EE_105_001 Lab 2 Report

Thomas Sullivan

TOTAL POINTS

100 / 100

QUESTION 1

1 Report **100 / 100**

Thomas Sullivan
862149154

Lab 2: Simulink Lab - Numeric Simulation

EE105 (022) - Jean-Bernard Uwineza

Pre-Lab

1. $G = 1$, $\zeta = 0.75$, $\omega_n = 4$, $\sigma = 3$, $\omega_d = 0.2$

2-3.

$$H(s) = \frac{16}{s^2+6s+16}$$

$$|H(s)|^2 = H(j\omega)\,H(-j\omega) = \frac{16}{-\omega^2+6j\omega+16} \cdot \frac{16}{-\omega^2-6j\omega+16}$$

$$= \frac{16^2}{\omega^4+4\omega^2+256}$$

$$|H(j\omega)| = \frac{16}{\sqrt{\omega^4+4\omega^2+256}}$$

$$\angle H(j\omega) = -\tan^{-1}\left(\frac{6\omega}{16-\omega^2}\right)$$

If $v(t) = \sin(0.1t)$, $\omega = 0.1$

$$|H(j0.1)| = \frac{16}{\sqrt{(0.1)^4+4(0.1)^2+256}} = 0.999 = 1$$

$$\angle H(j0.1) = -\tan^{-1}\left(\frac{0.6}{16-0.01}\right) = -0.038$$

$$\boxed{y(t) = 1\sin(0.1t - 0.038)}$$

$x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$

$$\frac{Y(s)}{U(s)} = \frac{16}{s^2 + 6s + 16}$$

$(s^2 + 6s + 16)\, Y(s) = 16\, U(s)$

$\boxed{\ddot{y} + 6\dot{y} + 16y = 16u}$, $\boxed{n = 2}$

$x_1 = y, \quad x_2 = \dot{y}$ $\qquad \ddot{y} = -6\dot{y} - 16y + 16u$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -16 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 16 \end{bmatrix} [u]$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]\,[u]$$

$$\boxed{A = \begin{bmatrix} 0 & 1 \\ -16 & -6 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 16 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad D = [0]}$$

Lab Report

Section 8.1

```
% Solution of the system of interest
% found using the lsim() function

H = tf(16, [1, 6, 16]);

A = [0, 1; -16, -6];
B = [0; 16];
C = [1, 0];
D = [0];

sys = ss(A, B, C, D);
[u, t] = gensig('sin', 10, 5, 0.01);
u = zeros(length(t), 1); % can be commented out to use u(t) = sin(.1t)

lsim(sys, u, t, [3, 0]);
```

