution values
    yd = conv(h, x);

    % create discrete time step vector
    yn = (nh(1) + nx(1)):(nh(end) + nx(end));
end

% function that creates a stem plot based on
function s_p = s_plot(n, x)
    s_p = stem(n, x);

    % for aesthetic purposes
    xlim([n(1) - length(n) * 0.2, n(end) + length(n) * 0.2]);
    ylim([0, max(x) + max(x) * 0.25]);
    xlabel('Discrete time-step n');
end
```


6
5
4
3
2
1
0
-2
-1
0
1
2
3
4
Discrete time-step n


6
5
4
3
2
1
0
-3
-2
-1
0
1
2
3
Discrete time-step n



Published with MATLAB® R2023b