ENGR 451 - Chapter 2 Laboratory

Authors: Juan Angeles Acuna and Moses Martinez

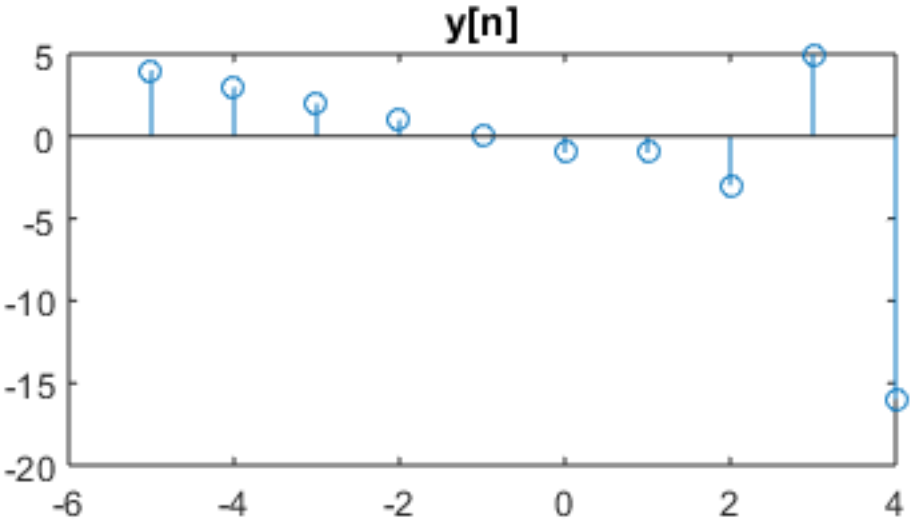
```
clear
x = sequence([1 2 3 4 5], 1);
y = sequence([5 3 1 -1 3 -2 2 3], -1);

% test plus
test_lab1('plus(x, y)')
test_lab1('plus(y, x)')
test_lab1('plus(1, x)')
test_lab1('plus(x, 1)')

y = sequence([5 3 1 0 3 -2 2 3], -4);
test_lab1('plus(x, y)')
test_lab1('plus(y, x)')

% test minustract
test_lab1('minus(x, y)')
test_lab1('minus(y, x)')
test_lab1('minus(1, x)')
test_lab1('minus(x, 1)')

% test timesiplication
test_lab1('times(x, y)')
test_lab1('times(3, x)')
test_lab1('times(x, 3)')

% test flip
test_lab1('flip(x)')

% test shift
test_lab1('shift(y, 2)')

%combinations
test_lab1('flip(minus(shift(plus(x, 2), 4), y))')
test_lab1('plus(flip(plus(x, y)), shift(y, -5))')
test_lab1('minus(plus(times(shift(flip(x), 4), shift(y, 3)), flip(y)),
 x)')

% test stem
set(clf, 'Position', [200 200 400 200])
stem(flip(2+(x-shift(y, -4).*y-3)))
title('y[n]');

% Program Listings
fprintf('\n\n')
disp('--- sequence.m --------------------------------')
type sequence
```

```
plus(x, y): sequence O.K.
plus(y, x): sequence O.K.
```

```
plus(1, x): sequence O.K.
plus(x, 1): sequence O.K.
plus(x, y): sequence O.K.
plus(y, x): sequence O.K.
minus(x, y): sequence O.K.
minus(y, x): sequence O.K.
minus(1, x): sequence O.K.
minus(x, 1): sequence O.K.
times(x, y): sequence O.K.
times(3, x): sequence O.K.
times(x, 3): sequence O.K.
flip(x): sequence O.K.
shift(y, 2): sequence O.K.
flip(minus(shift(plus(x, 2), 4), y)): sequence O.K.
plus(flip(plus(x, y)), shift(y, -5)): sequence O.K.
minus(plus(times(shift(flip(x), 4), shift(y, 3)), flip(y)), x):
 sequence O.K.


--- sequence.m --------------------------------

% Juan Angeles Acuna and Moses Martinez
classdef sequence
 properties
  data
  offset
 end

 methods
  function s = sequence(data, offset)
   s.data = data;
   s.offset = offset;
  end

  function display(s)
   var = inputname(1);
   if (isempty(var))
    disp('ans =');
   else
    disp([var '=']);
   end
   switch length(s.data)
    case 0
     disp('    data: []')
    case 1
     disp(['    data: ', num2str(s.data)])
    otherwise
     disp(['    data: [' num2str(s.data) ']'])
   end
   disp(['  offset: ' num2str(s.offset)])
  end

  function y = flip(x)
            Lin = length(x.data) + x.offset -1;
```

```
            data = fliplr(x.data);
            offset = -Lin;
            y = sequence(data, offset);
        end

    function y = shift(x, n0)
            offset = x.offset + n0;
            y = sequence(x.data, offset);
        end

        function [xdata, ydata] = seqData(x, y)
            Lx = length(x.data);
            Ly = length(y.data);
            ody= y.offset - x.offset;
            odx= x.offset - y.offset;
            xdata = [zeros(1, odx) x.data zeros(1, ody - (Lx - Ly))];
            ydata = [zeros(1, ody) y.data zeros(1, odx - (Ly - Lx))];
        end

        function z = sequenceTrimmer(x)
            while(x.data(1) == 0)
                x.data(1) = [];
                x.offset = x.offset + 1;
            end;

            while(x.data(end) == 0)
                x.data(end) = [];
            end;
            z = sequence(x.data, x.offset);
        end

    function z = plus(x, y)
            if((isa(x,'sequence')) && (isa(y,'sequence')))
                [x.data, y.data] = seqData(x,y);
                z_data = x.data + y.data;
                z_offset = min(x.offset, y.offset);
            else
                if(isa(x,'sequence'))
                    z_data = x.data + y;
                    z_offset= x.offset;
                elseif(isa(y,'sequence'))
                    z_data = y.data + x;
                    z_offset=y.offset;
                end
            end
            z = sequenceTrimmer(sequence(z_data, z_offset));
        end

    function z = minus(x, y)
            if((isa(x,'sequence')) && (isa(y,'sequence')))
                [x.data, y.data] = seqData(x,y);
                z_data = x.data - y.data;
                z_offset = min(x.offset, y.offset);
            else
```

```
                if(isa(x,'sequence'))
                    z_data = x.data - y;
                    z_offset= x.offset;
                elseif(isa(y,'sequence'))
                    z_data = x - y.data;
                    z_offset=y.offset;
                end
            end
            z = sequenceTrimmer(sequence(z_data, z_offset));
        end

    function z = times(x, y)
            if((isa(x,'sequence')) && (isa(y,'sequence')))
                [x.data, y.data] = seqData(x,y);
                z_data = x.data .* y.data;
                z_offset = min(x.offset, y.offset);
            else
                if(isa(x,'sequence'))
                    z_data = x.data * y;
                    z_offset= x.offset;
                elseif(isa(y,'sequence'))
                    z_data = x * y.data;
                    z_offset=y.offset;
                end
            end
            z = sequenceTrimmer(sequence(z_data, z_offset));
        end

    function stem(x)
            Lin = length(x.data) + x.offset -1;
            stem((x.offset:Lin), x.data);
    end
  end
end
```



Published with MATLAB® R2015a