University of California – Santa Cruz

Electrical Engineering Department

Digital Signal Processing

Spring 2018

Lab # 1 – Issued 4/3/2018, Due 4/11/2018

INSTRUCTIONS:

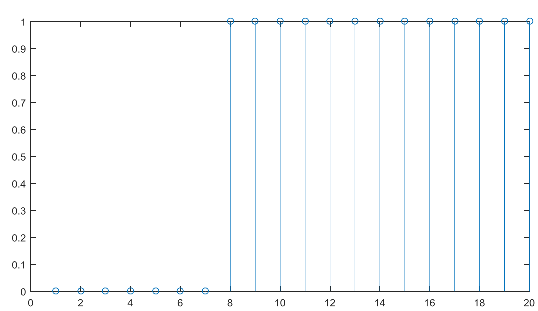
1. **Please submit your lab plots (in PDF) via Canvas**
2. **You are encouraged to discuss lab with others, including our Teaching Assistant in the Discussions Sessions**
3. **This lab problem set is worth 100 points**
4. **MATLAB functions are in Appendix and the mfiles are uploaded on Canvas.**

Exercise 1: Basic Signal Generation Function (30 points)

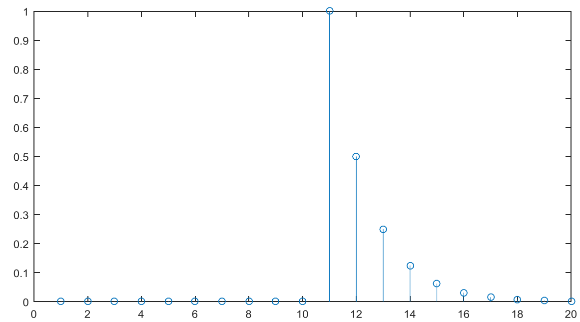
Included in this lab are several examples codes of MATLAB functions to generate various sequences: STEP, EXPONENTIAL, COSINE. Study the example codes to make sure you understand how the codes work.

For example:

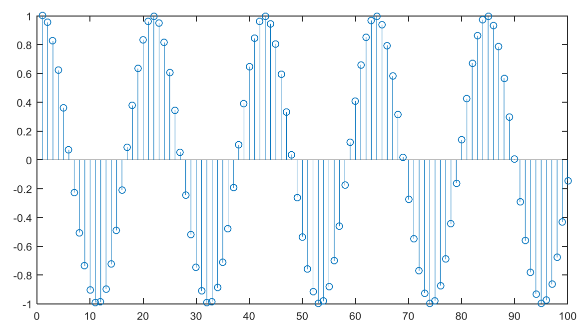
>> x = stepseq(20, 7), generated:



>> x = expseq(20, 10, 0.5), generated:



>> x = cosineseq(100, 0.3), generated:



In this lab exercise, use various parameters in the function calls for stepseq(), expseq(), and cosineseq() to demonstrate that you are able to generate these basic types of signals. You can modify the codes in any way you like.

Submit your plots and note down any observations or comments.

Exercise 2: Convolution (40 points)

Use the built-in MATLAB function: **conv(x, y)** to convolve:

a) A step sequence another step sequence;
b) A step sequence with an exponential sequence;
c) A step sequence with a cosine sequence;
d) An exponential sequence with another exponential sequence;
e) A cosine sequence with another cosine sequence (same frequency)
f) A cosine sequence with another cosine sequence (different frequencies)

Submit your plots and note down any observations or comments.

Exercise 3: Discrete-Time Fourier Transform (30 points)

Study the DTFT() function provided.

(a) Generate, x, a cosine sequence for ω = 0.4 and use the DTFT() for ω = [0 : 0.01 : 3.1426] to calucate the DTFT of x. Submit your plot for the magnitude of the X = DTFT(x, ω) versus ω.

(b) Generate, y, a exponential sequence for α = 0.5 and use the DTFT() for ω = [0 : 0.01 : 3.1426] to calculate the DTFT of y. Submit your plot for the magnitude and phase of the Y = DTFT(y, ω) versus ω.

(c) Generate, z, an exponential sequence for α = -0.5 and use the DTFT() for ω = [0 : 0.01 : 3.1426] to calculate the DTFT of z. Submit your plots for the sequence and for the magnitude and phase of the Z = DTFT(z, ω) versus ω.

APPENDIX: MATLAB FUNCTIONS

```
function x = stepseq(n, n0)
% n - length of the sequence
% n0 - how many zeros before ones

if ((n0 > n) | n0 < 0 | n < 0)
    error('Arguments must satisfy n > 0, n0 > 0, n0 < n')
end

x = [zeros(1,n0) ones(1, n-n0)];
stem(x)
```

```
function x = expseq(NN, n0, alpha)
% NN -- number of points
% how many preceding zeros
%  Generates x(n) = alpha^(n-n0)*u(n-n0)
%  -------------------------------------------
if ((n0 > NN) | n0 < 0 | NN < 0)
    error('Arguments must satisfy n > 0, n0 > 0, n0 < n')
end
for n=1:NN
x(n) = [((n-n0) > 0).*(alpha.^(n-n0-1))];
end

stem(x)
```

```
function x = cosineseq(n, omega)
% n - length of the sequence
% omega - frequency

x = cos(omega * (0:n-1));
stem(x)
```

```
function [X] = dtft(x, w)
% Computes Discrete-time Fourier Transform (DTFT)
% [X] = dtft(x,n,w)
%
% X = DTFT values computed at frequencies given by the w
% x = finite duration sequence
% w = frequency (in omega)
m = length(x);
n = 1:m;
numw = length(w);
X = zeros(1,numw);
for wi = 1:numw,
    X(wi) = sum(x.*exp(-j.*w(wi).*n));
end
stem(w, abs(X))
```