ELEC2400 Signals and Systems

Lab 1
(All Lab problems refer to the text)

Part 1: Introduction to Matlab

Material: Refer to Matlab notes

Things to learn:

- How to enter numbers, vectors, and matrices
- How to perform basic arithmetic operations (+,-,*,/, ^, .+, .-, .*, ./, .^)
- Basic mathematical functions (zeros, ones, cos, sin, exp, log, sqrt, sum,…)
- How to plot continuous-time waveforms and discrete-time waveforms
- How to solve difference equations

Exercises:

1. (20 minutes) Learn the basic operations in Matlab.
2. (10 minutes) Reproduce Figure 1.10 on p.10 in text, i.e., plot continuous-time signal
   $x(t) = \exp(-0.1t)\sin(2t/3)$ for $0<t<30$
   with increment of 0.1.
3. (10 minutes) Reproduce Figure 1.11 on p.12 in text, i.e., plot discrete-time signal
   $x[0]=1$, $x[1]=2$, $x[2]=1$, $x[3]=0$, $x[4]=-1$ and $x[n]=0$ for all other n.
   Range of n for the plot is [-2, 6].
4. (15 minutes) Learn how to produce .m files and how to solve difference equations.
   i. Study Example 2.6, i.e., solve the following difference equation numerically:

   $$y[n] - 1.5\, y[n-1] + y[n-2] = 2\, x[n-2]$$

   where $x[n] = u[n]$ (unit step function).
   ii. Recreate the recur.m file in Figure 2.8 for solving difference equations.
   iii. Use the recur.m file to reproduce Figure 2.9, which is the solution to the difference equation in Part i for n in [0, 20].

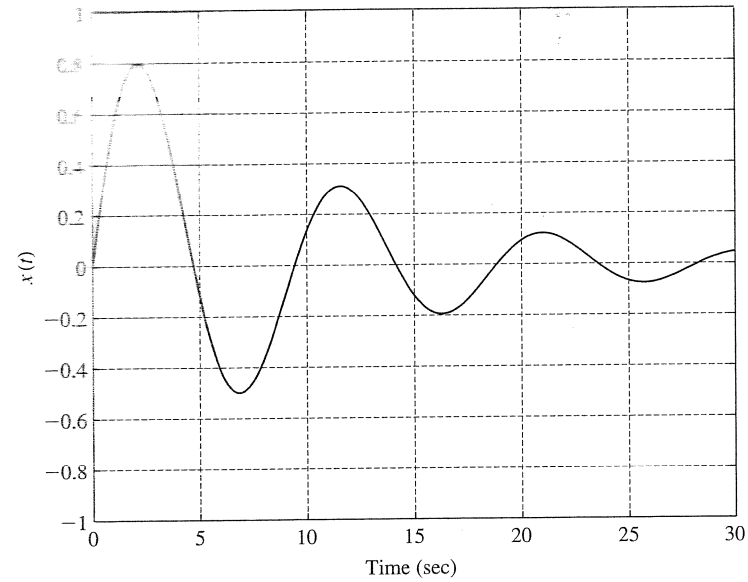

FIGURE 1.10
MATLAB plot of the signal $x(t) = e^{-0.1t} \sin\frac{2}{3}t$.

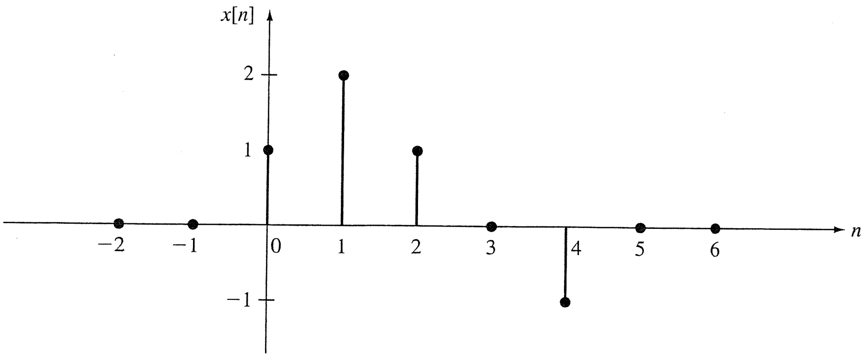

FIGURE 1.11
Stem plot of discrete-time signal.

