Session 3

Programming and Running Function M-Files

The main steps of programming function M-Files are identical to those of programming script M-File. These are writing and then saving the M-File. However, a function M-File always starts by a very significant line called the "declaration" line. The declaration line takes the following format:

function [OP1, OP2, …, Opn] = functionname(IP1, IP2,…, Ipn)

Very important to note is that a function's declaration line shall NEVER end with a semicolon.

Example : write Matlab function that generate impulse signal centered at n0 at start from n1 to n2

```
function [x,n] = impseq(n0,n1,n2)

% Generates x(n) = delta(n-n0); n1 <= n,n0 <= n2

% ----------------------------------------------

% [x,n] = impseq(n0,n1,n2)

%

if ((n0 < n1) | (n0 > n2) | (n1 > n2))

                  error('arguments must satisfy n1 <= n0 <= n2')

end

n = [n1:n2];

%x = [zeros(1,(n0-n1)), 1, zeros(1,(n2-n0))];

x = [(n-n0) == 0];
```

Example: write Matlab function that generate unit step signal rise at n0 and start from n1 to n2

```
function [x,n] = stepseq(n0,n1,n2)

% Generates x(n) = u(n-n0); n1 <= n,n0 <= n2

% -----------------------------------------

% [x,n] = stepseq(n0,n1,n2)

%

if ((n0 < n1) | (n0 > n2) | (n1 > n2))

                  error('arguments must satisfy n1 <= n0 <= n2')

end

n = [n1:n2];

%x = [zeros(1,(n0-n1)), ones(1,(n2-n0+1))];

x = [(n-n0) >= 0];
```

<u>Examples</u>:

Generate and plot each of the following sequences over the indicated interval.

a. $x(n) = 2\delta(n+2) - \delta(n-4), \quad -5 \le n \le 5.$
b. $x(n) = n[u(n)-u(n-10)] + 10e^{-0.3(n-10)}[u(n-10)-u(n-20)],\ 0 \le n \le 20.$

```
>> n = [-5:5];
>> x = 2*impseq(-2,-5,5) - impseq(4,-5,5);
>> stem(n,x); title('Sequence in Problem 2.1a')
>> xlabel('n'); ylabel('x(n)');
```

```
>> n = [0:20]; x1 = n.*(stepseq(0,0,20)-stepseq(10,0,20));
>> x2 = 10*exp(-0.3*(n-10)).*(stepseq(10,0,20)-stepseq(20,0,20));
>> x = x1+x2;
>> subplot(2,2,3); stem(n,x); title('Sequence in Problem 2.1b')
>> xlabel('n'); ylabel('x(n)');
```

Example: write Matlab function that add two signals together :x1's duration is n1 and x2's duration is n2

```
function [y,n] = sigadd(x1,n1,x2,n2)
% implements y(n) = x1(n)+x2(n)
% -----------------------------
% [y,n] = sigadd(x1,n1,x2,n2)
%   y = sum sequence over n, which includes n1 and n2
%  x1 = first sequence over n1
%  x2 = second sequence over n2 (n2 can be different from n1)
%
n = min(min(n1),min(n2)):max(max(n1),max(n2));  % duration of y(n)
y1 = zeros(1,length(n)); y2 = y1;               % initialization
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;      % x1 with duration of y
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;      % x2 with duration of y
y = y1+y2;                                      % sequence addition
```

Example: write Matlab function that multiply two signals together :x1's duration is n1 and x2's duration is n2

```
function [y,n] = sigmult(x1,n1,x2,n2)
% implements y(n) = x1(n)*x2(n)
% -----------------------------
% [y,n] = sigmult(x1,n1,x2,n2)
%   y = product sequence over n, which includes n1 and n2
%  x1 = first sequence over n1
%  x2 = second sequence over n2 (n2 can be different from n1)
%
```

```
n = min(min(n1),min(n2)):max(max(n1),max(n2));  % duration of y(n)
y1 = zeros(1,length(n)); y2 = y1;               %
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;      % x1 with duration of y
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;      % x2 with duration of y
y = y1 .* y2;                                   % sequence multiplication
```

Example: write Matlab function that shift signal at k ,duration of x is m and duration of y is n

```
function [y,n] = sigshift(x,m,k)
% implements y(n) = x(n-k)
% -------------------------
% [y,n] = sigshift(x,m,k)
%
n = m+k; y = x;
```

Example: write Matlab function that fold signal x

```
function [y,n] = sigfold(x,n)
% implements y(n) = x(-n)
% -----------------------
% [y,n] = sigfold(x,n)
%
y = fliplr(x); n = -fliplr(n);
```

<u>Examples:</u>

Let $x(n) = \{1, 2, \underset{\uparrow}{3}, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1\}$. Determine and plot the following sequences.

The sequence $x(n)$ is nonzero over $-2 \le n \le 10$. Hence

```
>> n = -2:10; x = [1:7,6:-1:1];
```

a. $x_1(n) = 2x(n-5) - 3x(n+4)$.
The first part is obtained by shifting $x(n)$ by 5 and the second part by shifting $x(n)$ by -4. This shifting and the addition can be easily done using the `sigshift` and the `sigadd` functions.

```
>> [x11,n11] = sigshift(x,n,5); [x12,n12] = sigshift(x,n,-4);
>> [x1,n1] = sigadd(2*x11,n11,-3*x12,n12);
>> subplot(2,1,1); stem(n1,x1); title('Sequence in Example 2.2a')
>> xlabel('n'); ylabel('x1(n)');
```

b. $x_2(n) = x(3-n) + x(n)x(n-2)$.
The first term can be written as $x(-(n-3))$. Hence it is obtained by first folding $x(n)$ and then shifting the result by 3. The second part is a multiplication of $x(n)$ and $x(n-2)$, both of which have the same length but different

```
>> [x21,n21] = sigfold(x,n); [x21,n21] = sigshift(x21,n21,3);
>> [x22,n22] = sigshift(x,n,2); [x22,n22] = sigmult(x,n,x22,n22);
>> [x2,n2] = sigadd(x21,n21,x22,n22);
>> subplot(2,1,2); stem(n2,x2); title('Sequence in Example 2.2b')
>> xlabel('n'); ylabel('x2(n)');
```

Assignment :

1.

Let $x[n] = \left[1, -2, 4, 6, \overset{\frown}{-5}, 8, 10\right]$.where $\frown$ is the zeroth index. Generate and plot samples using stem function for the following sequences.

i. $x_1[n] = 3x[n+2] + x[n-4] + 2x[n]$.

ii. $x_2[n] = x[n+4]x[n-1] + x[2-n]x[n]$.

2.

Generate and plot the samples (use stem function) of the following sequences using MATLAB:

$0 \le n \le 25$

$$x_2[n] = n^2[u(n+5) - u(n-6)] + 10\delta[n] + 20(0.5)^n[u(n-4) - u(n-10)].$$

3.

Plot the following signal with FS=100 HZ