Figure 1: MATLAB code using 'lsim()' to simulate the system of interest

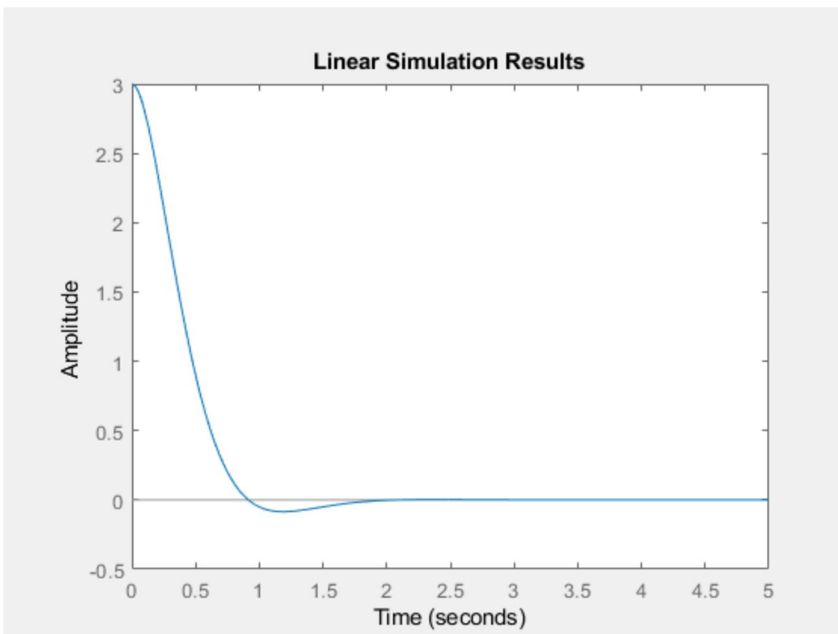

Figure 2: lsim() result for system of interest, initial condition of 3, and u(t) = 0

Section 8.2 - Figures

```
function dx = f(x, u, t)
    % State-space representation of the system
    % of interest given x and u.

    A = [0, 1; -16, -6];
    B = [0; 16];
    C = [1, 0];
    D = [0];

    dx = A*x + B*u;
end
-----------------------
function Euler(x0, T)
    % Numerically simulates the state-space system
    % of interest using recursion through Euler
    % integration approximation

    N = round(5/T) + 1;
    t = zeros(1,N);
    x = zeros(2,N);
    x(:,1) = x0;
    t = T*(0:N-1);

    tic
    for i=1:N
        dx = f(x(:,i),0,0);
        x(:,i+1) = x(:,i) + dx*T;
    end
    toc

    clf;
    plot(t(1:i)',x(1,1:i)');
    title("Simulation Using Euler Recursion (h = " + T + "s)");
    xlabel("Time, t, seconds");
    ylabel("Amplitude");
end
```