Example 2.6 Second-Order System

Consider the discrete-time system given by the second-order input/output difference equation

$$y[n] - 1.5y[n-1] + y[n-2] = 2x[n-2] \tag{2.27}$$

Writing (2.27) in the form (2.26) results in the input/output equation

$$y[n] = 1.5y[n-1] - y[n-2] + 2x[n-2] \tag{2.28}$$

To compute the unit-pulse response $h[n]$ of the system, set $x[n] = \delta[n]$ in (2.28) with the initial conditions $y[-1]$ and $y[-2]$ both equal to zero. This gives $h[0] = 0$, $h[1] = 0$, $h[2] = 2\delta[0] = 2$, $h[3] = (1.5)h[2] = 3$, $h[4] = (1.5)h[3] - h[2] = 2.5$, and so on.

Now suppose that the input $x[n]$ is the discrete-time unit-step function $u[n]$ and that the initial output values are $y[-2] = 2$ and $y[-1] = 1$. Then, setting $n = 0$ in (2.28) gives

$$y[0] = 1.5y[-1] - y[-2] + 2x[-2]$$
$$y[0] = (1.5)(1) - 2 + (2)(0) = -0.5$$

Setting $n = 1$ in (2.28) gives

$$y[1] = 1.5y[0] - y[-1] + 2x[-1]$$
$$y[1] = (1.5)(-0.5) - 1 + 2(0) = -1.75$$

Continuing the process yields

$$\begin{aligned} y[2] &= (1.5)y[1] - y[0] + 2x[0] \\ &= (1.5)(-1.75) + 0.5 + (2)(1) = -0.125 \\ y[3] &= (1.5)y[2] - y[1] + 2x[1] \\ &= (1.5)(-0.125) + 1.75 + (2)(1) = 3.5625 \end{aligned}$$

and so on.

```
N = length(a);
M = length(b)-1;
y = [y0 zeros(1,length(n))];
x = [x0 x];
a1 = a(length(a):-1:1);     % reverses the elements in a
b1 = b(length(b):-1:1);
for i=N+1:N+length(n),
    y(i) = -a1*y(i-N:i-1)' + b1*x(i-N:i-N+M)';
end
y = y(N+1:N+length(n));
```

FIGURE 2.8
MATLAB program `recur`.

The following commands demonstrate how `recur` is used to compute the output response when $x[n] = u[n]$ for the system in Example 2.6:

```
a = [-1.5 1]; b = [0 0 2];
y0 = [2 1]; x0 = [0 0];
n = 0:20;
x = ones(1, length(n));
y = recur (a, b, n, x, x0, y0);
stem(n,y,'filled')          % produces a "stem plot"
xlabel ('n')
ylabel ('y[n]')
```

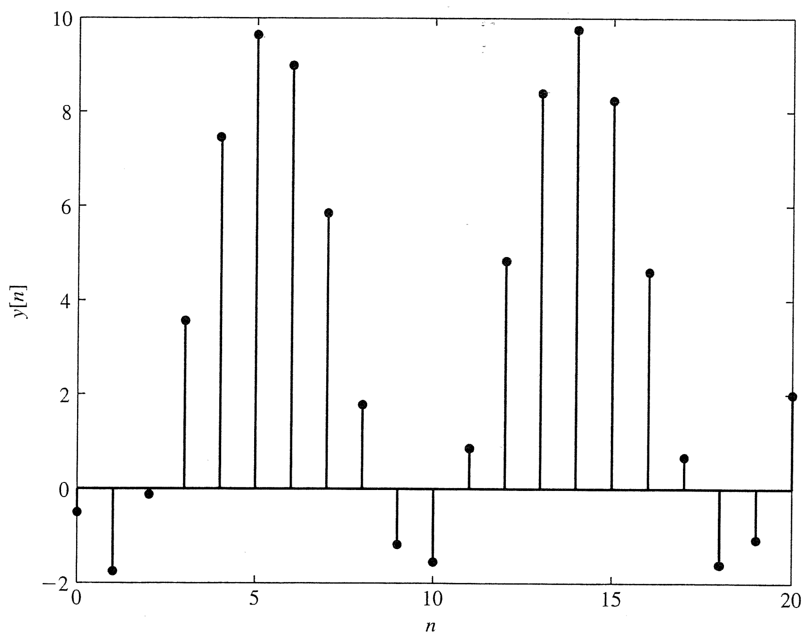

FIGURE 2.9
Plot of output response resulting from $x[n] = u[n]$ in Example 2.6.