Figure 3: MATLAB code for f.m and Euler.m

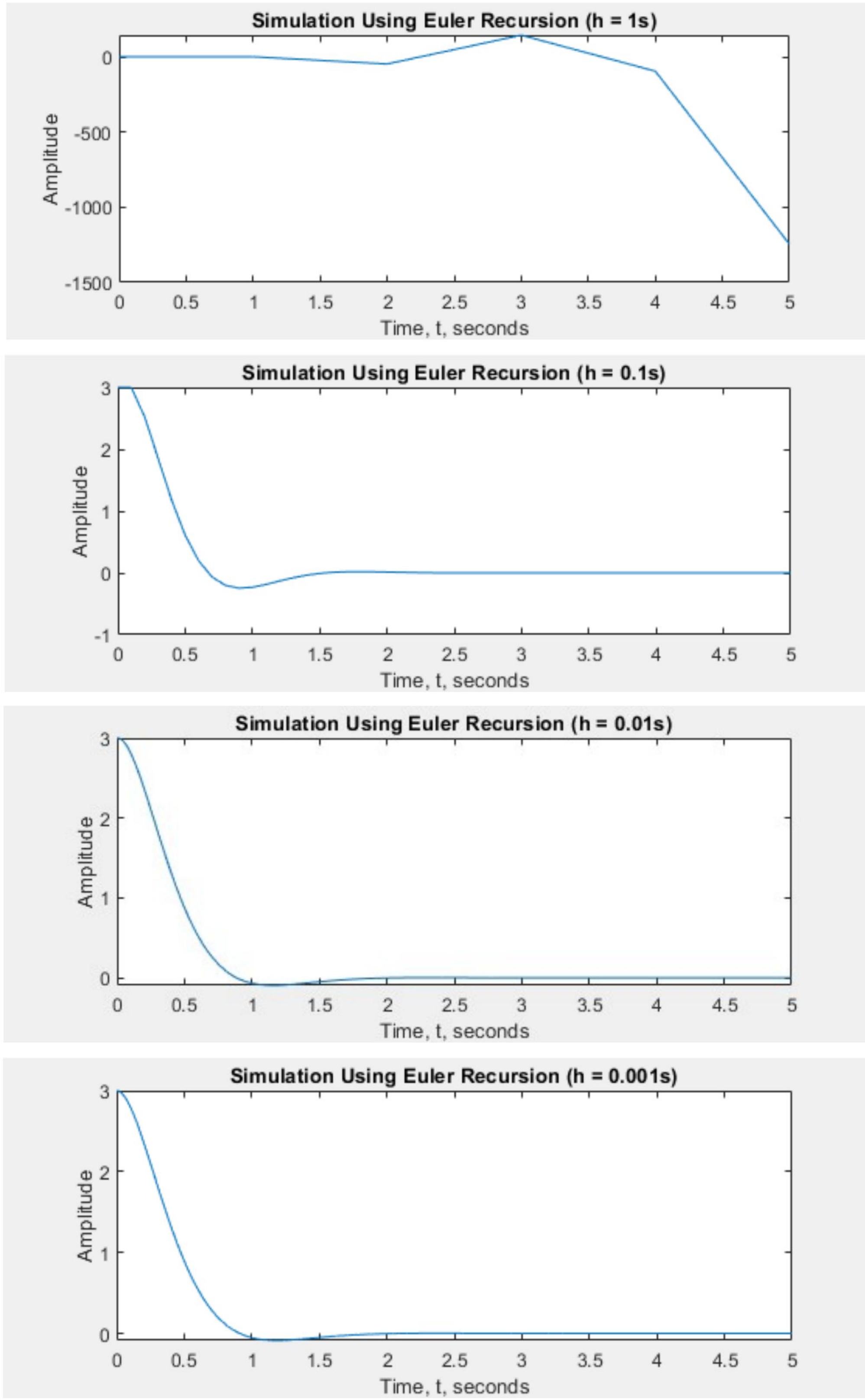

Figure 4: Results of Euler simulation with h = 1, 0.1, 0.01, 0.001

	h = 1.0 s		h = 0.1 s		h = 0.01s		h = 0.001s	
Time, t	k	x_k	k	x_k	k	x_k	k	x_k
0	0	3	0	3	0	3	0	3
1	1	3	10	-0.2323	100	-0.06787	1000	-0.05249
2	2	-45	20	0.01369	200	-0.00118	2000	-0.00281
3	3	147	30	-0.00064	300	0.00027	3000	0.00038
4	4	-93	40	0.00002	400	-0.00002	4000	-0.00003
5	5	-1245	50	0	500	0	5000	0
CPU time (ms)	0.051		0.087		0.258		2.098	

Figure 5: Table of values for results of Euler simulation

Section 8.2 - Discussion

Steps sizes were manually chosen based on powers of 10 starting from 1 (ie. 1s, 0.1s, 0.01s, etc.). Looking at Figure 2, we can assume from the lsim() simulation that the time to reach steady state is around 2 seconds. At 2 seconds, the only Euler simulations to be within 1% of steady state are when h = 0.01 and h = 0.001. This corroborates with Figure 5, which shows a very inaccurate estimation for h = 1, and a closer but still inaccurate curve for h = 0.1, when compared to Figure 2. Beyond this, there seems to be no significant benefit in estimation when comparing h = 0.01 and h = 0.001. Thus, h = 0.01 seems to be a more optimal step size, since this will save almost 1.8ms when compared to h = 0.001.

Section 8.3 - Figures

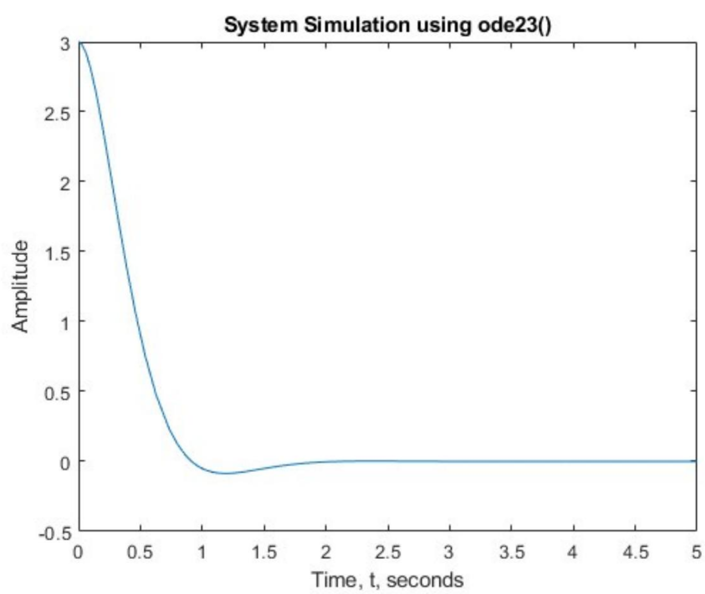

Figure 6: Output of ode23 vs. time

Section 8.3 - Discussion

The graph generated from the output of ode23 does not visually look different from Figure 2 or the simulations from the Euler approximation. The CPU time required for ode23() was 9.964ms. This is significantly larger than any of the CPU times required for the Euler approximation. However, using ode23 does not require a manual selection of a step size.

Section 8.4 - Figures

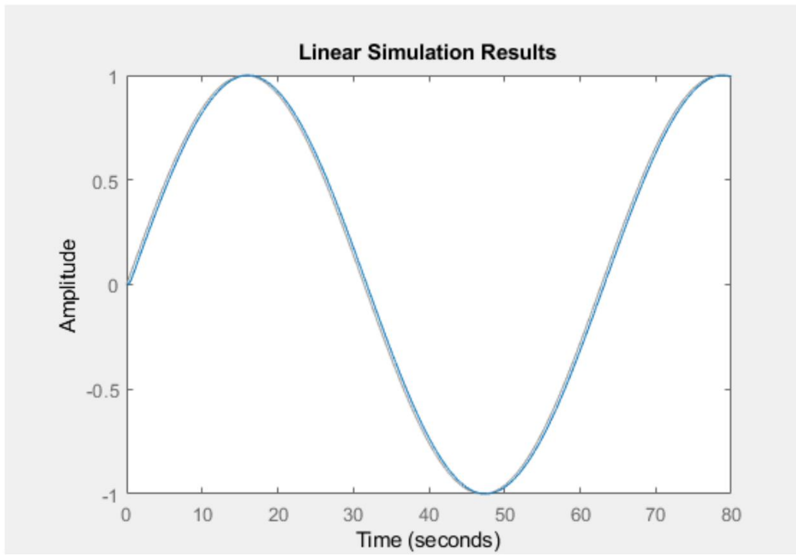

Figure 7: System simulation with input u(t) = sin(0.1t)

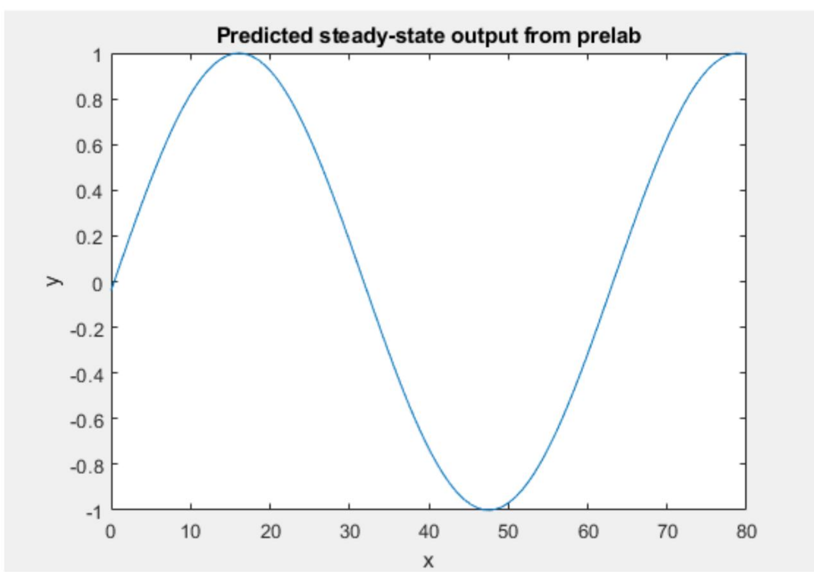

Figure 8: Pre-lab prediction of steady state output with input u(t) = sin(0.1t)

Section 8.4 - Discussion

The graphs generated from the simulation matches the predicted answer of the steady state output from the pre-lab. When comparing the output line (marked in blue) in Figure 7 to Figure 8, the graphs are identical.

Section 8.5 - Figures

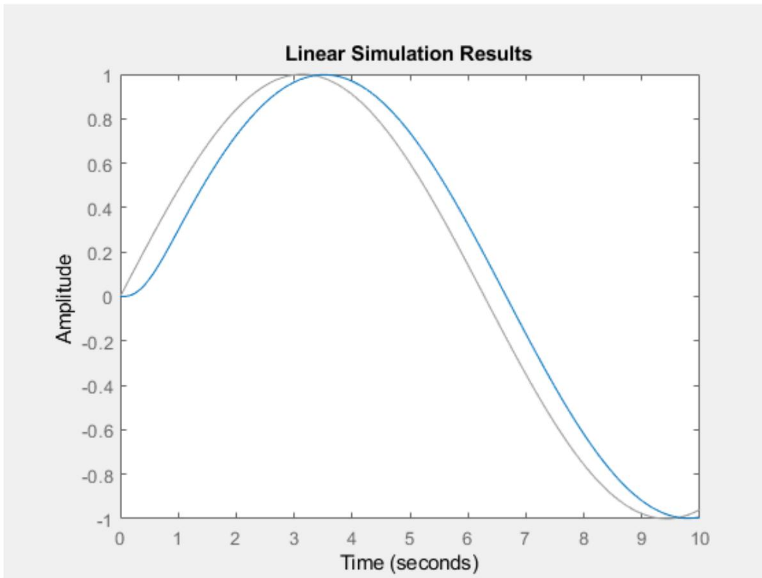

Figure 8: Simulated system for u(t) = sin(0.5t) where $0.5 < \omega_n$

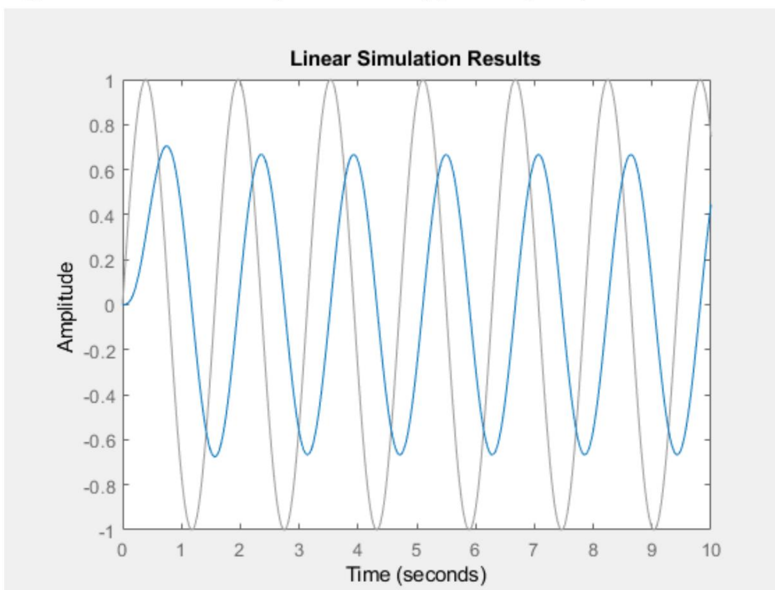

Figure 9: Simulated system for u(t) = sin(4t) where $4 = \omega_n$

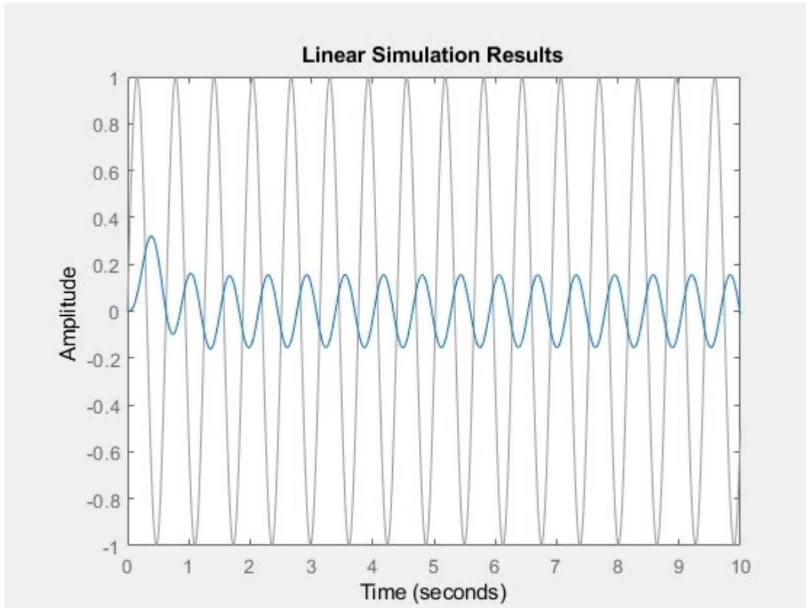

Figure 10: Simulated system for u(t) = sin(10t) where 10 > ω_n

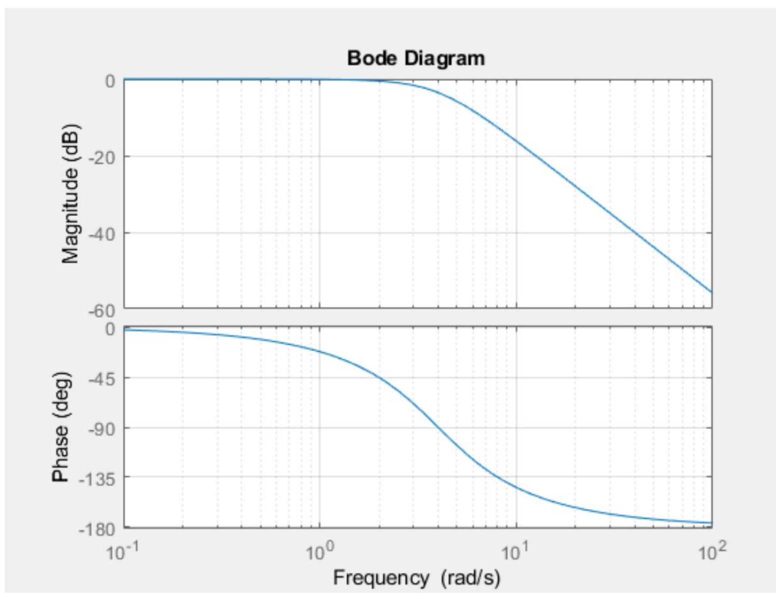

Figure 11: Bode plot for the system of interest

Section 8.5 - Discussion

As ω_n increases on the input, the output amplitude decreases significantly. For example, the output seen in Figure 8 has the same amplitude as the input, but the outputs in Figures 9 and 10 have lower amplitudes than the input, inversely proportional to the input frequency. The phase also seems to shift more depending on the value of the input frequency. The steady state values acquired here concur with the Bode plot (Figure 11) of the system. For frequency values below 1, the amplitude remains unchanged and the phase shifts slightly more. For frequency values between 1 and 10, the amplitude begins to decrease and the phase sharply shifts, which was seen from previous simulations.

1 Report 100 / 100